ABAP BASICS

\*user defined data types

data a type i VALUE 123.

\*WRITE a.

\*adding two variables

\* data :lv\_a TYPE i VALUE 10,  
\*      lv\_b TYPE i VALUE 20,  
\*      lv\_c TYPE i.  
\*  
\*lv\_c =  lv\_a + lv\_b.  
\*  
\*WRITE: 'the value is' , lv\_c.  
  
\* special parameters  
\*PARAMETERS:  p\_a TYPE i DEFAULT 10,  
\*             p\_b TYPE i DEFAULT 20.  
\*  
\*data  lv\_c TYPE i.  
\*  
\*lv\_c =  p\_a + p\_b.  
\*  
\*write lv\_c.  
  
\* system defined data types  
  
\*WRITE : / 'The current date is' ,  sy-datum,  
\*         'Time : '  , sy-uzeit,  
\*         'sap user', sy-uname.  
  
data: begin of b1,  " FIELD STRING  
        empid       type i value  123,  
        empname(20) type c value 'UST',  
      end of b1.  
data: begin of b2,  
        empid        type i,  
        empaddrs(30) type c,  
        empname(20)  type c,  
  
      end of b2.  
\*      move b1 to b2.  
move-corresponding b1 to b2.  
write :/ b2-empid , b2-empname , b2-empaddrs.

Control statements

\*&---------------------------------------------------------------------\*  
\*& Report Z353\_ABAP\_CNTL\_STMTS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT Z353\_ABAP\_CNTL\_STMTS.  
  
\* do loop  
\*do 5 times.  
\*  write / 'ust hyd'.  
\*enddo.  
  
\*\*\*\*\*while  
\*  
\*data m type i VALUE 2.  
\*  
\*WHILE  m < 10.  
\*  
\*WRITE / m.  
\*  
\*m = m  + 1.  
\*  
\*ENDWHILE.  
  
\*\*\*\*# 2  
\*\*\*\*\*\*\*EXIT /STOP.  
\*stop - no program will be executed after the stop  
  
\*do.  
\*  write sy-index.  
\*  if sy-index eq 10.  
\*   EXIT.  
\*\*    stop.  
\*  endif.  
\*enddo.  
\*  
\*write / 'SAP UST GLOBAL'.  
  
\*\*\*# 3  
\*\*\*\*\*CONTINUE : it will skip that statement and print the remaining   
\*do.  
\*  if sy-index eq 2.  
\*    continue.  
\*  endif.  
\*  
\*  write sy-index.  
\*  if sy-index eq 10.  
\*    exit.  
\*  endif.  
\*enddo.  
\*write / 'SAP UST GLOBAL'.  
  
\*\*\*\*\*\*Check #4 it will print that statement   
\*do.  
\*    write sy-index.  
\*  if sy-index eq 10.  
\*    exit.  
\*  endif.  
\*  CHECK sy-index eq 2.  
\*    write sy-index.  
\*enddo.  
\*write / 'SAP UST GLOBAL'.  
  
\*\*\*\*\*\*\*\*case  
\*PARAMETERS p\_val type i DEFAULT 2.  
\*  
\*CASE p\_val.  
\*  WHEN 1.  
\*    WRITE : ' first val' , p\_val.  
\*  WHEN 2.  
\*    WRITE :' second val' , p\_val.  
\*  
\*  WHEN OTHERS.  
\*    WRITE :' default val' , p\_val.  
\*  
\*ENDCASE.  
\*  
\*\*WRITE  p\_val.

Select statements

\*&---------------------------------------------------------------------\*  
\*& Report Z353\_ABAP\_SELECT\_STMTS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT Z353\_ABAP\_SELECT\_STMTS.  
  
\* for whole table records will print  
  
\*data: lv\_kunnr TYPE kna1-kunnr, " Customer Number  
\*      lv\_name1 type kna1-name1, " Name 1  
\*      lv\_ort01 TYPE kna1-ort01, " city  
\*      lv\_land1 TYPE kna1-land1. " country  
\*  
\*select kunnr name1 ort01 land1  
\*  into ( lv\_kunnr , lv\_name1 , lv\_ort01 , lv\_land1 )  
\*  from kna1.  
\*  
\*  write : / lv\_kunnr , lv\_name1, lv\_ort01 , lv\_land1.  
\*ENDSELECT.  
  
\* for selecting one entry  
  
\*data: lv\_kunnr TYPE kna1-kunnr, " Customer Number  
\*      lv\_name1 type kna1-name1, " Name 1  
\*      lv\_ort01 TYPE kna1-ort01, " city  
\*      lv\_land1 TYPE kna1-land1. " country  
\*  
\*select kunnr name1 ort01 land1  
\*  into ( lv\_kunnr , lv\_name1 , lv\_ort01 , lv\_land1 )  
\*  from kna1  
\*  where kunnr = '0000000001'.  
\*  
\*  write : / lv\_kunnr , lv\_name1, lv\_ort01 , lv\_land1.  
\*ENDSELECT.  
  
  
\*\*\*\*\*\*# 3  
\*\*  
\*TABLES kna1.  
\*  
\*DATA: lv\_kunnr TYPE kna1-kunnr, " Customer Number  
\*      lv\_name1 TYPE kna1-name1, " Name 1  
\*      lv\_ort01 TYPE kna1-ort01, " city  
\*      lv\_land1 TYPE kna1-land1. " country  
\*  
\*\*\*\*\*PARAMETERS p\_kunnr TYPE kna1-kunnr.  
\*  
\*select-OPTIONS s\_kunnr for lv\_kunnr.  
\*  
\*\*\*\*\*\*\*\*\*\*BREAK-POINT.  
\*  
\*SELECT kunnr name1 ort01 land1  
\*       INTO ( lv\_kunnr, lv\_name1 , lv\_ort01 , lv\_land1 )  
\*       FROM kna1  
\*\*\*\*       WHERE kunnr = p\_kunnr . """""'0000000006'.  
\*        WHERE kunnr in s\_kunnr . """""'0000000006'.  
\*  
\*  WRITE : / lv\_kunnr, lv\_name1 , lv\_ort01 , lv\_land1.  
\*  
\*ENDSELECT.  
\*  
\*  
\*\*\*\*\*# 4  
  
TABLES kna1.  
  
DATA: lv\_kunnr TYPE kna1-kunnr, " Customer Number  
      lv\_name1 TYPE kna1-name1, " Name 1  
      lv\_ort01 TYPE kna1-ort01, " city  
      lv\_land1 TYPE kna1-land1. " country  
  
  
select-OPTIONS s\_kunnr for lv\_kunnr.  " 1  to 100  
PARAMETERS p\_land1 TYPE kna1-land1.  " us  
  
  
SELECT kunnr name1 ort01 land1  
       INTO ( lv\_kunnr, lv\_name1 , lv\_ort01 , lv\_land1 )  
       FROM kna1  
        wHERE kunnr in s\_kunnr and  
              land1 eq p\_land1.  
  
  WRITE : / lv\_kunnr, lv\_name1 , lv\_ort01 , lv\_land1.  
  
ENDSELECT.

Strings

\*&---------------------------------------------------------------------\*

\*& Report ZTRUPEN\_UST\_STRINGS

\*&---------------------------------------------------------------------\*

\*&

\*&---------------------------------------------------------------------\*

REPORT ZTRUPEN\_UST\_STRINGS.

\*\*\*\*\*TRANSLTE #1

\*

TRANSLATE P\_A TO UPPER CASE .

TRANSLATE P\_A TO LOWER CASE .

\*WRITE P\_A.

\*

\*\*\*\* CONCATENATE

\*

\*PARAMETERS: P\_A(10),

\* P\_B(20),

\* P\_DEL.

\*

\*DATA LV\_DEST(30) TYPE C.

\*

\*\*\*\*\*\*CONCATENATE P\_A P\_B INTO LV\_DEST.

\*\*CONCATENATE P\_A P\_B INTO LV\_DEST SEPARATED BY '\*'.

\*

\*\*CONCATENATE P\_A P\_B INTO LV\_DEST SEPARATED BY SPACE.

\*

\*CONCATENATE P\_A P\_B INTO LV\_DEST SEPARATED BY P\_DEL.

\*

\*WRITE LV\_DEST.

\*\*\*\*SPLIT

\*

\*PARAMETERS p\_src(30).

\*

\*data: lv\_a(15), lv\_b(15).

\*

\*SPLIT p\_src at ',' INTO lv\_a lv\_b.

\*

\*WRITE : / lv\_a , lv\_b.

\*\*\*\*\*\*\*offsetting

\*PARAMETERS p\_a(20).

\*

\*data lv\_dest(2).

\*

\*\*lv\_dest = p\_a+3(4).

\*

\*lv\_dest = p\_a+0.

\*

\*

\*WRITE lv\_dest.

\*\*\*\*STRLEN

\*PARAMETERS P\_A(20).

\*

\*DATA LV\_LEN TYPE I.

\*

\*LV\_LEN = STRLEN( P\_A ) .

\*

\*WRITE LV\_LEN.

\*\*\*\*REPLACE

PARAMETERS P\_SRC(20).

\*DO.

IF P\_SRC CA ','.

REPLACE ',' WITH '$' INTO P\_SRC.

ELSE.

EXIT.

ENDIF.

\* ENDDO.

WRITE P\_SRC.

Loops

\*&---------------------------------------------------------------------\*  
\*& Report Z353\_ABAP\_JOINS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT Z353\_ABAP\_JOINS.  
  
\*types: BEGIN OF st\_tab,  
\*        vbeln TYPE vbeln\_va,  
\*         audat TYPE audat,  
\*         vbtyp TYPE vbtypl,  
\*         trvog TYPE trvog,  
\*         auart TYPE auart,  
\*       END OF st\_tab.  
\*  
\*data : it\_tab type STANDARD TABLE OF st\_tab,  
\*      wa\_tab type st\_tab.  
\*  
\*select-OPTIONS s\_vbeln for wa\_tab-vbeln.  
\*PARAMETERS p\_auart type auart.  
\*  
\*START-OF-SELECTION.  
\*  
\*SELECT vbeln audat vbtyp trvog auart  
\*  into table it\_tab  
\*  from vbak  
\*  where vbeln in s\_vbeln and  
\*  auart = p\_auart.  
\*  
\*  
\*end-of-SELECTION.  
\*  
\*loop at it\_tab into wa\_tab.  
\*  write : / wa\_tab-vbeln,  
\*            wa\_tab-audat,  
\*            wa\_tab-vbtyp,  
\*           wa\_tab-trvog,  
\*           wa\_tab-auart.  
\*ENDLOOP.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\* inner join  
  
\*types: BEGIN OF st\_tab,  
\*        vbeln TYPE vbeln\_va, " vbak  
\*         audat TYPE audat,  
\*         vbtyp TYPE vbtypl,  
\*         trvog TYPE trvog,  
\*         auart TYPE auart,  
\*  
\*        posnr TYPE posnr\_va,  "   vbap  
\*         matnr TYPE matnr,  
\*         matwa TYPE MATWA,  
\*       END OF st\_tab.  
\*  
\*data : it\_tab type STANDARD TABLE OF st\_tab,  
\*      wa\_tab type st\_tab.  
\*  
\*select-OPTIONS s\_vbeln for wa\_tab-vbeln.  
\*PARAMETERS p\_auart type auart.  
\*  
\*START-OF-SELECTION.  
\*  
\*SELECT vbak~vbeln audat vbtyp trvog auart posnr matnr matwa  
\*  into table it\_tab  
\*  from vbak  
\*  inner join vbap on  
\*  vbak~vbeln = vbap~vbeln  
\*  where vbak~vbeln in s\_vbeln and  
\*  auart = p\_auart.  
\*  
\*  
\*end-of-SELECTION.  
\*  
\*loop at it\_tab into wa\_tab.  
\*  write : / wa\_tab-vbeln,  
\*            wa\_tab-audat,  
\*            wa\_tab-vbtyp,  
\*           wa\_tab-trvog,  
\*           wa\_tab-auart,  
\*           wa\_tab-posnr,  
\*           wa\_tab-matnr,  
\*           wa\_tab-matwa.  
\*ENDLOOP.  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*inner join with alias  
  
\*\*\*\*\*\*# 2 two tables with alias  
\*TYPES: BEGIN OF ls\_tab,  
\*         vbeln TYPE vbeln\_va,  "  vbak  
\*         audat TYPE audat,  
\*         vbtyp TYPE vbtypl,  
\*         trvog TYPE trvog,  
\*         auart TYPE auart,  
\*  
\*         posnr TYPE posnr\_va,  "   vbap  
\*         matnr TYPE matnr,  
\*         matwa TYPE matwa,  
\*       END OF ls\_tab.  
\*  
\*DATA : lt\_tab TYPE STANDARD TABLE OF ls\_tab,  
\*       wa\_tab TYPE ls\_tab.  
\*  
\*SELECT-OPTIONS s\_vbeln FOR wa\_tab-vbeln.  
\*  
\*START-OF-SELECTION.  
\*  
\*  SELECT so\_h~vbeln  
\*    so\_h~audat  
\*    so\_h~vbtyp  
\*    so\_h~trvog  
\*    so\_h~auart  
\*  
\*     so\_i~posnr  
\*     so\_i~matnr  
\*    so\_i~matwa  
\*  
\*    INTO TABLE lt\_tab  
\*    FROM vbak AS so\_h  
\*  
\*    INNER JOIN vbap AS so\_i  
\*    ON so\_h~vbeln = so\_i~vbeln  
\*  
\*    WHERE so\_h~vbeln IN s\_vbeln.  
\*  
\*  IF sy-subrc EQ 0.  
\*    WRITE : 'sales order info ', sy-dbcnt.  
\*  ELSE.  
\*    WRITE : 'sales order records are not found ', sy-dbcnt.  
\*  ENDIF.  
\*  
\*END-OF-SELECTION.  
\*  LOOP AT lt\_tab INTO wa\_tab.  
\*    WRITE : / wa\_tab-vbeln,  
\*     wa\_tab-posnr,  
\*    wa\_tab-matnr,  
\*    wa\_tab-audat,  
\*    wa\_tab-vbtyp,  
\*    wa\_tab-trvog,  
\*    wa\_tab-auart,  
\*       wa\_tab-matwa.  
\*  
\*  ENDLOOP.  
  
  
\*\*\*\*\*\*\*\*left inner join  
  
  
types: BEGIN OF st\_tab,  
        vbeln TYPE vbeln\_va, " vbak  
         audat TYPE audat,  
         vbtyp TYPE vbtypl,  
         trvog TYPE trvog,  
         auart TYPE auart,  
  
        posnr TYPE posnr\_va,  "   vbap  
         matnr TYPE matnr,  
         matwa TYPE MATWA,  
       END OF st\_tab.  
  
data : it\_tab type STANDARD TABLE OF st\_tab,  
      wa\_tab type st\_tab.  
  
select-OPTIONS s\_vbeln for wa\_tab-vbeln.  
PARAMETERS p\_auart type auart.  
  
START-OF-SELECTION.  
  
SELECT vbak~vbeln audat vbtyp trvog auart posnr matnr matwa  
  into table it\_tab  
  from vbak  
  left outer join vbap on  
  vbak~vbeln = vbap~vbeln  
  where vbak~vbeln in s\_vbeln and  
  auart = p\_auart.  
  
  
end-of-SELECTION.  
  
loop at it\_tab into wa\_tab.  
  write : / wa\_tab-vbeln,  
            wa\_tab-audat,  
            wa\_tab-vbtyp,  
           wa\_tab-trvog,  
           wa\_tab-auart,  
           wa\_tab-posnr,  
           wa\_tab-matnr,  
           wa\_tab-matwa.  
ENDLOOP.

Formatted Tables

\*&---------------------------------------------------------------------\*  
\*& Report Z353\_ABAP\_FINAL\_TAB  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT Z353\_ABAP\_FINAL\_TAB.  
  
  
  
  
TYPES : BEGIN OF ls\_tab,   " LOCAL STRS  
          kunnr TYPE kunnr ,  "  Customer Number.  
          name1 TYPE name1\_gp ,  " name  
          ort01 TYPE ort01\_gp,  " city  
          land1 TYPE land1\_gp,  " Country/Region Key  
        END OF ls\_tab.  
  
DATA : LT\_tAB   TYPE STANDARD TABLE OF ls\_tab, " ITAB  
       wa\_tab   TYPE LS\_tAB,   " WORK AREA  
       lv\_kunnr TYPE kunnr.  
  
  
SELECT-OPTIONS s\_kunnr FOR lv\_kunnr OBLIGATORY.  " 1  to 100  
PARAMETERS p\_land1 TYPE kna1-land1.  " us  
  
START-OF-SELECTION.  
  
  SELECT  kunnr name1 ort01 land1  
    INTO TABLE LT\_tAB  
         FROM kna1  
          WHERE kunnr IN s\_kunnr AND  
                land1 EQ p\_land1.  
  
  IF sy-subrc EQ 0.  
    WRITE : / 'records are found in kna1 table ', sy-dbcnt.  
  ELSE.  
    WRITE : / 'records are not found in kna1 table ', sy-dbcnt.  
  ENDIF.  
\*\*\*\*SKIP .  
SKIP 3.  
  
WRITE : / 'ROW NO',  
         10 'CUST NO',  
         30 'CUST NAME',  
         50 'CITY',  
         65 'Region Key'.  
  
ULINE.  
  
end-of-SELECTION.  
FORMAT COLOR COL\_POSITIVE.  
  
  LOOP AT lt\_Tab INTO WA\_Tab.  " b to W  
    WRITE: / SY-TABIX,  
            10 WA\_tab-kunnr, " COLOR 1,   " W-f  
            30 WA\_tab-name1,  "  COLOR 2,  
            50 WA\_tab-ort01, "  COLOR COL\_GROUP,  
            65 WA\_tab-land1. "  COLOR COL\_POSITIVE.  
  ENDLOOP.  
  
FORMAT COLOR OFF.  
  
BACK.  
SKIP 2.  
    WRITE /120 'UST HYD'.  
BACK.  
SKIP 2.  
    WRITE /130 'UST USA'.  
  
\*

Internal tables

\*&---------------------------------------------------------------------\*  
\*& Report Z353\_ABAP\_TABLES  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT Z353\_ABAP\_TABLES.  
  
  
\*DATA: lv\_kunnr TYPE kunnr, " Customer Number  
\*      lv\_name1 TYPE NAME1\_GP, " Name 1  
\*      lv\_ort01 TYPE ORT01\_GP, " city  
\*      lv\_land1 TYPE LAND1\_GP. " country  
\*  
\*select-OPTIONS s\_kunnr for lv\_kunnr.  
\*PARAMETERS p\_land type land1\_gp.  
\*  
\*  
\*  
\*select kunnr name1 ort01 land1  
\*  into ( lv\_kunnr , lv\_name1 , lv\_ort01 , lv\_land1 )  
\*  from kna1  
\*  where kunnr in s\_kunnr and land1 = p\_land.  
\*  
\*  write : / lv\_kunnr , lv\_name1 , lv\_ort01 , lv\_land1.  
\*  
\*ENDSELECT.  
  
\*\*\*\*\*\*\*\*\*  
\*\*\*\*\*\*\*\*\*with header  
  
\*types: BEGIN OF st\_tab,  
\*      lv\_kunnr TYPE kunnr, " Customer Number  
\*      lv\_name1 TYPE NAME1\_GP, " Name 1  
\*      lv\_ort01 TYPE ORT01\_GP, " city  
\*      lv\_land1 TYPE LAND1\_GP, " country  
\*      end of st\_tab.  
\*  
\*data : it\_tab type st\_tab occurs 0 WITH HEADER LINE,  
\*      lv\_kunnr type kunnr.  
\*  
\*select-OPTIONS s\_kunnr for lv\_kunnr.  
\*PARAMETERS p\_land1 type land1.  
\*  
\*  
\*select kunnr name1 ort01 land1  
\*  into it\_tab  
\*  from kna1  
\*  where kunnr in s\_kunnr and land1 = p\_land1.  
\*  
\*  append it\_tab to it\_tab.  
\*ENDSELECT.  
\*  
\*loop at it\_Tab INTO it\_Tab.  " b to h  
\*\*  
\* WRITE: / it\_tab-lv\_kunnr,   " h-f  
\*          it\_tab-lv\_name1,  
\*          it\_tab-lv\_ort01,  
\*          it\_tab-lv\_land1.  
\*  
\*  ENDLOOP.  
  
  
\*TABLES MARA.  
\*\*DATA : LV\_KUNNR TYPE KUNNR , "CUST NUMBER  
\*\*      LV\_NAME1 TYPE NAME1,  " NAME1  
\*\*      LV\_ORT01 TYPE ORT01,  "CITY  
\*\*      LV\_LAND1 TYPE LAND1.  "COUNTRY  
\*DATA : LT\_TAB TYPE MARA OCCURS 0 WITH HEADER LINE,  
\*\*\*\*\*\*\*\* in this case itab body and header names are same.  
\*        LV\_MATNR TYPE MATNR.  
\*  
\*select-OPTIONS s\_MATNR for lv\_MATNR.  
\*PARAMETERS p\_MTART TYPE MTART.  
\*\*MATNR , MBRSH , MTART , MEINS, MATKL  
\*SELECT \*  
\*  INTO LT\_TAB FROM MARA  
\*  WHERE MATNR IN S\_MATNR AND MTART EQ P\_MTART.  
\*  APPEND LT\_TAB TO LT\_TAB.  
\*ENDSELECT.  
\*  
\*LOOP AT LT\_TAB INTO LT\_TAB.  
\*  WRITE : / lt\_tab-matnr.  
\*  ENDLOOP.  
  
  
\*\*\*\*\*\*\*withput header  
  
types: BEGIN OF st\_tab,  
      lv\_kunnr TYPE kunnr, " Customer Number  
      lv\_name1 TYPE NAME1\_GP, " Name 1  
      lv\_ort01 TYPE ORT01\_GP, " city  
      lv\_land1 TYPE LAND1\_GP, " country  
      end of st\_tab.  
  
data : it\_tab type STANDARD TABLE OF st\_tab,  
      lv\_kunnr type kunnr,  
      wa\_tab type st\_tab.  
  
select-OPTIONS s\_kunnr for lv\_kunnr.  
PARAMETERS p\_land1 type land1.  
  
  
select kunnr name1 ort01 land1  
  into table it\_tab  
  from kna1  
  where kunnr in s\_kunnr and land1 = p\_land1.  
  
  
loop at it\_Tab INTO wa\_Tab.  " b to h  
\*  
 WRITE: / wa\_tab-lv\_kunnr,   " h-f  
          wa\_tab-lv\_name1,  
          wa\_tab-lv\_ort01,  
          wa\_tab-lv\_land1.  
  
  ENDLOOP.