**CLASSICAL REPORT:**

REPORT Z391\_CLASSICAL\_REPORT NO STANDARD PAGE HEADING LINE-COUNT 20(3).  
  
LOAD-OF-PROGRAM.  
INCLUDE Z\_CLASSICAL\_REPORT\_TOP.  
  
SELECT-OPTIONS S\_MATNR FOR WA\_TAB-MATNR OBLIGATORY.  
  
INITIALIZATION.  
CLEAR :WA\_TAB, LT\_TAB.  
S\_MATNR-LOW = '1'.  
S\_MATNR-HIGH = '100'.  
APPEND S\_MATNR.  
  
"VALIDATION SQL CODE FOR INPUT­­­­­­­­­  
AT SELECTION-SCREEN ON S\_MATNR.  
  SELECT SINGLE MATNR INTO LV\_MATNR  
    FROM MARA  
    WHERE MATNR IN S\_MATNR.  
  
    IF SY-SUBRC EQ 0.  
      MESSAGE S000(ZSUHEL\_MESSAGES).  
      ELSE.  
        MESSAGE E001(ZSUHEL\_MESSAGES).  
    ENDIF.  
  
START-OF-SELECTION.  
PERFORM GET\_MAT\_DATA.  
END-OF-SELECTION.  
  
PERFORM DISPLAY\_MAT\_DATA.  
  
\*\*\*\*\* LOGIC FOR HEADER  
 data lv\_count type i.  
  lv\_count = sy-linct - sy-linno.  
  skip lv\_count.  
  
top-of-page.  
  uline.  
  write :/ 'Material No',  
         30 'Mat Ind',  
         40 'Mat Type',  
         55 'Mat UOM',  
         70 'Gross Weight',  
         90 'Net weight'.  
  uline.  
  
end-of-page.  
  write :/ 'current list page no:' , sy-pagno,  
          'Date: ', sy-datum,  
          'Time:', sy-uzeit.  
  
  
\*\*\*\*\*  LOGIC TO GET DATA FROM INTERNAK TABLE  
FORM GET\_MAT\_DATA.  
  SELECT MATNR MBRSH MTART MEINS BRGEW NTGEW  
    INTO TABLE LT\_TAB FROM MARA WHERE MATNR IN S\_MATNR.  
  
    IF SY-SUBRC EQ 0.  
      MESSAGE S000(ZSUHEL\_MESSAGES).  
      ELSE.  
        MESSAGE E001(ZSUHEL\_MESSAGES).  
    ENDIF.  
    ENDFORM.  
  
\*\*\*\*\*LOGIC TO GET DATA FROM INTERNAL TABLE TO WORK AREA.  
FORM DISPLAY\_MAT\_DATA.  
  LOOP AT LT\_TAB INTO WA\_TAB.  
    WRITE:/ WA\_TAB-MATNR,  
           WA\_TAB-MBRSH,  
            WA\_TAB-MTART,  
            WA\_TAB-MEINS,  
            WA\_TAB-BRGEW,  
            WA\_TAB-NTGEW.  
    ENDLOOP.  
    ENDFORM.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**INTERACTIVE REPORT:**

1. INTERACTIVE AT LINE:

INCLUDE PROGRAM:

TYPES : BEGIN OF LS\_TAB,        "STRUCTURE FOR BASIC LIST  
        MATNR TYPE MATNR,  
        MBRSH TYPE MBRSH,  
        MTART TYPE MTART,  
        MEINS TYPE MEINS,  
        BRGEW TYPE BRGEW,  
        NTGEW TYPE NTGEW,  
  END OF LS\_TAB,  
  
        begin of ls\_tab2,  " strs  marc             "SYTRUCTURE FOR SECONDARY LIST  
          matnr type matnr, " Material Number  
          werks type werks\_d, " plant  
          pstat type pstat\_d, " Maintenance status  
          lvorm type lvowk,  " flag for mat  
  
        end of ls\_tab2,  
  
        begin of ls\_tab3,  " strs  makt  
          matnr type matnr, " Material Number         "STRUCTURE FOR SECONDARY LIST 2  
          spras type  spras,  
          maktx type  maktx,  
          maktg type  maktg,  
  
        end of ls\_tab3.  
  
DATA : LT\_TAB TYPE STANDARD TABLE OF LS\_TAB,  
        WA\_TAB TYPE LS\_TAB,  
  
      LT\_TAB2 TYPE STANDARD TABLE OF LS\_TAB2,  
      WA\_TAB2 TYPE LS\_TAB2,  
  
      LT\_TAB3 TYPE STANDARD TABLE OF LS\_TAB3,  
      WA\_TAB3 TYPE LS\_TAB3.  
  
  
DATA :LV\_MATNR  TYPE MATNR,  
      LV\_COUNT TYPE I.

REPORT PROGRAM:

REPORT Z391\_INTERACTIVEATLINE\_FINAL NO STANDARD PAGE HEADING  
LINE-COUNT 20(3) LINE-SIZE 500  
MESSAGE-ID ZSUHEL\_MESSAGES.  
  
LOAD-OF-PROGRAM.  
  
INCLUDE Z391\_INTERACTIVEATLINE\_FINAL1.  
  
SELECT-OPTIONS S\_MATNR FOR WA\_TAB-MATNR OBLIGATORY.  
  
INITIALIZATION.  
  
CLEAR : WA\_TAB , LT\_TAB.  
S\_MATNR-LOW = '1'.  
S\_MATNR-HIGH = '100'.  
APPEND S\_MATNR.  
  
AT SELECTION-SCREEN ON S\_MATNR.  
  SELECT SINGLE MATNR  
    INTO LV\_MATNR FROM MARA  
    WHERE MATNR IN S\_MATNR.  
  
  IF SY-SUBRC EQ 0.  
    MESSAGE S000.  
    ELSE.  
      MESSAGE E001.  
   ENDIF.  
  
START-OF-SELECTION.  
PERFORM GET\_MAT\_DATA.  
END-OF-SELECTION.  
  
PERFORM DISPLAY\_MAT\_DATA.  
  
lv\_count = sy-linct - sy-linno.  
  skip lv\_count.  
  
top-of-page.  
  uline.  
  write :/ 'Records are found in marc table', 'the curent list index is :  ', sy-lsind.  
  write :/ 'Material No',  
         30 'Mat Ind',  
         40 'Mat Type',  
         55 'Mat UOM',  
         70 'Gross Weight',  
         90 'Net weight'.  
  uline.  
  
end-of-page.  
  write :/ 'current list page no:' , sy-pagno,  
          'Date: ', sy-datum,  
          'Time:', sy-uzeit.  
  
  
AT LINE-SELECTION.  
  CASE SY-LSIND.  
    WHEN 1.  
      SELECT MATNR WERKS PSTAT LVORM  
        INTO TABLE LT\_TAB2 FROM MARC  
        WHERE MATNR EQ WA\_TAB-MATNR.  
  
        loop at lt\_tab2 into wa\_tab2.  
        write : /3 wa\_tab2-matnr hotspot,  
                 30 wa\_tab2-werks,  
                 40 wa\_tab2-pstat,  
                 55 wa\_tab2-lvorm.  
        hide   wa\_tab2-matnr.  
        ENDLOOP.  
        WHEN 2.  
      select matnr spras maktx maktg  
             into table lt\_Tab3  
            from makt  
            where matnr eq wa\_tab2-matnr.   " hide stmt  
  
      loop at lt\_tab3 into wa\_tab3.  
        write : /3 wa\_tab3-matnr,  
                 30 wa\_tab3-spras,  
                 40 wa\_tab3-maktx,  
                 55 wa\_tab3-maktg.  
      endloop.  
  
      ENDCASE.  
  
top-of-page during line-selection.  
  
  case sy-lsind.  
    when 1.  
      write :/ 'Records are found in marc -plant  table', 'the curent list index is :  ', sy-lsind.  
      uline.  
      write :/3 'Material No',  
             30 'Mat plant',  
             40 'Mat stat',  
             55 'Mat flag'.  
      uline.  
  
    when 2.  
      write : / 'Records are found in makt- text table', 'the curent list index is :  ', sy-lsind.  
      uline.  
      write :/3 'Material No',  
             30 'Mat language',  
             40 'Mat description',  
             55 'Mat grp'.  
      uline.  
  endcase.  
  
  
  
\*\*\* FORM FOR GETTING THE DATA  
FORM GET\_MAT\_DATA.  
  SELECT MATNR MBRSH MTART MEINS BRGEW NTGEW  
    INTO TABLE LT\_TAB FROM MARA  
    WHERE MATNR IN S\_MATNR.  
   IF SY-SUBRC EQ 0.  
     message s000.  
  else.  
    message e001.  
  endif.  
  ENDFORM.  
  
  
  
\*\*\*  FORM TO DISPLAY THE DATA  
FORM DISPLAY\_MAT\_DATA.  
  LOOP AT LT\_TAB INTO WA\_TAB.  
    WRITE :/ WA\_TAB-MATNR HOTSPOT,  
             30,WA\_TAB-MBRSH,  
             40,WA\_TAB-MTART,  
             55,WA\_TAB-MEINS,  
             70,WA\_TAB-BRGEW,  
             90,WA\_TAB-NTGEW.  
    ENDLOOP.  
    ENDFORM.

**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**

2.INTERACTIVE AT USER

INCLUDE PROGRAM FOR AT USER

types : begin of ls\_tab,  " strs   mara  
          matnr type matnr, " Material Number  
          mbrsh type mbrsh, " Industry  
          mtart type mtart, " Material Number  
          meins type meins, " Base Unit of Measure  
          brgew type brgew, " Gross Weight  
          ntgew type ntgew, " Net Weight  
  
        end of ls\_tab,  
  
        begin of ls\_tab2,  " strs  marc  
          matnr type matnr, " Material Number  
          werks type werks\_d, " plant  
          pstat type pstat\_d, " Maintenance status  
          lvorm type lvowk,  " flag for mat  
  
        end of ls\_tab2,  
  
        begin of ls\_tab3,  " strs  makt  
          matnr type matnr, " Material Number  
          spras type  spras,  
          maktx type  maktx,  
          maktg type  maktg,  
  
        end of ls\_tab3.  
  
  
data : lt\_tab type standard table of ls\_tab,  " itab  
       wa\_tab type ls\_tab,   " work area  
  
      lt\_tab2 type standard table of ls\_tab2,  " itab  
       wa\_tab2 type ls\_tab2,   " work area  
  
  
      lt\_tab3 type standard table of ls\_tab3,  " itab  
       wa\_tab3 type ls\_tab3,   " work area  
  
       lv\_matnr type matnr,  
       lv\_count TYPE i.

REPORT PROGRAM FOR AT USER:

REPORT Z391\_INTERACTIVEATUSER no standard page heading line-count 20(3)  
line-size 500 message-id ZSUHEL\_MESSAGES.  
  
LOAD-OF-PROGRAM.  
INCLUDE Z391\_INTERACTIVEATUSER\_INCLUDE.  
  
  select-options s\_matnr for wa\_tab-matnr obligatory.  
  
initialization.  
  
  clear: wa\_Tab, lt\_Tab.  
  
  s\_matnr-low  = '1'.  
  s\_matnr-high  = '100'.  
  append s\_matnr.  
  
  at selection-screen on s\_matnr.   " 1 to 100  
  
  select MARA~matnr  
    into lv\_matnr  
     from mara  
     where matnr in s\_matnr.  
  
  if sy-subrc eq 0.  
    message s000.  " se91 - tcode.  
  else.  
    message e001.  " se91 - tcode.  
  endif.  
  
ENDSELECT.  
  
start-of-selection.  
SET PF-STATUS 'ZSUHEL\_PFS' EXCLUDING 'INFO'.  
  perform get\_mat\_data.  
  
end-of-selection.  
  perform disp\_mat\_data.  
  
  lv\_count = sy-linct - sy-linno.  
  skip lv\_count.  
  
top-of-page.  
  uline.  
  write :/ 'Records are found in marc table', 'the curent list index is :  ', sy-lsind.  
  write :/ 'Material No',  
         30 'Mat Ind',  
         40 'Mat Type',  
         55 'Mat UOM',  
         70 'Gross Weight',  
         90 'Net weight'.  
  uline.  
  
end-of-page.  
  write :/ 'current list page no:' , sy-pagno,  
          'Date: ', sy-datum,  
          'Time:', sy-uzeit.  
  
AT USER-COMMAND.  
  CASE :SY-UCOMM.  
    WHEN 'DISPLAY'.  
      WRITE 'UST-HYDERABAD'.  
    WHEN 'INFO'.  
      WRITE 'UST TRIVENDRAM'.  
    WHEN 'BACK'.  
      WRITE 'UST AMERICA'.  
    WHEN 'TCODE'.  
      WRITE 'UST CALIFORNIA'.  
    WHEN 'SAP'.  
      WRITE 'UST SAP'.  
    WHEN 'ABAP'.  
      WRITE 'UST ABAP'.  
   ENDCASE.  
  
  
form get\_mat\_data .  
  
  select matnr mbrsh mtart meins brgew ntgew  
      into table lt\_Tab  
      from mara  
      where matnr in  s\_matnr.  
  if sy-subrc eq 0.  
    write :/ 'Records are found in mara table'.  
  else.  
    write : / 'Records are not found in mara table'.  
  endif.  
  
endform.  
  
  
form disp\_mat\_data .  " basic list disp  
  loop at lt\_tab into wa\_tab.  
    write : / wa\_tab-matnr  HOTSPOT,  
             30 wa\_tab-mbrsh,  
             40 wa\_tab-mtart,  
             55 wa\_tab-meins,  
             70 wa\_tab-brgew,  
             90 wa\_tab-ntgew.  
  
    hide wa\_tab-matnr.  " for next where  
  
  endloop.  
endform.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

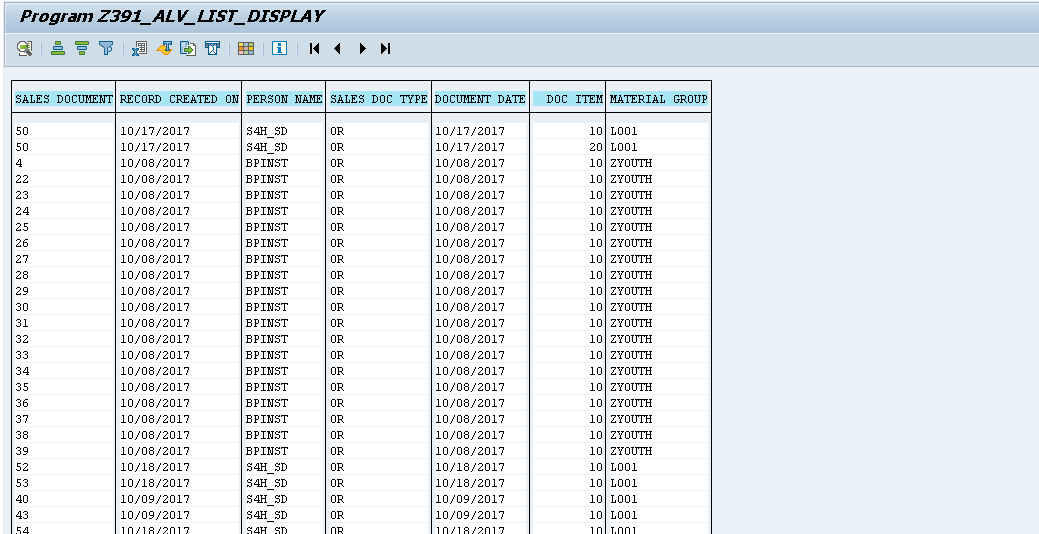
Description automatically generated

**ALV REPORTS**

*\*&---------------------------------------------------------------------\**  
*\*& Include          Z391\_ALV\_LIST\_DISPLAY\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
types : begin of ls\_tab,  *" strs   mara*  
          VBELN type VBELN\_VA, *" Material Number*  
          ERDAT type ERDAT, *" Industry*  
          ERNAM type ERNAM, *" Material Number*  
          AUART type AUART, *" Base Unit of Measure*  
          AUDAT type AUDAT, *" Gross Weight*  
          POSNR TYPE POSNR,  
          MATKL TYPE MATKL,  
  
        end of ls\_tab.  
  
    data : lt\_tab type standard table of ls\_tab,  *" itab*  
*\*\*\*\*\*\*           wa\_tab type ls\_tab,   " work area*  
  
           lv\_VBELN type VBELN,  
           WA\_FIELDCAT TYPE SLIS\_FIELDCAT\_ALV,  *" ALV FCAT FOR WA*  
           LT\_FIELDCAT TYPE SLIS\_T\_FIELDCAT\_ALV,  
           TY\_LAYOUT TYPE SLIS\_LAYOUT\_ALV.

REPORT PROGRAM

REPORT Z391\_ALV\_LIST\_DISPLAY no standard page heading line-count 20(3)  
line-size 500 message-id ZSUHEL\_MESSAGES.  
  
INCLUDE Z391\_ALV\_LIST\_DISPLAY\_TOP.  
  
SELECT-OPTIONS s\_VBELN for LV\_VBELN OBLIGATORY.  
  
INITIALIZATION.  
*"CODE FOR INPUT VALIDATION*  
  
  CLEAR: LT\_TAB, WA\_FIELDCAT, LT\_FIELDCAT.  
  s\_VBELN-low  = '1'.  
  s\_VBELN-high  = '100'.  
  append s\_VBELN.  
  
at selection-screen on s\_VBELN.   *" 1 to 100*  
  
  select single VBELN  
    into lv\_VBELN  
     from VBAK  
     where VBELN in s\_VBELN.  
  
  if sy-subrc eq 0.  
    message s000.  *" se91 - tcode.*  
  else.  
    message e001.  *" se91 - tcode.*  
  endif.  
  
start-of-selection.  
  
  perform get\_alv\_fcat.  
  
  perform get\_SALES\_data.  
  
  END-OF-SELECTION.  
  PERFORM DISPLAY\_SALES\_DATA.  
  
*\*  TO GET SALES DATA*  
  form get\_SALES\_data.  
  
  select VBAK~VBELN VBAK~ERDAT VBAK~ERNAM AUART AUDAT POSNR MATKL  
      into table lt\_Tab  
      from VBAK  
       INNER JOIN VBAP ON VBAP~VBELN = VBAK~VBELN  
      where VBAK~VBELN in  s\_VBELN.  
  
  if sy-subrc eq 0.  
    message s000.  
  else.  
    message e001.  
  endif.  
  
endform.  
  
*\*FORM TO GET DATA*  
  
form get\_alv\_fcat .  
  
*\*\*\*\*\*  VBELN*  
  wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'SALES DOCUMENT'.  
  wa\_fieldcat-fieldname = 'VBELN'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
*\*\*\*\*\*  ERDAT*  
  wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'RECORD CREATED ON'.  
  wa\_fieldcat-fieldname = 'ERDAT'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
*\*\*\*\*\*  ERNAM*  
  wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'PERSON NAME'.  
  wa\_fieldcat-fieldname = 'ERNAM'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  
*\*\*\*\*\*  AUART*  
  wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'SALES DOC TYPE'.  
  wa\_fieldcat-fieldname = 'AUART'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  
  
*\*\*\*\*\*  AUDAT*  
  wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = 'DOCUMENT DATE'.  
  wa\_fieldcat-fieldname = 'AUDAT'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
*\*\*\*\*  POSNR*  
  wa\_fieldcat-col\_pos = '6'.  
  wa\_fieldcat-seltext\_l = 'DOC ITEM'.  
  wa\_fieldcat-fieldname = 'POSNR'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
*\*  \*\*\*\*\*  MATKL*  
  wa\_fieldcat-col\_pos = '7'.  
  wa\_fieldcat-seltext\_l = 'MATERIAL GROUP'.  
  wa\_fieldcat-fieldname = 'MATKL'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  ENDFORM.  
  
*\*FORM TO DISPLAY DATA*  
FORM DISPLAY\_SALES\_DATA.  
CALL FUNCTION 'REUSE\_ALV\_LIST\_DISPLAY'  
 EXPORTING  
   I\_CALLBACK\_PROGRAM             = 'SY-REPID'  
  
   IT\_FIELDCAT                    = LT\_FIELDCAT  
  
  TABLES  
    t\_outtab                       = LT\_TAB  
 EXCEPTIONS  
   PROGRAM\_ERROR                  = 1  
   OTHERS                         = 2  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
  
ENDFORM.



GRID ALV

*\*&---------------------------------------------------------------------\**  
*\*& Include          Z391\_ALV\_LIST\_DISPLAY\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
types : begin of ls\_tab,  *" strs   mara*  
          VBELN type VBELN\_VA, *" Material Number*  
          ERDAT type ERDAT, *" Industry*  
          ERNAM type ERNAM, *" Material Number*  
          AUART type AUART, *" Base Unit of Measure*  
          AUDAT type AUDAT, *" Gross Weight*  
          POSNR TYPE POSNR,  
          MATKL TYPE MATKL,  
  
        end of ls\_tab.  
  
    data : lt\_tab type standard table of ls\_tab,  *" itab*  
*\*\*\*\*\*\*           wa\_tab type ls\_tab,   " work area*  
  
           lv\_VBELN type VBELN,  
           WA\_FIELDCAT TYPE SLIS\_FIELDCAT\_ALV,  *" ALV FCAT FOR WA*  
           LT\_FIELDCAT TYPE SLIS\_T\_FIELDCAT\_ALV,  
           TY\_LAYOUT TYPE SLIS\_LAYOUT\_ALV.

*\*&---------------------------------------------------------------------\**  
*\*& Report Z391\_ALV\_GRID\_DISPLAY*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT Z391\_ALV\_GRID\_DISPLAY no standard page heading line-count 20(3)  
line-size 500 message-id ZSUHEL\_MESSAGES.  
  
INCLUDE Z391\_ALV\_LIST\_DISPLAY\_TOP.  
  
SELECT-OPTIONS s\_VBELN for LV\_VBELN OBLIGATORY.  
  
INITIALIZATION.  
*"CODE FOR INPUT VALIDATION*  
  
  CLEAR: LT\_TAB, WA\_FIELDCAT, LT\_FIELDCAT.  
  s\_VBELN-low  = '1'.  
  s\_VBELN-high  = '100'.  
  append s\_VBELN.  
  
at selection-screen on s\_VBELN.   *" 1 to 100*  
  
  select single VBELN  
    into lv\_VBELN  
     from VBAK  
     where VBELN in s\_VBELN.  
  
  if sy-subrc eq 0.  
    message s000.  *" se91 - tcode.*  
  else.  
    message e001.  *" se91 - tcode.*  
  endif.  
  
start-of-selection.  
  
  perform get\_alv\_fcat.  
  
  perform get\_SALES\_data.  
  
  END-OF-SELECTION.  
  PERFORM DISPLAY\_SALES\_DATA.  
  
*\*  TO GET SALES DATA*  
  form get\_SALES\_data.  
  
  select VBELN ERDAT ERNAM AUART AUDAT  
      into table lt\_Tab  
      from VBAK  
      where VBELN in  s\_VBELN.  
  
  if sy-subrc eq 0.  
    message s000.  
  else.  
    message e001.  
  endif.  
  
endform.  
  
*\*FORM TO GET DATA*  
  
form get\_alv\_fcat .  
  
*\*\*\*\*\*  VBELN*  
  wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'SALES DOCUMENT'.  
  wa\_fieldcat-fieldname = 'VBELN'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
*\*\*\*\*\*  ERDAT*  
  wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'RECORD CREATED ON'.  
  wa\_fieldcat-fieldname = 'ERDAT'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
*\*\*\*\*\*  ERNAM*  
  wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'PERSON NAME'.  
  wa\_fieldcat-fieldname = 'ERNAM'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  
*\*\*\*\*\*  AUART*  
  wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'SALES DOC TYPE'.  
  wa\_fieldcat-fieldname = 'AUART'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  
  
*\*\*\*\*\*  AUDAT*  
  wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = 'DOCUMENT DATE'.  
  wa\_fieldcat-fieldname = 'AUDAT'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  ENDFORM.  
  
*\*FORM TO DISPLAY DATA*  
FORM DISPLAY\_SALES\_DATA.  
CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
 EXPORTING  
   I\_CALLBACK\_PROGRAM             = 'SY-REPID'  
  
   IT\_FIELDCAT                    = LT\_FIELDCAT  
  
  TABLES  
    t\_outtab                       = LT\_TAB  
 EXCEPTIONS  
   PROGRAM\_ERROR                  = 1  
   OTHERS                         = 2  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
  
ENDFORM.

A screenshot of a computer

Description automatically generated

GRID WITH LOGO

*\*&---------------------------------------------------------------------\**  
*\*& Include          Z391\_ALV\_LIST\_DISPLAY\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
types : begin of ls\_tab,  *" strs   mara*  
          VBELN type VBELN\_VA, *" Material Number*  
          ERDAT type ERDAT, *" Industry*  
          ERNAM type ERNAM, *" Material Number*  
          AUART type AUART, *" Base Unit of Measure*  
          AUDAT type AUDAT, *" Gross Weight*  
          POSNR TYPE POSNR,  
          MATKL TYPE MATKL,  
  
        end of ls\_tab.  
  
    data : lt\_tab type standard table of ls\_tab,  *" itab*  
*\*\*\*\*\*\*           wa\_tab type ls\_tab,   " work area*  
  
           lv\_VBELN type VBELN,  
           WA\_FIELDCAT TYPE SLIS\_FIELDCAT\_ALV,  *" ALV FCAT FOR WA*  
           LT\_FIELDCAT TYPE SLIS\_T\_FIELDCAT\_ALV,  
           TY\_LAYOUT TYPE SLIS\_LAYOUT\_ALV.

*\*&---------------------------------------------------------------------\**  
*\*& Report Z391\_ALV\_GRID\_DISPLAY*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT Z391\_ALV\_GRID\_DISPLAY no standard page heading line-count 20(3)  
line-size 500 message-id ZSUHEL\_MESSAGES.  
  
INCLUDE Z391\_ALV\_LIST\_DISPLAY\_TOP.  
  
SELECT-OPTIONS s\_VBELN for LV\_VBELN OBLIGATORY.  
  
INITIALIZATION.  
*"CODE FOR INPUT VALIDATION*  
  
  CLEAR: LT\_TAB, WA\_FIELDCAT, LT\_FIELDCAT.  
  s\_VBELN-low  = '1'.  
  s\_VBELN-high  = '100'.  
  append s\_VBELN.  
  
at selection-screen on s\_VBELN.   *" 1 to 100*  
  
  select single VBELN  
    into lv\_VBELN  
     from VBAK  
     where VBELN in s\_VBELN.  
  
  if sy-subrc eq 0.  
    message s000.  *" se91 - tcode.*  
  else.  
    message e001.  *" se91 - tcode.*  
  endif.  
  
start-of-selection.  
  
  perform get\_alv\_fcat.  
  
  perform get\_SALES\_data.  
  
  END-OF-SELECTION.  
  PERFORM DISPLAY\_SALES\_DATA.  
  
*\*  TO GET SALES DATA*  
  form get\_SALES\_data.  
  
  select VBELN ERDAT ERNAM AUART AUDAT  
      into table lt\_Tab  
      from VBAK  
      where VBELN in  s\_VBELN.  
  
  if sy-subrc eq 0.  
    message s000.  
  else.  
    message e001.  
  endif.  
  
endform.  
  
*\*FORM TO GET DATA*  
  
form get\_alv\_fcat .  
  
*\*\*\*\*\*  VBELN*  
  wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'SALES DOCUMENT'.  
  wa\_fieldcat-fieldname = 'VBELN'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  wa\_fieldcat-no\_zero = 'X'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
*\*\*\*\*\*  ERDAT*  
  wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'RECORD CREATED ON'.  
  wa\_fieldcat-fieldname = 'ERDAT'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
*\*\*\*\*\*  ERNAM*  
  wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'PERSON NAME'.  
  wa\_fieldcat-fieldname = 'ERNAM'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  
*\*\*\*\*\*  AUART*  
  wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'SALES DOC TYPE'.  
  wa\_fieldcat-fieldname = 'AUART'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  
  
*\*\*\*\*\*  AUDAT*  
  wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = 'DOCUMENT DATE'.  
  wa\_fieldcat-fieldname = 'AUDAT'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  ENDFORM.  
  
*\*&---------------------------------------------------------------------\**  
*\*& Form DISPLAY\_SALES\_DATA*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM display\_sales\_data .  
  DATA LV\_TITLE TYPE LVC\_TITLE VALUE 'SALES DOC DATA'.  
  TY\_LAYOUT-NO\_COLHEAD = 'X'.  
  TY\_LAYOUT-NO\_VLINE = 'X'.  
  TY\_LAYOUT-NO\_HLINE = 'X'.  
CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
 EXPORTING  
*\*   I\_INTERFACE\_CHECK                 = ' '*  
*\*   I\_BYPASSING\_BUFFER                = ' '*  
*\*   I\_BUFFER\_ACTIVE                   = ' '*  
    I\_CALLBACK\_PROGRAM                = SY-REPID  
*\*   I\_CALLBACK\_PF\_STATUS\_SET          = ' '*  
*\*   I\_CALLBACK\_USER\_COMMAND           = ' '*  
   I\_CALLBACK\_TOP\_OF\_PAGE            = 'UST\_LOGO'  
*\*   I\_CALLBACK\_HTML\_TOP\_OF\_PAGE       = ' '*  
*\*   I\_CALLBACK\_HTML\_END\_OF\_LIST       = ' '*  
*\*   I\_STRUCTURE\_NAME                  =*  
*\*   I\_BACKGROUND\_ID                   = ' '*  
  I\_GRID\_TITLE                      = LV\_TITLE  
*\*   I\_GRID\_SETTINGS                   =*  
*\*   IS\_LAYOUT                         =*  
   IT\_FIELDCAT                       = LT\_FIELDCAT  
*\*   IT\_EXCLUDING                      =*  
*\*   IT\_SPECIAL\_GROUPS                 =*  
*\*   IT\_SORT                           =*  
*\*   IT\_FILTER                         =*  
*\*   IS\_SEL\_HIDE                       =*  
*\*   I\_DEFAULT                         = 'X'*  
*\*   I\_SAVE                            = ' '*  
*\*   IS\_VARIANT                        =*  
*\*   IT\_EVENTS                         =*  
*\*   IT\_EVENT\_EXIT                     =*  
*\*   IS\_PRINT                          =*  
*\*   IS\_REPREP\_ID                      =*  
*\*   I\_SCREEN\_START\_COLUMN             = 0*  
*\*   I\_SCREEN\_START\_LINE               = 0*  
*\*   I\_SCREEN\_END\_COLUMN               = 0*  
*\*   I\_SCREEN\_END\_LINE                 = 0*  
*\*   I\_HTML\_HEIGHT\_TOP                 = 0*  
*\*   I\_HTML\_HEIGHT\_END                 = 0*  
*\*   IT\_ALV\_GRAPHICS                   =*  
*\*   IT\_HYPERLINK                      =*  
*\*   IT\_ADD\_FIELDCAT                   =*  
*\*   IT\_EXCEPT\_QINFO                   =*  
*\*   IR\_SALV\_FULLSCREEN\_ADAPTER        =*  
*\*   O\_PREVIOUS\_SRAL\_HANDLER           =*  
*\*   O\_COMMON\_HUB                      =*  
*\* IMPORTING*  
*\*   E\_EXIT\_CAUSED\_BY\_CALLER           =*  
*\*   ES\_EXIT\_CAUSED\_BY\_USER            =*  
  TABLES  
    t\_outtab                          = LT\_TAB  
 EXCEPTIONS  
   PROGRAM\_ERROR                     = 1  
   OTHERS                            = 2.  
  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
  
ENDFORM.  
  
FORM UST\_LOGO.  
  DATA: LT\_HEADER TYPE SLIS\_T\_LISTHEADER,  
        WA\_HEADER TYPE SLIS\_LISTHEADER.  
*\*  WA\_HEADER-TYP = 'S'.*  
  WA\_HEADER-TYP = 'A'.  
*\*  WA\_HEADER-TYP = 'H'*  
  
  WA\_HEADER-INFO = 'UST GLOBAL'.  
  WA\_HEADER-KEY = 'HYDERABAD'.  
  APPEND WA\_HEADER TO LT\_HEADER.  
  
  WA\_HEADER-TYP = 'S'.  
  WA\_HEADER-INFO = SY-DATUM.  
  WA\_HEADER-KEY = SY-UZEIT.  
  APPEND WA\_HEADER TO LT\_HEADER.  
  
  CALL FUNCTION 'REUSE\_ALV\_COMMENTARY\_WRITE'  
    EXPORTING  
      it\_list\_commentary       = LT\_HEADER  
     I\_LOGO                   = 'ZUST\_LOGO'  
*\*     I\_END\_OF\_LIST\_GRID       =*  
*\*     I\_ALV\_FORM               =*  
            .  
endform.

A screenshot of a computer

Description automatically generated

HIERSEQUENTIAL REPORT:

*\*&---------------------------------------------------------------------\**  
*\*& Report Z391\_ALV\_HEIRSEQ\_ALV*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT Z391\_ALV\_HEIRSEQ\_ALV NO STANDARD PAGE HEADING LINE-COUNT 20(3).  
  
TYPES : BEGIN OF ST\_VBAK,  
        UST(1),  
        VBELN TYPE VBELN\_VA,    *"SALES DOCUMENT*  
        ERDAT TYPE ERDAT,       *"RECORD CREATED ON*  
        ERZET TYPE ERZET,        *"ENTRY TIME*  
      END OF ST\_VBAK.  
  
DATA : IT\_VBAK TYPE  TABLE OF ST\_VBAK,     *" CREATE INTERNAL TABLE*  
      WA\_VBAK TYPE ST\_VBAK.      *" CREATE WORK AREA*  
  
TYPES : BEGIN OF ST\_VBAP,  
        VBELN TYPE VBELN\_VA,        *"SALES DOCUMENT*  
        POSNR TYPE POSNR\_VA,        *"SALES DOCUMNET ITEM*  
        MATNR TYPE MATNR,           *"MATERIAL NUMBER*  
  END OF ST\_VBAP.  
  
DATA : IT\_VBAP TYPE  TABLE OF ST\_VBAP,      *"CREATE INTERNAL TABLE*  
       WA\_VBAP TYPE ST\_VBAP,      *"CREATE WORK AREA FOR TAB 2.*  
  
F\_CAT TYPE SLIS\_T\_FIELDCAT\_ALV,  
WA\_FCAT LIKE LINE OF F\_CAT,  
  
IT\_KEY TYPE SLIS\_KEYINFO\_ALV,  
  
LAYOUT TYPE SLIS\_LAYOUT\_ALV.  
  
TABLES VBAK.  
  
SELECT-OPTIONS S\_VBELN FOR VBAK-VBELN.  
  
START-OF-SELECTION.  
PERFORM GET\_DATA.  
  
PERFORM FCAT\_DATA.  
PERFORM GET\_HEIRCHDATA.  
*\*&---------------------------------------------------------------------\**  
*\*& Form GET\_DATA*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_data.  
  
  SELECT VBELN ERDAT ERZET  
    FROM VBAK  
    INTO CORRESPONDING FIELDS OF TABLE  
    IT\_VBAK WHERE VBELN  
    IN S\_VBELN.  
  
    SELECT VBELN POSNR MATNR  
      FROM VBAP  
      INTO TABLE IT\_VBAP  
      WHERE VBELN IN S\_VBELN.  
  
ENDFORM.  
  
FORM fcat\_data .  
  
*\*\*\*\*  BUILD FCAT FOR VBAK*  
  
  WA\_FCAT-col\_pos = '1'.  
  WA\_FCAT-fieldname = 'VBELN'.  
  WA\_FCAT-tabname = 'IT\_VBAK'.  
  WA\_FCAT-ref\_tabname = 'VBAK'.  
  WA\_FCAT-ref\_fieldname = 'VBELN'.  
  
  APPEND WA\_FCAT TO F\_CAT.  
  CLEAR WA\_FCAT.  
  
  WA\_FCAT-col\_pos = '2'.  
  WA\_FCAT-fieldname = 'ERDAT'.  
  WA\_FCAT-tabname = 'IT\_VBAK'.  
  WA\_FCAT-ref\_tabname = 'VBAK'.  
  WA\_FCAT-ref\_fieldname = 'ERDAT'.  
  
  APPEND WA\_FCAT TO F\_CAT.  
  CLEAR WA\_FCAT.  
  
  WA\_FCAT-col\_pos = '3'.  
  WA\_FCAT-fieldname = 'ERZET'.  
  WA\_FCAT-tabname = 'IT\_VBAK'.  
  WA\_FCAT-ref\_tabname = 'VBAK'.  
  WA\_FCAT-ref\_fieldname = 'ERZET'.  
  
  APPEND WA\_FCAT TO F\_CAT.  
  CLEAR WA\_FCAT.  
  
*\*\*\*\*\*  BUILD FCAT FOR VBAP*  
  
  WA\_FCAT-col\_pos = '1'.  
  WA\_FCAT-fieldname = 'VBELN'.  
  WA\_FCAT-tabname = 'IT\_VBAP'.  
  WA\_FCAT-ref\_tabname = 'VBAP'.  
  WA\_FCAT-ref\_fieldname = 'VBELN'.  
  
  APPEND WA\_FCAT TO F\_CAT.  
  CLEAR WA\_FCAT.  
  
  WA\_FCAT-col\_pos = '2'.  
  WA\_FCAT-fieldname = 'POSNR'.  
  WA\_FCAT-tabname = 'IT\_VBAP'.  
  WA\_FCAT-ref\_tabname = 'VBAP'.  
  WA\_FCAT-ref\_fieldname = 'POSNR'.  
  
  APPEND WA\_FCAT TO F\_CAT.  
  CLEAR WA\_FCAT.  
  
  WA\_FCAT-col\_pos = '3'.  
  WA\_FCAT-fieldname = 'MATNR'.  
  WA\_FCAT-tabname = 'IT\_VBAP'.  
  WA\_FCAT-ref\_tabname = 'VBAP'.  
  WA\_FCAT-ref\_fieldname = 'MATNR'.  
  
  APPEND WA\_FCAT TO F\_CAT.  
  CLEAR WA\_FCAT.  
  
ENDFORM.  
  
  
FORM get\_heirchdata .  
  IF SY-SUBRC EQ 0.  
  IT\_KEY-HEADER01 = 'VBELN'.  
  LAYOUT-EXPAND\_FIELDNAME = 'UST'.  
  CALL FUNCTION 'REUSE\_ALV\_HIERSEQ\_LIST\_DISPLAY'  
  EXPORTING  
   I\_CALLBACK\_PROGRAM             = SY-REPID  
   IS\_LAYOUT                      = LAYOUT  
   IT\_FIELDCAT                    = F\_CAT[]  
    i\_tabname\_header               = 'IT\_VBAK'  
    i\_tabname\_item               = 'IT\_VBAP'  
    is\_keyinfo                     = IT\_KEY  
*\*   IS\_PRINT                       =*  
*\*   IS\_REPREP\_ID                   =*  
*\*   IS\_HIERSEQ\_SETTINGS            =*  
*\*   I\_BYPASSING\_BUFFER             =*  
*\*   I\_BUFFER\_ACTIVE                =*  
*\*   IR\_SALV\_HIERSEQ\_ADAPTER        =*  
*\*   IT\_EXCEPT\_QINFO                =*  
*\*   I\_SUPPRESS\_EMPTY\_DATA          = ABAP\_FALSE*  
*\*   I\_LIST\_DISPLAY\_ONLY            = ABAP\_FALSE*  
*\*   O\_COMMON\_HUB                   =*  
*\* IMPORTING*  
*\*   E\_EXIT\_CAUSED\_BY\_CALLER        =*  
*\*   ES\_EXIT\_CAUSED\_BY\_USER         =*  
  TABLES  
    t\_outtab\_header                = IT\_VBAK  
    t\_outtab\_item                  = IT\_VBAP  
 EXCEPTIONS  
   PROGRAM\_ERROR                  = 1  
   OTHERS                         = 2.  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
ELSE.  
  WRITE 'NO RECORDS FOUND'.  
  ENDIF.  
ENDFORM.

A screenshot of a computer

Description automatically generated

SUBROUTINES PROGRAMS

*\*&---------------------------------------------------------------------\**  
*\*& Report ZSUHEL\_SUBROUTINES*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZSUHEL\_SUBROUTINES.  
  
PARAMETERS : LV\_A TYPE I,  
            LV\_B TYPE I.  
  
  
DATA: LV\_SUM TYPE I,  
      LV\_MUL TYPE I,  
      LV\_SUB TYPE I,  
      LV\_DIV TYPE I.  
CALL FUNCTION 'ZSUHEL\_FUNCT\_MODULE'  
 EXPORTING  
   I\_A           = LV\_A  
   I\_B           = LV\_B  
 IMPORTING  
   E\_SUM         = LV\_SUM  
   E\_MUL         = LV\_MUL  
   E\_SUB         = LV\_SUB  
   E\_DIV         = LV\_DIV  
          .  
WRITE /: LV\_SUM , LV\_MUL , LV\_DIV , LV\_SUB.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

*\*&---------------------------------------------------------------------\**  
*\*& Report ZSUHEL\_SUBROUTINE\_PARA*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zsuhel\_subroutine\_para.  
  
*\*\* 1.CALL BY VALUE.*  
*\*DATA a TYPE i VALUE 20.*  
*\*PERFORM sub USING a.*  
*\*WRITE a.*  
*\*FORM sub USING VALUE(p\_a).*  
*\*  p\_a = 30.*  
*\*ENDFORM.*  
  
*\*2. CALL BY REFERENCE*  
*\*DATA a TYPE i VALUE 20.*  
*\*PERFORM sub changing a.*  
*\*WRITE a.*  
*\*FORM sub changing p\_a.*  
*\*  p\_a = 30.*  
*\*ENDFORM.*  
  
*\*3 call by value and return*  
  
*\*DATA a TYPE i VALUE 20.*  
*\*PERFORM sub changing a.*  
*\*WRITE a.*  
*\**  
*\*FORM sub  CHANGING value(p\_a).*  
*\*  p\_a = 50.*  
*\*ENDFORM.*  
  
*\*USING INCLUDES*  
  
*\*DATA a TYPE i VALUE 20.*  
*\*PERFORM sub changing a.*  
*\*WRITE a.*  
*\*INCLUDE ZSUHEL\_SUBROUTINE\_SF01.*

SELECTION SCREEN REPORT

*\*&---------------------------------------------------------------------\**  
*\*& Report ZSUHEL\_SELSCREEN\_REPORT*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZSUHEL\_SELSCREEN\_REPORT NO STANDARD PAGE HEADING  
MESSAGE-ID ZSUHEL\_MESSAGES  
LINE-COUNT 25(3) LINE-SIZE 500.  
  
  
LOAD-OF-PROGRAM.  
INCLUDE ZSUHEL\_SELSCREEN\_TOP\_INCLUDE.  
  
*\*BLOCK 1*  
  
SELECTION-SCREEN :BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-001 .  
SELECT-OPTIONS S\_MATNR FOR WA\_TAB-MATNR OBLIGATORY NO-EXTENSION.  
SELECTION-SCREEN END OF BLOCK B1.  
  
*\*BLOCK2*  
  
SELECTION-SCREEN : BEGIN OF BLOCK B2 WITH FRAME TITLE TEXT-002.  
  PARAMETERS : P\_MTART TYPE MTART OBLIGATORY.  
SELECTION-SCREEN END OF BLOCK B2.  
*\*BLOCK 3*  
  
SELECTION-SCREEN : BEGIN OF BLOCK B3 WITH FRAME TITLE TEXT-003.  
  PARAMETERS : R1 RADIOBUTTON GROUP RAD1,  
                R2 RADIOBUTTON GROUP RAD1 DEFAULT 'X',  
                R3 RADIOBUTTON GROUP RAD1.  
  SELECTION-SCREEN ULINE /1(79).  
  PARAMETERS :    R4 RADIOBUTTON GROUP RAD2,  
                R5 RADIOBUTTON GROUP RAD2.  
  SELECTION-SCREEN END OF BLOCK B3.  
  
  SELECTION-SCREEN: BEGIN OF BLOCK B4 WITH FRAME TITLE TEXT-004.  
    PARAMETERS : C1 AS CHECKBOX,  
                 C2 AS CHECKBOX.  
    SELECTION-SCREEN END OF BLOCK B4.  
  
  
  
  
INITIALIZATION.  
CLEAR :WA\_TAB, LT\_TAB.  
S\_MATNR-LOW = '1'.  
S\_MATNR-HIGH = '100'.  
APPEND S\_MATNR.  
  
*"VALIDATION SQL CODE FOR INPUT*  
AT SELECTION-SCREEN ON S\_MATNR.  
  SELECT SINGLE MATNR INTO LV\_MATNR  
    FROM MARA  
    WHERE MATNR IN S\_MATNR.  
  
    IF SY-SUBRC EQ 0.  
      MESSAGE S000(ZSUHEL\_MESSAGES).  
      ELSE.  
        MESSAGE E001(ZSUHEL\_MESSAGES).  
    ENDIF.  
  
START-OF-SELECTION.  
PERFORM GET\_MAT\_DATA.  
END-OF-SELECTION.  
  
PERFORM DISPLAY\_MAT\_DATA.  
  
*\*\*\*\*\* LOGIC FOR HEADER*  
 data lv\_count type i.  
  lv\_count = sy-linct - sy-linno.  
  skip lv\_count.  
  
top-of-page.  
  uline.  
  write :/ 'Material No',  
         30 'Mat Ind',  
         40 'Mat Type',  
         55 'Mat UOM',  
         70 'Gross Weight',  
         90 'Net weight'.  
  uline.  
  
end-of-page.  
  write :/ 'current list page no:' , sy-pagno,  
          'Date: ', sy-datum,  
          'Time:', sy-uzeit.  
  
  
*\*\*\*\*\*  LOGIC TO GET DATA FROM INTERNAK TABLE*  
FORM GET\_MAT\_DATA.  
  SELECT MATNR MBRSH MTART MEINS BRGEW NTGEW  
    INTO TABLE LT\_TAB FROM MARA WHERE MATNR IN S\_MATNR AND MTART EQ P\_MTART.  
  
    IF SY-SUBRC EQ 0.  
      MESSAGE S000(ZSUHEL\_MESSAGES).  
      ELSE.  
        MESSAGE E001(ZSUHEL\_MESSAGES).  
    ENDIF.  
    ENDFORM.  
  
*\*\*\*\*\*LOGIC TO GET DATA FROM INTERNAL TABLE TO WORK AREA.*  
FORM DISPLAY\_MAT\_DATA.  
  LOOP AT LT\_TAB INTO WA\_TAB.  
    WRITE:/ WA\_TAB-MATNR,  
           WA\_TAB-MBRSH,  
            WA\_TAB-MTART,  
            WA\_TAB-MEINS,  
            WA\_TAB-BRGEW,  
            WA\_TAB-NTGEW.  
    ENDLOOP.  
    ENDFORM.

*A screenshot of a computer

Description automatically generated*

*A screenshot of a computer

Description automatically generated*

SELECTION SCREEN OUTPUT

*\*&---------------------------------------------------------------------\**  
*\*& Report ZSUHEL\_SELSCREEN\_REPORT*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZSUHEL\_SELSCREEN\_REPORT NO STANDARD PAGE HEADING  
MESSAGE-ID ZSUHEL\_MESSAGES  
LINE-COUNT 25(3) LINE-SIZE 500.  
  
  
LOAD-OF-PROGRAM.  
INCLUDE ZSUHEL\_SELSCREEN\_TOP\_INCLUDE.  
  
*\*BLOCK 1*  
  
SELECTION-SCREEN :BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-001 .  
SELECT-OPTIONS S\_MATNR FOR WA\_TAB-MATNR OBLIGATORY NO-EXTENSION.  
SELECTION-SCREEN END OF BLOCK B1.  
  
*\*BLOCK2*  
  
SELECTION-SCREEN : BEGIN OF BLOCK B2 WITH FRAME TITLE TEXT-002.  
  PARAMETERS : P\_MTART TYPE MTART OBLIGATORY.  
SELECTION-SCREEN END OF BLOCK B2.  
*\*BLOCK 3*  
  
SELECTION-SCREEN : BEGIN OF BLOCK B3 WITH FRAME TITLE TEXT-003.  
  PARAMETERS : R1 RADIOBUTTON GROUP RAD1,  
                R2 RADIOBUTTON GROUP RAD1 DEFAULT 'X',  
                R3 RADIOBUTTON GROUP RAD1.  
  SELECTION-SCREEN ULINE /1(79).  
  PARAMETERS :    R4 RADIOBUTTON GROUP RAD2,  
                R5 RADIOBUTTON GROUP RAD2.  
  SELECTION-SCREEN END OF BLOCK B3.  
  
  SELECTION-SCREEN: BEGIN OF BLOCK B4 WITH FRAME TITLE TEXT-004.  
    PARAMETERS : C1 AS CHECKBOX,  
                 C2 AS CHECKBOX.  
    SELECTION-SCREEN END OF BLOCK B4.  
  
  
  
  
INITIALIZATION.  
CLEAR :WA\_TAB, LT\_TAB.  
S\_MATNR-LOW = '1'.  
S\_MATNR-HIGH = '100'.  
APPEND S\_MATNR.  
  
*"VALIDATION SQL CODE FOR INPUT*  
AT SELECTION-SCREEN ON S\_MATNR.  
  SELECT SINGLE MATNR INTO LV\_MATNR  
    FROM MARA  
    WHERE MATNR IN S\_MATNR.  
  
    IF SY-SUBRC EQ 0.  
      MESSAGE S000(ZSUHEL\_MESSAGES).  
      ELSE.  
        MESSAGE E001(ZSUHEL\_MESSAGES).  
    ENDIF.  
  
START-OF-SELECTION.  
PERFORM GET\_MAT\_DATA.  
END-OF-SELECTION.  
  
PERFORM DISPLAY\_MAT\_DATA.  
  
*\*\*\*\*\* LOGIC FOR HEADER*  
 data lv\_count type i.  
  lv\_count = sy-linct - sy-linno.  
  skip lv\_count.  
  
top-of-page.  
  uline.  
  write :/ 'Material No',  
         30 'Mat Ind',  
         40 'Mat Type',  
         55 'Mat UOM',  
         70 'Gross Weight',  
         90 'Net weight'.  
  uline.  
  
end-of-page.  
  write :/ 'current list page no:' , sy-pagno,  
          'Date: ', sy-datum,  
          'Time:', sy-uzeit.  
  
  
*\*\*\*\*\*  LOGIC TO GET DATA FROM INTERNAK TABLE*  
FORM GET\_MAT\_DATA.  
  SELECT MATNR MBRSH MTART MEINS BRGEW NTGEW  
    INTO TABLE LT\_TAB FROM MARA WHERE MATNR IN S\_MATNR AND MTART EQ P\_MTART.  
  
    IF SY-SUBRC EQ 0.  
      MESSAGE S000(ZSUHEL\_MESSAGES).  
      ELSE.  
        MESSAGE E001(ZSUHEL\_MESSAGES).  
    ENDIF.  
    ENDFORM.  
  
*\*\*\*\*\*LOGIC TO GET DATA FROM INTERNAL TABLE TO WORK AREA.*  
FORM DISPLAY\_MAT\_DATA.  
  LOOP AT LT\_TAB INTO WA\_TAB.  
    WRITE:/ WA\_TAB-MATNR,  
           WA\_TAB-MBRSH,  
            WA\_TAB-MTART,  
            WA\_TAB-MEINS,  
            WA\_TAB-BRGEW,  
            WA\_TAB-NTGEW.  
    ENDLOOP.  
    ENDFORM.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

ITAB PROGRAMS

*\*&---------------------------------------------------------------------\**  
*\*& Report Z391\_SUHELITAB\_PROGRAMS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT Z391\_SUHELITAB\_PROGRAMS.  
  
*\*DATA: BEGIN OF LINE,        "FIELD STRING*  
*\*      COL\_1(10) type C,*  
*\*      COL\_2(20) TYPE C,*  
*\*  END OF LINE.*  
*\**  
*\*DATA ITAB LIKE TABLE OF LINE.*  
*\**  
*\*LINE-COL\_1 = 'UST'.*  
*\*LINE-COL\_2 = 'TECHNOLOGIES'.*  
*\*APPEND LINE TO ITAB.*  
*\*CLEAR LINE.*  
*\**  
*\*IF ITAB IS INITIAL.*  
*\*  WRITE 'INTERNAL TABLE IS EMPTY'.*  
*\*ENDIF.*  
  
  
*\*\*  2 INSERT*  
  
*\*DATA : BEGIN OF LINE,     "FIELD STRING*  
*\*      NAME(20) TYPE C,*  
*\*      AGE TYPE I,*  
*\*      WEIGHT TYPE P DECIMALS 2,*  
*\*      LAND(5) TYPE C,*  
*\*  END OF LINE.*  
*\**  
*\*TYPES : BEGIN OF LINE,    "LOCAL STRUCTURE*  
*\*  NAME(20) TYPE C,*  
*\*  AGE TYPE I,*  
*\*  WEIGHT TYPE P DECIMALS 2,*  
*\*  LAND(5) TYPE C,*  
*\*  END OF LINE.*  
*\**  
*\**  
*\*\*DATA itab LIKE SORTED TABLE OF LINE  WITH NON-UNIQUE KEY name age weight land.*  
*\*DATA itab LIKE STANDARD TABLE OF LINE  WITH NON-UNIQUE KEY name age weight land.*  
*\**  
*\*LINE-NAME = 'JOHAN'.*  
*\*LINE-AGE = 35.*  
*\*LINE-WEIGHT = '80.00'.*  
*\*LINE-LAND = 'IND'.*  
*\*INSERT LINE INTO TABLE ITAB.*  
*\**  
*\*LINE-NAME = 'PETER'.*  
*\*LINE-AGE = 35.*  
*\*LINE-WEIGHT = '45.00'.*  
*\*LINE-LAND = 'USA'.*  
*\*INSERT LINE INTO TABLE ITAB.*  
*\**  
*\*LINE-NAME = 'DAVID'.*  
*\*LINE-AGE = 40.*  
*\*LINE-WEIGHT = '95.00'.*  
*\*LINE-LAND = 'DE'.*  
*\*INSERT LINE INTO TABLE ITAB.*  
*\**  
*\*SORT ITAB BY NAME.       " IF WE ARE USING SORTED TABLE WE CAN IGNORE THIS LINE*  
*\**  
*\*LOOP AT ITAB INTO LINE.*  
*\*  WRITE:/ LINE-NAME, LINE-AGE , LINE-WEIGHT , LINE-LAND.*  
*\*  ENDLOOP.*  
  
  
*\*\*---------3-----------INSERT LINE---------*  
DATA:BEGIN OF LINE,     *"FIELD STRING*  
  COL\_1 TYPE I,  
  COL\_2 TYPE I,  
  END OF LINE.  
  
DATA ITAB LIKE TABLE OF LINE.  
  
DO 2 TIMES.  
  LINE-COL\_1 = SY-INDEX.  
  LINE-COL\_2 = SY-INDEX \*\* 2.  
  APPEND LINE TO ITAB.  
  ENDDO.  
  
  LOOP AT ITAB INTO LINE.  
    LINE-COL\_1 = 3 \* SY-TABIX.  
    LINE-COL\_2 = 5 \* SY-TABIX.  
    INSERT LINE INTO ITAB.  
    ENDLOOP.  
    LOOP AT ITAB INTO LINE.  
      WRITE : / SY-TABIX, LINE-COL\_1, LINE-COL\_2.  
      ENDLOOP.  
  
  
*\*MODIFY*  
  
*\*DATA : BEGIN OF LINE,*  
*\*  COL1 TYPE I,*  
*\*  COL2 TYPE I,*  
*\*  END OF LINE.*  
*\**  
*\*  DATA ITAB LIKE TABLE OF LINE.*  
*\**  
*\*  DO 3 TIMES.*  
*\*    LINE-COL1 = SY-INDEX.*  
*\*    LINE-COL2 = SY-INDEX \*\* 2.*  
*\*    APPEND LINE TO ITAB.*  
*\*  ENDDO.*  
*\**  
*\*LOOP AT ITAB INTO LINE.*  
*\*  IF SY-TABIX = 2.*  
*\*    LINE-COL1 = SY-TABIX \* 10.*  
*\*    LINE-COL2 = ( SY-TABIX \* 10 ) \*\* 2.*  
*\*    MODIFY ITAB FROM LINE.*  
*\*  ENDIF.*  
*\*  ENDLOOP.*  
*\**  
*\*  LOOP AT ITAB INTO LINE.*  
*\*    WRITE : / SY-TABIX , LINE-COL1 , LINE-COL2.*  
*\*  ENDLOOP.*  
  
  
*\*DELETE*  
*\**  
*\*DATA: BEGIN OF LINE,*  
*\*  col1 TYPE I,*  
*\*  col2 TYPE I,*  
*\*END OF LINE.*  
*\*DATA itab LIKE HASHED TABLE OF LINE WITH UNIQUE key col1.*  
*\**  
*\*DO 4 TIMES.*  
*\*  LINE-col1 = sy-INDEX.*  
*\*  LINE-col2 = sy-INDEX \*\* 2.*  
*\*  INSERT LINE INTO TABLE itab.*  
*\*ENDDO.*  
*\*LINE-col1 = 1.*  
*\*DELETE TABLE itab: FROM LINE,*  
*\*WITH TABLE KEY col1 = 3.*  
*\**  
*\*LOOP AT itab INTO LINE.*  
*\*  WRITE: / LINE-col1, LINE-col2.*  
*\*ENDLOOP.*  
  
*\**  
*\*COLLECT*  
*\**  
*\*DATA: BEGIN OF LINE,*  
*\*  col\_1(3) TYPE C,*  
*\*  col\_2(2) TYPE n,*  
*\*  col\_3    TYPE I,*  
*\*END OF LINE.*  
*\**  
*\*DATA itab LIKE SORTED TABLE OF LINE*  
*\*      WITH NON-UNIQUE KEY col\_1  col\_2.*  
*\**  
*\*LINE-col\_1 = 'UST'.*  
*\*LINE-col\_2 = '12'.*  
*\*LINE-col\_3 = 3.*  
*\**  
*\*COLLECT LINE INTO itab.*  
*\*WRITE / sy-tabix.*  
*\**  
*\**  
*\*LINE-col\_1 = 'TECH'.*  
*\*LINE-col\_2 = '34'.*  
*\*LINE-col\_3 = 5.*  
*\*COLLECT LINE INTO itab.*  
*\*WRITE / sy-tabix.*  
*\**  
*\**  
*\*LINE-col\_1 = 'UST'.*  
*\*LINE-col\_2 = '22'.*  
*\*LINE-col\_3 = 7.*  
*\*COLLECT LINE INTO itab.*  
*\**  
*\*WRITE / sy-tabix.*  
*\*LOOP AT itab INTO LINE.*  
*\*  WRITE: / LINE-col\_1, LINE-col\_2, LINE-col\_3.*  
*\*ENDLOOP.*  
  
*\*Data Types*  
  
*\*\*--------------------------------------------------------------\**  
*\*TYPES: BEGIN OF ty\_student,*  
*\*  ID(5)    TYPE n,*  
*\*  name(10) TYPE C,*  
*\*END OF ty\_student.*  
*\**  
*\**  
*\*\*READ + INDEX + KEY*  
*\*\*--------------------------------------------------------------\**  
*\*\*Data Declaration*  
*\*\*--------------------------------------------------------------\**  
*\*DATA: gwa\_student TYPE ty\_student.*  
*\*DATA: it TYPE TABLE OF ty\_student.*  
*\**  
*\*gwa\_student-ID    = 1.*  
*\*gwa\_student-name  = 'JOHN'.*  
*\*APPEND gwa\_student TO it.*  
*\**  
*\*gwa\_student-ID    = 2.*  
*\*gwa\_student-name  = 'JIM'.*  
*\*APPEND gwa\_student TO it.*  
*\**  
*\*gwa\_student-ID    = 3.*  
*\*gwa\_student-name  = 'JACK'.*  
*\*APPEND gwa\_student TO it.*  
*\**  
*\*SORT IT BY ID.*  
*\*\*\*\* # 1*  
*\*READ TABLE it INTO gwa\_student INDEX 2.*  
*\**  
*\*IF sy-subrc = 0.*  
*\*  WRITE: gwa\_student-ID, gwa\_student-name.*  
*\*ELSE.*  
*\*  WRITE 'No Record Found'.*  
*\*ENDIF.*  
  
*\*\*\*LOOP AT*  
*\*\*\*    LOOP AT*  
*\*\*\*ENDLOOP.*  
  
  
*\*\*\*\*\*\* # 2*  
*\*READ TABLE it INTO gwa\_student WITH KEY name = 'JACK'.*  
*\*IF sy-subrc = 0.*  
*\*  WRITE: gwa\_student-ID, gwa\_student-name.*  
*\*ELSE.*  
*\*  WRITE 'No Record Found'.*  
*\*ENDIF.*  
  
  
*\*\*\*\*FIELD-SYMBOLS  &C*  
  
*\*\*\*\*\*\*\*\*\* # 1*  
*\*DATA : F1(20) TYPE C.*  
*\**  
*\*FIELD-SYMBOLS <FS\_UST>.*  
*\**  
*\*F1 = 'UST TECH'.*  
*\**  
*\*ASSIGN F1 TO <FS\_UST>.*  
*\**  
*\*WRITE : <FS\_UST>.*  
  
*\*----------------------------------*  
*\* # 2*  
*\*----------------------------------*  
*\*data: begin of line,  " field string*  
*\*        col\_1 type i,*  
*\*        col\_2 type i,*  
*\*      end of line.*  
*\**  
*\*data itab like sorted table of line with unique key col\_1.*  
*\**  
*\*field-symbols <fs> like line of itab.*  
*\**  
*\**  
*\*do 10 times.*  
*\*  line-col\_1   =   sy-index.*  
*\*  line-col\_2   =   2   \*   sy-index.*  
*\*  append line to itab.*  
*\*enddo.*  
*\**  
*\*LOOP AT itab INTO LINE  .  " itab to fld*  
*\*  WRITE: / LINE-col\_1, LINE-col\_2.  " LINE-weight, LINE-land.*  
*\*ENDLOOP.*  
*\**  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*READ TABLE<<<<<<<<<*  
*\**  
*\*read table itab assigning <fs> index 5.*  
*\*\*CLEAR <fs>.*  
*\*write:   sy-subrc, sy-tabix.*  
*\*write: / <fs>-col\_1, <fs>-col\_2.*

LOCK OBJECTS:

*\*&---------------------------------------------------------------------\**  
*\*& Report ZSUHEL\_EZSUHEL*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZSUHEL\_EZSUHEL.  
PARAMETERS P\_MATNR TYPE MATNR.  
  
CALL FUNCTION 'ENQUEUE\_EZSUHEL\_SUHEL'  
EXPORTING  
 MODE\_MARA            = 'E'  
MANDT                = SY-MANDT  
MATNR                =  P\_MATNR  
*\*   X\_MATNR              = ' '*  
*\*   \_SCOPE               = '2'*  
*\*   \_WAIT                = ' '*  
*\*   \_COLLECT             = ' '*  
*\* EXCEPTIONS*  
*\*   FOREIGN\_LOCK         = 1*  
*\*   SYSTEM\_FAILURE       = 2*  
*\*   OTHERS               = 3*  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
  
BREAK-POINT.  
  
CALL FUNCTION 'DEQUEUE\_EZSUHEL\_SUHEL'  
 EXPORTING  
  MODE\_MARA       = 'E'  
  MANDT           = SY-MANDT  
 MATNR           = P\_MATNR  
*\*   X\_MATNR         = ' '*  
*\*   \_SCOPE          = '3'*  
*\*   \_SYNCHRON       = ' '*  
*\*   \_COLLECT        = ' '*  
          .

**MM01 PROGRAM**

*\*&---------------------------------------------------------------------\**  
*\*& Report Z391\_MM01\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT Z391\_MM01\_PROG.  
INCLUDE Z391\_MM01\_INPROG.  
  
  
  SELECT-OPTIONS S\_MATNR FOR WA\_TAB-MATNR.  
  
START-OF-SELECTION.  
PERFORM Z\_391\_SUBROUTINE.  
END-OF-SELECTION.  
  
PERFORM Z\_391\_LOOP.  
SELECT SINGLE MATNR  
  INTO WA\_TAB-MATNR FROM MARA WHERE MARA~MATNR IN S\_MATNR.  
  IF SY-SUBRC EQ 0.  
    MESSAGE S000(ZSUHEL\_MESSAGES).  
    ELSE.  
      MESSAGE E001(ZSUHEL\_MESSAGES).  
    ENDIF.  
  
TOP-OF-PAGE.  
WRITE :/ 'NUMBER',20  
  'INDUSTRY',30  
  'TYPE',40  
  'UNIT',50  
  'GROUP',60  
  'DESCRIPTION'.  
  
FORM z\_391\_subroutine .  
  SELECT MARA~MATNR MARA~MBRSH MARA~MTART MARA~MEINS MARA~MATKL MAKT~MAKTX  
    INTO TABLE LT\_TAB FROM MARA INNER JOIN MAKT ON MARA~MATNR = MAKT~MATNR  
    WHERE MARA~MATNR IN S\_MATNR.  
ENDFORM.  
  
  
FORM z\_391\_loop .  
LOOP AT LT\_TAB INTO WA\_TAB.  
  WRITE:/ WA\_TAB-MATNR,   *"MATERIAL NUMBER*  
        WA\_TAB-MBRSH,      *"INDUSTRY*  
        WA\_TAB-MTART,       *"MATERIAL TYPE*  
        WA\_TAB-MEINS,       *"UNIT OF MEASURE*  
        WA\_TAB-MATKL,       *"MATERIAL GROUP*  
        WA\_TAB-MAKTX.       *"MATERIAL DESCRIPTION*  
  ENDLOOP.  
ENDFORM.

**A screenshot of a computer

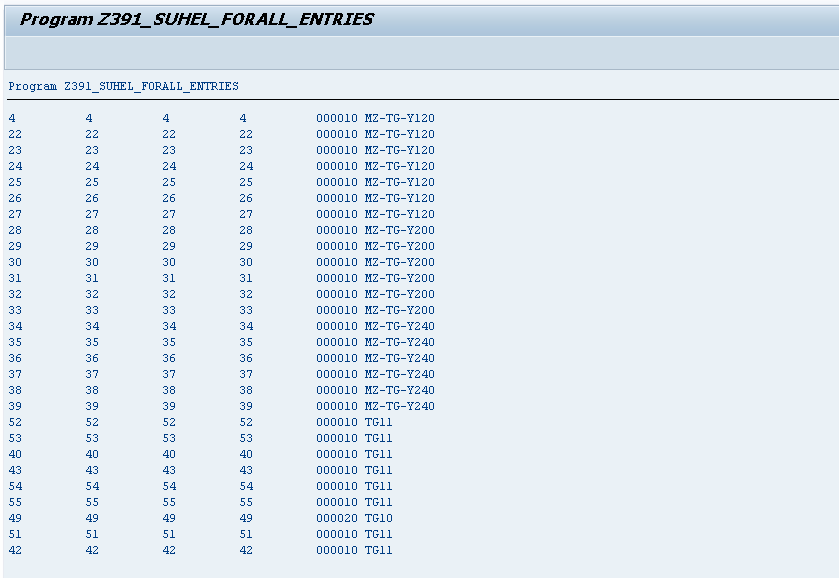
Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**FOR ALL ENTRIES**

*\*&---------------------------------------------------------------------\**  
*\*& Report Z391\_SUHEL\_FORALL\_ENTRIES*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT Z391\_SUHEL\_FORALL\_ENTRIES.  
  
INCLUDE Z391\_SUHEL\_FORALL\_ENTRIES\_INC.  
  
TABLES VBAK.  
  
SELECT-OPTIONS S\_VBELN FOR VBAK-VBELN.  
  
START-OF-SELECTION.  
SELECT VBELN ERDAT ERZET VBTYP  
  INTO TABLE IT\_TAB1 FROM VBAK  
  WHERE VBELN IN S\_VBELN.  
  
IF NOT IT\_TAB1[] IS INITIAL.          *"STEP 1*  
  SELECT VBELN POSNR MATNR  
    INTO TABLE IT\_TAB2 FROM VBAP  
  
    FOR ALL ENTRIES IN IT\_TAB1        *"STEP 2*  
    WHERE VBELN EQ IT\_TAB1-VBELN.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*loop at it\_tab1 into wa\_tab1.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*loop at it\_tab2 INTO wa\_tab2.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WRITE : wa\_tab1-matnr....*  
*\*\*\*\*\*\*\*\*\*\*\*\*\* ENDLOOP.*  
*\*\*\*\*\*\*\*\*\*\*\*\*ENDLOOP       " WE ARE AVOIDING THIS AS THIS DOESNOT VALIDATE THE ENTERED DATA IT WILL GIVE DUMP ALL THE DATA AS A DUMP*  
  
DELETE ADJACENT DUPLICATES FROM IT\_TAB1 COMPARING VBELN. *"STEP 3*  
  
SORT IT\_TAB2 BY VBELN.        *"STEP 4.*  
  
LOOP AT IT\_TAB1 INTO WA\_TAB1.  
  
READ TABLE IT\_TAB2 INTO WA\_TAB2 WITH KEY VBELN = WA\_TAB1-VBELN BINARY SEARCH.  
  IF SY-SUBRC = 0.  
    MOVE : WA\_TAB1-VBELN TO WA\_FINAL-VBELN,  
            WA\_TAB1-ERDAT TO WA\_FINAL-ERDAT,  
            WA\_TAB1-ERZET TO WA\_FINAL-ERZET,  
            WA\_TAB1-VBTYP TO WA\_FINAL-VBTYP,  
  
           WA\_TAB2-POSNR TO WA\_FINAL-POSNR,  
           WA\_TAB2-MATNR TO WA\_FINAL-MATNR.  
    APPEND WA\_FINAL TO IT\_FINAL.  
  
    ELSE.  
      CONTINUE.  
      ENDIF.  
      ENDLOOP.  
  
  
      LOOP AT IT\_FINAL INTO WA\_FINAL.  
        WRITE :/ WA\_FINAL-VBELN,  
                 WA\_FINAL-VBELN,  
                 WA\_FINAL-VBELN,  
                 WA\_FINAL-VBELN,  
  
                 WA\_FINAL-POSNR,  
                 WA\_FINAL-MATNR.  
       ENDLOOP.  
  
 ELSE.  
   WRITE 'SALES DOCS ARE NOT FOUND'.  
   ENDIF.

****

**GET CURSOR**

*\*&---------------------------------------------------------------------\**  
*\*& Report Z391\_SUHEL\_GETCURSOR*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT Z391\_SUHEL\_GETCURSOR.  
  
DATA LV\_VBELN TYPE VBELN\_VA.  
  
TYPES : BEGIN OF TY\_VBAK,  
        VBELN TYPE VBELN\_VA,  
        KUNNR TYPE KUNNR,  
        ERDAT TYPE ERDAT,  
  END OF TY\_VBAK.  
  
  DATA : IT\_VBAK TYPE TABLE OF TY\_VBAK,  
        WA\_VBAK TYPE TY\_VBAK.  
  SELECT-OPTIONS S\_VBELN FOR LV\_VBELN.  
  
  START-OF-SELECTION.  
  SELECT VBELN KUNNR ERDAT INTO TABLE IT\_VBAK  
    FROM VBAK WHERE VBELN IN S\_VBELN.  
  
    END-OF-SELECTION.  
  
    LOOP AT IT\_VBAK INTO WA\_VBAK.  
  
      FORMAT HOTSPOT ON.  
        WRITE :/ WA\_VBAK-VBELN,  
                  WA\_VBAK-KUNNR,  
*\*                  WA\_VBAK-ERDAT.*  
        FORMAT HOTSPOT OFF.  
        WRITE :/ WA\_VBAK-ERDAT.  
        ENDLOOP.  
  
TOP-OF-PAGE.  
   WRITE :/ 'SALES DOC ON',  
            'CUSTOMER NO',  
            'DATE'.  
   AT LINE-SELECTION.  
     DATA : FNAM(15),  
           FVAL(15).  
     GET CURSOR FIELD FNAM VALUE FVAL.  
     IF FNAM = 'WA\_VBAK-VBELN'.  
       SET PARAMETER ID 'AUN' FIELD FVAL.  
       CALL TRANSACTION 'VA03' AND SKIP FIRST SCREEN.  
     ELSEIF FNAM = 'WA\_VBAK-KUNNR'.  
       SET PARAMETER ID 'KUN' FIELD FVAL.  
       CALL TRANSACTION 'XD02' AND SKIP FIRST SCREEN.  
     ENDIF.

**A screenshot of a computer

Description automatically generated**

**GLOBAL CLASS ATTRIBUTES**

*\*&---------------------------------------------------------------------\**  
*\*& Report Z391\_SUHEL\_GLOBALCLASS\_ATTR*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT Z391\_SUHEL\_GLOBALCLASS\_ATTR.  
  
*\*WE CREATED THE GLOBAL CLASS AND DECLARED THE ATTRIBUTES THEN CREATED THE OBJECT GLOBALLY AND*  
*\*DATA I\_OBJ TYPE REF TO ZCL\_391\_GLOBAL\_CLASS.*  
*\*CREATE OBJECT I\_OBJ.*  
*\*START-OF-SELECTION.*  
*\*WRITE :/ I\_OBJ->i\_suhel1,*  
*\*         ZCL\_391\_GLOBAL\_CLASS=>s\_suhel2.*  
  
*\*\*\*\* LOCAL CLASS DECLARATION AND CALLING*  
  
*\*CLASS CL\_UST DEFINITION.*  
*\*  PUBLIC SECTION.*  
*\*  DATA I\_V1 TYPE I VALUE 50.*  
*\*  CLASS-DATA S\_V2 TYPE I VALUE 600.*  
*\*ENDCLASS.*  
*\**  
*\*\*\*\*\*\*\*\*\*\*\*CLASS CL\_UST IMPLEMENTATION." WE ARE NOT CALLING ANY FUNCTION BECAUSE WE DIDN'T ADD ANY METHOD.*  
*\*\*\*\*\*\*\*\*\*\*\**  
*\*\*\*\*\*\*\*\*\*\*\*  ENDCLASS.*  
*\**  
*\*DATA I\_OBJ TYPE REF TO CL\_UST.*  
*\*CREATE OBJECT I\_OBJ.*  
*\*START-OF-SELECTION.*  
*\*WRITE :/ I\_OBJ->i\_v1,*  
*\*        CL\_UST=>s\_v2.*

*EXAMPLE 2*

*\*&---------------------------------------------------------------------\**  
*\*& Report Z391\_SUHEL\_GLOBALCLASS\_METHOD*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT Z391\_SUHEL\_GLOBALCLASS\_METHOD.  
  
DATA I\_OBJ TYPE REF TO ZCL\_391\_GLOBAL\_CLASS.  
CREATE OBJECT I\_OBJ.  
  
START-OF-SELECTION.  
CALL METHOD I\_OBJ->I\_V1.  
        WRITE:/ 'INSTANCE ONE'.  
  
          SKIP 2.  
  
CALL METHOD ZCL\_391\_GLOBAL\_CLASS=>I\_V2.  
        WRITE:/ 'SPECIFIC TWO'.

**A screenshot of a computer

Description automatically generated**

*\*&---------------------------------------------------------------------\**  
*\*& Report ZCL\_391\_GL\_ASSIGNMENT1\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZCL\_391\_GL\_ASSIGNMENT1\_PROG.  
  
DATA IT\_TAB TYPE Z391\_TT\_FOR\_GL.  
DATA: I\_OBJ TYPE REF TO  
      ZCL\_391\_GL\_ASSIGNMENT,  
       LV\_VBELN TYPE VBELN.  
CREATE OBJECT I\_OBJ.  
SELECT-OPTIONS S\_VBELN FOR LV\_VBELN.  
  
*\**  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CALL METHOD i\_obj->get\_salesdata*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  EXPORTING*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    i\_vbeln1 = '5000'*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    i\_vbeln2 = '5010'*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  IMPORTING*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    gt\_tab   = it\_tab*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    .*  
*\**  
START-OF-SELECTION.  
CALL METHOD I\_OBJ->GET\_SALESDATA  
EXPORTING  
  I\_VBELN1 = S\_VBELN-LOW  
  I\_VBELN2 = S\_VBELN-HIGH  
  IMPORTING  
    GT\_TAB = IT\_TAB.  
CL\_DEMO\_OUTPUT=>DISPLAY\_DATA( IT\_TAB ).

*A screenshot of a computer

Description automatically generated*