S A P ABAP PROGRAMMING LANGUAGE

**CONTROL STATEMENTS:**

REPORT zcontrol\_stmts.

PARAMETERS p\_a(2) TYPE c.

PARAMETERS : a type i,

b type i.

\*IF p\_a < 7.

\* WRIte 'less than 7'.

\*elseif p\_a > 7 .

\* WRITE 'greater than 7'.

\* elseif p\_a = 7.

\* write 'eq'.

\*ENDIF.

\*

\*WHILE p\_a <= 100.

\* WRITE 'kalesha'.

\* p\_a = p\_a + 5.

\*ENDWHILE.

\*

\*WHILE p\_a < 11.

\* DATA(h) = a \* p\_a.

\* WRITE : 'a \* ', p\_a ,'=', h.

\* p\_a = p\_a + 1. "tables maths

\* ENDWHILE.

\*do.

\* write 'kalesha'.

\* if p\_a > 10.

\* exit.

\* ENDIF.

\* p\_a = p\_a + 1.

\* ENDDO.

\*while p\_a < 100.

\* write / a.

\* a = a + 3.

\* if a = 12.

\* CONTINUE.

\* ENDIF.

\*

\* p\_a = p\_a \* 3.

\* ENDWHILE.

CASE p\_a.

WHEN '+'.

DATA(h) = a + b.

write h .

WHEN '-'.

DATA(t) = a - b.

write t.

WHEN OTHERS.

WRITE 'it is not found'.

ENDCASE.

**DATA TYPE:**

REPORT ZDATATYPE.

data LV\_a type i value 45.

data LV\_b type i value 67.

data LV\_c type i.

LV\_c = LV\_a + LV\_b.

write : 'Adding ',LV\_c.

**INTERNAL TABLE 1:**

REPORT zinternal\_table1\_kal.

\*TABLES lfa1.

\*TYPES : BEGIN OF i\_tab,

\* lifnr TYPE lifnr,

\* land TYPE land1\_gp,

\* name1 TYPE NAME1\_Gp,

\* city TYPE ort01\_gp,

\* END OF i\_tab.

\*

\*DATA : in\_tab TYPE STANDARD TABLE OF i\_tab,

\* wa\_tab TYPE i\_tab,

\* lv\_lifnr TYPE lifnr.

\*

\*PARAMETERS p\_city TYPE land1\_gp.

\*SELECT-OPTIONS s\_lifnr FOR lv\_lifnr.

\*

\*SELECT lifnr land1 name1 ort01 INTO TABLE in\_tab FROM lfa1

\* WHERE LAND1 = p\_city AND lifnr IN s\_lifnr.

\*

\*LOOP AT in\_tab INTO wa\_tab.

\* WRITE : / wa\_tab-lifnr,

\* wa\_tab-land,

\* wa\_tab-name1,

\* wa\_tab-city.

\*

\*ENDLOOP.

**\*\*\*\* By using Structure and table type**

DATA : in\_tab TYPE TABLE OF ZKAL\_ST,

wa\_tab TYPE ZKAL\_ST,

lv\_lifnr TYPE lifnr.

PARAMETERS p\_city TYPE land1\_gp.

SELECT-OPTIONS s\_lifnr FOR lv\_lifnr.

SELECT lifnr land1 name1 ort01 INTO TABLE in\_tab FROM lfa1

WHERE LAND1 = p\_city AND lifnr IN s\_lifnr.

LOOP AT in\_tab INTO wa\_tab.

WRITE : / wa\_tab-lifnr,

wa\_tab-land1,

wa\_tab-name1,

wa\_tab-ort01.

ENDLOOP.

**INTERNAL TABLE JOIN:**

REPORT ZINTERNAL\_TABLE\_JOINS NO STANDARD PAGE HEADING.

types : begin of s\_tab,

vbeln type VBELN\_va,

ERDAT type ERDAT,

ERZET type ERZET,

ERNAM type ERnam,

ANGDT type ANGDT,

POSNR type Posnr,

MATNR type matnr,

MATWA type matwa,

PMATN type pmatn,

end of s\_tab.

data : i\_tab type STANDARD TABLE OF s\_tab,

wa\_tab type s\_tab.

ULINE.

select-OPTIONS s\_vbeln for wa\_tab-vbeln.

PARAMETERS p\_ernam type ernam.

START-OF-SELECTION.

SELECT vbak~VBELN

vbak~ERDAT vbak~ERZET

vbak~ERNAM

vbak~ANGDT

vbap~POSNR

vbap~MATNR

vbap~MATWA

vbap~PMATN into TABLE i\_tab from vbak INNER JOIN vbap

on vbak~vbeln = vbap~vbeln

where vbak~vbeln in s\_vbeln and

vbap~ernam eq p\_ernam.

skip.

skip.

SORT i\_tab by vbeln.

if sy-subrc eq 0.

WRITE : / 'it is working dude' ,sy-dbcnt.

else.

WRITE : / 'it is not working dude'.

ENDIF.

end-of-SELECTION.

ULINE.

FORMAt COLOR COL\_GROUP.

LOOP AT i\_tab into wa\_tab.

WRITE : /

wa\_tab-vbeln,

wa\_tab-erdat,

wa\_tab-erzet,

wa\_tab-ernam,

wa\_tab-angdt,

wa\_tab-POSNR,

wa\_tab-MATNR,

wa\_tab-MATWA,

wa\_tab-PMATN.

ENDLOOP.

FORMAT COLOR off.

uline.

**SELECT MYTABLE:**

REPORT ZSELECT\_MYTABLE.

TABLES ZKALESHA\_TABLE1C.

DATA : lv\_collegeid type ZCOLLEGE\_ID\_KAL,

lv\_dept TYPE ZDEPT\_KAL,

lv\_name TYPE ZNAME\_KAL,

lv\_phn TYPE ZPHN\_KAL.

PARAMETERS p\_colle TYPE ZCOLLEGE\_ID\_KAL.

select-OPTIONS s\_phn for lv\_phn.

SELECT COLLEGE\_ID\_KAL DEPT\_KAL NAME\_KAL PHN\_KAL into

(lv\_collegeid,lv\_dept,lv\_name,lv\_phn) from ZKALESHA\_TABLE1C

WHERE COLLEGE\_ID\_KAL = p\_colle and PHN\_KAL in s\_phn.

WRITE : / lv\_collegeid,lv\_dept,lv\_name,lv\_phn.

ENDSELECT.

**SELECT TABLE 2:**

REPORT ZSELECT\_MYTABLE2.

TABLES ZKALESHA\_TABLE2C.

DATA : lv\_uid TYPE ZUST\_UID\_KAL,

lv\_name TYPE ZUST\_NAME,

lv\_phn TYPE ZUST\_PHN,

lv\_add TYPE ZUST\_ADDRESS,

lv\_marks TYPE ZUST\_MARKS.

PARAMETERS p\_add TYPE ZUST\_ADDRESS.

SELECT-OPTIONS s\_phn for lv\_phn.

SELECT UST\_ID UST\_CANDI\_NAME UST\_CANDI\_PHN UST\_ADDRESS UST\_MARKS into

(lv\_uid,lv\_name,lv\_phn,lv\_add,lv\_marks) from ZKALESHA\_TABLE2C

WHERE UST\_ADDRESS = p\_add and UST\_CANDI\_PHN in s\_phn.

write :/ lv\_uid,lv\_name,lv\_phn,lv\_add,lv\_marks.

ENDSELECT.

**SELECT STATEMENT 3:**

REPORT ZSELECT\_STMT.

TABLES MARA. "MASTER TABLE

DATA : lv\_matnr TYPE MATNR,

lv\_ersda type ERSDA,

lv\_createdAtTime type CREATED\_AT\_TIME,

lv\_ernam type ERNAM,

lv\_laeda TYPE LAEDA.

PARAMETERS number TYPE Mara-ERSDA.

SELECT-OPTIONS s\_matnr for lv\_matnr.

SELECT MATNR ERSDA CREATED\_AT\_TIME ERNAM LAEDA into (lv\_matnr,lv\_ersda,

lv\_createdAtTime,lv\_ernam,lv\_laeda) from MARA

where ERSDA eq number AND MATNR in s\_matnr.

WRITE : / lv\_matnr,lv\_ersda,lv\_createdAtTime,lv\_ernam,lv\_laeda.

ENDSELECT.

**STRING:**

REPORT zstring\_func NO STANDARD PAGE HEADING.

\*PARAMETERS : p\_a(20) LOWER CASE OBLIGATORY,

\* p\_b(30) OBLIGATORY.

\*TRANSLATE p\_a TO UPPER CASE.

\*WRITE p\_a.

\*DATA lv\_c(60).

\*CONCATENATE p\_a p\_b into lv\_c SEPARATED BY space.

\*write lv\_c.

\*PARAMETERS p\_a(60).

\*DATA : lv\_b(30),

\* lv\_c(20).

\*SPLIT p\_a at ',' into lv\_b lv\_c.

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

\*PARAMETERS p\_a(60).

\*DATA lv\_b(20).

\*lv\_b = p\_a+0(7).

\*WRITE lv\_b.

\*PARAMETERS p\_a(60).

\*DATA lv\_b type i.

\*lv\_b = strlen( p\_a ).

\*WRITE lv\_b.

\*PARAMETERS p\_a(60).

\*DATA lv\_b type i.

\*lv\_b = strlen( p\_a ).

\*data m type i.

\*while m lt lv\_b.

\* IF p\_a ca ','.

\* REPLACE ',' WITH ' ' into p\_a.

\*else.

\* exit.

\* ENDIF.

\* m = m + 1.

\* endwhile.

\* WRITE p\_a.

\*PARAMETERS p\_a(20).

\*

\*SHIFT p\_a CIRCULAR.

\*WRITE p\_a.

**CLASSICAL INTERNAL TABLES :**

1. REPORT zkalesha\_innerjoin\_classic NO STANDARD PAGE HEADING LINE-COUNT 25(3) LINE-SIZE 500.  
   TYPES : BEGIN OF ls\_tab,  
             vbeln TYPE vbeln\_va,  
             erdat TYPE erdat,  
             angdt TYPE angdt\_v,  
             auart TYPE auart,  
             matnr TYPE matnr,  
             matkl TYPE matkl,  
           END OF ls\_tab.  
     
   DATA : i\_tab    TYPE STANDARD TABLE OF ls\_tab,  
          wa\_tab   TYPE ls\_tab,  
          lv\_vbeln TYPE vbeln\_va.  
   SELECT-OPTIONS s\_vbeln FOR lv\_vbeln OBLIGATORY.  
     
   INITIALIZATION.  
     s\_vbeln-low = '1'.  
     s\_vbeln-high = '100'.  
     APPEND s\_vbeln.  
     CLEAR : i\_tab,wa\_tab.  
     
   AT SELECTION-SCREEN ON s\_vbeln.  
     SELECT SINGLE vbak~vbeln INTO lv\_vbeln FROM vbak INNER JOIN vbap  
       ON vbak~vbeln = vbap~vbeln  
       WHERE vbak~vbeln IN S\_VBELn.  
     IF sy-subrc EQ 0.  
       MESSAGE s000(zkalesha\_message).  
     ELSE.  
       MESSAGE e001(zkalesha\_message).  
     ENDIF.  
     
   START-OF-SELECTION.  
     SELECT vbak~vbeln vbak~erdat vbak~angdt vbak~auart vbap~matnr vbap~matkl INTO TABLE i\_tab FROM vbak  
       INNER JOIN vbap  
       ON vbak~vbeln = vbap~vbeln  
       WHERE vbak~vbeln IN s\_vbeln.  
     
     IF sy-subrc EQ 0.  
       WRITE : 'RECORDS FOUND',sy-dbcnt.  
     ELSE.  
       WRITE 'not found'.  
     ENDIF.  
     
   TOP-OF-PAGE.  
     ULINE.  
     FORMAT COLOR COL\_POSITIVE.  
     WRITE : 'VBELN',sy-vline,20 'ERDAT',sy-vline,50 'ANGDT',sy-vline,80 'AUART',sy-vline,120 'MATNR',sy-vline,140 'MATKL',sy-vline.  
     FORMAT COLOR OFF.  
     ULINE.  
     
   END-OF-PAGE.  
     WRITE : / 'DATE :',sy-datum,'TIME :',sy-uzeit,'PAGENO :',sy-pagno.  
     
   END-OF-SELECTION.  
     FORMAT COLOR 1.  
     LOOP AT i\_tab INTO wa\_tab.  
       WRITE : /  
       sy-vline,  
       wa\_tab-vbeln COLOR 4,sy-vline,  
       20 wa\_tab-erdat,sy-vline,  
       40 wa\_tab-angdt,sy-vline,  
       70 wa\_tab-auart,sy-vline,  
       100 wa\_tab-matnr,sy-vline,  
      140 wa\_tab-matkl,  
       sy-vline.  
     
     ENDLOOP.  
     FORMAT COLOR OFF.

**2.**

REPORT ZKALESHA\_CLASSIC\_REP NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500.  
include ZKALESHA\_CLASSIC\_TD.  
include ZKALESHA\_CLASSIC\_EVENTS.  
  
PERFORM PRINT.  
  
\*&---------------------------------------------------------------------\*  
\*& Form DEFALUT  
  
FORM defalut .  
s\_vbeln-low = '1'.  
s\_vbeln-high = '100'.  
APPEND s\_vbeln.  
clear :  i\_tab,wa\_tab.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form SELECTION\_SCREEN\_VALID  
  
FORM selection\_screen\_valid .  
SELECT SINGLE vbeln into lv\_vbeln from vbak where vbeln in s\_vbeln.  
    if sy-subrc eq 0.  
      MESSAGE S000(ZKALESHA\_MESSAGE).  
      else.  
        MESSAGE E001(ZKALESHA\_MESSAGE).  
       ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form SELECT\_QUERY  
  
FORM select\_query .  
SELECT VBELN ERDAT ERZET ERNAM ANGDT into TABLE i\_tab from vbak  
  WHERE VBELN in s\_vbeln.  
  
  if sy-subrc eq 0.  
    WRITE : / 'Records are found',sy-dbcnt.  
    else.  
      WRITE : / 'Records are not found'.  
      ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form PRINT  
  
FORM print .  
  FORMAT COLOR 1.  
loop at i\_tab into wa\_tAB.  
 uline.  
  write : / wa\_tab-vbeln,sy-vline,  
  20 wa\_tab-erdat,sy-vline,  
  40 wa\_tab-erzet,sy-vline,  
  60 wa\_tab-ernam,sy-vline,  
  100 wa\_tab-angdt,sy-vline.  
  
  ENDLOOP.  
ENDFORM.

**3.**

REPORT zkalesha\_kna\_pm\_prog.  
include ZKALESHA\_INCLUDE\_KNA\_HOLE.

# **TABLES:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**VIEWS :**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated