1-------------SAP BASICS--------------1

\*&---------------------------------------------------------------------\*  
\*& Report Z391\_SAPBASICS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT Z391\_SAPBASICS.  
\* 1 - DECLARATION OF VALUE AND ACCESSING THE VALUE  
  
\*DATA A TYPE I VALUE 40.  
\*WRITE A.  
  
\*  2 - DECLARATION STRING AND RETURNING VALUE  
  
\*DATA LV\_VAR(20) TYPE C VALUE 'SUHEL SHAIK'.  
\*WRITE LV\_VAR.  
  
\*3 - DECLARATION USING CHAIN OPERATOR  
\*DATA : A TYPE I VALUE 40,  
\*      B TYPE I VALUE 80,  
\*      C TYPE I.  
\*C = A + B.  
\*WRITE : / A,  
\*B,  
\*C.  
  
  
\* 4 - DECLARATION USING PARAMETERS AND PROCESSING  
\*PARAMETERS : P\_A TYPE I DEFAULT 10,  
\*P\_B TYPE I DEFAULT 30,  
\*P\_C TYPE I.  
\*P\_C = P\_A + P\_B.  
\*  
\*WRITE :/ P\_C.  
  
\* 5 - USING SYSTEM DEFINED DATA TYPES  
  
\*WRITE : / SY-DATUM,  
\*          SY-UZEIT,  
\*          SY-UNAME.  
  
  
\*DATA : BEGIN OF B1, "FIELD STRINGS  
\*  EMPID TYPE I VALUE 123,  
\*  EMPNAME(30) TYPE C VALUE 'UST',  
\*  END OF B1.  
\*  
\*  DATA : BEGIN OF B2,  
\*    EMPID TYPE I,  
\*    EMPNAME(30) TYPE C,  
\*    EMPADDRESS(30) TYPE C VALUE 'SINGARAYAKONDA',  
\*    END OF B2.  
\*    MOVE B1 TO B2.  
\*  
\*WRITE :/ B2-EMPID, B2-EMPNAME, B2-EMPADDRESS.

2---------------SELECT STATEMENTS ------------2

SELECT STATEMENTS

\*&---------------------------------------------------------------------\*  
\*& Report Z391\_SAPBASICS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
\*REPORT Z391\_SAPBASICS.  
\* 1 - DECLARATION OF VALUE AND ACCESSING THE VALUE  
  
\*DATA A TYPE I VALUE 40.  
\*WRITE A.  
  
\*  2 - DECLARATION STRING AND RETURNING VALUE  
  
\*DATA LV\_VAR(20) TYPE C VALUE 'SUHEL SHAIK'.  
\*WRITE LV\_VAR.  
  
\*3 - DECLARATION USING CHAIN OPERATOR  
\*DATA : A TYPE I VALUE 40,  
\*      B TYPE I VALUE 80,  
\*      C TYPE I.  
\*C = A + B.  
\*WRITE : / A,  
\*B,  
\*C.  
  
  
\* 4 - DECLARATION USING PARAMETERS AND PROCESSING  
\*PARAMETERS : P\_A TYPE I DEFAULT 10,  
\*P\_B TYPE I DEFAULT 30,  
\*P\_C TYPE I.  
\*P\_C = P\_A + P\_B.  
\*  
\*WRITE :/ P\_C.  
  
\* 5 - USING SYSTEM DEFINED DATA TYPES  
  
\*WRITE : / SY-DATUM,  
\*          SY-UZEIT,  
\*          SY-UNAME.  
  
  
\*DATA : BEGIN OF B1, "FIELD STRINGS  
\*  EMPID TYPE I VALUE 123,  
\*  EMPNAME(30) TYPE C VALUE 'UST',  
\*  END OF B1.  
\*  DATA : BEGIN OF B2,  
\*    EMPID TYPE I,  
\*    EMPNAME(30) TYPE C,  
\*    EMPADDRESS(30) TYPE C VALUE 'SINGARAYAKONDA',  
\*    END OF B2.  
\*    MOVE B1 TO B2

\*WRITE :/ B2-EMPID, B2-EMPNAME, B2-EMPADDRESS.

3------------ CONTROL STATEMENTS-------------

\*&---------------------------------------------------------------------\*  
\*& Report Z391\_CTRLSTMTS  
\*&---------------------------------------------------------------------\*  
REPORT Z391\_CTRLSTMTS.  
\*1--- DO STATEMETS  
  
\*DO 5 TIMES.  
\*  WRITE :/ 'SUHEL SHAIK'.  
\*  ENDDO.  
  
\*2----- WHILE STATEMENTS.  
  
\*DATA A TYPE I VALUE 10.  
\*WHILE A < 30.  
\* WRITE / A.  
\* A = A + 1.  
\* ENDWHILE.  
  
  
\*3----------- EXIT STOP  
\*do.  
\*  write sy-index.  
\*  if sy-index = 10.  
\*    exit.  
\*    stop.  
\*    endif.  
\*enddo.  
\*write : 'suhel shaik'.  
  
  
\*4---------- CONTINUE------  
\*DO.  
\*  IF SY-INDEX EQ 2.  
\*    CONTINUE.  
\*    ENDIF.  
\*  
\*    WRITE SY-INDEX.  
\*    IF SY-INDEX  EQ 10.  
\*      EXIT.  
\*      ENDIF.  
\*      ENDDO.  
\*WRITE 'SUHEL SHAIK'.  
  
\*5--------- CASE  
\*PARAMETERS P\_VAL  TYPE I DEFAULT 2.  
\*  
\*CASE P\_VAL.  
\*  WHEN 1.  
\*    WRITE : 'FIRST CASE' ,P\_VAL.  
\*  WHEN 2.  
\*    WRITE : 'SECOND CASE', P\_VAL.  
\*  WHEN 3.  
\*    WRITE : 'THIRD CASE', P\_VAL.  
\*  ENDCASE.

*4-------------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.*

*----------4 -----TABLES---------*

*\*&---------------------------------------------------------------------\*  
\*& Report Z391\_TABLES  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT Z391\_TABLES.  
  
\*1------NO ITAB  
\*  
\*TABLES KNA1.  
\*DATA : LV\_KUNNR TYPE KUNNR , "CUST NUMBER  
\*      LV\_NAME1 TYPE NAME1,  " NAME1  
\*      LV\_ORT01 TYPE ORT01,  "CITY  
\*      LV\_LAND1 TYPE LAND1.  "COUNTRY  
\*select-OPTIONS s\_kunnr for lv\_kunnr.  
\*PARAMETERS p\_land1 TYPE kna1-land1.  
\*  
\*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.  
  
\*2------- WITH HEADER LINE  
\*  
\*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.  
\*DATA LV\_MATNR TYPE MATNR.  
\*select-OPTIONS s\_MATNR for lv\_MATNR.  
\*PARAMETERS p\_MTART TYPE MARA-MTART.  
\*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 :/ MARA-MATNR,  
\*          MARA-MBRSH,  
\*          MARA-MTART,  
\*          MARA-MEINS.  
\*  ENDLOOP.  
  
  
\*\*\*\*# 3   without header  
\*TABLES KNA1.  
\*DATA : LT\_TAB TYPE KNA1 OCCURS 0 WITH HEADER LINE.  
\*DATA LV\_KUNNR TYPE KUNNR.  
\*SELECT-OPTIONS S\_KUNNR FOR LV\_KUNNR.  
\*PARAMETERS P\_LAND1 TYPE KNA1-LAND1.  
\*  
\*SELECT \* INTO TABLE LT\_TAB  
\*  FROM KNA1 WHERE KUNNR  
\*  IN S\_KUNNR  
\*  AND LAND1 = P\_LAND1.  
\*  
\*  LOOP AT LT\_TAB INTO LT\_TAB.  
\*  WRITE : /  
\*        LT\_TAB-KUNNR,  
\*        LT\_TAB-NAME1,  
\*        LT\_TAB-ORT01,  
\*        LT\_TAB-LAND1.  
\*  ENDLOOP.  
  
\*\*\*\*# 4 WITH USER DEFINED WORK AREA  
  
  
\*TABLES kna1.  
\*  
\*TYPES : BEGIN OF LS\_TAB,  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  KUNNR TYPE KNA1-KUNNR. " V TYPE T-F  
\*      lv\_KUNNR TYPE KUNNR ,  "  V TYPE DTL.  
\*      NAME1 TYPE NAME1\_GP ,  
\*      ORT01 TYPE ORT01\_GP,  
\*      LAND1 TYPE LAND1\_GP,  
\*      END OF LS\_TAB.  
\*  
\*DATA : LT\_tAB TYPE STANDARD TABLE OF LS\_TAB, " ITAB  
\*       WA\_TAB TYPE LS\_tAB.   " WORK AREA  
\*  
\*data lv\_kunnr TYPE kunnr.  
\*select-OPTIONS s\_kunnr for lv\_kunnr.  " 1  to 100  
\*PARAMETERS p\_land1 TYPE kna1-land1.  " us  
\*  
\*SELECT  KUNNR NAME1 ORT01 LAND1  
\*       INTO WA\_Tab   " into WORK AREA  
\*       FROM kna1  
\*        wHERE kunnr in s\_kunnr and  
\*              land1 eq p\_land1.  
\*  
\* APPEND WA\_tab to lt\_tab.  " W to b  
\* ENDSELECT.  
\*  
\*loop at lt\_Tab INTO WA\_Tab.  " b to W  
\*  
\* WRITE: / WA\_tab-lv\_kunnr,   " W-f  
\*          WA\_tab-name1,  
\*          WA\_tab-ort01,  
\*          WA\_tab-land1.  
\*  
\*  ENDLOOP.  
  
  
\*------------------------ FINAL CODE ----------------------  
  
\*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'.*

*5-------------- JOINS------------*

*\*&---------------------------------------------------------------------\*  
\*& Report Z391\_JOINS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT Z391\_JOINS.  
\*\*\*\*\*# 1---------------   TABLE--------------  
\*TYPES: BEGIN OF ls\_tab,  
\*         vbeln TYPE vbeln\_va,  
\*         audat TYPE audat,  
\*         vbtyp TYPE vbtypl,  
\*         trvog TYPE trvog,  
\*         auart TYPE auart,  
\*       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 vbeln audat vbtyp trvog auart  
\*    INTO TABLE lt\_tab  
\*    FROM vbak  
\*    WHERE 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-audat,  
\*    wa\_tab-vbtyp,  
\*    wa\_tab-trvog,  
\*    wa\_tab-auart.  
\*ENDLOOP.  
  
  
\*\*\*\*\*\*# 2 two tables  
\*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 vbak~vbeln  
\*    vbak~audat  
\*    vbak~vbtyp  
\*    vbak~trvog  
\*    vbak~auart  
\*  
\*     vbap~posnr  
\*     vbap~matnr  
\*    vbap~matwa  
\*  
\*    INTO TABLE lt\_tab  
\*    FROM vbak  
\*  
\*    INNER JOIN vbap  
\*    ON vbak~vbeln = vbap~vbeln  
\*  
\*    WHERE vbak~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.*