Stored Procedure

```
-- ==================================================
-- Package Name : Automatic Estimation
-- Package ID   :
-- Create Date  : 24/Nov/1999
-- Replace Date : xx/xxx/xxxx
-- Author       : Nakamura
--
-- File         : @c:¥jica¥AutoEst;
-- Test                                :       BEGIN AutoEst.p_AtrbtUpd(2000, 1); commit;
END;
-- Test                                : BEGIN AutoEst.p_UnitPriceUpd(1998, 4); commit; EN
D;
-- Test                                :       BEGIN AutoEst.p_GrowthRateIns(1999, 5); com
mit; END;
-- Test                                :       BEGIN AutoEst.p_SurveyDataIns(1999, 4); com
mit; END;
-- Test                                :       BEGIN AutoEst.p_AutoEstI1(2000, 1); Commit;
END;
-- Test                                :       BEGIN AutoEst.p_AutoEstI2(1999, 5); commit;
END;
-- Test                                :       BEGIN AutoEst.p_AutoEstI3(1999, 5); commit;
END;
-- Test                                :       BEGIN AutoEst.p_AutoEstI4(1999, 5); commit;
END;
-- Test                                :       BEGIN AutoEst.p_AutoEstI6(1999, 5); commit;
END;
-- Test                                :       BEGIN AutoEst.p_UnitPriceUpd(1999, 5); comm
it; END;
-- Test                                :       BEGIN AutoEst.p_BatchCtlUpd(1998, 12, '2');
commit; END;
-- Test                                :       BEGIN AutoEst.main(1999, 4, '1'); commit; EN
D;
-- ==================================================
create or replace package AutoEst is

procedure p_AtrbtUpd (
        inYear                                          in      number,
        inMonth                                         in      number);

procedure p_UnitPriceUpd (
        inYear                                          in      number,
        inMonth                                         in      number);

procedure p_GrowthRateIns (
        inYear                                          in      number,
        inMonth                                         in      number);

procedure p_SurveyDataIns (
        inYear                                          in      number,
        inMonth                                         in      number);

procedure p_AutoEstI1 (
        inYear                                          in      number,
        inMonth                                         in      number);

procedure p_AutoEstI2 (
        inYear                                          in      number,
        inMonth                                         in      number);

procedure p_AutoEstI3 (
        inYear                                          in      number,
        inMonth                                         in      number);

procedure p_AutoEstI4 (
        inYear                                          in      number,
        inMonth                                         in      number);

procedure p_AutoEstI6 (
        inYear                                          in      number,
        inMonth                                         in      number);

procedure p_BatchCtlUpd (
        inYear                                          in      number,
        inMonth                                         in      number,
        inPre_Rev_Mark                  in      varchar2);

procedure main (
```

```
        inYear                                                  in      number,
        inMonth                                                 in      number,
        inPre_Rev_Mark                  in      varchar2);

end AutoEst;
/
show errors
/

create or replace package body AutoEst is
-------------------------------------------------------------
-- Growth Rate Calculation.
-------------------------------------------------------------
function f_GRCal (
        inCurValue              in      number,
        inPreValue              in      number) return number is

        nGR                                             number;

begin
        if      inPreValue = 0
        or      inPreValue is null then
                nGR := 1;
                return nGR;
        end if;

        nGR := inCurValue / inPreValue;

        if  nGR = 0
        or      nGR is null then
                nGR := 1;
                return nGR;
        end if;

        return nGR;

end;

-------------------------------------------------------------
-- Error Log Output
-------------------------------------------------------------
procedure p_ErrLogIns(
        inISIC                                          in Varchar2,
        inYear                                          in Number,
        inMonth                                         in Number,
        inRegistrationNo        in Varchar2,
        inItemType                              in Varchar2,
        inCommodityCode         in Varchar2,
        inMessageCode                   in Varchar2) is

begin
        Insert
  into  ErrorLog (
                ISIC,
                Year,
                Month,
                RegistrationNo,
                ItemType,
                CommodityCode,
                MessageCode)
  Values (
                inISIC,
                inYear,
                inMonth,
                inRegistrationNo,
                inItemType,
                inCommodityCode,
                inMessageCode);

exception
        when dup_val_on_index then
                null;
end;

-------------------------------------------------------------
-- Common Estimation and Error Log Output
-- inCalPatern = M:Month Y:Year G:Growth Rate
```

```
-----------------------------------------------------------
procedure p_CommEst(
        inISIC                                          in      Varchar2,
        inYear                                          in      Number,
        inMonth                                         in      Number,
        inRegistrationNo        in      Varchar2,
        inItemType                              in      Varchar2,
        inCommodityCode         in      Varchar2,
        inColumnID                              in      Varchar2,
        inCalPatern                             in      Varchar2,

        ioValue                                         in      out Number,
        ioAtrbt                                         in      out Varchar2,
        inPreValue                              in      Number,
        inPreAtrbt                              in      Varchar2,
        inGrowthRate                    in      Number) is

begin
-- Attribute Checking(This Month)
        if      ioAtrbt = 'X' then
                null;
        else
                return;
        end if;

-- Attribute Checking(Pre Month or Year)
        if      inPreAtrbt = 'X' or inPreAtrbt = 'Y' or inPreAtrbt = 'M' then
                p_ErrLogIns(inISIC, inYear, inMonth, inRegistrationNo, inItemType, inCommodityC
ode,
                                                                                        'X' || inItemType ||
inColumnID);
                return;
        end if;
        if      inPreAtrbt = 'U' then
                p_ErrLogIns(inISIC, inYear, inMonth, inRegistrationNo, inItemType, inCommodityC
ode,
                                                                                        'N' || inItemType ||
inColumnID);
                return;
        end if;
        if      inPreAtrbt = 'S' then
                p_ErrLogIns(inISIC, inYear, inMonth, inRegistrationNo, inItemType, inCommodityC
ode,
                                                                                        'S' || inItemType ||
inColumnID);
                return;
        end if;
        if      inPreAtrbt = 'V' then
                null;
        else
                p_ErrLogIns(inISIC, inYear, inMonth, inRegistrationNo, inItemType, inCommodityC
ode,
                                                                                        'Z' || inItemType ||
inColumnID);
                return;
        end if;

-- Calculation
        if inCalPatern = 'M' or inCalPatern = 'Y' then
                ioValue := inPreValue;
        else
                ioValue := round(inPreValue * inGrowthRate, 3);
        end if;

-- Estimated Attribute setting.
        ioAtrbt := 'Y';

end;

-----------------------------------------------------------
-- Reset Estimated Attribute.
-----------------------------------------------------------
procedure p_AtrbtUpd (
        inYear                                          in      number,
        inMonth                                         in      number) is

-- Record Area Define
```

```
        recsd           SurveyData%rowtype;

-- Work Area
        bUpdFlg         boolean;

-- Survey Data Cursor
        Cursor  csrsd is
                select  *
                        from    SurveyData
                        where Year      = inYear
                                and     Month = inMonth
                                and ItemType in('1','2','3','4','6');

Begin
--      DBMS_OUTPUT.PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Survey Data Open Fetch
        open    csrsd;
        fetch   csrsd   into    recsd;

-- Survey Data Cursor Fetch Looping
        while csrsd%found loop

-- Reset Updated Flg
                bUpdFlg := false;

-- Item Type 1 Checking
                if      recsd.ItemType = '1' then
                        if      recsd.Atrbt_BM_Inventory        = 'Y' then
                                recsd.Atrbt_BM_Inventory        := 'X';
                                recsd.BM_Inventory      := null;
                                bUpdFlg := true;
                        end if;
                        if      recsd.Atrbt_ProductionQty = 'Y' then
                                recsd.Atrbt_ProductionQty := 'X';
                                recsd.ProductionQty := null;
                                bUpdFlg := true;
                        end if;
                        if      recsd.Atrbt_Receipts = 'Y' then
                                recsd.Atrbt_Receipts := 'X';
                                recsd.Receipts := null;
                                bUpdFlg := true;
                        end if;
                        if      recsd.Atrbt_DomesticSales = 'Y' then
                                recsd.Atrbt_DomesticSales := 'X';
                                recsd.DomesticSales := null;
                                bUpdFlg := true;
                        end if;
                        if      recsd.Atrbt_Export = 'Y' then
                                recsd.Atrbt_Export := 'X';
                                recsd.Export := null;
                                bUpdFlg := true;
                        end if;
                        if      recsd.Atrbt_OtherSales = 'Y' then
                                recsd.Atrbt_OtherSales := 'X';
                                recsd.OtherSales := null;
                                bUpdFlg := true;
                        end if;
                        if      recsd.Atrbt_ME_Inventory = 'Y' then
                                recsd.Atrbt_ME_Inventory := 'X';
                                recsd.ME_Inventory := null;
                                bUpdFlg := true;
                        end if;
                        if      bUpdFlg then
                                update  SurveyData
                                        set             BM_Inventory    = recsd.BM_Inventor
y,
                                                                ProductionQty = recsd.Prod
uctionQty,
                                                                Receipts = recsd.Receipts,
                                                                DomesticSales = recsd.Dome
sticSales,
                                                                Export = recsd.Export,
                                                                OtherSales = recsd.OtherSa
les,
                                                                ME_Inventory = recsd.ME_In
ventory,
                                                                Atrbt_BM_Inventory      = r
```

```
ecsd.Atrbt_BM_Inventory,
                                                        Atrbt_ProductionQty = recs
d.Atrbt_ProductionQty,
                                                        Atrbt_Receipts = recsd.Atr
bt_Receipts,
                                                        Atrbt_DomesticSales = recs
d.Atrbt_DomesticSales,
                                                        Atrbt_Export = recsd.Atrbt
_Export,
                                                        Atrbt_OtherSales = recsd.A
trbt_OtherSales,
                                                        Atrbt_ME_Inventory = recs
d.Atrbt_ME_Inventory,
                                                        userid = 'soi2man'
                                        where   ISIC
                = recsd.ISIC
                                                and     Year
                        = recsd.Year
                                                and     Month
                = recsd.Month
                                                and     RegistrationNo  = recsd.Registratio
nNo
                                                and     ItemType
                = recsd.ItemType
                                                and     CommodityCode   = recsd.CommodityCo
de;
                        end if;

-- Item Type 2 Checking
                elsif   recsd.ItemType = '2' then
                        if      recsd.Atrbt_ShipmentValue = 'Y' then
                                recsd.Atrbt_ShipmentValue := 'X';
                                recsd.ShipmentValue := null;
                                bUpdFlg := true;
                        end if;
                        if      bUpdFlg then
                                update  SurveyData
                                        set             ShipmentValue = recsd.ShipmentValu
e,
                                                        Atrbt_ShipmentValue = recs
d.Atrbt_ShipmentValue
                                        where   ISIC
                = recsd.ISIC
                                                and     Year
                        = recsd.Year
                                                and     Month
                = recsd.Month
                                                and     RegistrationNo  = recsd.Registratio
nNo
                                                and     ItemType
                = recsd.ItemType
                                                and     CommodityCode   = recsd.CommodityCo
de;
                        end if;

-- Item Type 3 Checking
                elsif   recsd.ItemType = '3' then
                        if      recsd.Atrbt_LaborTotal = 'Y' then
                                recsd.Atrbt_LaborTotal := 'X';
                                recsd.LaborTotal := null;
                                bUpdFlg := true;
                        end if;
                        if      recsd.Atrbt_Labor_SC = 'Y' then
                                recsd.Atrbt_Labor_SC := 'X';
                                recsd.Labor_SC := null;
                                bUpdFlg := true;
                        end if;
                        if      bUpdFlg then
                                update  SurveyData
                                        set             LaborTotal = recsd.LaborTotal,
                                                                Labor_SC = recsd.Labor_SC,
                                                                Atrbt_LaborTotal = recsd.A
trbt_LaborTotal,
                                                                Atrbt_Labor_SC = recsd.Atr
bt_Labor_SC
                                        where   ISIC
                = recsd.ISIC
```

```
                                        and     Year
                    = recsd.Year
                                        and     Month
            = recsd.Month
                                        and     RegistrationNo  = recsd.Registratio
nNo
                                        and     ItemType
            = recsd.ItemType
                                        and     CommodityCode   = recsd.CommodityCo
de;
                    end if;

-- Item Type 4 Checking
            elsif   recsd.ItemType = '4' then
                    if      recsd.Atrbt_RwMt_MEInventory = 'Y' then
                            recsd.Atrbt_RwMt_MEInventory := 'X';
                            recsd.RwMt_MEInventory := null;
                            bUpdFlg := true;
                    end if;
                    if      recsd.Atrbt_RwMt_MEInventoryValue = 'Y' then
                            recsd.Atrbt_RwMt_MEInventoryValue := 'X';
                            recsd.RwMt_MEInventoryValue := null;
                            bUpdFlg := true;
                    end if;
                    if      bUpdFlg then
                            update  SurveyData
                                    set             RwMt_MEInventory = recsd.RwMt_MEIn
ventory,
                                                            RwMt_MEInventoryValue = re
csd.RwMt_MEInventoryValue,
                                                            Atrbt_RwMt_MEInventory = r
ecsd.Atrbt_RwMt_MEInventory,
                                                            Atrbt_RwMt_MEInventoryValue
= recsd.Atrbt_RwMt_MEInventoryValue
                                    where   ISIC
            = recsd.ISIC
                                        and     Year
                    = recsd.Year
                                        and     Month
            = recsd.Month
                                        and     RegistrationNo  = recsd.Registratio
nNo
                                        and     ItemType
            = recsd.ItemType
                                        and     CommodityCode   = recsd.CommodityCo
de;
                    end if;

-- Item Type 6 Checking
            elsif   recsd.ItemType = '6' then
                    if      recsd.Atrbt_Capacity = 'Y' then
                            recsd.Atrbt_Capacity := 'X';
                            recsd.Capacity := null;
                            bUpdFlg := true;
                    end if;
                    if      bUpdFlg then
                            update  SurveyData
                                    set             Capacity = recsd.Capacity,
                                                            Atrbt_Capacity = recsd.Atr
bt_Capacity
                                    where   ISIC
            = recsd.ISIC
                                        and     Year
                    = recsd.Year
                                        and     Month
            = recsd.Month
                                        and     RegistrationNo  = recsd.Registratio
nNo
                                        and     ItemType
            = recsd.ItemType
                                        and     CommodityCode   = recsd.CommodityCo
de;
                    end if;
            end if;

-- Fetch Survey Data
            fetch   csrsd   into    recsd;
```

```
        end loop;

        close csrsd;

        commit;

end;

-------------------------------------------------------
-- Unit Price Calculation For Actual Value.
-------------------------------------------------------
procedure p_UnitPriceUpd (
        inYear                                                  in      number,
        inMonth                                                 in      number) is

-- Record Area Define
        recsd           SurveyData%rowtype;

-- Work Area
        nShipment                                       number;

-- Survey Data Cursor
        Cursor  csrsd is
                select  a.ISIC,
                                        a.Year,
                                        a.Month,
                                        a.RegistrationNo,
                                        a.CommodityCode,
                                        a.DomesticSales,
                                        a.Atrbt_DomesticSales,
                                        a.Export,
                                        a.Atrbt_Export,
                                        b.ShipmentValue,
                                        b.Atrbt_ShipmentValue
                        from    SurveyData a, SurveyData b
                        where a.ISIC = b.ISIC
                                and     a.Year = b.Year
                                and     a.Month = b.Month
                                and     a.RegistrationNo = b.RegistrationNo
                                and     a.CommodityCode = b.CommodityCode
                                and     a.ItemType = '1'
                                and     b.ItemType = '2'
                                and     a.year = inYear
                                and     a.Month = inMonth;

Begin
--      DBMS_OUTPUT.PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Survey Data Open Fetch
        open    csrsd;
        fetch   csrsd
                into    recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo, recsd.Commo
dityCode,
                                        recsd.DomesticSales, recsd.Atrbt_DomesticSales,
                                        recsd.Export, recsd.Atrbt_Export,
                                        recsd.ShipmentValue, recsd.Atrbt_ShipmentValue;

-- Survey Data Cursor Fetch Looping
        while csrsd%found loop

-- Calculation Record Checking
                if      (recsd.Atrbt_DomesticSales      in('V','M','Y'))
                and     (recsd.Atrbt_Export                     in('V','M','Y'))
                and     (recsd.Atrbt_ShipmentValue      in('V','M','Y')) then
                        nShipment := nvl(recsd.DomesticSales,0) + nvl(recsd.Export,0);
                        if      nShipment = 0 then
                                recsd.UnitPrice := null;
                        else
                                recsd.UnitPrice := recsd.ShipmentValue / nShipment;
                        end if;
                else
                        recsd.UnitPrice := null;
                end if;

-- Update Unit Price
                update  SurveyData
                        set             UnitPrice = recsd.UnitPrice
```

```
                        where   ISIC
                = recsd. ISIC
                                                and     Year
                        = recsd. Year
                                                and     Month
                = recsd. Month
                                                and     RegistrationNo  = recsd. Registratio
nNo
                                                and     ItemType
                = '1'
                                                and     CommodityCode   = recsd. CommodityCo
de;

-- Fetch Survey Data
                fetch   csrsd
                        into    recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, rec
sd. CommodityCode,
                                                recsd. DomesticSales, recsd. Atrbt_DomesticSa
les,
                                                recsd. Export, recsd. Atrbt_Export,
                                                recsd. ShipmentValue, recsd. Atrbt_ShipmentVa
lue;

        end loop;

        close csrsd;

        commit;

end;

-------------------------------------------------------
-- Growth Rate Work Making
-------------------------------------------------------
procedure p_GrowthRateIns (
        inYear                                          in      number,
        inMonth                                         in      number) is

-- Record Area Define
        recsd           SurveyData%rowtype;
        recsdp  SurveyData%rowtype;
        recge           GrowthRateForEstimation%rowtype;

-- Work Area
        nPY             number;
        nPM             number;

-- Survey Data Cursor
        Cursor  csrsd is
        select
                a. ISIC, a. ItemType, a. CommodityCode,
                sum(a. ProductionQty * decode(a. Atrbt_ProductionQty, 'V', 1, 0) * decode(b. Atr
bt_ProductionQty, 'V', 1, 0)),
                sum(b. ProductionQty *   decode(a. Atrbt_ProductionQty, 'V', 1, 0) * decode(b. A
trbt_ProductionQty, 'V', 1, 0)),
                sum(a. Receipts * decode(a. Atrbt_Receipts, 'V', 1, 0) * decode(b. Atrbt_Receipt
s, 'V', 1, 0)),
                sum(b. Receipts * decode(a. Atrbt_Receipts, 'V', 1, 0) * decode(b. Atrbt_Receipt
s, 'V', 1, 0)),
                sum(a. DomesticSales * decode(a. Atrbt_DomesticSales, 'V', 1, 0) * decode(b. Atr
bt_DomesticSales, 'V', 1, 0)),
                sum(b. DomesticSales * decode(a. Atrbt_DomesticSales, 'V', 1, 0) * decode(b. Atr
bt_DomesticSales, 'V', 1, 0)),
                sum(a. Export * decode(a. Atrbt_Export, 'V', 1, 0) * decode(b. Atrbt_Export, 'V',
1, 0)),
                sum(b. Export * decode(a. Atrbt_Export, 'V', 1, 0) * decode(b. Atrbt_Export, 'V',
1, 0)),
                sum(a. OtherSales * decode(a. Atrbt_OtherSales, 'V', 1, 0) * decode(b. Atrbt_Oth
erSales, 'V', 1, 0)),
                sum(b. OtherSales * decode(a. Atrbt_OtherSales, 'V', 1, 0) * decode(b. Atrbt_Oth
erSales, 'V', 1, 0)),
                sum(a. ShipmentValue * decode(a. Atrbt_ShipmentValue, 'V', 1, 0) * decode(b. Atr
bt_ShipmentValue, 'V', 1, 0)),
                sum(b. ShipmentValue * decode(a. Atrbt_ShipmentValue, 'V', 1, 0) * decode(b. Atr
bt_ShipmentValue, 'V', 1, 0)),
                sum(a. LaborTotal * decode(a. Atrbt_LaborTotal, 'V', 1, 0) * decode(b. Atrbt_Lab
orTotal, 'V', 1, 0)),
```

```
                sum(b.LaborTotal * decode(a.Atrbt_LaborTotal,'V',1,0) * decode(b.Atrbt_Lab
orTotal,'V',1,0)),
                sum(a.Labor_SC * decode(a.Atrbt_Labor_SC,'V',1,0) * decode(b.Atrbt_Labor_S
C,'V',1,0)),
                sum(b.Labor_SC * decode(a.Atrbt_Labor_SC,'V',1,0) * decode(b.Atrbt_Labor_S
C,'V',1,0)),
                sum(a.RwMt_MEInventory * decode(a.Atrbt_RwMt_MEInventory,'V',1,0) * decode
(b.Atrbt_RwMt_MEInventory,'V',1,0)),
                sum(b.RwMt_MEInventory * decode(a.Atrbt_RwMt_MEInventory,'V',1,0) * decode
(b.Atrbt_RwMt_MEInventory,'V',1,0)),
                sum(a.RwMt_MEInventoryValue * decode(a.Atrbt_RwMt_MEInventoryValue,'V',1,0)
* decode(b.Atrbt_RwMt_MEInventoryValue,'V',1,0)),
                sum(b.RwMt_MEInventoryValue * decode(a.Atrbt_RwMt_MEInventoryValue,'V',1,0)
* decode(b.Atrbt_RwMt_MEInventoryValue,'V',1,0))
        from    SurveyData a,SurveyData b
        where   a.ISIC                                          = b.ISIC
                and     a.RegistrationNo        =       b.RegistrationNo
                and     a.ItemType                              = b.ItemType
                and     a.CommodityCode         = b.CommodityCode
                and     a.year                                          = inYear
                and     a.Month                                         = inMonth
                and     b.Year                                          = nPY
                and     b.Month                                         = nPM
                and     a.ItemType                              in('1','2','3','4')
        group by a.ISIC,a.ItemType,a.CommodityCode;

Begin
-- Get Pre Year and Month
        if      inMonth = 1 then
                nPY     := inYear - 1;
                nPM     := 12;
        else
                nPY     := inYear;
                nPM     := inMonth - 1;
        end if;

-- Delete Growth Rate For Estimation Table
        delete  from    GrowthRateForEstimation;

-- Survey Data Open Fetch
        open    csrsd;
        fetch   csrsd
                into    recsd.ISIC,recsd.ItemType,recsd.CommodityCode,
                recsd.ProductionQty,recsdp.ProductionQty,recsd.Receipts,recsdp.Receipts,
                recsd.DomesticSales,recsdp.DomesticSales,recsd.Export,recsdp.Export,
                recsd.OtherSales,recsdp.OtherSales,
                recsd.ShipmentValue,recsdp.ShipmentValue,
                recsd.LaborTotal,recsdp.LaborTotal,recsd.Labor_SC,recsdp.Labor_SC,

                recsd.RwMt_MEInventory,recsdp.RwMt_MEInventory,
                recsd.RwMt_MEInventoryValue,recsdp.RwMt_MEInventoryValue;

-- Survey Data Cursor Fetch Looping
        while csrsd%found loop

-- Unit Price Calculation
                if      recsd.ItemType = '1' then
                        select  sum(a.UnitPrice),
                                                        sum(c.UnitPrice)
                                into    recsd.UnitPrice,recsdp.UnitPrice
                                from    SurveyData a,SurveyData b,SurveyData c,SurveyData d
                                where a.ISIC = b.ISIC
                                        and     a.RegistrationNo = b.RegistrationNo
                                        and     a.CommodityCode = b.CommodityCode
                                        and     a.year = b.Year
                                        and     a.Month = b.Month
                                        and     a.ItemType = '1'
                                        and     a.Atrbt_DomesticSales = 'V'
                                        and     a.Atrbt_Export = 'V'
                                        and     a.Atrbt_OtherSales = 'V'
                                        and     a.year = inYear
                                        and     a.Month = inMonth
                                        and     b.ItemType = '2'
                                        and     b.Atrbt_ShipmentValue = 'V'

                                        and c.ISIC = d.ISIC
                                        and     c.Year = d.Year
```

```
                                        and     c.Month = d.Month
                                        and     c.RegistrationNo = d.RegistrationNo
                                        and     c.CommodityCode = d.CommodityCode
                                        and     c.ItemType = '1'
                                        and     c.Atrbt_DomesticSales = 'V'
                                        and     c.Atrbt_Export = 'V'
                                        and     c.Atrbt_OtherSales = 'V'
                                        and     c.year = nPY
                                        and     c.Month = nPM
                                        and     d.ItemType = '2'
                                        and     d.Atrbt_ShipmentValue = 'V'

                                        and a.ISIC = c.ISIC
                                        and     a.RegistrationNo = c.RegistrationNo
                                        and     a.CommodityCode = c.CommodityCode

                                        and a.ISIC = recsd.ISIC
                                        and     a.CommodityCode = recsd.CommodityCode;
                else
                        recsd.UnitPrice := null;
                        recsdp.UnitPrice := null;
                end if;

-- Growth Rate Calculation
                recge.ProductionQty                             :=      null;
                recge.Receipts                                                  :=      nul
l;
                recge.DomesticSales                             :=      null;
                recge.Export                                                            :
=       null;
                recge.OtherSales                                                :=      nul
l;
                recge.UnitPrice                                                 :=      nul
l;
                recge.ShipmentValue                             :=      null;
                recge.LaborTotal                                                :=      nul
l;
                recge.Labor_SC                                                  :=      nul
l;
                recge.RwMt_MEInventory                  :=      null;
                recge.RwMt_MEInventoryValue     :=      null;

                if recsd.ItemType = '1' then
                        recge.ProductionQty                     :=      f_GRCal(rec
sd.ProductionQty, recsdp.ProductionQty);
                        recge.Receipts                                                  :
=       f_GRCal(recsd.Receipts, recsdp.Receipts);
                        recge.DomesticSales                                     :=      f_G
RCal(recsd.DomesticSales, recsdp.DomesticSales);
                        recge.Export
                :=      f_GRCal(recsd.Export, recsdp.Export);
                        recge.OtherSales                                                :
=       f_GRCal(recsd.OtherSales, recsdp.OtherSales);
                        recge.UnitPrice                                                 :
=       f_GRCal(recsd.UnitPrice, recsdp.UnitPrice);
                elsif recsd.ItemType = '2' then
                        recge.ShipmentValue                                     :=      f_G
RCal(recsd.ShipmentValue, recsdp.ShipmentValue);
                elsif recsd.ItemType = '3' then
                        recge.LaborTotal                                                :
=       f_GRCal(recsd.LaborTotal, recsdp.LaborTotal);
                        recge.Labor_SC                                                  :
=       f_GRCal(recsd.Labor_SC, recsdp.Labor_SC);
                elsif recsd.ItemType = '4' then
                        recge.RwMt_MEInventory                  :=      f_GRcal(recsd.RwMt_
MEInventory, recsdp.RwMt_MEInventory);
                        recge.RwMt_MEInventoryValue     :=      f_GRcal(recsd.RwMt_MEInven
toryValue, recsdp.RwMt_MEInventoryValue);
                end if;

-- Insert Growth Rate For Estimation Table
                insert
                        into    GrowthRateForEstimation(
                                ISIC,
                                ItemType,
                                CommodityCode,
                                ProductionQty,
```

```
                                Receipts,
                                DomesticSales,
                                Export,
                                OtherSales,
                                UnitPrice,
                                ShipmentValue,
                                LaborTotal,
                                Labor_SC,
                                RwMt_MEInventory,
                                RwMt_MEInventoryValue)
                Values(
                                recsd. ISIC,
                                recsd. ItemType,
                                recsd. CommodityCode,
                                recge. ProductionQty,
                                recge. Receipts,
                                recge. DomesticSales,
                                recge. Export,
                                recge. OtherSales,
                                recge. UnitPrice,
                                recge. ShipmentValue,
                                recge. LaborTotal,
                                recge. Labor_SC,
                                recge. RwMt_MEInventory,
                                recge. RwMt_MEInventoryValue);

-- Fetch Survey Data Table
                fetch   csrsd
                        into    recsd. ISIC, recsd. ItemType, recsd. CommodityCode,
                        recsd. ProductionQty, recsdp. ProductionQty, recsd. Receipts, recsdp. Rec
eipts,
                        recsd. DomesticSales, recsdp. DomesticSales, recsd. Export, recsdp. Expor
t,
                        recsd. OtherSales, recsdp. OtherSales,
                        recsd. ShipmentValue, recsdp. ShipmentValue,
                        recsd. LaborTotal, recsdp. LaborTotal, recsd. Labor_SC, recsdp. Labor_S
C,
                        recsd. RwMt_MEInventory, recsdp. RwMt_MEInventory,
                        recsd. RwMt_MEInventoryValue, recsdp. RwMt_MEInventoryValue;

        end loop;

        close csrsd;

        commit;

end;

-------------------------------------------------------
-- Insert Survey Data.
-------------------------------------------------------
procedure p_SurveyDataIns (
        inYear                                          in      number,
        inMonth                                         in      number) is

-- Record Area Define
        recem           EstablishmentMaster%rowtype;
        reccm           CommodityMaster%rowtype;
        recsdp  SurveyData%rowtype;
        recsd           SurveyData%rowtype;

-- Work Area
        nCnt    number;
        nPY             number;
        nPM             number;

-- Establishment Master Cursor
        Cursor  csrem is
                select  EstablishmentMaster. ISIC,
                                        EstablishmentMaster. RegistrationNo,
                                        EstablishmentMaster. EstimationCode,

                                        CommodityMaster. ItemType,
                                        CommodityMaster. CommodityCode
                        from    EstablishmentMaster, CommodityMaster
                        where   EstablishmentMaster. ISIC = CommodityMaster. ISIC
                                and     EstablishmentMaster. ContinuousRespondent = '1';
```

```
Begin
--      DBMS_OUTPUT.PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Establishment and Commodity Master Open Fetch
        open    csrem;
        fetch   csrem
                into    'recem.ISIC, recem.RegistrationNo, recem.EstimationCode,
                                        reccm.ItemType, reccm.CommodityCode;

-- Establishment and Commodity Cursor Fetch Looping
        while csrem%found loop

-- Survey Data Existing Checking
                select  count(*)
                        into    nCnt
                        from    SurveyData
                        where   ISIC                                    = recem.ISIC
                                and     Year                                    = i
nYear
                                and     Month                           = inMonth
                                and     RegistrationNo  = recem.RegistrationNo
                                and     ItemType                        = reccm.Ite
mType
                                and     CommodityCode   = reccm.CommodityCode;
                if      nCnt > 0 then
                        goto FetchNext;
                end if;

-- Get Pre Month or Year
                if      recem.EstimationCode in('1','2') then
                        if      inMonth = 1 then
                                nPY     := inYear - 1;
                                nPM     := 12;
                        else
                                nPY     := inYear;
                                nPM     := inMonth - 1;
                        end if;
                else
                        nPY     := inYear - 1;
                        nPM     := inMonth;
                end if;

-- Pre Survey Data Existing Checking
                select  count(*)
                        into    nCnt
                        from    SurveyData
                        where   ISIC                                    = recem.ISIC
                                and     Year                                    = n
PY
                                and     Month                           = nPM
                                and     RegistrationNo  = recem.RegistrationNo;
                if      nCnt < 1 then
                        goto FetchNext;
                end if;

-- Pre Survey Data Select
                begin
                        select  *
                                into    recsdp
                                from    SurveyData
                                where   ISIC                                    = r
ecem.ISIC
                                        and     Year
                = nPY
                                        and     Month                           = n
PM
                                        and     RegistrationNo  = recem.RegistrationNo
                                        and     ItemType                        = r
eccm.ItemType
                                        and     CommodityCode   = reccm.CommodityCode;
                exception
                        when no_data_found then
                                recsdp.Atrbt_BM_Inventory
                        := 'U';
                                recsdp.Atrbt_ProductionQty
                := 'U';
                                recsdp.Atrbt_Receipts
```

```
                        := 'U';
                        recsdp.Atrbt_DomesticSales
:= 'U';
                        recsdp.Atrbt_Export
                                := 'U';
                        recsdp.Atrbt_OtherSales
                := 'U';
                        recsdp.Atrbt_ME_Inventory
                := 'U';
                        recsdp.Atrbt_ShipmentValue
:= 'U';
                        recsdp.Atrbt_SalesPlan
                := 'U';
                        recsdp.Atrbt_LaborTotal
                := 'U';
                        recsdp.Atrbt_Labor_SC
                        := 'U';
                        recsdp.Atrbt_RwMt_MEInventory                           :=
'U';
                        recsdp.Atrbt_RwMt_MEInventoryValue      := 'U';
                        recsdp.Atrbt_Capacity
                        := 'U';
end;

if      reccm.ItemType = '1' then
        if      recsdp.Atrbt_BM_Inventory = 'U'
        or      recsdp.Atrbt_BM_Inventory is null       then
                recsd.Atrbt_BM_Inventory := 'U';
        else
                recsd.Atrbt_BM_Inventory := 'X';
        end if;
        if      recsdp.Atrbt_ProductionQty = 'U'
        or      recsdp.Atrbt_ProductionQty is null      then
                recsd.Atrbt_ProductionQty := 'U';
        else
                recsd.Atrbt_ProductionQty := 'X';
        end if;
        if      recsdp.Atrbt_Receipts = 'U'
        or      recsdp.Atrbt_Receipts is null then
                recsd.Atrbt_Receipts := 'U';
        else
                recsd.Atrbt_Receipts := 'X';
        end if;
        if      recsdp.Atrbt_DomesticSales = 'U'
        or      recsdp.Atrbt_DomesticSales is null then
                recsd.Atrbt_DomesticSales := 'U';
        else
                recsd.Atrbt_DomesticSales := 'X';
        end if;
        if      recsdp.Atrbt_Export = 'U'
        or      recsdp.Atrbt_Export is null then
                recsd.Atrbt_Export := 'U';
        else
                recsd.Atrbt_Export := 'X';
        end if;
        if      recsdp.Atrbt_OtherSales = 'U'
        or      recsdp.Atrbt_OtherSales is null then
                recsd.Atrbt_OtherSales := 'U';
        else
                recsd.Atrbt_OtherSales := 'X';
        end if;
        if      recsdp.Atrbt_ME_Inventory = 'U'
        or      recsdp.Atrbt_ME_Inventory is null then
                recsd.Atrbt_ME_Inventory := 'U';
        else
                recsd.Atrbt_ME_Inventory := 'X';
        end if;

        insert
                into    SurveyData(
                        ISIC,
                        Year,
                        Month,
                        RegistrationNo,
                        ItemType,
                        CommodityCode,
                        Atrbt_BM_Inventory,
```

```
                        Atrbt_ProductionQty,
                        Atrbt_Receipts,
                        Atrbt_DomesticSales,
                        Atrbt_Export,
                        Atrbt_OtherSales,
                        Atrbt_ME_Inventory,
                        UserID,
                        RecDate)
                Values(
                        recem.ISIC,
                        inYear,
                        inMonth,
                        recem.RegistrationNo,
                        reccm.ItemType,
                        reccm.CommodityCode,
                        recsd.Atrbt_BM_Inventory,
                        recsd.Atrbt_ProductionQty,
                        recsd.Atrbt_Receipts,
                        recsd.Atrbt_DomesticSales,
                        recsd.Atrbt_Export,
                        recsd.Atrbt_OtherSales,
                        recsd.Atrbt_ME_Inventory,
                        'AutoEstIns',
                        sysdate);

elsif   reccm.ItemType = '2' then
        if      recsdp.Atrbt_ShipmentValue = 'U'
        or      recsdp.Atrbt_ShipmentValue is null then
                recsd.Atrbt_ShipmentValue := 'U';
        else
                recsd.Atrbt_ShipmentValue := 'X';
        end if;

        insert
                into    SurveyData(
                        ISIC,
                        Year,
                        Month,
                        RegistrationNo,
                        ItemType,
                        CommodityCode,
                        Atrbt_ShipmentValue,
                        Atrbt_SalesPlan,
                        UserID,
                        RecDate)
                Values(
                        recem.ISIC,
                        inYear,
                        inMonth,
                        recem.RegistrationNo,
                        reccm.ItemType,
                        reccm.CommodityCode,
                        recsd.Atrbt_ShipmentValue,
                        'U',
                        'AutoEstIns',
                        sysdate);
elsif   reccm.ItemType = '3' then
        if      reccm.CommodityCode = '010' then
                if      recsdp.Atrbt_LaborTotal = 'U'
                or      recsdp.Atrbt_LaborTotal is null then
                        recsd.Atrbt_LaborTotal := 'U';
                else
                        recsd.Atrbt_LaborTotal := 'X';
                end if;
                if      recsdp.Atrbt_Labor_SC = 'U'
                or      recsdp.Atrbt_Labor_SC is null then
                        recsd.Atrbt_Labor_SC := 'U';
                else
                        recsd.Atrbt_Labor_SC := 'X';
                end if;

                insert
                        into    SurveyData(
                                ISIC,
                                Year,
                                Month,
                                RegistrationNo,
```

```
                                ItemType,
                                CommodityCode,
                                Atrbt_LaborTotal,
                                Atrbt_Labor_SC,
                                UserID,
                                RecDate)
                        Values(
                                recem.ISIC,
                                inYear,
                                inMonth,
                                recem.RegistrationNo,
                                reccm.ItemType,
                                reccm.CommodityCode,
                                recsd.Atrbt_LaborTotal,
                                recsd.Atrbt_Labor_SC,
                                'AutoEstIns',
                                sysdate);
        else
                if      recsdp.Atrbt_LaborTotal = 'U'
                or      recsdp.Atrbt_LaborTotal is null then
                        recsd.Atrbt_LaborTotal := 'U';
                else
                        recsd.Atrbt_LaborTotal := 'X';
                end if;

                insert
                        into    SurveyData(
                                ISIC,
                                Year,
                                Month,
                                RegistrationNo,
                                ItemType,
                                CommodityCode,
                                Atrbt_LaborTotal,
                                UserID,
                                RecDate)
                        Values(
                                recem.ISIC,
                                inYear,
                                inMonth,
                                recem.RegistrationNo,
                                reccm.ItemType,
                                reccm.CommodityCode,
                                recsd.Atrbt_LaborTotal,
                                'AutoEstIns',
                                sysdate);
        end if;
elsif   reccm.ItemType = '4' then
        if      recsdp.Atrbt_RwMt_MEInventory = 'U'
        or      recsdp.Atrbt_RwMt_MEInventory is null then
                recsd.Atrbt_RwMt_MEInventory := 'U';
        else
                recsd.Atrbt_RwMt_MEInventory := 'X';
        end if;
        if      recsdp.Atrbt_RwMt_MEInventoryValue = 'U'
        or      recsdp.Atrbt_RwMt_MEInventoryValue is null then
                recsd.Atrbt_RwMt_MEInventoryValue := 'U';
        else
                recsd.Atrbt_RwMt_MEInventoryValue := 'X';
        end if;
        insert
                into    SurveyData(
                        ISIC,
                        Year,
                        Month,
                        RegistrationNo,
                        ItemType,
                        CommodityCode,
                        Atrbt_RwMt_MEInventory,
                        Atrbt_RwMt_MEInventoryValue,
                        UserID,
                        RecDate)
                Values(
                        recem.ISIC,
                        inYear,
                        inMonth,
                        recem.RegistrationNo,
```

```
                                        reccm.ItemType,
                                        reccm.CommodityCode,
                                        recsd.Atrbt_RwMt_MEInventory,
                                        recsd.Atrbt_RwMt_MEInventoryValue,
                                        'AutoEstIns',
                                        sysdate);
                elsif   reccm.ItemType = '5' then
                        insert
                                into    SurveyData(
                                        ISIC,
                                        Year,
                                        Month,
                                        RegistrationNo,
                                        ItemType,
                                        CommodityCode,
                                        UserID,
                                        RecDate)
                                Values(
                                        recem.ISIC,
                                        inYear,
                                        inMonth,
                                        recem.RegistrationNo,
                                        reccm.ItemType,
                                        reccm.CommodityCode,
                                        'AutoEstIns',
                                        sysdate);
                elsif   reccm.ItemType = '6' then
                        if      recsdp.Atrbt_Capacity = 'U'
                        or      recsdp.Atrbt_Capacity is null then
                                recsd.Atrbt_Capacity := 'U';
                        else
                                recsd.Atrbt_Capacity := 'X';
                        end if;

                        insert
                                into    SurveyData(
                                        ISIC,
                                        Year,
                                        Month,
                                        RegistrationNo,
                                        ItemType,
                                        CommodityCode,
                                        Atrbt_Capacity,
                                        UserID,
                                        RecDate)
                                Values(
                                        recem.ISIC,
                                        inYear,
                                        inMonth,
                                        recem.RegistrationNo,
                                        reccm.ItemType,
                                        reccm.CommodityCode,
                                        recsd.Atrbt_Capacity,
                                        'AutoEstIns',
                                        sysdate);
                end if;

-- Insert Survey Header
                begin
                        insert
                                into    SurveyHeader(
                                        ISIC,
                                        Year,
                                        Month,
                                        RegistrationNo,
                                        EstimatedMark,
                                        RevisedStatus,
                                        YearlyStatus)
                                values(
                                        recem.ISIC,
                                        inYear,
                                        inMonth,
                                        recem.RegistrationNo,
                                        '2',
                                        '0',
                                        '0');
                exception
```

```
                        when dup_val_on_index then
                                null;
                end;

-- Label
                <<FetchNext>>

-- Establishment and Commodity Cursor Fetch Looping
                fetch   csrem
                        into    recem.ISIC, recem.RegistrationNo, recem.EstimationCode,
                                                reccm.ItemType, reccm.CommodityCode;

        end loop;

        close csrem;

        commit;

end;

-------------------------------------------------------------
-- Automatic Estimation For Item Type 1.
-------------------------------------------------------------
procedure p_AutoEstI1 (
        inYear                                                  in      number,
        inMonth                                                 in      number) is

-- Record Area Define
        recge           GrowthRateForEstimation%rowtype;
        recsd           SurveyData%rowtype;
        recsdm  SurveyData%rowtype;
        recsdy  SurveyData%rowtype;
        recem           EstablishmentMaster%rowtype;

-- Work Area
        nPY_Y           number;
        nPY_M           number;
        nPM_Y           number;
        nPM_M           number;

-- Survey Data Cursor(Item Type 1)
        Cursor  csrsd is
                select  SurveyData.ISIC,
                                                SurveyData.Year,
                                                SurveyData.Month,
                                                SurveyData.RegistrationNo,
                                                SurveyData.CommodityCode,
                                                SurveyData.BM_Inventory,
                                                SurveyData.Atrbt_BM_Inventory,
                                                SurveyData.ProductionQty,
                                                SurveyData.Atrbt_ProductionQty,
                                                SurveyData.Receipts,
                                                SurveyData.Atrbt_Receipts,
                                                SurveyData.DomesticSales,
                                                SurveyData.Atrbt_DomesticSales,
                                                SurveyData.Export,
                                                SurveyData.Atrbt_Export,
                                                SurveyData.OtherSales,
                                                SurveyData.Atrbt_OtherSales,
                                                SurveyData.ME_Inventory,
                                                SurveyData.Atrbt_ME_Inventory,
                                                EstablishmentMaster.EstimationCode
                        from    SurveyData, EstablishmentMaster
                        where   SurveyData.ISIC =       EstablishmentMaster.ISIC
                                and     SurveyData.RegistrationNo =     EstablishmentMaste
r.RegistrationNo
                                and     year                    = inYear
                                and     Month           = inMonth
                                and     ItemType        = '1';

Begin
-- Delete Error Message Table
        delete  from    ErrorLog
                where   Year = inYear
                        and     Month   = inMonth;

-- Get Pre Year and Month
```

```
        if      inMonth = 1 then
                nPM_Y   := inYear - 1;
                nPM_M   := 12;
        else
                nPM_Y   := inYear;
                nPM_M   := inMonth - 1;
        end if;
        nPY_Y   := inYear - 1;
        nPY_M   := inMonth;

-- Survey Data Open and Fetch(Item Type 1)
        open    csrsd;
        fetch   csrsd
                into    recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, recsd. Commo
dityCode,
                                        recsd. BM_Inventory, recsd. Atrbt_BM_Inventory,
                                        recsd. ProductionQty, recsd. Atrbt_ProductionQty,
                                        recsd. Receipts, recsd. Atrbt_Receipts,
                                        recsd. DomesticSales, recsd. Atrbt_DomesticSales,
                                        recsd. Export, recsd. Atrbt_Export,
                                        recsd. OtherSales, recsd. Atrbt_OtherSales,
                                        recsd. ME_Inventory, recsd. Atrbt_ME_Inventory,
                                        recem. EstimationCode;

-- Survey Data Looping(Item Type 1)
        while csrsd%found loop

-- Skip Data(Estimated is Nothing)
                if      recsd. Atrbt_BM_Inventory       = 'X'
                or      recsd. Atrbt_ProductionQty = 'X'
                or      recsd. Atrbt_Receipts                   =       'X'
                or      recsd. Atrbt_DomesticSales      =       'X'
                or      recsd. Atrbt_Export                             =       'X'
                or      recsd. Atrbt_OtherSales         =       'X'
                or      recsd. Atrbt_ME_Inventory       =       'X'  then
                        null;
                else
                        goto FetchNext;
                end if;

-- Check and Replace Attribute
                if      recsd. Atrbt_BM_Inventory != 'X'  then
                        p_ErrLogIns(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationN
o,'1',
                                                                                        recsd. Commo
dityCode,'R11');
                        recsd. Atrbt_BM_Inventory := 'X';
                end if;
                if      recsd. Atrbt_ProductionQty != 'X'  then
                        p_ErrLogIns(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationN
o,'1',
                                                                                        recsd. Commo
dityCode,'R12');
                        recsd. Atrbt_ProductionQty := 'X';
                end if;
                if      recsd. Atrbt_Receipts != 'X'  then
                        p_ErrLogIns(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationN
o,'1',
                                                                                        recsd. Commo
dityCode,'R13');
                        recsd. Atrbt_Receipts := 'X';
                end if;
                if      recsd. Atrbt_DomesticSales != 'X'  then
                        p_ErrLogIns(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationN
o,'1',
                                                                                        recsd. Commo
dityCode,'R14');
                        recsd. Atrbt_DomesticSales := 'X';
                end if;
                if      recsd. Atrbt_Export != 'X'  then
                        p_ErrLogIns(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationN
o,'1',
                                                                                        recsd. Commo
dityCode,'R15');
                        recsd. Atrbt_Export := 'X';
                end if;
                if      recsd. Atrbt_OtherSales != 'X'  then
```

```
                        p_ErrLogIns(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationN
o, '1',
                                                                                        recsd. Commo
dityCode, 'R16');
                        recsd. Atrbt_OtherSales := 'X';
                end if;
                if      recsd. Atrbt_ME_Inventory != 'X' then
                        p_ErrLogIns(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationN
o, '1',
                                                                                        recsd. Commo
dityCode, 'R17');
                        recsd. Atrbt_ME_Inventory := 'X';
                end if;

-- Select Pre Month Survey Data
                begin
                        select  *
                                into    recsdm
                                from    SurveyData
                                where   ISIC                                            = r
ecsd. ISIC
                                        and     Year
                = nPM_Y
                                        and     Month                                   = n
PM_M
                                        and     RegistrationNo  = recsd. RegistrationNo
                                        and     ItemType                                = '
1'
                                        and     CommodityCode   = recsd. CommodityCode;
                exception
                        when no_data_found then
                                recsdm. Atrbt_BM_Inventory      := null;
                                recsdm. Atrbt_ProductionQty     := null;
                                recsdm. Atrbt_Receipts                  := null;
                                recsdm. Atrbt_DomesticSales     := null;
                                recsdm. Atrbt_Export                            := null;
                                recsdm. Atrbt_OtherSales                := null;
                end;

-- Estimate by Establishment Code.
                if      recem. EstimationCode = '1' then
                        p_CommEst(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, '
1',
                                                                        recsd. CommodityCode, '2', 'M
',
                                                                        recsd. ProductionQty, recsd.
Atrbt_ProductionQty,
                                                                        recsdm. ProductionQty, recsd
m. Atrbt_ProductionQty, null);
                        p_CommEst(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, '
1',
                                                                        recsd. CommodityCode, '3', 'M
',
                                                                        recsd. Receipts, recsd. Atrbt
_Receipts,
                                                                        recsdm. Receipts, recsdm. Atr
bt_Receipts, null);
                        p_CommEst(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, '
1',
                                                                        recsd. CommodityCode, '4', 'M
',
                                                                        recsd. DomesticSales, recsd.
Atrbt_DomesticSales,
                                                                        recsdm. DomesticSales, recsd
m. Atrbt_DomesticSales, null);
                        p_CommEst(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, '
1',
                                                                        recsd. CommodityCode, '5', 'M
',
                                                                        recsd. Export, recsd. Atrbt_E
xport,
                                                                        recsdm. Export, recsdm. Atrbt
_Export, null);
                        p_CommEst(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, '
1',
                                                                        recsd. CommodityCode, '6', 'M
',
```

```
                                                                recsd. OtherSales, recsd. Atr
bt_OtherSales,
                                                                recsdm. OtherSales, recsdm. A
trbt_OtherSales, null);

                elsif   recem. EstimationCode = '2' then
                        begin
                                select  *
                                        into    recge
                                        from    GrowthRateForEstimation
                                        where   ISIC
                = recsd. ISIC
                                                and     ItemType
                = '1'
                                                and     CommodityCode   = recsd. CommodityCo
de;
                        exception
                                when no_data_found then
                                        recge. ProductionQty    := null;
                                        recge. Receipts                 := null;
                                        recge. DomesticSales    := null;
                                        recge. Export                           := null;
                                        recge. OtherSales               := null;
                        end;

                        p_CommEst(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, '
1',
                                                                recsd. CommodityCode, '2', 'G
',
                                                                recsd. ProductionQty, recsd.
Atrbt_ProductionQty,
                                                                recsdm. ProductionQty, recsd
m. Atrbt_ProductionQty, recge. ProductionQty);
                        p_CommEst(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, '
1',
                                                                recsd. CommodityCode, '3', 'G
',
                                                                recsd. Receipts, recsd. Atrbt
_Receipts,
                                                                recsdm. Receipts, recsdm. Atr
bt_Receipts, recge. Receipts);
                        p_CommEst(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, '
1',
                                                                recsd. CommodityCode, '4', 'G
',
                                                                recsd. DomesticSales, recsd.
Atrbt_DomesticSales,
                                                                recsdm. DomesticSales, recsd
m. Atrbt_DomesticSales, recge. DomesticSales);
                        p_CommEst(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, '
1',
                                                                recsd. CommodityCode, '5', 'G
',
                                                                recsd. Export, recsd. Atrbt_E
xport,
                                                                recsdm. Export, recsdm. Atrbt
_Export, recge. Export);
                        p_CommEst(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, '
1',
                                                                recsd. CommodityCode, '6', 'G
',
                                                                recsd. OtherSales, recsd. Atr
bt_OtherSales,
                                                                recsdm. OtherSales, recsdm. A
trbt_OtherSales, recge. OtherSales);

                elsif   recem. EstimationCode = '3' then
                        begin
                                select  *
                                        into    recsdy
                                        from    SurveyData
                                        where   ISIC
                = recsd. ISIC
                                                and     Year
                        = nPY_Y
                                                and     Month
                = nPY_M
```

```
                                                and     RegistrationNo  = recsd.Registratio
nNo
                                                and     ItemType
                = '1'
                                                and     CommodityCode   = recsd.CommodityCo
de;
                        exception
                                when no_data_found then
                                        recsdy.Atrbt_ProductionQty      := null;
                                        recsdy.Atrbt_Receipts                   := null;
                                        recsdy.Atrbt_DomesticSales      := null;
                                        recsdy.Atrbt_Export                             :=
null;
                                        recsdy.Atrbt_OtherSales                 := null;
                        end;

                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
1',
                                                                recsd.CommodityCode,'2','Y
',
                                                                recsd.ProductionQty, recsd.
Atrbt_ProductionQty,
                                                                recsdy.ProductionQty, recsd
y.Atrbt_ProductionQty, null);
                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
1',
                                                                recsd.CommodityCode,'3','Y
',
                                                                recsd.Receipts, recsd.Atrbt
_Receipts,
                                                                recsdy.Receipts, recsdy.Atr
bt_Receipts, null);
                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
1',
                                                                recsd.CommodityCode,'4','Y
',
                                                                recsd.DomesticSales, recsd.
Atrbt_DomesticSales,
                                                                recsdy.DomesticSales, recsd
y.Atrbt_DomesticSales, null);
                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
1',
                                                                recsd.CommodityCode,'5','Y
',
                                                                recsd.Export, recsd.Atrbt_E
xport,
                                                                recsdy.Export, recsdy.Atrbt
_Export, null);
                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
1',
                                                                recsd.CommodityCode,'6','Y
',
                                                                recsd.OtherSales, recsd.Atr
bt_OtherSales,
                                                                recsdy.OtherSales, recsdy.A
trbt_OtherSales, null);

                end if;

-- Calculate Others

                p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'1',
                                                        recsd.CommodityCode,'1','M',
                                                        recsd.BM_Inventory, recsd.Atrbt_BM_
Inventory,
                                                        recsdm.ME_Inventory, recsdm.Atrbt_M
E_Inventory, null);

                recsd.ME_Inventory := recsd.ProductionQty + recsd.Receipts + recsd.BM_Inve
ntory -

                        (recsd.DomesticSales + recsd.Export + recsd.OtherSales);
                recsd.Atrbt_ME_Inventory := 'Y';

                if      recsd.ME_Inventory < 0 then
                        p_ErrLogIns(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationN
o,'1',
```

```
                                                                                recsd.CommodityCod
e,'L17');
                end if;

-- Update Survey Data
                if      recsd.Atrbt_BM_Inventory        = 'X'
                or      recsd.Atrbt_ProductionQty = 'X'
                or      recsd.Atrbt_Receipts                    =       'X'
                or      recsd.Atrbt_DomesticSales       =       'X'
                or      recsd.Atrbt_Export                              =       'X'
                or      recsd.Atrbt_OtherSales          =       'X'
                or      recsd.Atrbt_ME_Inventory        =       'X' then
                        recsd.Atrbt_BM_Inventory        :=      'X';
                        recsd.Atrbt_ProductionQty :=    'X';
                        recsd.Atrbt_Receipts                    :=      'X';
                        recsd.Atrbt_DomesticSales       :=      'X';
                        recsd.Atrbt_Export                              :=      'X';
                        recsd.Atrbt_OtherSales          :=      'X';
                        recsd.Atrbt_ME_Inventory        :=      'X';
                        recsd.BM_Inventory      := null;
                        recsd.ProductionQty     := null;
                        recsd.Receipts                  := null;
                        recsd.DomesticSales     := null;
                        recsd.Export                            := null;
                        recsd.OtherSales                := null;
                        recsd.ME_Inventory      := null;
                end if;

                update  SurveyData
                        set             BM_Inventory                            = recsd.BM_
Inventory,
                                        ProductionQty                   = recsd.Pro
ductionQty,
                                        Receipts
                        = recsd.Receipts,
                                        DomesticSales                           = r
ecsd.DomesticSales,
                                        Export
                        = recsd.Export,
                                        OtherSales
                = recsd.OtherSales,
                                        ME_Inventory                            = r
ecsd.ME_Inventory,
                                        Atrbt_BM_Inventory      = recsd.Atrbt_BM_In
ventory,
                                        Atrbt_ProductionQty = recsd.Atrbt_Producti
onQty,
                                        Atrbt_Receipts                  = recsd.Atr
bt_Receipts,
                                        Atrbt_DomesticSales     = recsd.Atrbt_Domes
ticSales,
                                        Atrbt_Export                            = r
ecsd.Atrbt_Export,
                                        Atrbt_OtherSales                = recsd.Atr
bt_OtherSales,
                                        Atrbt_ME_Inventory      = recsd.Atrbt_ME_In
ventory
                        where   ISIC
                = recsd.ISIC
                                        and     Year
                        = recsd.Year
                                        and     Month
                = recsd.Month
                                        and     RegistrationNo  = recsd.Registratio
nNo
                                        and     ItemType
                = '1'
                                        and     CommodityCode   = recsd.CommodityCo
de;
-- Label
                <<FetchNext>>

-- Survey Data Fetch(Item Type 1)
                fetch   csrsd
                        into    recsd.ISIC,recsd.Year,recsd.Month,recsd.RegistrationNo,rec
sd.CommodityCode,
                                        recsd.BM_Inventory,recsd.Atrbt_BM_Inventor
```

```
y,
                                                        recsd.ProductionQty, recsd.Atrbt_Production
Qty,
                                                        recsd.Receipts, recsd.Atrbt_Receipts,
                                                        recsd.DomesticSales, recsd.Atrbt_DomesticSa
les,
                                                        recsd.Export, recsd.Atrbt_Export,
                                                        recsd.OtherSales, recsd.Atrbt_OtherSales,
                                                        recsd.ME_Inventory, recsd.Atrbt_ME_Inventor
y,
                                                        recem.EstimationCode;
        end loop;

        close csrsd;

        commit;

end;

-------------------------------------------------------
-- Automatic Estimation For Item Type 2.
-------------------------------------------------------
procedure p_AutoEstI2 (
        inYear                                                  in      number,
        inMonth                                                 in      number) is

-- Record Area Define
        recge           GrowthRateForEstimation%rowtype;
        recsd           SurveyData%rowtype;
        recsdm  SurveyData%rowtype;
        recsdy  SurveyData%rowtype;
        recsdt  SurveyData%rowtype;
        recsdu  SurveyData%rowtype;
        recem           EstablishmentMaster%rowtype;

-- Work Area
        nPY_Y           number;
        nPY_M           number;
        nPM_Y           number;
        nPM_M           number;

        nShipment       number;

-- Survey Data Cursor(Item Type 2)
        Cursor  csrsd is
                select  SurveyData.ISIC,
                                                        SurveyData.Year,
                                                        SurveyData.Month,
                                                        SurveyData.RegistrationNo,
                                                        SurveyData.CommodityCode,
                                                        SurveyData.ShipmentValue,
                                                        SurveyData.Atrbt_ShipmentValue,
                                                        EstablishmentMaster.EstimationCode
                        from    SurveyData, EstablishmentMaster
                        where   SurveyData.ISIC =       EstablishmentMaster.ISIC
                                and     SurveyData.RegistrationNo =     EstablishmentMaste
r.RegistrationNo
                                and     year                    = inYear
                                and     Month           = inMonth
                                and     ItemType        = '2';

Begin
--      DBMS_OUTPUT.PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Get Pre Year and Month
        if      inMonth = 1 then
                nPM_Y   := inYear - 1;
                nPM_M   := 12;
        else
                nPM_Y   := inYear;
                nPM_M   := inMonth - 1;
        end if;
        nPY_Y   := inYear - 1;
        nPY_M   := inMonth;

-- Survey Data Open and Fetch(Item Type 2)
        open    csrsd;
        fetch   csrsd
```

```
                into    recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo, recsd.Commo
dityCode,
                                        recsd.ShipmentValue, recsd.Atrbt_ShipmentValue,
                                        recem.EstimationCode;

-- Survey Data Looping(Item Type 2)
        while csrsd%found loop

-- Skip Data(Estimated is Nothing)
                if      recsd.Atrbt_ShipmentValue       != 'X' then
                        goto FetchNext;
                end if;

-- Estimation by Establishment Code
                if      recem.EstimationCode = '1' then
                        begin
                                select  *
                                        into    recsdm
                                        from    SurveyData
                                        where   ISIC
                = recsd.ISIC
                                                and     Year
                        = nPM_Y
                                                and     Month
                = nPM_M
                                                and     RegistrationNo  = recsd.Registratio
nNo
                                                and     ItemType
                = '2'
                                                and     CommodityCode   = recsd.CommodityCo
de;
                        exception
                                when no_data_found then
                                        recsdm.Atrbt_ShipmentValue := null;
                        end;

                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
2',
                                        recsd.CommodityCode,'1','M',
                                        recsd.ShipmentValue, recsd.Atrbt_ShipmentValue,
                                        recsdm.ShipmentValue, recsdm.Atrbt_ShipmentValue, nu
ll);

                elsif   recem.EstimationCode = '2' then
                        begin
                                select  *
                                        into    recsdm
                                        from    SurveyData
                                        where   ISIC
                = recsd.ISIC
                                                and     Year
                        = nPM_Y
                                                and     Month
                = nPM_M
                                                and     RegistrationNo  = recsd.Registratio
nNo
                                                and     ItemType
                = '2'
                                                and     CommodityCode   = recsd.CommodityCo
de;
                        exception
                                when no_data_found then
                                        recsdm.Atrbt_ShipmentValue := null;
                        end;

                        if      recsd.CommodityCode = '999' then
                                begin
                                        select  *
                                                into    recge
                                                from    GrowthRateForEstimation
                                                where   ISIC
                        = recsd.ISIC
                                                        and     ItemType
                        = '2'
                                                        and     CommodityCode   = recsd.Com
modityCode;
                                exception
```

```
                                when no_data_found then
                                        recge.ShipmentValue := null;
                        end;

                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.Registra
tionNo,'2',
                                                                        recsd.CommodityCod
e,'1','G',
                                                                        recsd.ShipmentValu
e, recsd.Atrbt_ShipmentValue,
                                                                        recsdm.ShipmentValu
e, recsdm.Atrbt_ShipmentValue, recge.ShipmentValue);

                else
                        begin
                                select  *
                                        into    recge
                                        from    GrowthRateForEstimation
                                        where   ISIC
                = recsd.ISIC
                                                and     ItemType
                = '1'
                                                and     CommodityCode   = recsd.Com
modityCode;
                        exception
                                when no_data_found then
                                        recge.UnitPrice := null;
                        end;

                        begin
                                select  *
                                        into    recsdu
                                        from    SurveyData
                                        where   ISIC
                = recsd.ISIC
                                                and     Year
                        = nPM_Y
                                                and     Month
                = nPM_M
                                                and     RegistrationNo  = recsd.Reg
istrationNo
                                                and     ItemType
                = '1'
                                                and     CommodityCode   = recsd.Com
modityCode;
                        exception
                                when no_data_found then
                                        recsdu.UnitPrice := null;
                        end;

                        begin
                                select  *
                                        into    recsdt
                                        from    SurveyData
                                        where   ISIC
                = recsd.ISIC
                                                and     Year
                        = recsd.Year
                                                and     Month
                = recsd.Month
                                                and     RegistrationNo  = recsd.Reg
istrationNo
                                                and     ItemType
                = '1'
                                                and     CommodityCode   = recsd.Com
modityCode;

                        exception
                                when no_data_found then
                                        recsdt.DomesticSales    := null;
                                        recsdt.Export                   :=
null;
                        end;

                        nShipment :=    nvl(recsdt.DomesticSales,0) + nvl(recsdt.E
xport,0);
```

```
                                p_CommEst(recsd. ISIC, recsd. Year, recsd. Month, recsd. Registra
tionNo, '2',
                                                                                recsd. CommodityCod
e, '1', 'G',
                                                                                recsd. ShipmentValu
e, recsd. Atrbt_ShipmentValue,
                                                                                recsdu. UnitPrice *
nShipment,
                                                                                recsdm. Atrbt_Shipme
ntValue, recge. UnitPrice);

                        end if;

                elsif   recem. EstimationCode = '3' then
                        begin
                                select  *
                                        into    recsdy
                                        from    SurveyData
                                        where   ISIC
                = recsd. ISIC
                                                and     Year
                        = nPY_Y
                                                and     Month
                = nPY_M
                                                and     RegistrationNo  = recsd. Registratio
nNo
                                                and     ItemType
                = '2'
                                                and     CommodityCode   = recsd. CommodityCo
de;
                        exception
                                when no_data_found then
                                        recsdy. Atrbt_ShipmentValue := null;
                        end;

                        p_CommEst(recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, '
2',
                                        recsd. CommodityCode, '1', 'Y',
                                        recsd. ShipmentValue, recsd. Atrbt_ShipmentValue,
                                        recsdy. ShipmentValue, recsdy. Atrbt_ShipmentValue, nu
ll);

                end if;

-- Update Survey Data
                if      recsd. Atrbt_ShipmentValue = 'Y' then
                        update  SurveyData
                                set             ShipmentValue                           = r
ecsd. ShipmentValue,
                                                        Atrbt_ShipmentValue     = recsd. Atr
bt_ShipmentValue
                                where   ISIC
                        = recsd. ISIC
                                                        and     Year
                                = recsd. Year
                                                        and     Month
                        = recsd. Month
                                                        and     RegistrationNo  = recsd. Reg
istrationNo
                                                        and     ItemType
                        = '2'
                                                        and     CommodityCode   = recsd. Com
modityCode;
                end if;

-- Label
                <<FetchNext>>

-- Survey Data Fetch(Item Type 2)
                fetch   csrsd
                        into    recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, rec
sd. CommodityCode,
                                                ShipmentValue, recsd. Atrbt_ShipmentVa
lue,
                                                recem. EstimationCode;
        end loop;
```

```
        close csrsd;

        commit;

end;

-------------------------------------------------------
-- Automatic Estimation For Item Type 3.
-------------------------------------------------------
procedure p_AutoEstI3 (
        inYear                                          in      number,
        inMonth                                         in      number) is

-- Record Area Define
        recge           GrowthRateForEstimation%rowtype;
        recsd           SurveyData%rowtype;
        recsdm  SurveyData%rowtype;
        recsdy  SurveyData%rowtype;
        recem           EstablishmentMaster%rowtype;

-- Work Area
        nPY_Y           number;
        nPY_M           number;
        nPM_Y           number;
        nPM_M           number;

-- Survey Data Cursor(Item Type 3)
        Cursor  csrsd is
                select  SurveyData.ISIC,
                                        SurveyData.Year,
                                        SurveyData.Month,
                                        SurveyData.RegistrationNo,
                                        SurveyData.CommodityCode,
                                        SurveyData.LaborTotal,
                                        SurveyData.Atrbt_LaborTotal,
                                        SurveyData.Labor_SC,
                                        SurveyData.Atrbt_Labor_SC,
                                        EstablishmentMaster.EstimationCode
                        from    SurveyData,EstablishmentMaster
                        where   SurveyData.ISIC =       EstablishmentMaster.ISIC
                                and     SurveyData.RegistrationNo =     EstablishmentMaste
r.RegistrationNo
                                and     year                    = inYear
                                and     Month           = inMonth
                                and     ItemType        = '3';

Begin
--      DBMS_OUTPUT.PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Get Pre Year and Month
        if      inMonth = 1 then
                nPM_Y   := inYear - 1;
                nPM_M   := 12;
        else
                nPM_Y   := inYear;
                nPM_M   := inMonth - 1;
        end if;
        nPY_Y   := inYear - 1;
        nPY_M   := inMonth;

-- Survey Data Open and Fetch(Item Type 3)
        open    csrsd;
        fetch   csrsd
                into    recsd.ISIC,recsd.Year,recsd.Month,recsd.RegistrationNo,recsd.Commo
dityCode,
                                        recsd.LaborTotal,recsd.Atrbt_LaborTotal,
                                        recsd.Labor_SC,recsd.Atrbt_Labor_SC,
                                        recem.EstimationCode;

-- Survey Data Looping(Item Type 3)
        while csrsd%found loop

-- Skip Data(Estimated is Nothing)
                if      recsd.Atrbt_LaborTotal  = 'X'
                or      recsd.Atrbt_Labor_SC            = 'X' then
                        null;
                else
                        goto FetchNext;
```

```
end if;

if      recem.EstimationCode = '1' then
        begin
                select  *
                        into    recsdm
                        from    SurveyData
                        where   ISIC
= recsd.ISIC
                                and     Year
        = nPM_Y
                                and     Month
= nPM_M
                                and     RegistrationNo  = recsd.Registratio
nNo
                                and     ItemType
= '3'
                                and     CommodityCode   = recsd.CommodityCo
de;
        exception
                when no_data_found then
                        recsdm.Atrbt_LaborTotal := null;
                        recsdm.Atrbt_Labor_SC := null;
        end;

        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
3',
                        recsd.CommodityCode,'1','M',
                        recsd.LaborTotal, recsd.Atrbt_LaborTotal,
                        recsdm.LaborTotal, recsdm.Atrbt_LaborTotal, null);

        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
3',
                        recsd.CommodityCode,'2','M',
                        recsd.Labor_SC, recsd.Atrbt_Labor_SC,
                        recsdm.Labor_SC, recsdm.Atrbt_Labor_SC, null);

elsif   recem.EstimationCode = '2' then
        begin
                select  *
                        into    recsdm
                        from    SurveyData
                        where   ISIC
= recsd.ISIC
                                and     Year
        = nPM_Y
                                and     Month
= nPM_M
                                and     RegistrationNo  = recsd.Registratio
nNo
                                and     ItemType
= '3'
                                and     CommodityCode   = recsd.CommodityCo
de;
        exception
                when no_data_found then
                        recsdm.Atrbt_LaborTotal := null;
                        recsdm.Atrbt_Labor_SC := null;
        end;

        begin
                select  *
                        into    recge
                        from    GrowthRateForEstimation
                        where   ISIC
= recsd.ISIC
                                and     ItemType
= '3'
                                and     CommodityCode   = recsd.CommodityCo
de;
        exception
                when no_data_found then
                        recge.LaborTotal := null;
                        recge.Labor_SC := null;
        end;

        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
```

```
3',
                                        recsd.CommodityCode,'1','G',
                                        recsd.LaborTotal,recsd.Atrbt_LaborTotal,
                                        recsdm.LaborTotal,recsdm.Atrbt_LaborTotal,recge.La
borTotal);

                        p_CommEst(recsd.ISIC,recsd.Year,recsd.Month,recsd.RegistrationNo,'
3',
                                        recsd.CommodityCode,'2','G',
                                        recsd.Labor_SC,recsd.Atrbt_Labor_SC,
                                        recsdm.Labor_SC,recsdm.Atrbt_Labor_SC,recge.Labor_
SC);

                elsif   recem.EstimationCode = '3' then
                        begin
                                select  *
                                        into    recsdy
                                        from    SurveyData
                                        where   ISIC
                = recsd.ISIC
                                                and     Year
                        = nPY_Y
                                                and     Month
                = nPY_M
                                                and     RegistrationNo  = recsd.Registratio
nNo
                                                and     ItemType
                = '3'
                                                and     CommodityCode   = recsd.CommodityCo
de;
                        exception
                                when no_data_found then
                                        recsdy.LaborTotal := null;
                                        recsdy.Labor_SC := null;
                        end;

                        p_CommEst(recsd.ISIC,recsd.Year,recsd.Month,recsd.RegistrationNo,'
3',
                                        recsd.CommodityCode,'1','Y',
                                        recsd.LaborTotal,recsd.Atrbt_LaborTotal,
                                        recsdy.LaborTotal,recsdy.Atrbt_LaborTotal,null);

                        p_CommEst(recsd.ISIC,recsd.Year,recsd.Month,recsd.RegistrationNo,'
3',
                                        recsd.CommodityCode,'2','Y',
                                        recsd.Labor_SC,recsd.Atrbt_Labor_SC,
                                        recsdy.Labor_SC,recsdy.Atrbt_Labor_SC,null);

                end if;

-- Update Survey Data
                if      recsd.Atrbt_LaborTotal          = 'Y'
                or      recsd.Atrbt_Labor_SC                    = 'Y' then
                        update  SurveyData
                                set             LaborTotal
                = recsd.LaborTotal,
                                                Labor_SC
                                = recsd.Labor_SC,
                                                Atrbt_LaborTotal                = r
ecsd.Atrbt_LaborTotal,
                                                Atrbt_Labor_SC                  = r
ecsd.Atrbt_Labor_SC
                                where   ISIC
                        = recsd.ISIC
                                                and     Year
                                = recsd.Year
                                                and     Month
                        = recsd.Month
                                                and     RegistrationNo  = recsd.Reg
istrationNo
                                                and     ItemType
                        = '3'
                                                and     CommodityCode   = recsd.Com
modityCode;
                end if;

-- Label
```

```
                <<FetchNext>>

-- Survey Data Fetch(Item Type 3)
                fetch   csrsd
                        into    recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, rec
sd. CommodityCode,
                                                recsd. LaborTotal, recsd. Atrbt_LaborTotal,
                                                recsd. Labor_SC, recsd. Atrbt_Labor_SC,
                                                recem. EstimationCode;
        end loop;

        close csrsd;

        commit;

end;

-----------------------------------------------------------
-- Automatic Estimation For Item Type 4.
-----------------------------------------------------------
procedure p_AutoEstI4 (
        inYear                                                  in      number,
        inMonth                                                 in      number) is

-- Record Area Define
        recge           GrowthRateForEstimation%rowtype;
        recsd           SurveyData%rowtype;
        recsdm  SurveyData%rowtype;
        recsdy  SurveyData%rowtype;
        recem           EstablishmentMaster%rowtype;

-- Work Area
        nPY_Y           number;
        nPY_M           number;
        nPM_Y           number;
        nPM_M           number;

-- Survey Data Cursor(Item Type 4)
        Cursor  csrsd is
                select  SurveyData. ISIC,
                                                SurveyData. Year,
                                                SurveyData. Month,
                                                SurveyData. RegistrationNo,
                                                SurveyData. CommodityCode,
                                                SurveyData. RwMt_MEInventory,
                                                SurveyData. Atrbt_RwMt_MEInventory,
                                                SurveyData. RwMt_MEInventoryValue,
                                                SurveyData. Atrbt_RwMt_MEInventoryValue,
                                                EstablishmentMaster. EstimationCode
                        from    SurveyData, EstablishmentMaster
                        where   SurveyData. ISIC =      EstablishmentMaster. ISIC
                                and     SurveyData. RegistrationNo =    EstablishmentMaste
r. RegistrationNo
                                and     year                    = inYear
                                and     Month           = inMonth
                                and     ItemType        = '4';

Begin
--      DBMS_OUTPUT. PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Get Pre Year and Month
        if      inMonth = 1 then
                nPM_Y   := inYear - 1;
                nPM_M   := 12;
        else
                nPM_Y   := inYear;
                nPM_M   := inMonth - 1;
        end if;
        nPY_Y   := inYear - 1;
        nPY_M   := inMonth;

-- Survey Data Open and Fetch(Item Type 4)
        open    csrsd;
        fetch   csrsd
                into    recsd. ISIC, recsd. Year, recsd. Month, recsd. RegistrationNo, recsd. Commo
dityCode,
                                        recsd. RwMt_MEInventory, recsd. Atrbt_RwMt_MEInventor
y,
```

```
                                        recsd.RwMt_MEInventoryValue, recsd.Atrbt_RwMt_MEInv
entoryValue,
                                        recem.EstimationCode;

-- Survey Data Looping(Item Type 4)
        while csrsd%found loop

-- Skip Data(Estimated is Nothing)
                if      recsd.Atrbt_RwMt_MEInventory    = 'X'
                or      recsd.Atrbt_RwMt_MEInventoryValue       = 'X' then
                        null;
                else
                        goto FetchNext;
                end if;

                if      recem.EstimationCode = '1' then
                        begin
                                select  *
                                        into    recsdm
                                        from    SurveyData
                                        where   ISIC
                = recsd.ISIC
                                                and     Year
                        = nPM_Y
                                                and     Month
                = nPM_M
                                                and     RegistrationNo  = recsd.Registratio
nNo
                                                and     ItemType
                = '4'
                                                and     CommodityCode   = recsd.CommodityCo
de;
                        exception
                                when no_data_found then
                                        recsdm.Atrbt_RwMt_MEInventory := null;
                                        recsdm.Atrbt_RwMt_MEInventoryValue := null;
                        end;

                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
4',
                                        recsd.CommodityCode,'1','M',
                                        recsd.RwMt_MEInventory, recsd.Atrbt_RwMt_MEInventor
y,
                                        recsdm.RwMt_MEInventory, recsdm.Atrbt_RwMt_MEInvent
ory, null);

                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
4',
                                        recsd.CommodityCode,'2','M',
                                        recsd.RwMt_MEInventoryValue, recsd.Atrbt_RwMt_MEInv
entoryValue,
                                        recsdm.RwMt_MEInventoryValue, recsdm.Atrbt_RwMt_MEI
nventoryValue, null);

                elsif   recem.EstimationCode = '2' then
                        begin
                                select  *
                                        into    recsdm
                                        from    SurveyData
                                        where   ISIC
                = recsd.ISIC
                                                and     Year
                        = nPM_Y
                                                and     Month
                = nPM_M
                                                and     RegistrationNo  = recsd.Registratio
nNo
                                                and     ItemType
                = '4'
                                                and     CommodityCode   = recsd.CommodityCo
de;
                        exception
                                when no_data_found then
                                        recsdm.Atrbt_RwMt_MEInventory := null;
                                        recsdm.Atrbt_RwMt_MEInventoryValue := null;
                        end;
```

```
                        begin
                                select  *
                                        into    recge
                                        from    GrowthRateForEstimation
                                        where   ISIC
                = recsd.ISIC
                                                and     ItemType
                = '4'
                                                and     CommodityCode   = recsd.CommodityCo
de;
                        exception
                                when no_data_found then
                                        recge.RwMt_MEInventory := null;
                                        recge.RwMt_MEInventoryValue := null;
                        end;

                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
4',
                                        recsd.CommodityCode,'1','G',
                                        recsd.RwMt_MEInventory, recsd.Atrbt_RwMt_MEInventor
y,
                                        recsdm.RwMt_MEInventory, recsdm.Atrbt_RwMt_MEInvent
ory, recge.RwMt_MEInventory);

                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
4',
                                        recsd.CommodityCode,'2','G',
                                        recsd.RwMt_MEInventoryValue, recsd.Atrbt_RwMt_MEInv
entoryValue,
                                        recsdm.RwMt_MEInventoryValue, recsdm.Atrbt_RwMt_MEI
nventoryValue,
                                        recge.RwMt_MEInventoryValue);

                elsif   recem.EstimationCode = '3' then
                        begin
                                select  *
                                        into    recsdy
                                        from    SurveyData
                                        where   ISIC
                = recsd.ISIC
                                                and     Year
                        = nPY_Y
                                                and     Month
                = nPY_M
                                                and     RegistrationNo  = recsd.Registratio
nNo
                                                and     ItemType
                = '4'
                                                and     CommodityCode   = recsd.CommodityCo
de;
                        exception
                                when no_data_found then
                                        recsdy.Atrbt_RwMt_MEInventory := null;
                                        recsdy.Atrbt_RwMt_MEInventoryValue := null;
                        end;

                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
4',
                                        recsd.CommodityCode,'1','Y',
                                        recsd.RwMt_MEInventory, recsd.Atrbt_RwMt_MEInventor
y,
                                        recsdy.RwMt_MEInventory, recsdy.Atrbt_RwMt_MEInvent
ory, null);

                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
4',
                                        recsd.CommodityCode,'2','Y',
                                        recsd.RwMt_MEInventoryValue, recsd.Atrbt_RwMt_MEInv
entoryValue,
                                        recsdy.RwMt_MEInventoryValue, recsdy.Atrbt_RwMt_MEI
nventoryValue, null);

                end if;

-- Update Survey Data
                if      recsd.Atrbt_RwMt_MEInventory    = 'Y'
                or      recsd.Atrbt_RwMt_MEInventoryValue       = 'Y' then
```

```
                        update  SurveyData
                                set             RwMt_MEInventory
                                = recsd.RwMt_MEInventory,
                                                RwMt_MEInventoryValue
                = recsd.RwMt_MEInventoryValue,
                                                Atrbt_RwMt_MEInventory
                = recsd.Atrbt_RwMt_MEInventory,
                                                Atrbt_RwMt_MEInventoryValue = recs
d.Atrbt_RwMt_MEInventoryValue
                                where   ISIC
                        = recsd.ISIC
                                                and     Year
                                = recsd.Year
                                                and     Month
                        = recsd.Month
                                                and     RegistrationNo  = recsd.Reg
istrationNo
                                                and     ItemType
                        = '4'
                                                and     CommodityCode   = recsd.Com
modityCode;

                end if;
-- Label
                <<FetchNext>>

-- Survey Data Fetch(Item Type 4)
        fetch   csrsd
                into    recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo, recsd.Commo
dityCode,
                                        recsd.RwMt_MEInventory, recsd.Atrbt_RwMt_MEInventor
y,
                                        recsd.RwMt_MEInventoryValue, recsd.Atrbt_RwMt_MEInv
entoryValue,
                                        recem.EstimationCode;

        end loop;

        close csrsd;

        commit;

end;

-------------------------------------------------------
-- Automatic Estimation For Item Type 6.
-------------------------------------------------------
procedure p_AutoEstI6 (
        inYear                                          in      number,
        inMonth                                         in      number) is

-- Record Area Define
        recge           GrowthRateForEstimation%rowtype;
        recsd           SurveyData%rowtype;
        recsdm  SurveyData%rowtype;
        recsdy  SurveyData%rowtype;
        recem           EstablishmentMaster%rowtype;

-- Work Area
        nPY_Y           number;
        nPY_M           number;
        nPM_Y           number;
        nPM_M           number;

-- Survey Data Cursor(Item Type 6)
        Cursor  csrsd is
                select  SurveyData.ISIC,
                                        SurveyData.Year,
                                        SurveyData.Month,
                                        SurveyData.RegistrationNo,
                                        SurveyData.CommodityCode,
                                        SurveyData.Capacity,
                                        SurveyData.Atrbt_Capacity,
                                        EstablishmentMaster.EstimationCode
                        from    SurveyData, EstablishmentMaster
                        where   SurveyData.ISIC =       EstablishmentMaster.ISIC
                                and     SurveyData.RegistrationNo =     EstablishmentMaste
```

```
r.RegistrationNo
                                and     year                    = inYear
                                and     Month           = inMonth
                                and     ItemType        = '6';

Begin
--      DBMS_OUTPUT.PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Get Pre Year and Month
        if      inMonth = 1 then
                nPM_Y   := inYear - 1;
                nPM_M   := 12;
        else
                nPM_Y   := inYear;
                nPM_M   := inMonth - 1;
        end if;
        nPY_Y   := inYear - 1;
        nPY_M   := inMonth;

-- Survey Data Open and Fetch(Item Type 6)
        open    csrsd;
        fetch   csrsd
                into    recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo, recsd.Commo
dityCode,
                                        recsd.Capacity, recsd.Atrbt_Capacity,
                                        recem.EstimationCode;

-- Survey Data Looping(Item Type 6)
        while csrsd%found loop

-- Skip Data(Estimated is Nothing)
                if      recsd.Atrbt_Capacity    != 'X' then
                        goto FetchNext;
                end if;

                if      recem.EstimationCode = '1' or recem.EstimationCode = '2' then
                        begin
                                select  *
                                        into    recsdm
                                        from    SurveyData
                                        where   ISIC
                = recsd.ISIC
                                                and     Year
                        = nPM_Y
                                                and     Month
                = nPM_M
                                                and     RegistrationNo  = recsd.Registratio
nNo
                                                and     ItemType
                = '6'
                                                and     CommodityCode   = recsd.CommodityCo
de;

                        exception
                                when no_data_found then
                                        recsdm.Atrbt_Capacity := null;
                        end;

                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo,'
6',
                                        recsd.CommodityCode,'1','M',
                                        recsd.Capacity, recsd.Atrbt_Capacity,
                                        recsdm.Capacity, recsdm.Atrbt_Capacity, null);

                elsif   recem.EstimationCode = '3' then
                        begin
                                select  *
                                        into    recsdy
                                        from    SurveyData
                                        where   ISIC
                = recsd.ISIC
                                                and     Year
                        = nPY_Y
                                                and     Month
                = nPY_M
                                                and     RegistrationNo  = recsd.Registratio
nNo
                                                and     ItemType
```

```
                = '6'
                                                and     CommodityCode   = recsd.CommodityCo
de;
                        exception
                                when no_data_found then
                                        recsdy.Atrbt_Capacity := null;
                        end;

                        p_CommEst(recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo, '
6',
                                        recsd.CommodityCode, '1', 'Y',
                                        recsd.Capacity, recsd.Atrbt_Capacity,
                                        recsdy.Capacity, recsdy.Atrbt_Capacity, null);

                end if;

-- Update Survey Data
                if      recsd.Atrbt_Capacity = 'Y' then
                        update  SurveyData
                                set             Capacity                                = r
ecsd.Capacity,
                                                        Atrbt_Capacity  = recsd.Atrbt_Capac
ity
                                where   ISIC                                            = r
ecsd.ISIC
                                        and     Year
                = recsd.Year
                                        and     Month                                   = r
ecsd.Month
                                        and     RegistrationNo  = recsd.RegistrationNo
                                        and     ItemType                                = '
6'
                                        and     CommodityCode   = recsd.CommodityCode;
                end if;

-- Label
                <<FetchNext>>

-- Survey Data Fetch(Item Type 6)
                fetch   csrsd
                        into    recsd.ISIC, recsd.Year, recsd.Month, recsd.RegistrationNo, rec
sd.CommodityCode,
                                        recsd.Capacity, recsd.Atrbt_Capacity,
                                        recem.EstimationCode;

        end loop;

        close csrsd;

        commit;

end;

-------------------------------------------------------
-- Batch Control File Update.
-------------------------------------------------------
procedure p_BatchCtlUpd (
        inYear                                          in      number,
        inMonth                                         in      number,
        inPre_Rev_Mark                  in      varchar2) is

Begin
        if      inPre_Rev_Mark = '1' then
                update  BatchControl
                        set             Pre_EstimatedDate       = sysdate
                        where   Year                                            = i
nYear
                                and     Month                                   = i
nMonth;
        elsif   inPre_Rev_Mark = '2' then
                update  BatchControl
                        set             Rev_EstimatedDate       = sysdate
                        where   Year                                            = i
nYear
                                and     Month                                   = i
nMonth;
        end if;
```

```
end;

-------------------------------------------------------
-- Automatic Estimation Main.
-------------------------------------------------------
procedure Main (
        inYear                                          in      number,
        inMonth                                         in      number,
        inPre_Rev_Mark                  in      varchar2) is

Begin
        p_AtrbtUpd(inYear, inMonth);
        commit;
        p_UnitPriceUpd(inYear, inMonth);
        commit;
        p_GrowthRateIns(inYear, inMonth);
        commit;
        p_SurveyDataIns(inYear, inMonth);
        commit;
        p_AutoEstI1(inYear, inMonth);
        Commit;
        p_AutoEstI2(inYear, inMonth);
        Commit;
        p_AutoEstI3(inYear, inMonth);
        Commit;
        p_AutoEstI4(inYear, inMonth);
        Commit;
        p_AutoEstI6(inYear, inMonth);
        Commit;
        p_UnitPriceUpd(inYear, inMonth);
        commit;
        p_BatchCtlUpd(inYear, inMonth, inPre_Rev_Mark);
        commit;

end;

end;
/
show errors
/
```

```
-- ==================================================
-- Package Name : Balance Comparison Data Making
-- Package ID   : BalComMk
-- Create Date  : 29/Oct/1999
-- Replace Date : xx/xxx/xxxx
-- Author       : Nakamura
--
-- File         : @c:¥jica¥BalComMk;
-- Test                                : BEGIN BalComMk.p_BalCom(2,'0',1999,3,'1'); END;
-- Test                                :       delete from BalanceComparison;
-- Test                                : Update BalanceComparison set seq = 3 where seq =
2;
-- ==================================================
create or replace package BalComMk is
procedure p_BalCom (
        inSeq                                                   in      number,
        inComparisonTarget      in      varchar2,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2);

end BalComMk;
/
show errors
/

create or replace package body BalComMk is
-------------------------------------------------------
-- Insert Balance Comparison Report Table
-------------------------------------------------------
procedure p_InsBalCom (
        inSeq                                                                   i
n       number,
        inIndexType                                             in      varchar2,
        inComparisonTarget                      in      varchar2,
        inYear                                                          in      num
ber,
        inMonth                                                         in      num
ber,
        inISIC                                                          in      var
char2,
        inIndustryTypeShortName in      varchar2,
        inIndexPMY                                              in      number,
        inIndexPre                                              in      number,
        inIndexRev                                              in      number,
        inIndexBalance                          in      number,
        inGR_Pre                                                        in      num
ber,
        inGR_Rev                                                        in      num
ber,
        inCD_Pre                                                        in      num
ber,
        inCD_Rev                                                        in      num
ber,
        inCD_Balance                                    in      number,
        inSumIndexPMY                                   in      number,
        inSumIndexPre                                   in      number,
        inSumIndexRev                                   in      number,
        inSumIndexBalance                       in      number,
        inSumGR_Pre                                             in      number,
        inSumGR_Rev                                             in      number,
        inSumCD_Pre                                             in      number,
        inSumCD_Rev                                             in      number,
        inSumCD_Balance                         in      number) is

begin
        Insert
        into            BalanceComparison (
                                        Seq,
                                        IndexType,
                                        ComparisonTarget,
                                        Year,
                                        Month,
                                        ISIC,
                                        IndustryTypeShortName,
                                        IndexPMY,
                                        IndexPre,
```

```
                                        IndexRev,
                                        IndexBalance,
                                        GR_Pre,
                                        GR_Rev,
                                        CD_Pre,
                                        CD_Rev,
                                        CD_Balance,
                                        SumIndexPMY,
                                        SumIndexPre,
                                        SumIndexRev,
                                        SumIndexBalance,
                                        SumGR_Pre,
                                        SumGR_Rev,
                                        SumCD_Pre,
                                        SumCD_Rev,
                                        SumCD_Balance)
        Values (inSeq,
                                        inIndexType,
                                        inComparisonTarget,
                                        inYear,
                                        inMonth,
                                        inISIC,
                                        inIndustryTypeShortName,
                                        inIndexPMY,
                                        inIndexPre,
                                        inIndexRev,
                                        inIndexBalance,
                                        inGR_Pre,
                                        inGR_Rev,
                                        inCD_Pre,
                                        inCD_Rev,
                                        inCD_Balance,
                                        inSumIndexPMY,
                                        inSumIndexPre,
                                        inSumIndexRev,
                                        inSumIndexBalance,
                                        inSumGR_Pre,
                                        inSumGR_Rev,
                                        inSumCD_Pre,
                                        inSumCD_Rev,
                                        inSumCD_Balance);
end;

-------------------------------------------------------
-- Balance Comparison Report Data Get
-------------------------------------------------------
procedure p_BalCom (
        inSeq                                                   in      number,
        inComparisonTarget      in      varchar2,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2) is

-- Record Area Define
        recidp  Indices%rowtype;
        recidr  Indices%rowtype;
        recgrp  growthRate%rowtype;
        recgrr  growthRate%rowtype;
        recclp  ContributionForAll1%rowtype;
        recclr  ContributionForAll1%rowtype;
        recam           AggregationMaster%rowtype;
        recid           Indices%rowtype;

        recidps Indices%rowtype;
        recidrs Indices%rowtype;
        recgrps growthRate%rowtype;
        recgrrs growthRate%rowtype;
        recids  Indices%rowtype;

-- Work Area
        nPY                     number;
        nPM                     number;

-- Indices Table Cursor
        Cursor csrid is
                select  a.ISIC,
                        a.IndexPR1, a.IndexSP2, a.IndexIV3, a.IndexIR4, a.IndexCU5, a.IndexLP6,
```

```
                        b. IndexPR1, b. IndexSP2, b. IndexIV3, b. IndexIR4, b. IndexCU5, b. IndexLP6,
                                                  c. GR_PR1, c. GR_SP2, c. GR_IV3, c. GR_IR4, c. GR_C
U5, c. GR_LP6,
                                                  d. GR_PR1, d. GR_SP2, d. GR_IV3, d. GR_IR4, d. GR_C
U5, d. GR_LP6,
                                                  e. CD_PR1, e. CD_SP2, e. CD_IV3, e. CD_IR4, e. CD_C
U5, e. CD_LP6,
                                                  f. CD_PR1, f. CD_SP2, f. CD_IV3, f. CD_IR4, f. CD_C
U5, f. CD_LP6,
                                                  AggregationMaster. IndustryTypeShortName
                from    Indices a, Indices b, GrowthRate c, GrowthRate d,
                                                  ContributionForAll1 e, ContributionForAll1
f, AggregationMaster
                where   a. AggregationRange = b. AggregationRange
                and             a. ISIC = b. ISIC
                and     a. Year = b. Year
                and             a. Month = b. Month
                and     a. SurveyScope = b. SurveyScope
                and             b. Pre_Rev_Mark = '2'
                and     a. AggregationRange = c. AggregationRange
                and             a. ISIC = c. ISIC
                and     a. Year = c. Year
                and             a. Month = c. Month
                and     a. SurveyScope = c. SurveyScope
                and             c. Pre_Rev_Mark = '1'
                and             c. ComparisonTarget = inComparisonTarget
                and     a. AggregationRange = d. AggregationRange
                and             a. ISIC = d. ISIC
                and     a. Year = d. Year
                and             a. Month = d. Month
                and     a. SurveyScope = d. SurveyScope
                and             d. Pre_Rev_Mark = '2'
                and             d. ComparisonTarget = inComparisonTarget
                and             a. ISIC = e. ISIC
                and     a. Year = e. Year
                and             a. Month = e. Month
                and     a. SurveyScope = e. SurveyScope
                and             e. Pre_Rev_Mark = '1'
                and             e. ComparisonTarget = inComparisonTarget
                and             e. Seq = inSeq
                and             a. ISIC = f. ISIC
                and     a. Year = f. Year
                and             a. Month = f. Month
                and     a. SurveyScope = f. SurveyScope
                and             f. Pre_Rev_Mark = '2'
                and             f. ComparisonTarget = inComparisonTarget
                and             f. Seq = inSeq
                and     a. AggregationRange = AggregationMaster. AggregationRange
                and             a. ISIC = AggregationMaster. ISIC
                        and             a. AggregationRange = '3'
                        and             a. Year = inYear
                        and             a. Month = inMonth
                        and             a. SurveyScope = inSurveyScope
                        and             a. Pre_Rev_Mark = '1';

Begin
--      DBMS_OUTPUT. PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Initial Process
        if      inComparisonTarget = '0' then
                if      inMonth = 1 then
                        nPY     := inYear - 1;
                        nPM     := 12;
                else
                        nPY     := inYear;
                        nPM     := inMonth - 1;
                end if;
        else
                        nPY     := inYear - 1;
                        nPM     := inMonth;
        end if;

-- Summary Data Selecting
        begin
                select  a. IndexPR1, a. IndexSP2, a. IndexIV3, a. IndexIR4, a. IndexCU5, a. IndexLP6,
                        b. IndexPR1, b. IndexSP2, b. IndexIV3, b. IndexIR4, b. IndexCU5, b. IndexLP6,
                                                  c. GR_PR1, c. GR_SP2, c. GR_IV3, c. GR_IR4, c. GR_C
U5, c. GR_LP6,
```

```
                                                d. GR_PR1, d. GR_SP2, d. GR_IV3, d. GR_IR4, d. GR_C
U5, d. GR_LP6
                into            recidps. IndexPR1, recidps. IndexSP2, recidps. IndexIV3,
                                                recidps. IndexIR4, recidps. IndexCU5, recidps.
IndexLP6,
                        recidrs. IndexPR1, recidrs. IndexSP2, recidrs. IndexIV3,
                        recidrs. IndexIR4, recidrs. IndexCU5, recidrs. IndexLP6,
                                                recegrps. GR_PR1, recegrps. GR_SP2, recegrps. GR_I
V3,
                                                recegrps. GR_IR4, recegrps. GR_CU5, recegrps. GR_L
P6,
                                                recegrrs. GR_PR1, recegrrs. GR_SP2, recegrrs. GR_I
V3,
                                                recegrrs. GR_IR4, recegrrs. GR_CU5, recegrrs. GR_L
P6
                from    Indices a, Indices b, GrowthRate c, GrowthRate d
                where   a. AggregationRange = b. AggregationRange
                and             a. ISIC = b. ISIC
                and     a. Year = b. Year
                and             a. Month = b. Month
                and     a. SurveyScope = b. SurveyScope
                and             b. Pre_Rev_Mark = '2'
                and     a. AggregationRange = c. AggregationRange
                and             a. ISIC = c. ISIC
                and     a. Year = c. Year
                and             a. Month = c. Month
                and     a. SurveyScope = c. SurveyScope
                and             c. Pre_Rev_Mark = '1'
                and             c. ComparisonTarget = inComparisonTarget
                and     a. AggregationRange = d. AggregationRange
                and             a. ISIC = d. ISIC
                and     a. Year = d. Year
                and             a. Month = d. Month
                and     a. SurveyScope = d. SurveyScope
                and             d. Pre_Rev_Mark = '2'
                and             d. ComparisonTarget = inComparisonTarget
                        and             a. AggregationRange = '1'
                        and             a. Year = inYear
                        and             a. Month = inMonth
                        and             a. SurveyScope = inSurveyScope
                        and             a. Pre_Rev_Mark = '1';

        exception
                when no_data_found then
                        recidps. IndexPR1 := 0;
                        recidps. IndexSP2 := 0;
                        recidps. IndexIV3 := 0;
                        recidps. IndexIR4 := 0;
                        recidps. IndexCU5 := 0;
                        recidps. IndexLP6 := 0;
      recidrs. IndexPR1 := 0;
      recidrs. IndexSP2 := 0;
      recidrs. IndexIV3 := 0;
      recidrs. IndexIR4 := 0;
      recidrs. IndexCU5 := 0;
      recidrs. IndexLP6 := 0;
                        recegrps. GR_PR1 := 0;
                        recegrps. GR_SP2 := 0;
                        recegrps. GR_IV3 := 0;
                        recegrps. GR_IR4 := 0;
                        recegrps. GR_CU5 := 0;
                        recegrps. GR_LP6 := 0;
                        recegrrs. GR_PR1 := 0;
                        recegrrs. GR_SP2 := 0;
                        recegrrs. GR_IV3 := 0;
                        recegrrs. GR_IR4 := 0;
                        recegrrs. GR_CU5 := 0;
                        recegrrs. GR_LP6 := 0;
        end;

-- Summary Pre Data Selecting
        begin
                select  IndexPR1,
                        IndexSP2,
                        IndexIV3,
                        IndexIR4,
                        IndexCU5,
```

```
                        IndexLP6
                into            recids.IndexPR1,
                        recids.IndexSP2,
                        recids.IndexIV3,
                        recids.IndexIR4,
                        recids.IndexCU5,
                        recids.IndexLP6
                from    Indices
                where   AggregationRange = '1'
                        and     Year = nPY
                        and             Month = nPM
                        and     SurveyScope = inSurveyScope
                        and             Pre_Rev_Mark = '2';

        exception
                when no_data_found then
                        recids.IndexPR1 := 0;
      recids.IndexSP2   := 0;
      recids.IndexIV3   := 0;
      recids.IndexIR4   := 0;
      recids.IndexCU5   := 0;
      recids.IndexLP6   := 0;
        end;

-- Indices Cursor Open Fetch
        open    csrid;
        fetch   csrid
                into    recidp.ISIC, recidp.IndexPR1, recidp.IndexSP2, recidp.IndexIV3,
                                        recidp.IndexIR4, recidp.IndexCU5, recidp.IndexLP6,
                recidr.IndexPR1, recidr.IndexSP2, recidr.IndexIV3,
                recidr.IndexIR4, recidr.IndexCU5, recidr.IndexLP6,
                                        recgrp.GR_PR1, recgrp.GR_SP2, recgrp.GR_IV3,
                                        recgrp.GR_IR4, recgrp.GR_CU5, recgrp.GR_LP6,
                                        recgrr.GR_PR1, recgrr.GR_SP2, recgrr.GR_IV3,
                                        recgrr.GR_IR4, recgrr.GR_CU5, recgrr.GR_LP6,
                                        recclp.CD_PR1, recclp.CD_SP2, recclp.CD_IV3,
                                        recclp.CD_IR4, recclp.CD_CU5, recclp.CD_LP6,
                                        recclr.CD_PR1, recclr.CD_SP2, recclr.CD_IV3,
                                        recclr.CD_IR4, recclr.CD_CU5, recclr.CD_LP6,
                                        recam.IndustryTypeShortName;

-- Indices Cursor Fetch Looping
        while csrid%found loop

-- Pre Data Selecting
                begin
                        select
                                                IndexPR1,
                        IndexSP2,
                        IndexIV3,
                        IndexIR4,
                        IndexCU5,
                        IndexLP6
                        into    recid.IndexPR1,
                        recid.IndexSP2,
                        recid.IndexIV3,
                        recid.IndexIR4,
                        recid.IndexCU5,
                        recid.IndexLP6
                        from  Indices
                        where AggregationRange = '3'
                        and     ISIC = recidp.ISIC
                        and Year = nPY
                        and     Month = nPM
                        and SurveyScope = inSurveyScope
                                and     Pre_Rev_Mark = '2';

                exception
                        when no_data_found then
                                recid.IndexPR1  := 0;
        recid.IndexSP2  := 0;
        recid.IndexIV3  := 0;
        recid.IndexIR4  := 0;
        recid.IndexCU5  := 0;
        recid.IndexLP6  := 0;
                end;
```

```
-- Insert Index List Table(PR1)
                p_InsBalCom(inSeq, '1', inComparisonTarget, inYear, inMonth, substr(recidp. ISI
C, 1, 4),
                        recam. IndustryTypeShortName, recid. IndexPR1, recidp. IndexPR1, recidr.
IndexPR1,
                        recidr. IndexPR1 - recidp. IndexPR1, recgrp. GR_PR1, recgrr. GR_PR1,
                        recclp. CD_PR1, recclr. CD_PR1, recclr. CD_PR1 - recclp. CD_PR1,
                        recids. IndexPR1, recidps. IndexPR1, recidrs. IndexPR1, recidrs. IndexPR1
- recidps. IndexPR1,
                        recgrps. GR_PR1, recgrrs. GR_PR1,
                        recidps. IndexPR1 - recids. IndexPR1, recidrs. IndexPR1 - recids. Index
PR1,
                        (recidrs. IndexPR1 - recids. IndexPR1) - (recidps. IndexPR1 - recids.
IndexPR1));

-- Insert Index List Table(SP2)
                p_InsBalCom(inSeq, '2', inComparisonTarget, inYear, inMonth, substr(recidp. ISI
C, 1, 4),
                        recam. IndustryTypeShortName, recid. IndexSP2, recidp. IndexSP2, recidr.
IndexSP2,
                        recidr. IndexSP2 - recidp. IndexSP2, recgrp. GR_SP2, recgrr. GR_SP2,
                        recclp. CD_SP2, recclr. CD_SP2, recclr. CD_SP2 - recclp. CD_SP2,
                        recids. IndexSP2, recidps. IndexSP2, recidrs. IndexSP2, recidrs. IndexSP2
- recidps. IndexSP2,
                        recgrps. GR_SP2, recgrrs. GR_SP2,
                        recidps. IndexSP2 - recids. IndexSP2, recidrs. IndexSP2 - recids. Index
SP2,
                        (recidrs. IndexSP2 - recids. IndexSP2) - (recidps. IndexSP2 - recids.
IndexSP2));

-- Insert Index List Table(IV3)
                p_InsBalCom(inSeq, '3', inComparisonTarget, inYear, inMonth, substr(recidp. ISI
C, 1, 4),
                        recam. IndustryTypeShortName, recid. IndexIV3, recidp. IndexIV3, recidr.
IndexIV3,
                        recidr. IndexIV3 - recidp. IndexIV3, recgrp. GR_IV3, recgrr. GR_IV3,
                        recclp. CD_IV3, recclr. CD_IV3, recclr. CD_IV3 - recclp. CD_IV3,
                        recids. IndexIV3, recidps. IndexIV3, recidrs. IndexIV3, recidrs. IndexIV3
- recidps. IndexIV3,
                        recgrps. GR_IV3, recgrrs. GR_IV3,
                        recidps. IndexIV3 - recids. IndexIV3, recidrs. IndexIV3 - recids. Index
IV3,
                        (recidrs. IndexIV3 - recids. IndexIV3) - (recidps. IndexIV3 - recids.
IndexIV3));


-- Insert Index List Table(IR4)
                p_InsBalCom(inSeq, '4', inComparisonTarget, inYear, inMonth, substr(recidp. ISI
C, 1, 4),
                        recam. IndustryTypeShortName, recid. IndexIR4, recidp. IndexIR4, recidr.
IndexIR4,
                        recidr. IndexIR4 - recidp. IndexIR4, recgrp. GR_IR4, recgrr. GR_IR4,
                        recclp. CD_IR4, recclr. CD_IR4, recclr. CD_IR4 - recclp. CD_IR4,
                        recids. IndexIR4, recidps. IndexIR4, recidrs. IndexIR4, recidrs. IndexIR4
- recidps. IndexIR4,
                        recgrps. GR_IR4, recgrrs. GR_IR4,
                        recidps. IndexIR4 - recids. IndexIR4, recidrs. IndexIR4 - recids. Index
IR4,
                        (recidrs. IndexIR4 - recids. IndexIR4) - (recidps. IndexIR4 - recids.
IndexIR4));

-- Insert Index List Table(CU5)
                p_InsBalCom(inSeq, '5', inComparisonTarget, inYear, inMonth, substr(recidp. ISI
C, 1, 4),
                        recam. IndustryTypeShortName, recid. IndexCU5, recidp. IndexCU5, recidr.
IndexCU5,
                        recidr. IndexCU5 - recidp. IndexCU5, recgrp. GR_CU5, recgrr. GR_CU5,
                        recclp. CD_CU5, recclr. CD_CU5, recclr. CD_CU5 - recclp. CD_CU5,
                        recids. IndexCU5, recidps. IndexCU5, recidrs. IndexCU5, recidrs. IndexCU5
- recidps. IndexCU5,
                        recgrps. GR_CU5, recgrrs. GR_CU5,
                        recidps. IndexCU5 - recids. IndexCU5, recidrs. IndexCU5 - recids. Index
CU5,
                        (recidrs. IndexCU5 - recids. IndexCU5) - (recidps. IndexCU5 - recids.
IndexCU5));

-- Insert Index List Table(LP6)
```

```
                p_InsBalCom(inSeq, '6', inComparisonTarget, inYear, inMonth, substr(recidp. ISI
C, 1, 4),
                        recam. IndustryTypeShortName, recid. IndexLP6, recidp. IndexLP6, recidr.
IndexLP6,
                        recidr. IndexLP6 - recidp. IndexLP6, recgrp. GR_LP6, recgrr. GR_LP6,
                        recclp. CD_LP6, recclr. CD_LP6, recclr. CD_LP6 - recclp. CD_LP6,
                        recids. IndexLP6, recidps. IndexLP6, recidrs. IndexLP6, recidrs. IndexLP6
- recidps. IndexLP6,
                        recgrps. GR_LP6, recgrrs. GR_LP6,
                        recidps. IndexLP6 - recids. IndexLP6, recidrs. IndexLP6 - recids. Index
LP6,
                        (recidrs. IndexLP6 - recids. IndexLP6) - (recidps. IndexLP6 - recids.
IndexLP6));

-- Fetch Indices Table
        fetch   csrid
                into    recidp. ISIC, recidp. IndexPR1, recidp. IndexSP2, recidp. IndexIV3,
                                        recidp. IndexIR4, recidp. IndexCU5, recidp. IndexLP6,
                recidr. IndexPR1, recidr. IndexSP2, recidr. IndexIV3,
                recidr. IndexIR4, recidr. IndexCU5, recidr. IndexLP6,
                                        recgrp. GR_PR1, recgrp. GR_SP2, recgrp. GR_IV3,
                                        recgrp. GR_IR4, recgrp. GR_CU5, recgrp. GR_LP6,
                                        recgrr. GR_PR1, recgrr. GR_SP2, recgrr. GR_IV3,
                                        recgrr. GR_IR4, recgrr. GR_CU5, recgrr. GR_LP6,
                                        recclp. CD_PR1, recclp. CD_SP2, recclp. CD_IV3,
                                        recclp. CD_IR4, recclp. CD_CU5, recclp. CD_LP6,
                                        recclr. CD_PR1, recclr. CD_SP2, recclr. CD_IV3,
                                        recclr. CD_IR4, recclr. CD_CU5, recclr. CD_LP6,
                                        recam. IndustryTypeShortName;

                end loop;

        close csrid;

end;

end BalComMk;
/
show errors
/
--commit;
```

```
/*
 * Package Name : CalcIndex
 * Created Date : 30/NOV/99
 * Update  Date : 23/FEB/00 Add SurveyScope in WeightMaster
 * Update  Date : 24/FEB/00 Add Deflator
 * Update  Date : 28/FEB/00 Add SeasonalAdjustedIndices
 * Update  Date : 16/MAR/00 Modify bug found in expand survey
 * By  K.Shibamoto@JICA SYUDY TEAM
 * NOTE : Calculate Index.
 * TEST : (1)exec Weight.CreateWeightView;     --Create aggregated weight
          (2)exec CalcIndex.Main(1999,1,'2'); --Calculate index
 * Make : @C:¥shiba¥Index-PL-SQL¥calcindex
 */

CREATE OR REPLACE PACKAGE CalcIndex IS

/* ------------------------------------------------*
 * 0.MAIN                                          *
 * ------------------------------------------------*/
    /* MAIN
    ||  is called first of all before calculating indices.
    */
    PROCEDURE MAIN (
        pYear                IN  NUMBER,    -- Year
        pMonth               IN  NUMBER,    -- Month
        pPreRev              IN  VARCHAR2   -- Pre_RevMark(1:Preliminaly 2:Revised)
    );
/* ------------------------------------------------*
 * 1.Calculation of Production Index               *
 * ------------------------------------------------*/
    /* CalcPR_Main
    ||  Prepare for Index calculation
    ||  for ALL ISIC(6digit) + CommodityCode(Item1)
    ||  (AggregateRange 5)
    */
    FUNCTION  CalcPR_Main (
        pYear                IN  NUMBER,    -- Year
        pMonth               IN  NUMBER,    -- Month
        pScope               IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev              IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY               IN  NUMBER,    -- BaseYear
        pBaseM               IN  NUMBER     -- BaseMonth
    ) return NUMBER;
    /* CalcPR5
    ||  Calculate Production Index
    ||  for each ISIC(6digit) + CommodityCode(Item1)
    ||  (AggregateRange 5)
    */
    FUNCTION  CalcPR5
        pISIC                IN  VARCHAR2,  -- ISIC
        pCommodity           IN  VARCHAR2,  -- CommodityCode
        pYear                IN  NUMBER,    -- Year
        pMonth               IN  NUMBER,    -- Month
        pScope               IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev              IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY               IN  NUMBER,    -- BaseYear
        pBaseM               IN  NUMBER     -- BaseMonth
    ) return NUMBER;
/* ------------------------------------------------*
 * 2.Calculation of Shipment Index                 *
 * ------------------------------------------------*/
    /* CalcSP_Main
    ||  Prepare for Index calculation
    ||  for ALL ISIC(6digit) + CommodityCode(Item1)
    ||  (AggregateRange 5)
    */
    FUNCTION  CalcSP_Main (
        pYear                IN  NUMBER,    -- Year
        pMonth               IN  NUMBER,    -- Month
        pScope               IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev              IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY               IN  NUMBER,    -- BaseYear
        pBaseM               IN  NUMBER     -- BaseMonth
    ) return NUMBER;
    /* CalcSP5
    ||  Calculate Shipment Index
    ||  for each ISIC(6digit) + CommodityCode(Item1)
```

```
    ||  (AggregateRange 5)
    */
    FUNCTION  CalcSP5
        pISIC                   IN  VARCHAR2,  -- ISIC
        pCommodity              IN  VARCHAR2,  -- CommodityCode
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev                 IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY                  IN  NUMBER,    -- BaseYear
        pBaseM                  IN  NUMBER     -- BaseMonth
    ) return NUMBER;
/* ------------------------------------------------*
 * 3. Calculation of Inventory Index                *
 * ------------------------------------------------*/
    /* CalcIV_Main
    ||  Prepare for Index calculation
    ||  for ALL ISIC(6digit) + CommodityCode(Item1)
    ||  (AggregateRange 5)
    */
    FUNCTION  CalcIV_Main (
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev                 IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY                  IN  NUMBER,    -- BaseYear
        pBaseM                  IN  NUMBER     -- BaseMonth
    ) return NUMBER;
    /* CalcIV5
    ||  Calculate Inventory Index
    ||  for each ISIC(6digit) + CommodityCode(Item1)
    ||  (AggregateRange 5)
    */
    FUNCTION  CalcIV5
        pISIC                   IN  VARCHAR2,  -- ISIC
        pCommodity              IN  VARCHAR2,  -- CommodityCode
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev                 IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY                  IN  NUMBER,    -- BaseYear
        pBaseM                  IN  NUMBER     -- BaseMonth
    ) return NUMBER;
/* ------------------------------------------------*
 * 4. Calculation of Inventory Ratio Index          *
 * ------------------------------------------------*/
    FUNCTION  CalcIR_Main (
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev                 IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY                  IN  NUMBER,    -- BaseYear
        pBaseM                  IN  NUMBER     -- BaseMonth
    ) return NUMBER;
    /* CalcIR5
    ||  Calculate Inventory Ratio Index
    ||  for each ISIC(6digit) + CommodityCode(Item1)
    ||  (AggregateRange 5)
    */
    FUNCTION  CalcIR5
        pISIC                   IN  VARCHAR2,  -- ISIC
        pCommodity              IN  VARCHAR2,  -- CommodityCode
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev                 IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY                  IN  NUMBER,    -- BaseYear
        pBaseM                  IN  NUMBER     -- BaseMonth
    ) return NUMBER;
/* ------------------------------------------------*
 * 5. Calculation of Capacity Utilization Index     *
 * ------------------------------------------------*/
    FUNCTION  CalcCU_Main (
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev                 IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
```

```
        pBaseY                  IN  NUMBER,    -- BaseYear
        pBaseM                  IN  NUMBER     -- BaseMonth
    ) return NUMBER;
    /* CalcIR5
    ||  Calculate Capacity Utilization Index
    ||  for each ISIC(6digit) + CommodityCode(Item1)
    ||  (AggregateRange 5)
    */
    FUNCTION  CalcCU5
        pISIC                   IN  VARCHAR2,  -- ISIC
        pCommodity              IN  VARCHAR2,  -- CommodityCode
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev                 IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY                  IN  NUMBER,    -- BaseYear
        pBaseM                  IN  NUMBER     -- BaseMonth
    ) return NUMBER;
/* ------------------------------------------------*
 * 6. Calculation of Labor Productivity Index       *
 * ------------------------------------------------*/
    FUNCTION  CalcLP_Main (
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev                 IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY                  IN  NUMBER,    -- BaseYear
        pBaseM                  IN  NUMBER     -- BaseMonth
    ) return NUMBER;
    /* CalcLP4
    ||  Calculate Labor Productivity Index
    ||  for each ISIC(6digit)
    ||  (AggregateRange 4)
    */
    FUNCTION  CalcLP4
        pISIC                   IN  VARCHAR2,  -- ISIC
        --pCommodity              IN  VARCHAR2,  -- CommodityCode
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev                 IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY                  IN  NUMBER,    -- BaseYear
        pBaseM                  IN  NUMBER     -- BaseMonth
    ) return NUMBER;
/* ------------------------------------------------*
 * 7. Common routine for aggregation                *
 * ------------------------------------------------*/
    /* AggregateIndex
    ||  is called for calculate every Index.
    ||  Because the aggregation methods for each index are the same .
    ||  (AggregateRange 5->4->3->2->1)
    */
    FUNCTION  AggregateIndex (
        pRange                  IN  NUMBER,    -- Aggregation Range
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev                 IN  VARCHAR2   -- Pre_RevMark(1:Preliminaly 2:Revised)
    )   return NUMBER;
/* ------------------------------------------------*
 * 8. Common routine for getting BasePeriod         *
 * ------------------------------------------------*/
    /* GetBasePeriod
    ||  Get base year and month for each surveyscope.
    */
    Procedure  GetBasePeriod (
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
        oYear                   OUT NUMBER,    -- Year
        oMonth                  OUT NUMBER     -- Month
    );
-- ------------
END;
/
show errors
```

```
CREATE OR REPLACE PACKAGE BODY CalcIndex IS

    /* MAIN
    ||  is called first of all before calculating indices.
    */
    PROCEDURE MAIN (
        pYear   IN  NUMBER,    -- Year
        pMonth  IN  NUMBER,    -- Month
        pPreRev IN  VARCHAR2   -- Pre_RevMark(1:Preliminaly 2:Revised)
    )
    IS
        Ret NUMBER;
        MaxScope NUMBER;
        sBase  varchar2(6);
        nBaseY number;
        nBaseM number;
    BEGIN
        -- Get the maximum number of surveyscope
        select to_number(max(SurveyScope)) into MaxScope
        from SurveyScopeMaster;

        -- Crate WeightView
        -- Weight.CreateWeightView;

        -- Clear work table;
        DELETE FROM Index_Wk;
        COMMIT WORK;

        -- Create skeleton records for every index in work table.
        FOR i IN 1..MaxScope LOOP
            /* Aggregation Range 5 */
            insert into Index_Wk
            select '5',                        -- AggregationRange
                   ISIC,                       -- ISIC(6 digit)
                   ItemType,                   -- ItemType
                   CommodityCode,              -- IndexCommodity
                   pYear,                      -- Year
                   pMonth,                     -- Month
                   to_char(i),                 -- SurveyScope
                   pPreRev,                    -- Pre_Rev_Mark
                   NULL, NULL, NULL, NULL,     -- Index1..8
                   NULL, NULL, NULL, NULL
            from CommodityMaster
            where ItemType IN ('1','6');  -- Caution!!
            commit work;                  --  Only "Capacity Utilization Index" is based on
ItemType 6 .

            /* Aggregation Range 4 */
            insert into Index_Wk
            select '4',                        -- AggregationRange
                   ISIC,                       -- ISIC(6 digit)
                   'X',                        -- ItemType
                   'XXX',                      -- IndexCommodity
                   pYear,                      -- Year
                   pMonth,                     -- Month
                   to_char(i),                 -- SurveyScope
                   pPreRev,                    -- Pre_Rev_Mark
                   NULL, NULL, NULL, NULL,     -- Index1..8
                   NULL, NULL, NULL, NULL
            from CommodityMaster
            where ItemType = '5'               -- Business condition
            and   CommodityCode = '010';
            commit work;

            /* Aggregation Range 3 */
            insert into Index_Wk
            select distinct
                   '3',                        -- AggregationRange
                   substr(ISIC, 1, 4)||'XX',   -- ISIC
                   'X',                        -- ItemType
                   'XXX',                      -- IndexCommodity
                   pYear,                      -- Year
                   pMonth,                     -- Month
                   to_char(i),                 -- SurveyScope
                   pPreRev,                    -- Pre_Rev_Mark
                   NULL, NULL, NULL, NULL,     -- Index1..8
```

```
            NULL, NULL, NULL, NULL
    from CommodityMaster
    where ItemType = '5'                -- Business condition
    and   CommodityCode = '010';        -- is the same with each Questionnaire
    commit work;

    /* Aggregation Range 2 */
    insert into Index_Wk
    select distinct
            '2',                        -- AggregationRange
            substr(ISIC, 1, 2)||'XXXX', -- ISIC
            'X',                        -- ItemType
            'XXX',                      -- IndexCommodity
            pYear,                      -- Year
            pMonth,                     -- Month
            to_char(i),                 -- SurveyScope
            pPreRev,                    -- Pre_Rev_Mark
            NULL, NULL, NULL, NULL,     -- Index1..8
            NULL, NULL, NULL, NULL
    from CommodityMaster
    where ItemType = '5'                -- Business condition
    and   CommodityCode = '010';
    commit work;

    /* Aggregation Range 1 */
    insert into Index_Wk
    values(
            '1',                        -- AggregationRange
            'XXXXXX',                   -- ISIC
            'X',                        -- ItemType
            'XXX',                      -- IndexCommodity
            pYear,                      -- Year
            pMonth,                     -- Month
            to_char(i),                 -- SurveyScope
            pPreRev,                    -- Pre_Rev_Mark
            NULL, NULL, NULL, NULL,     -- Index1..8
            NULL, NULL, NULL, NULL
    );
    commit work;
END LOOP;

/* Create Index of PR, SP, IV, IR, CU
 *  for each SurveyScope from actual figures
 *  (Aggregation Range 5)
 */
FOR i IN 1..MaxScope LOOP
    GetBasePeriod(pYear, pMonth, i, nBaseY, nBaseM);
    Ret := CalcPR_Main(pYear, pMonth, TO_CHAR(i), pPreRev, nBaseY, nBaseM);
    COMMIT WORK;
    Ret := CalcSP_Main(pYear, pMonth, TO_CHAR(i), pPreRev, nBaseY, nBaseM);
    COMMIT WORK;
    Ret := CalcIV_Main(pYear, pMonth, TO_CHAR(i), pPreRev, nBaseY, nBaseM);
    COMMIT WORK;
    Ret := CalcIR_Main(pYear, pMonth, TO_CHAR(i), pPreRev, nBaseY, nBaseM);
    COMMIT WORK;
    Ret := CalcCU_Main(pYear, pMonth, TO_CHAR(i), pPreRev, nBaseY, nBaseM);
    COMMIT WORK;
END LOOP;

/* Aggregate Index (Range 5 -> 4) */
FOR i IN 1..MaxScope LOOP   -- "i" is SurveyScope.
    Ret := AggregateIndex(4, pYear, pMonth, TO_CHAR(i), pPreRev);
    commit work;
END LOOP;

/* Create Index of LP
 *  for each SurveyScope from actual figures
 *  (Aggregation Range 4)
 */
FOR i IN 1..MaxScope LOOP
    GetBasePeriod(pYear, pMonth, i, nBaseY, nBaseM);
    Ret := CalcLP_Main(pYear, pMonth, TO_CHAR(i), pPreRev, nBaseY, nBaseM);
    COMMIT WORK;
END LOOP;

/* Aggregate Index (Range 4 -> 3 -> 2 -> 1) */
FOR j IN 1..3 LOOP              -- "j" is Aggragation Range.
```

```
    FOR i IN 1..MaxScope LOOP   -- "i" is SurveyScope.
        Ret := AggregateIndex(4-j, pYear, pMonth, TO_CHAR(i), pPreRev);
        commit work;
    END LOOP;
END LOOP;

/* delete extra indices */
delete from Index_wk
where  Year = pYear
  and  Month = pMonth
  and  Pre_Rev_Mark = pPreRev
  and  indexpr1 is null
  and  indexsp2 is null
  and  indexiv3 is null
  and  indexir4 is null
  and  indexcu5 is null
  and  indexlp6 is null ;
commit work;

/* Export data to the genuine table */
delete from Indices
where  Year = pYear
  and  Month = pMonth
  and  Pre_Rev_Mark = pPreRev;
commit work;

delete from SeasonalAdjustedIndices
where  Year = pYear
  and  Month = pMonth
  and  Pre_Rev_Mark = pPreRev;
commit work;

/* Raw Indices */
insert into Indices
select
   AggregationRange,
   ISIC,
   ItemType,
   IndexCommodityCode,
   Year,
   Month,
   SurveyScope,
   Pre_Rev_Mark,
   NVL(IndexPR1 * 100, NULL),
   NVL(IndexSP2 * 100, NULL),
   NVL(IndexIV3 * 100, NULL),
   NVL(IndexIR4 * 100, NULL),
   NVL(IndexCU5 * 100, NULL),
   NVL(IndexLP6 * 100, NULL),
   NVL(IndexLI7 * 100, NULL),
   NVL(IndexRI8 * 100, NULL),
   SYSDATE
from Index_Wk;
COMMIT WORK;


/* seasonal adjusted indices */
insert into SeasonalAdjustedIndices
select
   W.AggregationRange,
   W.ISIC,
   W.ItemType,
   W.IndexCommodityCode,
   W.Year,
   W.Month,
   W.SurveyScope,
   W.Pre_Rev_Mark,
   NVL(W.IndexPR1 * 100, NULL) * SA.IndexPR1,
   NVL(W.IndexSP2 * 100, NULL) * SA.IndexSP2,
   NVL(W.IndexIV3 * 100, NULL) * SA.IndexIV3,
   NVL(W.IndexIR4 * 100, NULL) * SA.IndexIR4,
   NVL(W.IndexCU5 * 100, NULL) * SA.IndexCU5,
   NVL(W.IndexLP6 * 100, NULL) * SA.IndexLP6,
   NVL(W.IndexLI7 * 100, NULL) * SA.IndexLI7,
   NVL(W.IndexRI8 * 100, NULL) * SA.IndexRI8,
   SYSDATE
```

```
        from Index_Wk W, SeasonalAdjustmentMaster SA
        where W.AggregationRange = SA.AggregationRange
          and W.ISIC = SA.ISIC
          and W.ItemType = SA.ItemType
          and W.IndexCommodityCode = SA.IndexCommodityCode
          and W.Year = SA.Year
          and W.Month = SA.Month
          and W.SurveyScope = SA.SurveyScope
          and SA.Version = '1' ;
        COMMIT WORK;

        EXCEPTION
            WHEN OTHERS THEN
                --error_code := SQLCODE;
                --error_msg  := SQLERRM;
                DBMS_OUTPUT.PUT_LINE('  SQL ERROR --> '|| SQLERRM );
    END;


/* ------------------------------------------------*
 * 1.Caluculation of Production Index               *
 * ------------------------------------------------*/
    FUNCTION  CalcPR_Main (
        pYear                  IN  NUMBER,    -- Year
        pMonth                 IN  NUMBER,    -- Month
        pScope                 IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev                IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY                 IN  NUMBER,    -- BaseYear
        pBaseM                 IN  NUMBER     -- BaseMonth
    )
    return number /* 1:Success , 0:Error */
    IS
        -- This cursor gets "CommodityCode" for which
        -- IndexPR should be calculated .
        CURSOR Csr IS
            SELECT /*+ all_rows*/ ISIC ,CommodityCode
            FROM CommodityMaster
            WHERE ItemType = '1'
              AND IndexScopePR1 = '1'      -- should be calculated for IndexPR
            ORDER BY ISIC ,CommodityCode;
        Rec Csr%ROWTYPE;
        Ret NUMBER; -- Get return value of Func CalcPR5

    BEGIN
        /* Calculate Index for Aggregation range 5 */
        OPEN Csr;
        LOOP
            FETCH Csr INTO Rec;
            EXIT WHEN Csr%NOTFOUND;
     --DBMS_OUTPUT.PUT_LINE('--- ISIC = '|| Rec.ISIC || ' , CommodityCode = '|| Rec.Commod
ityCode);
            Ret := CalcPR5(Rec.ISIC, Rec.CommodityCode, pYear, pMonth, pScope, pPreRev, pBaseY, p
BaseM);
        END LOOP;
        CLOSE Csr;


        RETURN 1;

        EXCEPTION
            WHEN OTHERS THEN
                RETURN 0;
    END;

    FUNCTION  CalcPR5(
       pISIC                  IN  VARCHAR2,  -- ISIC
       pCommodity             IN  VARCHAR2,  -- CommodityCode
       pYear                  IN  NUMBER,    -- Year
       pMonth                 IN  NUMBER,    -- Month
       pScope                 IN  VARCHAR2,  -- Target Scope For Calculation
       pPreRev                IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
       pBaseY                 IN  NUMBER,    -- BaseYear
       pBaseM                 IN  NUMBER     -- BaseMonth
    )
    return number /* 0 : Process is suceeded , -1 : Process is not suceeded */
    IS
        -- Definition of Variables --
```

```
    nErr     NUMBER := 0; -- count missing values
    nThisPR  NUMBER;
    nBasePR  NUMBER;
    nIndexPR NUMBER;
    nLink    NUMBER;
    nDefl    NUMBER;
BEGIN
/* Get Base Value */
    select /*+ all_rows*/      --- This hint is incredibly
                               --- effective on response !!
           sum(BP.ProductionQTY)
    into nBasePR
    from BasePeriodFile BP, EstablishmentMaster EM
    where BP.ISIC = pISIC
    and   BP.Year = pBaseY
    and   BP.Month = pBaseM
    and   BP.SurveyScope = pScope    ---- Modify 16/MAR
    and   BP.Version = '1' -- Monthly Process
    and   BP.ItemType = '1' -- Fixed Value
    and   BP.CommodityCode = pCommodity
    and   BP.ISIC = EM.ISIC
    and   BP.RegistrationNo = EM.RegistrationNo
    and   EM.SurveyScope <= pScope
    and   EM.ContinuousRespondent = '1';

/* Get This Month Value */
    select /*+ all_rows*/
           sum(PR.ProductionQTY)
    into nThisPR --, nThisSP, nThisIV
    from PastRecord PR, EstablishmentMaster EM
    where PR.ISIC = pISIC
    and   PR.Year = pYear
    and   PR.Month = pMonth
    and   PR.Pre_Rev_Mark = pPreRev -- Monthly Process
    and   PR.ItemType = '1'         -- Fixed Value
    and   PR.CommodityCode = pCommodity
    and   PR.ISIC = EM.ISIC
    and   PR.RegistrationNo = EM.RegistrationNo
    and   EM.SurveyScope <= pScope
    and   EM.ContinuousRespondent = '1' ;

/* Get Link Coefficient */
    select ProductionQTY
    into   nLink
    from   LinkMaster ---0221
    where  ISIC = pISIC
      and  ItemType = '1'
      and  CommodityCode = pCommodity
      and  SurveyScope = pScope
      and  Year = pYear
      and  Month = pMonth
      and  Version = '1';
/* Get Deflator Coefficient */
    select ProductionQTY
    into   nDefl
    from   DeflatorMaster ---0221
    where  ISIC = pISIC
      and  ItemType = '1'
      and  CommodityCode = pCommodity
      and  SurveyScope = pScope
      and  Year = pYear
      and  Month = pMonth
      and  Version = '1';

/* Production */
    IF (nBasePR = 0) THEN
        nIndexPR := NULL;
    ELSE
        nIndexPR := (nThisPR * nLink / nDefl) / nBasePR;
    END IF;

/* Set Production Index for AggregateRange5 */
    update Index_Wk
    set IndexPR1 = nIndexPR
    where AggregationRange = '5'
    and   ISIC = pISIC
    and   ItemType = '1'
```

```
        and   IndexCommodityCode = pCommodity
        and   Year = pYear
        and   Month = pMonth
        and   SurveyScope = pScope
        and   Pre_Rev_Mark = pPreRev ;

        RETURN nErr;
        EXCEPTION
            WHEN OTHERS THEN
                RETURN -1;
    END;

/* ------------------------------------------------*
 * 2.Caluculation of Shipment Index                 *
 * ------------------------------------------------*/
    FUNCTION  CalcSP_Main (
        pYear                IN  NUMBER,    -- Year
        pMonth               IN  NUMBER,    -- Month
        pScope               IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev              IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY               IN  NUMBER,    -- BaseYear
        pBaseM               IN  NUMBER     -- BaseMonth
    )
    return number /* 1:Success , 0:Error */
    IS
        -- This cursor gets "CommodityCode" for which
        -- IndexSP should be calculated .
        CURSOR Csr IS
            SELECT /*+ all_rows*/ ISIC ,CommodityCode
            FROM CommodityMaster
            WHERE ItemType = '1'
              AND IndexScopeSP2 = '1'     -- should be calculated for IndexSP
            ORDER BY ISIC ,CommodityCode;
        Rec Csr%ROWTYPE;
        Ret NUMBER; -- Get return value of Func CalcPR5

    BEGIN
        /* Calculate Index for Aggregation range 5 */
        OPEN Csr;
        LOOP
            FETCH Csr INTO Rec;
            EXIT WHEN Csr%NOTFOUND;
            Ret := CalcSP5(Rec.ISIC, Rec.CommodityCode, pYear, pMonth, pScope, pPreRev, pBaseY, p
BaseM);
        END LOOP;
        CLOSE Csr;
        RETURN 1;

        EXCEPTION
            WHEN OTHERS THEN
                --error_code := SQLCODE;
                --error_msg  := SQLERRM;
                --DBMS_OUTPUT.PUT_LINE('  SQL ERROR --> '|| SQLERRM );
                RETURN 0;
    END;

    FUNCTION  CalcSP5(
       pISIC                 IN  VARCHAR2,  -- ISIC
       pCommodity            IN  VARCHAR2,  -- CommodityCode
       pYear                 IN  NUMBER,    -- Year
       pMonth                IN  NUMBER,    -- Month
       pScope                IN  VARCHAR2,  -- Target Scope For Calculation
       pPreRev               IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
       pBaseY                IN  NUMBER,    -- BaseYear
       pBaseM                IN  NUMBER     -- BaseMonth
    )
    return number /* 0 : Process is suceeded , -1 : Process is not suceeded */
    IS
        -- Definition of Variables --
        nErr     NUMBER := 0; -- count missing values

        nThisSP  NUMBER;
        nBaseSP  NUMBER;
        nIndexSP NUMBER;
        nLink    NUMBER;
        nDefl    NUMBER;
    BEGIN
```

```
/* Get Base Value */
    select /*+ all_rows*/
           sum(BP.DomesticSales + BP.Export)
    into nBaseSP
    from BasePeriodFile BP, EstablishmentMaster EM
    where BP.ISIC = pISIC
    and   BP.Year = pBaseY
    and   BP.Month = pBaseM
    and   BP.SurveyScope = pScope     ---- Modify 16/MAR
    and   BP.Version = '1' -- Monthly Process
    and   BP.ItemType = '1' -- Fixed Value
    and   BP.CommodityCode = pCommodity
    and   BP.ISIC = EM.ISIC
    and   BP.RegistrationNo = EM.RegistrationNo
    and   EM.SurveyScope <= pScope
    and   EM.ContinuousRespondent = '1' ;

/* Get This Month Value */
    select /*+ all_rows*/
         sum(PR.DomesticSales + PR.Export)
    into nThisSP
    from PastRecord PR, EstablishmentMaster EM
    where PR.ISIC = pISIC
    and   PR.Year = pYear
    and   PR.Month = pMonth
    and   PR.Pre_Rev_Mark = pPreRev -- Monthly Process
    and   PR.ItemType = '1'         -- Fixed Value
    and   PR.CommodityCode = pCommodity
    and   PR.ISIC = EM.ISIC
    and   PR.RegistrationNo = EM.RegistrationNo
    and   EM.SurveyScope <= pScope
    and   EM.ContinuousRespondent = '1' ;

/* Get Link Coefficient */
    select ShipmentQTY
    into   nLink
    from   LinkMaster  ---0221
    where  ISIC = pISIC
      and  ItemType = '1'
      and  CommodityCode = pCommodity
      and  SurveyScope = pScope
      and  Year = pYear
      and  Month = pMonth
      and  Version = '1';
/* Get Deflator Coefficient */
    select ShipmentQTY
    into   nDefl
    from   DeflatorMaster ---0221
    where  ISIC = pISIC
      and  ItemType = '1'
      and  CommodityCode = pCommodity
      and  SurveyScope = pScope
      and  Year = pYear
      and  Month = pMonth
      and  Version = '1';

/* Shipment */
    IF (nBaseSP = 0) THEN
        nIndexSP := NULL;
    ELSE
        nIndexSP := (nThisSP * nLink / nDefl) / nBaseSP;
    END IF;

/* Set Production Index for AggregateRange5 */
    update Index_Wk
    set IndexSP2 = nIndexSP
    where AggregationRange = '5'
    and   ISIC = pISIC
    and   ItemType = '1'
    and   IndexCommodityCode = pCommodity
    and   Year = pYear
    and   Month = pMonth
    and   SurveyScope = pScope
    and   Pre_Rev_Mark = pPreRev;

    RETURN nErr;
```

```
        EXCEPTION
            WHEN OTHERS THEN
                RETURN -1;
    END;

/* ------------------------------------------------*
 * 3. Caluculation of Inventory Index               *
 * ------------------------------------------------*/
    FUNCTION  CalcIV_Main (
        pYear                IN  NUMBER,    -- Year
        pMonth               IN  NUMBER,    -- Month
        pScope               IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev              IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY               IN  NUMBER,    -- BaseYear
        pBaseM               IN  NUMBER     -- BaseMonth
    )
    return number /* 1:Success , 0:Error */
    IS
        -- This cursor gets "CommodityCode" for which
        -- IndexIV should be calculated .
        CURSOR Csr IS
            SELECT /*+ all_rows*/ ISIC , CommodityCode
            FROM CommodityMaster
            WHERE ItemType = '1'
              AND IndexScopeIV3 = '1'      -- should be calculated for IndexIV
            ORDER BY ISIC , CommodityCode;
        Rec Csr%ROWTYPE;
        Ret NUMBER; -- Get return value of Func CalcIV5

    BEGIN
        /* Calculate Index for Aggregation range 5 */
        OPEN Csr;
        LOOP
            FETCH Csr INTO Rec;
            EXIT WHEN Csr%NOTFOUND;
            Ret := CalcIV5(Rec. ISIC, Rec. CommodityCode, pYear, pMonth, pScope, pPreRev, pBaseY, p
BaseM);
        END LOOP;
        CLOSE Csr;

        RETURN 1;

        EXCEPTION
            WHEN OTHERS THEN
                --error_code := SQLCODE;
                --error_msg  := SQLERRM;
                --DBMS_OUTPUT. PUT_LINE('  SQL ERROR --> '|| SQLERRM );
                RETURN 0;
    END;

    FUNCTION  CalcIV5(
       pISIC                 IN  VARCHAR2,  -- ISIC
       pCommodity            IN  VARCHAR2,  -- CommodityCode
       pYear                 IN  NUMBER,    -- Year
       pMonth                IN  NUMBER,    -- Month
       pScope                IN  VARCHAR2,  -- Target Scope For Calculation
       pPreRev               IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
       pBaseY                IN  NUMBER,    -- BaseYear
       pBaseM                IN  NUMBER     -- BaseMonth
    )
    return number /* 0 : Process is suceeded , -1 : Process is not suceeded */
    IS
        -- Definition of Variables --
        nErr     NUMBER := 0;

        nThisIV  NUMBER;
        nBaseIV  NUMBER;
        nIndexIV NUMBER;
        nLink    NUMBER;
        nDefl    NUMBER;
    BEGIN
    /* Get Base Value */
        select /*+ all_rows*/
            sum(BP. ME_Inventory)
        into nBaseIV
        from BasePeriodFile BP, EstablishmentMaster EM
        where BP. ISIC = pISIC
```

```
        and   BP.Year = pBaseY
        and   BP.Month = pBaseM
        and   BP.SurveyScope = pScope     ---- Modify 16/MAR
        and   BP.Version = '1' -- Monthly Process
        and   BP.ItemType = '1' -- Fixed Value
        and   BP.CommodityCode = pCommodity
        and   BP.ISIC = EM.ISIC
        and   BP.RegistrationNo = EM.RegistrationNo
        and   EM.SurveyScope <= pScope
        and   EM.ContinuousRespondent = '1' ;

    /* Get This Month Value */
        select /*+ all_rows*/
              sum(PR.ME_Inventory)
        into nThisIV
        from PastRecord PR, EstablishmentMaster EM
        where PR.ISIC = pISIC
        and   PR.Year = pYear
        and   PR.Month = pMonth
        and   PR.Pre_Rev_Mark = pPreRev -- Monthly Process
        and   PR.ItemType = '1'          -- Fixed Value
        and   PR.CommodityCode = pCommodity
        and   PR.ISIC = EM.ISIC
        and   PR.RegistrationNo = EM.RegistrationNo
        and   EM.SurveyScope <= pScope
        and   EM.ContinuousRespondent = '1' ;

    /* Get Link Coefficient */
        select ME_Inventory
        into   nLink
        from   LinkMaster --0221
        where  ISIC = pISIC
          and  ItemType = '1'
          and  CommodityCode = pCommodity
          and  SurveyScope = pScope
          and  Year = pYear
          and  Month = pMonth
          and  Version = '1';
    /* Get Deflator Coefficient */
        select ME_Inventory
        into   nDefl
        from   DeflatorMaster ---0221
        where  ISIC = pISIC
          and  ItemType = '1'
          and  CommodityCode = pCommodity
          and  SurveyScope = pScope
          and  Year = pYear
          and  Month = pMonth
          and  Version = '1';

    /* Production */
        IF (nBaseIV = 0) THEN
            nIndexIV := NULL;
        ELSE
            nIndexIV := (nThisIV * nLink / nDefl) / nBaseIV;
        END IF;

    /* Set Production Index for AggregateRange5 */
        update Index_Wk
        set IndexIV3 = nIndexIV
        where AggregationRange = '5'
        and   ISIC = pISIC
        and   ItemType = '1'
        and   IndexCommodityCode = pCommodity
        and   Year = pYear
        and   Month = pMonth
        and   SurveyScope = pScope
        and   Pre_Rev_Mark = pPreRev ;

        RETURN nErr;

        EXCEPTION
            WHEN OTHERS THEN
                RETURN -1;
    END;

/* ------------------------------------------------*
```

```
* 4. Caluculation of Inventory Ratio Index          *
* ---------------------------------------------------*/
   FUNCTION  CalcIR_Main (
       pYear                  IN  NUMBER,    -- Year
       pMonth                 IN  NUMBER,    -- Month
       pScope                 IN  VARCHAR2,  -- Target Scope For Calculation
       pPreRev                IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
       pBaseY                 IN  NUMBER,    -- BaseYear
       pBaseM                 IN  NUMBER     -- BaseMonth
   )
   return number /* 1:Success , 0:Error */
   IS
       -- This cursor gets "CommodityCode" for which
       -- IndexIR should be calculated .
       CURSOR Csr IS
           SELECT /*+ all_rows*/ ISIC , CommodityCode
           FROM CommodityMaster
           WHERE ItemType = '1'
             AND IndexScopeIR4 = '1'      -- should be calculated for IndexPR
           ORDER BY ISIC , CommodityCode;
       Rec Csr%ROWTYPE;
       Ret NUMBER; -- Get return value of Func CalcPR5

   BEGIN
       /* Calculate Index for Aggregation range 5 */
       OPEN Csr;
       LOOP
           FETCH Csr INTO Rec;
           EXIT WHEN Csr%NOTFOUND;
           Ret := CalcIR5(Rec. ISIC, Rec. CommodityCode, pYear, pMonth, pScope, pPreRev, pBaseY, p
BaseM);
       END LOOP;
       CLOSE Csr;

       RETURN 1;

       EXCEPTION
           WHEN OTHERS THEN
               RETURN 0;
   END;

   FUNCTION  CalcIR5(
      pISIC                   IN  VARCHAR2,  -- ISIC
      pCommodity              IN  VARCHAR2,  -- CommodityCode
      pYear                   IN  NUMBER,    -- Year
      pMonth                  IN  NUMBER,    -- Month
      pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
      pPreRev                 IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
      pBaseY                  IN  NUMBER,    -- BaseYear
      pBaseM                  IN  NUMBER     -- BaseMonth
   )
   return number /* 0 : Process is suceeded , -1 : Process is not suceeded */
   IS
       nErr      NUMBER := 0; -- count missing values
       nThisIV   NUMBER;
       nBaseIV   NUMBER;
       nThisSP   NUMBER;
       nBaseSP   NUMBER;
       nLinkSP   NUMBER;
       nLinkIV   NUMBER;
       nDefISP   NUMBER;
       nDefIIV   NUMBER;
       nIndexIR  NUMBER;
   BEGIN
   /* Get Base Value */
       select /*+ all_rows*/
           sum(BP. DomesticSales) + sum(BP. Export),
           sum(BP. ME_Inventory)
       into nBaseSP, nBaseIV
       from BasePeriodFile BP, EstablishmentMaster EM
       where BP. ISIC = pISIC
       and   BP. Year = pBaseY
       and   BP. Month = pBaseM
       and   BP. SurveyScope = pScope     ---- Modify 16/MAR
       and   BP. Version = '1'   -- Monthly Process
       and   BP. ItemType = '1'  -- Fixed Value
       and   BP. CommodityCode = pCommodity
```

```
        and   BP.ISIC = EM.ISIC
        and   BP.RegistrationNo = EM.RegistrationNo
        and   EM.SurveyScope <= pScope
        and   EM.ContinuousRespondent = '1' ;

    /* Get This Month Value */
        select /*+ all_rows*/
            sum(PR.DomesticSales) + sum(PR.Export),
            sum(PR.ME_Inventory)
        into nThisSP,nThisIV
        from PastRecord PR, EstablishmentMaster EM
        where PR.ISIC = pISIC
        and   PR.Year = pYear
        and   PR.Month = pMonth
        and   PR.Pre_Rev_Mark = pPreRev -- Monthly Process
        and   PR.ItemType = '1'         -- Fixed Value
        and   PR.CommodityCode = pCommodity
        and   PR.ISIC = EM.ISIC
        and   PR.RegistrationNo = EM.RegistrationNo
        and   EM.SurveyScope <= pScope
        and   EM.ContinuousRespondent = '1' ;

    /* Get Link Coefficient */
        select ShipmentQTY , ME_Inventory
        into   nLinkSP, nLinkIV
        from   LinkMaster --0221
        where  ISIC = pISIC
          and  ItemType = '1'
          and  CommodityCode = pCommodity
          and  SurveyScope = pScope
          and  Year = pYear
          and  Month = pMonth
          and  Version = '1';
    /* Get Deflator Coefficient */
        select ShipmentQTY , ME_Inventory
        into   nDeflSP, nDeflIV
        from   DeflatorMaster --0221
        where  ISIC = pISIC
          and  ItemType = '1'
          and  CommodityCode = pCommodity
          and  SurveyScope = pScope
          and  Year = pYear
          and  Month = pMonth
          and  Version = '1';

    /* Inventory Ratio */
        nIndexIR := ( (nThisIV * nLinkIV / nDeflIV) / (nThisSP * nLinkSP / nDeflSP) ) /
( nBaseIV / nBaseSP );

    /* Set Production Index for AggregateRange5 */
        update Index_Wk
        set   IndexIR4 = nIndexIR
        where AggregationRange = '5'
        and   ISIC = pISIC
        and   ItemType = '1'
        and   IndexCommodityCode = pCommodity
        and   Year = pYear
        and   Month = pMonth
        and   SurveyScope = pScope
        and   Pre_Rev_Mark = pPreRev ;

        RETURN nErr;

        EXCEPTION
            WHEN OTHERS THEN
                RETURN -1;
    END;

/* ------------------------------------------------*
 * 5.Calculation of Capacity Utilization Index     *
 * ------------------------------------------------*/
    FUNCTION  CalcCU_Main (
        pYear                IN  NUMBER,    -- Year
        pMonth               IN  NUMBER,    -- Month
        pScope               IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev              IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY               IN  NUMBER,    -- BaseYear
```

```
        pBaseM                  IN  NUMBER    -- BaseMonth
    )
    return number /* 1:Success , 0:Error */
    IS
        -- This cursor gets "CommodityCode" for which
        -- IndexCU should be calculated .
        CURSOR Csr IS
            SELECT /*+ all_rows*/ ISIC ,CommodityCode
            FROM CommodityMaster
            WHERE ItemType = '6'  -- Only this index is calculated from ItemType 6.
              AND IndexScopeCU5 = '1' -- should be calculated for IndexCU
            ORDER BY ISIC ,CommodityCode;
        Rec Csr%ROWTYPE;
        Ret NUMBER; -- Get return value of Func CalcPR5

    BEGIN
        /* Calculate Index for Aggregation range 5 */
        OPEN Csr;
        LOOP
            FETCH Csr INTO Rec;
            EXIT WHEN Csr%NOTFOUND;
            Ret := CalcCU5(Rec.ISIC, Rec.CommodityCode, pYear, pMonth, pScope, pPreRev, pBaseY, p
BaseM);
        END LOOP;
        CLOSE Csr;


        RETURN 1;

        EXCEPTION
            WHEN OTHERS THEN
                --error_code := SQLCODE;
                --error_msg  := SQLERRM;
                --DBMS_OUTPUT.PUT_LINE('  SQL ERROR --> '|| SQLERRM );
                RETURN 0;
    END;

    FUNCTION  CalcCU5(
       pISIC                IN  VARCHAR2, -- ISIC
       pCommodity           IN  VARCHAR2, -- CommodityCode
       pYear                IN  NUMBER,   -- Year
       pMonth               IN  NUMBER,   -- Month
       pScope               IN  VARCHAR2, -- Target Scope For Calculation
       pPreRev              IN  VARCHAR2, -- Pre_RevMark(1:Preliminaly 2:Revised)
       pBaseY               IN  NUMBER,   -- BaseYear
       pBaseM               IN  NUMBER    -- BaseMonth
    )
    return number /* 0 : Process is suceeded , -1 : Process is not suceeded */
    IS
        nErr     NUMBER := 0; -- count missing values

        nThisPR  NUMBER; -- production at this month
        nBasePR  NUMBER; -- production at the base month

        nThisCP  NUMBER; -- capacity at this month
        nBaseCP  NUMBER; -- capacity at the base month

        nIndexCU NUMBER;
    BEGIN
    /* Get Base Value */
        -- capacity from ItemType 6 --
        select /*+ all_rows*/
               sum(BP.Capacity)
        into nBaseCP
        from BasePeriodFile BP, EstablishmentMaster EM
        where BP.ISIC = pISIC
        and   BP.Year = pBaseY
        and   BP.Month = pBaseM
        and   BP.SurveyScope = pScope    ---- Modify 16/MAR
        and   BP.Version = '1'  -- Monthly Process
        and   BP.ItemType = '6' -- Fixed Value
        and   BP.CommodityCode = pCommodity
        and   BP.ISIC = EM.ISIC
        and   BP.RegistrationNo = EM.RegistrationNo
        and   EM.SurveyScope <= pScope
        and   EM.ContinuousRespondent = '1' ;
```

```
    -- production from ItemType 1 --
    select /*+ all_rows*/
           sum(BP.ProductionQTY)
    into nBasePR
    from BasePeriodFile BP, EstablishmentMaster EM
    where BP.ISIC = pISIC
    and   BP.Year = pBaseY
    and   BP.Month = pBaseM
    and   BP.SurveyScope = pScope     ---- Modify 16/MAR
    and   BP.Version = '1'  -- Monthly Process
    and   BP.ItemType = '1' -- Fixed Value
    and   BP.CommodityCode  IN
          (
           select /*+ all_rows*/ Commodi tyCode
           from CommodityMaster
           where ISIC = pISIC
           and   ItemType = '1'
           and   CapacityCode = pCommodity
          )
    and   BP.ISIC = EM.ISIC
    and   BP.RegistrationNo = EM.RegistrationNo
    and   EM.SurveyScope <= pScope
    and   EM.ContinuousRespondent = '1' ;

/* Get This Month Value */
    -- capacity from ItemType 6 --
    select /*+ all_rows*/
           sum(PR.Capacity * LM.Capacity / DM.Capacity)
    into nThisCP
    from PastRecord PR, EstablishmentMaster EM, LinkMaster LM, DeflatorMaster DM
    where PR.ISIC = pISIC
    and   PR.Year = pYear
    and   PR.Month = pMonth
    and   PR.Pre_Rev_Mark = pPreRev
    and   PR.ItemType = '6' -- Fixed Value
    and   PR.CommodityCode = pCommodity
    and   PR.ISIC = EM.ISIC
    and   PR.RegistrationNo = EM.RegistrationNo
    and   EM.SurveyScope <= pScope
    and   EM.ContinuousRespondent = '1'
    and   LM.ISIC = PR.ISIC
    and   LM.ItemType = PR.ItemType
    and   LM.CommodityCode = PR.CommodityCode
    and   LM.Year = PR.Year
    and   LM.Month = PR.Month
    and   LM.SurveyScope = pScope
    and   LM.Version = '1'
    and   DM.ISIC          = LM.ISIC
    and   DM.ItemType      = LM.ItemType
    and   DM.CommodityCode = LM.CommodityCode
    and   DM.Year          = LM.Year
    and   DM.Month         = LM.Month
    and   DM.SurveyScope   = LM.SurveyScope
    and   DM.Version       = LM.Version;

    -- production from ItemType 1 --
    select /*+ all_rows*/
           sum(PR.ProductionQTY * LM.ProductionQTY / DM.ProductionQTY)
    into nThisPR
    from PastRecord PR, EstablishmentMaster EM, LinkMaster LM, DeflatorMaster DM
    where PR.ISIC = pISIC
    and   PR.Year = pYear
    and   PR.Month = pMonth
    and   PR.Pre_Rev_Mark = pPreRev
    and   PR.ItemType = '1' -- Fixed Value
    and   PR.CommodityCode  IN
          (
           select /*+ all_rows*/ CommodityCode
           from CommodityMaster
           where ISIC = pISIC
           and   ItemType = '1'
           and   CapacityCode = pCommodity
          )
    and   PR.ISIC = EM.ISIC
    and   PR.RegistrationNo = EM.RegistrationNo
    and   EM.SurveyScope <= pScope
    and   EM.ContinuousRespondent = '1'
```

```
        and   LM.ISIC = PR.ISIC
        and   LM.ItemType = PR.ItemType
        and   LM.CommodityCode = PR.CommodityCode
        and   LM.Year = PR.Year
        and   LM.Month = PR.Month
        and   LM.SurveyScope = pScope
        and   LM.Version = '1'
        and   DM.ISIC          = LM.ISIC
        and   DM.ItemType      = LM.ItemType
        and   DM.CommodityCode = LM.CommodityCode
        and   DM.Year          = LM.Year
        and   DM.Month         = LM.Month
        and   DM.SurveyScope   = LM.SurveyScope
        and   DM.Version       = LM.Version;

    /* Capacity Utilization */
        nIndexCU := (nThisPR / nThisCP  ) / ( nBasePR / nBaseCP );

    /* Set Production Index for AggregateRange5 */
        update Index_Wk
        set   IndexCU5 = nIndexCU
        where AggregationRange = '5'
        and   ISIC = pISIC
        and   ItemType = '6'
        and   IndexCommodityCode = pCommodity
        and   Year = pYear
        and   Month = pMonth
        and   SurveyScope = pScope
        and   Pre_Rev_Mark = pPreRev ;

        RETURN nErr;

        EXCEPTION
            WHEN OTHERS THEN
                --error_code := SQLCODE;
                --error_msg  := SQLERRM;
                RETURN -1;
    END;
/* ------------------------------------------------*
 * 6.Caluculation of Labor Productivity Index      *
 * ------------------------------------------------*/
    FUNCTION  CalcLP_Main (
        pYear                IN  NUMBER,    -- Year
        pMonth               IN  NUMBER,    -- Month
        pScope               IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev              IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
        pBaseY               IN  NUMBER,    -- BaseYear
        pBaseM               IN  NUMBER     -- BaseMonth
    )
    return number /* 1:Success , 0:Error */
    IS
        -- This cursor gets "CommodityCode" for which
        -- IndexIR should be calculated .
        CURSOR Csr IS
            SELECT /*+ all_rows*/ ISIC
            FROM ClassMaster
            ORDER BY ISIC ;
        Rec Csr%ROWTYPE;
        Ret NUMBER; -- Get return value of Func CalcPR5

    BEGIN
        /* Calculate Index for Aggregation range 5 */
        OPEN Csr;
        LOOP
            FETCH Csr INTO Rec;
            EXIT WHEN Csr%NOTFOUND;
            Ret := CalcLP4(Rec.ISIC, pYear, pMonth, pScope, pPreRev, pBaseY, pBaseM);
        END LOOP;
        CLOSE Csr;

        RETURN 1;

        EXCEPTION
            WHEN OTHERS THEN
                --error_code := SQLCODE;
                --error_msg  := SQLERRM;
                --DBMS_OUTPUT.PUT_LINE('  SQL ERROR --> '|| SQLERRM );
```

```
            RETURN 0;
END;

FUNCTION  CalcLP4(
   pISIC                    IN  VARCHAR2,  -- ISIC
   pYear                    IN  NUMBER,    -- Year
   pMonth                   IN  NUMBER,    -- Month
   pScope                   IN  VARCHAR2,  -- Target Scope For Calculation
   pPreRev                  IN  VARCHAR2,  -- Pre_RevMark(1:Preliminaly 2:Revised)
   pBaseY                   IN  NUMBER,    -- BaseYear
   pBaseM                   IN  NUMBER     -- BaseMonth
)
return number /* 0 : Process is suceeded , -1 : Process is not suceeded */
IS
    nErr     NUMBER := 0; -- count missing values
    nThisPV  NUMBER; -- Production Value at current month
    nBasePV  NUMBER; -- Production Value at base month
    nThisLI  NUMBER; -- Labor Input at current month
    nBaseLI  NUMBER; -- Labor Input at base month
    nIndexLP NUMBER;
BEGIN
/* Get Base Value */
-- 1. Production Value
    select /*+ all_rows*/
        sum(BP.ProductionQTY * BP.UnitPrice)
    into  nBasePV
    from BasePeriodFile BP, EstablishmentMaster EM , CommodityMaster CM
    where BP.ISIC = pISIC
    and   BP.Year = pBaseY
    and   BP.Month = pBaseM
    and   BP.SurveyScope = pScope    ---- Modify 16/MAR
    and   BP.Version = '1'  -- Monthly Process
    and   BP.ItemType = '1' -- Fixed Value
    and   BP.ISIC = EM.ISIC
    and   BP.RegistrationNo = EM.RegistrationNo
    and   EM.SurveyScope <= pScope
    and   EM.ContinuousRespondent = '1'
    and   CM.ISIC = BP.ISIC
    and   CM.ItemType = BP.ItemType
    and   CM.CommodityCode = BP.CommodityCode
    and   CM.IndexScopeLP6 = '1'
    group by BP.ISIC  ;

-- 2. Labor Input
    select /*+ all_rows*/
      SUM(BP1.Labor_SC * BP2.LaborTotal * BP3.LaborTotal)
    into  nBaseLI
    from BasePeriodFile BP1, BasePeriodFile BP2,BasePeriodFile BP3 ,
         EstablishmentMaster EM
    where BP1.ISIC = pISIC
    and   BP1.Year = pBaseY
    and   BP1.Month = pBaseM
    and   BP1.ItemType = '3' -- Fixed Value
    and   BP1.CommodityCode = '010' -- Fixed Value
    and   BP1.SurveyScope = pScope    ---- Modify 16/MAR
    and   BP1.Version = '1'  -- Monthly Process
    --
    and   BP2.ISIC = BP1.ISIC
    and   BP2.Year = BP1.Year
    and   BP2.Month = BP1.Month
    and   BP2.RegistrationNo = BP1.RegistrationNo
    and   BP2.ItemType = BP1.ItemType
    and   BP2.CommodityCode = '020' -- Fixed Value
    and   BP2.SurveyScope = pScope    ---- Modify 16/MAR
    and   BP2.Version = BP1.Version
    --
    and   BP3.ISIC = BP1.ISIC
    and   BP3.Year = BP1.Year
    and   BP3.Month = BP1.Month
    and   BP3.RegistrationNo = BP1.RegistrationNo
    and   BP3.ItemType = BP1.ItemType
    and   BP3.CommodityCode = '030' -- Fixed Value
    and   BP3.SurveyScope = pScope    ---- Modify 16/MAR
    and   BP3.Version = BP1.Version
    --
    and   EM.ISIC = BP1.ISIC
    and   EM.RegistrationNo = BP1.RegistrationNo
```

```
    and   EM.SurveyScope <= pScope
    and   EM.ContinuousRespondent = '1' ;

/* Get This Month Value */
-- 1. Production Value
    select /*+ all_rows*/
     SUM((PR.ProductionQTY * LM.ProductionQTY / DM.ProductionQTY) * BP.UnitPrice)
    into nThisPV
    from PastRecord PR, EstablishmentMaster EM , BasePeriodFile BP ,
         CommodityMaster CM, LinkMaster LM, DeflatorMaster DM
    where PR.ISIC = pISIC
    and   PR.Year = pYear
    and   PR.Month = pMonth
    and   PR.Pre_Rev_Mark = pPreRev
    and   PR.ItemType = '1' -- Fixed Value
    --
    and   BP.ISIC = PR.ISIC
    and   BP.Year = pBaseY
    and   BP.Month = pBaseM
    and   BP.RegistrationNo = PR.RegistrationNo
    and   BP.ItemType = PR.ItemType
    and   BP.CommodityCode = PR.CommodityCode
    and   BP.SurveyScope = pScope    ---- Modify 16/MAR
    and   BP.Version = '1' -- Fixed Value
    --
    and   EM.ISIC = PR.ISIC
    and   EM.RegistrationNo = PR.RegistrationNo
    and   EM.SurveyScope <= pScope
    and   EM.ContinuousRespondent = '1'
    --
    and   CM.ISIC = PR.ISIC
    and   CM.ItemType = PR.ItemType
    and   CM.CommodityCode = PR.CommodityCode
    and   CM.IndexScopeLP6 = '1'
    --
    and   LM.ISIC = PR.ISIC
    and   LM.ItemType = PR.ItemType
    and   LM.CommodityCode = PR.CommodityCode
    and   LM.Year  = PR.Year
    and   LM.Month = PR.Month
    and   LM.SurveyScope = pScope
    and   LM.Version = '1'
    --
    and   DM.ISIC          = LM.ISIC
    and   DM.ItemType      = LM.ItemType
    and   DM.CommodityCode = LM.CommodityCode
    and   DM.Year          = LM.Year
    and   DM.Month         = LM.Month
    and   DM.SurveyScope   = LM.SurveyScope
    and   DM.Version       = LM.Version
    group by PR.ISIC ;

-- 2. Labor Input
    select /*+ all_rows*/
      SUM( (PR1.Labor_SC * LM1.Worker)
           * (PR2.LaborTotal * LM2.WorkingHour)
             * (PR3.LaborTotal * LM3.WorkingDay) )
    into  nThisLI
    from PastRecord PR1, PastRecord PR2, PastRecord PR3 ,
         LinkMaster LM1, LinkMaster LM2, LinkMaster LM3 ,
         EstablishmentMaster EM
    where PR1.ISIC = pISIC
    and   PR1.Year = pYear
    and   PR1.Month = pMonth
    and   PR1.ItemType = '3' -- Fixed Value
    and   PR1.CommodityCode = '010' -- Fixed Value
    and   PR1.Pre_Rev_Mark = pPreRev
    --
    and   PR2.ISIC = PR1.ISIC
    and   PR2.Year = PR1.Year
    and   PR2.Month = PR1.Month
    and   PR2.RegistrationNo = PR1.RegistrationNo
    and   PR2.ItemType = PR1.ItemType
    and   PR2.CommodityCode = '020' -- Fixed Value
    and   PR2.Pre_Rev_Mark = PR1.Pre_Rev_Mark
    --
    and   PR3.ISIC = PR1.ISIC
```

```
        and   PR3.Year = PR1.Year
        and   PR3.Month = PR1.Month
        and   PR3.RegistrationNo = PR1.RegistrationNo
        and   PR3.ItemType = PR1.ItemType
        and   PR3.CommodityCode = '030' -- Fixed Value
        and   PR3.Pre_Rev_Mark = PR1.Pre_Rev_Mark
        --
        and   LM1.ISIC = PR1.ISIC
        and   LM1.ItemType = PR1.ItemType
        and   LM1.CommodityCode = PR1.CommodityCode
        and   LM1.Year = PR1.Year
        and   LM1.Month = PR1.Month
        and   LM1.SurveyScope = pScope
        and   LM1.Version = '1'
        --
        and   LM2.ISIC = PR2.ISIC
        and   LM2.ItemType = PR2.ItemType
        and   LM2.CommodityCode = PR2.CommodityCode
        and   LM2.Year = PR2.Year
        and   LM2.Month = PR2.Month
        and   LM2.SurveyScope = pScope
        and   LM2.Version = '1'
        --
        and   LM3.ISIC = PR3.ISIC
        and   LM3.ItemType = PR3.ItemType
        and   LM3.CommodityCode = PR3.CommodityCode
        and   LM3.Year = PR3.Year
        and   LM3.Month = PR3.Month
        and   LM3.SurveyScope = pScope
        and   LM3.Version = '1'
        --
        and   EM.ISIC = PR1.ISIC
        and   EM.RegistrationNo = PR1.RegistrationNo
        and   EM.SurveyScope <= pScope
        and   EM.ContinuousRespondent = '1' ;

    /* Labor Productivity */
        nIndexLP := (nThisPV / nThisLI) / (nBasePV / nBaseLI);

    /* Set Production Index for AggregateRange5 */
        update Index_Wk
        set   IndexLP6 = nIndexLP
        where AggregationRange = '4'
        and   ISIC = pISIC
        and   ItemType = 'X'
        and   IndexCommodityCode = 'XXX'
        and   Year = pYear
        and   Month = pMonth
        and   SurveyScope = pScope
        and   Pre_Rev_Mark = pPreRev ;
        RETURN nErr;

        EXCEPTION
            WHEN OTHERS THEN
                RETURN -1;
    END;

/* ------------------------------------------------*
 * 7.Common routine for aggregation                *
 * ------------------------------------------------*/
    FUNCTION  AggregateIndex (
        pRange                IN  NUMBER,    -- Aggregation Range
        pYear                 IN  NUMBER,    -- Year
        pMonth                IN  NUMBER,    -- Month
        pScope                IN  VARCHAR2,  -- Target Scope For Calculation
        pPreRev               IN  VARCHAR2   -- Pre_RevMark(1:Preliminaly 2:Revised)
    )   return NUMBER
    IS
        -- Definition of Cursor and Record Type and Variable --
        CURSOR Csr IS
            SELECT I.AggregationRange AS AggregationRange,
                   I.ISIC             AS ISIC,
                   I.ItemType         AS ItemType,
                   I.IndexCommodityCode AS IndexCommodityCode,
                   NVL(I.IndexPR1,0)  AS IndexPR1,
                   NVL(I.IndexSP2,0)  AS IndexSP2,
                   NVL(I.IndexIV3,0)  AS IndexIV3,
```

```
                NVL(I.IndexIR4,0) AS IndexIR4,
                NVL(I.IndexCU5,0) AS IndexCU5,
                NVL(I.IndexLP6,0) AS IndexLP6,
                NVL(I.IndexLI7,0) AS IndexLI7,
                NVL(I.IndexRI8,0) AS IndexRI8,
                V.WeightPR1       AS WeightPR,
                V.WeightSP2       AS WeightSP,
                V.WeightIV3       AS WeightIV,
                V.WeightIR4       AS WeightIR,
                V.WeightCU5       AS WeightCU,
                V.WeightLP6       AS WeightLP,
                V.WeightLI7       AS WeightLI,
                V.WeightRI8       AS WeightRI,
                V.IntegratedKey   AS IntegratedKey
            FROM  Index_Wk I, WeightAggregation V
            WHERE I.AggregationRange = TO_CHAR(pRange + 1)
              AND I.Year = pYear
              AND I.Month = pMonth
              AND I.SurveyScope = pScope
              AND I.Pre_Rev_Mark = pPreRev
              AND I.AggregationRange = V.AggregationRange
              AND I.ISIC = V.ISIC
              AND I.ItemType = V.ItemType
              AND I.IndexCommodityCode = V.CommodityCode
              AND I.SurveyScope = V.SurveyScope          --0221
              AND V.Version = '1'                        --0221
            ORDER BY I.ISIC, I.ItemType, I.IndexCommodityCode;
        Rec Csr%ROWTYPE;
        TotalWeightPR NUMBER := 0;
        TotalWeightSP NUMBER := 0;
        TotalWeightIV NUMBER := 0;
        TotalWeightIR NUMBER := 0;
        TotalWeightCU NUMBER := 0;
        TotalWeightLP NUMBER := 0;
        TotalWeightLI NUMBER := 0;
        TotalWeightRI NUMBER := 0;
        PrevKey VARCHAR(8) := 'START';
        SameKeyRecCnt  NUMBER := 0;
        Ret NUMBER := 0;
        PR  NUMBER := 0; --1
        SP  NUMBER := 0; --2
        IV  NUMBER := 0; --3
        IR  NUMBER := 0; --4
        CU  NUMBER := 0; --5
        LP  NUMBER := 0; --6
        LI  NUMBER := 0; --7
        RI  NUMBER := 0; --8

    BEGIN
        OPEN Csr;
        LOOP
            FETCH Csr INTO Rec;
            EXIT WHEN Csr%NOTFOUND;

            /* If the KEY breaks, set the value(index) on the database. */
            IF (PrevKey != 'START') AND (PrevKey != Rec.IntegratedKey) THEN
                UPDATE Index_Wk
                  SET IndexPR1 = PR,
                      IndexSP2 = SP,
                      IndexIV3 = IV,
                      IndexIR4 = IR,
                      IndexCU5 = CU,
                      IndexLP6 = LP,
                      IndexLI7 = LI,
                      IndexRI8 = RI
                WHERE AggregationRange = TO_CHAR(pRange)
                  AND ISIC = PrevKey
                  AND ItemType = 'X'
                  AND IndexCommodityCode = 'XXX'
                  AND Year = pYear
                  AND Month = pMonth
                  AND SurveyScope = pScope
                  AND Pre_Rev_Mark = pPreRev;

              /* Reset the counter and index of previous key */
                SameKeyRecCnt := 0;
                PR := 0;
```

```
        SP := 0;
        IV := 0;
        IR := 0;
        CU := 0;
        LP := 0;
        LI := 0;
        RI := 0;
END IF;

/* Save the KEY for Key-Break controlling */
PrevKey := Rec.IntegratedKey;

/* Get the Total Weight for the curent KEY(ISIC) */
IF SameKeyRecCnt = 0 THEN
    SELECT WeightPR1, WeightSP2, WeightIV3,
           WeightIR4, WeightCU5, WeightLP6,
           WeightLI7, WeightRI8
    INTO   TotalWeightPR, TotalWeightSP, TotalWeightIV,
           TotalWeightIR, TotalWeightCU, TotalWeightLP,
           TotalWeightLI, TotalWeightRI
    FROM   WeightAggregation
    WHERE  AggregationRange = TO_CHAR(pRange)
    AND    SurveyScope = pScope  --0221
    AND    Version = '1'         --0221
    AND    ISIC = Rec.IntegratedKey
    AND    ItemType = 'X'
    AND    ROWNUM = 1 ;
END IF;

/* Store the data */
SameKeyRecCnt := SameKeyRecCnt + 1;

-- 1. Production Index
IF TotalWeightPR = 0 THEN
    PR := 0;
ELSE
    PR := PR + (Rec.WeightPR / TotalWeightPR) * Rec.IndexPR1;
END IF;

-- 2. Shipment Index
IF TotalWeightSP = 0 THEN
    SP := 0;
ELSE
    SP := SP + (Rec.WeightSP / TotalWeightSP) * Rec.IndexSP2;
END IF;

-- 3. Inventory Index
IF TotalWeightIV = 0 THEN
    IV := 0;
ELSE
    IV := IV + (Rec.WeightIV / TotalWeightIV) * Rec.IndexIV3;
END IF;

-- 4. Inventory Ratio Index
IF TotalWeightIR = 0 THEN
    IR := 0;
ELSE
    IR := IR + (Rec.WeightIR / TotalWeightIR) * Rec.IndexIR4;
END IF;

-- 5. Capacity Utilization Index
IF TotalWeightCU = 0 THEN
    CU := 0;
ELSE
    CU := CU + (Rec.WeightCU / TotalWeightCU) * Rec.IndexCU5;
END IF;

-- 6. Labor Productivity Index
IF TotalWeightLP = 0 THEN
    LP := 0;
ELSE
    LP := LP + (Rec.WeightLP / TotalWeightLP) * Rec.IndexLP6;
END IF;

-- 7. Labor Input Index
IF TotalWeightLI = 0 THEN
    LI := 0;
```

```
            ELSE
                LI := LI + (Rec.WeightLI / TotalWeightLI) * Rec.IndexLI7;
            END IF;

            -- 8. Raw Material Inventory Index
            IF TotalWeightRI = 0 THEN
                RI := 0;
            ELSE
                RI := RI + (Rec.WeightRI / TotalWeightRI) * Rec.IndexRI8;
            END IF;
        END LOOP;

       /* Update the final data */
        IF (PrevKey != 'START') THEN
            UPDATE Index_Wk
              SET IndexPR1 = PR,
                  IndexSP2 = SP,
                  IndexIV3 = IV,
                  IndexIR4 = IR,
                  IndexCU5 = CU,
                  IndexLP6 = LP,
                  IndexLI7 = LI,
                  IndexRI8 = RI
            WHERE AggregationRange = TO_CHAR(pRange)
              AND ISIC = PrevKey
              AND ItemType = 'X'
              AND IndexCommodityCode = 'XXX'
              AND Year = pYear
              AND Month = pMonth
              AND SurveyScope = pScope
              AND Pre_Rev_Mark = pPreRev;
        END IF;

        CLOSE Csr;
        RETURN 1;

        EXCEPTION
            WHEN OTHERS THEN
                RETURN 0;
    END;

/* ------------------------------------------------------*
 * 8.Common routine for getting BasePeriod               *
 * ------------------------------------------------------*/
    /* GetBasePeriod
    ||  Get base year and month for each surveyscope.
    */
    Procedure  GetBasePeriod (
        pYear                IN  NUMBER,    -- Year
        pMonth               IN  NUMBER,    -- Month
        pScope               IN  VARCHAR2,  -- Target Scope For Calculation
        oYear                OUT NUMBER,    -- Year
        oMonth               OUT NUMBER     -- Month
    )
    IS
        sBase  varchar2(6);
    BEGIN
        select /*+ all_rows*/
            to_char(max(Year*100 + Month)) -- Formatting 'YYYYMM'
        into sBase
        from   BasePeriodFile
        where  (Year*100 + Month) <= (pYear*100 + pMonth)
          and  SurveyScope = pScope;

        oYear  := substr(sBase, 1, 4);
        oMonth := substr(sBase, 5, 2);
    END;

/*================== end of package body =======================*/
END;
/*==============================================================*/

/
show errors
```

```
-- ==================================================
-- Package Name : Survey Data Checking
-- Package ID   : cksv
-- Create Date  : 02/Sep/1999
-- Replace Date : xx/xxx/xxxx
-- Author       : Nakamura
--
-- File         : @c:¥jica¥cksv
-- Test         : select * from warninglog
-- ==================================================
create or replace package cksv is
procedure p_cksurvey (
        inYear                                  In number,
        inMonth                                 In number,
        inISIC                          In varchar2,
        inRegistrationNo        In varchar2);

end;
/
show errors
/

create or replace package body cksv is
-------------------------------------------------------
-- Full Input Checking
-------------------------------------------------------
function f_fullinput(
        inBM_Inventory  in number,
        inProductionQTY in number,
        inReceipts                      in number,
        inDomesticSales in number,
        inExport                                in number,
        inOtherSales            in number,
        inME_Inventory  in number,
        inShipmentValue in number) return boolean is

        bReturn  boolean := false;
begin
        if      inBM_Inventory is null
        and inProductionQTY is null
        and inReceipts is null
        and inDomesticSales is null
        and inExport is null
        and inOtherSales is null
        and inME_Inventory is null
        and inShipmentValue is null then
        bReturn := true;
                return bReturn;
        end if;

        if      inBM_Inventory is not null
        and inProductionQTY is not null
        and inReceipts is not null
        and inDomesticSales is not null
        and inExport is not null
        and inOtherSales is not null
        and inME_Inventory is not null
        and inShipmentValue is not null then
                bReturn := true;
                return bReturn;
        end if;

        return bReturn;

exception
        when others then
                return bReturn;
end;

-------------------------------------------------------
-- Logical Ballance Checking
-------------------------------------------------------
function f_ballance(
        inBM_Inventory  in number,
        inProductionQTY in number,
        inReceipts              in number,
        inDomesticSales in number,
```

```
        inExport                                in number,
        inOtherSales            in number,
        inME_Inventory  in number) return boolean is

        bReturn   boolean := false;
        BM_Inventory  number;
        ProductionQTY  number;
        Receipts   number;
        DomesticSales number;
        Export    number;
        OtherSales   number;
        ME_Inventory  number;

begin
-- Data Change
        BM_Inventory  := nvl(inBM_Inventory, 0);
        ProductionQTY := nvl(inProductionQTY, 0);
        Receipts                        := nvl(inReceipts, 0);
        DomesticSales   := nvl(inDomesticSales, 0);
        Export                          := nvl(inExport, 0);
        OtherSales              := nvl(inOtherSales, 0);
        ME_Inventory    := nvl(inME_Inventory, 0);

-- Logical ballance caluculation
        if  BM_Inventory + (ProductionQTY + Receipts) - (DomesticSales + Export + OtherSal
es)
                between ME_Inventory - 1 and ME_Inventory + 1 then
                bReturn := true;
                return bReturn;
        end if;

        return bReturn;

exception
        when others then
  return bReturn;
end;

-----------------------------------------------------------
-- Operation Days Checking
-----------------------------------------------------------
function f_opedays(
        inLaborTotal  in number,
        inYear    in number,
        inMonth    in number)  return boolean is

        bReturn    boolean := false;
        dLastDay   date;
        nLastDay   number;

begin
        if      inLaborTotal is null or inLaborTotal = 0 then
                bReturn := true;
                return bReturn;
        end if;

        dLastDay := last_day(to_date(to_char(inYear,'9999') || to_char(inMonth,'09') || '0
1','yyyymmdd'));
        nLastDay := to_number(to_char(dLastDay,'dd'));

        if inLaborTotal > nLastDay then
        return bReturn;
        end if;

        bReturn := true;
        return bReturn;

exception
        when others then
                null;
end;

-----------------------------------------------------------
-- Comparison Check
-----------------------------------------------------------
function f_compprev(
        inCurrValue in  number,
```

```
        inPrevValue in  number,
        inDevRange in number,
        inPrevExist in boolean) return boolean is
        bReturn    boolean := false;
        nUnderPrevValue  number;
        nOverPrevValue  number;
begin
-- Current Data Null
        if inCurrValue is null then
                bReturn := true;
                return bReturn;
        end if;

-- Previous data not found
        if inPrevExist = false then
                bReturn := true;
                return bReturn;
        end if;

-- Previous Data null or 0
        if      inPrevValue is null or inPrevValue = 0 then
                bReturn := true;
                return bReturn;
        end if;

-- Check range
        nUnderPrevValue := inPrevValue * (100 - inDevRange) / 100;
        nOverPrevValue  := inPrevValue * (100 + inDevRange) / 100;
        if      inCurrValue >= nUnderPrevValue and inCurrValue <= nOverPrevValue then
                bReturn := true;
                return bReturn;
        end if;

        return bReturn;

exception
        when others then
                return bReturn;
end;

-------------------------------------------------------
-- Continuous response checking
-------------------------------------------------------
function f_continue(
        inCurrValue in number,
        inPrevValue in number,
        inPrevExist in boolean) return boolean is
        bReturn  boolean := false;

begin
-- Current Month data is inputed
        if      inCurrValue is not null then
                bReturn := true;
                return bReturn;
        end if;
-- Previous month data not found
        if inPrevExist = false then
                bReturn := true;
                return bReturn;
        end if;
-- Previous month data value is null
        if      inPrevValue is null then
                bReturn := true;
                return bReturn;
        end if;

        return bReturn;

exception
        when others then
                return bReturn;
end;

-------------------------------------------------------
-- Inventory Checking
-------------------------------------------------------
function f_inventory(
```

```
        inBM_Inventory in number,
        inME_Inventory in number,
        inPrevExist in boolean) return boolean is
        bReturn  boolean := false;

begin
-- Current Month data is inputed
        if      inBM_Inventory is null then
                bReturn := true;
                return bReturn;
        end if;

-- Previous month data not found
        if inPrevExist = false then
                bReturn := true;
                return bReturn;
        end if;

-- Previous month data value is null
        if      inME_Inventory is null then
                bReturn := true;
                return bReturn;
        end if;

-- Equal Checking
        if      inBM_Inventory = inME_Inventory then
                bReturn := true;
                return bReturn;
        end if;

        return bReturn;

exception
        when others then
                return bReturn;
end;

-------------------------------------------------------
-- Unit Price Check And Caluculation
-------------------------------------------------------
function f_uprice(
        inShipmentValue         in number,
        inDomesticSales         in number,
        inExport                                        in number,
        outUnitPrice            out number) return boolean is

        bReturn                                 boolean := false;
        ShipmentValue  number;
        DomesticSales  number;
        Export    number;

begin
-- All Null Checking
        if      inShipmentValue is null
        and inDomesticSales is null
        and inExport is null then
                bReturn := true;
                outUnitPrice := 0;
                return bReturn;
        end if;

-- All Zero Checking
        if      inShipmentValue = 0
        and inDomesticSales = 0
        and inExport = 0 then
                bReturn := true;
                outUnitPrice := 0;
                return bReturn;
        end if;

-- Data Change
        ShipmentValue := nvl(inShipmentValue, 0);
        DomesticSales := nvl(inDomesticSales, 0);
        Export := nvl(inExport, 0);

-- Zero Checking
        if ShipmentValue = 0 then
```

```
                outUnitPrice := 0;
                return bReturn;
        end if;

        if DomesticSales + Export = 0 then
                outUnitPrice := 0;
                return bReturn;
        end if;

-- Unit Price Caluculation
        outUnitPrice := ShipmentValue / (DomesticSales + Export);

        bReturn := true;
        return bReturn;

exception
        when others then
                return bReturn;
end;

-------------------------------------------------------
-- Worning Log Output
-------------------------------------------------------
procedure p_warningout(
        inMessageCode                   in Varchar2,
        inISIC                                          in Varchar2,
        inRegistrationNo        in Varchar2,
        inYear                                          in Number,
        inMonth                                         in Number,
        inItemType                              in Varchar2,
        inCommodityCode         in Varchar2) is

begin
        Insert
  into  WarningLog (
                ISIC,
                RegistrationNo,
                Year,
                Month,
                ItemType,
                CommodityCode,
                MessageCode)
  Values (
                inISIC,
                inRegistrationNo,
                inYear,
                inMonth,
                inItemType,
                inCommodityCode,
                inMessageCode);

exception
        when dup_val_on_index then
                null;
end;

-------------------------------------------------------
-- Survey Data Checking
-------------------------------------------------------
procedure p_cksurvey (
        inYear                                          In number,
        inMonth                                         In number,
        inISIC                                          In varchar2,
  inRegistrationNo      In varchar2) is

-- Record Area Define
        recsd                   SurveyData%rowtype;
        recpmpr         PastRecord%rowtype;
        recpypr         PastRecord%rowtype;
        recdrm          DeviationRangeMaster%rowtype;
        recsdfg         vw_SurveyDataFG%rowtype;

-- Flg Area Define
        bPMPastRecord   boolean;
        bPYPastRecord   boolean;

-- Date Check Area
```

```
        nPMYear      SurveyData.Year%type;
        nPMMonth     SurveyData.Month%type;
        nPYYear      SurveyData.Year%type;
        nPYMonth     SurveyData.Month%type;

-- Work Area Define
        nUnitPrice     number;

-- Survey F/G Data Cursor
        Cursor csrSurveyFG is
                select  *
        from    vw_SurveyDataFG
                where   ISIC     = inISIC
        and             RegistrationNo = inRegistrationNo
        and             Year    = inYear
        and             Month    = inMonth;

-- Past Record Unit Cursor(Last Month)
        Cursor csrPMUPastRecord is
                select *
                        from PastRecord
                        where ISIC    = inISIC
                        and RegistrationNo = inRegistrationNo
                        and Year    = nPMYear
                        and Month    = nPMMonth
                        and ItemType  = '1'
                        and CommodityCode = recsdfg.CommodityCode
                        and Pre_Rev_Mark in ('1','2')
                        order by Pre_Rev_Mark desc;

-- Past Record Unit Cursor(Last Year Month)
        Cursor csrPYUPastRecord is
                select *
                        from PastRecord
                        where ISIC    = inISIC
    and RegistrationNo = inRegistrationNo
    and Year    = nPYYear
    and Month    = nPYMonth
    and ItemType  = '1'
    and CommodityCode = recsdfg.CommodityCode
    and Pre_Rev_Mark in ('1','2')
     order by Pre_Rev_Mark desc;

-- Survey Data Cursor
        Cursor csrSurveyData is
   select *
   from SurveyData
   where ISIC    = inISIC
   and  RegistrationNo = inRegistrationNo
   and  Year    = inYear
   and  Month    = inMonth;

-- Past Record Cursor(Last Month)
        Cursor csrPMPastRecord is
  select *
   from PastRecord
   where ISIC    = inISIC
    and RegistrationNo = inRegistrationNo
    and Year    = nPMYear
    and Month    = nPMMonth
    and ItemType  = recsd.ItemType
    and CommodityCode = recsd.CommodityCode
    and Pre_Rev_Mark in ('1','2')
     order by Pre_Rev_Mark desc;

-- Past Record Cursor(Last Year Month)
        Cursor csrPYPastRecord is
  select *
   from PastRecord
   where ISIC    = inISIC
    and RegistrationNo = inRegistrationNo
    and Year    = nPYYear
    and Month    = nPYMonth
    and ItemType  = recsd.ItemType
    and CommodityCode = recsd.CommodityCode
    and Pre_Rev_Mark in ('1','2')
     order by Pre_Rev_Mark desc;
```

```
Begin
-- DBMS_OUTPUT.PUT_LINE('Start PGM');
-- Warning Log Delete
        delete from WarningLog
                where ISIC                                          = inISIC
                        and RegistrationNo  = inRegistrationNo
                        and Year                                            = inYear
                        and Month                                           = inMonth;

-- Date Caluculation
        if      inMonth = 1 Then
                nPMYear  := inYear - 1;
                nPMMonth := 12;
        else
                nPMYear  := inYear;
                nPMMonth := inMonth - 1;
        end if;
        nPYYear  := inYear - 1;
        nPYMonth := inMonth;

-- Deviation Range Master Selecting
        begin
                Select *
                        into    recdrm
                        from    DeviationRangeMaster
                        where ISIC  = inISIC;
        exception
                when no_data_found then
                        recdrm.DR_ComparePreMonth := 10;
                        recdrm.DR_ComparePreYear := 10;
                        recdrm.DR_UnitPrice    := 10;
                        recdrm.DR_LaborInput   := 10;
                        recdrm.DR_LaborDays    := 10;
        end;

-- Survey F/G Open -> Fetch
        open csrSurveyFG;
        fetch csrSurveyFG into recsdfg;
        while csrSurveyFG%found loop
-- Full Input Check
                if  not f_fullinput (recsdfg.BM_Inventory, recsdfg.ProductionQTY, recsdfg.Re
ceipts,
        recsdfg.DomesticSales, recsdfg.Export, recsdfg.OtherSales,
        recsdfg.ME_Inventory, recsdfg.ShipmentValue) then
                        p_warningout ('I11', inISIC, inRegistrationNo, inYear,
        inMonth, '1', recsdfg.CommodityCode);
        end if;
-- Unit Price Checking
                if  not f_uprice (recsdfg.ShipmentValue, recsdfg.DomesticSales, recsdfg.Expo
rt,
                nUnitPrice) then
                        p_warningout ('U21', inISIC, inRegistrationNo, inYear,
        inMonth, '2', recsdfg.CommodityCode);
        end if;
-- Comparison Check for Unit Price
        if      nUnitPrice = 0 then
                null;
        else
                bPMPastRecord := false;
                open csrPMUPastRecord;
                fetch csrPMUPastRecord  into recpmpr;
                if      csrPMUPastRecord%found then
                bPMPastRecord := true;
                end if;
                close csrPMUPastRecord;

                bPYPastRecord := false;
                open csrPYUPastRecord;
                fetch csrPYUPastRecord  into recpypr;
                if      csrPYUPastRecord%found then
                        bPYPastRecord := true;
                end if;
                close csrPYUPastRecord;

                if  not f_compprev (nUnitPrice, recpmpr.UnitPrice,
        recdrm.DR_UnitPrice, bPMPastRecord) then
```

```
                p_warningout    ('M23', inISIC, inRegistrationNo, inYear,
        inMonth, '2', recsdfg.CommodityCode);
                end if;
                if  not f_compprev (nUnitPrice, recpypr.UnitPrice,
        recdrm.DR_UnitPrice, bPYPastRecord) then
                p_warningout    ('Y23', inISIC, inRegistrationNo, inYear,
        inMonth, '2', recsdfg.CommodityCode);
                end if;
        end if;

        fetch csrSurveyFG into recsdfg;
        end loop;
        close csrSurveyFG;

-- Survey Data Open -> Fetch
        open csrSurveyData;
        fetch csrSurveyData into recsd;
        while csrSurveyData%found loop

-- Past Record Selecting(Last Month)
        bPMPastRecord := false;
        open csrPMPastRecord;
        fetch csrPMPastRecord  into recpmpr;
        if      csrPMPastRecord%found then
                bPMPastRecord := true;
        end if;
        close csrPMPastRecord;

-- Past Record Selecting(Last Year)
        bPYPastRecord := false;
        open csrPYPastRecord;
        fetch csrPYPastRecord  into recpypr;
        if      csrPYPastRecord%found then
                bPYPastRecord := true;
        end if;
        close csrPYPastRecord;

-- Check survey data by item type
        if  recsd.ItemType = '1' then
-- Logical ballance checking
                if  not f_ballance (recsd.BM_Inventory, recsd.ProductionQTY, recsd.Receipts,
                recsd.DomesticSales, recsd.Export, recsd.OtherSales,
                recsd.ME_Inventory) then
                p_warningout('B11', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;

-- Comparison Check(Previous Month)
                if  not f_compprev (recsd.BM_Inventory, recpmpr.BM_Inventory,
                recdrm.DR_ComparePreMonth, bPMPastRecord) then
                p_warningout('M11', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                if  not f_compprev (recsd.ProductionQTY, recpmpr.ProductionQTY,
          recdrm.DR_ComparePreMonth, bPMPastRecord) then
                p_warningout('M12', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                if  not f_compprev (recsd.Receipts, recpmpr.Receipts,
          recdrm.DR_ComparePreMonth, bPMPastRecord) then
                p_warningout('M13', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                if  not f_compprev (recsd.DomesticSales, recpmpr.DomesticSales,
        recdrm.DR_ComparePreMonth, bPMPastRecord) then
                p_warningout('M14', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                  if    not f_compprev (recsd.Export, recpmpr.Export,
                      recdrm.DR_ComparePreMonth, bPMPastRecord) then
                    p_warningout('M15', inISIC, inRegistrationNo, inYear,
                      inMonth, recsd.ItemType, recsd.CommodityCode);
                  end if;
                  if  not f_compprev (recsd.OtherSales, recpmpr.OtherSales,
                      recdrm.DR_ComparePreMonth, bPMPastRecord) then
                    p_warningout('M16', inISIC, inRegistrationNo, inYear,
                        inMonth, recsd.ItemType, recsd.CommodityCode);
```

```
                end if;
                if  not f_compprev (recsd. ME_Inventory, recpmpr. ME_Inventory,
                    recdrm. DR_ComparePreMonth, bPMPastRecord) then
                  p_warningout('M17', inISIC, inRegistrationNo, inYear,
                    inMonth, recsd. ItemType, recsd. CommodityCode);
                end if;

-- Comparison Check(Month of previous year)
              if      not f_compprev (recsd. BM_Inventory, recpypr. BM_Inventory,
              recdrm. DR_ComparePreYear, bPYPastRecord) then
              p_warningout('Y11', inISIC, inRegistrationNo, inYear,
              inMonth, recsd. ItemType, recsd. CommodityCode);
              end if;
              if      not f_compprev (recsd. ProductionQTY, recpypr. ProductionQTY,
              recdrm. DR_ComparePreYear, bPYPastRecord) then
              p_warningout('Y12', inISIC, inRegistrationNo, inYear,
      inMonth, recsd. ItemType, recsd. CommodityCode);
              end if;
              if      not f_compprev (recsd. Receipts, recpypr. Receipts,
      recdrm. DR_ComparePreYear, bPYPastRecord) then
              p_warningout('Y13', inISIC, inRegistrationNo, inYear,
              inMonth, recsd. ItemType, recsd. CommodityCode);
              end if;
              if      not f_compprev (recsd. DomesticSales, recpypr. DomesticSales,
      recdrm. DR_ComparePreYear, bPYPastRecord) then
              p_warningout('Y14', inISIC, inRegistrationNo, inYear,
              inMonth, recsd. ItemType, recsd. CommodityCode);
              end if;
                      if      not f_compprev (recsd. Export, recpypr. Export,
              recdrm. DR_ComparePreYear, bPYPastRecord) then
                      p_warningout('Y15', inISIC, inRegistrationNo, inYear,
              inMonth, recsd. ItemType, recsd. CommodityCode);
              end if;
              if      not f_compprev (recsd. OtherSales, recpypr. OtherSales,
              recdrm. DR_ComparePreYear, bPYPastRecord) then
              p_warningout('Y16', inISIC, inRegistrationNo, inYear,
              inMonth, recsd. ItemType, recsd. CommodityCode);
              end if;
              if      not f_compprev (recsd. ME_Inventory, recpypr. ME_Inventory,
              recdrm. DR_ComparePreYear, bPYPastRecord) then
              p_warningout('Y17', inISIC, inRegistrationNo, inYear,
              inMonth, recsd. ItemType, recsd. CommodityCode);
              end if;

-- Continuous checking
              if not f_continue (recsd. BM_Inventory, recpmpr. BM_Inventory, bPMPastRecord)
then
              p_warningout    ('C11', inISIC, inRegistrationNo, inYear,
      inMonth, recsd. ItemType, recsd. CommodityCode);
              end if;
                if not f_continue (recsd. ProductionQTY, recpmpr. ProductionQTY, bPMPastReco
rd) then
                  p_warningout    ('C12', inISIC, inRegistrationNo, inYear,
                      inMonth, recsd. ItemType, recsd. CommodityCode);
                end if;
                if not f_continue (recsd. Receipts, recpmpr. Receipts, bPMPastRecord) then
                  p_warningout    ('C13', inISIC, inRegistrationNo, inYear,
                      inMonth, recsd. ItemType, recsd. CommodityCode);
                end if;
                if not f_continue (recsd. DomesticSales, recpmpr. DomesticSales, bPMPastReco
rd) then
                  p_warningout    ('C14', inISIC, inRegistrationNo, inYear,
                      inMonth, recsd. ItemType, recsd. CommodityCode);
                end if;
                if not f_continue (recsd. Export, recpmpr. Export, bPMPastRecord) then
                  p_warningout    ('C15', inISIC, inRegistrationNo, inYear,
                      inMonth, recsd. ItemType, recsd. CommodityCode);
                end if;
                if not f_continue (recsd. OtherSales, recpmpr. OtherSales, bPMPastRecord) th
en
                  p_warningout    ('C16', inISIC, inRegistrationNo, inYear,
                      inMonth, recsd. ItemType, recsd. CommodityCode);
                end if;
                if not f_continue (recsd. ME_Inventory, recpmpr. ME_Inventory, bPMPastRecord)
then
                  p_warningout    ('C17', inISIC, inRegistrationNo, inYear,
                      inMonth, recsd. ItemType, recsd. CommodityCode);
```

```
                end if;

-- Inventory Check
                if  not f_inventory (recsd.BM_Inventory, recpmpr.ME_Inventory,
                                        bPMPastRecord) then
                p_warningout('E11', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;

        elsif recsd.ItemType  = '2' then
-- Comparison Check(Previous Month)
                if  not f_compprev (recsd.ShipmentValue, recpmpr.ShipmentValue,
        recdrm.DR_ComparePreMonth, bPMPastRecord) then
                p_warningout    ('M21', inISIC, inRegistrationNo, inYear,
        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                if  not f_compprev (recsd.SalesPlan, recpmpr.SalesPlan,
          recdrm.DR_ComparePreMonth, bPMPastRecord) then
        p_warningout    ('M22', inISIC, inRegistrationNo, inYear,
        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;

-- Comparison Check(Month of previous year)
                if not f_compprev (recsd.ShipmentValue, recpypr.ShipmentValue,
        recdrm.DR_ComparePreYear, bPYPastRecord) then
                p_warningout    ('Y21', inISIC, inRegistrationNo, inYear,
        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                if not f_compprev (recsd.SalesPlan, recpypr.SalesPlan,
        recdrm.DR_ComparePreYear, bPYPastRecord) then
                p_warningout    ('Y22', inISIC, inRegistrationNo, inYear,
        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;

-- Continuous checking
                if not f_continue (recsd.ShipmentValue, recpmpr.ShipmentValue, bPMPastRecord)
then
                p_warningout    ('C21', inISIC, inRegistrationNo, inYear,
        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                if not f_continue (recsd.SalesPlan, recpmpr.SalesPlan, bPMPastRecord) then
                p_warningout    ('C22', inISIC, inRegistrationNo, inYear,
        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;

        elsif recsd.ItemType  = '3' then
-- Operation Days Checking
                if recsd.CommodityCode = '030' then
                if  not f_opedays (recsd.LaborTotal, inYear, inMonth) then
                        p_warningout    ('D31', inISIC, inRegistrationNo, inYear,
                        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                end if;
-- Comparison Check Of Labor Input(Previous Month)
                if recsd.CommodityCode = '010' then
                if  not f_compprev (recsd.LaborTotal, recpmpr.LaborTotal,
                        recdrm.DR_LaborInput, bPMPastRecord) then
                        p_warningout('M31', inISIC, inRegistrationNo, inYear,
                        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                                if      not f_compprev (recsd.Labor_SC, recpmpr.Labor_SC,
                recdrm.DR_LaborInput, bPMPastRecord) then
                        p_warningout('M32', inISIC, inRegistrationNo, inYear,
                        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                end if;
-- Comparison Check Of Labor Hours(Previous Month)
                if recsd.CommodityCode = '020' then
                if      not f_compprev (recsd.LaborTotal, recpmpr.LaborTotal,
                recdrm.DR_LaborHour, bPMPastRecord) then
                        p_warningout('M33', inISIC, inRegistrationNo, inYear,
                        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                end if;
-- Comparison Check Of Labor Days(Previous Month)
                if recsd.CommodityCode = '030' then
                if  not f_compprev (recsd.LaborTotal, recpmpr.LaborTotal,
```

```
                recdrm.DR_LaborDays, bPMPastRecord) then
                        p_warningout('M34', inISIC, inRegistrationNo, inYear,
                        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                end if;
-- Comparison Check of Labor Input(Month of previous year)
                if recsd.CommodityCode = '010' then
                if  not f_compprev (recsd.LaborTotal, recpypr.LaborTotal,
                        recdrm.DR_LaborInput, bPYPastRecord) then
                        p_warningout    ('Y31', inISIC, inRegistrationNo, inYear,
                        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                if  not f_compprev (recsd.Labor_SC, recpypr.Labor_SC,
                recdrm.DR_LaborInput, bPYPastRecord) then
                        p_warningout    ('Y32', inISIC, inRegistrationNo, inYear,
                        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                end if;
-- Comparison Check Of Labor Days(Month of previous year)
                if recsd.CommodityCode = '020' then
                if      not f_compprev (recsd.LaborTotal, recpypr.LaborTotal,
                        recdrm.DR_LaborHour, bPYPastRecord) then
                        p_warningout    ('Y33', inISIC, inRegistrationNo, inYear,
                        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                end if;
-- Comparison Check Of Labor Days(Month of previous year)
                if      recsd.CommodityCode = '030' then
                if  not f_compprev (recsd.LaborTotal, recpypr.LaborTotal,
                        recdrm.DR_LaborDays, bPYPastRecord) then
                        p_warningout    ('Y34', inISIC, inRegistrationNo, inYear,
                        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                end if;
-- Continuous checking
                if      not f_continue (recsd.LaborTotal, recpmpr.LaborTotal, bPMPastRecord)
then
                        p_warningout    ('C31', inISIC, inRegistrationNo, inYear,
                        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                if      not f_continue (recsd.Labor_SC, recpmpr.Labor_SC, bPMPastRecord) then
                        p_warningout    ('C32', inISIC, inRegistrationNo, inYear,
                        inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;

        elsif recsd.ItemType  = '4' then
-- Comparison Check(Previous Month)
                if      not f_compprev (recsd.RwMt_MEInventory, recpmpr.RwMt_MEInventory,
            recdrm.DR_ComparePreMonth, bPMPastRecord) then
                p_warningout    ('M41', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                if      not f_compprev (recsd.RwMt_MEInventoryValue, recpmpr.RwMt_MEInvento
ryValue,
            recdrm.DR_ComparePreMonth, bPMPastRecord) then
                p_warningout    ('M42', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;

-- Comparison Check(Month of previous year)
                if      not f_compprev (recsd.RwMt_MEInventory, recpypr.RwMt_MEInventory,
                recdrm.DR_ComparePreYear, bPYPastRecord) then
                p_warningout    ('Y41', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
                if      not f_compprev (recsd.RwMt_MEInventoryValue, recpypr.RwMt_MEInvento
ryValue,
                recdrm.DR_ComparePreYear, bPYPastRecord) then
                p_warningout    ('Y42', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;

-- Continuous checking
                if      not f_continue (recsd.RwMt_MEInventory, recpmpr.RwMt_MEInventory, bP
MPastRecord) then
                p_warningout    ('C41', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
```

```
                end if;
                if      not f_continue (recsd.RwMt_MEInventoryValue, recpmpr.RwMt_MEInvento
ryValue, bPMPastRecord) then
                p_warningout    ('C42', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;

        elsif recsd.ItemType = '5' then
                null;
        elsif recsd.ItemType = '6' then
-- Comparison Check(Previous Month)
                if      not f_compprev (recsd.Capacity, recpmpr.Capacity,
                recdrm.DR_ComparePreMonth, bPMPastRecord) then
                p_warningout    ('M61', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
-- Comparison Check(Month of previous year)
                if not f_compprev (recsd.Capacity, recpypr.Capacity,
                recdrm.DR_ComparePreYear, bPYPastRecord) then
                p_warningout    ('Y61', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;
-- Continuous checking
                if      not f_continue (recsd.Capacity, recpmpr.Capacity, bPMPastRecord) then
                p_warningout    ('C61', inISIC, inRegistrationNo, inYear,
                inMonth, recsd.ItemType, recsd.CommodityCode);
                end if;

        end if;

  fetch csrSurveyData into recsd;
        end loop;

        close csrSurveyData;
end;

end;
/
show errors
/
```

```
-- ==================================================
-- Package Name : Contribution Ratio, Degree Calcuration
-- Package ID   : CtbCal
-- Create Date  : 04/Nov/1999
-- Replace Date : 15/Feb/2000
-- Author       : T. Nakamura
--
-- File         : @g:¥storedproc¥CtbCal;
-- Test                                  :       BEGIN CtbCal.p_IndustryInAll(77, 1999, 2, '1
', '2', '0'); commit; END;
-- Test                                  : BEGIN CtbCal.p_CommInAll(77, 1999, 2, '1', '2', '0');
commit; END;
-- Test                                  :       BEGIN CtbCal.p_CommInIndustry(77, '1', 'XXXX
XX', 1999, 2, '1', '2', '0'); END;
-- Test                                  :       BEGIN CtbCal.p_EstInComm(2, '4', '151210', 19
99, 12, '2', '1', '0'); END;
-- ==================================================
create or replace package CtbCal is

procedure p_IndustryInAll (
        inSeq                                                   in      number,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2,
        inComparisonTarget      in      varchar2);

procedure p_CommInAll (
        inSeq                                                   in      number,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2,
        inComparisonTarget      in      varchar2);

procedure p_CommInIndustry (
        inSeq                                                   in      number,
        inAggregationRange      in      Varchar2,
        inISIC                                          in      Varchar2,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2,
        inComparisonTarget      in      varchar2);

procedure p_EstInComm (
        inSeq                                                   in  number,
        inAggregationRange      in      varchar2,
        inISIC                                          in      varchar2,
        inYear                                          in  number,
        inMonth                                         in      number,
        inPre_Rev_Mark                  in      varchar2,
        inSurveyScope                           in      varchar2,
        inComparisonTarget      in      varchar2);


end CtbCal;
/
show errors
/

create or replace package body CtbCal is
-------------------------------------------------------
-- Contribution Degree Calculation
-------------------------------------------------------
function f_CtbDegreeCal (
        inIndexCur                                              in      number,
        inIndexPre                                              in      number,
        inWeight                                                        in      num
ber,
        inIntIndexPre                                   in      number,
        inWeightTtl                                             in      number) ret
urn number is

        nCtbDegree      number;

begin
```

```
        if      inIntIndexPre = 0 or inWeightTtl = 0 then
                nCtbDegree := null;
                return nCtbDegree;
        end if;

        nCtbDegree := ((inIndexCur * inWeight) - (inIndexPre * inWeight)) /
                                                        (inIntIndexPre * inWeightT
tl) * 100;

        if      nCtbDegree = 0 then
                nCtbDegree := null;
        end if;

        return nCtbDegree;

end;

-------------------------------------------------------
-- Contribution Ratio Calculation
-------------------------------------------------------
function f_CtbRatioCal (
        inCD                                                                    i
n       number,
        inCDTtl                                                         in      num
ber) return number is

        nCtbRatio               number;

begin
        if      inCDTtl = 0 then
                nCtbRatio := null;
                return nCtbRatio;
        end if;

        nCtbRatio := inCD / inCDTtl * 100;

        if      nCtbRatio = 0 then
                nCtbRatio := null;
        end if;

        return nCtbRatio;

end;

-------------------------------------------------------
-- Industry In All Calculation
-------------------------------------------------------
procedure p_IndustryInAll (
        inSeq                                                   in      number,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2,
        inComparisonTarget      in      varchar2) is

-- Record Area Define
        recid           Indices%rowtype;
        recidp  Indices%rowtype;
        recidpi Indices%rowtype;
        recc1           ContributionForAll1%rowtype;
        recc1t  ContributionForAll1%rowtype;
        recwa           WeightAggregation%rowtype;
        recwat  WeightAggregation%rowtype;

-- Work Area
        nPM             number;
        nPY             number;

-- Indices Table Cursor
        Cursor  csrid is
                select  ISIC, IndexPR1, IndexSP2, IndexIV3, IndexIR4, IndexCU5, IndexLP6
                        from  Indices
                        where   AggregationRange = '3'
                                and     Year = inYear
                                and     Month = inMonth
                                and     SurveyScope = inSurveyScope
                                and     Pre_Rev_Mark = inPre_Rev_Mark;
```

```
-- Contribution For All 1 Cursor
        Cursor  csrc1 is
                select  CD_PR1, CD_SP2, CD_IV3, CD_IR4, CD_CU5, CD_LP6
                        from  ContributionForAll1
                        where Seq = inSeq
                                and     ComparisonTarget = inComparisonTarget
                                for update;

Begin
--      DBMS_OUTPUT.PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Get Pre Year and Month
        if      inComparisonTarget = '0'        then
                if      inMonth = 1 then
                        nPY     := inYear - 1;
                        nPM     := 12;
                else
                        nPY     := inYear;
                        nPM     := inMonth - 1;
                end if;
        else
                        nPY     := inYear - 1;
                        nPM     := inMonth;
        end if;

-- Get Weight Aggregation Total
        begin
                Select  WeightPR1, WeightSP2, WeightIV3, WeightIR4, WeightCU5, WeightLP6
                        into    recwat.WeightPR1, recwat.WeightSP2, recwat.WeightIV3,
                                                recwat.WeightIR4, recwat.WeightCU5, recwat.W
eightLP6
                        from    WeightAggregation
                        where   AggregationRange = '1'
                                and     SurveyScope = inSurveyScope
                                and     Version = '1';

        exception
                when no_data_found then
                        recwat.WeightPR1 := null;
                        recwat.WeightSP2 := null;
                        recwat.WeightIV3 := null;
                        recwat.WeightIR4 := null;
                        recwat.WeightCU5 := null;
                        recwat.WeightLP6 := null;
        end;

-- Get Pre Integrated Data Selecting
        begin
                select  IndexPR1, IndexSP2, IndexIV3,
                                IndexIR4, IndexCU5, IndexLP6
                        into    recidpi.IndexPR1, recidpi.IndexSP2, recidpi.IndexIV3,
                                recidpi.IndexIR4, recidpi.IndexCU5, recidpi.IndexLP6
                        from  Indices
                        where AggregationRange = '1'
                                and Year = nPY
                                and     Month = nPM
                                and SurveyScope = inSurveyScope
                                and     Pre_Rev_Mark = '2';

        exception
                when no_data_found then
                        recidpi.IndexPR1        := 0;
      recidpi.IndexSP2  := 0;
      recidpi.IndexIV3  := 0;
      recidpi.IndexIR4  := 0;
      recidpi.IndexCU5  := 0;
      recidpi.IndexLP6  := 0;
        end;

-- Indices Cursor Open Fetch
        open    csrid;
        fetch   csrid
                into    recid.ISIC,
                                        recid.IndexPR1, recid.IndexSP2, recid.IndexIV3,
                                        recid.IndexIR4, recid.IndexCU5, recid.IndexLP6;

-- Indices Cursor Fetch Looping
```

```
        while csrid%found loop

-- Pre Data Selecting
                begin
                        select  IndexPR1, IndexSP2, IndexIV3, IndexIR4, IndexCU5, IndexLP6
                        into            recidp. IndexPR1, recidp. IndexSP2, recidp. IndexIV3,
                                recidp. IndexIR4, recidp. IndexCU5, recidp. IndexLP6
                        from            Indices
                        where   AggregationRange = '3'
                        and             ISIC = recid. ISIC
                        and             Year = nPY
                        and             Month = nPM
                        and     SurveyScope = inSurveyScope
                                and             Indices. Pre_Rev_Mark = '2';

                exception
                        when no_data_found then
                                recidp. IndexPR1 := 0;
        recidp. IndexSP2 := 0;
        recidp. IndexIV3 := 0;
        recidp. IndexIR4 := 0;
        recidp. IndexCU5 := 0;
        recidp. IndexLP6 := 0;
                end;

-- Weight ISIC Selecting
                begin
                        Select  WeightPR1, WeightSP2, WeightIV3, WeightIR4, WeightCU5, WeightLP6
                                into    recwa. WeightPR1, recwa. WeightSP2, recwa. WeightIV3,
                                                        recwa. WeightIR4, recwa. WeightCU5, re
cwa. WeightLP6
                                from    WeightAggregation
                                where AggregationRange = '3'
                                        and     ISIC = recid. ISIC
                                        and     SurveyScope = inSurveyScope
                                        and     Version = '1';

                exception
                        when no_data_found then
                                recwa. WeightPR1 := null;
                                recwa. WeightSP2 := null;
                                recwa. WeightIV3 := null;
                                recwa. WeightIR4 := null;
                                recwa. WeightCU5 := null;
                                recwa. WeightLP6 := null;
                end;

-- Contribution Degree Calculation
                recc1. CD_PR1 := f_CtbDegreeCal(recid. IndexPR1, recidp. IndexPR1,
                        recwa. WeightPR1, recidpi. IndexPR1, recwat. WeightPR1);
                recc1. CD_SP2 := f_CtbDegreeCal(recid. IndexSP2, recidp. IndexSP2,
                        recwa. WeightSP2, recidpi. IndexSP2, recwat. WeightSP2);
                recc1. CD_IV3 := f_CtbDegreeCal(recid. IndexIV3, recidp. IndexIV3,
                        recwa. WeightIV3, recidpi. IndexIV3, recwat. WeightIV3);
                recc1. CD_IR4 := f_CtbDegreeCal(recid. IndexIR4, recidp. IndexIR4,
                        recwa. WeightIR4, recidpi. IndexIR4, recwat. WeightIR4);
                recc1. CD_CU5 := f_CtbDegreeCal(recid. IndexCU5, recidp. IndexCU5,
                        recwa. WeightCU5, recidpi. IndexCU5, recwat. WeightCU5);
                recc1. CD_LP6 := f_CtbDegreeCal(recid. IndexLP6, recidp. IndexLP6,
                        recwa. WeightLP6, recidpi. IndexLP6, recwat. WeightLP6);

-- Insert Contribution Table
                Insert
                        into            ContributionForAll1 (
                                                        ISIC,
                                                        Year,
                                                        Month,
                                                        SurveyScope,
                                                        Pre_Rev_mark,
                                                        ComparisonTarget,
                                                        Seq,
                                                        CD_PR1,
                                                        CD_SP2,
                                                        CD_IV3,
                                                        CD_IR4,
                                                        CD_CU5,
                                                        CD_LP6)
```

```
                        Values (recid.ISIC,
                                                        inYear,
                                                        inMonth,
                                                        inSurveyScope,
                                                        inPre_Rev_mark,
                                                        inComparisonTarget,
                                                        inSeq,
                                                        recc1.CD_PR1,
                                                        recc1.CD_SP2,
                                                        recc1.CD_IV3,
                                                        recc1.CD_IR4,
                                                        recc1.CD_CU5,
                                                        recc1.CD_LP6);

-- Fetch Indices Table
                        fetch           csrid
                                into    recid.ISIC,
                                                        recid.IndexPR1, recid.IndexSP2, reci
d.IndexIV3,
                                                        recid.IndexIR4, recid.IndexCU5, reci
d.IndexLP6;

                end loop;

        close csrid;

-- Get Total Contributio Degree
                select  sum(CD_PR1), sum(CD_SP2), sum(CD_IV3), sum(CD_IR4), sum(CD_CU5), sum(CD
_LP6)
                        into    recc1t.CD_PR1, recc1t.CD_SP2, recc1t.CD_IV3,
                                                recc1t.CD_IR4, recc1t.CD_CU5, recc1t.CD_LP6
                        from  ContributionForAll1
                        where Seq = inSeq
                                and     ComparisonTarget = inComparisonTarget;

-- Contribution Cursor Open Fetch
        open    csrc1;
        fetch   csrc1
                into    recc1.CD_PR1, recc1.CD_SP2, recc1.CD_IV3,
                                        recc1.CD_IR4, recc1.CD_CU5, recc1.CD_LP6;

-- Indices Cursor Fetch Looping
        while csrc1%found loop
-- Contribution Ratio Calcuration
                recc1.CR_PR1 := f_CtbRatioCal(recc1.CD_PR1, recc1t.CD_PR1);
                recc1.CR_SP2 := f_CtbRatioCal(recc1.CD_SP2, recc1t.CD_SP2);
                recc1.CR_IV3 := f_CtbRatioCal(recc1.CD_IV3, recc1t.CD_IV3);
                recc1.CR_IR4 := f_CtbRatioCal(recc1.CD_IR4, recc1t.CD_IR4);
                recc1.CR_CU5 := f_CtbRatioCal(recc1.CD_CU5, recc1t.CD_CU5);
                recc1.CR_LP6 := f_CtbRatioCal(recc1.CD_LP6, recc1t.CD_LP6);

-- Update Contribution Table
                update  ContributionForAll1
                        set             CR_PR1 = recc1.CR_PR1,
                                                CR_SP2 = recc1.CR_SP2,
                                                CR_IV3 = recc1.CR_IV3,
                                                CR_IR4 = recc1.CR_IR4,
                                                CR_CU5 = recc1.CR_CU5,
                                                CR_LP6 = recc1.CR_LP6
                        where   current of csrc1;

-- Contribution Cursor Fetch
                fetch   csrc1
                        into    recc1.CD_PR1, recc1.CD_SP2, recc1.CD_IV3,
                                                recc1.CD_IR4, recc1.CD_CU5, recc1.CD_LP6;

        end loop;

        close csrc1;

end;

-------------------------------------------------------
-- Commodity In All Calculation
-------------------------------------------------------
procedure p_CommInAll (
        inSeq                                                   in      number,
```

```
        inYear                                                  in      number,
        inMonth                                                 in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2,
        inComparisonTarget      in      varchar2) is

-- Record Area Define
        recid           Indices%rowtype;
        recidp  Indices%rowtype;
        recidpi Indices%rowtype;
        recc2           ContributionForAll2%rowtype;
        recc2t  ContributionForAll2%rowtype;
        recwa           WeightAggregation%rowtype;
        recwat  WeightAggregation%rowtype;

-- Work Area
        nPM             number;
        nPY             number;

-- Indices Table Cursor
        Cursor  csrid is
                select  ISIC, ItemType, IndexCommodityCode, IndexPR1, IndexSP2, IndexIV3, IndexI
R4, IndexCU5, IndexLP6
                        from  Indices
                        where   AggregationRange = '5'
                                and     Year = inYear
                                and     Month = inMonth
                                and     SurveyScope = inSurveyScope
                                and     Pre_Rev_Mark = inPre_Rev_Mark;

-- Contribution For All 2 Cursor
        Cursor  csrc2 is
                select  CD_PR1, CD_SP2, CD_IV3, CD_IR4, CD_CU5, CD_LP6
                        from  ContributionForAll2
                        where Seq = inSeq
                                and     ComparisonTarget = inComparisonTarget
                                for update;

Begin
--      DBMS_OUTPUT. PUT_LINE('Start');
-- Get Pre Year and Month
        if      inComparisonTarget = '0'        then
                if      inMonth = 1 then
                        nPY     := inYear - 1;
                        nPM     := 12;
                else
                        nPY     := inYear;
                        nPM     := inMonth - 1;
                end if;
        else
                        nPY     := inYear - 1;
                        nPM     := inMonth;
        end if;

-- Get Weight Total
        begin
                Select  WeightPR1, WeightSP2, WeightIV3, WeightIR4, WeightCU5, WeightLP6
                        into    recwat. WeightPR1, recwat. WeightSP2, recwat. WeightIV3,
                                                recwat. WeightIR4, recwat. WeightCU5, recwat. W
eightLP6
                        from    WeightAggregation
                        where AggregationRange = '1'
                                and     SurveyScope = inSurveyScope
                                and     Version = '1';

        exception
                when no_data_found then
                        recwat. WeightPR1 := null;
                        recwat. WeightSP2 := null;
                        recwat. WeightIV3 := null;
                        recwat. WeightIR4 := null;
                        recwat. WeightCU5 := null;
                        recwat. WeightLP6 := null;
        end;

-- Get Pre Integrated Data Selecting
        begin
```

```
                select  IndexPR1, IndexSP2, IndexIV3,
                                IndexIR4, IndexCU5, IndexLP6
                        into    recidpi. IndexPR1, recidpi. IndexSP2, recidpi. IndexIV3,
                                recidpi. IndexIR4, recidpi. IndexCU5, recidpi. IndexLP6
                        from  Indices
                        where AggregationRange = '1'
                                and Year = nPY
                                and     Month = nPM
                                and SurveyScope = inSurveyScope
                                and     Pre_Rev_Mark = '2';

                exception
                        when no_data_found then
                                recidpi. IndexPR1        := 0;
        recidpi. IndexSP2       := 0;
        recidpi. IndexIV3       := 0;
        recidpi. IndexIR4       := 0;
        recidpi. IndexCU5       := 0;
        recidpi. IndexLP6       := 0;
                end;

-- Indices Cursor Open Fetch
        open    csrid;
        fetch   csrid
                into    recid. ISIC, recid. ItemType, recid. IndexCommodityCode,
                                        recid. IndexPR1, recid. IndexSP2, recid. IndexIV3,
                                        recid. IndexIR4, recid. IndexCU5, recid. IndexLP6;

-- Indices Cursor Fetch Looping
        while csrid%found loop

-- Pre Data Selecting
                begin
                        select  IndexPR1, IndexSP2, IndexIV3, IndexIR4, IndexCU5, IndexLP6
                        into            recidp. IndexPR1, recidp. IndexSP2, recidp. IndexIV3,
                                recidp. IndexIR4, recidp. IndexCU5, recidp. IndexLP6
                        from            Indices
                        where   AggregationRange = '5'
                        and             ISIC = recid. ISIC
                        and             ItemType = recid. ItemType
                        and             IndexCommodityCode = recid. IndexCommodityCode
                        and             Year = nPY
                        and             Month = nPM
                        and     SurveyScope = inSurveyScope
                                and             Indices. Pre_Rev_Mark = '2';

                exception
                        when no_data_found then
                                recidp. IndexPR1 := 0;
        recidp. IndexSP2 := 0;
        recidp. IndexIV3 := 0;
        recidp. IndexIR4 := 0;
        recidp. IndexCU5 := 0;
        recidp. IndexLP6 := 0;
                end;

-- Weight Selecting by Commodity Level.
                begin
                        Select  WeightPR1, WeightSP2, WeightIV3, WeightIR4, WeightCU5, WeightLP6
                                into    recwa. WeightPR1, recwa. WeightSP2, recwa. WeightIV3,
                                                        recwa. WeightIR4, recwa. WeightCU5, re
cwa. WeightLP6
                                from    WeightAggregation
                                where AggregationRange = '5'
                                        and     ISIC = recid. ISIC
                                and     ItemType = recid. ItemType
                                and     CommodityCode = recid. IndexCommodityCode
                                and     SurveyScope = inSurveyScope
                                and     Version = '1';

                exception
                        when no_data_found then
                                recwa. WeightPR1 := null;
                                recwa. WeightSP2 := null;
                                recwa. WeightIV3 := null;
                                recwa. WeightIR4 := null;
                                recwa. WeightCU5 := null;
```

```
                                recwa.WeightLP6 := null;
                end;

-- Contribution Degree Calculation
                recc2.CD_PR1 := f_CtbDegreeCal(recid.IndexPR1, recidp.IndexPR1,
                        recwa.WeightPR1, recidpi.IndexPR1, recwat.WeightPR1);
                recc2.CD_SP2 := f_CtbDegreeCal(recid.IndexSP2, recidp.IndexSP2,
                        recwa.WeightSP2, recidpi.IndexSP2, recwat.WeightSP2);
                recc2.CD_IV3 := f_CtbDegreeCal(recid.IndexIV3, recidp.IndexIV3,
                        recwa.WeightIV3, recidpi.IndexIV3, recwat.WeightIV3);
                recc2.CD_IR4 := f_CtbDegreeCal(recid.IndexIR4, recidp.IndexIR4,
                        recwa.WeightIR4, recidpi.IndexIR4, recwat.WeightIR4);
                recc2.CD_CU5 := f_CtbDegreeCal(recid.IndexCU5, recidp.IndexCU5,
                        recwa.WeightCU5, recidpi.IndexCU5, recwat.WeightCU5);
                recc2.CD_LP6 := f_CtbDegreeCal(recid.IndexLP6, recidp.IndexLP6,
                        recwa.WeightLP6, recidpi.IndexLP6, recwat.WeightLP6);

-- Insert Contribution Table
                insert
                        into            ContributionForAll2 (
                                                        ISIC,
                                                        ItemType,
                                                        CommodityCode,
                                                        Year,
                                                        Month,
                                                        SurveyScope,
                                                        Pre_Rev_mark,
                                                        ComparisonTarget,
                                                        Seq,
                                                        CD_PR1,
                                                        CD_SP2,
                                                        CD_IV3,
                                                        CD_IR4,
                                                        CD_CU5,
                                                        CD_LP6)
                        Values (recid.ISIC,
                                                        recid.ItemType,
                                                        recid.IndexCommodityCode,

                                                        inYear,
                                                        inMonth,
                                                        inSurveyScope,
                                                        inPre_Rev_mark,
                                                        inComparisonTarget,
                                                        inSeq,
                                                        recc2.CD_PR1,
                                                        recc2.CD_SP2,
                                                        recc2.CD_IV3,
                                                        recc2.CD_IR4,
                                                        recc2.CD_CU5,
                                                        recc2.CD_LP6);

-- Fetch Indices Table
                        fetch           csrid
                                into    recid.ISIC, recid.ItemType, recid.IndexCommodityCode,
                                                        recid.IndexPR1, recid.IndexSP2, reci
d.IndexIV3,
                                                        recid.IndexIR4, recid.IndexCU5, reci
d.IndexLP6;

                end loop;

        close csrid;

-- Get Total Contributio Degree
                select  sum(CD_PR1), sum(CD_SP2), sum(CD_IV3), sum(CD_IR4), sum(CD_CU5), sum(CD
_LP6)
                        into    recc2t.CD_PR1, recc2t.CD_SP2, recc2t.CD_IV3,
                                                recc2t.CD_IR4, recc2t.CD_CU5, recc2t.CD_LP6
                        from  ContributionForAll2
                        where Seq = inSeq
                                and     ComparisonTarget = inComparisonTarget;

-- Contribution Cursor Open Fetch
        open    csrc2;
        fetch   csrc2
                into    recc2.CD_PR1, recc2.CD_SP2, recc2.CD_IV3,
```

```
                                        recc2.CD_IR4, recc2.CD_CU5, recc2.CD_LP6;

-- Indices Cursor Fetch Looping
        while csrc2%found loop

-- Contribution Ratio Calcuration
                recc2.CR_PR1 := f_CtbRatioCal(recc2.CD_PR1, recc2t.CD_PR1);
                recc2.CR_SP2 := f_CtbRatioCal(recc2.CD_SP2, recc2t.CD_SP2);
                recc2.CR_IV3 := f_CtbRatioCal(recc2.CD_IV3, recc2t.CD_IV3);
                recc2.CR_IR4 := f_CtbRatioCal(recc2.CD_IR4, recc2t.CD_IR4);
                recc2.CR_CU5 := f_CtbRatioCal(recc2.CD_CU5, recc2t.CD_CU5);
                recc2.CR_LP6 := f_CtbRatioCal(recc2.CD_LP6, recc2t.CD_LP6);

-- Update Contribution Table
                update  ContributionForAll2
                        set             CR_PR1 = recc2.CR_PR1,
                                                CR_SP2 = recc2.CR_SP2,
                                                CR_IV3 = recc2.CR_IV3,
                                                CR_IR4 = recc2.CR_IR4,
                                                CR_CU5 = recc2.CR_CU5,
                                                CR_LP6 = recc2.CR_LP6
                        where   current of csrc2;

-- Contribution Cursor Fetch
                fetch   csrc2
                        into    recc2.CD_PR1, recc2.CD_SP2, recc2.CD_IV3,
                                                recc2.CD_IR4, recc2.CD_CU5, recc2.CD_LP6;

        end loop;

        close csrc2;

end;

-------------------------------------------------------
-- Commodity In Industry
-------------------------------------------------------
procedure p_CommInIndustry (
        inSeq                                                   in      number,
        inAggregationRange      in      varchar2,
        inISIC                                          in      varchar2,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2,
        inComparisonTarget      in      varchar2) is

-- Record Area Define
        recid           Indices%rowtype;
        recidp  Indices%rowtype;
        recidpi Indices%rowtype;
        recci           ContributionForIndustry%rowtype;
        reccit  ContributionForIndustry%rowtype;
        recwa           WeightAggregation%rowtype;
        recwat  WeightAggregation%rowtype;

-- Work Area
        nPM             number;
        nPY             number;

        sISIC2d         Varchar2(2);
        sISIC4d         Varchar2(4);

        sOldISIC        Varchar2(4);

-- Indices Table Cursor
        Cursor  csrid is
                select  ISIC, ItemType, IndexCommodityCode,
                                                IndexPR1, IndexSP2, IndexIV3, IndexIR4, IndexC
U5, IndexLP6
                        from  Indices
                        where   AggregationRange = '5'
                                and     Year = inYear
                                and     Month = inMonth
                                and     SurveyScope = inSurveyScope
                                and     Pre_Rev_Mark = inPre_Rev_Mark
                        order by ISIC;
```

```
-- Contribution For Industry Cursor
        Cursor  csrci is
                select  ISIC, CD_PR1, CD_SP2, CD_IV3, CD_IR4, CD_CU5, CD_LP6
                        from  ContributionForIndustry
                        where Seq = inSeq
                                and     ComparisonTarget = inComparisonTarget
                        order by ISIC
                                for update;

Begin
--      DBMS_OUTPUT. PUT_LINE('Start');
-- Get Pre Year and Month
        if      inComparisonTarget = '0'        then
                if      inMonth = 1 then
                        nPY     := inYear - 1;
                        nPM     := 12;
                else
                        nPY     := inYear;
                        nPM     := inMonth - 1;
                end if;
        else
                        nPY     := inYear - 1;
                        nPM     := inMonth;
        end if;

        sISIC2d := substr(inISIC, 1, 2);
        sISIC4d := substr(inISIC, 1, 4);

        sOldISIC := null;

-- Indices Cursor Open Fetch
        open    csrid;
        fetch   csrid
                into    recid. ISIC, recid. ItemType, recid. IndexCommodityCode,
                                        recid. IndexPR1, recid. IndexSP2, recid. IndexIV3,
                                        recid. IndexIR4, recid. IndexCU5, recid. IndexLP6;

-- Indices Cursor Fetch Looping
        while csrid%found loop

-- Check Aggregation Range
                if      (inAggregationRange = '1')
                or      (inAggregationRange = '2' and substr(recid. ISIC, 1, 2) = sISIC2d)
                or      (inAggregationRange = '3' and substr(recid. ISIC, 1, 4) = sISIC4d) th
en
                        if      sOldISIC = substr(recid. ISIC, 1, 4) then
                                null;
                        else
                                sOldISIC := substr(recid. ISIC, 1, 4);

-- Get Weight Total
                                begin
                                        Select  WeightPR1, WeightSP2, WeightIV3, WeightIR4, We
ightCU5, WeightLP6
                                                into    recwat. WeightPR1, recwat. WeightSP2,
recwat. WeightIV3,
                                                                        recwat. WeightIR4, re
cwat. WeightCU5, recwat. WeightLP6
                                                from    WeightAggregation
                                                where   AggregationRange = '3'
                                                        and     ISIC like substr(recid. ISI
C, 1, 4) || '%'
                                                        and     SurveyScope = inSurveyScope
                                                        and     Version = '1';

                                exception
                                        when no_data_found then
                                                recwat. WeightPR1 := null;
                                                recwat. WeightSP2 := null;
                                                recwat. WeightIV3 := null;
                                                recwat. WeightIR4 := null;
                                                recwat. WeightCU5 := null;
                                                recwat. WeightLP6 := null;
                                end;

-- Get Pre Integrated Data Selecting
```

```
                                begin
                                        select  IndexPR1, IndexSP2, IndexIV3,
                                                        IndexIR4, IndexCU5, IndexLP6
                                                into    recidpi. IndexPR1, recidpi. IndexSP2,
recidpi. IndexIV3,
                                                        recidpi. IndexIR4, recidpi. IndexCU5,
recidpi. IndexLP6
                                                from  Indices
                                                where AggregationRange = '3'
                                                        and     ISIC like substr(recid. ISI
C, 1, 4) || '%'
                                                        and Year = nPY
                                                        and     Month = nPM
                                                        and SurveyScope = inSurveyScope
                                                        and     Pre_Rev_Mark = '2';
                                exception
                                        when no_data_found then
                                                recidpi. IndexPR1        := 0;
                        recidpi. IndexSP2       := 0;
                        recidpi. IndexIV3       := 0;
                        recidpi. IndexIR4       := 0;
                        recidpi. IndexCU5       := 0;
                        recidpi. IndexLP6       := 0;
                                end;
                        end if;

-- Pre Data Selecting
                        begin
                                select  IndexPR1, IndexSP2, IndexIV3, IndexIR4, IndexCU5, Index
LP6
                                        into    recidp. IndexPR1, recidp. IndexSP2, recidp. Ind
exIV3,
                                        recidp. IndexIR4, recidp. IndexCU5, recidp. IndexLP6
                                        from    Indices
                                        where AggregationRange = '5'
                                        and     ISIC = recid. ISIC
                                        and     ItemType = recid. ItemType
                                        and     IndexCommodityCode = recid. IndexCommodityC
ode
                                        and     Year = nPY
                                        and     Month = nPM
                                        and SurveyScope = inSurveyScope
                                                and     Indices. Pre_Rev_Mark = '2';

                        exception
                                when no_data_found then
                                        recidp. IndexPR1 := 0;
                recidp. IndexSP2 := 0;
                recidp. IndexIV3 := 0;
                recidp. IndexIR4 := 0;
                recidp. IndexCU5 := 0;
                recidp. IndexLP6 := 0;
                        end;

-- Weight Selecting by Commodity Level.
                        begin
                                Select  WeightPR1, WeightSP2, WeightIV3, WeightIR4, WeightCU5,
WeightLP6
                                        into    recwa. WeightPR1, recwa. WeightSP2, recwa. Weig
htIV3,
                                                                        recwa. WeightIR4, recwa. Weig
htCU5, recwa. WeightLP6
                                        from    WeightAggregation
                                        where AggregationRange = '5'
                                                and     ISIC = recid. ISIC
                                        and     ItemType = recid. ItemType
                                        and     CommodityCode = recid. IndexCommodityCode
                                                and     SurveyScope = inSurveyScope
                                                and     Version = '1';

                        exception
                                when no_data_found then
                                        recwa. WeightPR1 := null;
                                        recwa. WeightSP2 := null;
                                        recwa. WeightIV3 := null;
                                        recwa. WeightIR4 := null;
                                        recwa. WeightCU5 := null;
```

```
                                        recwa.WeightLP6 := null;
                        end;

-- Contribution Degree Calculation
                        recci.CD_PR1 := f_CtbDegreeCal(recid.IndexPR1, recidp.IndexPR1,
                                recwa.WeightPR1, recidpi.IndexPR1, recwat.WeightPR1);
                        recci.CD_SP2 := f_CtbDegreeCal(recid.IndexSP2, recidp.IndexSP2,
                                recwa.WeightSP2, recidpi.IndexSP2, recwat.WeightSP2);
                        recci.CD_IV3 := f_CtbDegreeCal(recid.IndexIV3, recidp.IndexIV3,
                                recwa.WeightIV3, recidpi.IndexIV3, recwat.WeightIV3);
                        recci.CD_IR4 := f_CtbDegreeCal(recid.IndexIR4, recidp.IndexIR4,
                                recwa.WeightIR4, recidpi.IndexIR4, recwat.WeightIR4);
                        recci.CD_CU5 := f_CtbDegreeCal(recid.IndexCU5, recidp.IndexCU5,
                                recwa.WeightCU5, recidpi.IndexCU5, recwat.WeightCU5);
                        recci.CD_LP6 := f_CtbDegreeCal(recid.IndexLP6, recidp.IndexLP6,
                                recwa.WeightLP6, recidpi.IndexLP6, recwat.WeightLP6);

-- Insert Contribution Table
                        Insert
                                into    ContributionForIndustry (
                                                        ISIC,
                                                        ItemType,
                                                        CommodityCode,
                                                        Year,
                                                        Month,
                                                        SurveyScope,
                                                        Pre_Rev_mark,
                                                        ComparisonTarget,
                                                        Seq,
                                                        CD_PR1,
                                                        CD_SP2,
                                                        CD_IV3,
                                                        CD_IR4,
                                                        CD_CU5,
                                                        CD_LP6)
                        Values (recid.ISIC,
                                                        recid.ItemType,
                                                        recid.IndexCommodityCode,

                                                        inYear,
                                                        inMonth,
                                                        inSurveyScope,
                                                        inPre_Rev_mark,
                                                        inComparisonTarget,
                                                        inSeq,
                                                        recci.CD_PR1,
                                                        recci.CD_SP2,
                                                        recci.CD_IV3,
                                                        recci.CD_IR4,
                                                        recci.CD_CU5,
                                                        recci.CD_LP6);
                end if;

-- Fetch Indices Table
                fetch           csrid
                        into    recid.ISIC, recid.ItemType, recid.IndexCommodityCode,
                                recid.IndexPR1, recid.IndexSP2, recid.IndexIV3,
                                recid.IndexIR4, recid.IndexCU5, recid.IndexLP6;

        end loop;

        close csrid;

-- Initialize Process
        sOldISIC := null;

-- Contribution Cursor Open Fetch
        open    csrci;
        fetch   csrci
                into    recci.ISIC, recci.CD_PR1, recci.CD_SP2, recci.CD_IV3,
                                        recci.CD_IR4, recci.CD_CU5, recci.CD_LP6;

-- Indices Cursor Fetch Looping
        while csrci%found loop
                if      sOldISIC = substr(recci.ISIC, 1, 4) then
                        null;
                else
```

```
                        sOldISIC := substr(recci.ISIC, 1, 4);

-- Get Total Contributio Degree
                        select  sum(CD_PR1), sum(CD_SP2), sum(CD_IV3), sum(CD_IR4), sum(CD_CU
5), sum(CD_LP6)
                                into    reccit.CD_PR1, reccit.CD_SP2, reccit.CD_IV3,
                                                reccit.CD_IR4, reccit.CD_CU5, recci
t.CD_LP6
                                from  ContributionForIndustry
                                where Seq = inSeq
                                        and     ISIC like substr(recci.ISIC, 1, 4) || '%'
                                        and     ComparisonTarget = inComparisonTarget;
                end if;

-- Contribution Ratio Calcuration
                recci.CR_PR1 := f_CtbRatioCal(recci.CD_PR1, reccit.CD_PR1);
                recci.CR_SP2 := f_CtbRatioCal(recci.CD_SP2, reccit.CD_SP2);
                recci.CR_IV3 := f_CtbRatioCal(recci.CD_IV3, reccit.CD_IV3);
                recci.CR_IR4 := f_CtbRatioCal(recci.CD_IR4, reccit.CD_IR4);
                recci.CR_CU5 := f_CtbRatioCal(recci.CD_CU5, reccit.CD_CU5);
                recci.CR_LP6 := f_CtbRatioCal(recci.CD_LP6, reccit.CD_LP6);

-- Update Contribution Table
                update  ContributionForIndustry
                        set             CR_PR1 = recci.CR_PR1,
                                                CR_SP2 = recci.CR_SP2,
                                                CR_IV3 = recci.CR_IV3,
                                                CR_IR4 = recci.CR_IR4,
                                                CR_CU5 = recci.CR_CU5,
                                                CR_LP6 = recci.CR_LP6
                        where   current of csrci;

-- Contribution Cursor Fetch
                fetch   csrci
                        into    recci.ISIC, recci.CD_PR1, recci.CD_SP2, recci.CD_IV3,
                                                recci.CD_IR4, recci.CD_CU5, recci.CD_LP6;

        end loop;

        close csrci;

end;

/* ==========================================================
   Calculate Contribution for each Establishment
========================================================== */
function FCalCtb (
        ppCurAV                         in      number,
  ppPreAV                               in      number,
  ppCurTtl                      in      number,
  ppPreTtl                      in      number,
        ppCurLink                       in      number,
        ppPreLink                       in      number,
        ppCurDeflator   in      number,
        ppPreDeflator   in      number) return number is

  nResult               number;
  nBalAV                number;
  nBalTtl               number;

BEGIN
--      DBMS_OUTPUT.PUT_LINE('CurAV = '|| ppCurAV || ' PreAV = ' || ppPreAV);
--      DBMS_OUTPUT.PUT_LINE('CurTtl = '|| ppCurTtl || ' PreTtl = ' || ppPreTtl);
--      DBMS_OUTPUT.PUT_LINE('CurLink = '|| ppCurLink || ' PreLink = ' || ppPreLink);
--      DBMS_OUTPUT.PUT_LINE('CurDef = '|| ppCurDeflator || ' PreDef = ' || ppPreDeflator);

        nBalAV  := ppPreAV * ppPreLink / ppPreDeflator - ppCurAV * ppCurLink / ppCurDeflat
or;
        nBalTtl := ppPreTtl * ppPreLink  / ppPreDeflator - ppCurTtl * ppCurLink / ppCurDef
lator;

        if      nBalTtl = 0 then
                nResult := null;
                return nResult;
        end if;

  nResult := (nBalAV / nBalTtl) * 100;
```

```
        if      nResult = 0 then
                nResult := null;
        end if;

        return nResult;

END;

/* ==========================================================
    Contribution Calculation(Establishment by Commodity)
========================================================== */
PROCEDURE p_EstInComm (
        inSeq                                   IN  NUMBER,
  inAggregationRange    IN      Varchar2,
  inISIC                                                        IN      Varchar2,
  inYear                                IN  NUMBER,
  inMonth                               IN  NUMBER,
  inPre_Rev_Mark        IN  VARCHAR2,
  inSurveyScope                         IN  VARCHAR2,
  inComparisonTarget    IN      VARCHAR2) IS

  ISIC_wk         ContributionForCommodity.ISIC%TYPE;
  ItemType_wk     ContributionForCommodity.ItemType%TYPE;
  Com_wk          ContributionForCommodity.CommodityCode%TYPE;
  Reg_wk          ContributionForCommodity.RegistrationNo%TYPE;

  nPreMonth       NUMBER;
  nPreYear        NUMBER;

  sISIC2d                                       Varchar2(2);
        sISIC4d                                 Varchar2(4);

        sOldISIC                                        varchar2(6);
        sOldItemType                    varchar2(1);
        sOldCommodityCode       varchar2(3);

        recIm                                           LinkMaster%rowtype;
        recImp                                  LinkMaster%rowtype;
        recdf                                           DeflatorMaster%rowtype;
        recdfp                                  DeflatorMaster%rowtype;
        reccc                                           ContributionForCommodity%rowtype;
        recpr                                           PastRecord%rowtype;
        recprp                                  PastRecord%rowtype;
        recprs                                  PastRecord%rowtype;
        recprsp                                 PastRecord%rowtype;


  CURSOR CsrCom IS
        SELECT  CM.ISIC, CM.Itemtype, CM.CommodityCode, EM.RegistrationNo
                FROM    CommodityMaster CM, EstablishmentMaster EM
                WHERE CM.ISIC = EM.ISIC
                        AND CM.ItemType in ('1','4')
                        AND EM.SurveyScope <= inSurveyScope
                order by CM.ISIC, CM.Itemtype, CM.CommodityCode;

BEGIN
-- DBMS_OUTPUT.ENABLE(600000);
--      Previous Year or Month Get
        if      inComparisonTarget = '0' then
    if inMonth = 1 THEN
      nPreMonth := 12;
      nPreYear := inYear - 1;
    else
      nPreMonth := inMonth - 1;
      nPreYear := inYear;
    end if;
        else
        nPreMonth := inMonth;
    nPreYear := inYear - 1;
        end if;

-- ISIC Work Edit
  sISIC2d := substr(inISIC,1,2);
        sISIC4d := substr(inISIC,1,4);

-- Old Key Area Clear
```

```
        sOldISIC                                        := null;
        sOldItemType                    := null;
        sOldCommodityCode       := null;

-- Open Cursor and Loop Start
  OPEN CsrCom;
  LOOP

-- Fetch and Exit
    FETCH CsrCom INTO ISIC_wk, Itemtype_wk, Com_wk, Reg_wk;
    EXIT WHEN CsrCom%NOTFOUND;

-- Aggregation Range Checking
    if  (inAggregationRange = '1')
                or      (inAggregationRange = '2' and substr(ISIC_wk, 1, 2) = sISIC2d)
                or      (inAggregationRange = '3' and substr(ISIC_wk, 1, 4) = sISIC4d)
                or      (inAggregationRange = '4' and ISIC_wk = inISIC) then

-- Compair Key
                        if      sOldISIC                                        = ISIC_wk
                        and     sOldItemType                    = Itemtype_wk
                        and     sOldCommodityCode       = Com_wk then
                                null;
                        else

-- Set Old Key
                                sOldISIC                                        := ISIC_wk;
                                sOldItemType                    := Itemtype_wk;
                                sOldCommodityCode       := Com_wk;

-- Get Link Coefficient Data
                                begin
                                        select  *
                                                into    recIm
                                                from    LinkMaster
                                                where   ISIC
                = ISIC_wk
                                                        and Year
                        = inYear
                                                        and Month
                = inMonth
                                                        and ItemType                    = I
temtype_wk
                                                        and CommodityCode = Com_wk
                                                        and SurveyScope         = inSurveyS
cope
                                                        and     Version
                =       '1';
                                        select  *
                                                into    recImp
                                                from    LinkMaster
                                                where   ISIC
                = ISIC_wk
                                                        and Year
                        = nPreYear
                                                        and Month
                = nPreMonth
                                                        and ItemType                    = I
temtype_wk
                                                        and CommodityCode = Com_wk
                                                        and SurveyScope         = inSurveyS
cope
                                                        and     Version
                =       '1';
                                exception
                                        when no_data_found then
                                                recIm.ProductionQty             := 0;
                                                recIm.ShipmentQty
                := 0;
                                                recIm.ME_Inventory                      :=
0;
                                                recIm.ShipmentValue                     :=
0;
                                                recIm.RwMt_MEInventory  := 0;
                                                recImp.ProductionQty            := 0;
                                                recImp.ShipmentQty                      :=
0;
```

```
                                        recImp.ME_Inventory                     := 0;
                                        recImp.ShipmentValue            := 0;
                                        recImp.RwMt_MEInventory := 0;
                                end;

-- Deflator Master Select
                        begin
                                select  *
                                        into    recdf
                                        from    DeflatorMaster
                                        where   ISIC                                    = I
SIC_wk
                                                and     Year
                = inYear
                                                and     Month                           = i
nMonth
                                                and ItemType                    = Itemtype_
wk
                                                and     CommodityCode   = Com_wk
                                                and SurveyScope         = inSurveyScope
                                                and     Version                         = '
1';

                                select  *
                                        into    recdfp
                                        from    DeflatorMaster
                                        where   ISIC                                    = I
SIC_wk
                                                and Year
                = nPreYear
                                                and Month                               = n
PreMonth
                                                and ItemType                    = Itemtype_
wk
                                                and     CommodityCode   = Com_wk
                                                and SurveyScope         = inSurveyScope
                                                and     Version                         = '
1';

                        exception
                                when no_data_found then
                                        recdf.ProductionQty             := 1;
                                        recdf.ShipmentQty                               :=
1;
                                        recdf.ME_Inventory                      := 1;
                                recdf.Capacity                                          := 1;
                                        recdf.RwMt_MEInventory  := 1;
                                        recdfp.ProductionQty            := 1;
                                        recdfp.ShipmentQty                      := 1;
                                recdfp.ME_Inventory                     := 1;
                                recdfp.Capacity                                 := 1;
                                recdfp.RwMt_MEInventory := 1;
                        end;

-- Summary Of Commodity Level
                                if      Itemtype_wk = '1' then

-- This Month Total Past Record Select(Item Type 1)
                                        SELECT  SUM(PR.ProductionQTY), SUM(PR.DomesticSale
s), SUM(PR.Export),
                                                                        SUM(PR.ME_Inventory)
                                                INTO    recprs.ProductionQTY, recprs.Domest
icSales, recprs.Export,
                                                                        recprs.ME_Inventory
                                                FROM    PastRecord PR,  EstablishmentMaster
EM
                                                WHERE   PR.ISIC = ISIC_wk
                                                        AND  PR.Year = inYear
                                                        AND  PR.Month = inMonth
                                                  AND   PR.ItemType = '1'
                                                  AND   PR.CommodityCode = Com_wk
                                                  AND   PR.Pre_Rev_Mark  = inPre_Rev_Mark
                                                  AND   PR.ISIC = EM.ISIC
                                                  AND   PR.RegistrationNo = EM.Registration
No
                                                  AND   EM.SurveyScope <= inSurveyScope;
```

```
-- This Month Total Past Record Select(Item Type 2)
                                SELECT        SUM(PR. ShipmentValue)
                                      INTO    recprs. ShipmentValue
                                      FROM    PastRecord PR, EstablishmentMaster EM
                                      WHERE PR. ISIC = ISIC_wk
                                                      AND PR. Year = inYear
                                                      AND PR. Month = inMonth
                                              AND PR. ItemType = '2'
                                              AND PR. CommodityCode = Com_wk
                                              AND PR. Pre_Rev_Mark  = inPre_Rev_Mark
                                              AND PR. ISIC = EM. ISIC
                                              AND PR. RegistrationNo = EM. RegistrationNo
                                              AND     EM. SurveyScope <= inSurveyScope;

-- Pre Month Total Past Record Select(Item Type 1)
                                      SELECT  SUM(PR. ProductionQTY), SUM(PR. DomesticSale
s), SUM(PR. Export),
                                                                      SUM(PR. ME_Inventory)
                                              INTO    recprsp. ProductionQTY, recprsp. Dome
sticSales, recprsp. Export,
                                                                      recprsp. ME_Inventory
                                              FROM  PastRecord PR, EstablishmentMaster EM
                                              WHERE PR. ISIC = ISIC_wk
                                                      AND     PR. Year = nPreYear
                                                      AND PR. Month = nPreMonth
                                                AND PR. ItemType = '1'
                                                AND PR. CommodityCode = Com_wk
                                                AND PR. Pre_Rev_Mark  = '2'
                                                AND PR. ISIC = EM. ISIC
                                                AND PR. RegistrationNo = EM. RegistrationNo
                                                AND EM. SurveyScope <= inSurveyScope;

-- Pre Month Total Past Record Select(Item Type 2)
                                SELECT        SUM(PR. ShipmentValue)
                                      INTO    recprsp. ShipmentValue
                                      FROM    PastRecord PR, EstablishmentMaster EM
                                      WHERE PR. ISIC = ISIC_wk
                                              AND PR. Year = nPreYear
                                              AND PR. Month = nPreMonth
                                              AND PR. ItemType = '2'
                                              AND PR. CommodityCode = Com_wk
                                              AND PR. Pre_Rev_Mark  = '2'
                                              AND PR. ISIC = EM. ISIC
                                              AND PR. RegistrationNo = EM. RegistrationNo
                                              AND EM. SurveyScope <= inSurveyScope;

                              elsif Itemtype_wk = '4' then
-- This Month Total Past Record Select(Item Type 4)
                                SELECT        SUM(PR. RwMt_MEInventory)
                                      INTO    recprs. RwMt_MEInventory
                                      FROM    PastRecord PR, EstablishmentMaster EM
                                      WHERE   PR. ISIC = ISIC_wk
                                                      AND  PR. Year = inYear
                                                      AND  PR. Month = inMonth
                                              AND     PR. ItemType = '4'
                                              AND     PR. CommodityCode = Com_wk
                                              AND     PR. Pre_Rev_Mark  = inPre_Rev_Mark
                                              AND     PR. ISIC = EM. ISIC
                                              AND     PR. RegistrationNo = EM. Registratio
nNo
                                              AND     EM. SurveyScope <= inSurveyScope;

-- Pre Month Total Past Record Select(Item Type 4)
                                SELECT        SUM(PR. RwMt_MEInventory)
                                      INTO    recprsp. RwMt_MEInventory
                                      FROM    PastRecord PR, EstablishmentMaster EM
                                      WHERE   PR. ISIC = ISIC_wk
                                              AND     PR. Year = nPreYear
                                              AND     PR. Month = nPreMonth
                                              AND     PR. ItemType = '4'
                                              AND     PR. CommodityCode = Com_wk
                                              AND     PR. Pre_Rev_Mark  = '2'
                                              AND     PR. ISIC = EM. ISIC
                                              AND     PR. RegistrationNo = EM. Registratio
nNo
                                              AND     EM. SurveyScope <= inSurveyScope;
                              end if;
```

```
                        end if;

                if      Itemtype_wk = '1' then

-- Select This Month Past Record(Item Type 1)
                                begin
                                        SELECT  ProductionQTY, DomesticSales, Export, ME_Inve
ntory
                                                INTO    recpr. ProductionQTY, recpr. Domestic
Sales, recpr. Export, recpr. ME_Inventory
                                        FROM    PastRecord
                                        WHERE   ISIC = ISIC_wk
                                                AND     Year = inYear
                                        AND Month = inMonth
                                        AND RegistrationNo = Reg_wk
                                        AND ItemType = '1'
                                        AND CommodityCode = Com_wk
                                        AND Pre_Rev_Mark  = inPre_Rev_Mark;

                                        recpr. ProductionQTY     := nvl(recpr. ProductionQT
Y, 0);
                                        recpr. DomesticSales := nvl(recpr. DomesticSales, 0);
                                        recpr. Export                           := nvl(recp
r. Export, 0);
                                        recpr. ME_Inventory      := nvl(recpr. ME_Inventory,
0);

                                exception
                                        when no_data_found then
                                                recpr. ProductionQTY := 0;
                                                recpr. DomesticSales := 0;
                                                recpr. Export                           :=
0;
                                                recpr. ME_Inventory := 0;
                                END;

-- Select Pre Month Past Record(Item Type 1)
                                begin
                                        SELECT  ProductionQTY, DomesticSales, Export, ME_Inve
ntory
                                                INTO    recprp. ProductionQTY, recprp. Domest
icSales, recprp. Export, recprp. ME_Inventory
                                        FROM    PastRecord
                                        WHERE   ISIC = ISIC_wk
                                                AND     Year = nPreYear
                                        AND Month = nPreMonth
                                        AND RegistrationNo = Reg_wk
                                        AND ItemType = '1'
                                        AND CommodityCode = Com_wk
                                        AND Pre_Rev_Mark  = '2';

                                        recprp. ProductionQTY    := nvl(recprp. ProductionQT
Y, 0);
                                        recprp. DomesticSales    := nvl(recprp. DomesticSale
s, 0);
                                        recprp. Export                          :=
nvl(recprp. Export, 0);
                                        recprp. ME_Inventory             := nvl(recprp. ME_In
ventory, 0);

                                exception
                                        when no_data_found then
                                                recprp. ProductionQTY    := 0;
                                                recprp. DomesticSales    := 0;
                                                recprp. Export
                := 0;
                                                recprp. ME_Inventory             := 0;
                                END;

-- Select This Month Past Record(Item Type 2)
                                begin
                                        SELECT  ShipmentValue
                                                INTO    recpr. ShipmentValue
                                        FROM    PastRecord
                                        WHERE   ISIC = ISIC_wk
                                                AND     Year = inYear
                                        AND Month = inMonth
```

```
                                        AND RegistrationNo = Reg_wk
                                        AND ItemType = '2'
                                        AND CommodityCode = Com_wk
                                        AND Pre_Rev_Mark  = inPre_Rev_Mark;

                                        recpr.ShipmentValue := nvl(recpr.ShipmentValue,0);

                                exception
                                        when no_data_found then
                                                recpr.ShipmentValue := 0;
                                END;

-- Select Pre Month Past Record(Item Type 2)
                                begin
                                        SELECT  ShipmentValue
                                                INTO    recprp.ShipmentValue
                                        FROM    PastRecord
                                        WHERE   ISIC = ISIC_wk
                                                AND     Year = nPreYear
                                        AND Month = nPreMonth
                                        AND RegistrationNo = Reg_wk
                                        AND ItemType = '2'
                                        AND CommodityCode = Com_wk
                                        AND Pre_Rev_Mark  = '2';

                                        recprp.ShipmentValue := nvl(recprp.ShipmentValue,
0);

                                exception
                                        when no_data_found then
                                                recprp.ShipmentValue := 0;
                                END;

-- Calculation
                        reccc.CR_Production     := FCalCtb (recpr.ProductionQTY, recprp.Pro
ductionQTY,

                                                                                        recprs.Prod
uctionQTY, recprsp.ProductionQTY,

                                                                                        recim.Produ
ctionQty, recimp.ProductionQty,

                                                                                        recdf.Produ
ctionQty, recdfp.ProductionQty);
                        reccc.CR_ShipmentQty    := FCalCtb (recpr.DomesticSales + recpr.Ex
port,

                                                                                        recprp.Dome
sticSales + recprp.Export,

                                                                                        nvl(recprs.
DomesticSales,0) + nvl(recprs.Export,0),

                                                                                        nvl(recprs
p.DomesticSales,0) + nvl(recprsp.Export,0),

                                                                                        recim.Shipm
entQty, recimp.ShipmentQty,

                                                                                        recdf.Shipm
entQty, recdfp.ShipmentQty);
                        reccc.CR_MEInventory    := FCalCtb (recpr.ME_Inventory, recprp.ME_I
nventory,

                                                                                        recprs.ME_I
nventory, recprsp.ME_Inventory,

                                                                                        recim.ME_In
ventory, recimp.ME_Inventory,

                                                                                        recdf.ME_In
ventory, recdfp.ME_Inventory);
                        reccc.CR_ShipmentVal    := FCalCtb (recpr.ShipmentValue, recprp.Shi
pmentValue,

                                                                                        recprs.Ship
```

```
mentValue, recprsp. ShipmentValue,
                                                                                                    reclm. Shipm
entValue, reclmp. ShipmentValue, 1, 1);

--Insert Table(Item Type 1)
                INSERT
                        INTO    ContributionForCommodity(
                                ISIC, ItemType, CommodityCode, RegistrationNo, Year, Month,
                                SurveyScope, Pre_Rev_Mark, ComparisonTarget, Seq,
                                CR_Production,  CR_ShipmentQty,  CR_MEInventory, CR_ShipmentV
al)
                VALUES(
                        ISIC_wk, '1', Com_wk, Reg_wk, inYear, inMonth,
                        inSurveyScope, inPre_Rev_Mark, inComparisonTarget, inSeq,
                        reccc. CR_Production, reccc. CR_ShipmentQty,
                        reccc. CR_MEInventory, reccc. CR_ShipmentVal);

        elsif Itemtype_wk = '4' THEN

-- Select This Month Past Record(Item Type 4)
                                begin
                                        SELECT  RwMt_MEInventory
                                                INTO    recpr. RwMt_MEInventory
                                        FROM    PastRecord
                                        WHERE   ISIC = ISIC_wk
                                                AND     Year = inYear
                                        AND Month = inMonth
                                        AND RegistrationNo = Reg_wk
                                        AND ItemType = '4'
                                        AND CommodityCode = Com_wk
                                        AND Pre_Rev_Mark  = inPre_Rev_Mark;

                        recpr. RwMt_MEInventory := nvl(recpr. RwMt_MEInventory, 0);

                                exception
                                        when no_data_found then
                                                recpr. RwMt_MEInventory := 0;
                                END;

-- Select Pre Month Past Record(Item Type 4)
                                begin
                                        SELECT  RwMt_MEInventory
                                                INTO    recprp. RwMt_MEInventory
                                        FROM    PastRecord
                                        WHERE   ISIC = ISIC_wk
                                                AND     Year = nPreYear
                                        AND Month = nPreMonth
                                        AND RegistrationNo = Reg_wk
                                        AND ItemType = '4'
                                        AND CommodityCode = Com_wk
                                        AND Pre_Rev_Mark  = '2';

                        recprp. RwMt_MEInventory := nvl(recprp. RwMt_MEInventory, 0);

                                exception
                                        when no_data_found then
                                                recprp. RwMt_MEInventory := 0;
                                END;

-- Calculation
--      DBMS_OUTPUT. PUT_LINE(recpr. RwMt_MEInventory || recprp. RwMt_MEInventory
--                                                                                                    ||
recprs. RwMt_MEInventory || recprsp. RwMt_MEInventory);

                        reccc. CR_RawMaterial     := FCalCtb (recpr. RwMt_MEInventory, recprp.
RwMt_MEInventory,
                                                                                                    recprs. RwMt
_MEInventory, recprsp. RwMt_MEInventory,
                                                                                                    reclm. RwMt_
MEInventory, reclmp. RwMt_MEInventory,
                                                                                                    recdf. RwMt_
MEInventory, recdfp. RwMt_MEInventory);
```

```
--Insert Table(Item Type 4)
                                INSERT
                                        INTO    ContributionForCommodity(
                                        ISIC, ItemType, CommodityCode, RegistrationNo, Year, Mo
nth,
                                        SurveyScope, Pre_Rev_Mark, ComparisonTarget, Seq,
                                        CR_RawMaterial)
                        VALUES(
                                        ISIC_wk, '4', Com_wk, Reg_wk, inYear, inMonth,
                                        inSurveyScope, inPre_Rev_Mark, inComparisonTarget, in
Seq,
                                        reccc.CR_RawMaterial);

                        END IF;

        END IF;

  END LOOP;

  CLOSE CsrCom;

END p_EstInComm;

end CtbCal;
/
show errors
/
```

```
/* ============================================================
    Package Name GetSeq
            Create Date 1999/10/14
              By: Rattana J.
            Last Update 1999/11/04
              By: K. Shibamoto@JICA STUDY TEAM
    File name : @c:¥back¥GetSeq.sql
    Call string :   DECLARE AA NUMBER;
                    BEGIN GetSeq.SeqNo(:AA); END;
============================================================ */
CREATE OR REPLACE PACKAGE GetSeq IS

    PROCEDURE SeqNo (
        pSeqMax     OUT SeqControl.SeqMax%TYPE);

    PROCEDURE ClearTemp (
        pSeq        IN  NUMBER);

END;

/
show errors
/

CREATE OR REPLACE PACKAGE BODY GetSeq IS

/* ============================================================
    Main of Program
============================================================  */
    PROCEDURE SeqNo (
        pSeqMax     OUT SeqControl.SeqMax%TYPE)

    IS
        CURSOR Csr IS
            SELECT SeqMax
            FROM SeqControl
            WHERE OutputID = '00'
            FOR UPDATE;  -- LOCK THE FILE !!
        Rec Csr%ROWTYPE;
    BEGIN
        -- Select SeqMax from table
        OPEN Csr;
        FETCH Csr INTO Rec;

        -- Set the new value for SeqMax
            --DBMS_OUTPUT.PUT_LINE('before SeqMax=' ||pSeqMax);
        IF Rec.SeqMax >= 1000 THEN
            pSeqMax := 0;
        ELSE
            pSeqMax := Rec.SeqMax + 1;
        END IF;
        -- Update the new SeqMax in table
        UPDATE SeqControl
        SET SeqMax = pSeqMax
        WHERE CURRENT OF Csr;

        CLOSE Csr;
        COMMIT WORK;
             --DBMS_OUTPUT.PUT_LINE('After SeqMax=' ||pSeqMax);
    END;-- end of Main Procedure

/* ============================================================
    Clear temporary tables
============================================================  */
    PROCEDURE ClearTemp (
        pSeq        IN  NUMBER)

    IS
    BEGIN
        delete from ContributionForAll1 where Seq = pSeq;
        delete from ContributionForAll2 where Seq = pSeq;
        delete from ContributionForIndustry where Seq = pSeq;
        delete from ContributionForCommodity where Seq = pSeq;
        delete from IndexList where Seq = pSeq;
        delete from TimeFigures where Seq = pSeq;
        delete from CollectionRate where Seq = pSeq;
        delete from AnnualCollectionStatus where Seq = pSeq;
```

```
        delete from CPMYRawEst where Seq = pSeq;
        delete from CPMYRawISIC where Seq = pSeq;
    END;
END; -- end of package body
/
show errors
/
```

```
-- ==================================================
-- Package Name : Growth Rate Calcuration
-- Package ID   : GRCal
-- Create Date  : 9/Nov/1999
-- Replace Date : xx/xxx/xxxx
-- Author       : Nakamura
--
-- File         : @c:¥jica¥GRCal;
-- Test                                  :       BEGIN GRCal.p_GRQtyCal(1,'2','15XXXX',199
9,2,'1','2','0'); END;
-- Test                                  : BEGIN GRCal.p_GREleComm(1999,12,'2'); commit; EN
D;
-- Test                                  : BEGIN GRCal.p_GREleIndustry(1999,2,'2'); commit;
END;
-- Test                                  : BEGIN GRCal.p_GRIndices(2000,2,'1'); commit; END;
-- ==================================================
create or replace package GRCal is

procedure p_GRQtyCal (
        inSeq                                                   in      number,
        inAggregationRange      in      varchar2,
        inISIC                                          in      varchar2,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2,
        inComparisonTarget      in      varchar2);

procedure p_GREleIndustry (
        inYear                                          in      number,
        inMonth                                         in      number,
        inPre_Rev_Mark                  in      varchar2);

procedure p_GREleComm (
        inYear                                          in      number,
        inMonth                                         in      number,
        inPre_Rev_Mark                  in      varchar2);

procedure p_GRIndices(
  pYear                         in number,
  pMonth                        in number,
  pPreRev                       in varchar2);

end GRCal;
/
show errors
/

create or replace package body GRCal is
-------------------------------------------------------
-- Growth Rate Qty Table Insert
-------------------------------------------------------
procedure p_GRQtyIns (
        inISIC                                          in      varchar2,
        inItemType                                      in      varchar2,
        inCommodityCode                 in      varchar2,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_mark                  in      varchar2,
        inComparisonTarget      in      varchar2,
        inSeq                                                   in      number,
        inGR_Production                 in      number,
        inGR_ShipmentQty                in      number,
        inGR_MEInventory                in      number,
        inGR_ShipmentVal                in      number,
        inGR_RawMaterial                in      number) is

begin
        Insert
                into            GrowthRateQty (
                                                ISIC,
                                                ItemType,
                                                CommodityCode,
                                                Year,
                                                Month,
                                                SurveyScope,
```

```
                                                Pre_Rev_mark,
                                                ComparisonTarget,
                                                Seq,
                                                GR_Production,
                                                GR_ShipmentQty,
                                                GR_MEInventory,
                                                GR_ShipmentVal,
                                                GR_RawMaterial)
                Values (inISIC,
                                                inItemType,
                                                inCommodityCode,
                                                inYear,
                                                inMonth,
                                                inSurveyScope,
                                                inPre_Rev_mark,
                                                inComparisonTarget,
                                                inSeq,
                                                inGR_Production,
                                                inGR_ShipmentQty,
                                                inGR_MEInventory,
                                                inGR_ShipmentVal,
                                                inGR_RawMaterial);
end;

-------------------------------------------------------
-- Growth Rate Calculation
-------------------------------------------------------
function f_GrowthRate (
        inAVCurr                                                        in      number,
        inAVPre                                                         in      number,
        inLKCurr                                                        in      number,
        inLKPre                                                         in      number,
        inDFCurr                                                        in      number,
        inDFPre                                                         in      number) ret
urn number is

        nGrowthRate                                             number;

begin

        if      inAVPre = 0
        or      inLKPre = 0 then
                nGrowthRate := null;
                return nGrowthRate;
        end if;

        nGrowthRate :=  ((inAVCurr * inLKCurr / inDFCurr) - (inAVPre * inLKPre / inDFPre))
/                                                                       (inAVPre * inLKPre /
inDFPre) * 100;

        if      nGrowthRate = 0 then
                nGrowthRate := null;
        end if;

        return nGrowthRate;

end;

-------------------------------------------------------
-- Growth Rate Qty Calculation
-------------------------------------------------------
procedure p_GRQtyCal (
        inSeq                                                           in      number,
        inAggregationRange      in      varchar2,
        inISIC                                                  in      varchar2,
        inYear                                                  in      number,
        inMonth                                                 in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2,
        inComparisonTarget      in      varchar2) is

-- Record Area Define
        reccm           CommodityMaster%rowtype;
        reclm           LinkMaster%rowtype;
        reclmp  LinkMaster%rowtype;
        recdf           DeflatorMaster%rowtype;
```

```
        recdfp  DeflatorMaster%rowtype;
        recpr           PastRecord%rowtype;
        recprp  PastRecord%rowtype;
        recgq           GrowthRateQTY%rowtype;

-- Work Area
        nPY                     number;
        nPM                     number;

        sISIC2d         Varchar2(2);
        sISIC4d         Varchar2(4);

--  CommodityMaster Cursor
        Cursor  csrcm is
                select  ISIC, ItemType, CommodityCode
                        from  CommodityMaster
                        where   ItemType in('1','4');

Begin
--      DBMS_OUTPUT.ENABLE(60000);
--      DBMS_OUTPUT.PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Get Pre Year and Month
        if      inComparisonTarget = '0' then
                if      inMonth = 1 then
                        nPY     := inYear - 1;
                        nPM     := 12;
                else
                        nPY     := inYear;
                        nPM     := inMonth - 1;
                end if;
        else
                nPY     := inYear - 1;
                nPM     := inMonth;
        end if;

-- For Grouping Range Check
        sISIC2d := substr(inISIC, 1, 2);
        sISIC4d := substr(inISIC, 1, 4);

-- Commodity Master Cursor Open and Fetch
        open    csrcm;
        fetch   csrcm
                into    reccm.ISIC, reccm.ItemType, reccm.CommodityCode;
        while csrcm%found loop

-- Grouping Range Check
                if      (inAggregationRange = '1')
                or      (inAggregationRange = '2' and substr(reccm.ISIC, 1, 2) = sISIC2d)
                or      (inAggregationRate = '3' and substr(reccm.ISIC, 1, 4) = sISIC4d)
                or      (inAggregationRange = '4' and reccm.ISIC = inISIC) then

-- Get LinkMaster Coefficient(This Month and Pre Month)
                        begin
                                select  *
                                        into    recIm
                                        from    LinkMaster
                                        where   ISIC                                    = r
eccm.ISIC
                                                and     Year
                = inYear
                                                and     Month                           = i
nMonth
                                                and ItemType                    = reccm.Ite
mType
                                                and     CommodityCode   = reccm.CommodityCo
de
                                                and SurveyScope         = inSurveyScope
                                                and     Version                         = '
1';

                                select  *
                                        into    recImp
                                        from    LinkMaster
                                        where   ISIC                                    = r
eccm.ISIC
                                                and     Year
                = nPY
```

```
                                        and     Month                           = n
PM
                                        and ItemType                    = reccm.Ite
mType
                                        and     CommodityCode   = reccm.CommodityCo
de
                                        and SurveyScope         = inSurveyScope
                                        and     Version                         = '
1';

                exception
                        when no_data_found then
                                recim.ProductionQty             := 0;
        recim.ShipmentQty                               := 0;
        recim.ME_Inventory                      := 0;
        recim.ShipmentValue                     := 0;
        recim.RwMt_MEInventory  := 0;
                                recimp.ProductionQty            := 0;
        recimp.ShipmentQty                      := 0;
        recimp.ME_Inventory                     := 0;
        recimp.ShipmentValue            := 0;
        recimp.RwMt_MEInventory := 0;
                end;

-- Deflator Master Select
                begin
                        select  *
                                into    recdf
                                from    DeflatorMaster
                                where   ISIC                                    = r
eccm.ISIC
                                        and     Year
        = inYear
                                        and     Month                           = i
nMonth
                                        and ItemType                    = reccm.Ite
mType
                                        and     CommodityCode   = reccm.CommodityCo
de
                                        and SurveyScope         = inSurveyScope
                                        and     Version                         = '
1';

                        select  *
                                into    recdfp
                                from    DeflatorMaster
                                where   ISIC                                    = r
eccm.ISIC
                                        and     Year
        = nPY
                                        and     Month                           = n
PM
                                        and ItemType                    = reccm.Ite
mType
                                        and     CommodityCode   = reccm.CommodityCo
de
                                        and SurveyScope         = inSurveyScope
                                        and     Version                         = '
1';

                exception
                        when no_data_found then
                                recdf.ProductionQty             := 1;
                                recdf.ShipmentQty                               :=
1;
                                recdf.ME_Inventory                      := 1;
                        recdf.Capacity                                  := 1;
                                recdf.RwMt_MEInventory  := 1;
                                recdfp.ProductionQty            := 1;
                                recdfp.ShipmentQty                      := 1;
                        recdfp.ME_Inventory                     := 1;
                        recdfp.Capacity                                 := 1;
                        recdfp.RwMt_MEInventory := 1;
                end;

                if      reccm.ItemType = '1' then
-- Get Past Record(This Month, Item Type 1)
```

```
                                select  sum(ProductionQty), sum(DomesticSales), sum(Export),
sum(ME_Inventory)
                                        into    recpr.ProductionQty, recpr.DomesticSales, re
cpr.Export, recpr.ME_Inventory
                                        from    VW_PastRecordScope
                                        where   ISIC                                    = r
eccm.ISIC
                                                and     Year
                = inYear
                                                and     Month                           = i
nMonth
                                                and ItemType                    = '1'
                                                and     CommodityCode   = reccm.CommodityCo
de
                                                and     Pre_Rev_Mark    = inPre_Rev_mark
                                                and     SurveyScope     <= inSurveyScope;

-- Get Past Record(Pre Month and Year, Item Type 1)
                                select  sum(ProductionQty), sum(DomesticSales), sum(Export),
sum(ME_Inventory)
                                        into    recprp.ProductionQty, recprp.DomesticSales,
recprp.Export, recprp.ME_Inventory
                                        from    VW_PastRecordScope
                                        where   ISIC                                    = r
eccm.ISIC
                                                and     Year
                = nPY
                                                and     Month                           = n
PM
                                                and ItemType                    = '1'
                                                and     CommodityCode   = reccm.CommodityCo
de
                                                and     Pre_Rev_Mark    = '2'
                                                and     SurveyScope     <= inSurveyScope;

-- Get Past Record(This Month, Item Type 2)
                                select  sum(ShipmentValue)
                                        into    recpr.ShipmentValue
                                        from    VW_PastRecordScope
                                        where   ISIC                                    = r
eccm.ISIC
                                                and     Year
                = inYear
                                                and     Month                           = i
nMonth
                                                and ItemType                    = '2'
                                                and     CommodityCode   = reccm.CommodityCo
de
                                                and     Pre_Rev_Mark    = inPre_Rev_mark
                                                and     SurveyScope     <= inSurveyScope;

-- Get Past Record(Pre Month and Year, Item Type 2)
                                select  sum(ShipmentValue)
                                        into    recprp.ShipmentValue
                                        from    VW_PastRecordScope
                                        where   ISIC                                    = r
eccm.ISIC
                                                and     Year
                = nPY
                                                and     Month                           = n
PM
                                                and ItemType                    = '2'
                                                and     CommodityCode   = reccm.CommodityCo
de
                                                and     Pre_Rev_Mark    = '2'
                                                and     SurveyScope     <= inSurveyScope;

-- Growth Rate Calculation
                                recgq.GR_Production     := f_GrowthRate(recpr.ProductionQt
y, recprp.ProductionQty,

                        reclm.ProductionQty, reclmp.ProductionQty,

                        recdf.ProductionQty, recdfp.ProductionQty);
```

```
                                if      recpr.DomesticSales is null
                                and recpr.Export is null then
                                        null;
                                else
                                        recpr.DomesticSales     := nvl(recpr.DomesticSale
s,0);
                                        recpr.Export                                    :=
nvl(recpr.Export,0);
                                end if;
                                if      recprp.DomesticSales is null
                                and recprp.Export is null then
                                        null;
                                else
                                        recprp.DomesticSales    := nvl(recprp.DomesticSale
s,0);
                                        recprp.Export                           := nvl(recp
rp.Export,0);
                                end if;

                                recgq.GR_ShipmentQty    := f_GrowthRate(recpr.DomesticSales
+ recpr.Export,


                        recprp.DomesticSales + recprp.Export,


                        recim.ShipmentQty, recimp.ShipmentQty,


                        recdf.ShipmentQty, recdfp.ShipmentQty);
                                recgq.GR_MEInventory    := f_GrowthRate(recpr.ME_Inventor
y, recprp.ME_Inventory,


                        recim.ME_Inventory, recimp.ME_Inventory,


                        recdf.ME_Inventory, recdfp.ME_Inventory);
                                recgq.GR_ShipmentVal    := f_GrowthRate(recpr.ShipmentValu
e, recprp.ShipmentValue,


                        recim.ShipmentValue, recimp.ShipmentValue,


                        1,1);

-- Insert Growth Rate Qty Table
                                p_GRQtyIns (reccm.ISIC, reccm.ItemType, reccm.CommodityCode,
inYear, inMonth,
                                                                                inSurveySco
pe, inPre_Rev_mark, inComparisonTarget, inSeq,
                                                                                recgq.GR_Pr
oduction, recgq.GR_ShipmentQty,
                                                                                recgq.GR_ME
Inventory, recgq.GR_ShipmentVal, null);

                        elsif   reccm.ItemType = '4' then
-- Get Past Record(This Month, Item Type 4)
                                select  sum(RwMt_MEInventory)
                                        into    recpr.RwMt_MEInventory
                                        from    VW_PastRecordScope
                                        where   ISIC                                    = r
eccm.ISIC
                                                and     Year
                = inYear
                                                and     Month                           = i
nMonth
                                                and ItemType                    = '4'
                                                and     CommodityCode   = reccm.CommodityCo
de
                                                and     Pre_Rev_Mark    = inPre_Rev_mark
                                                and     SurveyScope     <= inSurveyScope;

-- Get Past Record(Pre Month and Year, Item Type 4)
                                select  sum(RwMt_MEInventory)
                                        into    recprp.RwMt_MEInventory
```

```
                                        from    VW_PastRecordScope
                                        where   ISIC                                    = r
eccm.ISIC
                                                and     Year
                = nPY
                                                and     Month                           = n
PM
                                                and ItemType                    = '4'
                                                and     CommodityCode   = reccm.CommodityCo
de
                                                and     Pre_Rev_Mark    = '2'
                                                and     SurveyScope     <= inSurveyScope;

-- Growth Rate Calculation
                                recgq.GR_RawMaterial    := f_GrowthRate(recpr.RwMt_MEInven
tory, recprp.RwMt_MEInventory,


                                reclm.RwMt_MEInventory, reclmp.RwMt_MEInventory,



                                recdf.RwMt_MEInventory, recdfp.RwMt_MEInventory);

                                p_GRQtyIns (reccm.ISIC, reccm.ItemType, reccm.CommodityCode,
inYear, inMonth,
                                                                                        inSurveySco
pe, inPre_Rev_mark, inComparisonTarget, inSeq,
                                                                                        null, null, n
ull, null, recgq.GR_RawMaterial);

                        end if;
                end if;

-- Fetch Indices Table
                fetch   csrcm
                        into    reccm.ISIC, reccm.ItemType, reccm.CommodityCode;

        end loop;

        close csrcm;

end;

-------------------------------------------------------
-- Growth Rate Of Ratio Index Elements Calculation By Commodity.
-------------------------------------------------------
procedure p_GREleComm (
        inYear                                          in      number,
        inMonth                                         in      number,
        inPre_Rev_Mark                  in      varchar2) is

-- Record Area Define
        recgr           GrowthRate%rowtype;
        reclm           LinkMaster%rowtype;
        reclmp  LinkMaster%rowtype;
        recdf           DeflatorMaster%rowtype;
        recdfp  DeflatorMaster%rowtype;
        recpr           PastRecord%rowtype;
        recprp  PastRecord%rowtype;

-- Work Area
        nPM                     number;
        nPY                     number;

--  Growth Rate Table Cursor
        Cursor  csrgr is
                select  *
                        from  GrowthRate
                        where   AggregationRange = '5'
                                and     ItemType in('1','6')
                                and     Year = inYear
                                and     Month = inMonth
                                and     Pre_Rev_Mark = inPre_Rev_Mark
                        for update;

Begin
```

```
        DBMS_OUTPUT.ENABLE(60000);
        open    csrgr;
        fetch   csrgr   into    recgr;
        while csrgr%found loop

-- Get Pre Year and Month
                if      recgr.ComparisonTarget = '0' then
                        if      inMonth = 1 then
                                nPY     := inYear - 1;
                                nPM     := 12;
                        else
                                nPY     := inYear;
                                nPM     := inMonth - 1;
                        end if;
                else
                        nPY     := inYear - 1;
                        nPM     := inMonth;
                end if;

-- Get LinkMaster Coefficient(This Month and Pre Month)
                begin
                        select  *
                                into    recIm
                                from    LinkMaster
                                where   ISIC                                    = recgr.ISIC
                                        and     Year                                    = i
nYear
                                        and     Month                           = inMonth
                                        and ItemType                    = recgr.ItemType
                                        and     CommodityCode   = recgr.IndexCommodityCode
                                        and SurveyScope         = recgr.SurveyScope
                                        and     Version                         = '1';

                        select  *
                                into    recImp
                                from    LinkMaster
                                where   ISIC                                    = recgr.ISIC
                                        and     Year                                    = n
PY
                                        and     Month                           = nPM
                                        and ItemType                    = recgr.ItemType
                                        and     CommodityCode   = recgr.IndexCommodityCode
                                        and SurveyScope         = recgr.SurveyScope
                                        and     Version                         = '1';

                exception
                        when no_data_found then
                                recIm.ProductionQty             := 0;
        recIm.ShipmentQty                               := 0;
        recIm.ME_Inventory                      := 0;
        recIm.Capacity                                  := 0;
                                recImp.ProductionQty            := 0;
        recImp.ShipmentQty                      := 0;
        recImp.ME_Inventory                     := 0;
        recImp.Capacity                                 := 0;
                end;

-- Select Deflator Master
                begin
                        select  *
                                into    recdf
                                from    DeflatorMaster
                                where   ISIC                                    = recgr.ISIC
                                        and     Year                                    = i
nYear
                                        and     Month                           = inMonth
                                        and ItemType                    = recgr.ItemType
                                        and     CommodityCode   = recgr.IndexCommodityCode
                                        and SurveyScope         = recgr.SurveyScope
                                        and     Version                         = '1';

                        select  *
                                into    recdfp
                                from    DeflatorMaster
                                where   ISIC                                    = recgr.ISIC
                                        and     Year                                    = n
PY
```

```
                                        and     Month                           = nPM
                                        and ItemType                    = recgr. ItemType
                                        and     CommodityCode   = recgr. IndexCommodityCode
                                        and SurveyScope         = recgr. SurveyScope
                                        and     Version                         = '1';

                exception
                        when no_data_found then
                                recdf. ProductionQty            := 1;
                                recdf. ShipmentQty                              := 1;
                                recdf. ME_Inventory                     := 1;
                        recdf. Capacity                                 := 1;
                                recdf. RwMt_MEInventory  := 1;
                                recdfp. ProductionQty           := 1;
                                recdfp. ShipmentQty                     := 1;
                        recdfp. ME_Inventory                    := 1;
                        recdfp. Capacity                                := 1;
                        recdfp. RwMt_MEInventory := 1;
                end;

                if      recgr. ItemType = '1' then

-- Get Past Record(This Month, Item Type 1)
                        select  sum(DomesticSales), sum(Export), sum(ME_Inventory)
                                into    recpr. DomesticSales, recpr. Export, recpr. ME_Inventory
                                from    VW_PastRecordScope
                                where   ISIC                                    = recgr. ISIC
                                        and     Year                                    = i
nYear
                                        and     Month                           = inMonth
                                        and ItemType                    = '1'
                                        and     CommodityCode   = recgr. IndexCommodityCode
                                        and     Pre_Rev_Mark    = inPre_Rev_mark
                                        and     SurveyScope     <= recgr. SurveyScope;

-- Get Past Record(Pre Month and Year, Item Type 1)
                        select  sum(DomesticSales), sum(Export), sum(ME_Inventory)
                                into    recprp. DomesticSales, recprp. Export, recprp. ME_Inven
tory
                                from    VW_PastRecordScope
                                where   ISIC                                    = recgr. ISIC
                                        and     Year                                    = n
PY
                                        and     Month                           = nPM
                                        and ItemType                    = '1'
                                        and     CommodityCode   = recgr. IndexCommodityCode
                                        and     Pre_Rev_Mark    = '2'
                                        and     SurveyScope     <= recgr. SurveyScope;

-- Growth Rate Calculation
                        recgr. GR_InventoryQty_IR       := f_GrowthRate(recpr. ME_Inventor
y, recprp. ME_Inventory,


                                recim. ME_Inventory, recimp. ME_Inventory,


                                recdf. ME_Inventory, recdfp. ME_Inventory);

                        if      recpr. DomesticSales is null
                        and recpr. Export is null then
                                null;
                        else
                                recpr. DomesticSales := nvl(recpr. DomesticSales, 0);
                                recpr. Export := nvl(recpr. Export, 0);
                        end if;
                        if      recprp. DomesticSales is null
                        and recprp. Export is null then
                                null;
                        else
                                recprp. DomesticSales   := nvl(recprp. DomesticSales, 0);
                                recprp. Export := nvl(recprp. Export, 0);
                        end if;
                        recgr. GR_ShipmentQty_IR        := f_GrowthRate(recpr. DomesticSales
+ recpr. Export,
```

```
                                recprp.DomesticSales + recprp.Export,

                                reclm.ShipmentQty, reclmp.ShipmentQty,

                                recdf.ShipmentQty, recdfp.ShipmentQty);
                        recgr.GR_ProductionQty_CU        :=      null;
                        recgr.GR_Capacity_CU                     :=      null;

                elsif   recgr.ItemType = '6' then
-- Get Past Record(This Month, Item Type 6)
                        select  sum(Capacity)
                                into    recpr.Capacity
                                from    VW_PastRecordScope
                                where   ISIC                                    = recgr.ISIC
                                        and     Year                                    = i
nYear
                                        and     Month                           = inMonth
                                        and ItemType                    = '6'
                                        and     CommodityCode   = recgr.IndexCommodityCode
                                        and     Pre_Rev_Mark    = inPre_Rev_mark
                                        and     SurveyScope     <= recgr.SurveyScope;

-- Get Past Record(Pre Month and Year, Item Type 4)
                        select  sum(Capacity)
                                into    recprp.Capacity
                                from    VW_PastRecordScope
                                where   ISIC                                    = recgr.ISIC
                                        and     Year                                    = n
PY
                                        and     Month                           = nPM
                                        and ItemType                    = '6'
                                        and     CommodityCode   = recgr.IndexCommodityCode
                                        and     Pre_Rev_Mark    = '2'
                                        and     SurveyScope     <= recgr.SurveyScope;

-- Get Production Qty(This Month)
                        select  sum(PastRecord.ProductionQty * LinkMaster.ProductionQty
                                                                        / DeflatorMaster.P
roductionQty)
                                into    recpr.ProductionQty
                                from    PastRecord, CommodityMaster, EstablishmentMaster, Lin
kMaster, DeflatorMaster
                                where   PastRecord.ISIC
                                        = CommodityMaster.ISIC
                                        and     PastRecord.ItemType
                                = CommodityMaster.ItemType
                                        and     PastRecord.CommodityCode
                        = CommodityMaster.CommodityCode
                                        and     PastRecord.ISIC
                                                = EstablishmentMaster.ISIC
                                        and     PastRecord.RegistrationNo
                = EstablishmentMaster.RegistrationNo
                                        and     PastRecord.ISIC
                                                = LinkMaster.ISIC
                                        and     PastRecord.ItemType
                                = LinkMaster.ItemType
                                        and     PastRecord.CommodityCode
                        = LinkMaster.CommodityCode
                                        and     PastRecord.Year
                                                = LinkMaster.Year
                                        and     PastRecord.Month
                                                = LinkMaster.Month
                                        and     LinkMaster.Version
                                        =       '1'
                                        and     PastRecord.ISIC
                                                = DeflatorMaster.ISIC
                                        and     PastRecord.ItemType
                                = DeflatorMaster.ItemType
                                        and     PastRecord.CommodityCode
                        = DeflatorMaster.CommodityCode
                                        and     PastRecord.Year
                                                = DeflatorMaster.Year
                                        and     PastRecord.Month
                                                = DeflatorMaster.Month
                                        and     DeflatorMaster.Version
```

```
                        =       '1'
                                        and     PastRecord. ISIC
                                                = recgr. ISIC
                                        and     PastRecord. Year
                                                = inYear
                                        and     PastRecord. Month
                                                = inMonth
                                        and PastRecord. ItemType
                                = '1'
                                        and     PastRecord. Pre_Rev_Mark
                        = inPre_Rev_mark
                                        and     CommodityMaster. CapacityCode
=       recgr. IndexCommodityCode
                                        and     EstablishmentMaster. SurveyScope <= recgr. S
urveyScope
                                        and     LinkMaster. SurveyScope
                        = recgr. SurveyScope
                                        and     DeflatorMaster. SurveyScope
                = recgr. SurveyScope;

--      DBMS_OUTPUT. PUT_LINE('code= ' || recgr. IndexCommodityCode);

--      DBMS_OUTPUT. PUT_LINE('recpr-ProductionQty= ' || recpr. ProductionQty);

-- Get Production Qty(Pre Month)
                        select  sum(PastRecord. ProductionQty * LinkMaster. ProductionQty
                                                                                / DeflatorMaster. P
roductionQty)
                                into    recprp. ProductionQty
                                from    PastRecord, CommodityMaster, EstablishmentMaster, Lin
kMaster, DeflatorMaster
                                where   PastRecord. ISIC
                                        = CommodityMaster. ISIC
                                        and     PastRecord. ItemType
                                = CommodityMaster. ItemType
                                        and     PastRecord. CommodityCode
                        = CommodityMaster. CommodityCode
                                        and     PastRecord. ISIC
                                                = EstablishmentMaster. ISIC
                                        and     PastRecord. RegistrationNo
                = EstablishmentMaster. RegistrationNo
                                        and     PastRecord. ISIC
                                                = LinkMaster. ISIC
                                        and     PastRecord. ItemType
                                = LinkMaster. ItemType
                                        and     PastRecord. CommodityCode
                        = LinkMaster. CommodityCode
                                        and     PastRecord. Year
                                                = LinkMaster. Year
                                        and     PastRecord. Month
                                                = LinkMaster. Month
                                        and     LinkMaster. Version
                                        =       '1'
                                        and     PastRecord. ISIC
                                                = DeflatorMaster. ISIC
                                        and     PastRecord. ItemType
                                = DeflatorMaster. ItemType
                                        and     PastRecord. CommodityCode
                        = DeflatorMaster. CommodityCode
                                        and     PastRecord. Year
                                                = DeflatorMaster. Year
                                        and     PastRecord. Month
                                                = DeflatorMaster. Month
                                        and     DeflatorMaster. Version
                        =       '1'
                                        and     PastRecord. ISIC
                                                = recgr. ISIC
                                        and     PastRecord. Year
                                                = nPY
                                        and     PastRecord. Month
                                                = nPM
                                        and PastRecord. ItemType
                                = '1'
                                        and     PastRecord. Pre_Rev_Mark
                        = '2'
                                        and     CommodityMaster. CapacityCode
=       recgr. IndexCommodityCode
```

```
                                        and     EstablishmentMaster.SurveyScope <= recgr.S
urveyScope
                                        and     LinkMaster.SurveyScope
                        = recgr.SurveyScope
                                        and     DeflatorMaster.SurveyScope
                = recgr.SurveyScope;


-- Growth Rate Calculation
                        recgr.GR_Capacity_CU                    := f_GrowthRate(recpr.Capa
city, recprp.Capacity,


                                recim.Capacity, recimp.Capacity,


                                recdf.Capacity, recdfp.Capacity);
                        recgr.GR_ProductionQty_CU       := f_GrowthRate(recpr.ProductionQt
y, recprp.ProductionQty, 1, 1, 1, 1);

--      DBMS_OUTPUT.PUT_LINE('recgr-ProductionQty= ' || recgr.GR_ProductionQty_CU);

                        recgr.GR_InventoryQty_IR        := null;
                        recgr.GR_ShipmentQty_IR         := null;

                end if;

-- Growth Rate Master
                update  GrowthRate
                        set             GR_InventoryQty_IR      = recgr.GR_InventoryQty_IR,
                                                GR_ShipmentQty_IR       = recgr.GR_Shipment
Qty_IR,
                                                GR_Capacity_CU                  = recgr.GR_
Capacity_CU,
                                                GR_ProductionQty_CU = recgr.GR_ProductionQ
ty_CU,
                                                GR_ProductionVal_LP     = null,
                                                GR_ManHour_LP   = null
                        where   current Of csrgr;

-- Fetch Indices Table
                fetch   csrgr   into    recgr;

        end loop;

        close csrgr;

end;

-------------------------------------------------------
-- Growth Rate Of Ratio Index Elements Calculation By Industry.
-------------------------------------------------------
procedure p_GREleIndustry (
        inYear                                                  in      number,
        inMonth                                                 in      number,
        inPre_Rev_Mark                  in      varchar2) is

-- Record Area Define
        recgr           GrowthRate%rowtype;
        recim           LinkMaster%rowtype;
        recimp  LinkMaster%rowtype;
        recpr           PastRecord%rowtype;
        recprp  PastRecord%rowtype;

-- Work Area
        nPM                     number;
        nPY                     number;

        nCM_ProductionValue             number;
        nCM_ManHours                                    number;
        nPM_ProductionValue             number;
        nPM_ManHours                                    number;

        nBPYear                 number;
        nBPMonth                number;

--  Growth Rate Table Cursor
```

```
        Cursor  csrgr is
                select  *
                        from  GrowthRate
                        where   AggregationRange = '4'
                                and     Year = inYear
                                and     Month = inMonth
                                and     Pre_Rev_Mark = inPre_Rev_Mark
                        for update;

Begin
--      DBMS_OUTPUT.PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Commodity Master Cursor Open and Fetch
        open    csrgr;
        fetch   csrgr   into    recgr;
        while csrgr%found loop

-- Get Pre Year and Month
                if      recgr.ComparisonTarget = '0' then
                        if      inMonth = 1 then
                                nPY     := inYear - 1;
                                nPM     := 12;
                        else
                                nPY     := inYear;
                                nPM     := inMonth - 1;
                        end if;
                else
                        nPY     := inYear - 1;
                        nPM     := inMonth;
                end if;

-- Get Base Piriod Year and Month.
                CalcIndex.GetBasePeriod(inYear, inMonth, recgr.SurveyScope, nBPYear, nBPMonth);

-- Get Production Value(This Month)
                select  sum(PastRecord.ProductionQty * BasePeriodFile.UnitPrice *
                                                                        LinkMaster.ProductionQty /
DeflatorMaster.ProductionQty)
                        into    nCM_ProductionValue
                        from    PastRecord, EstablishmentMaster, LinkMaster, BasePeriodFile, D
eflatorMaster
                        where   PastRecord.ISIC
                                = EstablishmentMaster.ISIC
                                and     PastRecord.RegistrationNo                       = E
stablishmentMaster.RegistrationNo
                                and     PastRecord.ISIC
                                        = LinkMaster.ISIC
                                and     PastRecord.ItemType
                        = LinkMaster.ItemType
                                and     PastRecord.CommodityCode
                = LinkMaster.CommodityCode
                                and     PastRecord.Year
                                        = LinkMaster.Year
                                and     PastRecord.Month
                                        = LinkMaster.Month
                                and     LinkMaster.Version
                                = '1'
                                and     PastRecord.ISIC
                                        = DeflatorMaster.ISIC
                                and     PastRecord.ItemType
                        = DeflatorMaster.ItemType
                                and     PastRecord.CommodityCode
                = DeflatorMaster.CommodityCode
                                and     PastRecord.Year
                                        = DeflatorMaster.Year
                                and     PastRecord.Month
                                        = DeflatorMaster.Month
                                and     DeflatorMaster.Version
                = '1'
                                and     PastRecord.ISIC
                                        = recgr.ISIC
                                and     PastRecord.Year
                                        = inYear
                                and     PastRecord.Month
                                        = inMonth
                                and PastRecord.ItemType
                        = '1'
                                and     PastRecord.Pre_Rev_Mark
```

```
                = inPre_Rev_mark
                                and     EstablishmentMaster. SurveyScope <= recgr. SurveySco
pe
                                and     LinkMaster. SurveyScope
                = recgr. SurveyScope
                                and     DeflatorMaster. SurveyScope                     = r
ecgr. SurveyScope
                                and     PastRecord. ISIC
                                        = BasePeriodFile. ISIC
                                and     PastRecord. ItemType
                        = BasePeriodFile. ItemType
                                and     PastRecord. CommodityCode
                = BasePeriodFile. CommodityCode
                                and     PastRecord. RegistrationNo                      = B
asePeriodFile. RegistrationNo
                                and     BasePeriodFile. Year
                        = nBPYear
                                and     BasePeriodFile. Month
                        = nBPMonth
                                and BasePeriodFile. Version = '1';

-- Get Production Value(Pre Month)
                select  sum(PastRecord. ProductionQty * BasePeriodFile. UnitPrice *
                                                                        LinkMaster. Producti
onQty  / DeflatorMaster. ProductionQty)
                        into    nPM_ProductionValue
                        from    PastRecord, EstablishmentMaster, LinkMaster, BasePeriodFile, D
eflatorMaster
                        where   PastRecord. ISIC
                                = EstablishmentMaster. ISIC
                                and     PastRecord. RegistrationNo                      = E
stablishmentMaster. RegistrationNo
                                and     PastRecord. ISIC
                                        = LinkMaster. ISIC
                                and     PastRecord. ItemType
                        = LinkMaster. ItemType
                                and     PastRecord. CommodityCode
                = LinkMaster. CommodityCode
                                and     PastRecord. Year
                                        = LinkMaster. Year
                                and     PastRecord. Month
                                        = LinkMaster. Month
                                and     LinkMaster. Version
                                = '1'
                                and     PastRecord. ISIC
                                        = DeflatorMaster. ISIC
                                and     PastRecord. ItemType
                        = DeflatorMaster. ItemType
                                and     PastRecord. CommodityCode
                = DeflatorMaster. CommodityCode
                                and     PastRecord. Year
                                        = DeflatorMaster. Year
                                and     PastRecord. Month
                                        = DeflatorMaster. Month
                                and     DeflatorMaster. Version
                = '1'
                                and     PastRecord. ISIC
                                        = recgr. ISIC
                                and     PastRecord. Year
                                        = nPY
                                and     PastRecord. Month
                                        = nPM
                                and PastRecord. ItemType
                        = '1'
                                and     PastRecord. Pre_Rev_Mark
                = '2'
                                and     EstablishmentMaster. SurveyScope <= recgr. SurveySco
pe
                                and     LinkMaster. SurveyScope
                = recgr. SurveyScope
                                and     DeflatorMaster. SurveyScope                     = r
ecgr. SurveyScope
                                and     PastRecord. ISIC
                                        = BasePeriodFile. ISIC
                                and     PastRecord. ItemType
                        = BasePeriodFile. ItemType
                                and     PastRecord. CommodityCode
```

```
                = BasePeriodFile.CommodityCode
                                and     PastRecord.RegistrationNo                       = B
asePeriodFile.RegistrationNo
                                and     BasePeriodFile.Year
                        = nBPYear
                                and     BasePeriodFile.Month
                        = nBPMonth
                                and BasePeriodFile.Version = '1';

-- Get Production Value(This Month)
                select  sum(a.Labor_SC * b.LaborTotal * c.LaborTotal *
                                                        d.Worker * e.WorkingHour *
f.WorkingDay)
                        into    nCM_ManHours
                        from    PastRecord a, PastRecord b, PastRecord c,
                                        linkMaster d, linkMaster e, linkMaster f, Est
ablishmentMaster
                        where a.ISIC                                    = b.ISIC
                                and     a.Year                                  =
b.Year
                                and     a.Month                                 =
b.Month
                                and     a.RegistrationNo        = b.RegistrationNo
                                and     a.ItemType                      =       b.I
temType
                                and     a.Pre_Rev_Mark          =       b.Pre_Rev_Mark
                                and     b.CommodityCode         =       '020'
                                and a.ISIC                                      =
c.ISIC
                                and     a.Year                                  =
c.Year
                                and     a.Month                                 =
c.Month
                                and     a.RegistrationNo        = c.RegistrationNo
                                and     a.ItemType                      =       c.I
temType
                                and     a.Pre_Rev_Mark          =       c.Pre_Rev_Mark
                                and     c.CommodityCode         =       '030'
                                and     a.ISIC                                  =
d.ISIC
                                and     a.ItemType                      = d.ItemType
                                and     a.Year
=       d.Year
                                and     a.Month                                 =
d.Month
                                and     d.CommodityCode         = '010'
                                and     d.SurveyScope                   = recgr.SurveyScope
                                and     d.Version                               = '
1'
                                and     a.ISIC                                  =
e.ISIC
                                and     a.ItemType                      = e.ItemType
                                and     a.Year
=       e.Year
                                and     a.Month                                 =
e.Month
                                and     e.CommodityCode         = '020'
                                and     e.SurveyScope                   = recgr.SurveyScope
                                and     e.Version                               = '
1'
                                and     a.ISIC                                  =
f.ISIC
                                and     a.ItemType                      = f.ItemType
                                and     a.Year
=       f.Year
                                and     a.Month                                 =
f.Month
                                and     f.CommodityCode         = '030'
                                and     f.SurveyScope                   = recgr.SurveyScope
                                and     f.Version                               = '
1'
                                and     a.ISIC                                  = E
stablishmentMaster.ISIC
                                and     a.RegistrationNo        = EstablishmentMaster.Regi
strationNo
                                and     EstablishmentMaster.SurveyScope <= recgr.SurveySco
pe
```

```
                                and     a. ISIC                                         = r
ecgr. ISIC
                                and     a. Year                                         = i
nYear
                                and     a. Month                                        = i
nMonth
                                and a. ItemType                         = '3'
                                and     a. CommodityCode        =       '010'
                                and     a. Pre_Rev_Mark         = inPre_Rev_mark;

-- Get Production Value(Pre Month)
                select  sum(a. Labor_SC * b. LaborTotal * c. LaborTotal *
                                                        d. Worker * e. WorkingHour *
f. WorkingDay)
                        into    nPM_ManHours
                        from    PastRecord a, PastRecord b, PastRecord c,
                                        linkMaster d, linkMaster e, linkMaster f, Est
ablishmentMaster
                        where a. ISIC                                           = b. ISIC
                                and     a. Year                                         =
b. Year
                                and     a. Month                                        =
b. Month
                                and     a. RegistrationNo       = b. RegistrationNo
                                and     a. ItemType                             =       b. I
temType
                                and     a. Pre_Rev_Mark         =       b. Pre_Rev_Mark
                                and     b. CommodityCode        =       '020'
                                and a. ISIC                                             =
c. ISIC
                                and     a. Year                                         =
c. Year
                                and     a. Month                                        =
c. Month
                                and     a. RegistrationNo       = c. RegistrationNo
                                and     a. ItemType                             =       c. I
temType
                                and     a. Pre_Rev_Mark         =       c. Pre_Rev_Mark
                                and     c. CommodityCode        =       '030'
                                and     a. ISIC                                         =
d. ISIC
                                and     a. ItemType                             = d. ItemType
                                and     a. Year
=       d. Year
                                and     a. Month                                        =
d. Month
                                and     d. CommodityCode        = '010'
                                and     d. SurveyScope                  = recgr. SurveyScope
                                and     d. Version                                      = '
1'
                                and     a. ISIC                                         =
e. ISIC
                                and     a. ItemType                             = e. ItemType
                                and     a. Year
=       e. Year
                                and     a. Month                                        =
e. Month
                                and     e. CommodityCode        = '020'
                                and     e. SurveyScope                  = recgr. SurveyScope
                                and     e. Version                                      = '
1'
                                and     a. ISIC                                         =
f. ISIC
                                and     a. ItemType                             = f. ItemType
                                and     a. Year
=       f. Year
                                and     a. Month                                        =
f. Month
                                and     f. CommodityCode        = '030'
                                and     f. SurveyScope                  = recgr. SurveyScope
                                and     f. Version                                      = '
1'
                                and     a. ISIC                                         = E
stablishmentMaster. ISIC
                                and     a. RegistrationNo       = EstablishmentMaster. Regi
strationNo
                                and     EstablishmentMaster. SurveyScope <= recgr. SurveySco
```

```
pe
                                and     a.ISIC                                          = r
ecgr.ISIC
                                and     a.Year                                          = n
PY
                                and     a.Month                                         = n
PM
                                and a.ItemType                          = '3'
                                and     a.CommodityCode         =       '010'
                                and     a.Pre_Rev_Mark          = '2';

-- Growth Rate Calculation
                        recgr.GR_ProductionVal_LP       := f_GrowthRate(nCM_ProductionValu
e, nPM_ProductionValue, 1, 1, 1, 1);
                        recgr.GR_ManHour_LP                             := f_GrowthRate(nCM
_ManHours, nPM_ManHours, 1, 1, 1, 1);

-- Growth Rate Master
                update  GrowthRate
                        set             GR_ProductionVal_LP     = recgr.GR_ProductionVal_L
P,
                                                GR_ManHour_LP   = recgr.GR_ManHour_LP,
                                                GR_InventoryQty_IR      = null,
                                                GR_ShipmentQty_IR       = null,
                                                GR_Capacity_CU                  = null,
                                                GR_ProductionQty_CU = null
                        where   current Of csrgr;

-- Fetch Indices Table
                fetch   csrgr   into    recgr;

        end loop;

        close csrgr;

end;

/* ==============================================================
    Compair data with the previous data
==============================================================  */
FUNCTION Fcpm (
          nPre    IN NUMBER,
          nCur    IN NUMBER)
      RETURN number
        IS
--        DECLARE
           nValue NUMBER;
        BEGIN
            IF nPre IS NULL OR nCur IS NULL THEN
                nValue := NULL;
                RETURN nValue;
            ELSIF nPre = 0 THEN
                nValue := NULL;
                RETURN nValue;
            ELSE
                nValue := ((nCur/nPre) - 1) * 100;
                RETURN nValue;
            END IF;
END; -- the end of function Fcpm

/* ==============================================================
    Insert data to GrowthRate Table.
==============================================================  */
PROCEDURE p_GRIndices(
      pYear           IN NUMBER,
      pMonth          IN NUMBER,
      pPreRev         IN VARCHAR2)
                IS

--    DECLARE
        CURSOR CsrIndex IS
            SELECT *
            FROM Indices
            WHERE Year = pYear
            AND Month = pMonth
            AND Pre_Rev_Mark = pPreRev
            ORDER BY ISIC, ItemType;
```

```
        RecIndex     CsrIndex%ROWTYPE;

        RecGR        GrowthRate%ROWTYPE;
        nCnt         NUMBER;
        nPreMonth    NUMBER;
        nYear        NUMBER;
        nPreYear     NUMBER;

        CURSOR CsrPM IS
            SELECT *
            FROM Indices
            WHERE AggregationRange     = RecIndex.AggregationRange
              AND ISIC       = RecIndex.ISIC
              AND ItemType   = RecIndex.ItemType
              AND IndexCommodityCode  = RecIndex.IndexCommodityCode
              AND Year  = nYear
              AND Month = nPreMonth
              AND SurveyScope  = RecIndex.SurveyScope
              AND Pre_Rev_Mark = '2';
        RecPM      CsrPM%ROWTYPE;

        CURSOR CsrPY IS
            SELECT *
            FROM Indices
            WHERE AggregationRange     = RecIndex.AggregationRange
              AND ISIC       = RecIndex.ISIC
              AND ItemType   = RecIndex.ItemType
              AND IndexCommodityCode  = RecIndex.IndexCommodityCode
              AND Year  = nPreYear
              AND Month = RecIndex.Month
              AND SurveyScope  = RecIndex.SurveyScope
              AND Pre_Rev_Mark = '2';
        RecPY      CsrPY%ROWTYPE;

BEGIN
-- DBMS_OUTPUT.ENABLE(60000);
-- Delete Growth Rate Table
                                  delete  from GrowthRate
                where   Year = pYear
            AND Month = pMonth
            AND Pre_Rev_Mark = pPreRev;

        OPEN CsrIndex;
        LOOP
            FETCH CsrIndex INTO RecIndex;
            EXIT WHEN CsrIndex%NOTFOUND;

            RecGR.AggregationRange := RecIndex.AggregationRange;
            RecGR.ISIC  := RecIndex.ISIC;
            RecGR.ItemType := RecIndex.ItemType;
            RecGR.IndexCommodityCode  := RecIndex.IndexCommodityCode;
            RecGR.Year := RecIndex.Year;
            RecGR.Month := RecIndex.Month;
            RecGR.SurveyScope := RecIndex.SurveyScope;
            RecGR.Pre_Rev_Mark := RecIndex.Pre_Rev_Mark;

            nPreYear := RecIndex.Year - 1;
            IF RecIndex.Month = 1 THEN
                nPreMonth := 12;
                nYear := RecIndex.Year - 1;
            ELSE
                nPreMonth := RecIndex.Month - 1;
                nYear := RecIndex.Year;
            END IF;

 -- Find out the compair data in each indices.
            -- Data in Previous Month
               OPEN CsrPM;
               FETCH CsrPM INTO RecPM;
               RecGR.ComparisonTarget := '0';
               IF CsrPM%NOTFOUND THEN
                    RecGR.GR_PR1 := NULL;
                    RecGR.GR_SP2 := NULL;
                    RecGR.GR_IV3 := NULL;
                    RecGR.GR_IR4 := NULL;
                    RecGR.GR_CU5 := NULL;
                    RecGR.GR_LP6 := NULL;
```

```
        RecGR.GR_LI7 := NULL;
        RecGR.GR_RI8 := NULL;
    ELSE
        RecGR.GR_PR1 := Fcpm(RecPM.IndexPR1, RecIndex.IndexPR1);
        RecGR.GR_SP2 := Fcpm(RecPM.IndexSP2, RecIndex.IndexSP2);
        RecGR.GR_IV3 := Fcpm(RecPM.IndexIV3, RecIndex.IndexIV3);
        RecGR.GR_IR4 := Fcpm(RecPM.IndexIR4, RecIndex.IndexIR4);
        RecGR.GR_CU5 := Fcpm(RecPM.IndexCU5, RecIndex.IndexCU5);
        RecGR.GR_LP6 := Fcpm(RecPM.IndexLP6, RecIndex.IndexLP6);
        RecGR.GR_LI7 := Fcpm(RecPM.IndexLI7, RecIndex.IndexLI7);
        RecGR.GR_RI8 := Fcpm(RecPM.IndexRI8, RecIndex.IndexRI8);
    END IF;
    CLOSE CsrPM;


INSERT INTO GrowthRate (
AggregationRange,
ISIC,
ItemType,
IndexCommodityCode,
Year,
Month,
SurveyScope,
Pre_Rev_Mark,
ComparisonTarget,
GR_PR1,
GR_SP2,
GR_IV3,
GR_IR4,
GR_CU5,
GR_LP6,
GR_LI7,
GR_RI8)
VALUES (
RecGR.AggregationRange,
RecGR.ISIC,
RecGR.ItemType,
RecGR.IndexCommodityCode,
RecGR.Year,
RecGR.Month,
RecGR.SurveyScope,
RecGR.Pre_Rev_Mark,
RecGR.ComparisonTarget,
RecGR.GR_PR1,
RecGR.GR_SP2,
RecGR.GR_IV3,
RecGR.GR_IR4,
RecGR.GR_CU5,
RecGR.GR_LP6,
RecGR.GR_LI7,
RecGR.GR_RI8);

-- Data in Same Month in Previous Year
    OPEN CsrPY;
    FETCH CsrPY INTO RecPY;
    RecGR.ComparisonTarget := '1';
    IF CsrPY%NOTFOUND THEN
        RecGR.GR_PR1 := NULL;
        RecGR.GR_SP2 := NULL;
        RecGR.GR_IV3 := NULL;
        RecGR.GR_IR4 := NULL;
        RecGR.GR_CU5 := NULL;
        RecGR.GR_LP6 := NULL;
        RecGR.GR_LI7 := NULL;
        RecGR.GR_RI8 := NULL;


    ELSE
        RecGR.GR_PR1 := Fcpm(RecPY.IndexPR1, RecIndex.IndexPR1);
        RecGR.GR_SP2 := Fcpm(RecPY.IndexSP2, RecIndex.IndexSP2);
        RecGR.GR_IV3 := Fcpm(RecPY.IndexIV3, RecIndex.IndexIV3);
        RecGR.GR_IR4 := Fcpm(RecPY.IndexIR4, RecIndex.IndexIR4);
        RecGR.GR_CU5 := Fcpm(RecPY.IndexCU5, RecIndex.IndexCU5);
        RecGR.GR_LP6 := Fcpm(RecPY.IndexLP6, RecIndex.IndexLP6);
        RecGR.GR_LI7 := Fcpm(RecPY.IndexLI7, RecIndex.IndexLI7);
        RecGR.GR_RI8 := Fcpm(RecPY.IndexRI8, RecIndex.IndexRI8);
    END IF;
```

```
            CLOSE CsrPY;

          INSERT INTO GrowthRate (
          AggregationRange,
          ISIC,
          ItemType,
          IndexCommodityCode,
          Year,
          Month,
          SurveyScope,
          Pre_Rev_Mark,
          ComparisonTarget,
          GR_PR1,
          GR_SP2,
          GR_IV3,
          GR_IR4,
          GR_CU5,
          GR_LP6,
          GR_LI7,
          GR_RI8)
          VALUES (
          RecGR.AggregationRange,
          RecGR.ISIC,
          RecGR.ItemType,
          RecGR.IndexCommodityCode,
          RecGR.Year,
          RecGR.Month,
          RecGR.SurveyScope,
          RecGR.Pre_Rev_Mark,
          RecGR.ComparisonTarget,
          RecGR.GR_PR1,
          RecGR.GR_SP2,
          RecGR.GR_IV3,
          RecGR.GR_IR4,
          RecGR.GR_CU5,
          RecGR.GR_LP6,
          RecGR.GR_LI7,
          RecGR.GR_RI8);

        END LOOP;  -- fetch data from Indeies.
      CLOSE CsrIndex;

END; -- end of p_GRIndices procedure

end GRCal;
/
show errors
/
```

```
-- ==================================================
-- Package Name : Index List of The Growth Rate Data Making
-- Package ID   : IndexListMk
-- Create Date  : 20/Oct/1999
-- Replace Date : xx/xxx/xxxx
-- Author       : Nakamura
--
-- File         : @c:¥jica¥indexlistmk;
-- Test                                : BEGIN IndexListMk.p_CommInAll(1, 1999, 3, '1', '2');
END;
-- Test                                :       BEGIN IndexListMk.p_IndustryInAll(10, 1999,
3, '1', '2'); END;
-- Test                                :       BEGIN IndexListMk.p_CommInIndustry(20, '1',
'1512', 1999, 3, '1', '2'); END;
-- Test                                : Delete from IndexList where seq = 1;
-- Test                                : Update IndexList set seq = 24 where seq = 20;
-- ==================================================
create or replace package IndexListMk is
procedure p_CommInAll (
        inSeq                                                   in      number,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2);

procedure p_IndustryInAll (
        inSeq                                                   in      number,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2);

procedure p_CommInIndustry (
        inSeq                                                   in      number,
        inAggregationRange      in      Varchar2,
        inISIC                                          in      Varchar2,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2);

end IndexListMk;
/
show errors
/

create or replace package body IndexListMk is
-----------------------------------------------------
-- Insert Temporary Index List Table
-----------------------------------------------------
procedure p_InsIndexList (
        inSeq                                                                   i
n       number,
        inReportType                                    in      varchar2,
        inIndexType                                             in      varchar
2,
        inYear                                                          in      num
ber,
        inMonth                                                         in      num
ber,
        inISIC                                                          in      var
char2,
        inCommodityCode                                 in      varchar2,
        inIndustryTypeShortName in      varchar2,
        inCommodityShortName            in      varchar2,
        inIndexCM                                                       in      num
ber,
        inIndexPM                                                       in      num
ber,
        inIndexPY                                                       in      num
ber,
        inGR_CPM                                                        in      num
ber,
        inGR_CPY                                                        in      num
ber,
        inCD_CPM                                                        in      num
ber,
```

```
        inCD_CPY                                                        in      num
ber,
        inCR_CPM                                                        in      num
ber,
        inCR_CPY                                                        in      num
ber,
        inSumIndexCM                                    in      number,
        inSumIndexPM                                    in      number,
        inSumIndexPY                                    in      number,
        inSumGR_CPM                                             in      number,
        inSumGR_CPY                                             in      number,
        inSumCD_CPM                                             in      number,
        inSumCD_CPY                                             in      number,
        inSumCR_CPM                                             in      number,
        inSumCR_CPY                                             in      number) is

begin
        Insert
        into            IndexList (
                                        Seq,
                                        ReportType,
                                        IndexType,
                                        Year,
                                        Month,
                                        ISIC,
                                        CommodityCode,
                                        IndustryTypeShortName,
                                        CommodityShortName,
                                        IndexCM,
                                        IndexPM,
                                        IndexPY,
                                        GR_CPM,
                                        GR_CPY,
                                        CD_CPM,
                                        CD_CPY,
                                        CR_CPM,
                                        CR_CPY,
                                        SumIndexCM,
                                        SumIndexPM,
                                        SumIndexPY,
                                        SumGR_CPM,
                                        SumGR_CPY,
                                        SumCD_CPM,
                                        SumCD_CPY,
                                        SumCR_CPM,
                                        SumCR_CPY)
        Values (inSeq,
                                        inReportType,
                                        inIndexType,
                                        inYear,
                                        inMonth,
                                        inISIC,
                                        inCommodityCode,
                                        inIndustryTypeShortName,
                                        inCommodityShortName,
                                        inIndexCM,
                                        inIndexPM,
                                        inIndexPY,
                                        inGR_CPM,
                                        inGR_CPY,
                                        inCD_CPM,
                                        inCD_CPY,
                                        inCR_CPM,
                                        inCR_CPY,
                                        inSumIndexCM,
                                        inSumIndexPM,
                                        inSumIndexPY,
                                        inSumGR_CPM,
                                        inSumGR_CPY,
                                        inSumCD_CPM,
                                        inSumCD_CPY,
                                        inSumCR_CPM,
                                        inSumCR_CPY);
end;
-------------------------------------------------------------
-- Commodity In All Manifacturing Data Get
```

```
---------------------------------------------------------
procedure p_CommInAll (
        inSeq                                                   in      number,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2) is

-- Record Area Define
        recid           Indices%rowtype;
        recidm  Indices%rowtype;
        recidy  Indices%rowtype;
        recgrm  growthRate%rowtype;
        recgry  growthRate%rowtype;
        recc2m  ContributionForAll2%rowtype;
        recc2y  ContributionForAll2%rowtype;
        reccm           CommodityMaster%rowtype;

        recids  Indices%rowtype;
        recidms Indices%rowtype;
        recidys Indices%rowtype;
        recgrms growthRate%rowtype;
        recgrys growthRate%rowtype;
        recc2ms ContributionForAll2%rowtype;
        recc2ys ContributionForAll2%rowtype;

-- Work Area
        nPM_Y           number;
        nPM_M           number;
        nPY_Y           number;
        nPY_M           number;

-- Indices Table Cursor
        Cursor csrid is
                select  Indices.ISIC, Indices.ItemType, Indices.IndexCommodityCode,
                        Indices.INDEXPR1, Indices.INDEXSP2, Indices.INDEXIV3,
                        Indices.INDEXIR4, Indices.INDEXCU5, Indices.INDEXLP6,
                                                a.GR_PR1, a.GR_SP2, a.GR_IV3, a.GR_IR4, a.GR_C
U5, a.GR_LP6,
                                                b.GR_PR1, b.GR_SP2, b.GR_IV3, b.GR_IR4, b.GR_C
U5, b.GR_LP6,
                                                c.CD_PR1, c.CD_SP2, c.CD_IV3, c.CD_IR4, c.CD_C
U5, c.CD_LP6,
                                                c.CR_PR1, c.CR_SP2, c.CR_IV3, c.CR_IR4, c.CR_C
U5, c.CR_LP6,
                                                d.CD_PR1, d.CD_SP2, d.CD_IV3, d.CD_IR4, d.CD_C
U5, d.CD_LP6,
                                                d.CR_PR1, d.CR_SP2, d.CR_IV3, d.CR_IR4, d.CR_C
U5, d.CR_LP6,
                                                CommodityMaster.CommodityShortName
        from  Indices, GrowthRate a, GrowthRate b,
                                ContributionForAll2 c, ContributionForAll2 d, CommodityMaster
 where  Indices.AggregationRange = a.AggregationRange
   and  Indices.ISIC = a.ISIC
   and  Indices.ItemType = a.ItemType
   and  Indices.IndexCommodityCode = a.IndexCommodityCode
   and  Indices.Year = a.Year
   and  Indices.Month = a.Month
   and  Indices.SurveyScope = a.SurveyScope
   and  Indices.Pre_Rev_Mark = a.Pre_Rev_Mark
   and  a.ComparisonTarget = '0'
   and  Indices.AggregationRange = b.AggregationRange
   and  Indices.ISIC = b.ISIC
   and  Indices.ItemType = b.ItemType
   and  Indices.IndexCommodityCode = b.IndexCommodityCode
   and  Indices.Year = b.Year
   and  Indices.Month = b.Month
   and  Indices.SurveyScope = b.SurveyScope
   and  Indices.Pre_Rev_Mark = b.Pre_Rev_Mark
   and  b.ComparisonTarget = '1'
   and  Indices.ISIC = c.ISIC
   and  Indices.ItemType = c.ItemType
   and  Indices.IndexCommodityCode = c.CommodityCode
   and  Indices.Year = c.Year
   and  Indices.Month = c.Month
   and  Indices.SurveyScope = c.SurveyScope
   and  Indices.Pre_Rev_Mark = c.Pre_Rev_Mark
```

```
        and     c.ComparisonTarget = '0'
        and     c.Seq = inSeq
   and  Indices.ISIC = d.ISIC
   and  Indices.ItemType = d.ItemType
   and  Indices.IndexCommodityCode = d.CommodityCode
   and  Indices.Year = d.Year
   and  Indices.Month = d.Month
   and  Indices.SurveyScope = d.SurveyScope
   and  Indices.Pre_Rev_Mark = d.Pre_Rev_Mark
        and     d.ComparisonTarget = '1'
        and     d.Seq = inSeq
   and  Indices.ISIC = CommodityMaster.ISIC
   and  Indices.ItemType = CommodityMaster.ItemType
   and  Indices.IndexCommodityCode = CommodityMaster.CommodityCode
        and     Indices.AggregationRange = '5'
        and     Indices.ItemType in('1','6')
        and     Indices.Year = inYear
        and     Indices.Month = inMonth
        and     Indices.SurveyScope = inSurveyScope
        and     Indices.Pre_Rev_Mark = inPre_Rev_Mark;

Begin
--      DBMS_OUTPUT.PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Initial Process
        nPY_Y   := inYear - 1;
        nPY_M   := inMonth;

        if      inMonth = 1 then
                        nPM_Y   := inYear - 1;
                        nPM_M   := 12;
        else
                        nPM_Y   := inYear;
                        nPM_M   := inMonth - 1;
        end if;

-- Summary Current Month Data Selecting
        begin
                select  Indices.INDEXPR1, Indices.INDEXSP2, Indices.INDEXIV3,
                        Indices.INDEXIR4, Indices.INDEXCU5, Indices.INDEXLP6,
                                                a.GR_PR1, a.GR_SP2, a.GR_IV3, a.GR_IR4, a.GR_C
U5, a.GR_LP6,
                                                b.GR_PR1, b.GR_SP2, b.GR_IV3, b.GR_IR4, b.GR_C
U5, b.GR_LP6
                into            recids.INDEXPR1, recids.INDEXSP2, recids.INDEXIV3,
                        recids.INDEXIR4, recids.INDEXCU5, recids.INDEXLP6,
                                                recgrms.GR_PR1, recgrms.GR_SP2, recgrms.GR_I
V3,
                                                recgrms.GR_IR4, recgrms.GR_CU5, recgrms.GR_L
P6,
                                                recgrys.GR_PR1, recgrys.GR_SP2, recgrys.GR_I
V3,
                                                recgrys.GR_IR4, recgrys.GR_CU5, recgrys.GR_L
P6
                from  Indices, GrowthRate a, GrowthRate b
                where  Indices.AggregationRange = a.AggregationRange
            and  Indices.ISIC = a.ISIC
            and  Indices.Year = a.Year
            and  Indices.Month = a.Month
            and  Indices.SurveyScope = a.SurveyScope
            and  Indices.Pre_Rev_Mark = a.Pre_Rev_Mark
            and  a.ComparisonTarget = '0'
            and  Indices.AggregationRange = b.AggregationRange
            and  Indices.ISIC = b.ISIC
            and  Indices.Year = b.Year
            and  Indices.Month = b.Month
            and  Indices.SurveyScope = b.SurveyScope
            and  Indices.Pre_Rev_Mark = b.Pre_Rev_Mark
            and  b.ComparisonTarget = '1'
                 and     Indices.AggregationRange = '1'
                 and     Indices.Year = inYear
                 and     Indices.Month = inMonth
                 and     Indices.SurveyScope = inSurveyScope
                 and     Indices.Pre_Rev_Mark = inPre_Rev_Mark;

        exception
                when no_data_found then
                        recids.INDEXPR1 := 0;
```

```
                        recids.INDEXSP2 := 0;
                        recids.INDEXIV3 := 0;
                        recids.INDEXIR4 := 0;
                        recids.INDEXCU5 := 0;
                        recids.INDEXLP6 := 0;
                        recgrms.GR_PR1  := 0;
                        recgrms.GR_SP2  := 0;
                        recgrms.GR_IV3  := 0;
                        recgrms.GR_IR4  := 0;
                        recgrms.GR_CU5  := 0;
                        recgrms.GR_LP6  := 0;
                        recgrys.GR_PR1  := 0;
                        recgrys.GR_SP2  := 0;
                        recgrys.GR_IV3  := 0;
                        recgrys.GR_IR4  := 0;
                        recgrys.GR_CU5  := 0;
                        recgrys.GR_LP6  := 0;
        end;

-- Summary Pre Month Data Selecting
        begin
                select  IndexPR1,
                        IndexSP2,
                        IndexIV3,
                        IndexIR4,
                        IndexCU5,
                        IndexLP6
                        into    recidms.IndexPR1,
                        recidms.IndexSP2,
                        recidms.IndexIV3,
                        recidms.IndexIR4,
                        recidms.IndexCU5,
                        recidms.IndexLP6
                        from  Indices
                        where AggregationRange = '1'
                        and Year = nPM_Y
                        and     Month = nPM_M
                        and SurveyScope = inSurveyScope
                                and     Indices.Pre_Rev_Mark = '2';

        exception
                when no_data_found then
                        recidms.IndexPR1        := 0;
      recidms.IndexSP2  := 0;
      recidms.IndexIV3  := 0;
      recidms.IndexIR4  := 0;
      recidms.IndexCU5  := 0;
      recidms.IndexLP6  := 0;
        end;

-- Summary Pre Year Data Selecting
        begin
                select  IndexPR1,
                        IndexSP2,
                        IndexIV3,
                        IndexIR4,
                        IndexCU5,
                        IndexLP6
                        into    recidys.IndexPR1,
                        recidys.IndexSP2,
                        recidys.IndexIV3,
                        recidys.IndexIR4,
                        recidys.IndexCU5,
                        recidys.IndexLP6
                        from  Indices
                        where AggregationRange = '1'
                        and Year = nPY_Y
                        and     Month = nPY_M
                        and SurveyScope = inSurveyScope
                                and     Indices.Pre_Rev_Mark = '2';

        exception
                when no_data_found then
                        recidys.IndexPR1        := null;
      recidys.IndexSP2  := null;
      recidys.IndexIV3  := null;
      recidys.IndexIR4  := null;
```

```
      recidys.IndexCU5  := null;
      recidys.IndexLP6  := null;
        end;

-- Summary Contribution CPM Data Selecting
        begin
                select  sum(CD_PR1), sum(CD_SP2), sum(CD_IV3), sum(CD_IR4), sum(CD_CU5), sum(CD
_LP6),
                                                sum(CR_PR1), sum(CR_SP2), sum(CR_IV3), sum(CR
_IR4), sum(CR_CU5), sum(CR_LP6)
                into            recc2ms.CD_PR1, recc2ms.CD_SP2, recc2ms.CD_IV3,
                                                recc2ms.CD_IR4, recc2ms.CD_CU5, recc2ms.CD_L
P6,
                                                recc2ms.CR_PR1, recc2ms.CR_SP2, recc2ms.CR_I
V3,
                                                recc2ms.CR_IR4, recc2ms.CR_CU5, recc2ms.CR_L
P6
                from    ContributionForAll2
                where           Year = inYear
                        and             Month = inMonth
                        and             SurveyScope = inSurveyScope
                        and             Pre_Rev_Mark = inPre_Rev_Mark
                        and             ComparisonTarget = '0'
                        and             Seq = inSeq;

        exception
                when no_data_found then
                        recc2ms.CD_PR1 := 0;
                        recc2ms.CD_SP2 := 0;
                        recc2ms.CD_IV3 := 0;
                        recc2ms.CD_IR4 := 0;
                        recc2ms.CD_CU5 := 0;
                        recc2ms.CD_LP6 := 0;
                        recc2ms.CR_PR1 := 0;
                        recc2ms.CR_SP2 := 0;
                        recc2ms.CR_IV3 := 0;
                        recc2ms.CR_IR4 := 0;
                        recc2ms.CR_CU5 := 0;
                        recc2ms.CR_LP6 := 0;
        end;

-- Summary Contribution CPY Data Selecting
        begin
                select  sum(CD_PR1), sum(CD_SP2), sum(CD_IV3), sum(CD_IR4), sum(CD_CU5), sum(CD
_LP6),
                                                sum(CR_PR1), sum(CR_SP2), sum(CR_IV3), sum(CR
_IR4), sum(CR_CU5), sum(CR_LP6)
                into            recc2ys.CD_PR1, recc2ys.CD_SP2, recc2ys.CD_IV3,
                                                recc2ys.CD_IR4, recc2ys.CD_CU5, recc2ys.CD_L
P6,
                                                recc2ys.CR_PR1, recc2ys.CR_SP2, recc2ys.CR_I
V3,
                                                recc2ys.CR_IR4, recc2ys.CR_CU5, recc2ys.CR_L
P6
                from    ContributionForAll2
                where           Year = inYear
                        and             Month = inMonth
                        and             SurveyScope = inSurveyScope
                        and             Pre_Rev_Mark = inPre_Rev_Mark
                        and             ComparisonTarget = '1'
                        and             Seq = inSeq;

        exception
                when no_data_found then
                        recc2ys.CD_PR1 := 0;
                        recc2ys.CD_SP2 := 0;
                        recc2ys.CD_IV3 := 0;
                        recc2ys.CD_IR4 := 0;
                        recc2ys.CD_CU5 := 0;
                        recc2ys.CD_LP6 := 0;
                        recc2ys.CR_PR1 := 0;
                        recc2ys.CR_SP2 := 0;
                        recc2ys.CR_IV3 := 0;
                        recc2ys.CR_IR4 := 0;
                        recc2ys.CR_CU5 := 0;
                        recc2ys.CR_LP6 := 0;
        end;
```

```
-- Indices Cursor Open Fetch
        open    csrid;
        fetch   csrid
                into    recid. ISIC, recid. ItemType, recid. IndexCommodityCode,
                recid. IndexPR1, recid. IndexSP2, recid. IndexIV3,
                                        recid. IndexIR4, recid. IndexCU5, recid. IndexLP6,
                                        recgrm. GR_PR1, recgrm. GR_SP2, recgrm. GR_IV3,
                                        recgrm. GR_IR4, recgrm. GR_CU5, recgrm. GR_LP6,
                                        recgry. GR_PR1, recgry. GR_SP2, recgry. GR_IV3,
                                        recgry. GR_IR4, recgry. GR_CU5, recgry. GR_LP6,
                                        recc2m. CD_PR1, recc2m. CD_SP2, recc2m. CD_IV3,
                                        recc2m. CD_IR4, recc2m. CD_CU5, recc2m. CD_LP6,
                                        recc2m. CR_PR1, recc2m. CR_SP2, recc2m. CR_IV3,
                                        recc2m. CR_IR4, recc2m. CR_CU5, recc2m. CR_LP6,
                                        recc2y. CD_PR1, recc2y. CD_SP2, recc2y. CD_IV3,
                                        recc2y. CD_IR4, recc2y. CD_CU5, recc2y. CD_LP6,
                                        recc2y. CR_PR1, recc2y. CR_SP2, recc2y. CR_IV3,
                                        recc2y. CR_IR4, recc2y. CR_CU5, recc2y. CR_LP6, reccm. Co
mmodityShortName;

-- Indices Cursor Fetch Looping
        while csrid%found loop

-- Pre Month Data Selecting
                begin
                        select
                                                IndexPR1,
                        IndexSP2,
                        IndexIV3,
                        IndexIR4,
                        IndexCU5,
                        IndexLP6
                        into    recidm. IndexPR1,
                        recidm. IndexSP2,
                        recidm. IndexIV3,
                        recidm. IndexIR4,
                        recidm. IndexCU5,
                        recidm. IndexLP6
                        from  Indices
                        where AggregationRange = '5'
                        and     ISIC = recid. ISIC
                        and ItemType = recid. ItemType
                        and IndexCommodityCode = recid. IndexCommodityCode
                        and Year = nPM_Y
                        and     Month = nPM_M
                        and SurveyScope = inSurveyScope
                                and     Indices. Pre_Rev_Mark = '2';

                exception
                        when no_data_found then
                                recidm. IndexPR1 := 0;
        recidm. IndexSP2 := 0;
        recidm. IndexIV3 := 0;
        recidm. IndexIR4 := 0;
        recidm. IndexCU5 := 0;
        recidm. IndexLP6 := 0;
                end;

-- Pre Year Data Selecting
                begin
                        select
                                                IndexPR1,
                        IndexSP2,
                        IndexIV3,
                        IndexIR4,
                        IndexCU5,
                        IndexLP6
                        into    recidy. IndexPR1,
                        recidy. IndexSP2,
                        recidy. IndexIV3,
                        recidy. IndexIR4,
                        recidy. IndexCU5,
                        recidy. IndexLP6
                        from  Indices
                        where AggregationRange = '5'
                        and     ISIC = recid. ISIC
```

```
                        and ItemType = recid.ItemType
                        and IndexCommodityCode = recid.IndexCommodityCode
                        and Year = nPY_Y
                        and     Month = nPY_M
                        and SurveyScope = inSurveyScope
                                and     Indices.Pre_Rev_Mark = '2';

                exception
                        when no_data_found then
                                recidy.IndexPR1 := null;
        recidy.IndexSP2 := null;
        recidy.IndexIV3 := null;
        recidy.IndexIR4 := null;
        recidy.IndexCU5 := null;
        recidy.IndexLP6 := null;
                end;

-- Item Type 1 Process
                if      recid.ItemType = '1' then

-- Insert Index List Table(PR1)
                        p_InsIndexList(inSeq,'1','1',inYear,inMonth,recid.ISIC,recid.Index
CommodityCode,
                                null,reccm.CommodityShortName,
                                recid.IndexPR1,recidm.IndexPR1,recidy.IndexPR1,recgrm.GR_P
R1,recgry.GR_PR1,
                                recc2m.CD_PR1,recc2y.CD_PR1,recc2m.CR_PR1,recc2y.CR_PR1,
                                recids.IndexPR1,recidms.IndexPR1,recidys.IndexPR1,recgrms.
GR_PR1,recgrys.GR_PR1,
                                recc2ms.CD_PR1,recc2ys.CD_PR1,recc2ms.CR_PR1,recc2ys.CR_PR
1);

-- Insert Index List Table(SP2)
                        p_InsIndexList(inSeq,'1','2',inYear,inMonth,recid.ISIC,recid.Index
CommodityCode,
                                null,reccm.CommodityShortName,
                                recid.IndexSP2,recidm.IndexSP2,recidy.IndexSP2,recgrm.GR_S
P2,recgry.GR_SP2,
                                recc2m.CD_SP2,recc2y.CD_SP2,recc2m.CR_SP2,recc2y.CR_SP2,
                                recids.IndexSP2,recidms.IndexSP2,recidys.IndexSP2,recgrms.
GR_SP2,recgrys.GR_SP2,
                                recc2ms.CD_SP2,recc2ys.CD_SP2,recc2ms.CR_SP2,recc2ys.CR_SP
2);

-- Insert Index List Table(IV3)
                        p_InsIndexList(inSeq,'1','3',inYear,inMonth,recid.ISIC,recid.Index
CommodityCode,
                                null,reccm.CommodityShortName,
                                recid.IndexIV3,recidm.IndexIV3,recidy.IndexIV3,recgrm.GR_I
V3,recgry.GR_IV3,
                                recc2m.CD_IV3,recc2y.CD_IV3,recc2m.CR_IV3,recc2y.CR_IV3,
                                recids.IndexIV3,recidms.IndexIV3,recidys.IndexIV3,recgrms.
GR_IV3,recgrys.GR_IV3,
                                recc2ms.CD_IV3,recc2ys.CD_IV3,recc2ms.CR_IV3,recc2ys.CR_IV
3);

-- Insert Index List Table(IR4)
                        p_InsIndexList(inSeq,'1','4',inYear,inMonth,recid.ISIC,recid.Index
CommodityCode,
                                null,reccm.CommodityShortName,
                                recid.IndexIR4,recidm.IndexIR4,recidy.IndexIR4,recgrm.GR_I
R4,recgry.GR_IR4,
                                recc2m.CD_IR4,recc2y.CD_IR4,recc2m.CR_IR4,recc2y.CR_IR4,
                                recids.IndexIR4,recidms.IndexIR4,recidys.IndexIR4,recgrms.
GR_IR4,recgrys.GR_IR4,
                                recc2ms.CD_IR4,recc2ys.CD_IR4,recc2ms.CR_IR4,recc2ys.CR_IR
4);

-- Item Type 6 Process
                elsif   recid.ItemType = '6' then

-- Insert Index List Table(CU5)
                        p_InsIndexList(inSeq,'1','5',inYear,inMonth,recid.ISIC,recid.Index
CommodityCode,
                                null,reccm.CommodityShortName,
                                recid.IndexCU5,recidm.IndexCU5,recidy.IndexCU5,recgrm.GR_C
U5,recgry.GR_CU5,
```

```
                                        recc2m. CD_CU5, recc2y. CD_CU5, recc2m. CR_CU5, recc2y. CR_CU5,
                                        recids. IndexCU5, recidms. IndexCU5, recidys. IndexCU5, recgrms.
GR_CU5, recgrys. GR_CU5,
                                        recc2ms. CD_CU5, recc2ys. CD_CU5, recc2ms. CR_CU5, recc2ys. CR_CU
5);
                end if;

-- Fetch Indices Table
                fetch   csrid
                        into    recid. ISIC, recid. ItemType, recid. IndexCommodityCode,
                recid. IndexPR1, recid. IndexSP2, recid. IndexIV3,
                                                recid. IndexIR4, recid. IndexCU5, recid. IndexL
P6,
                                                recgrm. GR_PR1, recgrm. GR_SP2, recgrm. GR_IV3,
                                                recgrm. GR_IR4, recgrm. GR_CU5, recgrm. GR_LP6,
                                                recgry. GR_PR1, recgry. GR_SP2, recgry. GR_IV3,
                                                recgry. GR_IR4, recgry. GR_CU5, recgry. GR_LP6,
                                                recc2m. CD_PR1, recc2m. CD_SP2, recc2m. CD_IV3,
                                                recc2m. CD_IR4, recc2m. CD_CU5, recc2m. CD_LP6,
                                                recc2m. CR_PR1, recc2m. CR_SP2, recc2m. CR_IV3,
                                                recc2m. CR_IR4, recc2m. CR_CU5, recc2m. CR_LP6,
                                                recc2y. CD_PR1, recc2y. CD_SP2, recc2y. CD_IV3,
                                                recc2y. CD_IR4, recc2y. CD_CU5, recc2y. CD_LP6,
                                                recc2y. CR_PR1, recc2y. CR_SP2, recc2y. CR_IV3,
                                                recc2y. CR_IR4, recc2y. CR_CU5, recc2y. CR_LP6,
reccm. CommodityShortName;

                end loop;

        close csrid;

end;

-----------------------------------------------------------
-- Industry In All Manifacturing Data Get
-----------------------------------------------------------
procedure p_IndustryInAll (
        inSeq                                                   in      number,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2) is

-- Record Area Define
        recid           Indices%rowtype;
        recidm  Indices%rowtype;
        recidy  Indices%rowtype;
        recgrm  growthRate%rowtype;
        recgry  growthRate%rowtype;
        recc1m  ContributionForAll1%rowtype;
        recc1y  ContributionForAll2%rowtype;
        recam           AggregationMaster%rowtype;

        recids  Indices%rowtype;
        recidms Indices%rowtype;
        recidys Indices%rowtype;
        recgrms growthRate%rowtype;
        recgrys growthRate%rowtype;
        recc1ms ContributionForAll2%rowtype;
        recc1ys ContributionForAll2%rowtype;

-- Work Area
        nPM_Y           number;
        nPM_M           number;
        nPY_Y           number;
        nPY_M           number;

-- Indices Table Cursor
        Cursor csrid is
                select  Indices. ISIC, Indices. INDEXPR1, Indices. INDEXSP2, Indices. INDEXIV3,
                        Indices. INDEXIR4, Indices. INDEXCU5, Indices. INDEXLP6,
                                                a. GR_PR1, a. GR_SP2, a. GR_IV3, a. GR_IR4, a. GR_C
U5, a. GR_LP6,
                                                b. GR_PR1, b. GR_SP2, b. GR_IV3, b. GR_IR4, b. GR_C
U5, b. GR_LP6,
                                                c. CD_PR1, c. CD_SP2, c. CD_IV3, c. CD_IR4, c. CD_C
U5, c. CD_LP6,
```

```
                                                        c. CR_PR1, c. CR_SP2, c. CR_IV3, c. CR_IR4, c. CR_C
U5, c. CR_LP6,
                                                        d. CD_PR1, d. CD_SP2, d. CD_IV3, d. CD_IR4, d. CD_C
U5, d. CD_LP6,
                                                        d. CR_PR1, d. CR_SP2, d. CR_IV3, d. CR_IR4, d. CR_C
U5, d. CR_LP6,
                                                        AggregationMaster. IndustryTypeShortName
        from  Indices, GrowthRate a, GrowthRate b,
                                ContributionForAll1 c, ContributionForAll1 d, AggregationMas
ter
 where  Indices. AggregationRange = a. AggregationRange
   and  Indices. ISIC = a. ISIC
   and  Indices. Year = a. Year
   and  Indices. Month = a. Month
   and  Indices. SurveyScope = a. SurveyScope
   and  Indices. Pre_Rev_Mark = a. Pre_Rev_Mark
   and  a. ComparisonTarget = '0'
   and  Indices. AggregationRange = b. AggregationRange
   and  Indices. ISIC = b. ISIC
   and  Indices. Year = b. Year
   and  Indices. Month = b. Month
   and  Indices. SurveyScope = b. SurveyScope
   and  Indices. Pre_Rev_Mark = b. Pre_Rev_Mark
   and  b. ComparisonTarget = '1'
   and  Indices. ISIC = c. ISIC
   and  Indices. Year = c. Year
   and  Indices. Month = c. Month
   and  Indices. SurveyScope = c. SurveyScope
   and  Indices. Pre_Rev_Mark = c. Pre_Rev_Mark
         and    c. ComparisonTarget = '0'
         and    c. Seq = inSeq
   and  Indices. ISIC = d. ISIC
   and  Indices. Year = d. Year
   and  Indices. Month = d. Month
   and  Indices. SurveyScope = d. SurveyScope
   and  Indices. Pre_Rev_Mark = d. Pre_Rev_Mark
         and    d. ComparisonTarget = '1'
         and    d. Seq = inSeq
   and  Indices. AggregationRange = AggregationMaster. AggregationRange
   and  Indices. ISIC = AggregationMaster. ISIC
         and    Indices. AggregationRange = '3'
         and    Indices. Year = inYear
         and    Indices. Month = inMonth
         and    Indices. SurveyScope = inSurveyScope
         and    Indices. Pre_Rev_Mark = inPre_Rev_Mark;

Begin
--      DBMS_OUTPUT. PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Initial Process
        nPY_Y   := inYear - 1;
        nPY_M   := inMonth;

        if      inMonth = 1 then
                        nPM_Y   := inYear - 1;
                        nPM_M   := 12;
        else
                        nPM_Y   := inYear;
                        nPM_M   := inMonth - 1;
        end if;

-- Summary Current Month Data Selecting
        begin
                select  Indices. INDEXPR1, Indices. INDEXSP2, Indices. INDEXIV3,
                        Indices. INDEXIR4, Indices. INDEXCU5, Indices. INDEXLP6,
                                                        a. GR_PR1, a. GR_SP2, a. GR_IV3, a. GR_IR4, a. GR_C
U5, a. GR_LP6,
                                                        b. GR_PR1, b. GR_SP2, b. GR_IV3, b. GR_IR4, b. GR_C
U5, b. GR_LP6
                into            recids. INDEXPR1, recids. INDEXSP2, recids. INDEXIV3,
                        recids. INDEXIR4, recids. INDEXCU5, recids. INDEXLP6,
                                                        recgrms. GR_PR1, recgrms. GR_SP2, recgrms. GR_I
V3,
                                                        recgrms. GR_IR4, recgrms. GR_CU5, recgrms. GR_L
P6,
                                                        recgrys. GR_PR1, recgrys. GR_SP2, recgrys. GR_I
V3,
                                                        recgrys. GR_IR4, recgrys. GR_CU5, recgrys. GR_L
```

P6

```
                from  Indices, GrowthRate a, GrowthRate b
                where  Indices. AggregationRange = a. AggregationRange
            and  Indices. ISIC = a. ISIC
            and  Indices. Year = a. Year
            and  Indices. Month = a. Month
            and  Indices. SurveyScope = a. SurveyScope
            and  Indices. Pre_Rev_Mark = a. Pre_Rev_Mark
            and  a. ComparisonTarget = '0'
            and  Indices. AggregationRange = b. AggregationRange
            and  Indices. ISIC = b. ISIC
            and  Indices. Year = b. Year
            and  Indices. Month = b. Month
            and  Indices. SurveyScope = b. SurveyScope
            and  Indices. Pre_Rev_Mark = b. Pre_Rev_Mark
            and  b. ComparisonTarget = '1'
                 and    Indices. AggregationRange = '1'
                 and    Indices. Year = inYear
                 and    Indices. Month = inMonth
                 and    Indices. SurveyScope = inSurveyScope
                 and    Indices. Pre_Rev_Mark = inPre_Rev_Mark;

        exception
                when no_data_found then
                        recids. INDEXPR1 := 0;
                        recids. INDEXSP2 := 0;
                        recids. INDEXIV3 := 0;
                        recids. INDEXIR4 := 0;
                        recids. INDEXCU5 := 0;
                        recids. INDEXLP6 := 0;
                        recgrms. GR_PR1  := 0;
                        recgrms. GR_SP2  := 0;
                        recgrms. GR_IV3  := 0;
                        recgrms. GR_IR4  := 0;
                        recgrms. GR_CU5  := 0;
                        recgrms. GR_LP6  := 0;
                        recgrys. GR_PR1  := 0;
                        recgrys. GR_SP2  := 0;
                        recgrys. GR_IV3  := 0;
                        recgrys. GR_IR4  := 0;
                        recgrys. GR_CU5  := 0;
                        recgrys. GR_LP6  := 0;
        end;

-- Summary Pre Month Data Selecting
        begin
                select  IndexPR1,
                        IndexSP2,
                        IndexIV3,
                        IndexIR4,
                        IndexCU5,
                        IndexLP6
                        into    recidms. IndexPR1,
                        recidms. IndexSP2,
                        recidms. IndexIV3,
                        recidms. IndexIR4,
                        recidms. IndexCU5,
                        recidms. IndexLP6
                        from  Indices
                        where AggregationRange = '1'
                        and Year = nPM_Y
                        and     Month = nPM_M
                        and SurveyScope = inSurveyScope
                                and     Indices. Pre_Rev_Mark = '2';

        exception
                when no_data_found then
                        recidms. IndexPR1        := 0;
      recidms. IndexSP2  := 0;
      recidms. IndexIV3  := 0;
      recidms. IndexIR4  := 0;
      recidms. IndexCU5  := 0;
      recidms. IndexLP6  := 0;
        end;

-- Summary Pre Year Data Selecting
        begin
```

```
                select  IndexPR1,
                        IndexSP2,
                        IndexIV3,
                        IndexIR4,
                        IndexCU5,
                        IndexLP6
                        into    recidys.IndexPR1,
                        recidys.IndexSP2,
                        recidys.IndexIV3,
                        recidys.IndexIR4,
                        recidys.IndexCU5,
                        recidys.IndexLP6
                        from  Indices
                        where AggregationRange = '1'
                        and Year = nPY_Y
                        and     Month = nPY_M
                        and SurveyScope = inSurveyScope
                                and     Indices.Pre_Rev_Mark = '2';

        exception
                when no_data_found then
                        recidys.IndexPR1        := null;
      recidys.IndexSP2  := null;
      recidys.IndexIV3  := null;
      recidys.IndexIR4  := null;
      recidys.IndexCU5  := null;
      recidys.IndexLP6  := null;
        end;

-- Summary Contribution CPM Data Selecting
        begin
                select  sum(CD_PR1), sum(CD_SP2), sum(CD_IV3), sum(CD_IR4), sum(CD_CU5), sum(CD
_LP6),
                                                sum(CR_PR1), sum(CR_SP2), sum(CR_IV3), sum(CR
_IR4), sum(CR_CU5), sum(CR_LP6)
                into            recclms.CD_PR1, recclms.CD_SP2, recclms.CD_IV3,
                                                recclms.CD_IR4, recclms.CD_CU5, recclms.CD_L
P6,
                                                recclms.CR_PR1, recclms.CR_SP2, recclms.CR_I
V3,
                                                recclms.CR_IR4, recclms.CR_CU5, recclms.CR_L
P6
                from    ContributionForAll1
                where           Year = inYear
                        and             Month = inMonth
                        and             SurveyScope = inSurveyScope
                        and             Pre_Rev_Mark = inPre_Rev_Mark
                        and             ComparisonTarget = '0'
                        and             Seq = inSeq;

        exception
                when no_data_found then
                        recclms.CD_PR1 := 0;
                        recclms.CD_SP2 := 0;
                        recclms.CD_IV3 := 0;
                        recclms.CD_IR4 := 0;
                        recclms.CD_CU5 := 0;
                        recclms.CD_LP6 := 0;
                        recclms.CR_PR1 := 0;
                        recclms.CR_SP2 := 0;
                        recclms.CR_IV3 := 0;
                        recclms.CR_IR4 := 0;
                        recclms.CR_CU5 := 0;
                        recclms.CR_LP6 := 0;
        end;

-- Summary Contribution CPY Data Selecting
        begin
                select  sum(CD_PR1), sum(CD_SP2), sum(CD_IV3), sum(CD_IR4), sum(CD_CU5), sum(CD
_LP6),
                                                sum(CR_PR1), sum(CR_SP2), sum(CR_IV3), sum(CR
_IR4), sum(CR_CU5), sum(CR_LP6)
                into            recclys.CD_PR1, recclys.CD_SP2, recclys.CD_IV3,
                                                recclys.CD_IR4, recclys.CD_CU5, recclys.CD_L
P6,
                                                recclys.CR_PR1, recclys.CR_SP2, recclys.CR_I
V3,
```

```
                                                recclys. CR_IR4, recclys. CR_CU5, recclys. CR_L
P6
                from    ContributionForAll1
                where           Year = inYear
                        and             Month = inMonth
                        and             SurveyScope = inSurveyScope
                        and             Pre_Rev_Mark = inPre_Rev_Mark
                        and             ComparisonTarget = '1'
                        and             Seq = inSeq;

        exception
                when no_data_found then
                        recclys. CD_PR1 := 0;
                        recclys. CD_SP2 := 0;
                        recclys. CD_IV3 := 0;
                        recclys. CD_IR4 := 0;
                        recclys. CD_CU5 := 0;
                        recclys. CD_LP6 := 0;
                        recclys. CR_PR1 := 0;
                        recclys. CR_SP2 := 0;
                        recclys. CR_IV3 := 0;
                        recclys. CR_IR4 := 0;
                        recclys. CR_CU5 := 0;
                        recclys. CR_LP6 := 0;
        end;

-- Indices Cursor Open Fetch
        open    csrid;
        fetch   csrid
                into    recid. ISIC, recid. IndexPR1, recid. IndexSP2, recid. IndexIV3,
                                        recid. IndexIR4, recid. IndexCU5, recid. IndexLP6,
                                        recgrm. GR_PR1, recgrm. GR_SP2, recgrm. GR_IV3,
                                        recgrm. GR_IR4, recgrm. GR_CU5, recgrm. GR_LP6,
                                        recgry. GR_PR1, recgry. GR_SP2, recgry. GR_IV3,
                                        recgry. GR_IR4, recgry. GR_CU5, recgry. GR_LP6,
                                        recclm. CD_PR1, recclm. CD_SP2, recclm. CD_IV3,
                                        recclm. CD_IR4, recclm. CD_CU5, recclm. CD_LP6,
                                        recclm. CR_PR1, recclm. CR_SP2, recclm. CR_IV3,
                                        recclm. CR_IR4, recclm. CR_CU5, recclm. CR_LP6,
                                        reccly. CD_PR1, reccly. CD_SP2, reccly. CD_IV3,
                                        reccly. CD_IR4, reccly. CD_CU5, reccly. CD_LP6,
                                        reccly. CR_PR1, reccly. CR_SP2, reccly. CR_IV3,
                                        reccly. CR_IR4, reccly. CR_CU5, reccly. CR_LP6, recam. In
dustryTypeShortName;

-- Indices Cursor Fetch Looping
        while csrid%found loop

-- Pre Month Data Selecting
                begin
                        select
                                                IndexPR1,
                        IndexSP2,
                        IndexIV3,
                        IndexIR4,
                        IndexCU5,
                        IndexLP6
                        into    recidm. IndexPR1,
                        recidm. IndexSP2,
                        recidm. IndexIV3,
                        recidm. IndexIR4,
                        recidm. IndexCU5,
                        recidm. IndexLP6
                        from  Indices
                        where AggregationRange = '3'
                        and     ISIC = recid. ISIC
                        and Year = nPM_Y
                        and     Month = nPM_M
                        and SurveyScope = inSurveyScope
                                and     Indices. Pre_Rev_Mark = '2';

                exception
                        when no_data_found then
                                recidm. IndexPR1 := 0;
        recidm. IndexSP2 := 0;
        recidm. IndexIV3 := 0;
        recidm. IndexIR4 := 0;
```

```
        recidm.IndexCU5 := 0;
        recidm.IndexLP6 := 0;
                end;

-- Pre Year Data Selecting
                begin
                        select
                                                IndexPR1,
                        IndexSP2,
                        IndexIV3,
                        IndexIR4,
                        IndexCU5,
                        IndexLP6
                        into    recidy.IndexPR1,
                        recidy.IndexSP2,
                        recidy.IndexIV3,
                        recidy.IndexIR4,
                        recidy.IndexCU5,
                        recidy.IndexLP6
                        from  Indices
                        where AggregationRange = '3'
                        and     ISIC = recid.ISIC
                        and Year = nPY_Y
                        and     Month = nPY_M
                        and SurveyScope = inSurveyScope
                                and     Indices.Pre_Rev_Mark = '2';

                exception
                        when no_data_found then
                                recidy.IndexPR1 := null;
        recidy.IndexSP2 := null;
        recidy.IndexIV3 := null;
        recidy.IndexIR4 := null;
        recidy.IndexCU5 := null;
        recidy.IndexLP6 := null;
                end;

-- Insert Index List Table(PR1)
                p_InsIndexList(inSeq,'2','1',inYear,inMonth,substr(recid.ISIC,1,4),'XXX',
                        recam.IndustryTypeShortName,null,
                        recid.IndexPR1,recidm.IndexPR1,recidy.IndexPR1,recgrm.GR_PR1,recgr
y.GR_PR1,
                        recclm.CD_PR1,recccly.CD_PR1,recclm.CR_PR1,recccly.CR_PR1,
                        recids.IndexPR1,recidms.IndexPR1,recidys.IndexPR1,recgrms.GR_PR1,r
ecgrys.GR_PR1,
                        recclms.CD_PR1,recclys.CD_PR1,recclms.CR_PR1,recclys.CR_PR1);

-- Insert Index List Table(SP2)
                p_InsIndexList(inSeq,'2','2',inYear,inMonth,substr(recid.ISIC,1,4),'XXX',
                        recam.IndustryTypeShortName,null,
                        recid.IndexSP2,recidm.IndexSP2,recidy.IndexSP2,recgrm.GR_SP2,recgr
y.GR_SP2,
                        recclm.CD_SP2,recccly.CD_SP2,recclm.CR_SP2,recccly.CR_SP2,
                        recids.IndexSP2,recidms.IndexSP2,recidys.IndexSP2,recgrms.GR_SP2,r
ecgrys.GR_SP2,
                        recclms.CD_SP2,recclys.CD_SP2,recclms.CR_SP2,recclys.CR_SP2);

-- Insert Index List Table(IV3)
                p_InsIndexList(inSeq,'2','3',inYear,inMonth,substr(recid.ISIC,1,4),'XXX',
                        recam.IndustryTypeShortName,null,
                        recid.IndexIV3,recidm.IndexIV3,recidy.IndexIV3,recgrm.GR_IV3,recgr
y.GR_IV3,
                        recclm.CD_IV3,recccly.CD_IV3,recclm.CR_IV3,recccly.CR_IV3,
                        recids.IndexIV3,recidms.IndexIV3,recidys.IndexIV3,recgrms.GR_IV3,r
ecgrys.GR_IV3,
                        recclms.CD_IV3,recclys.CD_IV3,recclms.CR_IV3,recclys.CR_IV3);

-- Insert Index List Table(IR4)
                p_InsIndexList(inSeq,'2','4',inYear,inMonth,substr(recid.ISIC,1,4),'XXX',
                        recam.IndustryTypeShortName,null,
                        recid.IndexIR4,recidm.IndexIR4,recidy.IndexIR4,recgrm.GR_IR4,recgr
y.GR_IR4,
                        recclm.CD_IR4,recccly.CD_IR4,recclm.CR_IR4,recccly.CR_IR4,
                        recids.IndexIR4,recidms.IndexIR4,recidys.IndexIR4,recgrms.GR_IR4,r
ecgrys.GR_IR4,
                        recclms.CD_IR4,recclys.CD_IR4,recclms.CR_IR4,recclys.CR_IR4);
```

```
-- Insert Index List Table(CU5)
                p_InsIndexList(inSeq, '2', '5', inYear, inMonth, substr(recid. ISIC, 1, 4), 'XXX',
                        recam. IndustryTypeShortName, null,
                        recid. IndexCU5, recidm. IndexCU5, recidy. IndexCU5, recgrm. GR_CU5, recgr
y. GR_CU5,
                        recclm. CD_CU5, reccly. CD_CU5, recclm. CR_CU5, reccly. CR_CU5,
                        recids. IndexCU5, recidms. IndexCU5, recidys. IndexCU5, recgrms. GR_CU5, r
ecgrys. GR_CU5,
                        recclms. CD_CU5, recclys. CD_CU5, recclms. CR_CU5, recclys. CR_CU5);

-- Insert Index List Table(LP6)
                p_InsIndexList(inSeq, '2', '6', inYear, inMonth, substr(recid. ISIC, 1, 4), 'XXX',
                        recam. IndustryTypeShortName, null,
                        recid. IndexLP6, recidm. IndexLP6, recidy. IndexLP6, recgrm. GR_LP6, recgr
y. GR_LP6,
                        recclm. CD_LP6, reccly. CD_LP6, recclm. CR_LP6, reccly. CR_LP6,
                        recids. IndexLP6, recidms. IndexLP6, recidys. IndexLP6, recgrms. GR_LP6, r
ecgrys. GR_LP6,
                        recclms. CD_LP6, recclys. CD_LP6, recclms. CR_LP6, recclys. CR_LP6);

-- Fetch Indices Table
        fetch   csrid
                into    recid. ISIC, recid. IndexPR1, recid. IndexSP2, recid. IndexIV3,
                                        recid. IndexIR4, recid. IndexCU5, recid. IndexLP6,
                                        recgrm. GR_PR1, recgrm. GR_SP2, recgrm. GR_IV3,
                                        recgrm. GR_IR4, recgrm. GR_CU5, recgrm. GR_LP6,
                                        recgry. GR_PR1, recgry. GR_SP2, recgry. GR_IV3,
                                        recgry. GR_IR4, recgry. GR_CU5, recgry. GR_LP6,
                                        recclm. CD_PR1, recclm. CD_SP2, recclm. CD_IV3,
                                        recclm. CD_IR4, recclm. CD_CU5, recclm. CD_LP6,
                                        recclm. CR_PR1, recclm. CR_SP2, recclm. CR_IV3,
                                        recclm. CR_IR4, recclm. CR_CU5, recclm. CR_LP6,
                                        reccly. CD_PR1, reccly. CD_SP2, reccly. CD_IV3,
                                        reccly. CD_IR4, reccly. CD_CU5, reccly. CD_LP6,
                                        reccly. CR_PR1, reccly. CR_SP2, reccly. CR_IV3,
                                        reccly. CR_IR4, reccly. CR_CU5, reccly. CR_LP6, recam. In
dustryTypeShortName;

                end loop;

        close csrid;

end;

-------------------------------------------------------
-- Commodity In Industry Data Making
-------------------------------------------------------
procedure p_CommInIndustry (
        inSeq                                                   in      number,
        inAggregationRange      in      Varchar2,
        inISIC                                          in      Varchar2,
        inYear                                          in      number,
        inMonth                                         in      number,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2) is

-- Record Area Define
        recid           Indices%rowtype;
        recidm  Indices%rowtype;
        recidy  Indices%rowtype;
        recgrm  growthRate%rowtype;
        recgry  growthRate%rowtype;
        reccim  ContributionForIndustry%rowtype;
        recciy  ContributionForIndustry%rowtype;
        reccm           CommodityMaster%rowtype;

        recids  Indices%rowtype;
        recidms Indices%rowtype;
        recidys Indices%rowtype;
        recgrms growthRate%rowtype;
        recgrys growthRate%rowtype;
        reccims ContributionForIndustry%rowtype;
        recciys ContributionForIndustry%rowtype;
        recams  AggregationMaster%rowtype;

-- Work Area
        nPM_Y           number;
```

```
        nPM_M           number;
        nPY_Y           number;
        nPY_M           number;

        sISIC2d         Varchar2(2);
        sISIC4d         Varchar2(4);

        sOldISIC        Varchar2(4);

-- Indices Table Cursor
        Cursor csrid is
                select  Indices.ISIC, Indices.ItemType, Indices.IndexCommodityCode,
                        Indices.INDEXPR1, Indices.INDEXSP2, Indices.INDEXIV3,
                        Indices.INDEXIR4, Indices.INDEXCU5, Indices.INDEXLP6,
                                                a.GR_PR1, a.GR_SP2, a.GR_IV3, a.GR_IR4, a.GR_C
U5, a.GR_LP6,
                                                b.GR_PR1, b.GR_SP2, b.GR_IV3, b.GR_IR4, b.GR_C
U5, b.GR_LP6,
                                                c.CD_PR1, c.CD_SP2, c.CD_IV3, c.CD_IR4, c.CD_C
U5, c.CD_LP6,
                                                c.CR_PR1, c.CR_SP2, c.CR_IV3, c.CR_IR4, c.CR_C
U5, c.CR_LP6,
                                                d.CD_PR1, d.CD_SP2, d.CD_IV3, d.CD_IR4, d.CD_C
U5, d.CD_LP6,
                                                d.CR_PR1, d.CR_SP2, d.CR_IV3, d.CR_IR4, d.CR_C
U5, d.CR_LP6,
                                                CommodityMaster.CommodityShortName
        from  Indices, GrowthRate a, GrowthRate b,
                                ContributionForIndustry c, ContributionForIndustry d, Commod
ityMaster
        where Indices.AggregationRange = a.AggregationRange
   and  Indices.ISIC = a.ISIC
   and  Indices.ItemType = a.ItemType
   and  Indices.IndexCommodityCode = a.IndexCommodityCode
   and  Indices.Year = a.Year
   and  Indices.Month = a.Month
   and  Indices.SurveyScope = a.SurveyScope
   and  Indices.Pre_Rev_Mark = a.Pre_Rev_Mark
   and  a.ComparisonTarget = '0'
   and  Indices.AggregationRange = b.AggregationRange
   and  Indices.ISIC = b.ISIC
   and  Indices.ItemType = b.ItemType
   and  Indices.IndexCommodityCode = b.IndexCommodityCode
   and  Indices.Year = b.Year
   and  Indices.Month = b.Month
   and  Indices.SurveyScope = b.SurveyScope
   and  Indices.Pre_Rev_Mark = b.Pre_Rev_Mark
   and  b.ComparisonTarget = '1'
   and  Indices.ISIC = c.ISIC
   and  Indices.ItemType = c.ItemType
   and  Indices.IndexCommodityCode = c.CommodityCode
   and  Indices.Year = c.Year
   and  Indices.Month = c.Month
   and  Indices.SurveyScope = c.SurveyScope
   and  Indices.Pre_Rev_Mark = c.Pre_Rev_Mark
         and    c.ComparisonTarget = '0'
         and    c.Seq = inSeq
   and  Indices.ISIC = d.ISIC
   and  Indices.ItemType = d.ItemType
   and  Indices.IndexCommodityCode = d.CommodityCode
   and  Indices.Year = d.Year
   and  Indices.Month = d.Month
   and  Indices.SurveyScope = d.SurveyScope
   and  Indices.Pre_Rev_Mark = d.Pre_Rev_Mark
         and    d.ComparisonTarget = '1'
         and    d.Seq = inSeq
   and  Indices.ISIC = CommodityMaster.ISIC
   and  Indices.ItemType = CommodityMaster.ItemType
   and  Indices.IndexCommodityCode = CommodityMaster.CommodityCode
         and    Indices.AggregationRange = '5'
         and    Indices.ItemType in('1','6')
         and    Indices.Year = inYear
         and    Indices.Month = inMonth
         and    Indices.SurveyScope = inSurveyScope
         and    Indices.Pre_Rev_Mark = inPre_Rev_Mark
        order by Indices.ISIC;
```

```
Begin
--      DBMS_OUTPUT.PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Initial Process
        nPY_Y   := inYear - 1;
        nPY_M   := inMonth;

        if      inMonth = 1 then
                        nPM_Y   := inYear - 1;
                        nPM_M   := 12;
        else
                        nPM_Y   := inYear;
                        nPM_M   := inMonth - 1;
        end if;

        sISIC2d := substr(inISIC, 1, 2);
        sISIC4d := substr(inISIC, 1, 4);

        sOldISIC := null;

-- Indices Cursor Open Fetch
        open    csrid;
        fetch   csrid
                into    recid.ISIC, recid.ItemType, recid.IndexCommodityCode,
                recid.IndexPR1, recid.IndexSP2, recid.IndexIV3,
                                        recid.IndexIR4, recid.IndexCU5, recid.IndexLP6,
                                        recgrm.GR_PR1, recgrm.GR_SP2, recgrm.GR_IV3,
                                        recgrm.GR_IR4, recgrm.GR_CU5, recgrm.GR_LP6,
                                        recgry.GR_PR1, recgry.GR_SP2, recgry.GR_IV3,
                                        recgry.GR_IR4, recgry.GR_CU5, recgry.GR_LP6,
                                        reccim.CD_PR1, reccim.CD_SP2, reccim.CD_IV3,
                                        reccim.CD_IR4, reccim.CD_CU5, reccim.CD_LP6,
                                        reccim.CR_PR1, reccim.CR_SP2, reccim.CR_IV3,
                                        reccim.CR_IR4, reccim.CR_CU5, reccim.CR_LP6,
                                        recciy.CD_PR1, recciy.CD_SP2, recciy.CD_IV3,
                                        recciy.CD_IR4, recciy.CD_CU5, recciy.CD_LP6,
                                        recciy.CR_PR1, recciy.CR_SP2, recciy.CR_IV3,
                                        recciy.CR_IR4, recciy.CR_CU5, recciy.CR_LP6, reccm.Co
mmodityShortName;

-- Indices Cursor Fetch Looping
        while csrid%found loop
                if      (inAggregationRange = '1')
                or      (inAggregationRange = '2' and substr(recid.ISIC, 1, 2) = sISIC2d)
                or      (inAggregationRange = '3' and substr(recid.ISIC, 1, 4) = sISIC4d) th
en
                        if      sOldISIC = substr(recid.ISIC, 1, 4) then
                                null;
                        else
                                sOldISIC := substr(recid.ISIC, 1, 4);

-- Summary Current Month Data Selecting
                                begin
                                        select  Indices.INDEXPR1, Indices.INDEXSP2, Indices.
INDEXIV3,
                                                Indices.INDEXIR4, Indices.INDEXCU5, Indices.
INDEXLP6,
                                                                        a.GR_PR1, a.GR_SP2,
a.GR_IV3, a.GR_IR4, a.GR_CU5, a.GR_LP6,
                                                                        b.GR_PR1, b.GR_SP2,
b.GR_IV3, b.GR_IR4, b.GR_CU5, b.GR_LP6,
                                                                        AggregationMaster.I
ndustryTypeShortName
                                        into            recids.INDEXPR1, recids.INDEXSP2, re
cids.INDEXIV3,
                                                recids.INDEXIR4, recids.INDEXCU5, recids.IND
EXLP6,
                                                                        recgrms.GR_PR1, recg
rms.GR_SP2, recgrms.GR_IV3,
                                                                        recgrms.GR_IR4, recg
rms.GR_CU5, recgrms.GR_LP6,
                                                                        recgrys.GR_PR1, recg
rys.GR_SP2, recgrys.GR_IV3,
                                                                        recgrys.GR_IR4, recg
rys.GR_CU5, recgrys.GR_LP6,
                                                                        recams.IndustryType
ShortName
```

```
                                        from    Indices, GrowthRate a, GrowthRate b, Aggregat
ionMaster
                                        where  Indices. AggregationRange = a. AggregationRan
ge
                                  and  Indices. ISIC = a. ISIC
                                  and  Indices. Year = a. Year
                                  and  Indices. Month = a. Month
                                  and  Indices. SurveyScope = a. SurveyScope
                                  and  Indices. Pre_Rev_Mark = a. Pre_Rev_Mark
                                  and  a. ComparisonTarget = '0'
                                  and  Indices. AggregationRange = b. AggregationRange
                                  and  Indices. ISIC = b. ISIC
                                  and  Indices. Year = b. Year
                                  and  Indices. Month = b. Month
                                  and  Indices. SurveyScope = b. SurveyScope
                                  and  Indices. Pre_Rev_Mark = b. Pre_Rev_Mark
                                  and  b. ComparisonTarget = '1'
                                  and  Indices. AggregationRange = AggregationMaster. Aggre
gationRange
                                and  Indices. ISIC = AggregationMaster. ISIC
                                        and    Indices. AggregationRange = '3'
                                        and    Indices. ISIC like substr(recid. ISIC, 1, 4) ||
'%'
                                        and    Indices. Year = inYear
                                        and    Indices. Month = inMonth
                                        and    Indices. SurveyScope = inSurveyScope
                                        and    Indices. Pre_Rev_Mark = inPre_Rev_Mark;

                                exception
                                        when no_data_found then
                                                recids. INDEXPR1 := 0;
                                                recids. INDEXSP2 := 0;
                                                recids. INDEXIV3 := 0;
                                                recids. INDEXIR4 := 0;
                                                recids. INDEXCU5 := 0;
                                                recids. INDEXLP6 := 0;
                                                recgrms. GR_PR1  := 0;
                                                recgrms. GR_SP2  := 0;
                                                recgrms. GR_IV3  := 0;
                                                recgrms. GR_IR4  := 0;
                                                recgrms. GR_CU5  := 0;
                                                recgrms. GR_LP6  := 0;
                                                recgrys. GR_PR1  := 0;
                                                recgrys. GR_SP2  := 0;
                                                recgrys. GR_IV3  := 0;
                                                recgrys. GR_IR4  := 0;
                                                recgrys. GR_CU5  := 0;
                                                recgrys. GR_LP6  := 0;
                                                recams. IndustryTypeShortName := null;
                                end;

-- Summary Pre Month Data Selecting
                                begin
                                        select  IndexPR1,
                                                IndexSP2,
                                                IndexIV3,
                                                IndexIR4,
                                                IndexCU5,
                                                IndexLP6
                                                into    recidms. IndexPR1,
                                                recidms. IndexSP2,
                                                recidms. IndexIV3,
                                                recidms. IndexIR4,
                                                recidms. IndexCU5,
                                                recidms. IndexLP6
                                                from  Indices
                                                where AggregationRange = '3'
                                                and     ISIC like substr(recid. ISIC, 1, 4) ||
'%'
                                                and Year = nPM_Y
                                                and     Month = nPM_M
                                                and SurveyScope = inSurveyScope
                                                        and     Indices. Pre_Rev_Mark = '2';

                                exception
                                        when no_data_found then
                                                recidms. IndexPR1        := 0;
```

```
                                recidms. IndexSP2  := 0;
                                recidms. IndexIV3  := 0;
                                recidms. IndexIR4  := 0;
                                recidms. IndexCU5  := 0;
                                recidms. IndexLP6  := 0;
                                  end;

-- Summary Pre Year Data Selecting
                                  begin
                                          select  IndexPR1,
                                                  IndexSP2,
                                                  IndexIV3,
                                                  IndexIR4,
                                                  IndexCU5,
                                                  IndexLP6
                                                  into    recidys. IndexPR1,
                                                  recidys. IndexSP2,
                                                  recidys. IndexIV3,
                                                  recidys. IndexIR4,
                                                  recidys. IndexCU5,
                                                  recidys. IndexLP6
                                                  from  Indices
                                                  where AggregationRange = '3'
                                                  and     ISIC like substr(recid. ISIC, 1, 4) ||
'%'
                                                  and Year = nPY_Y
                                                  and     Month = nPY_M
                                                  and SurveyScope = inSurveyScope
                                                          and     Indices. Pre_Rev_Mark = '2';

                                  exception
                                          when no_data_found then
                                                  recidys. IndexPR1        := null;
                                recidys. IndexSP2  := null;
                                recidys. IndexIV3  := null;
                                recidys. IndexIR4  := null;
                                recidys. IndexCU5  := null;
                                recidys. IndexLP6  := null;
                                  end;

-- Summary Contribution CPM Data Selecting
                                  begin
                                          select  sum(CD_PR1), sum(CD_SP2), sum(CD_IV3), sum(CD
_IR4), sum(CD_CU5), sum(CD_LP6),
                                                                                  sum(CR_PR1), sum(CR_
SP2), sum(CR_IV3), sum(CR_IR4), sum(CR_CU5), sum(CR_LP6)
                                          into            reccims. CD_PR1, reccims. CD_SP2, recc
ims. CD_IV3,
                                                                                  reccims. CD_IR4, recc
ims. CD_CU5, reccims. CD_LP6,
                                                                                  reccims. CR_PR1, recc
ims. CR_SP2, reccims. CR_IV3,
                                                                                  reccims. CR_IR4, recc
ims. CR_CU5, reccims. CR_LP6
                                          from    ContributionForIndustry
                                          where           ISIC like substr(recid. ISIC, 1, 4) ||
'%'
                                            and           Year = inYear
                                                  and             Month = inMonth
                                                  and             SurveyScope = inSurveyScope
                                                  and             Pre_Rev_Mark = inPre_Rev_M
ark
                                                  and             ComparisonTarget = '0'
                                                  and             Seq = inSeq;

                                  exception
                                          when no_data_found then
                                                  reccims. CD_PR1 := 0;
                                                  reccims. CD_SP2 := 0;
                                                  reccims. CD_IV3 := 0;
                                                  reccims. CD_IR4 := 0;
                                                  reccims. CD_CU5 := 0;
                                                  reccims. CD_LP6 := 0;
                                                  reccims. CR_PR1 := 0;
                                                  reccims. CR_SP2 := 0;
                                                  reccims. CR_IV3 := 0;
                                                  reccims. CR_IR4 := 0;
```

```
                                        reccims.CR_CU5 := 0;
                                        reccims.CR_LP6 := 0;
                        end;

-- Summary Contribution CPY Data Selecting
                        begin
                                select  sum(CD_PR1), sum(CD_SP2), sum(CD_IV3), sum(CD
_IR4), sum(CD_CU5), sum(CD_LP6),
                                                                sum(CR_PR1), sum(CR_
SP2), sum(CR_IV3), sum(CR_IR4), sum(CR_CU5), sum(CR_LP6)
                                into            recciys.CD_PR1, recciys.CD_SP2, recc
iys.CD_IV3,
                                                                recciys.CD_IR4, recc
iys.CD_CU5, recciys.CD_LP6,
                                                                recciys.CR_PR1, recc
iys.CR_SP2, recciys.CR_IV3,
                                                                recciys.CR_IR4, recc
iys.CR_CU5, recciys.CR_LP6
                                from    ContributionForIndustry
                                where           ISIC like substr(recid.ISIC, 1, 4) ||
'%'
                                        and             Year = inYear
                                        and             Month = inMonth
                                        and             SurveyScope = inSurveyScope
                                        and             Pre_Rev_Mark = inPre_Rev_M
ark
                                        and             ComparisonTarget = '1'
                                        and             Seq = inSeq;

                        exception
                                when no_data_found then
                                        recciys.CD_PR1 := 0;
                                        recciys.CD_SP2 := 0;
                                        recciys.CD_IV3 := 0;
                                        recciys.CD_IR4 := 0;
                                        recciys.CD_CU5 := 0;
                                        recciys.CD_LP6 := 0;
                                        recciys.CR_PR1 := 0;
                                        recciys.CR_SP2 := 0;
                                        recciys.CR_IV3 := 0;
                                        recciys.CR_IR4 := 0;
                                        recciys.CR_CU5 := 0;
                                        recciys.CR_LP6 := 0;
                        end;
                end if;

-- Pre Month Data Selecting
                begin
                        select
                                        IndexPR1,
                        IndexSP2,
                        IndexIV3,
                        IndexIR4,
                        IndexCU5,
                        IndexLP6
                        into    recidm.IndexPR1,
                        recidm.IndexSP2,
                        recidm.IndexIV3,
                        recidm.IndexIR4,
                        recidm.IndexCU5,
                        recidm.IndexLP6
                        from  Indices
                        where AggregationRange = '5'
                        and     ISIC = recid.ISIC
                        and ItemType = recid.ItemType
                        and IndexCommodityCode = recid.IndexCommodityCode
                        and Year = nPM_Y
                        and     Month = nPM_M
                        and SurveyScope = inSurveyScope
                                and     Indices.Pre_Rev_Mark = '2';

                exception
                        when no_data_found then
                                recidm.IndexPR1 := 0;
        recidm.IndexSP2         := 0;
        recidm.IndexIV3         := 0;
        recidm.IndexIR4         := 0;
```

```
        recidm. IndexCU5        := 0;
        recidm. IndexLP6        := 0;
            end;

-- Pre Year Data Selecting
            begin
                select
                                                IndexPR1,
                IndexSP2,
                IndexIV3,
                IndexIR4,
                IndexCU5,
                IndexLP6
                into    recidy. IndexPR1,
                recidy. IndexSP2,
                recidy. IndexIV3,
                recidy. IndexIR4,
                recidy. IndexCU5,
                recidy. IndexLP6
                from  Indices
                where AggregationRange = '5'
                and     ISIC = recid. ISIC
                and ItemType = recid. ItemType
                and IndexCommodityCode = recid. IndexCommodityCode
                and Year = nPY_Y
                and     Month = nPY_M
                and SurveyScope = inSurveyScope
                        and     Indices. Pre_Rev_Mark = '2';

            exception
                when no_data_found then
                        recidy. IndexPR1 := null;
        recidy. IndexSP2        := null;
        recidy. IndexIV3        := null;
        recidy. IndexIR4        := null;
        recidy. IndexCU5        := null;
        recidy. IndexLP6        := null;
            end;

-- Item Type 1 Process
                if      recid. ItemType = '1' then

-- Insert Index List Table(PR1)
                p_InsIndexList(inSeq, '3', '1', inYear, inMonth, recid. ISIC, recid. Index
CommodityCode,
                        recams. IndustryTypeShortName, reccm. CommodityShortName,
                        recid. IndexPR1, recidm. IndexPR1, recidy. IndexPR1, recgrm. GR_P
R1, recgry. GR_PR1,
                        reccim. CD_PR1, recciy. CD_PR1, reccim. CR_PR1, recciy. CR_PR1,
                                recids. IndexPR1, recidms. IndexPR1, recidys. IndexPR1,
recgrms. GR_PR1, recgrys. GR_PR1,
                                reccims. CD_PR1, recciys. CD_PR1, reccims. CR_PR1, recci
ys. CR_PR1);

-- Insert Index List Table(SP2)
                p_InsIndexList(inSeq, '3', '2', inYear, inMonth, recid. ISIC, recid. Index
CommodityCode,
                        recams. IndustryTypeShortName, reccm. CommodityShortName,
                        recid. IndexSP2, recidm. IndexSP2, recidy. IndexSP2, recgrm. GR_S
P2, recgry. GR_SP2,
                        reccim. CD_SP2, recciy. CD_SP2, reccim. CR_SP2, recciy. CR_SP2,
                                recids. IndexSP2, recidms. IndexSP2, recidys. IndexSP2,
recgrms. GR_SP2, recgrys. GR_SP2,
                                reccims. CD_SP2, recciys. CD_SP2, reccims. CR_SP2, recci
ys. CR_SP2);

-- Insert Index List Table(IV3)
                p_InsIndexList(inSeq, '3', '3', inYear, inMonth, recid. ISIC, recid. Index
CommodityCode,
                        recams. IndustryTypeShortName, reccm. CommodityShortName,
                        recid. IndexIV3, recidm. IndexIV3, recidy. IndexIV3, recgrm. GR_I
V3, recgry. GR_IV3,
                        reccim. CD_IV3, recciy. CD_IV3, reccim. CR_IV3, recciy. CR_IV3,
                                recids. IndexIV3, recidms. IndexIV3, recidys. IndexIV3,
recgrms. GR_IV3, recgrys. GR_IV3,
                                reccims. CD_IV3, recciys. CD_IV3, reccims. CR_IV3, recci
ys. CR_IV3);
```

```
-- Insert Index List Table(IR4)
                        p_InsIndexList(inSeq, '3', '4', inYear, inMonth, recid. ISIC, recid. Index
CommodityCode,
                                recams. IndustryTypeShortName, reccm. CommodityShortName,
                                recid. IndexIR4, recidm. IndexIR4, recidy. IndexIR4, recgrm. GR_I
R4, recgry. GR_IR4,
                                reccim. CD_IR4, recciy. CD_IR4, reccim. CR_IR4, recciy. CR_IR4,
                                        recids. IndexIR4, recidms. IndexIR4, recidys. IndexIR4,
recgrms. GR_IR4, recgrys. GR_IR4,
                                        reccims. CD_IR4, recciys. CD_IR4, reccims. CR_IR4, recci
ys. CR_IR4);

-- Item Type 6 Process
                        elsif   recid. ItemType = '6' then

-- Insert Index List Table(CU5)
                        p_InsIndexList(inSeq, '3', '5', inYear, inMonth, recid. ISIC, recid. Index
CommodityCode,
                                recams. IndustryTypeShortName, reccm. CommodityShortName,
                                recid. IndexCU5, recidm. IndexCU5, recidy. IndexCU5, recgrm. GR_C
U5, recgry. GR_CU5,
                                reccim. CD_CU5, recciy. CD_CU5, reccim. CR_CU5, recciy. CR_CU5,
                                        recids. IndexCU5, recidms. IndexCU5, recidys. IndexCU5,
recgrms. GR_CU5, recgrys. GR_CU5,
                                        reccims. CD_CU5, recciys. CD_CU5, reccims. CR_CU5, recci
ys. CR_CU5);
                end if;
                end if;
-- Fetch Indices Table
                fetch   csrid
                        into    recid. ISIC, recid. ItemType, recid. IndexCommodityCode,
                recid. IndexPR1, recid. IndexSP2, recid. IndexIV3,
                                        recid. IndexIR4, recid. IndexCU5, recid. IndexLP6,
                                        recgrm. GR_PR1, recgrm. GR_SP2, recgrm. GR_IV3,
                                        recgrm. GR_IR4, recgrm. GR_CU5, recgrm. GR_LP6,
                                        recgry. GR_PR1, recgry. GR_SP2, recgry. GR_IV3,
                                        recgry. GR_IR4, recgry. GR_CU5, recgry. GR_LP6,
                                        reccim. CD_PR1, reccim. CD_SP2, reccim. CD_IV3,
                                        reccim. CD_IR4, reccim. CD_CU5, reccim. CD_LP6,
                                        reccim. CR_PR1, reccim. CR_SP2, reccim. CR_IV3,
                                        reccim. CR_IR4, reccim. CR_CU5, reccim. CR_LP6,
                                        recciy. CD_PR1, recciy. CD_SP2, recciy. CD_IV3,
                                        recciy. CD_IR4, recciy. CD_CU5, recciy. CD_LP6,
                                        recciy. CR_PR1, recciy. CR_SP2, recciy. CR_IV3,
                                        recciy. CR_IR4, recciy. CR_CU5, recciy. CR_LP6, reccm. Co
mmodityShortName;

        end loop;

        close csrid;

end;

end IndexListMk;
/
show errors
/
--commit;
```

```
/*
||==================================================================
||Package Name InputFunction
||      Create Date 1999/09/01
||      Update Date 1999/11/02
||      K. SHIBAMOTO @ JICA STUDY TEAM
||------------------------------------------------------------------
|| Functions and Procedures For Survey Data Input
|| 1. Caluculation of Response Rate for each survey scope
|| 2. Create Collection rate list in temporary table
|| @e:¥userjica¥storedproc¥inputfunction
||------------------------------------------------------------------
*/

CREATE OR REPLACE PACKAGE InputFunction IS
-- ---------------------------------------------------
-- Caluculation of Response Rate for each survey scope
-- ---------------------------------------------------
    PROCEDURE  CalcResponseRate(
       pISIC                  IN  VARCHAR2,  -- ISIC
       pYear                  IN  NUMBER,    -- Year
       pMonth                 IN  NUMBER,    -- Month
       pScope                 IN  VARCHAR2,  -- Target Scope For Calculation
       pContinuousRespondent  IN  VARCHAR2,  -- ContinuousRespondent
       pPopulationNumber      OUT NUMBER,    -- Number of Target Establishments
       pCollectedNumber       OUT NUMBER,    -- Number of Establishments that answered the q
uestionaire
       PCollectionRate        OUT NUMBER     -- (pCollectedNumber / pPopulationNumber) * 100
    ) ;

-- ------------
-- ---------------------------------------------------
-- Create Collection rate list in temporary table
-- ---------------------------------------------------
    PROCEDURE  CreateResponseRate(
       pSeq                   IN  SMALLINT,  -- Seq
       pISIC                  IN  VARCHAR2,  -- ISIC
       pYear                  IN  NUMBER,    -- Year
       pMonth                 IN  NUMBER,    -- Month
       pScope                 IN  VARCHAR2,  -- Target Scope For Calculation
       pContinuousRespondent  IN  VARCHAR2,  -- ContinuousRespondent
       pReturn                OUT NUMBER     -- 0: OK , 1:Error
    ) ;

-- ------------
END;
/
show errors

/*
||==================================================================
||Package Name: InputFunction
||      Create Date 1999/09/01
||      Update Date 1999/11/03
||      K. SHIBAMOTO @JICA STUDY TEAM
||------------------------------------------------------------------
|| Functions and Procedures For Survey Data Input
|| 1. Caluculation of Response Rate for each survey scope
|| 2. Create Collection rate list in temporary table
 TEST - 1 -
 variable pC number;
 variable pP number;
 variable pR number;
 exec InputFunction. CalcResponseRate('151201', 1999, 2, '2', '0', :pP, :pC, :pR);
 print pP;
 print pC;
 print pCol;
 TEST - 2 -
 variable Num Number;
 exec InputFunction. CreateResponseRate(10, '151201', 1999, 2, 1, 1, :Num);
||------------------------------------------------------------------
*/
CREATE OR REPLACE PACKAGE BODY InputFunction IS
-- ---------------------------------------------------
-- Caluculation of Response Rate for each survey scope
-- ---------------------------------------------------
    PROCEDURE  CalcResponseRate(
```

```
        pISIC                   IN  VARCHAR2,  -- ISIC
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
        pContinuousRespondent IN  VARCHAR2,  -- ContinuousRespondent(0 : All , 1 : Continuo
us Only)
        pPopulationNumber       OUT NUMBER,    -- Number of Target Establishments
        pCollectedNumber        OUT NUMBER,    -- Number of Establishments that answered the
questionaire
        PCollectionRate         OUT NUMBER     -- (pCollectedNumber / pPopulationNumber) * 100
    )
    --return number

    IS
    -- Definition of Variables --
        nPopulation     NUMBER;
        nCollected      NUMBER;
        nCollectionRate NUMBER;
        --sSurveyScope    VARCHAR2(1);

    BEGIN
        -- Get number of target establishments
        /* ContinuousRespondent
         ||  0 -> 0 or 1 (All)
         ||  1 -> 1       (Only Continuous Answering Establishment)
        */
        select count(*) into nPopulation
        from EstablishmentMaster
        where ISIC = pISIC
        and SurveyScope <= pScope
        and ContinuousRespondent >= pContinuousRespondent;
--DBMS_OUTPUT.PUT_LINE('  nPopulation --> ' ||to_char(nPopulation) );

        -- Get number of collected establishments
        select count(*) into nCollected
        from EstablishmentMaster EM, SurveyHeader SH
        where EM.ISIC = pISIC
          and EM.SurveyScope <= pScope
          and EM.ContinuousRespondent >= pContinuousRespondent
          and EM.ISIC = SH.ISIC
          and EM.RegistrationNo = SH.RegistrationNo
          and SH.Year = pYear
          and SH.Month = pMonth
          and SH.EstimatedMark IN (0,1); -- ManualInput and Modified Estimated Questionnai
re

--DBMS_OUTPUT.PUT_LINE('  nCollected --> ' ||to_char(nCollected) );
        -- Caluculate Collected Rate
        IF nPopulation != 0 THEN
            nCollectionRate := ROUND((nCollected / nPopulation)* 100,2) ;
        ELSE
            nCollectionRate := 0;
        END IF;
--DBMS_OUTPUT.PUT_LINE('  nCollectionRate --> ' ||to_char(nCollectionRate) );
        -- Set Output Arguments
        pPopulationNumber := nPopulation;
        pCollectedNumber := nCollected;
        PCollectionRate := nCollectionRate;

    EXCEPTION
        WHEN OTHERS THEN
            pPopulationNumber := 0;
            pCollectedNumber := 0;
            PCollectionRate := 0;
            --error_code := SQLCODE;
            --error_msg  := SQLERRM;
    END;

-- ------------------------------------------------
-- Create Collection rate list in temporary table
-- ------------------------------------------------
    PROCEDURE  CreateResponseRate(
        pSeq                    IN  SMALLINT,  -- Seq
        pISIC                   IN  VARCHAR2,  -- ISIC
        pYear                   IN  NUMBER,    -- Year
        pMonth                  IN  NUMBER,    -- Month
        pScope                  IN  VARCHAR2,  -- Target Scope For Calculation
```

```
        pContinuousRespondent IN  VARCHAR2, -- ContinuousRespondent(0 : All , 1 : Continuous
Only)
        pReturn               OUT NUMBER    -- 0: OK , 1:Error
    )
    IS
    BEGIN
        delete from CollectionRate where Seq = pSeq;

        insert into CollectionRate
        select pSeq,              --Seq
               pISIC,             --ISIC
               E.RegistrationNo,    --RegistrationNo
               E.EstablishmentName, --EstablishmentName
               E.EstablishmentTel,  --Tel
               E.EstablishmentFax,  --Fax
               NameOfRespondent,  --Respondent
               Null,              --Year
               Null               --Month
        from EstablishmentMaster E
        where E.ISIC = pISIC
          and E.SurveyScope <= pScope
          and E.ContinuousRespondent >= pContinuousRespondent;
        commit work;

        /* ContinuousRespondent
         ||  0 -> 0 or 1 (All)
         ||  1 -> 1      (Only Continuous Answering Establishment)
        */
        update CollectionRate
           set Year  = pYear,
               Month = pMonth
        where  Seq = pSeq
          and  RegistrationNo IN
               (select E.RegistrationNo
                  from SurveyHeader S, EstablishmentMaster E
                  where S.ISIC = pISIC
                    and S.Year = pYear
                    and S.Month = PMonth
                    and S.ISIC = E.ISIC
                    and S.RegistrationNo = E.RegistrationNo
                    and S.EstimatedMark IN (0, 1)
                    and E.SurveyScope <= pScope
                    and E.ContinuousRespondent >= pContinuousRespondent
               );
        commit work;
        pReturn := 0;

    EXCEPTION
        WHEN OTHERS THEN
            pReturn := 1;

    END;

END;
/
show errors
```

```
-- ==================================================
-- Package Name : Monthly Process
-- Package ID   : MonthProc
-- Create Date  : 1/Dec/1999
-- Replace Date : xx/xxx/xxxx
-- Author               : Nakamura
--
-- test         : @c:¥jica¥MonthProc
-- test                                :       begin MonthProc.p_PastRecMk('1',1998,5); e
nd;
-- test                                :       begin MonthProc.p_LinkCopy(1999,3); commit;
end;
-- test                                :       begin MonthProc.p_BatchCtlUpd('2',1998,12);
commit; end;
-- ==================================================
create or replace package MonthProc is
procedure p_PastRecMk (
        inPre_Rev_Mark  in      Varchar2,
        inYear                                  in      Number,
        inMonth                                 in      Number);

procedure p_LinkCopy (
        inYear                                  in      Number,
        inMonth                                 in      Number);

procedure p_BatchCtlUpd (
        inPre_Rev_Mark  in      Varchar2,
        inYear                                  in      Number,
        inMonth                                 in      Number);

procedure p_DeleteAllTemps;

end;
/
show errors
/

create or replace package body MonthProc is
-------------------------------------------------------
-- Past Record Data Making
--      inPre_Rev_Mark :
--              1 = Preliminary Report Process
--              2 = Revised Report Process
-------------------------------------------------------
procedure p_PastRecMk (
        inPre_Rev_Mark  in      Varchar2,
        inYear                                  in      Number,
        inMonth                                 in      Number) is

-- Record Area Define
        recsd                           SurveyData%rowtype;
        recbc                           BatchControl%rowtype;
        recpr                           PastRecord%rowtype;

Begin
-- DBMS_OUTPUT.PUT_LINE('Start PGM');
-- Pre_Rev_Mark Checking
        if      inPre_Rev_Mark = '1' or inPre_Rev_Mark = '2' then
                null;
        else
                return;
        end if;

-- Past Record Delete
        delete  from    PastRecord
                where   Year                                    = inYear
                 and    Month                                   = inMonth
                 and    Pre_Rev_Mark    = inPre_Rev_Mark;

        commit;

-- Past Record Insert
        insert
                into    PastRecord(
                                        ISIC,
                                        YEAR,
                                        MONTH,
```

```
                    REGISTRATIONNO,
                    ITEMTYPE,
                    COMMODITYCODE,
                    PRE_REV_MARK,
                    BM_INVENTORY,
                    ATRBT_BM_INVENTORY,
                    PRODUCTIONQTY,
                    ATRBT_PRODUCTIONQTY,
                    RECEIPTS,
                    ATRBT_RECEIPTS,
                    DOMESTICSALES,
                    ATRBT_DOMESTICSALES,
                    EXPORT,
                    ATRBT_EXPORT,
                    OTHERSALES,
                    ATRBT_OTHERSALES,
                    ME_INVENTORY,
                    ATRBT_ME_INVENTORY,
                    CAPACITY,
                    ATRBT_CAPACITY,
                    SHIPMENTVALUE,
                    ATRBT_SHIPMENTVALUE,
                    SALESPLAN,
                    ATRBT_SALESPLAN,
                    LABORTOTAL,
                    ATRBT_LABORTOTAL,
                    LABOR_SC,
                    ATRBT_LABOR_SC,
                    RWMT_MEINVENTORY,
                    ATRBT_RWMT_MEINVENTORY,
                    RWMT_MEINVENTORYValue,
                    ATRBT_RWMT_MEINVENTORYValue,
                    BIZCONDITION,
                    UNITPRICE,
                    REMARKS,
                    ESTIMATEDMARK,
                    USERID,
                    RECDATE)
select  ISIC,
            YEAR,
            MONTH,
            REGISTRATIONNO,
            ITEMTYPE,
            COMMODITYCODE,
            inPRE_REV_MARK,
            BM_INVENTORY,
            ATRBT_BM_INVENTORY,
            PRODUCTIONQTY,
            ATRBT_PRODUCTIONQTY,
            RECEIPTS,
            ATRBT_RECEIPTS,
            DOMESTICSALES,
            ATRBT_DOMESTICSALES,
            EXPORT,
            ATRBT_EXPORT,
            OTHERSALES,
            ATRBT_OTHERSALES,
            ME_INVENTORY,
            ATRBT_ME_INVENTORY,
            CAPACITY,
            ATRBT_CAPACITY,
            SHIPMENTVALUE,
            ATRBT_SHIPMENTVALUE,
            SALESPLAN,
            ATRBT_SALESPLAN,
            LABORTOTAL,
            ATRBT_LABORTOTAL,
            LABOR_SC,
            ATRBT_LABOR_SC,
            RWMT_MEINVENTORY,
            ATRBT_RWMT_MEINVENTORY,
            RWMT_MEINVENTORYValue,
            ATRBT_RWMT_MEINVENTORYValue,
            BIZCONDITION,
            UNITPRICE,
            REMARKS,
            '0',
```

```
                                USERID,
                                sysdate
        from    SurveyData
        where   Year                    = inYear
        and             Month                   = inMonth;

end;

-----------------------------------------------------------
-- Link Master Copy To Next Month
-----------------------------------------------------------
procedure p_LinkCopy (
        inYear                                  in      Number,
        inMonth                                 in      Number) is

-- Work Area
        nNY                     number;
        nNM                     number;

Begin
-- DBMS_OUTPUT.PUT_LINE('Start PGM');
-- Get Pre Month
        if      inMonth = 12 then
                nNY     := inYear + 1;
                nNM     := 1;
        else
                nNY     := inYear;
                nNM     := inMonth + 1;
        end if;

-- Link Master Insert
        begin
                insert
                        into    LinkMaster(
                                                ISIC,
                                                ITEMTYPE,
                                                COMMODITYCODE,
                                                YEAR,
                                                MONTH,
                                                SURVEYSCOPE,
                                                Version,
                                                PRODUCTIONQTY,
                                                SHIPMENTQTY,
                                                ME_INVENTORY,
                                                CAPACITY,
                                                SHIPMENTVALUE,
                                                WORKER,
                                                WORKINGHOUR,
                                                WORKINGDAY,
                                                RWMT_MEINVENTORY)
                select  ISIC,
                                                ITEMTYPE,
                                                COMMODITYCODE,
                                                nNY,
                                                nNM,
                                                SURVEYSCOPE,
                                                Version,
                                                PRODUCTIONQTY,
                                                SHIPMENTQTY,
                                                ME_INVENTORY,
                                                CAPACITY,
                                                SHIPMENTVALUE,
                                                WORKER,
                                                WORKINGHOUR,
                                                WORKINGDAY,
                                                RWMT_MEINVENTORY
                from    LinkMaster
                where   Year    = inYear
                and             Month   = inMonth;
        exception
                when    dup_val_on_index then
                        null;
        end;

-- Deflator Master Insert
        begin
                insert
```

```
                    into    DeflatorMaster(
                                        ISIC,
                                        ITEMTYPE,
                                        COMMODITYCODE,
                                        YEAR,
                                        MONTH,
                                        SURVEYSCOPE,
                                        Version,
                                        PRODUCTIONQTY,
                                        SHIPMENTQTY,
                                        ME_INVENTORY,
                                        CAPACITY,
                                        RWMT_MEINVENTORY)
            select  ISIC,
                                        ITEMTYPE,
                                        COMMODITYCODE,
                                        nNY,
                                        nNM,
                                        SURVEYSCOPE,
                                        Version,
                                        PRODUCTIONQTY,
                                        SHIPMENTQTY,
                                        ME_INVENTORY,
                                        CAPACITY,
                                        RWMT_MEINVENTORY
            from    DeflatorMaster
            where   Year    = inYear
            and             Month   = inMonth;
        exception
            when    dup_val_on_index then
                    null;
        end;

-- Seasonal Adjustment Master Insert
        begin
            insert
                    into    SeasonalAdjustmentMaster(
                                        AGGREGATIONRANGE,
                                        ISIC,
                                        ITEMTYPE,
                                        INDEXCOMMODITYCODE,
                                        YEAR,
                                        MONTH,
                                        SURVEYSCOPE,
                                        VERSION,
                                        INDEXPR1,
                                        INDEXSP2,
                                        INDEXIV3,
                                        INDEXIR4,
                                        INDEXCU5,
                                        INDEXLP6,
                                        INDEXLI7,
                                        INDEXRI8)
            select  AGGREGATIONRANGE,
                                        ISIC,
                                        ITEMTYPE,
                                        INDEXCOMMODITYCODE,
                                        nNY,
                                        nNM,
                                        SURVEYSCOPE,
                                        VERSION,
                                        INDEXPR1,
                                        INDEXSP2,
                                        INDEXIV3,
                                        INDEXIR4,
                                        INDEXCU5,
                                        INDEXLP6,
                                        INDEXLI7,
                                        INDEXRI8
            from    SeasonalAdjustmentMaster
            where   Year    = inYear
            and             Month   = inMonth;
        exception
            when    dup_val_on_index then
                    null;
        end;
```

```
end;

-------------------------------------------------------
-- Batch Control File Update
-------------------------------------------------------
procedure p_BatchCtlUpd (
        inPre_Rev_Mark  in      Varchar2,
        inYear                                  in      Number,
        inMonth                                 in      Number) is

Begin
        if      inPre_Rev_Mark = '1' then
                update  BatchControl
                        set             PreliminaryStatus = '0',
                                                PreliminaryDate         = sysdate
                        where   Year    = inYear
                        and             Month   = inMonth;
        elsif   inPre_Rev_Mark = '2' then
                update  BatchControl
                        set             RevisedStatus = '0',
                                                RevisedDate     = sysdate
                        where   Year    = inYear
                        and             Month   = inMonth;
        end if;

end;

/* ============================================================
    Delete All temporary tables
============================================================  */
    PROCEDURE p_DeleteAllTemps

    IS
    BEGIN
        delete from ContributionForAll1;
        delete from ContributionForAll2;
        delete from ContributionForIndustry;
        delete from ContributionForCommodity;
        delete from IndexList;
        delete from TimeFigures;
        delete from CollectionRate;
        delete from AnnualcollectionStatus;
        delete from CPMYRawEst;
        delete from CPMYRawISIC;
    END;

end;
/
show errors
/
```

```
-- ==================================================
-- Package Name : Survey Data Control
-- Package ID   : sdctl
-- Create Date  : 17/Sep/1999
-- Replace Date : xx/xxx/xxxx
-- Author       : Nakamura
--
-- File         : @c:¥jica¥sdctl
-- Test         : select * from surveyheader
-- Test         : select * from batchcontrol
-- ==================================================
create or replace package sdctl is
procedure       p_EntryCk (
        inYear                                          in      Number,
        inMonth                                         in      Number,
        inISIC                                          in      Varchar2,
        inRegistrationNo        in      Varchar2,
        outUserInputed          out Varchar2);

procedure       p_HeaderMk (
        inYear                                          in      Number,
        inMonth                                         in      Number,
        inISIC                                          in      Varchar2,
  inRegistrationNo      in      Varchar2);

procedure       p_SurveyHead (
        inYear                                          in      Number,
        inMonth                                         in      Number,
        inISIC                                          in      Varchar2,
  inRegistrationNo      in      Varchar2,
        outUserInputed          out     Varchar2);

end     sdctl;
/
show errors
/

create or replace package body sdctl is
-------------------------------------------------------
-- None Input Checking
--  outUserInputed:
--   User Data Inputed   = 1
--   User Data Not Input = 0
-------------------------------------------------------
procedure p_EntryCk (
        inYear                                          in      Number,
        inMonth                                         in      Number,
        inISIC                                          in      Varchar2,
        inRegistrationNo        in      Varchar2,
        outUserInputed          out Varchar2) is

-- Record Area Define
        recsd   SurveyData%rowtype;

-- Work Area Define
        nRecordCount                    number := 0;
        nNoneInputed            number := 0;

-- Survey Data Cursor
        Cursor  csrSurveyData is
        select  *
                from    SurveyData
                        where ISIC      = inISIC
                and     RegistrationNo = inRegistrationNo
                and     Year    = inYear
                and     Month           = inMonth;

Begin
-- DBMS_OUTPUT.PUT_LINE('Start PGM');
-- Get Record Count
        open csrSurveyData;

        fetch csrSurveyData into recsd;

        while csrSurveyData%found loop
-- Reccord Counter Up
        nRecordCount := nRecordCount + 1;
```

```
-- Data Input Checking
                if      recsd.PRODUCTIONQTY                                    is null
                and recsd.RECEIPTS                                     is null
                and recsd.DOMESTICSALES                                is null
                and recsd.EXPORT                                  is null
                and recsd.OTHERSALES                              is null
                and recsd.ME_INVENTORY                            is null
                and recsd.CAPACITY                                is null
                and recsd.SHIPMENTVALUE                           is null
                and recsd.SALESPLAN                               is null
                and recsd.LABORTOTAL                              is null
                and recsd.LABOR_SC                                is null
                and recsd.RWMT_MEINVENTORY                        is null
                and recsd.RWMT_MEINVENTORYValue is null then
                nNoneInputed := nNoneInputed + 1;
        end if;

        fetch csrSurveyData into recsd;

        end loop;

        close csrSurveyData;

-- Data input checking
        if      nRecordCount = nNoneInputed then
                outUserInputed := '0';
        else
        outUserInputed := '1';
        end if;

end;

-------------------------------------------------------------
-- Survey Header Data Maintenance
-------------------------------------------------------------
procedure p_HeaderMk (
        inYear                                          In Number,
        inMonth                                         In Number,
        inISIC                                          In Varchar2,
  inRegistrationNo      In Varchar2) is

-- Record Area Define
        recsd           SurveyData%rowtype;
        recsh           SurveyHeader%rowtype;
        recbc           BatchControl%rowtype;

-- Work Area Define
        nRecordCount                    number := 0;
        nAllEstimated                   number := 0;
        nIncludeEstimated       number := 0;
        sRevisedStatus          Varchar2(1) := '0';
        sYearlyStatus                   Varchar2(1) := '0';

-- Survey Data Cursor
        Cursor  csrSurveyData is
        select  *
                from    SurveyData
                where   ISIC  = inISIC
                and     RegistrationNo = inRegistrationNo
                and     Year  = inYear
                and     Month   = inMonth;

Begin
-- DBMS_OUTPUT.PUT_LINE('Start PGM');
-- Batch Control Table Selecting
        begin
        Select  *
                into    recbc
                from    BatchControl
                where Year = inYear
        and     Month = inMonth;
        exception
        when no_data_found then
                recbc.RevisedDate  := null;
                recbc.RevisedStatus := '0';
                recbc.YearlyDate  := null;
```

```
                recbc.YearlyStatus  := '0';
        end;

-- Survey Data Open -> Fetch
        open csrSurveyData;
        fetch csrSurveyData into recsd;
        while csrSurveyData%found loop

-- Reccord Counter Up
        nRecordCount := nRecordCount + 1;

-- Data All Estamated Checking
                if recsd.ItemType = '1' then
                        if      (recsd.Atrbt_BM_INVENTORY       = 'X'   or recsd.Atrbt_BM_I
NVENTORY        = 'Y')
                and (recsd.Atrbt_PRODUCTIONQTY  = 'X' or recsd.Atrbt_PRODUCTIONQTY      = '
Y')
                and (recsd.Atrbt_RECEIPTS                       = 'X' or recsd.Atrbt_RECEI
PTS                     = 'Y')
                and (recsd.Atrbt_DOMESTICSALES  = 'X' or recsd.Atrbt_DOMESTICSALES  = 'Y')
                and (recsd.Atrbt_EXPORT                                         = 'X' or recsd.Atrb
t_EXPORT                                = 'Y')
                and (recsd.Atrbt_OTHERSALES             = 'X' or recsd.Atrbt_EXPORT
                                = 'Y')
                and (recsd.Atrbt_ME_INVENTORY   = 'X' or recsd.Atrbt_ME_INVENTORY       = '
Y') then
                nAllEstimated := nAllEstimated + 1;
                end if;
                elsif recsd.ItemType = '6' then
                        if      (recsd.Atrbt_CAPACITY = 'X' or recsd.Atrbt_CAPACITY = 'Y')
then
                nAllEstimated := nAllEstimated + 1;
                end if;
        elsif recsd.ItemType = '2' then
                if      (recsd.Atrbt_SHIPMENTVALUE      = 'X' or recsd.Atrbt_SHIPMENTVALUE
        = 'Y')
                and (recsd.Atrbt_SALESPLAN                      = 'X' or recsd.Atrbt_SALES
PLAN                    = 'Y') then
                        nAllEstimated := nAllEstimated + 1;
                end if;
        elsif recsd.ItemType = '3' then
                        if      (recsd.Atrbt_LABORTOTAL = 'X' or        recsd.Atrbt_LABORTO
TAL     = 'Y')
                and (recsd.Atrbt_LABOR_SC       = 'X'   or      recsd.Atrbt_LABOR_SC
                = 'Y') then
                nAllEstimated := nAllEstimated + 1;
                end if;
        elsif recsd.ItemType = '4' then
                if      (recsd.Atrbt_RWMT_MEINVENTORY                   = 'X'   or      rec
sd.Atrbt_RWMT_MEINVENTORY                       = 'Y')
                and (recsd.Atrbt_RWMT_MEINVENTORYValue  = 'X' or        recsd.Atrbt_RWMT_ME
INVENTORYValue = 'Y') then
                nAllEstimated := nAllEstimated + 1;
                end if;
        end if;

-- Data Include Estamated Checking
                if  recsd.ItemType = '1' then
                if      (recsd.Atrbt_BM_INVENTORY       = 'X' or        recsd.Atrbt_BM_INVE
NTORY   = 'Y')
                or      (recsd.Atrbt_PRODUCTIONQTY      = 'X' or        recsd.Atrbt_PRODUCT
IONQTY = 'Y')
                or      (recsd.Atrbt_RECEIPTS                   = 'X' or        recsd.Atrbt
_RECEIPTS                       = 'Y')
                or      (recsd.Atrbt_DOMESTICSALES  = 'X' or    recsd.Atrbt_DOMESTICSALES =
'Y')
                or      (recsd.Atrbt_EXPORT                                     = 'X' or recsd.Atrb
t_EXPORT                                = 'Y')
                or      (recsd.Atrbt_OTHERSALES                 = 'X' or recsd.Atrbt_EXPORT
                                = 'Y')
                or      (recsd.Atrbt_ME_INVENTORY       = 'X' or recsd.Atrbt_ME_INVENTORY
        = 'Y') then
                nIncludeEstimated := nIncludeEstimated + 1;
                end if;
        elsif recsd.ItemType = '6' then
                if (recsd.Atrbt_CAPACITY = 'X' or recsd.Atrbt_CAPACITY = 'Y') then
        nIncludeEstimated := nIncludeEstimated + 1;
```

```
                end if;
        elsif recsd.ItemType = '2' then
                if (recsd.Atrbt_SHIPMENTVALUE = 'X' or recsd.Atrbt_SHIPMENTVALUE        = '
Y')
                or (recsd.Atrbt_SALESPLAN               = 'X' or recsd.Atrbt_SALESPLAN
                        = 'Y') then
                nIncludeEstimated := nIncludeEstimated + 1;
                end if;
        elsif   recsd.ItemType = '3' then
                if (recsd.Atrbt_LABORTOTAL      = 'X' or recsd.Atrbt_LABORTOTAL = 'Y')
                        or (recsd.Atrbt_LABOR_SC                = 'X' or recsd.Atrbt_LABOR
_SC     = 'Y') then
        nIncludeEstimated := nIncludeEstimated + 1;
                end if;
        elsif recsd.ItemType = '4' then
                if      (recsd.Atrbt_RWMT_MEINVENTORY = 'X' or recsd.Atrbt_RWMT_MEINVENTORY
= 'Y')
                or      (recsd.Atrbt_RWMT_MEINVENTORYValue = 'X' or recsd.Atrbt_RWMT_MEINV
ENTORYValue = 'Y') then
                        nIncludeEstimated := nIncludeEstimated + 1;
                end if;
        end if;

        fetch csrSurveyData into recsd;
        end loop;

        close csrSurveyData;

-- Header Data Insert or Update
        if nRecordCount = nAllEstimated then
        recsh.EstimatedMark := '2';
        else
        if  nIncludeEstimated > 0 then
                recsh.EstimatedMark := '1';
        else
                recsh.EstimatedMark := '0';
        end if;
        end if;

        if recbc.RevisedDate is null then
        recsh.RevisedStatus := '0';
        else
                recsh.RevisedStatus := '1';
        end if;

        if recbc.YearlyDate is null then
        recsh.YearlyStatus := '0';
  else
        recsh.YearlyStatus := '1';
        end if;

        begin
        Insert
                into            SurveyHeader(
                                        ISIC,
                                        RegistrationNo,
                                        Year,
                                        Month,
                                        EstimatedMark,
                                        RevisedStatus,
                                        YearlyStatus)
                Values (inISIC,
                                        inRegistrationNo,
                                        inYear,
                                        inMonth,
                                        recsh.EstimatedMark,
                                        recsh.RevisedStatus,
                                                        recsh.YearlyStatus);
        exception
        when dup_val_on_index then
                update  SurveyHeader
                set     EstimatedMark = recsh.EstimatedMark,
                                        RevisedStatus = recsh.RevisedStatus,
                                        YearlyStatus  = recsh.YearlyStatus
                        where ISIC = inISIC
                        and     RegistrationNo = inRegistrationNo
                        and     Year  = inYear
```

```
                and     Month = inMonth;
        end;

-- Batch Control Update
        if  recbc.RevisedDate is null then
        null;
        else
        if recbc.RevisedStatus = '1' then
                null;
        else
                sRevisedStatus := '1';
        end if;
        end if;

        if  recbc.YearlyDate is null then
        null;
        else
        if recbc.YearlyStatus = '1' then
                null;
        else
                sYearlyStatus := '1';
        end if;
        end if;

        if sRevisedStatus = '1'
        or sYearlyStatus = '1' then
        update  BatchControl
                set     RevisedStatus  = sRevisedStatus,
                                YearlyStatus   = sYearlyStatus
                where Year  = inYear
                and     Month   = inMonth;
        end if;
end;

-------------------------------------------------------------
-- Main Survey Data Entry Check And Survey Header Making
-------------------------------------------------------------
procedure p_SurveyHead (
        inYear                                          in      Number,
        inMonth                                         in      Number,
        inISIC                                          in      Varchar2,
        inRegistrationNo        in      Varchar2,
        outUserInputed          out Varchar2) is

        sUserInputed                    Varchar2(1) := '0';

Begin
-- Entry Check
        p_entryck (inYear, inMonth, inISIC, inRegistrationNo, sUserInputed);

        outUserInputed := sUserInputed;

        if outUserInputed = '0' then
                return;
        end if;

-- Header Data Making
        p_HeaderMk(inYear, inMonth, inISIC, inRegistrationNo);

end;

end;
/
show errors
/
```

```
-- ==================================================
-- Package Name : Time Series List of Figures by Commodity Data Making
-- Package ID   : temprep
-- Create Date  : 20/Oct/1999
-- Replace Date : xx/xxx/xxxx
-- Author       : Nakamura
--
-- File         : @c:¥jica¥timeseriesmk;
-- Test                                : BEGIN TimeSeriesMk. p_FigCommMk(1, 2000, '4', '15121
0', '2', '2'); END;
-- Test                                : Delete from TimeFigures;
-- ==================================================
create or replace package TimeSeriesMk is
procedure p_FigCommMk (
        inSeq                                                   in      number,
        inYear                                          in      number,
        inAggregationRange      in      varchar2,
        inISIC                                          in      varchar2,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2);
end TimeSeriesMk;
/
show errors
/

create or replace package body TimeSeriesMk is
-----------------------------------------------------------
-- Insert Temporary Figures Commodity Table
-----------------------------------------------------------
procedure p_InsFigComm (
        inSeq                                                                   i
n       NUMBER,
        inFigureType                                    in      VARCHAR2,
        inReportType                                    in      VARCHAR2,
        inYear                  in      NUMBER,
        inISIC                                                          in      VAR
CHAR2,
        inCommodityCode                         in      VARCHAR2,
        inIndustryTypeShortName in      VARCHAR2,
        inCommodityShortName            in      VARCHAR2,
        inUnitName                                              in      VARCHAR2,
        inPYA                                                                   i
n       NUMBER,
        inPYT                                                                   i
n       NUMBER,
        inCYT                                                                   i
n       NUMBER,
        inCYM01                                                         in      NUM
BER,
        inCYM02                                                         in      NUM
BER,
        inCYM03                                                         in      NUM
BER,
        inCYM04                                                         in      NUM
BER,
        inCYM05                                                         in      NUM
BER,
        inCYM06                                                         in      NUM
BER,
        inCYM07                                                         in      NUM
BER,
        inCYM08                                                         in      NUM
BER,
        inCYM09                                                         in      NUM
BER,
        inCYM10                                                         in      NUM
BER,
        inCYM11                                                         in      NUM
BER,
        inCYM12                                                         in      NUM
BER,
        inPY2T                                                          in      NUM
BER,
        inPYM01                                                         in      NUM
BER,
        inPYM02                                                         in      NUM
BER,
```

```
        inPYM03                                                         in      NUM
BER,
        inPYM04                                                         in      NUM
BER,
        inPYM05                                                         in      NUM
BER,
        inPYM06                                                         in      NUM
BER,
        inPYM07                                                         in      NUM
BER,
        inPYM08                                                         in      NUM
BER,
        inPYM09                                                         in      NUM
BER,
        inPYM10                                                         in      NUM
BER,
        inPYM11                                                         in      NUM
BER,
        inPYM12                                                         in      NUM
BER) is
begin
        Insert
        into            TimeFigures (
                                        Seq,
                                        FigureType,
                ReportType,
                Year,
                ISIC,
                CommodityCode,
                IndustryTypeShortName,
                CommodityShortName,
                UnitName,
                PYA,
                PYT,
                CYT,
                CYM01,
                CYM02,
                CYM03,
                CYM04,
                CYM05,
                CYM06,
                CYM07,
                CYM08,
                CYM09,
                CYM10,
                CYM11,
                CYM12,
                                        PY2T,
                PYM01,
                PYM02,
                PYM03,
                PYM04,
                PYM05,
                PYM06,
                PYM07,
                PYM08,
                PYM09,
                PYM10,
                PYM11,
                PYM12)
        Values (inSeq,
                                        inFigureType,
                inReportType,
                inYear,
                inISIC,
                inCommodityCode,
                inIndustryTypeShortName,
                inCommodityShortName,
                inUnitName,
                inPYA,
                inPYT,
                inCYT,
                inCYM01,
                inCYM02,
                inCYM03,
                inCYM04,
                inCYM05,
```

```
                inCYM06,
                inCYM07,
                inCYM08,
                inCYM09,
                inCYM10,
                inCYM11,
                inCYM12,
                                        inPY2T,
                inPYM01,
                inPYM02,
                inPYM03,
                inPYM04,
                inPYM05,
                inPYM06,
                inPYM07,
                inPYM08,
                inPYM09,
                inPYM10,
                inPYM11,
                inPYM12);
end;

-------------------------------------------------------
-- Growth Rate Against Caluculation
-------------------------------------------------------
function f_CalGRA(
        inCurFigure     in number,
        inPreFigure in number) return number is

        nGRA                            number;

begin
        if      (inCurFigure is null)
        or      (inPreFigure is null) then
                        nGRA := null;
                        return nGRA;
        end if;

        if      inPreFigure = 0 then
                        nGRA := 100;
                        return nGRA;
        end if;

        nGRA := ((inCurFigure - inPreFigure) / inPreFigure) * 100;

        return nGRA;
end;

-------------------------------------------------------
-- Actual Value Caluculation.
-------------------------------------------------------
procedure p_ActualValue (
        inSeq                                                   in      number,
        inYear                                          in      number,
        inAggregationRange      in      varchar2,
        inISIC                                          in      varchar2,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2) is

-- Record Area Define
        reccm                                                   CommodityMaster%row
type;
        recum                                                   UnitMaster%rowtype;
        reccl                                                   ClassMaster%rowtype;
        recbc                                                   BatchControl%rowtyp
e;

        ProductionQty                           TimeFigures%rowtype;
        Shipment                                        TimeFigures%rowtype;
        Me_Inventory                            TimeFigures%rowtype;
        ShipmentValue                           TimeFigures%rowtype;
        Capacity                                        TimeFigures%rowtype;
        RwMt_MEInventory                TimeFigures%rowtype;

-- Work Area Define
        ProductionQtyCYM                TimeFigures.CYM01%type;
        ShipmentCYM                                     TimeFigures.CYM01%type;
```

```
        Me_InventoryCYM                 TimeFigures.CYM01%type;
        ShipmentValueCYM                TimeFigures.CYM01%type;
        CapacityCYM                                     TimeFigures.CYM01%type;
        RwMt_MEInventoryCYM     TimeFigures.CYM01%type;

        ProductionQtyPYM                TimeFigures.PYM01%type;
        ShipmentPYM                                     TimeFigures.PYM01%type;
        Me_InventoryPYM                 TimeFigures.PYM01%type;
        ShipmentValuePYM                TimeFigures.PYM01%type;
        CapacityPYM                                     TimeFigures.PYM01%type;
        RwMt_MEInventoryPYM     TimeFigures.PYM01%type;

        i                               number;
        nPY                     number;
        nPY2            number;

        sISIC2d         Varchar2(2);
        sISIC4d         Varchar2(4);

        nCurMonth       number;

-- Commodity Master Cursor
        Cursor csrcm is
                select  CommodityMaster.ISIC,
                                                CommodityMaster.ItemType,
                                                CommodityMaster.CommodityCode,
                                                CommodityMaster.CommodityShortName,
                                                ClassMaster.IndustryTypeShortName,
                                                UnitMaster.UnitName
    from                CommodityMaster, ClassMaster, UnitMaster
        where           CommodityMaster.ISIC = ClassMaster.ISIC
                and             CommodityMaster.UnitCode = UnitMaster.UnitCode
         order by CommodityMaster.ISIC, CommodityMaster.ItemType,
                                                CommodityMaster.CommodityCode;

Begin
        DBMS_OUTPUT.PUT_LINE('Start p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));
-- Initial Process
        sISIC2d := substr(inISIC,1,2);
        sISIC4d := substr(inISIC,1,4);
        nPY                     := inYear - 1;
        nPY2            := inYear - 2;

-- Current Month Get
        select  max(Month)
                into    recbc.Month
                from    BatchControl
                where   Year = inYear
                        and     RevisedDate is not null;

        if      recbc.Month is null then
                nCurMonth := 1;
        elsif   recbc.Month >= 12 then
                nCurMonth := 12;
        else
                nCurMonth := recbc.Month + 1;
        end if;

-- Commodity Master Open Fetch
        open csrcm;
        fetch csrcm into        reccm.ISIC, reccm.ItemType, reccm.CommodityCode,
                                reccm.CommodityShortName, reccl.IndustryTypeShortName, recu
m.UnitName;

-- Commodity Master Fetch Looping
        while csrcm%found loop
                if      (inAggregationRange = '1')
                or      (inAggregationRange = '2' and substr(reccm.ISIC,1,2) = sISIC2d)
                or      (inAggregationRange = '3' and substr(reccm.ISIC,1,4) = sISIC4d)
                or      (inAggregationRange = '4' and substr(reccm.ISIC,1,6) = inISIC) then

--      DBMS_OUTPUT.PUT_LINE(to_char(sysdate,'HH24:MI:SS') || ' ' || reccm.ISIC || ' ' ||
reccm.ItemType ||
--
                        ' ' || reccm.CommodityCode);

-- Item Type 1
```

```
                    if      reccm.ItemType = '1' then

-- Current Year Total Data Select
                            select  sum(ProductionQty),
                                                    sum(DomesticSales) + sum(E
xport),
                                                    sum(Me_Inventory)
                            into            ProductionQty.CYT,
                                                    Shipment.CYT,
                                                    Me_Inventory.CYT
                    from            VW_PastRecordScope
                            where           ISIC = reccm.ISIC
                            and             Year = inYear
                            and             CommodityCode = reccm.CommodityCode
                                    and             ItemType = reccm.ItemType
                                    and             SurveyScope <= inSurveyScope
                                    and             ((Month = nCurMonth and Pre_Rev_Mark
= inPre_Rev_Mark)
                                    or              (Month < nCurMonth      and     Pre
_Rev_Mark = '2'));

-- Previous Year Total Data Select
                            select  sum(ProductionQty),
                                                    sum(DomesticSales) + sum(E
xport),
                                                    sum(Me_Inventory)
                            into            ProductionQty.PYT,
                                                    Shipment.PYT,
                                                    Me_Inventory.PYT
                    from            VW_PastRecordScope
                            where           ISIC = reccm.ISIC
                            and             Year = nPY
                            and             CommodityCode = reccm.CommodityCode
                              and           ItemType = reccm.ItemType
                              and           Pre_Rev_Mark = '2'
                              and           SurveyScope <= inSurveyScope;

-- Check Output Data
                            if      ProductionQty.CYT       is null
                            and     Shipment.CYT                    is null
                            and     Me_Inventory.CYT        is null
                            and     ProductionQty.PYT       is null
                            and     Shipment.PYT                    is null
                            and     Me_Inventory.PYT        is null then
                                    null;
                            else
-- Month Data Select
                                    ProductionQty.CYM01 := null;
                                    Shipment.CYM01                  := null;
                                    Me_Inventory.CYM01      := null;
                                    ProductionQty.PYM01 := null;
                                    Shipment.PYM01                  := null;
                                    Me_Inventory.PYM01      := null;
                                    ProductionQty.CYM02 := null;
                                    Shipment.CYM02                  := null;
                                    Me_Inventory.CYM02      := null;
                                    ProductionQty.PYM02 := null;
                                    Shipment.PYM02                  := null;
                                    Me_Inventory.PYM02      := null;
                                    ProductionQty.CYM03 := null;
                                    Shipment.CYM03                  := null;
                                    Me_Inventory.CYM03      := null;
                                    ProductionQty.PYM03 := null;
                                    Shipment.PYM03                  := null;
                                    Me_Inventory.PYM03      := null;
                                    ProductionQty.CYM04 := null;
                                    Shipment.CYM04                  := null;
                                    Me_Inventory.CYM04      := null;
                                    ProductionQty.PYM04 := null;
                                    Shipment.PYM04                  := null;
                                    Me_Inventory.PYM04      := null;
                                    ProductionQty.CYM05 := null;
                                    Shipment.CYM05                  := null;
                                    Me_Inventory.CYM05      := null;
                                    ProductionQty.PYM05 := null;
                                    Shipment.PYM05                  := null;
                                    Me_Inventory.PYM05      := null;
```

```
ProductionQty. CYM06 := null;
Shipment. CYM06                    := null;
Me_Inventory. CYM06       := null;
ProductionQty. PYM06 := null;
Shipment. PYM06                    := null;
Me_Inventory. PYM06       := null;
ProductionQty. CYM07 := null;
Shipment. CYM07                    := null;
Me_Inventory. CYM07       := null;
ProductionQty. PYM07 := null;
Shipment. PYM07                    := null;
Me_Inventory. PYM07       := null;
ProductionQty. CYM08 := null;
Shipment. CYM08                    := null;
Me_Inventory. CYM08       := null;
ProductionQty. PYM08 := null;
Shipment. PYM08                    := null;
Me_Inventory. PYM08       := null;
ProductionQty. CYM09 := null;
Shipment. CYM09                    := null;
Me_Inventory. CYM09       := null;
ProductionQty. PYM09 := null;
Shipment. PYM09                    := null;
Me_Inventory. PYM09       := null;
ProductionQty. CYM10 := null;
Shipment. CYM10                    := null;
Me_Inventory. CYM10       := null;
ProductionQty. PYM10 := null;
Shipment. PYM10                    := null;
Me_Inventory. PYM10       := null;
ProductionQty. CYM11 := null;
Shipment. CYM11                    := null;
Me_Inventory. CYM11       := null;
ProductionQty. PYM11 := null;
Shipment. PYM11                    := null;
Me_Inventory. PYM11       := null;
ProductionQty. CYM12 := null;
Shipment. CYM12                    := null;
Me_Inventory. CYM12       := null;
ProductionQty. PYM12 := null;
Shipment. PYM12                    := null;
Me_Inventory. PYM12       := null;

for i in 1..nCurMonth loop
        if      i = nCurMonth   then
                select  sum(ProductionQty),
                                                        sum
(DomesticSales) + sum(Export),
                                                        sum
(Me_Inventory)
                into            ProductionQtyCYM,
                                                        Shi
pmentCYM,
                                                        Me_
InventoryCYM
                from            VW_PastRecordScope
                where           ISIC = reccm. ISIC
                  and           Year = inYear
          and           Month = i
          and           CommodityCode = reccm. Comm
odityCode
                  and           ItemType = reccm. It
emType
                  and           Pre_Rev_Mark = inPr
e_Rev_Mark
                  and           SurveyScope <= inSu
rveyScope;

        else
                select  sum(ProductionQty),
                                                        sum
(DomesticSales) + sum(Export),
                                                        sum
(Me_Inventory)
                into            ProductionQtyCYM,
                                                        Shi
pmentCYM,
```

```
InventoryCYM
                                                                    Me_
                                        from            VW_PastRecordScope
                                        where           ISIC = reccm.ISIC
                                          and           Year = inYear
                                  and           Month = i
                                  and           CommodityCode = reccm.Comm
oditycode
                                          and           ItemType = reccm.It
emtype
                                          and           Pre_Rev_Mark = '2'
                                          and           SurveyScope <= inSu
rveyscope;
                                end if;

                                select  sum(ProductionQty),
                                                                sum(Domesti
cSales) + sum(Export),
                                                                sum(Me_Inve
ntory)
                                        into    ProductionQtyPYM,
                                                                ShipmentPYM,
                                                                Me_Inventor
yPYM
                                        from    VW_PastRecordScope
                                        where   ISIC = reccm.ISIC
                                        and     Year = nPY
                                and     Month = i
                                and     CommodityCode = reccm.CommodityCode
                                        and     ItemType = reccm.ItemType
                                        and     Pre_Rev_Mark = '2'
                                        and     SurveyScope <= inSurveysco
pe;

                                if      i = 1 then
                                                ProductionQty.CYM01 := Pro
ductionQtyCYM;
                                                Shipment.CYM01
                                                Me_Inventory.CYM01      :=
Me_InventoryCYM;
                                                ProductionQty.PYM01 := Pro
ductionQtyPYM;
                                                Shipment.PYM01
                                                Me_Inventory.PYM01      :=
Me_InventoryPYM;
                                elsif   i = 2 then
                                                ProductionQty.CYM02 := Pro
ductionQtyCYM;
                                                Shipment.CYM02
                                                Me_Inventory.CYM02      :=
Me_InventoryCYM;
                                                ProductionQty.PYM02 := Pro
ductionQtyPYM;
                                                Shipment.PYM02
                                                Me_Inventory.PYM02      :=
Me_InventoryPYM;
                                elsif   i = 3 then
                                                ProductionQty.CYM03 := Pro
ductionQtyCYM;
                                                Shipment.CYM03
                                                Me_Inventory.CYM03      :=
Me_InventoryCYM;
                                                ProductionQty.PYM03 := Pro
ductionQtyPYM;
                                                Shipment.PYM03
                                                Me_Inventory.PYM03      :=
Me_InventoryPYM;
                                elsif   i = 4 then
                                                ProductionQty.CYM04 := Pro
ductionQtyCYM;
                                                Shipment.CYM04
```

```
                        := ShipmentCYM;
Me_InventoryCYM;
ductionQtyPYM;
                        := ShipmentPYM;
Me_InventoryPYM;

ductionQtyCYM;
                        := ShipmentCYM;
Me_InventoryCYM;
ductionQtyPYM;
                        := ShipmentPYM;
Me_InventoryPYM;

ductionQtyCYM;
                        := ShipmentCYM;
Me_InventoryCYM;
ductionQtyPYM;
                        := ShipmentPYM;
Me_InventoryPYM;

ductionQtyCYM;
                        := ShipmentCYM;
Me_InventoryCYM;
ductionQtyPYM;
                        := ShipmentPYM;
Me_InventoryPYM;

ductionQtyCYM;
                        := ShipmentCYM;
Me_InventoryCYM;
ductionQtyPYM;
                        := ShipmentPYM;
Me_InventoryPYM;

ductionQtyCYM;
                        := ShipmentCYM;
Me_InventoryCYM;
ductionQtyPYM;
                        := ShipmentPYM;
Me_InventoryPYM;

ductionQtyCYM;
```

```
                Me_Inventory.CYM04       :=
                ProductionQty.PYM04 := Pro
                Shipment.PYM04
                Me_Inventory.PYM04       :=
elsif   i = 5 then
                ProductionQty.CYM05 := Pro
                Shipment.CYM05
                Me_Inventory.CYM05       :=
                ProductionQty.PYM05 := Pro
                Shipment.PYM05
                Me_Inventory.PYM05       :=
elsif   i = 6 then
                ProductionQty.CYM06 := Pro
                Shipment.CYM06
                Me_Inventory.CYM06       :=
                ProductionQty.PYM06 := Pro
                Shipment.PYM06
                Me_Inventory.PYM06       :=
elsif   i = 7 then
                ProductionQty.CYM07 := Pro
                Shipment.CYM07
                Me_Inventory.CYM07       :=
                ProductionQty.PYM07 := Pro
                Shipment.PYM07
                Me_Inventory.PYM07       :=
elsif   i = 8 then
                ProductionQty.CYM08 := Pro
                Shipment.CYM08
                Me_Inventory.CYM08       :=
                ProductionQty.PYM08 := Pro
                Shipment.PYM08
                Me_Inventory.PYM08       :=
elsif   i = 9 then
                ProductionQty.CYM09 := Pro
                Shipment.CYM09
                Me_Inventory.CYM09       :=
                ProductionQty.PYM09 := Pro
                Shipment.PYM09
                Me_Inventory.PYM09       :=
elsif   i = 10 then
                ProductionQty.CYM10 := Pro
```

```
                                                        Shipment.CYM10
                := ShipmentCYM;
                                                        Me_Inventory.CYM10      :=
Me_InventoryCYM;
                                                        ProductionQty.PYM10 := Pro
ductionQtyPYM;
                                                        Shipment.PYM10
                := ShipmentPYM;
                                                        Me_Inventory.PYM10      :=
Me_InventoryPYM;
                                        elsif   i = 11 then
                                                        ProductionQty.CYM11 := Pro
ductionQtyCYM;
                                                        Shipment.CYM11
                := ShipmentCYM;
                                                        Me_Inventory.CYM11      :=
Me_InventoryCYM;
                                                        ProductionQty.PYM11 := Pro
ductionQtyPYM;
                                                        Shipment.PYM11
                := ShipmentPYM;
                                                        Me_Inventory.PYM11      :=
Me_InventoryPYM;
                                        elsif   i = 12 then
                                                        ProductionQty.CYM12 := Pro
ductionQtyCYM;
                                                        Shipment.CYM12
                := ShipmentCYM;
                                                        Me_Inventory.CYM12      :=
Me_InventoryCYM;
                                                        ProductionQty.PYM12 := Pro
ductionQtyPYM;
                                                        Shipment.PYM12
                := ShipmentPYM;
                                                        Me_Inventory.PYM12      :=
Me_InventoryPYM;
                                        end if;
                                    end loop;

-- Previous Year Average Data Select
                                    select  avg(ProductionQty),
                                                        avg(DomesticSales) +
avg(Export),
                                                        avg(Me_Inventory)
                                    into            ProductionQty.PYA,
                                                        Shipment.PYA,
                                                        Me_Inventory.PYA
                                from        VW_PastRecordScope
                                    where           ISIC = reccm.ISIC
                                    and             Year = nPY
                                    and             CommodityCode = reccm.CommodityCode
                                      and           ItemType = reccm.ItemType
                                      and           Pre_Rev_Mark = '2'
                                      and           SurveyScope <= inSurveyScope;

-- Previous 2 Year Total Data Select
                                    select  sum(ProductionQty),
                                                        sum(DomesticSales) +
sum(Export),
                                                        sum(Me_Inventory)
                                    into            ProductionQty.PY2T,
                                                        Shipment.PY2T,
                                                        Me_Inventory.PY2T
                                from        VW_PastRecordScope
                                    where           ISIC = reccm.ISIC
                                    and             Year = nPY2
                                    and             CommodityCode = reccm.CommodityCode
                                      and           ItemType = reccm.ItemType
                                      and           Pre_Rev_Mark = '2'
                                      and           SurveyScope <= inSurveyScope;

-- Insert Actual Value Report(Production)
                                    p_InsFigComm(inSeq,'1','1',inYear,reccm.ISIC,recc
m.CommodityCode,
                        reccl.IndustryTypeShortName,reccm.CommodityShortName,recum.UnitNam
e,
                        ProductionQty.PYA,ProductionQty.PYT,ProductionQty.CYT,
```

```
                        ProductionQty. CYM01, ProductionQty. CYM02, ProductionQty. CYM03,
                        ProductionQty. CYM04, ProductionQty. CYM05, ProductionQty. CYM06,
                        ProductionQty. CYM07, ProductionQty. CYM08, ProductionQty. CYM09,
                        ProductionQty. CYM10, ProductionQty. CYM11, ProductionQty. CYM12,
                                              ProductionQty. PY2T,
                        ProductionQty. PYM01, ProductionQty. PYM02, ProductionQty. PYM03,
                        ProductionQty. PYM04, ProductionQty. PYM05, ProductionQty. PYM06,
                        ProductionQty. PYM07, ProductionQty. PYM08, ProductionQty. PYM09,
                        ProductionQty. PYM10, ProductionQty. PYM11, ProductionQty. PYM12);

-- Insert Actual Value Report(Shipment)
                                        p_InsFigComm(inSeq, '2', '1', inYear, reccm. ISIC, recc
m. CommodityCode,
                        reccl. IndustryTypeShortName, reccm. CommodityShortName, recum. UnitNam
e,
                        Shipment. PYA, Shipment. PYT, Shipment. CYT,
                        Shipment. CYM01, Shipment. CYM02, Shipment. CYM03,
                        Shipment. CYM04, Shipment. CYM05, Shipment. CYM06,
                        Shipment. CYM07, Shipment. CYM08, Shipment. CYM09,
                        Shipment. CYM10, Shipment. CYM11, Shipment. CYM12,
                                              Shipment. PY2T,
                        Shipment. PYM01, Shipment. PYM02, Shipment. PYM03,
                        Shipment. PYM04, Shipment. PYM05, Shipment. PYM06,
                        Shipment. PYM07, Shipment. PYM08, Shipment. PYM09,
                        Shipment. PYM10, Shipment. PYM11, Shipment. PYM12);

-- Insert Actual Value Report(Inventory)
                                        p_InsFigComm(inSeq, '3', '1', inYear, reccm. ISIC, recc
m. CommodityCode,
                        reccl. IndustryTypeShortName, reccm. CommodityShortName, recum. UnitNam
e,
                        Me_Inventory. PYA, Me_Inventory. PYT, Me_Inventory. CYT,
                        Me_Inventory. CYM01, Me_Inventory. CYM02, Me_Inventory. CYM03,
                        Me_Inventory. CYM04, Me_Inventory. CYM05, Me_Inventory. CYM06,
                        Me_Inventory. CYM07, Me_Inventory. CYM08, Me_Inventory. CYM09,
                        Me_Inventory. CYM10, Me_Inventory. CYM11, Me_Inventory. CYM12,
                                              Me_Inventory. PY2T,
                        Me_Inventory. PYM01, Me_Inventory. PYM02, Me_Inventory. PYM03,
                        Me_Inventory. PYM04, Me_Inventory. PYM05, Me_Inventory. PYM06,
                        Me_Inventory. PYM07, Me_Inventory. PYM08, Me_Inventory. PYM09,
                        Me_Inventory. PYM10, Me_Inventory. PYM11, Me_Inventory. PYM12);

                                end if;

-- Item Type 2
                        elsif   reccm. ItemType = '2' then
-- Current Year Total Data Select
                                select  sum(ShipmentValue)
                                into            ShipmentValue. CYT
                          from          VW_PastRecordScope
                                where           ISIC = reccm. ISIC
                                and             Year = inYear
                                and             CommodityCode = reccm. CommodityCode
                                        and             ItemType = reccm. ItemType
                                        and             SurveyScope <= inSurveyScope
                                        and             ((Month = nCurMonth and Pre_Rev_Mark
= inPre_Rev_Mark)
                                        or              (Month < nCurMonth      and     Pre
_Rev_Mark = '2'));

-- Previous Year Total Data Select
                                select  sum(ShipmentValue)
                                into            ShipmentValue. PYT
                          from          VW_PastRecordScope
                                where           ISIC = reccm. ISIC
                                and             Year = nPY
                                and             CommodityCode = reccm. CommodityCode
                                  and           ItemType = reccm. ItemType
                                  and           Pre_Rev_Mark = '2'
                                  and           SurveyScope <= inSurveyScope;

-- Input Check
                                if      ShipmentValue. CYT is null
                                and     ShipmentValue. PYT is null then
                                        null;
                                else
```

```
-- Month Data Select
                                        ShipmentValue.CYM01      := null;
                                        ShipmentValue.PYM01      := null;
                                        ShipmentValue.CYM02      := null;
                                        ShipmentValue.PYM02      := null;
                                        ShipmentValue.CYM03      := null;
                                        ShipmentValue.PYM03      := null;
                                        ShipmentValue.CYM04      := null;
                                        ShipmentValue.PYM04      := null;
                                        ShipmentValue.CYM05      := null;
                                        ShipmentValue.PYM05      := null;
                                        ShipmentValue.CYM06      := null;
                                        ShipmentValue.PYM06      := null;
                                        ShipmentValue.CYM07      := null;
                                        ShipmentValue.PYM07      := null;
                                        ShipmentValue.CYM08      := null;
                                        ShipmentValue.PYM08      := null;
                                        ShipmentValue.CYM09      := null;
                                        ShipmentValue.PYM09      := null;
                                        ShipmentValue.CYM10      := null;
                                        ShipmentValue.PYM10      := null;
                                        ShipmentValue.CYM11      := null;
                                        ShipmentValue.PYM11      := null;
                                        ShipmentValue.CYM12      := null;
                                        ShipmentValue.PYM12      := null;

                                        for i in 1..nCurMonth loop
                                                if      i = nCurMonth   then
                                                        select  sum(ShipmentValue)
                                                        into            ShipmentValueCYM
                                                        from            VW_PastRecordScope
                                                        where           ISIC = reccm.ISIC
                                                          and           Year = inYear
                                                  and           Month = i
                                                  and           CommodityCode = reccm.Comm
odityCode
                                                          and           ItemType = reccm.It
emType
                                                          and           Pre_Rev_Mark = inPr
e_Rev_Mark
                                                          and           SurveyScope <= inSu
rveyScope;

                                                else
                                                        select  sum(ShipmentValue)
                                                                into    ShipmentValueCYM
                                                                from    VW_PastRecordScope
                                                                where   ISIC = reccm.ISIC
                                                                and     Year = inYear
                                                        and     Month = i
                                                        and     CommodityCode = reccm.Comm
odityCode
                                                                and     ItemType = reccm.It
emType
                                                                and     Pre_Rev_Mark = '2'
                                                                and     SurveyScope <= inSu
rveyScope;

                                                end if;

                                                select  sum(ShipmentValue)
                                                        into    ShipmentValuePYM
                                                        from    VW_PastRecordScope
                                                        where   ISIC = reccm.ISIC
                                                          and   Year = nPY
                                                        and     Month = i
                                                and     CommodityCode = reccm.CommodityCode
                                                        and     ItemType = reccm.ItemType
                                                        and     Pre_Rev_Mark = '2'
                                                        and     SurveyScope <= inSurveySco
pe;

                                                if      i = 1 then
                                                                ShipmentValue.CYM01      :=
ShipmentValueCYM;
                                                                ShipmentValue.PYM01      :=
ShipmentValuePYM;
```

```
                                        elsif   i = 2 then
                                                ShipmentValue.CYM02     :=
ShipmentValueCYM;
                                                ShipmentValue.PYM02     :=
ShipmentValuePYM;
                                        elsif   i = 3 then
                                                ShipmentValue.CYM03     :=
ShipmentValueCYM;
                                                ShipmentValue.PYM03     :=
ShipmentValuePYM;
                                        elsif   i = 4 then
                                                ShipmentValue.CYM04     :=
ShipmentValueCYM;
                                                ShipmentValue.PYM04     :=
ShipmentValuePYM;
                                        elsif   i = 5 then
                                                ShipmentValue.CYM05     :=
ShipmentValueCYM;
                                                ShipmentValue.PYM05     :=
ShipmentValuePYM;
                                        elsif   i = 6 then
                                                ShipmentValue.CYM06     :=
ShipmentValueCYM;
                                                ShipmentValue.PYM06     :=
ShipmentValuePYM;
                                        elsif   i = 7 then
                                                ShipmentValue.CYM07     :=
ShipmentValueCYM;
                                                ShipmentValue.PYM07     :=
ShipmentValuePYM;
                                        elsif   i = 8 then
                                                ShipmentValue.CYM08     :=
ShipmentValueCYM;
                                                ShipmentValue.PYM08     :=
ShipmentValuePYM;
                                        elsif   i = 9 then
                                                ShipmentValue.CYM09     :=
ShipmentValueCYM;
                                                ShipmentValue.PYM09     :=
ShipmentValuePYM;
                                        elsif   i = 10 then
                                                ShipmentValue.CYM10     :=
ShipmentValueCYM;
                                                ShipmentValue.PYM10     :=
ShipmentValuePYM;
                                        elsif   i = 11 then
                                                ShipmentValue.CYM11     :=
ShipmentValueCYM;
                                                ShipmentValue.PYM11     :=
ShipmentValuePYM;
                                        elsif   i = 12 then
                                                ShipmentValue.CYM12     :=
ShipmentValueCYM;
                                                ShipmentValue.PYM12     :=
ShipmentValuePYM;
                                        end if;
                                end loop;

-- Previous Year Average Data Select
                                select  avg(ShipmentValue)
                                        into    ShipmentValue.PYA
                                from    VW_PastRecordScope
                                        where   ISIC = reccm.ISIC
                                        and     Year = nPY
                                        and     CommodityCode = reccm.CommodityCode
                                        and     ItemType = reccm.ItemType
                                        and     Pre_Rev_Mark = '2'
                                        and     SurveyScope <= inSurveyScope;

-- Previous 2 Year Total Data Select
                                select  sum(ShipmentValue)
                                        into    ShipmentValue.PY2T
                                from    VW_PastRecordScope
                                        where   ISIC = reccm.ISIC
                                        and     Year = nPY2
                                        and     CommodityCode = reccm.CommodityCode
                                        and     ItemType = reccm.ItemType
```

```
                                        and     Pre_Rev_Mark = '2'
                                        and     SurveyScope <= inSurveyScope;

-- Insert Actual Value Report(ShipmentValue)
                                p_InsFigComm(inSeq, '4', '1', inYear, reccm.ISIC, recc
m.CommodityCode,
                    reccl.IndustryTypeShortName, reccm.CommodityShortName, recum.UnitNam
e,
                    ShipmentValue.PYA, ShipmentValue.PYT, ShipmentValue.CYT,
                    ShipmentValue.CYM01, ShipmentValue.CYM02, ShipmentValue.CYM03,
                    ShipmentValue.CYM04, ShipmentValue.CYM05, ShipmentValue.CYM06,
                    ShipmentValue.CYM07, ShipmentValue.CYM08, ShipmentValue.CYM09,
                    ShipmentValue.CYM10, ShipmentValue.CYM11, ShipmentValue.CYM12,
                                ShipmentValue.PY2T,
                    ShipmentValue.PYM01, ShipmentValue.PYM02, ShipmentValue.PYM03,
                    ShipmentValue.PYM04, ShipmentValue.PYM05, ShipmentValue.PYM06,
                    ShipmentValue.PYM07, ShipmentValue.PYM08, ShipmentValue.PYM09,
                    ShipmentValue.PYM10, ShipmentValue.PYM11, ShipmentValue.PYM12);

                        end if;

-- Item Type 6
                    elsif   reccm.ItemType = '6' then
-- Current Year Total Data Select
                        select  sum(Capacity)
                        into            Capacity.CYT
                  from          VW_PastRecordScope
                        where           ISIC = reccm.ISIC
                        and             Year = inYear
                        and             CommodityCode = reccm.CommodityCode
                                and             ItemType = reccm.ItemType
                                and             SurveyScope <= inSurveyScope
                                and             ((Month = nCurMonth and Pre_Rev_Mark
= inPre_Rev_Mark)
                                or              (Month < nCurMonth      and     Pre
_Rev_Mark = '2'));

-- Previous Year Total Data Select
                        select  sum(Capacity)
                        into            Capacity.PYT
                  from          VW_PastRecordScope
                        where           ISIC = reccm.ISIC
                        and             Year = nPY
                        and             CommodityCode = reccm.CommodityCode
                          and           ItemType = reccm.ItemType
                          and           Pre_Rev_Mark = '2'
                          and           SurveyScope <= inSurveyScope;

                        if      Capacity.CYT    is null
                        and     Capacity.PYT    is null then
                                null;
                        else

-- Month Data Select
                                Capacity.CYM01  := null;
                                Capacity.PYM01  := null;
                                Capacity.CYM02  := null;
                                Capacity.PYM02  := null;
                                Capacity.CYM03  := null;
                                Capacity.PYM03  := null;
                                Capacity.CYM04  := null;
                                Capacity.PYM04  := null;
                                Capacity.CYM05  := null;
                                Capacity.PYM05  := null;
                                Capacity.CYM06  := null;
                                Capacity.PYM06  := null;
                                Capacity.CYM07  := null;
                                Capacity.PYM07  := null;
                                Capacity.CYM08  := null;
                                Capacity.PYM08  := null;
                                Capacity.CYM09  := null;
                                Capacity.PYM09  := null;
                                Capacity.CYM10  := null;
                                Capacity.PYM10  := null;
                                Capacity.CYM11  := null;
                                Capacity.PYM11  := null;
                                Capacity.CYM12  := null;
```

```
                                        Capacity.PYM12  := null;

                                        for i in 1..nCurMonth loop
                                                if      i = nCurMonth then
                                                        select  sum(Capacity)
                                                                into    CapacityCYM
                                                                from    VW_PastRecordScope
                                                                where   ISIC = reccm.ISIC
                                                                and     Year = inYear
                                                        and     Month = i
                                                        and     CommodityCode = reccm.Comm
oditycode
                                                                and     ItemType = reccm.It
emtype
                                                                and     Pre_Rev_Mark = inPr
e_Rev_Mark
                                                                and     SurveyScope <= inSu
rveyScope;

                                                else
                                                        select  sum(Capacity)
                                                                into    CapacityCYM
                                                                from    VW_PastRecordScope
                                                                where   ISIC = reccm.ISIC
                                                                and     Year = inYear
                                                        and     Month = i
                                                        and     CommodityCode = reccm.Comm
oditycode
                                                                and     ItemType = reccm.It
emtype
                                                                and     Pre_Rev_Mark = '2'
                                                                and     SurveyScope <= inSu
rveyScope;
                                                end if;

                                                select  sum(Capacity)
                                                        into    CapacityPYM
                                                        from    VW_PastRecordScope
                                                        where   ISIC = reccm.ISIC
                                                        and     Year = nPY
                                                and     Month = i
                                                and     CommodityCode = reccm.CommodityCode
                                                        and     ItemType = reccm.ItemType
                                                        and     Pre_Rev_Mark = '2'
                                                        and     SurveyScope <= inSurveySco
pe;

                                                if      i = 1 then
                                                                Capacity.CYM01  := Capacity
CYM;
                                                                Capacity.PYM01  := Capacity
PYM;
                                                elsif   i = 2 then
                                                                Capacity.CYM02  := Capacity
CYM;
                                                                Capacity.PYM02  := Capacity
PYM;
                                                elsif   i = 3 then
                                                                Capacity.CYM03  := Capacity
CYM;
                                                                Capacity.PYM03  := Capacity
PYM;
                                                elsif   i = 4 then
                                                                Capacity.CYM04  := Capacity
CYM;
                                                                Capacity.PYM04  := Capacity
PYM;
                                                elsif   i = 5 then
                                                                Capacity.CYM05  := Capacity
CYM;
                                                                Capacity.PYM05  := Capacity
PYM;
                                                elsif   i = 6 then
                                                                Capacity.CYM06  := Capacity
CYM;
                                                                Capacity.PYM06  := Capacity
PYM;
```

```
                                        elsif   i = 7 then
                                                Capacity.CYM07  := Capacity
CYM;
                                                Capacity.PYM07  := Capacity
PYM;
                                        elsif   i = 8 then
                                                Capacity.CYM08  := Capacity
CYM;
                                                Capacity.PYM08  := Capacity
PYM;
                                        elsif   i = 9 then
                                                Capacity.CYM09  := Capacity
CYM;
                                                Capacity.PYM09  := Capacity
PYM;
                                        elsif   i = 10 then
                                                Capacity.CYM10  := Capacity
CYM;
                                                Capacity.PYM10  := Capacity
PYM;
                                        elsif   i = 11 then
                                                Capacity.CYM11  := Capacity
CYM;
                                                Capacity.PYM11  := Capacity
PYM;
                                        elsif   i = 12 then
                                                Capacity.CYM12  := Capacity
CYM;
                                                Capacity.PYM12  := Capacity
PYM;
                                        end if;
                                end loop;

-- Previous Year Average Data Select
                                select  avg(Capacity)
                                into            Capacity.PYA
                        from            VW_PastRecordScope
                                where           ISIC = reccm.ISIC
                                and             Year = nPY
                                and             CommodityCode = reccm.CommodityCode
                                  and           ItemType = reccm.ItemType
                                  and           Pre_Rev_Mark = '2'
                                  and           SurveyScope <= inSurveyScope;

-- Previous 2 Year Total Data Select
                                select  sum(Capacity)
                                into            Capacity.PY2T
                        from            VW_PastRecordScope
                                where           ISIC = reccm.ISIC
                                and             Year = nPY2
                                and             CommodityCode = reccm.CommodityCode
                                  and           ItemType = reccm.ItemType
                                  and           Pre_Rev_Mark = '2'
                                  and           SurveyScope <= inSurveyScope;

-- Insert Actual Value Report(Capacity)
                                p_InsFigComm(inSeq,'5','1',inYear,reccm.ISIC,recc
m.CommodityCode,
                        reccl.IndustryTypeShortName,reccm.CommodityShortName,recum.UnitNam
e,
                        Capacity.PYA,Capacity.PYT,Capacity.CYT,
                        Capacity.CYM01,Capacity.CYM02,Capacity.CYM03,
                        Capacity.CYM04,Capacity.CYM05,Capacity.CYM06,
                        Capacity.CYM07,Capacity.CYM08,Capacity.CYM09,
                        Capacity.CYM10,Capacity.CYM11,Capacity.CYM12,
                                        Capacity.PY2T,
                        Capacity.PYM01,Capacity.PYM02,Capacity.PYM03,
                        Capacity.PYM04,Capacity.PYM05,Capacity.PYM06,
                        Capacity.PYM07,Capacity.PYM08,Capacity.PYM09,
                        Capacity.PYM10,Capacity.PYM11,Capacity.PYM12);

                                end if;

-- Item Type 4
                        elsif   reccm.ItemType = '4' then
-- Current Year Total Data Select
                                select  sum(RwMt_MEInventory)
```

```
                                        into    RwMt_MEInventory.CYT
                                from    VW_PastRecordScope
                                        where   ISIC = reccm.ISIC
                                        and     Year = inYear
                                        and     CommodityCode = reccm.CommodityCode
                                                and     ItemType = reccm.ItemType
                                                and     SurveyScope <= inSurveyScope
                                                and     ((Month = nCurMonth and Pre_Rev_Mark
= inPre_Rev_Mark)
                                                or      (Month < nCurMonth      and     Pre
_Rev_Mark = '2'));

-- Previous Year Total Data Select
                                select  sum(RwMt_MEInventory)
                                        into    RwMt_MEInventory.PYT
                                from    VW_PastRecordScope
                                        where   ISIC = reccm.ISIC
                                        and     Year = nPY
                                        and     CommodityCode = reccm.CommodityCode
                                        and     ItemType = reccm.ItemType
                                        and     Pre_Rev_Mark = '2'
                                        and     SurveyScope <= inSurveyScope;

-- Output Checking
                                if      RwMt_MEInventory.CYT is null
                                and     RwMt_MEInventory.PYT is null then
                                        null;
                                else

-- Month Data Select
                                        RwMt_MEInventory.CYM01  := null;
                                        RwMt_MEInventory.PYM01  := null;
                                        RwMt_MEInventory.CYM02  := null;
                                        RwMt_MEInventory.PYM02  := null;
                                        RwMt_MEInventory.CYM03  := null;
                                        RwMt_MEInventory.PYM03  := null;
                                        RwMt_MEInventory.CYM04  := null;
                                        RwMt_MEInventory.PYM04  := null;
                                        RwMt_MEInventory.CYM05  := null;
                                        RwMt_MEInventory.PYM05  := null;
                                        RwMt_MEInventory.CYM06  := null;
                                        RwMt_MEInventory.PYM06  := null;
                                        RwMt_MEInventory.CYM07  := null;
                                        RwMt_MEInventory.PYM07  := null;
                                        RwMt_MEInventory.CYM08  := null;
                                        RwMt_MEInventory.PYM08  := null;
                                        RwMt_MEInventory.CYM09  := null;
                                        RwMt_MEInventory.PYM09  := null;
                                        RwMt_MEInventory.CYM10  := null;
                                        RwMt_MEInventory.PYM10  := null;
                                        RwMt_MEInventory.CYM11  := null;
                                        RwMt_MEInventory.PYM11  := null;
                                        RwMt_MEInventory.CYM12  := null;
                                        RwMt_MEInventory.PYM12  := null;

                                        for i in 1..nCurMonth loop
                                                if      i = nCurMonth then
                                                        select  sum(RwMt_MEInventory)
                                                        into            RwMt_MEInventoryCYM
                                                        from            VW_PastRecordScope
                                                        where           ISIC = reccm.ISIC
                                                          and           Year = inYear
                                                  and           Month = i
                                                  and           CommodityCode = reccm.Comm
odityCode
                                                          and           ItemType = reccm.It
emType
                                                          and           Pre_Rev_Mark = inPr
e_Rev_Mark
                                                          and           SurveyScope <= inSu
rveyScope;
                                                else
                                                        select  sum(RwMt_MEInventory)
                                                        into            RwMt_MEInventoryCYM
                                                        from            VW_PastRecordScope
                                                        where           ISIC = reccm.ISIC
                                                          and           Year = inYear
```

```
oditycode

emtype

rveyscope;

pe;

RwMt_MEInventoryCYM;
RwMt_MEInventoryPYM;
RwMt_MEInventoryCYM;
RwMt_MEInventoryPYM;
RwMt_MEInventoryCYM;
RwMt_MEInventoryPYM;
RwMt_MEInventoryCYM;
RwMt_MEInventoryPYM;
RwMt_MEInventoryCYM;
RwMt_MEInventoryPYM;
RwMt_MEInventoryCYM;
RwMt_MEInventoryPYM;
RwMt_MEInventoryCYM;
RwMt_MEInventoryPYM;
RwMt_MEInventoryCYM;
RwMt_MEInventoryPYM;
RwMt_MEInventoryCYM;
RwMt_MEInventoryPYM;
RwMt_MEInventoryCYM;
```

```
  and          Month = i
  and          CommodityCode = reccm.Comm
          and          ItemType = reccm.It
          and          Pre_Rev_Mark = '2'
          and          SurveyScope <= inSu
end if;

select  sum(RwMt_MEInventory)
        into    RwMt_MEInventoryPYM
        from    VW_PastRecordScope
        where   ISIC = reccm.ISIC
          and   Year = nPY
  and   Month = i
  and   CommodityCode = reccm.CommodityCode
          and   ItemType = reccm.ItemType
          and   Pre_Rev_Mark = '2'
          and   SurveyScope <= inSurveySco

if      i = 1 then
                RwMt_MEInventory.CYM01  :=
                RwMt_MEInventory.PYM01  :=
elsif   i = 2 then
                RwMt_MEInventory.CYM02  :=
                RwMt_MEInventory.PYM02  :=
elsif   i = 3 then
                RwMt_MEInventory.CYM03  :=
                RwMt_MEInventory.PYM03  :=
elsif   i = 4 then
                RwMt_MEInventory.CYM04  :=
                RwMt_MEInventory.PYM04  :=
elsif   i = 5 then
                RwMt_MEInventory.CYM05  :=
                RwMt_MEInventory.PYM05  :=
elsif   i = 6 then
                RwMt_MEInventory.CYM06  :=
                RwMt_MEInventory.PYM06  :=
elsif   i = 7 then
                RwMt_MEInventory.CYM07  :=
                RwMt_MEInventory.PYM07  :=
elsif   i = 8 then
                RwMt_MEInventory.CYM08  :=
                RwMt_MEInventory.PYM08  :=
elsif   i = 9 then
                RwMt_MEInventory.CYM09  :=
                RwMt_MEInventory.PYM09  :=
elsif   i = 10 then
                RwMt_MEInventory.CYM10  :=
                RwMt_MEInventory.PYM10  :=
elsif   i = 11 then
                RwMt_MEInventory.CYM11  :=
                RwMt_MEInventory.PYM11  :=
```

```
RwMt_MEInventoryPYM;
                                                elsif   i = 12 then
                                                                RwMt_MEInventory.CYM12  :=
RwMt_MEInventoryCYM;
                                                                RwMt_MEInventory.PYM12  :=
RwMt_MEInventoryPYM;
                                                end if;
                                        end loop;

-- Previous Year Average Data Select
                                        select  avg(RwMt_MEInventory)
                                        into            RwMt_MEInventory.PYA
                                from            VW_PastRecordScope
                                        where           ISIC = reccm.ISIC
                                        and             Year = nPY
                                        and             CommodityCode = reccm.CommodityCode
                                          and           ItemType = reccm.ItemType
                                          and           Pre_Rev_Mark = '2'
                                          and           SurveyScope <= inSurveyScope;

-- Previous 2 Year Total Data Select
                                        select  sum(RwMt_MEInventory)
                                        into            RwMt_MEInventory.PY2T
                                from            VW_PastRecordScope
                                        where           ISIC = reccm.ISIC
                                        and             Year = nPY2
                                        and             CommodityCode = reccm.CommodityCode
                                          and           ItemType = reccm.ItemType
                                          and           Pre_Rev_Mark = '2'
                                          and           SurveyScope <= inSurveyScope;

-- Insert Actual Value Report(RwMt_MEInventory)
                                        p_InsFigComm(inSeq,'6','1',inYear,reccm.ISIC,recc
m.CommodityCode,
                        reccl.IndustryTypeShortName,reccm.CommodityShortName,recum.UnitNam
e,
                        RwMt_MEInventory.PYA,RwMt_MEInventory.PYT,RwMt_MEInventory.CYT,
                        RwMt_MEInventory.CYM01,RwMt_MEInventory.CYM02,RwMt_MEInventory.CYM
03,
                        RwMt_MEInventory.CYM04,RwMt_MEInventory.CYM05,RwMt_MEInventory.CYM
06,
                        RwMt_MEInventory.CYM07,RwMt_MEInventory.CYM08,RwMt_MEInventory.CYM
09,
                        RwMt_MEInventory.CYM10,RwMt_MEInventory.CYM11,RwMt_MEInventory.CYM
12,
                                                RwMt_MEInventory.PY2T,
                        RwMt_MEInventory.PYM01,RwMt_MEInventory.PYM02,RwMt_MEInventory.PYM
03,
                        RwMt_MEInventory.PYM04,RwMt_MEInventory.PYM05,RwMt_MEInventory.PYM
06,
                        RwMt_MEInventory.PYM07,RwMt_MEInventory.PYM08,RwMt_MEInventory.PYM
09,
                        RwMt_MEInventory.PYM10,RwMt_MEInventory.PYM11,RwMt_MEInventory.PYM
12);

                                end if;

                        end if;

                end if;

                fetch csrcm into        reccm.ISIC,reccm.ItemType,reccm.CommodityCode,
                                        reccm.CommodityShortName,reccl.IndustryTypeShortNa
me,recum.UnitName;

        end loop;

        close csrcm;

        DBMS_OUTPUT.PUT_LINE('Ended p_ActualValue ' || to_char(sysdate,'HH24:MI:SS'));

end;

-------------------------------------------------------
-- Growth Rate Against
-------------------------------------------------------
procedure p_GrowthRateAgainst (
```

```
        inSeq                                                   in      number) is

-- Record Area Define
        rectf                                                           TimeFigures%rowtype;
        rectfPM                                                 TimeFigures%rowtype;
        rectfPY                                                 TimeFigures%rowtype;

-- Time Figures Cursor
        Cursor csrtf is
                select  *
    from                TimeFigures
        where           Seq = inSeq;

Begin
-- Commodity Master Open Fetch
        open csrtf;
        fetch csrtf into        rectf;

-- Time Figures Fetch Looping
        while csrtf%found loop

-- Previous Month Calculation
                rectfPM. CYM01 := f_CalGRA(rectf. CYM01, rectf. PYM12);
                rectfPM. CYM02 := f_CalGRA(rectf. CYM02, rectf. CYM01);
                rectfPM. CYM03 := f_CalGRA(rectf. CYM03, rectf. CYM02);
                rectfPM. CYM04 := f_CalGRA(rectf. CYM04, rectf. CYM03);
                rectfPM. CYM05 := f_CalGRA(rectf. CYM05, rectf. CYM04);
                rectfPM. CYM06 := f_CalGRA(rectf. CYM06, rectf. CYM05);
                rectfPM. CYM07 := f_CalGRA(rectf. CYM07, rectf. CYM06);
                rectfPM. CYM08 := f_CalGRA(rectf. CYM08, rectf. CYM07);
                rectfPM. CYM09 := f_CalGRA(rectf. CYM09, rectf. CYM08);
                rectfPM. CYM10 := f_CalGRA(rectf. CYM10, rectf. CYM09);
                rectfPM. CYM11 := f_CalGRA(rectf. CYM11, rectf. CYM10);
                rectfPM. CYM12 := f_CalGRA(rectf. CYM12, rectf. CYM11);

                p_InsFigComm(rectf. seq, rectf. FigureType, '2', rectf. Year, rectf. ISIC,
                        rectf. CommodityCode, rectf. IndustryTypeShortName, rectf. CommoditySho
rtName,
                        rectf. UnitName, null, null, null,
        rectfPM. CYM01, rectfPM. CYM02, rectfPM. CYM03, rectfPM. CYM04,
                        rectfPM. CYM05, rectfPM. CYM06, rectfPM. CYM07, rectfPM. CYM08,
                        rectfPM. CYM09, rectfPM. CYM10, rectfPM. CYM11, rectfPM. CYM12,
                        null, null, null, null, null, null, null, null, null, null, null, null, null);

-- Same Month Of Previous Year Caluculation
                rectfPY. PYT     := f_CalGRA(rectf. PYT, rectf. PY2T);
                rectfPY. CYT     := f_CalGRA(rectf. CYT, rectf. PYT);

                rectfPY. CYM01 := f_CalGRA(rectf. CYM01, rectf. PYM01);
                rectfPY. CYM02 := f_CalGRA(rectf. CYM02, rectf. PYM02);
                rectfPY. CYM03 := f_CalGRA(rectf. CYM03, rectf. PYM03);
                rectfPY. CYM04 := f_CalGRA(rectf. CYM04, rectf. PYM04);
                rectfPY. CYM05 := f_CalGRA(rectf. CYM05, rectf. PYM05);
                rectfPY. CYM06 := f_CalGRA(rectf. CYM06, rectf. PYM06);
                rectfPY. CYM07 := f_CalGRA(rectf. CYM07, rectf. PYM07);
                rectfPY. CYM08 := f_CalGRA(rectf. CYM08, rectf. PYM08);
                rectfPY. CYM09 := f_CalGRA(rectf. CYM09, rectf. PYM09);
                rectfPY. CYM10 := f_CalGRA(rectf. CYM10, rectf. PYM10);
                rectfPY. CYM11 := f_CalGRA(rectf. CYM11, rectf. PYM11);
                rectfPY. CYM12 := f_CalGRA(rectf. CYM12, rectf. PYM12);

                p_InsFigComm(rectf. seq, rectf. FigureType, '3', rectf. Year, rectf. ISIC,
                        rectf. CommodityCode, rectf. IndustryTypeShortName, rectf. CommoditySho
rtName,
                        rectf. UnitName, null, rectfPY. PYT, rectfPY. CYT,
        rectfPY. CYM01, rectfPY. CYM02, rectfPY. CYM03, rectfPY. CYM04,
                        rectfPY. CYM05, rectfPY. CYM06, rectfPY. CYM07, rectfPY. CYM08,
                        rectfPY. CYM09, rectfPY. CYM10, rectfPY. CYM11, rectfPY. CYM12,
                        null, null, null, null, null, null, null, null, null, null, null, null, null);

                fetch csrtf into        rectf;

        end loop;

        close csrtf;

end;
```

```
-----------------------------------------------------------
-- Time Series List of Figures by Commodity Main
-----------------------------------------------------------
procedure p_FigCommMk (
        inSeq                                                   in      number,
        inYear                                          in      number,
        inAggregationRange      in      varchar2,
        inISIC                                          in      varchar2,
        inSurveyScope                           in      varchar2,
        inPre_Rev_Mark                  in      varchar2) is
begin
        p_ActualValue(inSeq, inYear, inAggregationRange, inISIC, inSurveyScope, inPre_Rev_Mark);
        p_GrowthRateAgainst(inSeq);
end;

end TimeSeriesMk;
/
show errors
/
```

```
-- ==================================================
-- Package Name : User Request Report
-- Package ID   : UserRep
-- Create Date  : 25/Feb/2000
-- Replace Date : xx/xxx/xxxx
-- Author       : T. Nakamura
--
-- File         : @c:¥jica¥UserRep;
-- Test                                :       BEGIN UserRep. p_AnnualColSt(77, '1', 'XXXXXX
', '1', 1999, '1'); commit; END;
-- Test                                :       BEGIN UserRep. p_CPMYRawEst(2, '171110', nul
l, 1999, 12, '2', '1'); commit; END;
-- Test                                :       BEGIN UserRep. p_CPMYRawISIC(1, '2', '151210
', 1999, 12, '1', '1'); commit; END;
-- ==================================================
create or replace package UserRep is

procedure p_AnnualColSt (
        inSeq
                in      number,
        inAggregationRange                              in      varchar2,
        inISIC                                                                  i
n       varchar2,
        inSelContinuousRespondent       in      varchar2,
        inYear                                                                  i
n       number,
        inSurveyScope                                                   in      varchar2);

procedure p_CPMYRawEst (
        inSeq
                in      number,
        inISIC                                                                  i
n       varchar2,
        inRegistrationNo                                        in      varchar2,       --
Null:All Data Select    Not Null:Select
                                        -- (Ignore Input Parameter about inSelContinuousRe
spondent and inSurveyScope)
        inYear                                                                  i
n       number,
        inMonth                                                                 i
n       number,
        inSelContinuousRespondent       in      varchar2,       -- 0:None       1:Only 2:All
        inSurveyScope                                                   in      varchar2);

procedure p_CPMYRawISIC (
        inSeq
                in      number,
        inAggregationRange                              in      varchar2,
        inISIC                                                                  i
n       varchar2,
        inYear                                                                  i
n       number,
        inMonth                                                                 i
n       number,
        inSelContinuousRespondent       in      varchar2,       -- 0:None       1:Only 2:All
        inSurveyScope                                                   in      varchar2);

end UserRep;
/
show errors
/

create or replace package body UserRep is
--------------------------------------------------------
-- Collection Status of Establishment
--
-- inSelContinuousRespondent
--      0:None
--      1:Only
--      2:All
--------------------------------------------------------
procedure p_AnnualColSt (
        inSeq
                in      number,
        inAggregationRange                              in      varchar2,
        inISIC                                                                  i
n       varchar2,
```

```
        inSelContinuousRespondent       in      varchar2,
        inYear                                                                  i
n       number,
        inSurveyScope                                                   in      varchar2) is

-- Record Area Define
        recsh           SurveyHeader%rowtype;
        recem           EstablishmentMaster%rowtype;
        recac           AnnualCollectionStatus%rowtype;

-- Work Area Define
        sISIC2d         Varchar2(2);
        sISIC4d         Varchar2(4);
        sCurYM          number;

-- Establishment Master Table Cursor
        Cursor  csrem is
                select  ISIC, RegistrationNo, EstablishmentName, CompanyTel, ContinuousRespond
ent
                        from  EstablishmentMaster
                        where   SurveyScope <= inSurveyScope;

begin
-- ISIC Edit
        sISIC2d := substr(inISIC, 1, 2);
        sISIC4d := substr(inISIC, 1, 4);

-- Set Annual Collection Status Table For Header
        recac.Seq
                        := inSeq;
  recac.Year                                                                    :=
inYear;
  recac.SurveyScope                                             := inSurveyScope;
        if      inSelContinuousRespondent = '0' then
                recac.SelContinuousRespondent   := 'None';
        elsif   inSelContinuousRespondent = '1' then
                recac.SelContinuousRespondent   := 'Only';
        elsif   inSelContinuousRespondent = '2' then
                recac.SelContinuousRespondent   := 'All';
        end if;

-- Get Current Batch Process
        select  max(Year * 100 + Month)
                into    sCurYM
                from    BatchControl
                where   PreliminaryDate is not null
                        or      RevisedDate is not null;

-- Establishment Master Cursor Open Fetch
        open    csrem;
        fetch   csrem
                into    recem.ISIC, recem.RegistrationNo,
                                        recem.EstablishmentName, recem.CompanyTel, recem.Con
tinuousRespondent;

-- Establishment Master Cursor Fetch Looping
        while csrem%found loop

-- Check Aggregation Range
                if      (inAggregationRange = '1')
                or      (inAggregationRange = '2' and substr(recem.ISIC, 1, 2) = sISIC2d)
                or      (inAggregationRange = '3' and substr(recem.ISIC, 1, 4) = sISIC4d)
                or      (inAggregationRange = '4' and substr(recem.ISIC, 1, 6) = inISIC) then

-- Check Continuous Respondent
                        if      (inSelContinuousRespondent = '0' and recem.ContinuousRespo
ndent = '0')
                        or      (inSelContinuousRespondent = '1' and recem.ContinuousRespo
ndent = '1')
                        or      (inSelContinuousRespondent = '2') then

-- Set Annual Collection Status Column
                recac.ISIC
                        := recem.ISIC;
                recac.RegistrationNo                                    := recem.Registrati
onNo;
                recac.EstablishmentName                 := recem.EstablishmentName;
```

```
                                recac.ContinuousRespondent              := recem.Continuous
Respondent;
                                recac.CompanyTel
                        := recem.CompanyTel;
                                recac.EstimatedMark01                                   :=
null;
                                recac.EstimatedMark02                                   :=
null;
                                recac.EstimatedMark03                                   :=
null;
                                recac.EstimatedMark04                                   :=
null;
                                recac.EstimatedMark05                                   :=
null;
                                recac.EstimatedMark06                                   :=
null;
                                recac.EstimatedMark07                                   :=
null;
                                recac.EstimatedMark08                                   :=
null;
                                recac.EstimatedMark09                                   :=
null;
                                recac.EstimatedMark10                                   :=
null;
                                recac.EstimatedMark11                                   :=
null;
                                recac.EstimatedMark12                                   :=
null;

-- Check Survey Header
                                for i in 1..12 loop
                                        begin
                                                select  EstimatedMark
                                                        into    recsh.EstimatedMark
                                                        from    SurveyHeader
                                                        where   ISIC = recem.ISIC
                                                        and     Year = inYear
                                                and     Month = i
                                                and     RegistrationNo = recem.Registratio
nNo;

                                                if      recsh.EstimatedMark = '0'       then
                                                        recsh.EstimatedMark := '/';
                                                elsif   recsh.EstimatedMark = '1'       then
                                                        recsh.EstimatedMark := 'I';
                                                elsif   recsh.EstimatedMark = '2'       then
                                                        recsh.EstimatedMark := 'A';
                                                end if;

                                        exception
                                                when no_data_found then
                                                        if inYear * 100 + i > sCurYM then
                                                                recsh.EstimatedMark := nul
l;
                                                        else
                                                                recsh.EstimatedMark := 'X';
                                                        end if;
                                        end;

-- Set EstimatedMark
                                        if      i = 1 then
                                                recac.EstimatedMark01   := recsh.EstimatedM
ark;
                                        elsif   i = 2 then
                                                recac.EstimatedMark02   := recsh.EstimatedM
ark;
                                        elsif   i = 3 then
                                                recac.EstimatedMark03   := recsh.EstimatedM
ark;
                                        elsif   i = 4 then
                                                recac.EstimatedMark04   := recsh.EstimatedM
ark;
                                        elsif   i = 5 then
                                                recac.EstimatedMark05   := recsh.EstimatedM
ark;
                                        elsif   i = 6 then
                                                recac.EstimatedMark06   := recsh.EstimatedM
```

```
ark;
                                        elsif   i = 7 then
                                                recac. EstimatedMark07   := recsh. EstimatedM
ark;
                                        elsif   i = 8 then
                                                recac. EstimatedMark08   := recsh. EstimatedM
ark;
                                        elsif   i = 9 then
                                                recac. EstimatedMark09   := recsh. EstimatedM
ark;
                                        elsif   i = 10 then
                                                recac. EstimatedMark10   := recsh. EstimatedM
ark;
                                        elsif   i = 11 then
                                                recac. EstimatedMark11   := recsh. EstimatedM
ark;
                                        elsif   i = 12 then
                                                recac. EstimatedMark12   := recsh. EstimatedM
ark;
                                        end if;

                                end loop;

-- Insert Annual Collection Status Table
                                insert  into    AnnualCollectionStatus(
                                                Seq,
                                Year,
                                ISIC,
                                SurveyScope,
                                                SelContinuousRespondent,
                              RegistrationNo,
                              CompanyTel,
                                EstablishmentName,
                                                ContinuousRespondent,
                                                EstimatedMark01,
                                                EstimatedMark02,
                                                EstimatedMark03,
                                                EstimatedMark04,
                                                EstimatedMark05,
                                                EstimatedMark06,
                                                EstimatedMark07,
                                                EstimatedMark08,
                                                EstimatedMark09,
                                                EstimatedMark10,
                                                EstimatedMark11,
                                                EstimatedMark12)
                                        values(
                                                recac. Seq,
                                recac. Year,
                                recac. ISIC,
                                recac. SurveyScope,
                                                recac. SelContinuousRespondent,
                              recac. RegistrationNo,
                              recac. CompanyTel,
                                recac. EstablishmentName,
                                                recac. ContinuousRespondent,
                                                recac. EstimatedMark01,
                                                recac. EstimatedMark02,
                                                recac. EstimatedMark03,
                                                recac. EstimatedMark04,
                                                recac. EstimatedMark05,
                                                recac. EstimatedMark06,
                                                recac. EstimatedMark07,
                                                recac. EstimatedMark08,
                                                recac. EstimatedMark09,
                                                recac. EstimatedMark10,
                                                recac. EstimatedMark11,
                                                recac. EstimatedMark12);

                        end if;
                end if;

                fetch   csrem
                        into    recem. ISIC, recem. RegistrationNo,
                        recem. EstablishmentName, recem. CompanyTel,
                        recem. ContinuousRespondent;
```

```
                end loop;

                close csrem;

        end;

-------------------------------------------------------
-- Insert CPM/CPY of the Raw by Establishment Table
-------------------------------------------------------
procedure p_InsCPMYRawEst(
        inseq                                                                   num
ber,
        inISIC                                                          varchar2,
        inYear                                                          number,
        inMonth                                                         number,
        inRegistrationNo                                varchar2,
        inOutputCnt                                                     number,
        inIndustryTypeShortName varchar2,
        inEstablishmentName                     varchar2,
        inContinuousRespondent  varchar2,
        inCommodityCodeWk                               varchar2,
        inDataTypeWK                                            varchar2,
        inQuestionnaireWK1                      varchar2,
        inQuestionnaireWK2                      varchar2,
        inQuestionnaireWK3                      varchar2,
        inQuestionnaireWK4                      varchar2,
        inQuestionnaireWK5                      varchar2,
        inQuestionnaireWK6                      varchar2,
        inQuestionnaireWK7                      varchar2) is

begin
        insert
                into    CPMYRawEst(
                        seq,
                ISIC,
                Year,
                Month,
                RegistrationNo,
                        OutputCnt,
                        IndustryTypeShortName,
                        EstablishmentName,
                        ContinuousRespondent,
                        CommodityCodeWk,
                        DataTypeWK,
                        QuestionnaireWK1,
                        QuestionnaireWK2,
                        QuestionnaireWK3,
                        QuestionnaireWK4,
                        QuestionnaireWK5,
                        QuestionnaireWK6,
                        QuestionnaireWK7)
                values(
                        inseq,
                inISIC,
                inYear,
                inMonth,
                inRegistrationNo,
                        inOutputCnt,
                        inIndustryTypeShortName,
                        inEstablishmentName,
                        inContinuousRespondent,
                        inCommodityCodeWk,
                        inDataTypeWK,
                        inQuestionnaireWK1,
                        inQuestionnaireWK2,
                        inQuestionnaireWK3,
                        inQuestionnaireWK4,
                        inQuestionnaireWK5,
                        inQuestionnaireWK6,
                        inQuestionnaireWK7);
end;

-------------------------------------------------------
-- Calculation CPM and CPY
-------------------------------------------------------
function f_calCP(
        inCur                                                           number,
```

```
        inPre                                                   number) return varc
har2 is

        sCPMY           varchar2(20);

begin
        if      inPre = 0
        or      inPre is null
        or      inCur = 0
        or      inCur is null then
                sCPMY := null;
        else
                sCPMY := to_char((inCur / inPre - 1) * 100,'99,990.00') || '%';
        end if;

        return sCPMY;

end;

-------------------------------------------------------
-- Format For Number Type
-------------------------------------------------------
function f_format(
        inNum           number) return varchar2 is

        sNum            varchar2(20);

begin
        if      inNum = 0 then
                sNum := '0.00';
        else
                sNum := to_char(inNum,'9,999,990.00');
        end if;

        return sNum;

end;

-------------------------------------------------------
-- Edit Continuous Respondent
-------------------------------------------------------
function f_EditCR(
        inContinuousRespondent  varchar2) return varchar2 is

        sContinuousRespondent           varchar2(1);

begin
        if      inContinuousRespondent = '1' then
                sContinuousRespondent := '*';
        else
                sContinuousRespondent := null;
        end if;

        return sContinuousRespondent;

end;

-------------------------------------------------------
-- CPM/CPY of the Raw Data Making by Establishment
-------------------------------------------------------
procedure p_CPMYRawEst (
        inSeq
                in      number,
        inISIC                                                          i
n       varchar2,
        inRegistrationNo                                        in      varchar2,
        inYear                                                          i
n       number,
        inMonth                                                         i
n       number,
        inSelContinuousRespondent       in      varchar2,
        inSurveyScope                                           in      varchar2) is

-- Record Area Define
        recsd           SurveyData%rowtype;
        recsdy  SurveyData%rowtype;
        recsdm  SurveyData%rowtype;
```

```
        reccp           CPMYRawEst%rowtype;
        recem           EstablishmentMaster%rowtype;
        reccl           ClassMaster%rowtype;
        reccm           CommodityMaster%rowtype;

-- Work Area
        nPMY    number;
        nPMM    number;

        nPYY    number;
        nPYM    number;

        sOldRegistrationNo      Varchar2(20) := null;
        sOldItemType                         Varchar2(1) := null;

-- Establishment Master Table Cursor
        Cursor  csrsd is
                select  *
                        from  SurveyData
                        where   ISIC    = inISIC
                                and     Year    = inYear
                                and     Month = inMonth
                                and     ItemType in('1','2','3','4','6')
                        order by RegistrationNo, ItemType, CommodityCode;

begin
-- DBMS_OUTPUT.ENABLE(600000);

-- PM and PY Calculation
        if      inMonth = 1 then
                nPMY    := inYear - 1;
                nPMM    := 12;
        else
                nPMY    := inYear;
                nPMM    := inMonth - 1;
        end if;

        nPYY    := inYear - 1;
        nPYM    := inMonth;

-- Select Class Master
        begin
                select  IndustryTypeShortName
                        into    reccl.IndustryTypeShortName
                        from    ClassMaster
                        where   ISIC = inISIC;

        exception
                when no_data_found then
     reccl.IndustryTypeShortName        := null;
        end;

-- Survey Data Cursor Open Fetch
        open    csrsd;
        fetch   csrsd
                into    recsd;

-- Survey Data Cursor Fetch Looping
        while csrsd%found loop
-- Input Parameter Checking(1)
                if      inRegistrationNo is null        then
                        null;
                else
                        if      inRegistrationNo = recsd.RegistrationNo then
                                null;
                        else
                                goto FetchNext;
                        end if;
                end if;

-- Set Output Counter
                if      recsd.RegistrationNo = sOldRegistrationNo then
                        null;
                else
                        reccp.OutputCnt := 0;
                        sOldRegistrationNo := recsd.RegistrationNo;
                        sOldItemType := null;
```

```
                        begin
                                select  *
                                        into    recem
                                        from    EstablishmentMaster
                                        where   ISIC = recsd.ISIC
                                                and     RegistrationNo = recsd.Registratio
nNo;

                        exception
                                when no_data_found then
                recem.EstablishmentName         := null;
                recem.ContinuousRespondent      := null;
                        end;

                end if;

-- Input Parameter Checking(2)
--      DBMS_OUTPUT.PUT_LINE('check start' || recsd.RegistrationNo);

                if      inRegistrationNo is null        then
                        if      inSelContinuousRespondent = '0' then
                                if      recem.ContinuousRespondent = '1' then
--      DBMS_OUTPUT.PUT_LINE('1 ' || recem.ContinuousRespondent);
                                        goto FetchNext;
                                else
--      DBMS_OUTPUT.PUT_LINE('0 ' || recem.ContinuousRespondent);
                                        null;
                                end if;
                        elsif   inSelContinuousRespondent = '1' then
                                if      recem.ContinuousRespondent = '1' then
                                        null;
                                else
                                        goto FetchNext;
                                end if;
                        elsif   inSelContinuousRespondent = '2' then
                                null;
                        end if;
                        if      inSurveyScope = recem.SurveyScope then
                                null;
                        else
                                goto FetchNext;
                        end if;
                end if;

-- Item Type Check
                if      recsd.ItemType = sOldItemType then
                        null;
                else
                        reccp.OutputCnt := reccp.OutputCnt + 1;
                        if      recsd.ItemType = '1' then
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                '1.Finished Goods(Quantity)',

                null,'BM_Inventory','Production','Receipt',

                'Domestic','Export','Others','ME_Inventory');
                        elsif recsd.ItemType = '2' then
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                '2.Finished Goods(Values)',

                null,'Shipment','Sales Plan',

                null, null, null, null, null);
                        elsif recsd.ItemType = '3' then
```

```
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                '3.Labor',

                null,'Labor Total','Labor SC',

                null,null,null,null,null);
                        elsif recsd.ItemType = '4' then
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                '4.Row Materials',

                null,'RwMtME_Inventory','RwMtME_InvValue',

                null,null,null,null,null);
                        elsif recsd.ItemType = '6' then
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                '6.Capacity',

                null,'Capacity',null,null,null,null,null,null);
                        end if;

                        sOldItemType := recsd.ItemType;

                end if;

-- Get CPM Data
                begin
                        select  *
                                into    recsdm
                                from    SurveyData
                                where   ISIC                                            = r
ecsd.ISIC
                                        and     ItemType                                =
recsd.ItemType
                                        and     CommodityCode           = recsd.CommodityCo
de
                                        and     RegistrationNo  = recsd.RegistrationNo
                                and     Year                                            = n
PMY
                                and     Month                                           = n
PMM;

                exception
                        when no_data_found then
        recsdm.BM_Inventory                                     := null;
        recsdm.ProductionQTY                                    := null;
        recsdm.Receipts                                                 := null;
        recsdm.DomesticSales                                    := null;
        recsdm.Export                                                           := null;
        recsdm.OtherSales                                               := null;
        recsdm.ME_Inventory                                     := null;
        recsdm.Capacity                                                 := null;
        recsdm.ShipmentValue                                    := null;
        recsdm.SalesPlan                                                := null;
        recsdm.LaborTotal                                               := null;
        recsdm.Labor_SC                                                 := null;
        recsdm.RwMt_MEInventory                         := null;
        recsdm.RwMt_MEInventoryValue    := null;
                end;
```

```
-- Get CPY Data
                begin
                        select  *
                                into    recsdy
                                from    SurveyData
                                where   ISIC                                    = r
ecsd.ISIC
                                        and     ItemType                        = r
ecsd.ItemType
                                        and     CommodityCode           = recsd.CommodityCo
de
                                        and     RegistrationNo  = recsd.RegistrationNo
                                and     Year                                    = n
PYY
                                and     Month                                   = n
PYM;

                exception
                        when no_data_found then
        recsdy.BM_Inventory                                     := null;
        recsdy.ProductionQTY                                    := null;
        recsdy.Receipts                                                 := null;
        recsdy.DomesticSales                                    := null;
        recsdy.Export                                                           := null;
        recsdy.OtherSales                                               := null;
        recsdy.ME_Inventory                                     := null;
        recsdy.Capacity                                                 := null;
        recsdy.ShipmentValue                                    := null;
        recsdy.SalesPlan                                                := null;
        recsdy.LaborTotal                                               := null;
        recsdy.Labor_SC                                                 := null;
        recsdy.RwMt_MEInventory                 := null;
        recsdy.RwMt_MEInventoryValue    := null;
                end;

-- Select Commodity Master
                begin
                        select  CommodityShortName
                                into    reccm.CommodityShortName
                                from    CommodityMaster
                                where   ISIC                                    = r
ecsd.ISIC
                                        and     ItemType                        = r
ecsd.ItemType
                                        and     CommodityCode           = recsd.CommodityCo
de;

                exception
                        when no_data_found then
            reccm.CommodityShortName    := null;
                end;

-- Item Type 1 Edit
                if      recsd.ItemType = '1' then
-- Check Input Data
                        if      recsd.BM_Inventory              is null
                        and     recsd.ProductionQty     is null
                        and     recsd.Receipts                          is null
                        and     recsd.DomesticSales             is null
                        and     recsd.Export                                    is null
                        and     recsd.OtherSales                        is null
                        and     recsd.ME_Inventory              is null
                        and     recsdm.BM_Inventory     is null
                        and     recsdm.ProductionQty    is null
                        and     recsdm.Receipts                         is null
                        and     recsdm.DomesticSales    is null
                        and     recsdm.Export                                   is null
                        and     recsdm.OtherSales                       is null
                        and     recsdm.ME_Inventory             is null
                        and     recsdy.BM_Inventory     is null
                        and     recsdy.ProductionQty    is null
                        and     recsdy.Receipts                         is null
                        and     recsdy.DomesticSales    is null
                        and     recsdy.Export                                   is null
                        and     recsdy.OtherSales                       is null
                        and     recsdy.ME_Inventory             is null then
                                null;
```

```
                    else
-- This Month Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,
            reccl.IndustryTypeShortName, recem.EstablishmentName,
            f_editCR(recem.ContinuousRespondent), .
            ' ' || recsd.CommodityCode || ' ' || substr(reccm.CommodityShortName, 1, 2
5), 'This Month',
            f_format(recsd.BM_Inventory), f_format(recsd.ProductionQty),
            f_format(recsd.Receipts), f_format(recsd.DomesticSales),
            f_format(recsd.Export), f_format(recsd.OtherSales),
            f_format(recsd.ME_Inventory));
-- Previous Month Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,
            reccl.IndustryTypeShortName, recem.EstablishmentName,
            f_editCR(recem.ContinuousRespondent),
            null, 'Pre Month',
            f_format(recsdm.BM_Inventory), f_format(recsdm.ProductionQty),
            f_format(recsdm.Receipts), f_format(recsdm.DomesticSales),
            f_format(recsdm.Export), f_format(recsdm.OtherSales),
            f_format(recsdm.ME_Inventory));
-- Previous Year Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,
            reccl.IndustryTypeShortName, recem.EstablishmentName,
            f_editCR(recem.ContinuousRespondent),
            null, 'Pre Year',
            f_format(recsdy.BM_Inventory), f_format(recsdy.ProductionQty),
            f_format(recsdy.Receipts), f_format(recsdy.DomesticSales),
            f_format(recsdy.Export), f_format(recsdy.OtherSales),
            f_format(recsdy.ME_Inventory));
-- CPM Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,
            reccl.IndustryTypeShortName, recem.EstablishmentName,
            f_editCR(recem.ContinuousRespondent),
            null, 'CPM(%)',
            f_CalCP(recsd.BM_Inventory, recsdm.BM_Inventory),
            f_CalCP(recsd.ProductionQty, recsdm.ProductionQty),
            f_CalCP(recsd.Receipts, recsdm.Receipts),
```

```
                f_CalCP(recsd.DomesticSales, recsdm.DomesticSales),

                f_CalCP(recsd.Export, recsdm.Export),

                f_CalCP(recsd.OtherSales, recsdm.OtherSales),

                f_CalCP(recsd.ME_Inventory, recsdm.ME_Inventory));
-- CPY Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null,'CPY(%)',

                f_CalCP(recsd.BM_Inventory, recsdy.BM_Inventory),

                f_CalCP(recsd.ProductionQty, recsdy.ProductionQty),

                f_CalCP(recsd.Receipts, recsdy.Receipts),

                f_CalCP(recsd.DomesticSales, recsdy.DomesticSales),

                f_CalCP(recsd.Export, recsdy.Export),

                f_CalCP(recsd.OtherSales, recsdy.OtherSales),

                f_CalCP(recsd.ME_Inventory, recsdy.ME_Inventory));
                        end if;
-- Item Type 2 Edit
                elsif   recsd.ItemType = '2' then
-- Check Input Data
                        if      recsd.ShipmentValue             is null
                        and     recsd.SalesPlan                         is null
                        and     recsdm.ShipmentValue    is null
                        and     recsdm.SalesPlan                        is null
                        and     recsdy.ShipmentValue    is null
                        and     recsdy.SalesPlan                        is null then
                                null;
                        else
-- This Month Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                ' ' || recsd.CommodityCode || ' ' || substr(reccm.CommodityShortName, 1, 2
5),'This Month',

                f_format(recsd.ShipmentValue), f_format(recsd.SalesPlan),

                null, null, null, null, null);
-- Previous Month Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null,'Pre Month',

                f_format(recsdm.ShipmentValue), f_format(recsdm.SalesPlan),

                null, null, null, null, null);
```

```
-- Previous Year Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null,'Pre Year',

                f_format(recsdy.ShipmentValue), f_format(recsdy.SalesPlan),

                null,null,null,null,null);

-- CPM Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null,'CPM(%)',

                f_CalCP(recsd.ShipmentValue, recsdm.ShipmentValue),

                f_CalCP(recsd.SalesPlan, recsdm.SalesPlan),

                null,null,null,null,null);

-- CPY Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null,'CPY(%)',

                f_CalCP(recsd.ShipmentValue, recsdy.ShipmentValue),

                f_CalCP(recsd.SalesPlan, recsdy.SalesPlan),

                null,null,null,null,null);
                        end if;

-- Item Type 3 Edit
                elsif   recsd.ItemType = '3' then
-- Check Input Data
                        if      recsd.LaborTotal        is null
                        and     recsd.Labor_SC          is null
                        and     recsdm.LaborTotal is null
                        and     recsdm.Labor_SC         is null
                        and     recsdy.LaborTotal is null
                        and     recsdy.Labor_SC         is null then
                                null;
                        else
-- This Month Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                ' ' || recsd.CommodityCode || ' ' || substr(reccm.CommodityShortName, 1, 2
5),'This Month',

                f_format(recsd.LaborTotal), f_format(recsd.Labor_SC),

                null,null,null,null,null);
```

```
-- Previous Month Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccI.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null,'Pre Month',

                f_format(recsdm.LaborTotal), f_format(recsdm.Labor_SC),

                null, null, null, null, null);

-- Previous Year Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccI.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null,'Pre Year',

                f_format(recsdy.LaborTotal), f_format(recsdy.Labor_SC),

                null, null, null, null, null);

-- CPM Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccI.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null,'CPM(%)',

                f_CalCP(recsd.LaborTotal, recsdm.LaborTotal),

                f_CalCP(recsd.Labor_SC, recsdm.Labor_SC),

                null, null, null, null, null);

-- CPY Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccI.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null,'CPY(%)',

                f_CalCP(recsd.LaborTotal, recsdy.LaborTotal),

                f_CalCP(recsd.Labor_SC, recsdy.Labor_SC),

                null, null, null, null, null);
                        end if;

-- Item Type 4 Edit
                elsif   recsd.ItemType = '4' then
-- Check Input Data
                        if      recsd.RwMt_MEInventory                          is null
                        and     recsd.RwMt_MEInventoryValue             is null
                        and     recsdm.RwMt_MEInventory                         is null
                        and     recsdm.RwMt_MEInventoryValue    is null
                        and     recsdy.RwMt_MEInventory                         is null
                        and     recsdy.RwMt_MEInventoryValue    is null then
                                null;
                        else
-- This Month Data Insert
```

```
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                ' ' || recsd.CommodityCode || ' ' || substr(reccm.CommodityShortName, 1, 2
5), 'This Month',

                f_format(recsd.RwMt_MEInventory),

                f_format(recsd.RwMt_MEInventoryValue),

                null, null, null, null, null);

-- Previous Month Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null, 'Pre Month',

                f_format(recsdm.RwMt_MEInventory),

                f_format(recsdm.RwMt_MEInventoryValue),

                null, null, null, null, null);

-- Previous Year Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null, 'Pre Year',

                f_format(recsdy.RwMt_MEInventory),

                f_format(recsdy.RwMt_MEInventoryValue),

                null, null, null, null, null);

-- CPM Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null, 'CPM(%)',

                f_CalCP(recsd.RwMt_MEInventory, recsdm.RwMt_MEInventory),

                f_CalCP(recsd.RwMt_MEInventoryValue, recsdm.RwMt_MEInventoryValue),

                null, null, null, null, null);

-- CPY Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                reccl.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),
```

```
                null,'CPY(%)',

                f_CalCP(recsd.RwMt_MEInventory,recsdy.RwMt_MEInventory),

                f_CalCP(recsd.RwMt_MEInventoryValue,recsdy.RwMt_MEInventoryValue),

                null,null,null,null,null);

                        end if;

-- Item Type 6 Edit
                elsif   recsd.ItemType = '6' then
-- Check Input Data
                        if      recsd.Capacity  is null
                        and     recsdm.Capacity is null
                        and     recsdy.Capacity is null then
                                null;
                        else
-- This Month Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq,inISIC,inYear,inMonth,recsd.Registra
tionNo,reccp.OutputCnt,

                reccl.IndustryTypeShortName,recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                ' ' || recsd.CommodityCode || ' ' || substr(reccm.CommodityShortName,1,2
5),'This Month',

                f_format(recsd.Capacity),

                null,null,null,null,null,null);

-- Previous Month Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq,inISIC,inYear,inMonth,recsd.Registra
tionNo,reccp.OutputCnt,

                reccl.IndustryTypeShortName,recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null,'Pre Month',

                f_format(recsdm.Capacity),

                null,null,null,null,null,null);

-- Previous Year Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq,inISIC,inYear,inMonth,recsd.Registra
tionNo,reccp.OutputCnt,

                reccl.IndustryTypeShortName,recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null,'Pre Year',

                f_format(recsdy.Capacity),

                null,null,null,null,null,null);

-- CPM Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq,inISIC,inYear,inMonth,recsd.Registra
tionNo,reccp.OutputCnt,

                reccl.IndustryTypeShortName,recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null,'CPM(%)',

                f_CalCP(recsd.Capacity,recsdm.Capacity),
```

```
                null, null, null, null, null, null);

-- CPY Data Insert
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                p_InsCPMYRawEst(inSeq, inISIC, inYear, inMonth, recsd.Registra
tionNo, reccp.OutputCnt,

                recci.IndustryTypeShortName, recem.EstablishmentName,

                f_editCR(recem.ContinuousRespondent),

                null, 'CPY(%)',

                f_CalCP(recsd.Capacity, recsdy.Capacity),

                null, null, null, null, null, null);
                        end if;

                end if;

                <<FetchNext>>

                fetch           csrsd
                        into    recsd;

        end loop;

        close csrsd;

end;

-------------------------------------------------------
-- Insert CPM/CPY of the Raw by ISIC Table
-------------------------------------------------------
procedure p_InsCPMYRawISIC(
        inseq                                                                   num
ber,
        inSurveyScope                                           varchar2,
        inISIC                                                          varchar2,
        inYear                                                          number,
        inMonth                                                         number,
        inOutputCnt                                                     number,
        inIndustryTypeShortName varchar2,
        inCommodityCodeWk                               varchar2,
        inDataTypeWK                                            varchar2,
        inQuestionnaireWK1                      varchar2,
        inQuestionnaireWK2                      varchar2,
        inQuestionnaireWK3                      varchar2,
        inQuestionnaireWK4                      varchar2,
        inQuestionnaireWK5                      varchar2,
        inQuestionnaireWK6                      varchar2,
        inQuestionnaireWK7                      varchar2) is

begin
-- DBMS_OUTPUT.PUT_LINE(inISIC || ' ' || inOutputCnt);

        insert
                into    CPMYRawISIC(
                        seq,
                        SurveyScope,
                ISIC,
                Year,
                Month,
                        OutputCnt,
                        IndustryTypeShortName,
                        CommodityCodeWk,
                        DataTypeWK,
                        QuestionnaireWK1,
                        QuestionnaireWK2,
                        QuestionnaireWK3,
                        QuestionnaireWK4,
                        QuestionnaireWK5,
                        QuestionnaireWK6,
                        QuestionnaireWK7)
                values(
                        inseq,
                        inSurveyScope,
```

```
                inISIC,
                inYear,
                inMonth,
                        inOutputCnt,
                        inIndustryTypeShortName,
                        inCommodityCodeWk,
                        inDataTypeWK,
                        inQuestionnaireWK1,
                        inQuestionnaireWK2,
                        inQuestionnaireWK3,
                        inQuestionnaireWK4,
                        inQuestionnaireWK5,
                        inQuestionnaireWK6,
                        inQuestionnaireWK7);
end;

-------------------------------------------------------
-- CPM/CPY of the Raw Data Making by ISIC
-------------------------------------------------------
procedure p_CPMYRawISIC (
        inSeq
                in      number,
        inAggregationRange                              in      varchar2,
        inISIC                                                                  i
n       varchar2,
        inYear                                                                  i
n       number,
        inMonth                                                                 i
n       number,
        inSelContinuousRespondent       in      varchar2,
        inSurveyScope                                                   in      varchar2) is

-- Record Area Define
        reccm           CommodityMaster%rowtype;
        recsd           SurveyData%rowtype;
        recsdy  SurveyData%rowtype;
        recsdm  SurveyData%rowtype;
        reccp           CPMYRawISIC%rowtype;
        reccl           ClassMaster%rowtype;

-- Work Area
        nPMY    number;
        nPMM    number;

        nPYY    number;
        nPYM    number;

        sOldISIC                                                        Varchar2(8) := null;
        sOldItemType                                    Varchar2(1) := null;

        sISIC2d                                                         Varchar2(2);
        sISIC4d                                                         Varchar2(4);

        sContinuousRespondent   varchar2(1);

-- Establishment Master Table Cursor
        Cursor  csrcm is
                select  *
                        from  CommodityMaster
                        where   ItemType in('1','2','3','4','6')
                        order by ISIC, ItemType, CommodityCode;

begin
-- ISIC Edit
        sISIC2d := substr(inISIC, 1, 2);
        sISIC4d := substr(inISIC, 1, 4);

-- PM and PY Calculation
        if      inMonth = 1 then
                nPMY    := inYear - 1;
                nPMM    := 12;
        else
                nPMY    := inYear;
                nPMM    := inMonth - 1;
        end if;

        nPYY    := inYear - 1;
```

```
        nPYM    := inMonth;

-- Edit Condition for ContinuousRespondent
        if      inSelContinuousRespondent = '0' then
                sContinuousRespondent := '0';
        elsif   inSelContinuousRespondent = '1' then
                sContinuousRespondent := '1';
        elsif   inSelContinuousRespondent = '2' then
                sContinuousRespondent := '%';
        end if;

-- Select Class Master
        begin
                select  IndustryTypeShortName
                        into    recci.IndustryTypeShortName
                        from    ClassMaster
                        where   ISIC = inISIC;

        exception
                when no_data_found then
     recci.IndustryTypeShortName        := null;
        end;

-- Survey Data Cursor Open Fetch
        open    csrcm;
        fetch   csrcm
                into    reccm;

-- Survey Data Cursor Fetch Looping
        while csrcm%found loop

-- Check Aggregation Range
                if      (inAggregationRange = '1')
                or      (inAggregationRange = '2' and substr(reccm.ISIC, 1, 2) = sISIC2d)
                or      (inAggregationRate = '3' and substr(reccm.ISIC, 1, 4) = sISIC4d)
                or      (inAggregationRange = '4' and substr(reccm.ISIC, 1, 6) = inISIC) then

-- Set Output Counter
                        if      reccm.ISIC = sOldISIC then
                                null;
                        else
                                reccp.OutputCnt := 0;
                                sOldISIC := reccm.ISIC;
                                sOldItemType := null;
                        end if;

-- Item Type Check
                        if      reccm.ItemType = sOldItemType then
                                null;
                        else
                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                if      reccm.ItemType = '1' then
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        recci.IndustryTypeShortName,

                        '1.Finished Goods(Quantity)',

                        null,'BM_Inventory','Production','Receipt',

                        'Domestic','Export','Others','ME_Inventory');
                                elsif reccm.ItemType = '2' then
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        recci.IndustryTypeShortName,

                        '2.Finished Goods(Values)',null,'Shipment','Sales Plan',

                        null,null,null,null,null);
                                elsif reccm.ItemType = '3' then
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        recci.IndustryTypeShortName,
```

```
                        '3. Labor',

                        null, 'Labor Total', 'Labor SC',

                        null, null, null, null, null);
                                elsif reccm. ItemType = '4' then
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm. ISIC, in
Year, inMonth, reccp. OutputCnt,

                        recci. IndustryTypeShortName,

                        '4. Row Materials',

                        null, 'RwMt ME_Inventory', 'RwMt ME_Inventory VI',

                        null, null, null, null, null);
                                elsif reccm. ItemType = '6' then
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm. ISIC, in
Year, inMonth, reccp. OutputCnt,

                        recci. IndustryTypeShortName,

                        '6. Capacity',

                        null, 'Capacity', null, null, null, null, null, null);
                                end if;

                                sOldItemType := reccm. ItemType;

                        end if;

-- Get Current Month Data
                        begin
                                select  sum(SurveyData. BM_Inventory),
                                        sum(SurveyData. ProductionQTY),
                                        sum(SurveyData. Receipts),
                                        sum(SurveyData. DomesticSales),
                                        sum(SurveyData. Export),
                                        sum(SurveyData. OtherSales),
                                        sum(SurveyData. ME_Inventory),
                                        sum(SurveyData. Capacity),
                                        sum(SurveyData. ShipmentValue),
                                        sum(SurveyData. SalesPlan),
                                        sum(SurveyData. LaborTotal),
                                        sum(SurveyData. Labor_SC),
                                        sum(SurveyData. RwMt_MEInventory),
                                        sum(SurveyData. RwMt_MEInventoryValue)
                                        into    recsd. BM_Inventory,
                                        recsd. ProductionQTY,
                                        recsd. Receipts,
                                        recsd. DomesticSales,
                                        recsd. Export,
                                        recsd. OtherSales,
                                        recsd. ME_Inventory,
                                        recsd. Capacity,
                                        recsd. ShipmentValue,
                                        recsd. SalesPlan,
                                        recsd. LaborTotal,
                                        recsd. Labor_SC,
                                        recsd. RwMt_MEInventory,
                                        recsd. RwMt_MEInventoryValue
                                        from    SurveyData, EstablishmentMaster
                                        where   SurveyData. ISIC
                        = EstablishmentMaster. ISIC
                                                and     SurveyData. RegistrationNo = Establ
ishmentMaster. RegistrationNo
                                                and     SurveyData. ISIC
                                = reccm. ISIC
                                                and     SurveyData. ItemType
                        =       reccm. ItemType
                                                and     SurveyData. CommodityCode       = r
eccm. CommodityCode
                                        and     Year
                                                        = inYear
                                        and     Month
                                                        = inMonth
                                        and     EstablishmentMaster. SurveyScope <= inSurve
```

```
yScope
                                        and     EstablishmentMaster.ContinuousRespondent l
ike sContinuousRespondent;

                        exception
                                when no_data_found then
                recsd.BM_Inventory                                              := null;
                recsd.ProductionQTY                                     := null;
                recsd.Receipts                                                          :=
null;
                recsd.DomesticSales                                     := null;
                recsd.Export
                 := null;
                recsd.OtherSales                                                := null;
                recsd.ME_Inventory                                              := null;
                recsd.Capacity                                                          :=
null;
                recsd.ShipmentValue                                     := null;
                recsd.SalesPlan                                                 := null;
                recsd.LaborTotal                                                := null;
                recsd.Labor_SC                                                          :=
null;
                recsd.RwMt_MEInventory                          := null;
                recsd.RwMt_MEInventoryValue     := null;
                        end;

-- Get CPM Data
                        begin
                                select  sum(SurveyData.BM_Inventory),
                                        sum(SurveyData.ProductionQTY),
                                        sum(SurveyData.Receipts),
                                        sum(SurveyData.DomesticSales),
                                        sum(SurveyData.Export),
                                        sum(SurveyData.OtherSales),
                                        sum(SurveyData.ME_Inventory),
                                        sum(SurveyData.Capacity),
                                        sum(SurveyData.ShipmentValue),
                                        sum(SurveyData.SalesPlan),
                                        sum(SurveyData.LaborTotal),
                                        sum(SurveyData.Labor_SC),
                                        sum(SurveyData.RwMt_MEInventory),
                                        sum(SurveyData.RwMt_MEInventoryValue)
                                        into    recsdm.BM_Inventory,
                                        recsdm.ProductionQTY,
                                        recsdm.Receipts,
                                        recsdm.DomesticSales,
                                        recsdm.Export,
                                        recsdm.OtherSales,
                                        recsdm.ME_Inventory,
                                        recsdm.Capacity,
                                        recsdm.ShipmentValue,
                                        recsdm.SalesPlan,
                                        recsdm.LaborTotal,
                                        recsdm.Labor_SC,
                                        recsdm.RwMt_MEInventory,
                                        recsdm.RwMt_MEInventoryValue
                                        from    SurveyData,EstablishmentMaster
                                        where   SurveyData.ISIC
                        = EstablishmentMaster.ISIC
                                                and     SurveyData.RegistrationNo = Establ
ishmentMaster.RegistrationNo
                                                and     SurveyData.ISIC
                                = reccm.ISIC
                                                and     SurveyData.ItemType
                        =       reccm.ItemType
                                                and     SurveyData.CommodityCode        = r
eccm.CommodityCode
                                        and     Year
                                                        = nPMY
                                        and     Month
                                                        = nPMM
                                        and     EstablishmentMaster.SurveyScope <= inSurve
yScope
                                        and     EstablishmentMaster.ContinuousRespondent l
ike sContinuousRespondent;

                        exception
```

```
                        when no_data_found then
                recsdm.BM_Inventory                                     := null;
                recsdm.ProductionQTY                                    := null;
                recsdm.Receipts                                                 := null;
                recsdm.DomesticSales                                    := null;
                recsdm.Export                                                           :=
null;
                recsdm.OtherSales                                               := null;
                recsdm.ME_Inventory                                     := null;
                recsdm.Capacity                                                 := null;
                recsdm.ShipmentValue                                    := null;
                recsdm.SalesPlan                                                := null;
                recsdm.LaborTotal                                               := null;
                recsdm.Labor_SC                                                 := null;
                recsdm.RwMt_MEInventory                 := null;
                recsdm.RwMt_MEInventoryValue    := null;
                        end;

-- Get CPY Data
                        begin
                                select  sum(SurveyData.BM_Inventory),
                                        sum(SurveyData.ProductionQTY),
                                        sum(SurveyData.Receipts),
                                        sum(SurveyData.DomesticSales),
                                        sum(SurveyData.Export),
                                        sum(SurveyData.OtherSales),
                                        sum(SurveyData.ME_Inventory),
                                        sum(SurveyData.Capacity),
                                        sum(SurveyData.ShipmentValue),
                                        sum(SurveyData.SalesPlan),
                                        sum(SurveyData.LaborTotal),
                                        sum(SurveyData.Labor_SC),
                                        sum(SurveyData.RwMt_MEInventory),
                                        sum(SurveyData.RwMt_MEInventoryValue)
                                        into    recsdy.BM_Inventory,
                                        recsdy.ProductionQTY,
                                        recsdy.Receipts,
                                        recsdy.DomesticSales,
                                        recsdy.Export,
                                        recsdy.OtherSales,
                                        recsdy.ME_Inventory,
                                        recsdy.Capacity,
                                        recsdy.ShipmentValue,
                                        recsdy.SalesPlan,
                                        recsdy.LaborTotal,
                                        recsdy.Labor_SC,
                                        recsdy.RwMt_MEInventory,
                                        recsdy.RwMt_MEInventoryValue
                                        from    SurveyData,EstablishmentMaster
                                        where   SurveyData.ISIC
                        = EstablishmentMaster.ISIC
                                                and     SurveyData.RegistrationNo = Establ
ishmentMaster.RegistrationNo
                                                and     SurveyData.ISIC
                                = reccm.ISIC
                                                and     SurveyData.ItemType
                        =       reccm.ItemType
                                                and     SurveyData.CommodityCode        = r
eccm.CommodityCode
                                        and     Year
                                                        = nPYY
                                        and     Month
                                                        = nPYM
                                        and     EstablishmentMaster.SurveyScope <= inSurve
yScope
                                        and     EstablishmentMaster.ContinuousRespondent l
ike sContinuousRespondent;

                        exception
                                when no_data_found then
                recsdy.BM_Inventory                                     := null;
                recsdy.ProductionQTY                                    := null;
                recsdy.Receipts                                                 := null;
                recsdy.DomesticSales                                    := null;
                recsdy.Export                                                           :=
null;
                recsdy.OtherSales                                               := null;
```

```
                recsdy.ME_Inventory                                     := null;
                recsdy.Capacity                                                 := null;
                recsdy.ShipmentValue                                    := null;
                recsdy.SalesPlan                                                := null;
                recsdy.LaborTotal                                               := null;
                recsdy.Labor_SC                                                 := null;
                recsdy.RwMt_MEInventory                 := null;
                recsdy.RwMt_MEInventoryValue    := null;
                        end;

-- Item Type 1 Edit
                        if      reccm.ItemType = '1' then
-- Check Input Data
                                if      recsd.BM_Inventory              is null
                                and     recsd.ProductionQty     is null
                                and     recsd.Receipts                          is null
                                and     recsd.DomesticSales             is null
                                and     recsd.Export                                            is
null
                                and     recsd.OtherSales                        is null
                                and     recsd.ME_Inventory              is null
                                and     recsdm.BM_Inventory     is null
                                and     recsdm.ProductionQty    is null
                                and     recsdm.Receipts                         is null
                                and     recsdm.DomesticSales    is null
                                and     recsdm.Export                                           is
null
                                and     recsdm.OtherSales                       is null
                                and     recsdm.ME_Inventory             is null
                                and     recsdy.BM_Inventory     is null
                                and     recsdy.ProductionQty    is null
                                and     recsdy.Receipts                         is null
                                and     recsdy.DomesticSales    is null
                                and     recsdy.Export                                           is
null
                                and     recsdy.OtherSales                       is null
                                and     recsdy.ME_Inventory             is null then
                                        null;
                                else
-- This Month Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        ' ' || reccm.CommodityCode || ' ' || substr(reccm.CommodityShortNa
me, 1, 25), 'This Month',

                        f_format(recsd.BM_Inventory), f_format(recsd.ProductionQty),

                        f_format(recsd.Receipts), f_format(recsd.DomesticSales),

                        f_format(recsd.Export), f_format(recsd.OtherSales),

                        f_format(recsd.ME_Inventory));

-- Previous Month Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null, 'Pre Month',

                        f_format(recsdm.BM_Inventory), f_format(recsdm.ProductionQty),

                        f_format(recsdm.Receipts), f_format(recsdm.DomesticSales),

                        f_format(recsdm.Export), f_format(recsdm.OtherSales),

                        f_format(recsdm.ME_Inventory));

-- Previous Year Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
```

```
Year, inMonth, reccp. OutputCnt,
                        reccl. IndustryTypeShortName,
                        null, 'Pre Year',
                        f_format(recsdy. BM_Inventory), f_format(recsdy. ProductionQty),
                        f_format(recsdy. Receipts), f_format(recsdy. DomesticSales),
                        f_format(recsdy. Export), f_format(recsdy. OtherSales),
                        f_format(recsdy. ME_Inventory));
-- CPM Data Insert
                                        reccp. OutputCnt := reccp. OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm. ISIC, in
Year, inMonth, reccp. OutputCnt,
                        reccl. IndustryTypeShortName,
                        null, 'CPM(%)',
                        f_CalCP(recsd. BM_Inventory, recsdm. BM_Inventory),
                        f_CalCP(recsd. ProductionQty, recsdm. ProductionQty),
                        f_CalCP(recsd. Receipts, recsdm. Receipts),
                        f_CalCP(recsd. DomesticSales, recsdm. DomesticSales),
                        f_CalCP(recsd. Export, recsdm. Export),
                        f_CalCP(recsd. OtherSales, recsdm. OtherSales),
                        f_CalCP(recsd. ME_Inventory, recsdm. ME_Inventory));
-- CPY Data Insert
                                        reccp. OutputCnt := reccp. OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm. ISIC, in
Year, inMonth, reccp. OutputCnt,
                        reccl. IndustryTypeShortName,
                        null, 'CPY(%)',
                        f_CalCP(recsd. BM_Inventory, recsdy. BM_Inventory),
                        f_CalCP(recsd. ProductionQty, recsdy. ProductionQty),
                        f_CalCP(recsd. Receipts, recsdy. Receipts),
                        f_CalCP(recsd. DomesticSales, recsdy. DomesticSales),
                        f_CalCP(recsd. Export, recsdy. Export),
                        f_CalCP(recsd. OtherSales, recsdy. OtherSales),
                        f_CalCP(recsd. ME_Inventory, recsdy. ME_Inventory));
                                end if;
-- Item Type 2 Edit
                        elsif   reccm. ItemType = '2' then
-- Check Input Data
                                if      recsd. ShipmentValue            is null
                                and     recsd. SalesPlan                        is null
                                and     recsdm. ShipmentValue   is null
                                and     recsdm. SalesPlan                       is null
                                and     recsdy. ShipmentValue   is null
                                and     recsdy. SalesPlan                       is null then
                                        null;
                                else
-- This Month Data Insert
                                        reccp. OutputCnt := reccp. OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm. ISIC, in
Year, inMonth, reccp. OutputCnt,
```

```
                        reccl.IndustryTypeShortName,

                        ' ' || reccm.CommodityCode || ' ' || substr(reccm.CommodityShortNa
me,1,25),'This Month',

                        f_format(recsd.ShipmentValue),f_format(recsd.SalesPlan),

                        null,null,null,null,null);

-- Previous Month Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq,inSurveyScope,reccm.ISIC,in
Year,inMonth,reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null,'Pre Month',

                        f_format(recsdm.ShipmentValue),f_format(recsdm.SalesPlan),

                        null,null,null,null,null);

-- Previous Year Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq,inSurveyScope,reccm.ISIC,in
Year,inMonth,reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null,'Pre Year',

                        f_format(recsdy.ShipmentValue),f_format(recsdy.SalesPlan),

                        null,null,null,null,null);

-- CPM Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq,inSurveyScope,reccm.ISIC,in
Year,inMonth,reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null,'CPM(%)',

                        f_CalCP(recsd.ShipmentValue,recsdm.ShipmentValue),

                        f_CalCP(recsd.SalesPlan,recsdm.SalesPlan),

                        null,null,null,null,null);

-- CPY Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq,inSurveyScope,reccm.ISIC,in
Year,inMonth,reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null,'CPY(%)',

                        f_CalCP(recsd.ShipmentValue,recsdy.ShipmentValue),

                        f_CalCP(recsd.SalesPlan,recsdy.SalesPlan),

                        null,null,null,null,null);
                                end if;

-- Item Type 3 Edit
                        elsif   reccm.ItemType = '3' then
-- Check Input Data
                                if      recsd.LaborTotal        is null
                                and     recsd.Labor_SC          is null
                                and     recsdm.LaborTotal is null
                                and     recsdm.Labor_SC         is null
                                and     recsdy.LaborTotal is null
                                and     recsdy.Labor_SC         is null then
                                        null;
```

```
                                        else
-- This Month Data Insert
                                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                                p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        ' ' || reccm.CommodityCode || ' ' || substr(reccm.CommodityShortNa
me, 1, 25), 'This Month',

                        f_format(recsd.LaborTotal), f_format(recsd.Labor_SC),

                        null, null, null, null, null);

-- Previous Month Data Insert
                                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                                p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null, 'Pre Month',

                        f_format(recsdm.LaborTotal), f_format(recsdm.Labor_SC),

                        null, null, null, null, null);

-- Previous Year Data Insert
                                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                                p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null, 'Pre Year',

                        f_format(recsdy.LaborTotal), f_format(recsdy.Labor_SC),

                        null, null, null, null, null);

-- CPM Data Insert
                                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                                p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null, 'CPM(%)',

                        f_CalCP(recsd.LaborTotal, recsdm.LaborTotal),

                        f_CalCP(recsd.Labor_SC, recsdm.Labor_SC),

                        null, null, null, null, null);

-- CPY Data Insert
                                                reccp.OutputCnt := reccp.OutputCnt + 1;
                                                p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null, 'CPY(%)',

                        f_CalCP(recsd.LaborTotal, recsdy.LaborTotal),

                        f_CalCP(recsd.Labor_SC, recsdy.Labor_SC),

                        null, null, null, null, null);

                                        end if;

-- Item Type 4 Edit
                        elsif   reccm.ItemType = '4' then
                                        if      recsd.RwMt_MEInventory                          is
null
```

```
                                and     recsd.RwMt_MEInventoryValue             is null
                                and     recsdm.RwMt_MEInventory                         is
null
                                and     recsdm.RwMt_MEInventoryValue    is null
                                and     recsdy.RwMt_MEInventory                         is
null
                                and     recsdy.RwMt_MEInventoryValue    is null then
                                        null;
                                else
-- This Month Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        recci.IndustryTypeShortName,

                        ' ' || reccm.CommodityCode || ' ' || substr(reccm.CommodityShortNa
me, 1, 25), 'This Month',

                        f_format(recsd.RwMt_MEInventory), f_format(recsd.RwMt_MEInventoryVa
lue),

                        null, null, null, null, null);

-- Previous Month Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        recci.IndustryTypeShortName,

                        null, 'Pre Month',

                        f_format(recsdm.RwMt_MEInventory), f_format(recsdm.RwMt_MEInventory
Value),

                        null, null, null, null, null);

-- Previous Year Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        recci.IndustryTypeShortName,

                        null, 'Pre Year',

                        f_format(recsdy.RwMt_MEInventory), f_format(recsdy.RwMt_MEInventory
Value),

                        null, null, null, null, null);

-- CPM Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        recci.IndustryTypeShortName,

                        null, 'CPM(%)',

                        f_CalCP(recsd.RwMt_MEInventory, recsdm.RwMt_MEInventory),

                        f_CalCP(recsd.RwMt_MEInventoryValue, recsdm.RwMt_MEInventoryValue),

                        null, null, null, null, null);

-- CPY Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        recci.IndustryTypeShortName,

                        null, 'CPY(%)',

                        f_CalCP(recsd.RwMt_MEInventory, recsdy.RwMt_MEInventory),
```

```
                        f_CalCP(recsd.RwMt_MEInventoryValue, recsdy.RwMt_MEInventoryValue),

                        null, null, null, null, null);
                                end if;

-- Item Type 6 Edit
                        elsif   reccm.ItemType = '6' then
-- Check Input Data
                                if      recsd.Capacity  is null
                                and     recsdm.Capacity is null
                                and     recsdy.Capacity is null then
                                        null;
                                else
-- This Month Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        ' ' || reccm.CommodityCode || ' ' || substr(reccm.CommodityShortNa
me, 1, 25), 'This Month',

                        f_format(recsd.Capacity),

                        null, null, null, null, null, null);

-- Previous Month Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null, 'Pre Month',

                        f_format(recsdm.Capacity),

                        null, null, null, null, null, null);

-- Previous Year Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null, 'Pre Year',

                        f_format(recsdy.Capacity),

                        null, null, null, null, null, null);

-- CPM Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null, 'CPM(%)',

                        f_CalCP(recsd.Capacity, recsdm.Capacity),

                        null, null, null, null, null, null);

-- CPY Data Insert
                                        reccp.OutputCnt := reccp.OutputCnt + 1;
                                        p_InsCPMYRawISIC(inSeq, inSurveyScope, reccm.ISIC, in
Year, inMonth, reccp.OutputCnt,

                        reccl.IndustryTypeShortName,

                        null, 'CPY(%)',

                        f_CalCP(recsd.Capacity, recsdy.Capacity),
```

```
                        null, null, null, null, null, null);
                                end if;

                        end if;

                end if;

                fetch           csrcm
                        into    reccm;

        end loop;

        close csrcm;

end;

end UserRep;
/
show errors
/
```

```
/*
 * Package Name : Weight
 * Created Date : 18/NOV/99
 * Updated Date : 22/FEB/00
 *          By  K.Shibamoto@JICA SYUDY TEAM
 * NOTE : Calculate integrated weight for each level.
 * TEST : exec Weight.CreateWeightView;
 * Make : @c:¥shiba¥weight
 */

CREATE OR REPLACE PACKAGE Weight IS

    PROCEDURE CreateWeightView;
-- ------------
END;
/
show errors

CREATE OR REPLACE PACKAGE BODY Weight IS
    /* CreateWeightView
         This procedure create weight view for
         each AggregationRange and ISIC .
     */
    PROCEDURE CreateWeightView
    IS
    BEGIN
        DELETE FROM WeightAggregation ;
        COMMIT WORK;

        /* Weight for Aggregation Range 5 == PR1, SP2, IV3, IR4, LP6 == */
        INSERT INTO WeightAggregation
        SELECT '5'                                                  ,-- AggregationRange,
               W.ISIC                                               ,-- ISIC,
               W.ItemType                                           ,-- ItemType,
               W.CommodityCode                                      ,-- CommodityCode,
               W.SurveyScope                                        ,-- SurveyScope
               W.Version                                            ,-- Version
               nvl(WeightPR,0) * decode(IndexScopePR1,'1',1,0)      ,-- WeightPR1
               nvl(WeightSP,0) * decode(IndexScopeSP2,'1',1,0)      ,-- WeightSP2
               nvl(WeightIV,0) * decode(IndexScopeIV3,'1',1,0)      ,-- WeightIV3
               nvl(WeightIV,0) * decode(IndexScopeIR4,'1',1,0)      ,-- WeightIR4
               0                                                    ,-- WeightCU5
               nvl(WeightPR,0) * decode(IndexScopeLP6,'1',1,0)      ,-- WeightLP6
               0/* Replace here appropriate data */                 ,-- WeightLI7
               0/* Replace here appropriate data */                 ,-- WeightRI8
               W.ISIC                                                -- IntegratedKey
        FROM WeightMaster W , CommodityMaster C
        WHERE W.ISIC = C.ISIC
          AND W.Itemtype = '1'  -- Caution!!!
          AND W.ItemType = C.ItemType
          AND W.CommodityCode = C.CommodityCode
          AND W.Version = '1';

        COMMIT;

        /* Weight for Aggregation Range 5 == CU5 == */
        INSERT INTO WeightAggregation
        SELECT '5'                                                  ,-- AggregationRange,
               C6.ISIC                                              ,-- ISIC,
               C6.ItemType                                          ,-- ItemType,
               C6.CommodityCode                                     ,-- CommodityCode,
               W.SurveyScope                                        ,-- SurveyScope
               W.Version                                            ,-- Version
               0                                                    ,-- WeightPR1
               0                                                    ,-- WeightSP2
               0                                                    ,-- WeightIV3
               0                                                    ,-- WeightIR4
               SUM(nvl(W.WeightPR,0) * decode(C6.IndexScopeCU5,'1',1,0)) ,-- WeightCU5
               0                                                    ,-- WeightLP6
               0                                                    ,-- WeightLI7
               0                                                    ,-- WeightRI8
               C6.ISIC                                               -- IntegratedKey
        FROM WeightMaster W , CommodityMaster C1, CommodityMaster C6
        WHERE C6.Itemtype = '6'
          AND C6.ISIC = C1.ISIC
          AND C1.Itemtype = '1'
          AND C1.CapacityCode = C6.CommodityCode
```

```
  AND W.ISIC = C1.ISIC
  AND W.ItemType = C1.ItemType
  AND W.CommodityCode = C1.CommodityCode
  AND W.Version = '1'
Group By C6.ISIC, C6.ItemType , C6.CommodityCode, W.SurveyScope, W.Version ;

COMMIT;

/* Weight for Aggregation Range 4 */
INSERT INTO WeightAggregation
SELECT '4'              ,-- AggregationRange,
       IntegratedKey    ,-- ISIC,
       'X'              ,-- ItemType,
       'XXX'            ,-- CommodityCode,
       SurveyScope      ,-- SurveyScope
       Version          ,-- Version
       SUM(WeightPR1)   ,-- WeightPR1
       SUM(WeightSP2)   ,-- WeightSP2
       SUM(WeightIV3)   ,-- WeightIV3
       SUM(WeightIR4)   ,-- WeightIR4
       SUM(WeightCU5)   ,-- WeightCU5
       SUM(WeightLP6)   ,-- WeightLP6
       SUM(WeightLI7)   ,-- WeightLI7
       SUM(WeightRI8)   ,-- WeightRI8
       SUBSTR(IntegratedKey,1,4)||'XX'   -- IntegratedKey
FROM WeightAggregation
WHERE AggregationRange = '5'
GROUP BY IntegratedKey, SurveyScope, Version ;
COMMIT;

/* Weight for Aggregation Range 3 */
INSERT INTO WeightAggregation
SELECT '3'              ,-- AggregationRange,
       IntegratedKey    ,-- ISIC,
       'X'              ,-- ItemType,
       'XXX'            ,-- CommodityCode,
       SurveyScope      ,-- SurveyScope
       Version          ,-- Version
       SUM(WeightPR1)   ,-- WeightPR1
       SUM(WeightSP2)   ,-- WeightSP2
       SUM(WeightIV3)   ,-- WeightIV3
       SUM(WeightIR4)   ,-- WeightIR4
       SUM(WeightCU5)   ,-- WeightCU5
       SUM(WeightLP6)   ,-- WeightLP6
       SUM(WeightLI7)   ,-- WeightLI7
       SUM(WeightRI8)   ,-- WeightRI8
       SUBSTR(IntegratedKey,1,2)||'XXXX'-- IntegratedKey
FROM WeightAggregation
WHERE AggregationRange = '4'
GROUP BY IntegratedKey, SurveyScope, Version;
COMMIT;

/* Weight for Aggregation Range 2 */
INSERT INTO WeightAggregation
SELECT '2'              ,-- AggregationRange,
       IntegratedKey    ,-- ISIC,
       'X'              ,-- ItemType,
       'XXX'            ,-- CommodityCode,
       SurveyScope      ,-- SurveyScope
       Version          ,-- Version
       SUM(WeightPR1)   ,-- WeightPR1
       SUM(WeightSP2)   ,-- WeightSP2
       SUM(WeightIV3)   ,-- WeightIV3
       SUM(WeightIR4)   ,-- WeightIR4
       SUM(WeightCU5)   ,-- WeightCU5
       SUM(WeightLP6)   ,-- WeightLP6
       SUM(WeightLI7)   ,-- WeightLI7
       SUM(WeightRI8)   ,-- WeightRI8
       'XXXXXX'          -- IntegratedKey
FROM WeightAggregation
WHERE AggregationRange = '3'
GROUP BY IntegratedKey, SurveyScope, Version;
COMMIT;

/* Weight for Aggregation Range 1 */
INSERT INTO WeightAggregation
SELECT '1'              ,-- AggregationRange,
```

```
            IntegratedKey  ,-- ISIC,
            'X'            ,-- ItemType,
            'XXX'          ,-- CommodityCode,
            SurveyScope    ,-- SurveyScope
            Version        ,-- Version
            SUM(WeightPR1) ,-- WeightPR1
            SUM(WeightSP2) ,-- WeightSP2
            SUM(WeightIV3) ,-- WeightIV3
            SUM(WeightIR4) ,-- WeightIR4
            SUM(WeightCU5) ,-- WeightCU5
            SUM(WeightLP6) ,-- WeightLP6
            SUM(WeightLI7) ,-- WeightLI7
            SUM(WeightRI8) ,-- WeightRI8
            NULL            -- IntegratedKey
        FROM WeightAggregation
        WHERE AggregationRange = '2'
        GROUP BY IntegratedKey, SurveyScope, Version;
        COMMIT;

        EXCEPTION
            WHEN OTHERS THEN
                ROLLBACK;
        END;

    END;
/
show errors
```

```
-- ==================================================
-- Package Name : Yearly Proc
-- Package ID   : YearProc
-- Create Date  : 25/Feb/2000
-- Replace Date : xx/xxx/xxxx
-- Author       : Nakamura
--
-- File         : @c:¥jica¥YearProc;
-- Test                                :       BEGIN YearProc.p_AnnualCal(1999); Commit;
END;
-- test                                :       begin YearProc.p_BatchCtlUpd(1999, 1); comm
it; end;
-- ==================================================
create or replace package YearProc is

procedure p_AnnualCal (
        inYear                                                  in      number);

procedure p_BatchCtlUpd (
        inYear                                  in      Number,
        inMonth                                 in      Number);

end YearProc;
/
show errors
/

create or replace package body YearProc is
--------------------------------------------------------
-- Annual Indices Calculation
--------------------------------------------------------
procedure p_AnnualCal (
        inYear                                                  in      number) is

begin
-- All Annual Indices File Delete
        delete
                from    AnnualIndices
                where   Year = inYear;

-- Insert Annual Indices
        Insert
                into    AnnualIndices(
                                        AGGREGATIONRANGE,
                                        ISIC,
                                        ITEMTYPE,
                                        INDEXCOMMODITYCODE,
                                        YEAR,
                                        SURVEYSCOPE,
                                        INDEXPR1,
                                        INDEXSP2,
                                        INDEXIV3,
                                        INDEXIR4,
                                        INDEXCU5,
                                        INDEXLP6,
                                        INDEXLI7,
                                        INDEXRI8,
                                        RECDATE)
                select  AGGREGATIONRANGE,
                                        ISIC,
                                        ITEMTYPE,
                                        INDEXCOMMODITYCODE,
                                        inYear,
                                        SURVEYSCOPE,
                                        avg(INDEXPR1),
                                        avg(INDEXSP2),
                                        avg(INDEXIV3),
                                        avg(INDEXIR4),
                                        avg(INDEXCU5),
                                        avg(INDEXLP6),
                                        avg(INDEXLI7),
                                        avg(INDEXRI8),
                                        sysdate
                        from    indices
                        where   Year = inYear
                        and             Pre_Rev_Mark = '2'
                        group by AggregationRange, ISIC, ItemType, IndexCommodityCode, SurveyS
```

```
cope;

end;

-------------------------------------------------------
-- Batch Control File Update
-------------------------------------------------------
procedure p_BatchCtlUpd (
        inYear                                  in      Number,
        inMonth                                 in      Number) is

Begin
        update  BatchControl
                set             YearlyStatus    = '0',
                                        YearlyDate      = sysdate
                where   Year    = inYear
                        and     Month   = inMonth;

end;

end YearProc;
/
show errors
/
```

Excel Macro Module

Code	Name	YEAR	MONTH	INDEXPR1	INDEXSP2	INDEXIV3	INDEXIR4	INDEXCU5	INDEXLP6	INDEXLI7	INDEXRI8
15	Food&Bev	1999	2	100.9667	113.3893456	101.0299	90.72503	102.1822	107.3845	0	0
15	Food&Bev	1999	3	135.7276	142.9845514	121.468	81.39177	134.1613	122.3704	0	0
15	Food&Bev	1999	4	121.9903	136.356219	128.9539	92.01177	117.1153	126.487	0	0
15	Food&Bev	1999	5	117.8782	112.3317889	168.6285	142.3927	120.49	115.5639	0	0
15	Food&Bev	1999	6	110.2448	117.5258258	180.9476	149.7605	113.0372	103.0992	0	0
15	Food&Bev	1999	7	101.4882	116.1695135	170.5791	144.1565	106.6476	86.67588	0	0
15	Food&Bev	1999	8	103.2028	120.3268425	155.7912	126.9087	108.0978	84.42898	0	0
15	Food&Bev	1999	9	108.6906	128.1171684	169.59	128.062	119.3772	89.07267	0	0
15	Food&Bev	1999	10	104.2039	131.8370986	147.5084	108.8869	115.8308	93.14061	0	0
15	Food&Bev	1999	11	124.5578	147.5936105	134.9618	90.30466	133.4566	105.3496	0	0
15	Food&Bev	1999	12	122.3302	133.5188	145.77	110.6823	122.1513	109.2766	0	0
15	Food&Bev	2000	1	106.4024	120.6365973	157.5407	129.4458	108.1604	93.60368	0	0
17	Textile	1999	2	100.3783	112.8155847	92.98155	85.90725	96.90853	102.5326	0	0
17	Textile	1999	3	115.0961	130.5720148	87.92558	65.82741	105.6536	111.2238	0	0
17	Textile	1999	4	110.3612	101.3145371	67.64567	59.68298	96.56969	96.70571	0	0
17	Textile	1999	5	115.4747	142.3963029	60.14205	48.05489	102.2675	104.4049	0	0
17	Textile	1999	6	124.1412	137.427165	65.93492	48.20764	105.4673	112.8671	0	0
17	Textile	1999	7	124.8609	132.0540879	80.41822	59.81318	107.1159	116.0119	0	0
17	Textile	1999	8	120.2224	131.6778498	76.84322	72.32119	102.0767	111.6573	0	0
17	Textile	1999	9	128.0765	153.8496691	77.79278	46.40889	106.6944	119.0662	0	0
17	Textile	1999	10	125.6209	152.1924867	74.36313	51.33424	102.4939	115.1381	0	0
17	Textile	1999	11	130.8404	144.7043068	61.16484	62.0048	105.2911	121.1509	0	0
17	Textile	1999	12	120.6615	129.3160293	74.24566	70.98847	103.0955	112.1599	0	0
17	Textile	2000	1	122.6054	135.5627409	65.27772	53.5816	110.4557	99.75274	0	0
18	Apparel	1999	2	94.16097	92.67743315	98.92345	107.1386	94.29606	96.9601	0	0
18	Apparel	1999	3	95.03271	83.22301488	130.4894	157.0807	95.55373	91.84449	0	0
18	Apparel	1999	4	86.41644	93.72502046	119.1489	123.2773	96.78816	92.40544	0	0
18	Apparel	1999	5	104.3236	103.1396323	128.2777	125.9683	113.2302	76.86318	0	0
18	Apparel	1999	6	104.5253	99.87859129	137.0575	139.6646	107.4224	71.94024	0	0
18	Apparel	1999	7	111.371	115.2569922	145.3985	129.1481	104.4623	73.77557	0	0
18	Apparel	1999	8	108.3039	112.0117242	144.9316	135.6672	106.4904	79.16016	0	0
18	Apparel	1999	9	101.4003	105.8655634	147.54	139.4643	102.3013	67.52772	0	0
18	Apparel	1999	10	118.1297	120.9760965	157.7609	130.8712	106.1038	93.94679	0	0
18	Apparel	1999	11	118.4341	128.6152318	147.1903	114.8313	105.5691	86.38977	0	0
18	Apparel	1999	12	119.0365	133.2371458	135.9295	101.6831	116.0418	93.7825	0	0
18	Apparel	2000	1	116.8165	113.4127863	131.5289	121.074	119.1759	97.52076	0	0
23	Coke and	1999	2	95.50237	88.47006132	115.5884	134.2213	0	98.81748	0	0
23	Coke and	1999	3	107.259	106.2594316	104.6391	96.93655	0	105.9913	0	0
23	Coke and	1999	4	112.5579	96.90076382	118.7928	120.3008	0	146.6567	0	0
23	Coke and	1999	5	109.2282	91.39455266	134.595	151.6959	0	128.4716	0	0
23	Coke and	1999	6	105.4855	88.49856249	139.6602	159.7563	0	106.0439	0	0
23	Coke and	1999	7	101.4396	90.03541655	128.5967	144.088	0	106.0425	0	0
23	Coke and	1999	8	102.4648	98.12616475	112.1842	113.5618	0	105.9717	0	0
23	Coke and	1999	9	97.11693	88.83802022	119.3041	134.6896	0	98.08047	0	0
23	Coke and	1999	10	99.56008	86.2256634	128.0966	150.8993	0	104.7791	0	0
23	Coke and	1999	11	100.6498	92.07062587	136.3954	150.9458	0	98.66778	0	0
23	Coke and	1999	12	93.11092	91.91689618	109.8591	122.1935	0	90.81831	0	0
23	Coke and	2000	1	101.1471	85.17067945	115.5218	137.4358	0	109.4787	0	0
26	Other Non	1999	2	97.17187	103.5101542	97.20125	101.3753	96.08353	104.2245	0	0
26	Other Non	1999	3	109.0264	118.7376699	89.83939	76.1505	105.7624	103.0922	0	0
26	Other Non	1999	4	94.04585	89.82605234	93.57711	102.674	90.5044	104.0757	0	0
26	Other Non	1999	5	103.6384	111.3322978	82.80894	72.41179	98.5217	112.5033	0	0
26	Other Non	1999	6	113.2726	112.0837938	76.57927	68.48892	106.3602	112.0233	0	0
26	Other Non	1999	7	113.8977	120.9032954	71.4608	60.58155	103.1171	113.466	0	0
26	Other Non	1999	8	114.1928	119.1528075	77.45944	73.91183	101.7166	115.2989	0	0
26	Other Non	1999	9	111.637	112.9255444	73.70835	75.70076	101.0683	110.0048	0	0
26	Other Non	1999	10	97.73744	86.42638993	94.17058	117.8249	88.37395	102.5324	0	0
26	Other Non	1999	11	76.60887	75.64741074	97.48407	173.5545	69.7887	75.93449	0	0
26	Other Non	1999	12	86.37007	92.84505145	90.09056	119.3139	77.83425	89.2389	0	0
26	Other Non	2000	1	102.4435	104.0077261	90.63425	85.40432	97.09317	111.2948	0	0
32	Radio,TV,(	1999	2	107.1444	110.3738952	121.3187	114.7802	115.962	115.971	0	0
32	Radio,TV,(	1999	3	134.3398	135.6655611	124.4224	97.29806	143.5605	144.2414	0	0
32	Radio,TV,(	1999	4	103.4603	107.3645099	133.3698	129.7876	106.8004	112.0998	0	0
32	Radio,TV,(	1999	5	124.2926	126.0429672	164.8632	134.2469	131.1653	124.0478	0	0
32	Radio,TV,(	1999	6	139.9695	144.0769273	163.8773	115.871	146.9097	137.0947	0	0
32	Radio,TV,(	1999	7	144.7436	149.898041	192.819	131.1345	145.9741	133.0807	0	0
32	Radio,TV,(	1999	8	145.8663	150.1271055	182.5105	122.5545	152.0285	192.1215	0	0
32	Radio,TV,(	1999	9	156.4571	160.0513461	196.1374	125.2487	157.2056	140.9398	0	0
32	Radio,TV,(	1999	10	148.2172	156.109052	165.7615	107.1333	161.1076	130.0046	0	0
32	Radio,TV,(	1999	11	157.8841	158.7857222	213.9165	134.9747	169.0736	149.9476	0	0
32	Radio,TV,(	1999	12	152.1951	162.4658402	189.4529	125.4228	163.4002	144.6215	0	0
32	Radio,TV,(	2000	1	148.6998	154.0117498	234.3575	151.9423	163.7932	138.3322	0	0
34	Vehicles, (	1999	2	113.3527	106.6370215	104.1207	104.2504	99.00609	115.7219	0	0
34	Vehicles, (	1999	3	127.4625	115.6496124	114.226	100.2896	99.84788	119.6347	0	0
34	Vehicles, (	1999	4	114.7034	129.0546998	91.63817	72.03191	82.83898	149.1043	0	0
34	Vehicles, (	1999	5	138.2931	132.3736522	93.09497	69.23075	98.55972	134.6703	0	0
34	Vehicles, (	1999	6	158.7173	159.6053942	84.52004	51.444	105.8897	129.4566	0	0
34	Vehicles, (	1999	7	176.9934	177.4334334	78.95558	43.31807	115.603	140.468	0	0
34	Vehicles, (	1999	8	186.3557	169.7662344	93.83724	54.83228	128.7276	144.2606	0	0
34	Vehicles, (	1999	9	197.8965	202.0464148	81.72676	42.78323	128.4322	146.7533	0	0
34	Vehicles, (	1999	10	195.4334	180.6598328	95.4855	53.25678	127.7995	144.9711	0	0
34	Vehicles, (	1999	11	217.9862	177.8292254	139.0755	78.22651	173.0129	159.0056	0	0
34	Vehicles, (	1999	12	160.9904	212.3491599	65.94003	46.54048	173.6258	144.4421	0	0
34	Vehicles, (	2000	1	73.7679	71.33330176	36.74879	62.90854	130.5002	77.66044	0	0

List of the Time-Series Indices by Commodity /0501P1

ISIC : **SurveyScope :**

Commodity Group : **Report Type :**

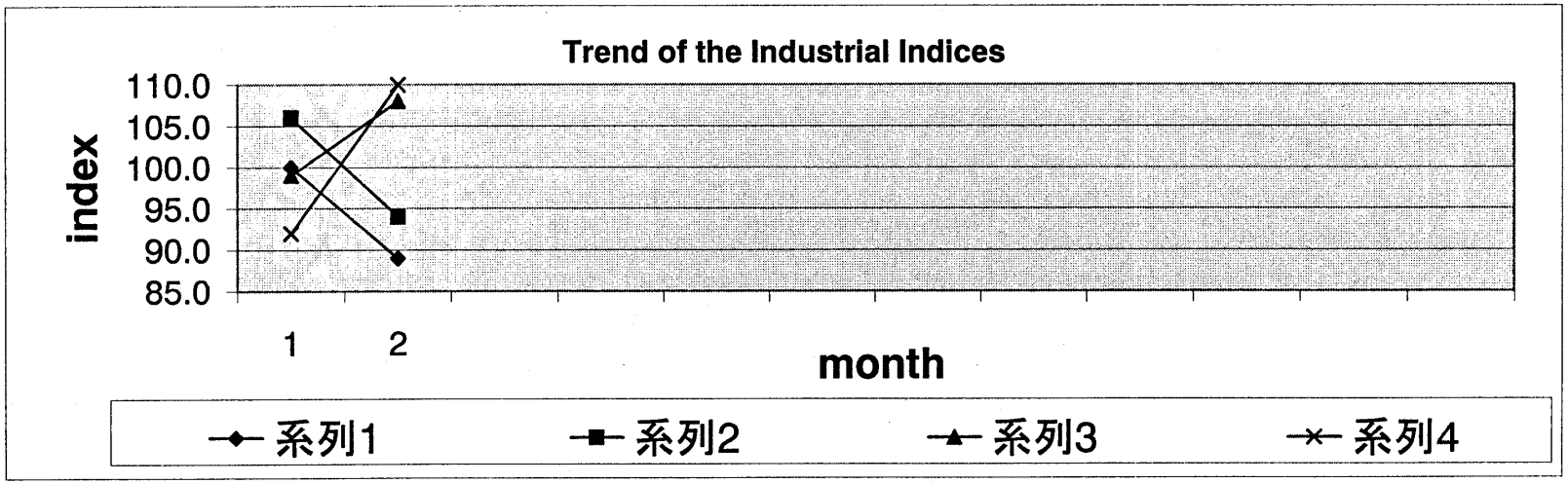
Commodity														
		jan	feb	mar	apr	may	jun	jul	aug	sep	oct	nov	dec	jan

Graph & Table for the General Explanation 1001A

ISIC : **SurveyScope :**

Commodity Group : **Report Type :**

Indices												
	1	2										
Production	100.0	89.0										
Shipment	106.0	94.0										
Inventory	99.0	108.0										
Inventory Ratio	92.0	110.0										

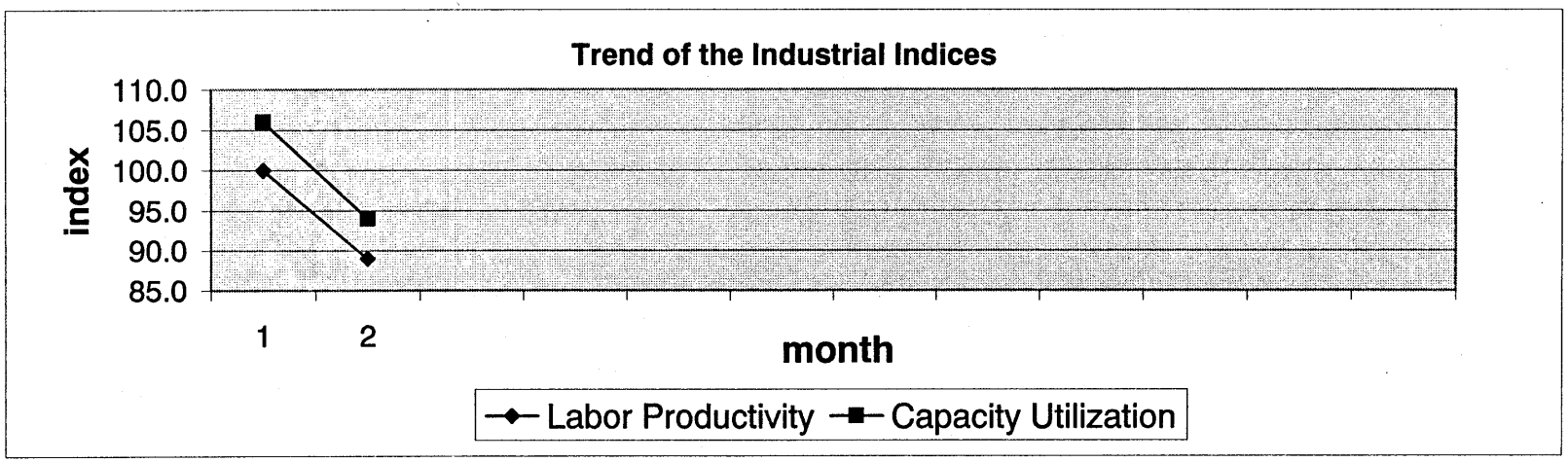


Graph & Table for the General Explanation /1001B

ISIC : **SurveyScope :**

Commodity Group : **Report Type :**

Indices												
	1	2										
Labor Productivity	100.0	89.0										
Capacity Utilizatio	106.0	94.0										



Indices of Industrial Production /0901

Year: **Month:**
Survey Scope: **Report Type:**

Outline of the trend

Index,1999 = 100

Indices	index	Growth Rate(%)	
Production			
Shipment			
Inventory			
Inventor Ratio			
Labor Productivity			
Capacity Utilization			

Detail Coment

(1)- - -

Conversion Program

```
/*
 1    ISIC in SurveyHeader are exactly same as ISIC in SurveyData? -> OK
 */
select distinct isic||RegistrationNO||Year||Month from surveyheader
minus
select distinct isic||RegistrationNO||Year||Month from surveydata;

select distinct isic||RegistrationNO||Year||Month from surveydata
minus
select distinct isic||RegistrationNO||Year||Month from surveyheader;


/*
 2    Number of Item 1 & 2 ->OK
 */
select count(*) from surveydata where itemtype = '1';
select count(*) from surveydata where itemtype = '2' and commoditycode != '999';
select count(*) from surveydata where itemtype = '2' and commoditycode  = '999';

select /*+ all_rows*/ ISIC||'-'||YEAR||'-'||MONTH||'-'||RegistrationNo||'-'||commoditycode
from surveydata
where itemtype = '1'
and ISIC||'-'||YEAR||'-'||MONTH||'-'||RegistrationNo||'-'||commoditycode
    not in
    (
    select /*+ all_rows*/ ISIC||'-'||YEAR||'-'||MONTH||'-'||RegistrationNo||'-'||commodity
code
    from surveydata
    where itemtype = '2'
     and  commoditycode != '999'
    );

select /*+ all_rows*/ ISIC||'-'||YEAR||'-'||MONTH||'-'||RegistrationNo||'-'||commoditycode
from surveydata
where itemtype = '2'
and ISIC||'-'||YEAR||'-'||MONTH||'-'||RegistrationNo||'-'||commoditycode
    not in
    (
    select /*+ all_rows*/ ISIC||'-'||YEAR||'-'||MONTH||'-'||RegistrationNo||'-'||commodity
code
    from surveydata
    where itemtype = '1'
     and  commoditycode != '999'
    );

select /*+ all_rows*/ ISIC||'-'||YEAR||'-'||MONTH||'-'||RegistrationNo||'-'||commoditycode
from surveydata where itemtype = '2' and commoditycode != '999'
minus
select /*+ all_rows*/ ISIC||'-'||YEAR||'-'||MONTH||'-'||RegistrationNo||'-'||commoditycode
from surveydata where itemtype = '1' ;

select /*+ all_rows*/ ISIC||'-'||YEAR||'-'||MONTH||'-'||RegistrationNo||'-'||commoditycode
from surveydata where itemtype = '1'
minus
select /*+ all_rows*/ ISIC||'-'||YEAR||'-'||MONTH||'-'||RegistrationNo||'-'||commoditycode
from surveydata where itemtype = '2' and commoditycode != '999' ;



/*
 3 All RegistrationNo are really registered in EstablishmentMaster? -> NO
 */
select distinct (ISIC||' - '||RegistrationNO)from SurveyHeader
where --year = 2000
   ISIC||' - '||RegistrationNO not IN
    ( select ISIC||' - '||RegistrationNO from EstablishmentMaster)
order by ISIC||' - '||RegistrationNO;

/*
 4 Number of Item3's records should be (3 * number of surveyheader). -> OK
 */
select count(*) from surveyheader;
select count(*) from surveydata where itemtype = '3';
--select count(*) from surveydata where itemtype = '3' and commoditycode = '010';
--select count(*) from surveydata where itemtype = '3' and commoditycode = '020';
--select count(*) from surveydata where itemtype = '3' and commoditycode = '030';
```

```
/*
 5 Number of Item6's records should be  number of item1. -> NO
 */
select count(*) from surveydata where itemtype = '1';
select count(*) from surveydata where itemtype = '6';
select distinct ISIC||CommodityCode from surveydata where itemtype = '1'
minus
select distinct ISIC||CommodityCode from surveydata where itemtype = '6';

select distinct ISIC||'-'||CommodityCode||'-'||Year||'-'||Month||'-'||RegistrationNO from
surveydata where itemtype = '1'
minus
select distinct ISIC||'-'||CommodityCode||'-'||Year||'-'||Month||'-'||RegistrationNO from
surveydata where itemtype = '6';

/*
 6    Number of collected Establishments
 */

select count(distinct registrationno) as JAN from surveyheader where year = 1999 and month
= 1;
select count(distinct registrationno) as FEB from surveyheader where year = 1999 and month
= 2;
select count(distinct registrationno) as MAR from surveyheader where year = 1999 and month
= 3;
select count(distinct registrationno) as APR from surveyheader where year = 1999 and month
= 4;
select count(distinct registrationno) as MAY from surveyheader where year = 1999 and month
= 5;
select count(distinct registrationno) as JUN from surveyheader where year = 1999 and month
= 6;
select count(distinct registrationno) as JUL from surveyheader where year = 1999 and month
= 7;
select count(distinct registrationno) as AUG from surveyheader where year = 1999 and month
= 8;
select count(distinct registrationno) as SEP from surveyheader where year = 1999 and month
= 9;
select count(distinct registrationno) as OCT from surveyheader where year = 1999 and month
= 10;
select count(distinct registrationno) as NOV from surveyheader where year = 1999 and month
= 11;
select count(distinct registrationno) as DEC from surveyheader where year = 1999 and month
= 12;
--
--select count(*) from surveyheader where isic = '341001';
--select count(*) from commoditymaster where isic = '341001';
--select count(*) from commoditymaster where isic = '341010';

/*
 7 Number of transfered ISIC. -> NO
 */
select distinct isic from surveyheader;

151210
155310
171110
173010
181010
232010
269410
321010
321020
323010
341001

11 rows selected.
```

```
-- ==================================================
-- Module Name  : Conversion Module from Pilot System.
-- Create Date  : 03/Feb/2000
-- Replace Date : xx/xxx/xxxx
-- Author       : Nakamura
-- ==================================================
----------------------------------------------------
-- Step 06 : Conversion To Survey Data
----------------------------------------------------
-- 1. Item Type 1 Conversion
        insert into surveydata
                select  c.NewISIC,
                                                s.YEAR,
                                                -- YEAR
                                                s.MONTH,
                                                -- MONTH
                                                s.REGISTRATIONNO,
                        -- REGISTRATIONNO
                                                c.NewITEMTYPE,
                        -- ITEMTYPE
                                                c.NewCOMMODITYCODE,
                -- COMMODITYCODE
                                                null,
                                                        -- BM_INVENTORY(1)
                                                'U',
                                                        -- ATRBT_BM_INVENTORY
                                                s.PRODUCTIONQTY,
                        -- PRODUCTIONQTY(1)
                                                decode(s.PRODUCTIONQTY, null, 'U', 'V'),  -- A
TRBT_PRODUCTIONQTY
                                                null,
                                                        -- RECEIPTS(1)
                                                'U',
                                                        -- ATRBT_RECEIPTS
                                                s.SALESQTY,
                        -- DOMESTICSALES(1)
                                                decode(s.SALESQTY, null, 'U', 'V'),       -- A
TRBT_DOMESTICSALES
                                                s.EXPORTQTY,
                        -- EXPORT(1)
                                                decode(s.EXPORTQTY, null, 'U', 'V'),      -- A
TRBT_EXPORT
                                                s.IntraUseQTY,
                        -- OTHERSALES(1)
                                                decode(s.IntraUseQTY, null, 'U', 'V'),    -- A
TRBT_OTHERSALES
                                                s.MONTHENDINVENTORY,                      -- M
E_INVENTORY(1)
                                                decode(s.MONTHENDINVENTORY, null, 'U', 'V'),
-- ATRBT_ME_INVENTORY
                                                null,
                                        -- CAPACITY(6)
                                                null,

                -- ATRBT_CAPACITY
                                                null,
                                        -- SHIPMENTVALUE(2)
                                                null,

                        -- ATRBT_SHIPMENTVALUE
                                                null,
                                                -- SALESPLAN(2)
                                                null,
                                                -- ATRBT_SALESPLAN
                                                null,
                                        -- LABORTOTAL(3)
                                                null,
                                                                                        --
ATRBT_LABORTOTAL
                                                null,
                                        -- LABOR_SC(3)
                                                null,
                                                                                        --
ATRBT_LABOR_SC
                                                null,
                                                                        -- RWMT_MEINVENTORY
(4)
```

```
                                        null,
                                                                                -- ATRBT_RW
MT_MEINVENTOR
                                        null,
                                        -- RWMT_MEINVENTORYVALUE(4)
                                        null,
                                                -- ATRBT_RWMT_MEINVENTOR
                                        null,
                                -- BIZCONDITION(5)
                                        null,

                -- UNITPRICE(1)
                                        s.REMARKS,
                -- REMARKS
                                        'convsan',
                -- USERID
                                sysdate                                         -- RECDATE
                from    surveydata_wk1 s, ConvMaster_wk1 c
                where s.QuestionnaireNo = c.OldquestionnaireNo
        and     s.ItemType = c.OldItemType
        and             s.CommodityCode = c.OldCommodityCode
                and             c.NewItemType = '1'
                and             c.NewDelFlg is null ;
                --and           s.year = 1999;
                --and           s.month = 1;

-- 2.Item Type 2 Conversion
        insert into surveydata
        select  c.NewISIC,
                                        s.YEAR,
                                        -- YEAR
                                        s.MONTH,
                                        -- MONTH
                                        s.REGISTRATIONNO,
                -- REGISTRATIONNO
                                        c.NewITEMTYPE,
                -- ITEMTYPE
                                        c.NewCOMMODITYCODE,
        -- COMMODITYCODE
                                        null,
                                                -- BM_INVENTORY(1)
                                        null,
                                                -- ATRBT_BM_INVENTORY
                                        null,
                                                -- PRODUCTIONQTY(1)
                                        null,

        -- ATRBT_PRODUCTIONQTY
                                        null,
                                                -- RECEIPTS(1)
                                        null,
                                                -- ATRBT_RECEIPTS
                                        null,
                                        -- DOMESTICSALES(1)
                                        null,
                                                                                --
ATRBT_DOMESTICSALES
                                        null,
                                        -- EXPORT(1)
                                        null,
                                                                                --
ATRBT_EXPORT
                                        null,
                                        -- OTHERSALES(1)
                                        null,
                                        -- ATRBT_OTHERSALES
                                        null,
                                        -- ME_INVENTORY(1)
                                        null,

        -- ATRBT_ME_INVENTORY
                                        null,
                                -- CAPACITY(6)
                                        null,

        -- ATRBT_CAPACITY
                                        SHIPMENTTOTALVALUE,                     -- S
```

```
HIPMENTVALUE(2)
                                                decode(SHIPMENTTOTALVALUE, null, 'U', 'V'), --
ATRBT_SHIPMENTVALUE
                                                null,
                                                -- SALESPLAN(2)
                                                'U',
                                                -- ATRBT_SALESPLAN
                                                null,
                                        -- LABORTOTAL(3)
                                                null,
                                                                                                --
ATRBT_LABORTOTAL
                                                null,
                                        -- LABOR_SC(3)
                                                null,
                                                                                                --
ATRBT_LABOR_SC
                                                null,
                                                                        -- RWMT_MEINVENTORY
(4)
                                                null,
                                                                                -- ATRBT_RW
MT_MEINVENTOR
                                                null,
                                                -- RWMT_MEINVENTORYVALUE(4)
                                                null,
                                                        -- ATRBT_RWMT_MEINVENTOR
                                                null,
        -- BIZCONDITION(5)
                                                null,

                        -- UNITPRICE(1)
                                                s.REMARKS,
                        -- REMARKS
                                                'convsan',
                        -- USERID
                                        sysdate                                 -- RECDATE
                        from    surveydata_wk1 s, ConvMaster_wk1 c
                        where s.QuestionnaireNo = c.OldquestionnaireNo
                and     s.ItemType = c.OldItemType
                and             s.CommodityCode = c.OldCommodityCode
                        and             c.NewItemType = '2'
                        and             c.NewDelFlg is null ;
                        --and           s.year = 1999
                        --and           s.month = 1;

-- 3.Item Type 3 Conversion
        insert into surveydata
                select  c.NewISIC,
                                                s.YEAR,
                                                -- YEAR
                                                s.MONTH,
                                                -- MONTH
                                                s.REGISTRATIONNO,
                        -- REGISTRATIONNO
                                                c.NewITEMTYPE,
                        -- ITEMTYPE
                                                c.NewCOMMODITYCODE,
                -- COMMODITYCODE
                                                null,
                                                        -- BM_INVENTORY(1)
                                                null,
                                                        -- ATRBT_BM_INVENTORY
                                                null,
                                                        -- PRODUCTIONQTY(1)
                                                null,

                -- ATRBT_PRODUCTIONQTY
                                                null,
                                                        -- RECEIPTS(1)
                                                null,
                                                        -- ATRBT_RECEIPTS
                                                null,
                                                -- DOMESTICSALES(1)
                                                null,
                                                                                                --
ATRBT_DOMESTICSALES
```

```
                                        null,
                                        -- EXPORT(1)
                                        null,
                                                                                --
ATRBT_EXPORT
                                        null,
                                        -- OTHERSALES(1)
                                        null,
                                        -- ATRBT_OTHERSALES
                                        null,
                                        -- ME_INVENTORY(1)
                                        null,

            -- ATRBT_ME_INVENTORY
                                        null,
                                -- CAPACITY(6)
                                        null,

            -- ATRBT_CAPACITY
                                        null,
                                        -- SHIPMENTVALUE(2)
                                        null,

                -- ATRBT_SHIPMENTVALUE
                                        null,
                                        -- SALESPLAN(2)
                                        null,
                                        -- ATRBT_SALESPLAN
                                        QTY_FORLABOR     ,                      -- L
ABORTOTAL(3)
                                        decode(QTY_FORLABOR, null, 'U', 'V'),   -- A
TRBT_LABORTOTAL
                                        QTY_FORLABOR     ,                      -- L
ABOR_SC(3)
                                        decode(QTY_FORLABOR, null, 'U', 'V'),   -- A
TRBT_LABOR_SC
                                        null,
                                                                -- RWMT_MEINVENTORY
(4)
                                        null,
                                                                        -- ATRBT_RW
MT_MEINVENTOR
                                        null,
                                        -- RWMT_MEINVENTORYVALUE(4)
                                        null,
                                                -- ATRBT_RWMT_MEINVENTOR
                                        null,
        -- BIZCONDITION(5)
                                        null,

                -- UNITPRICE(1)
                                        s.REMARKS,
                -- REMARKS
                                        'convsan',
                -- USERID
                                sysdate                                         -- RECDATE
                from    surveydata_wk1 s, ConvMaster_wk1 c
                where s.QuestionnaireNo = c.OldquestionnaireNo
            and     s.ItemType = c.OldItemType
            and             s.CommodityCode = c.OldCommodityCode
                and             c.NewItemType = '3'
                and             c.NewDelFlg is null ;
                -- and          s.year = 1999
                -- and          s.month = 1;

-- 4.Item Type 4 Conversion
        insert into surveydata
            select  c.NewISIC,
                                        s.YEAR,
                                        -- YEAR
                                        s.MONTH,
                                        -- MONTH
                                        s.REGISTRATIONNO,
                -- REGISTRATIONNO
                                        c.NewITEMTYPE,
                -- ITEMTYPE
                                        c.NewCOMMODITYCODE,
```

```
                -- COMMODITYCODE
                                        null,
                                                -- BM_INVENTORY(1)
                                        null,
                                                -- ATRBT_BM_INVENTORY
                                        null,
                                                -- PRODUCTIONQTY(1)
                                        null,

                -- ATRBT_PRODUCTIONQTY
                                        null,
                                                -- RECEIPTS(1)
                                        null,
                                                -- ATRBT_RECEIPTS
                                        null,
                                        -- DOMESTICSALES(1)
                                        null,
                                                                                --
ATRBT_DOMESTICSALES
                                        null,
                                        -- EXPORT(1)
                                        null,
                                                                                --
ATRBT_EXPORT
                                        null,
                                        -- OTHERSALES(1)
                                        null,
                                        -- ATRBT_OTHERSALES
                                        null,
                                        -- ME_INVENTORY(1)
                                        null,

                -- ATRBT_ME_INVENTORY
                                        null,
                                -- CAPACITY(6)
                                        null,

                -- ATRBT_CAPACITY
                                        null,
                                        -- SHIPMENTVALUE(2)
                                        null,

                        -- ATRBT_SHIPMENTVALUE
                                        null,
                                        -- SALESPLAN(2)
                                        null,
                                        -- ATRBT_SALESPLAN
                                        null,
                                -- LABORTOTAL(3)
                                        null,

                -- ATRBT_LABORTOTAL
                                        null,
                                        -- LABOR_SC(3)
                                        null,

                        -- ATRBT_LABOR_SC
                                        MEINVENTORY_FORROWMATERIALS,            -- R
WMT_MEINVENTORY(4)
                                        decode(MEINVENTORY_FORROWMATERIALS, null, 'U
','V'),   -- ATRBT_RWMT_MEINVENTOR
                                        null,
                                                -- RWMT_MEINVENTORYVALUE(4)
                                        'U',

                        -- ATRBT_RWMT_MEINVENTOR
                                        null,
                                -- BIZCONDITION(5)
                                        null,

                        -- UNITPRICE(1)
                                        s.REMARKS,
                        -- REMARKS
                                        'convsan',
                        -- USERID
                                sysdate                                         -- RECDATE
                        from    surveydata_wk1 s, ConvMaster_wk1 c
```

```
                where s.QuestionnaireNo = c.OldquestionnaireNo
        and     s.ItemType = c.OldItemType
        and             s.CommodityCode = c.OldCommodityCode
                and             c.NewItemType = '4'
                and             c.NewDelFlg is null
                -- and          s.year = 1999
                -- and          s.month = 1;

-- 5.Item Type 5 Conversion
        insert into surveydata
        select  c.NewISIC,
                                        s.YEAR,
                                        -- YEAR
                                        s.MONTH,
                                        -- MONTH
                                        s.REGISTRATIONNO,
                -- REGISTRATIONNO
                                        c.NewITEMTYPE,
                -- ITEMTYPE
                                        c.NewCOMMODITYCODE,
        -- COMMODITYCODE
                                        null,
                                                -- BM_INVENTORY(1)
                                        null,
                                                -- ATRBT_BM_INVENTORY
                                        null,
                                                -- PRODUCTIONQTY(1)
                                        null,

        -- ATRBT_PRODUCTIONQTY
                                        null,
                                                -- RECEIPTS(1)
                                        null,
                                                -- ATRBT_RECEIPTS
                                        null,
                                        -- DOMESTICSALES(1)
                                        null,
                                                                        --
ATRBT_DOMESTICSALES
                                        null,
                                        -- EXPORT(1)
                                        null,
                                                                        --
ATRBT_EXPORT
                                        null,
                                        -- OTHERSALES(1)
                                        null,
                                        -- ATRBT_OTHERSALES
                                        null,
                                        -- ME_INVENTORY(1)
                                        null,

        -- ATRBT_ME_INVENTORY
                                        null,
                                -- CAPACITY(6)
                                        null,

        -- ATRBT_CAPACITY
                                        null,
                                        -- SHIPMENTVALUE(2)
                                        null,

                -- ATRBT_SHIPMENTVALUE
                                        null,
                                        -- SALESPLAN(2)
                                        null,
                                        -- ATRBT_SALESPLAN
                                        null,
                        -- LABORTOTAL(3)
                                        null,

        -- ATRBT_LABORTOTAL
                                        null,
                                        -- LABOR_SC(3)
                                        null,

                -- ATRBT_LABOR_SC
```

```
                                        null,
                                                                        -- RWMT_MEINVENTORY
(4)
                                        null,

                -- ATRBT_RWMT_MEINVENTOR
                                        null,
                                                -- RWMT_MEINVENTORYVALUE(4)
                                        null,

                -- ATRBT_RWMT_MEINVENTOR
                                        BIZCONDITION,
        -- BIZCONDITION(5)
                                        null,

                -- UNITPRICE(1)
                                        s.REMARKS,
                -- REMARKS
                                        'convsan',
                -- USERID
                                sysdate                                 -- RECDATE
                from    surveydata_wk1 s, ConvMaster_wk1 c
                where s.QuestionnaireNo = c.OldquestionnaireNo
        and     s.ItemType = c.OldItemType
        and             s.CommodityCode = c.OldCommodityCode
                and             c.NewItemType = '5'
                and             c.NewDelFlg is null
                -- and          s.year = 1999
                -- and          s.month = 1;

-- 6.Item Type 6 Conversion(Convert From ItemType1 )
        insert into surveydata
        select  c.NewISIC,
                                        s.YEAR,
                                        -- YEAR
                                        s.MONTH,
                                        -- MONTH
                                        s.REGISTRATIONNO,
                -- REGISTRATIONNO
                                        '6',

                -- ITEMTYPE
                                        c.NewCOMMODITYCODE,
        -- COMMODITYCODE
                                        null,
                                                -- BM_INVENTORY(1)
                                        null,
                                                -- ATRBT_BM_INVENTORY
                                        null,
                                                -- PRODUCTIONQTY(1)
                                        null,

        -- ATRBT_PRODUCTIONQTY
                                        null,
                                                -- RECEIPTS(1)
                                        null,
                                                -- ATRBT_RECEIPTS
                                        null,
                                        -- DOMESTICSALES(1)
                                        null,
                                                                        --
ATRBT_DOMESTICSALES
                                        null,
                                        -- EXPORT(1)
                                        null,
                                                                        --
ATRBT_EXPORT
                                        null,
                                        -- OTHERSALES(1)
                                        null,
                                        -- ATRBT_OTHERSALES
                                        null,
                                        -- ME_INVENTORY(1)
                                        null,

        -- ATRBT_ME_INVENTORY
                                        ANNUALCAPACITY   ,                      -- C
```

```
APACITY(6)
                                        decode(ANNUALCAPACITY, null, 'U', 'V'),   -- A
TRBT_CAPACITY
                                        null,
                                        -- SHIPMENTVALUE(2)
                                        null,

                -- ATRBT_SHIPMENTVALUE
                                        null,
                                        -- SALESPLAN(2)
                                        null,
                                        -- ATRBT_SALESPLAN
                                        null,
                        -- LABORTOTAL(3)
                                        null,

        -- ATRBT_LABORTOTAL
                                        null,
                                        -- LABOR_SC(3)
                                        null,

                -- ATRBT_LABOR_SC
                                        null,
                                                                -- RWMT_MEINVENTORY
(4)
                                        null,

        -- ATRBT_RWMT_MEINVENTOR
                                        null,
                                                -- RWMT_MEINVENTORYVALUE(4)
                                        null,

                -- ATRBT_RWMT_MEINVENTOR
                                        null,
                                -- BIZCONDITION(5)
                                        null,

                -- UNITPRICE(1)
                                        s.REMARKS,
                -- REMARKS
                                        'convsan',
                -- USERID
                                sysdate                         -- RECDATE
                from    surveydata_wk1 s, ConvMaster_wk1 c
                where s.QuestionnaireNo = c.OldquestionnaireNo
        and     s.ItemType = c.OldItemType
        and             s.CommodityCode = c.OldCommodityCode
                and             c.NewItemType = '1'
                and             c.NewDelFlg is null
                -- and          s.year = 1999
                -- and          s.month = 1;

-- 7.Item Type 2 Summary Data Create
        insert into surveydata
        select  ISIC,
                                        YEAR,
                                        MONTH,
                                        REGISTRATIONNO,
                                        ItemType,
                                        '999',
                                        null,
                                        null,
                                        null,
                                        null,
                                        null,
                                        null,
                                        null,
                                        null,
                                        null,
                                        null,
                                        null,
                                        null,
                                        null,
                                        null,
                                        null,
                                        null,
                                sum(shipmentvalue),
```

```
                                                null,
                                                null,
                                                null,
                                                null,
                                                null,
                                                null,
                                                null,
                                                null,
                                                null,
                                                null,
                                                null,
                                                null,
                                                null,
                                                null,
                                                'CONVSAN',
                                                sysdate
                        from    surveydata
                        where itemtype = 2
                        --and   year = 1999
                        --and   month = 1
                        and             CommodityCode != '999'
                        group by ISIC, Year, Month, RegistrationNo, itemtype;

-- 8. Set Attribute for Item Type 2 Summary Data
        update  surveydata
                set             ATRBT_SHIPMENTVALUE = 'V'
                where   ITEMTYPE = '2'
                        AND     SHIPMENTVALUE is not null
                        and     CommodityCode = '999'
                        and     year = 1999;

        update  surveydata
                set             ATRBT_SHIPMENTVALUE = 'U'
                where   ITEMTYPE = '2'
                        AND     SHIPMENTVALUE is null
                        and     CommodityCode = '999'
                        and     year = 1999;

-- 9. clear labor_sc of itemtype 3
        update  surveydata
                set     labor_sc = null,
                ATRBT_LABOR_SC = 'U'
         where  itemtype = 3
         and    commodityCode in ('020','030');
         --and          year = 1999
         --and          month = 1;

-------------------------------------------------------
-- Step 07 : Survey Header Data Create
-------------------------------------------------------
        insert into surveyheader
                select distinct
                        isic, year, month, registrationNo, 0, 0, 0
                        from  surveydata
                        --where year = 1999
                        --and month = 1
                        and itemtype = 1;

-------------------------------------------------------
-- Step 08 : Copy Data For Separated ISIC in Item Type 1
-------------------------------------------------------
        insert into     SurveyData
                select  '181010',
                                                YEAR,
                                                MONTH,
                                                REGISTRATIONNO,
                                                ITEMTYPE,
                                                COMMODITYCODE,
                                                BM_INVENTORY,
                                                ATRBT_BM_INVENTORY,
                                                PRODUCTIONQTY,
                                                ATRBT_PRODUCTIONQTY,
                                                RECEIPTS,
                                                ATRBT_RECEIPTS,
                                                DOMESTICSALES,
                                                ATRBT_DOMESTICSALES,
                                                EXPORT,
```

```
                                                ATRBT_EXPORT,
                                                OTHERSALES,
                                                ATRBT_OTHERSALES,
                                                ME_INVENTORY,
                                                ATRBT_ME_INVENTORY,
                                                CAPACITY,
                                                ATRBT_CAPACITY,
                                                SHIPMENTVALUE,
                                                ATRBT_SHIPMENTVALUE,
                                                SALESPLAN,
                                                ATRBT_SALESPLAN,
                                                LABORTOTAL,
                                                ATRBT_LABORTOTAL,
                                                LABOR_SC,
                                                ATRBT_LABOR_SC,
                                                RWMT_MEINVENTORY,
                                                ATRBT_RWMT_MEINVENTORY,
                                                RWMT_MEINVENTORYVALUE,
                                                ATRBT_RWMT_MEINVENTORYVALUE,
                                                BIZCONDITION,
                                                UNITPRICE,
                                                REMARKS,
                                                USERID,
                                                RECDATE
                        from    SurveyData
                        where   ISIC = '173010'
                                --and    Year    = 1999
                                --and    Month = 1
                                and     ItemType in ('3','5');

-------------------------------------------------------
-- Step 09 : BM_Inventory Automatic Setting
-------------------------------------------------------
-- Set BM_Inventory
        update  SurveyData a
                set             BM_Inventory =
                        (select ME_Inventory
                                from SurveyData b
                                where a.ISIC = b.ISIC
                                        and     a.Year = b.Year
                                        and a.Month - 1 = b.Month
                                        and a.RegistrationNo = b.RegistrationNo
                                        and a.ItemType = b.ItemType
                                        and a.commodityCode = b.CommodityCode)
                where   a.ItemType = '1';
                        and     a.Year = 1999
                        and a.Month in(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12);

-- Set Attribute
        update  SurveyData
                set             Atrbt_BM_Inventory = 'V'
                where   BM_Inventory is not null
                        and     Year = 1999
                        and Month in(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12);

-------------------------------------------------------
-- Step 10 : Set BM_Inventory, Receipts, OtherSales
-------------------------------------------------------
update surveydata
set bm_inventory = 0,
    atrbt_bm_inventory = 'V'
where (atrbt_productionqty = 'V'
   or atrbt_domesticsales = 'V'
   or atrbt_export = 'V'
   or atrbt_me_inventory = 'V')
   and bm_inventory is null
   and year = 1999;


update surveydata
set receipts= 0,
    atrbt_receipts = 'V'
where (atrbt_productionqty = 'V'
   or atrbt_domesticsales = 'V'
   or atrbt_export = 'V'
   or atrbt_me_inventory = 'V')
   and receipts is null
```

```
   and year = 1999;

update surveydata
set othersales= 0,
    atrbt_othersales = 'V'
where (atrbt_productionqty = 'V'
   or atrbt_domesticsales = 'V'
   or atrbt_export = 'V'
   or atrbt_me_inventory = 'V')
   and othersales is null
   and year = 1999;
```

```
select ISIC , count(*) from establishmentmaster
where continuousrespondent = '1'
group by isic;


select ISIC , count(*) from establishmentmaster
group by isic;



select count(DISTINCT REGISTRATIONNO) from establishmentmaster
where continuousrespondent = '1' ;

select count(REGISTRATIONNO) from establishmentmaster ;


select ISIC , count(*), SurveyScope from establishmentmaster
group by isic, SurveyScope;

select count(DISTINCT REGISTRATIONNO) from establishmentmaster
where continuousrespondent = '1'
  and isic in ('173010','181010')
  and surveyscope = '1';
```

```
--
-- Auto Matic Yearly Process
-- Sabai Mark Mark Proc
--
--      @g:¥conversion¥GRCal;
--

truncate table pastrecord;
/

begin Weight.CreateWeightView; end;
/

-- 1
begin AutoEst.p_UnitPriceUpd(1999, 1); commit; end;
/
begin MonthProc.p_PastRecMk('2', 1999, 1);commit; end;
/
begin CalcIndex.Main(1999, 1, '2'); commit; end;
/
begin GRCal.p_GRIndices(1999, 1, '2'); commit; end;
/
begin GRCal.p_GREleComm(1999, 1, '2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999, 1, '2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999, 1); commit; end;
/

-- 2
begin AutoEst.p_UnitPriceUpd(1999, 2); commit; end;
/
begin MonthProc.p_PastRecMk('2', 1999, 2);commit; end;
/
begin CalcIndex.Main(1999, 2, '2'); commit; end;
/
begin GRCal.p_GRIndices(1999, 2, '2'); commit; end;
/
begin GRCal.p_GREleComm(1999, 2, '2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999, 2, '2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999, 2); commit; end;
/

-- 3
begin AutoEst.p_UnitPriceUpd(1999, 3); commit; end;
/
begin MonthProc.p_PastRecMk('2', 1999, 3);commit; end;
/
begin CalcIndex.Main(1999, 3, '2'); commit; end;
/
begin GRCal.p_GRIndices(1999, 3, '2'); commit; end;
/
begin GRCal.p_GREleComm(1999, 3, '2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999, 3, '2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999, 3); commit; end;
/

-- 4
begin AutoEst.p_UnitPriceUpd(1999, 4); commit; end;
/
begin MonthProc.p_PastRecMk('2', 1999, 4);commit; end;
/
begin CalcIndex.Main(1999, 4, '2'); commit; end;
/
begin GRCal.p_GRIndices(1999, 4, '2'); commit; end;
/
begin GRCal.p_GREleComm(1999, 4, '2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999, 4, '2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999, 4); commit; end;
/
```

```
-- 5
begin AutoEst.p_UnitPriceUpd(1999,5); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,5);commit; end;
/
begin CalcIndex.Main(1999,5,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,5,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,5,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,5,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,5); commit; end;
/

-- 6
begin AutoEst.p_UnitPriceUpd(1999,6); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,6);commit; end;
/
begin CalcIndex.Main(1999,6,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,6,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,6,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,6,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,6); commit; end;
/

-- 7
begin AutoEst.p_UnitPriceUpd(1999,7); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,7);commit; end;
/
begin CalcIndex.Main(1999,7,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,7,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,7,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,7,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,7); commit; end;
/

-- 8
begin AutoEst.p_UnitPriceUpd(1999,8); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,8);commit; end;
/
begin CalcIndex.Main(1999,8,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,8,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,8,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,8,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,8); commit; end;
/

-- 9
begin AutoEst.p_UnitPriceUpd(1999,9); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,9);commit; end;
/
begin CalcIndex.Main(1999,9,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,9,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,9,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,9,'2'); commit; end;
/
```

```
begin YearProc.p_BatchCtlUpd(1999,9); commit; end;
/

-- 10
begin AutoEst.p_UnitPriceUpd(1999,10); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,10);commit; end;
/
begin CalcIndex.Main(1999,10,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,10,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,10,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,10,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,10); commit; end;
/

-- 11
begin AutoEst.p_UnitPriceUpd(1999,11); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,11);commit; end;
/
begin CalcIndex.Main(1999,11,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,11,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,11,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,11,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,11); commit; end;
/

-- 12
begin AutoEst.p_UnitPriceUpd(1999,12); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,12);commit; end;
/
begin CalcIndex.Main(1999,12,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,12,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,12,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,12,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,12); commit; end;
/
BEGIN YearProc.p_AnnualCal(1999); Commit; END;
/

-- 2000-1
BEGIN AutoEst.main(2000,1,'2'); commit; END;
/
begin MonthProc.p_PastRecMk('2',2000,1);commit; end;
/
begin CalcIndex.Main(2000,1,'2'); commit; end;
/
begin GRCal.p_GRIndices(2000,1,'2'); commit; end;
/
begin GRCal.p_GREleComm(2000,1,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(2000,1,'2'); commit; end;
/
begin MonthProc.p_BatchCtlUpd('2',2000,1); commit; end;
/

-- 2000-2
BEGIN AutoEst.main(2000,2,'2'); commit; END;
/
begin MonthProc.p_PastRecMk('2',2000,2);commit; end;
/
begin CalcIndex.Main(2000,2,'2'); commit; end;
/
begin GRCal.p_GRIndices(2000,2,'2'); commit; end;
```

```
/
begin GRCal.p_GREleComm(2000,2,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(2000,2,'2'); commit; end;
/
begin MonthProc.p_BatchCtlUpd('2',2000,2); commit; end;
/

-- 2000-3
BEGIN AutoEst.main(2000,3,'2'); commit; END;
/
begin MonthProc.p_PastRecMk('2',2000,3);commit; end;
/
begin CalcIndex.Main(2000,3,'2'); commit; end;
/
begin GRCal.p_GRIndices(2000,3,'2'); commit; end;
/
begin GRCal.p_GREleComm(2000,3,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(2000,3,'2'); commit; end;
/
begin MonthProc.p_BatchCtlUpd('2',2000,3); commit; end;
/
```

```
--
-- Auto Matic Yearly Process
-- Sabai Mark Mark Proc
--
--      @E:¥userjica¥conversion¥ProcYear2.sql;
--

--truncate table pastrecord;
/

--begin Weight.CreateWeightView; end;
/

begin MonthProc.p_LinkCopy(2000,2); commit; end;

-- 1
/*
begin AutoEst.p_UnitPriceUpd(1999,1); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,1);commit; end;
/
begin CalcIndex.Main(1999,1,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,1,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,1,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,1,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,1); commit; end;
/

-- 2
begin AutoEst.p_UnitPriceUpd(1999,2); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,2);commit; end;
/
begin CalcIndex.Main(1999,2,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,2,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,2,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,2,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,2); commit; end;
/
*/

-- 3
begin AutoEst.p_UnitPriceUpd(1999,3); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,3);commit; end;
/
begin CalcIndex.Main(1999,3,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,3,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,3,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,3,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,3); commit; end;
/

-- 4
begin AutoEst.p_UnitPriceUpd(1999,4); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,4);commit; end;
/
begin CalcIndex.Main(1999,4,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,4,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,4,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,4,'2'); commit; end;
```

```
/
begin YearProc.p_BatchCtlUpd(1999,4); commit; end;
/

-- 5
begin AutoEst.p_UnitPriceUpd(1999,5); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,5);commit; end;
/
begin CalcIndex.Main(1999,5,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,5,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,5,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,5,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,5); commit; end;
/

-- 6
begin AutoEst.p_UnitPriceUpd(1999,6); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,6);commit; end;
/
begin CalcIndex.Main(1999,6,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,6,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,6,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,6,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,6); commit; end;
/

-- 7
begin AutoEst.p_UnitPriceUpd(1999,7); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,7);commit; end;
/
begin CalcIndex.Main(1999,7,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,7,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,7,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,7,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,7); commit; end;
/

-- 8
begin AutoEst.p_UnitPriceUpd(1999,8); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,8);commit; end;
/
begin CalcIndex.Main(1999,8,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,8,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,8,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,8,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,8); commit; end;
/

-- 9
begin AutoEst.p_UnitPriceUpd(1999,9); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,9);commit; end;
/
begin CalcIndex.Main(1999,9,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,9,'2'); commit; end;
/
```

```
begin GRCal.p_GREleComm(1999,9,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,9,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,9); commit; end;
/

-- 10
begin AutoEst.p_UnitPriceUpd(1999,10); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,10);commit; end;
/
begin CalcIndex.Main(1999,10,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,10,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,10,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,10,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,10); commit; end;
/

-- 11
begin AutoEst.p_UnitPriceUpd(1999,11); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,11);commit; end;
/
begin CalcIndex.Main(1999,11,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,11,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,11,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,11,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,11); commit; end;
/

-- 12
begin AutoEst.p_UnitPriceUpd(1999,12); commit; end;
/
begin MonthProc.p_PastRecMk('2',1999,12);commit; end;
/
begin CalcIndex.Main(1999,12,'2'); commit; end;
/
begin GRCal.p_GRIndices(1999,12,'2'); commit; end;
/
begin GRCal.p_GREleComm(1999,12,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(1999,12,'2'); commit; end;
/
begin YearProc.p_BatchCtlUpd(1999,12); commit; end;
/
BEGIN YearProc.p_AnnualCal(1999); Commit; END;
/

-- 2000-1
/*
BEGIN AutoEst.main(2000,1,'1'); commit; END;
/
begin MonthProc.p_PastRecMk('1',2000,1);commit; end;
/
begin CalcIndex.Main(2000,1,'1'); commit; end;
/
begin GRCal.p_GRIndices(2000,1,'1'); commit; end;
/
begin GRCal.p_GREleComm(2000,1,'1'); commit; end;
/
begin GRCal.p_GREleIndustry(2000,1,'1'); commit; end;
/
begin MonthProc.p_BatchCtlUpd('1',2000,1); commit; end;
/

-- 2000-2
BEGIN AutoEst.main(2000,2,'2'); commit; END;
/
```

```
begin MonthProc.p_PastRecMk('2',2000,2);commit; end;
/
begin CalcIndex.Main(2000,2,'2'); commit; end;
/
begin GRCal.p_GRIndices(2000,2,'2'); commit; end;
/
begin GRCal.p_GREleComm(2000,2,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(2000,2,'2'); commit; end;
/
begin MonthProc.p_BatchCtlUpd('2',2000,2); commit; end;
/

-- 2000-3
BEGIN AutoEst.main(2000,3,'2'); commit; END;
/
begin MonthProc.p_PastRecMk('2',2000,3);commit; end;
/
begin CalcIndex.Main(2000,3,'2'); commit; end;
/
begin GRCal.p_GRIndices(2000,3,'2'); commit; end;
/
begin GRCal.p_GREleComm(2000,3,'2'); commit; end;
/
begin GRCal.p_GREleIndustry(2000,3,'2'); commit; end;
/
begin MonthProc.p_BatchCtlUpd('2',2000,3); commit; end;
/
*/
```