REPORT Z392\_BASICS.  
  
\*data a type i value 20.  
\*write a.  
  
\*data b type i value 123.  
\*write b.  
  
\*#2  
\*data lv\_a(20) type c value 'dharani'.  
\*write lv\_a.  
  
\*data lv\_d(30) type c value 'hello'.  
\*write lv\_d.  
  
\*#3  
\*data : lv\_a type i value 10,  
\*       lv\_b type i value 20,  
\*       lv\_c type i.  
\*lv\_c = lv\_a + lv\_b.  
\*write  lv\_c.  
  
\*data : lv\_a type i value 30,  
\*      lv\_b type i value 40,  
\*      lv\_c type i.  
\*lv\_c = lv\_a + lv\_b.  
\*write : 'lv\_c of is', lv\_c.  
  
\*#4  
\*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.  
  
\*parameters: p\_a type i default 50,  
\*            p\_b type i DEFAULT 50.  
\*data: lv\_c type i.  
\*lv\_c = p\_a + p\_b.  
\*write: 'lv\_c of is', lv\_c.  
  
\*#5  
\*write: 'the date is', sy-datum.  
  
\*write : 'time', sy-uzeit,  
\*        'user', sy-uname,  
\*        'logon lan', sy-langu.  
  
\*data: begin of b1,  
\*      empid type i value 123,  
\*      empname(7) type c value 'dharani',  
\*  end of b1.  
\*data: begin of b2,  
\*     empid type i,  
\*     empadd(30) type c,  
\*     empname(7) type c,  
\* end of b2.  
\* move b1 to b2.  
\* write :/ b2-empid,b2-empname,b2-empadd.

REPORT Z392\_CNTLSTMTS.  
  
\*#1  
\*do 5 times.  
\*  write:/ 'dharani'.  
\*enddo.  
  
\*do 10 times.  
\*  write:/ 'ust global'.  
\*enddo.  
  
\*\*while  
\*data m type i value 1.  
\*while m <  10.  
\*  write / m.  
\*m = m + 1.  
\*ENDWHILE.  
  
\*data m type i value 5.  
\*while m < 50.  
\*   write / m.  
\*  m = m + 5.  
\*  ENDWHILE.  
  
\*data a type i value 2.  
\*while a < 101.  
\*  write / a.  
\* a = a + 2.  
\* ENDWHILE.  
  
\*\*exit/stop  
\*do.  
\*  write sy-index.  
\*  if sy-index eq 10.  
\* exit.  
\* stop.  
\* ENDIF.  
\* ENDDO.  
\* write / 'ust'.  
  
\*do.  
\*  write sy-index.  
\*  if sy-index eq 5.  
\*    exit.  
\*    stop.  
\*    ENDIF.  
\*    ENDDO.  
\*  
  
\*\*continue  
\*do.  
\*  if sy-index eq 2.  
\*    continue.  
\*   ENDIF.  
\*   write sy-index.  
\*   if sy-index eq 10.  
\*     exit.  
\*     ENDIF.  
\*     enddo.  
  
  
\*do.  
\*  if sy-index eq 5.  
\*  CONTINUE.  
\*  ENDIF.  
\* write sy-index.  
\* if sy-index eq 10.  
\*   exit.  
\*   endif.  
\*   enddo.  
  
  
\*check  
\*do.  
\*  write sy-index.  
\*  if sy-index eq 10.  
\*    exit.  
\*    ENDIF.  
\*    check sy-index eq 5.  
\*    write sy-index.  
\*    ENDDO.  
  
  
\*parameters  
\*parameters p\_val type i default 100.  
\*case p\_val.  
\*  when 1.  
\*    write : 'first value ', p\_val.  
\*   when 2.  
\*     write : 'second val', p\_val.  
\*  
\*   when others.  
\*     write: 'default val', p\_val.  
\* endcase.  
\* write p\_val.

*\*tables kna1.*  
*\**  
*\*data : lv\_kunnr type kunnr,*  
*\*       lv\_name1 type name1\_GP,*  
*\*       lv\_ort01 type ort01\_GP,*  
*\*       LV\_LAND1 type land1\_gp.*  
*\**  
*\*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.*  
  
  
*\*tables kna1.*  
*\**  
*\*data : it\_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 it\_tab*  
*\*  from kna1*  
*\*  where kunnr in s\_kunnr and land1 eq p\_land1.*  
*\**  
*\*  append it\_tab to it\_tab.*  
*\*  endselect.*  
*\**  
*\*  loop at it\_tab into it\_tab.*  
*\**  
*\*  write : / it\_tab-kunnr,it\_tab-name1,it\_tab-ort01,it\_tab-land1.*  
*\*  ENDloop.*  
*\**  
  
  
*\*tables kna1.*  
*\**  
*\*data : it\_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 it\_tab*  
*\*  from kna1*  
*\*  where kunnr in s\_kunnr and land1 eq p\_land1.*  
*\**  
*\**  
*\*  loop at it\_tab into it\_tab.*  
*\**  
*\*  write : / it\_tab-kunnr,it\_tab-name1,it\_tab-ort01,it\_tab-land1.*  
*\*  ENDloop.*  
*\*  ENDSELECT.*  
  
  
  
*\*tables kna1.*  
*\**  
*\*types : begin of ls\_tab,*  
*\*  lv\_kunnr type kunnr,*  
*\*  name1 type name1,*  
*\*  ort01 type ort01,*  
*\*  land1 type land1,*  
*\*  end of ls\_tab.*  
*\**  
*\**  
*\*data : lt\_tab type STANDARD TABLE OF ls\_tab,*  
*\*      wa\_tab type ls\_tab.*  
*\**  
*\**  
*\*data lv\_kunnr type kunnr.*  
*\*SELECT-OPTIONS s\_kunnr for lv\_kunnr.*  
*\*PARAMETERS p\_land1 type kna1-land1.*  
*\**  
*\**  
*\**  
*\*select kunnr name1 ort01 land1*  
*\*  into wa\_tab*  
*\*  from kna1*  
*\*  where kunnr in s\_kunnr and land1 eq p\_land1.*  
*\**  
*\*  APPEND wa\_tab to lt\_tab.*  
*\*  endselect.*  
*\**  
*\**  
*\*  loop at lt\_tab into wa\_tab.*  
*\**  
*\*  write : / wa\_tab-lv\_kunnr,wa\_tab-name1,wa\_tab-ort01,wa\_tab-land1.*  
*\*  ENDloop.*  
  
  
tables kna1.  
  
types : begin of ls\_tab,  
  lv\_kunnr type kunnr,  
  name1 type name1,  
  ort01 type ort01,  
  land1 type land1,  
  end of ls\_tab.  
  
  
data : lt\_tab type STANDARD TABLE OF ls\_tab,  
      wa\_tab type ls\_tab.  
  
  
data lv\_kunnr type kunnr.  
SELECT-OPTIONS s\_kunnr for lv\_kunnr.  
PARAMETERS p\_land1 type kna1-land1.  
  
  
  
select kunnr name1 ort01 land1  
  into wa\_tab  
  from kna1  
  where kunnr in s\_kunnr and land1 eq p\_land1.  
  
  APPEND wa\_tab to lt\_tab.  
  endselect.  
  
  
  loop at lt\_tab into wa\_tab.  
  
  write : / wa\_tab-lv\_kunnr,wa\_tab-name1,wa\_tab-ort01,wa\_tab-land1.  
  ENDloop.

*\*\*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(10) , LV\_B(10).*  
*\*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).*  
*\**  
*\*IF P\_SRC CA ','.*  
*\*  REPLACE ',' WITH '$' INTO P\_SRC.*  
*\*  ELSE.*  
*\*    EXIT.*  
*\*    ENDIF.*  
*\*    ENDDO.*  
*\*    WRITE P\_SRC.*  
*\**  
*\**  
*\**

REPORT Z392\_JOINS.  
  
*\*#1*  
*\*TYPES : BEGIN OF LS\_TAB,*  
*\*  VBELN TYPE VBELN\_VA,*  
*\*  AUDAT TYPE audat,*  
*\*  vbtyp type vbtyp,*  
*\*  trvog type trvog,*  
*\*  auart type auart,*  
*\*  end of LS\_TAB.*  
*\**  
*\*DATA: IT\_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 IT\_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 IT\_TAB INTO WA\_TAB.*  
*\*    WRITE:/ WA\_TAB-VBELN, WA\_TAB-AUDAT,WA\_TAB-VBTYP,WA\_TAB-TRVOG,WA\_TAB-AUART.*  
*\*    ENDLOOP.*  
*\**  
  
*\*# WITH 2 TABLES*  
*\*TYPES : BEGIN OF LS\_TAB,*  
*\*  VBELN TYPE VBELN\_VA,*  
*\*  AUDAT TYPE audat,*  
*\*  vbtyp type vbtyp,*  
*\*  trvog type trvog,*  
*\*  auart type auart,*  
*\**  
*\*  POSNR TYPE POSNR\_VA,*  
*\*  MATNR TYPE MATNR,*  
*\*  MATWA TYPE MATWA,*  
*\*  end of LS\_TAB.*  
*\**  
*\*DATA: IT\_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 IT\_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 IT\_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.*  
  
*\*#32 TABLES WITH ALIAS*  
*\*TYPES : BEGIN OF LS\_TAB,*  
*\*  VBELN TYPE VBELN\_VA,*  
*\*  AUDAT TYPE audat,*  
*\*  vbtyp type vbtyp,*  
*\*  trvog type trvog,*  
*\*  auart type auart,*  
*\**  
*\*  POSNR TYPE POSNR\_VA,*  
*\*  MATNR TYPE MATNR,*  
*\*  MATWA TYPE MATWA,*  
*\*  end of LS\_TAB.*  
*\**  
*\*DATA: IT\_TAB TYPE STANDARD TABLE OF LS\_TAB,*  
*\*      WA\_TAB TYPE LS\_TAB.*  
*\**  
*\*SELECT-OPTIONS S\_VBELN FOR WA\_TAB-VBELN.*  
*\**  
*\*START-OF-SELECTION.*  
*\*  SELECT A\_S~VBELN*  
*\*    A\_S~AUDAT*  
*\*    A\_S~VBTYP*  
*\*    A\_S~TRVOG*  
*\*    A\_S~AUART*  
*\**  
*\*    A\_I~POSNR*  
*\*    A\_I~MATNR*  
*\*    A\_I~MATWA*  
*\**  
*\*    INTO TABLE IT\_TAB*  
*\*    FROM VBAK AS A\_S*  
*\*    INNER JOIN VBAP AS A\_I ON A\_S~VBELN = A\_I~VBELN*  
*\*    WHERE A\_S~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 IT\_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 outer join*  
  
*\*TYPES : BEGIN OF LS\_TAB,*  
*\*  VBELN TYPE VBELN\_VA,*  
*\*  AUDAT TYPE audat,*  
*\*  vbtyp type vbtyp,*  
*\*  trvog type trvog,*  
*\*  auart type auart,*  
*\**  
*\*  POSNR TYPE POSNR\_VA,*  
*\*  MATNR TYPE MATNR,*  
*\*  MATWA TYPE MATWA,*  
*\*  end of LS\_TAB.*  
*\**  
*\*DATA: IT\_TAB TYPE STANDARD TABLE OF LS\_TAB,*  
*\*      WA\_TAB TYPE LS\_TAB.*  
*\**  
*\*SELECT-OPTIONS S\_VBELN FOR WA\_TAB-VBELN.*  
*\**  
*\*START-OF-SELECTION.*  
*\*  SELECT A\_S~VBELN*  
*\*    A\_S~AUDAT*  
*\*    A\_S~VBTYP*  
*\*    A\_S~TRVOG*  
*\*    A\_S~AUART*  
*\**  
*\*    A\_I~POSNR*  
*\*    A\_I~MATNR*  
*\*    A\_I~MATWA*  
*\**  
*\*    INTO TABLE IT\_TAB*  
*\*    FROM VBAK AS A\_S*  
*\*    left outer JOIN VBAP AS A\_I ON A\_S~VBELN = A\_I~VBELN*  
*\*    WHERE A\_S~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 IT\_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.*

REPORT Z392\_JOINS.  
  
*\*#1*  
*\*TYPES : BEGIN OF LS\_TAB,*  
*\*  VBELN TYPE VBELN\_VA,*  
*\*  AUDAT TYPE audat,*  
*\*  vbtyp type vbtyp,*  
*\*  trvog type trvog,*  
*\*  auart type auart,*  
*\*  end of LS\_TAB.*  
*\**  
*\*DATA: IT\_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 IT\_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 IT\_TAB INTO WA\_TAB.*  
*\*    WRITE:/ WA\_TAB-VBELN, WA\_TAB-AUDAT,WA\_TAB-VBTYP,WA\_TAB-TRVOG,WA\_TAB-AUART.*  
*\*    ENDLOOP.*  
*\**  
  
*\*# WITH 2 TABLES*  
*\*TYPES : BEGIN OF LS\_TAB,*  
*\*  VBELN TYPE VBELN\_VA,*  
*\*  AUDAT TYPE audat,*  
*\*  vbtyp type vbtyp,*  
*\*  trvog type trvog,*  
*\*  auart type auart,*  
*\**  
*\*  POSNR TYPE POSNR\_VA,*  
*\*  MATNR TYPE MATNR,*  
*\*  MATWA TYPE MATWA,*  
*\*  end of LS\_TAB.*  
*\**  
*\*DATA: IT\_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 IT\_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 IT\_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.*  
  
*\*#32 TABLES WITH ALIAS*  
*\*TYPES : BEGIN OF LS\_TAB,*  
*\*  VBELN TYPE VBELN\_VA,*  
*\*  AUDAT TYPE audat,*  
*\*  vbtyp type vbtyp,*  
*\*  trvog type trvog,*  
*\*  auart type auart,*  
*\**  
*\*  POSNR TYPE POSNR\_VA,*  
*\*  MATNR TYPE MATNR,*  
*\*  MATWA TYPE MATWA,*  
*\*  end of LS\_TAB.*  
*\**  
*\*DATA: IT\_TAB TYPE STANDARD TABLE OF LS\_TAB,*  
*\*      WA\_TAB TYPE LS\_TAB.*  
*\**  
*\*SELECT-OPTIONS S\_VBELN FOR WA\_TAB-VBELN.*  
*\**  
*\*START-OF-SELECTION.*  
*\*  SELECT A\_S~VBELN*  
*\*    A\_S~AUDAT*  
*\*    A\_S~VBTYP*  
*\*    A\_S~TRVOG*  
*\*    A\_S~AUART*  
*\**  
*\*    A\_I~POSNR*  
*\*    A\_I~MATNR*  
*\*    A\_I~MATWA*  
*\**  
*\*    INTO TABLE IT\_TAB*  
*\*    FROM VBAK AS A\_S*  
*\*    INNER JOIN VBAP AS A\_I ON A\_S~VBELN = A\_I~VBELN*  
*\*    WHERE A\_S~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 IT\_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 outer join*  
  
*\*TYPES : BEGIN OF LS\_TAB,*  
*\*  VBELN TYPE VBELN\_VA,*  
*\*  AUDAT TYPE audat,*  
*\*  vbtyp type vbtyp,*  
*\*  trvog type trvog,*  
*\*  auart type auart,*  
*\**  
*\*  POSNR TYPE POSNR\_VA,*  
*\*  MATNR TYPE MATNR,*  
*\*  MATWA TYPE MATWA,*  
*\*  end of LS\_TAB.*  
*\**  
*\*DATA: IT\_TAB TYPE STANDARD TABLE OF LS\_TAB,*  
*\*      WA\_TAB TYPE LS\_TAB.*  
*\**  
*\*SELECT-OPTIONS S\_VBELN FOR WA\_TAB-VBELN.*  
*\**  
*\*START-OF-SELECTION.*  
*\*  SELECT A\_S~VBELN*  
*\*    A\_S~AUDAT*  
*\*    A\_S~VBTYP*  
*\*    A\_S~TRVOG*  
*\*    A\_S~AUART*  
*\**  
*\*    A\_I~POSNR*  
*\*    A\_I~MATNR*  
*\*    A\_I~MATWA*  
*\**  
*\*    INTO TABLE IT\_TAB*  
*\*    FROM VBAK AS A\_S*  
*\*    left outer JOIN VBAP AS A\_I ON A\_S~VBELN = A\_I~VBELN*  
*\*    WHERE A\_S~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 IT\_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.*

REPORT Z392\_P\_ASS.  
  
data : lv\_vbeln type vbeln,  
      it\_tab type Z392\_ST\_TT,  
      wa\_tab type Z392\_ST\_FT.  
  
select-options s\_vbeln for lv\_vbeln.  
  
  
CALL FUNCTION 'Z392\_FM\_ASS'  
 EXPORTING  
   I\_A           = s\_vbeln-low  
   I\_B           = s\_vbeln-high  
  TABLES  
    gt\_tab        = it\_tab  
          .  
  
perform disp\_vbap\_data.  
  
*\*loop at it\_tab into gt\_tab.*  
*\*  write :/ wa\_tab-*  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_vbap\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_vbap\_data .  
  
  loop at it\_tab into wa\_tab.  
    write : / wa\_tab-vbeln.  
  
  ENDLOOP.  
  
ENDFORM.