**Modularization**

**1.Subroutines:**

REPORT ZNAGESH\_SUBROUTINES.

**1.Call By value:**  
  
DATA A TYPE I VALUE 20.  
  
PERFORM SUB USING A.  
  
WRITE a.  
  
FORM sub  USING VALUE(p\_a).  
  
p\_a  = 50.  
  
ENDFORM.

OUTPUT : 20

**2.call by reference**  
  
DATA A TYPE I VALUE 20.  
  
PERFORM SUB1 CHANGING A.  
  
WRITE a.  
FORM sub1  CHANGING p\_a.  
  
p\_a  = 50.  
  
ENDFORM.

OUTPUT : 50  
  
  
**3.Call By Value and Return**

DATA A TYPE I VALUE 20.  
  
PERFORM SUB1 CHANGING A.  
  
WRITE a.

FORM sub1  CHANGING VALUE(p\_a).  
  
p\_a  = 50.  
  
ENDFORM.

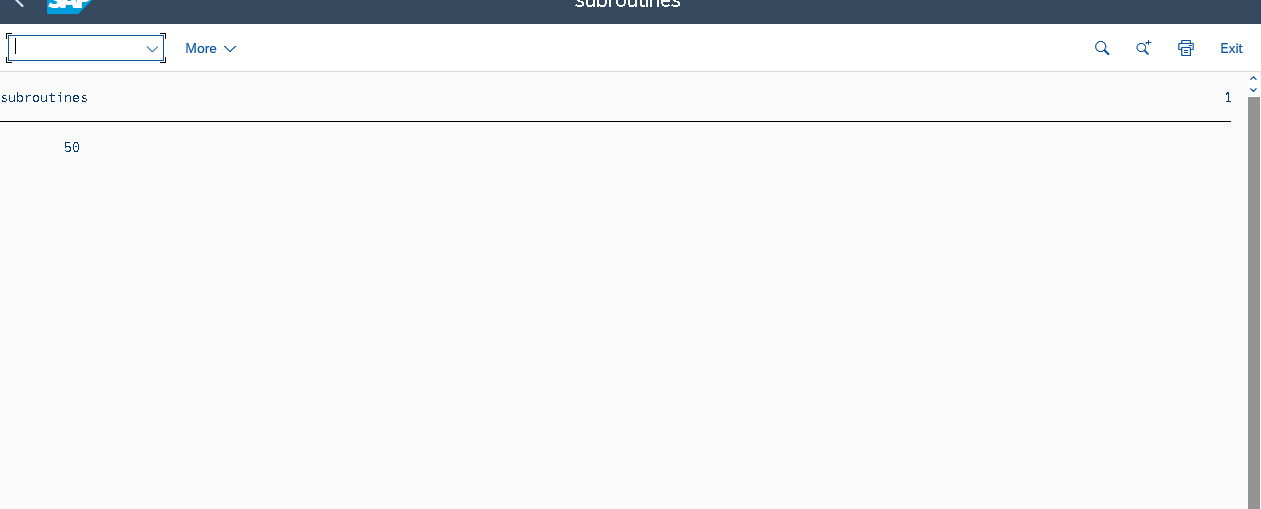
OUTPUT : 50

**4.WE can include globally**  
  
DATA A TYPE I VALUE 20.  
  
PERFORM SUB CHANGING A.  
  
WRITE a.  
  
INCLUDE ZNAGESH\_UST\_SUBROUTINES\_1.

**Include Program:**

\*& Include          ZNAGESH\_UST\_SUBROUTINES\_1  
  
FORM sub  CHANGING VALUE(p\_a).  
p\_a  = 50.  
ENDFORM.

OUTPUT:

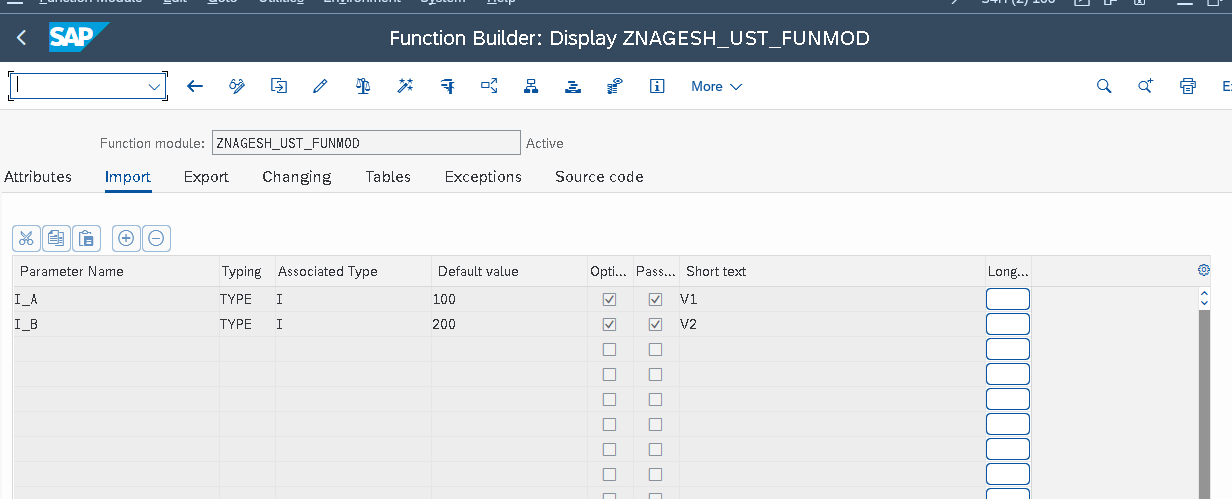


**2.Function Modules**

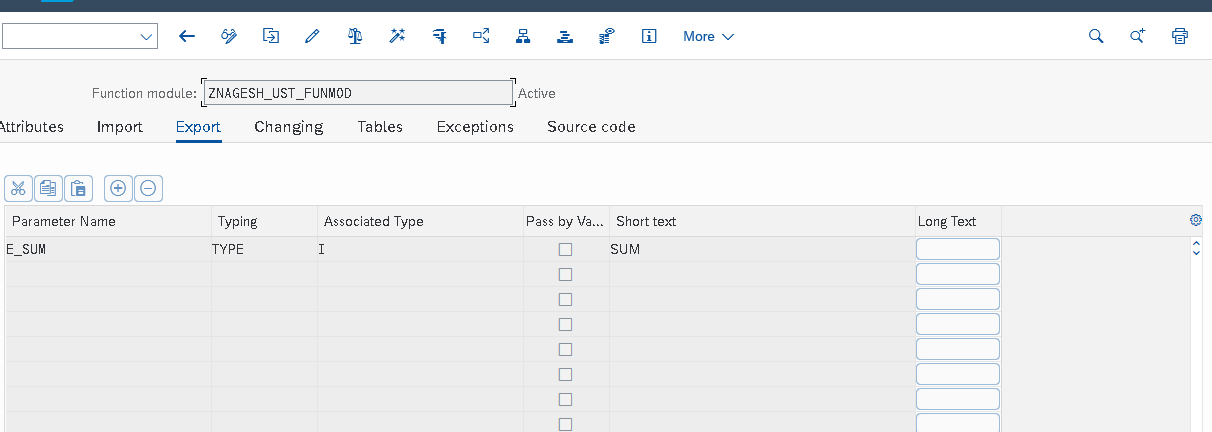
**1.Basic Program**

REPORT ZNAGESH\_USING\_MODULE.  
PARAMETERS : P\_a type I,  
p\_b type I.  
data lvsum type I.  
CALL FUNCTION 'ZNAGESH\_UST\_FUNMOD'  
 EXPORTING  
   I\_A           = p\_a  
   I\_B           = P\_b  
 IMPORTING  
   E\_SUM         = lvsum  
          .  
write lvsum.

IMPORT:



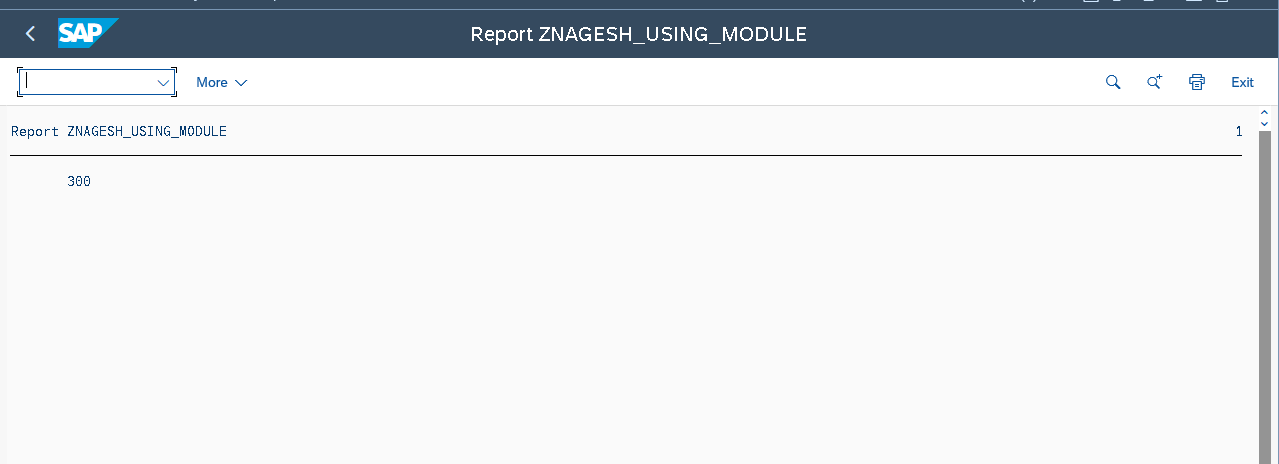
EXPORT:



SOURCE CODE:

FUNCTION ZNAGESH\_UST\_FUNMOD.  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  IMPORTING  
\*"     VALUE(I\_A) TYPE  I DEFAULT 100  
\*"     VALUE(I\_B) TYPE  I DEFAULT 200  
\*"  EXPORTING  
\*"     REFERENCE(E\_SUM) TYPE  I  
\*"----------------------------------------------------------------------  
E\_SUM = I\_A + I\_B.  
  
ENDFUNCTION.

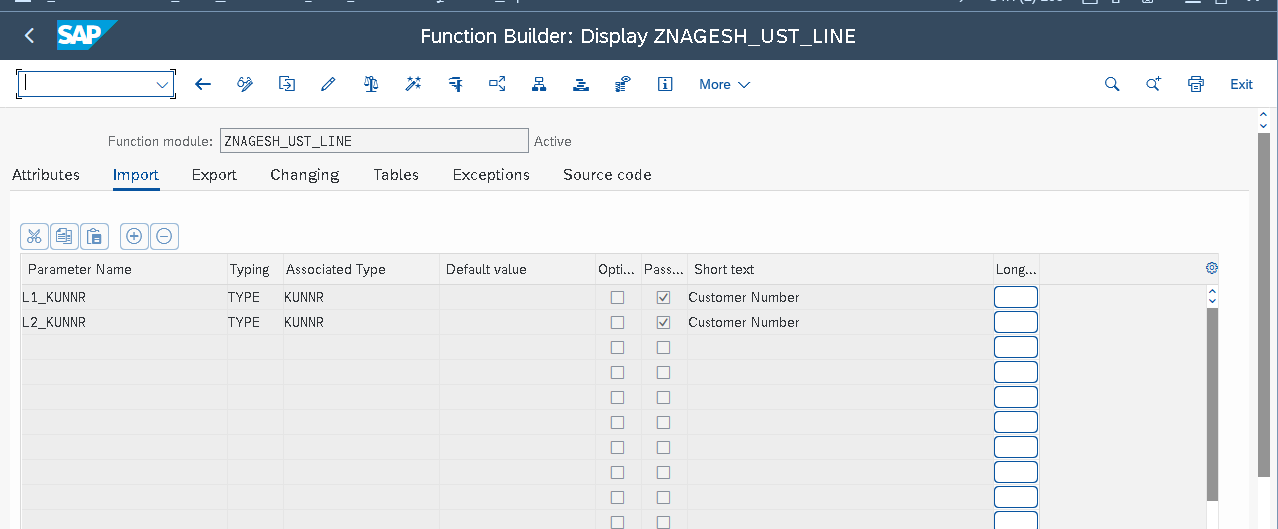
OUTPUT:



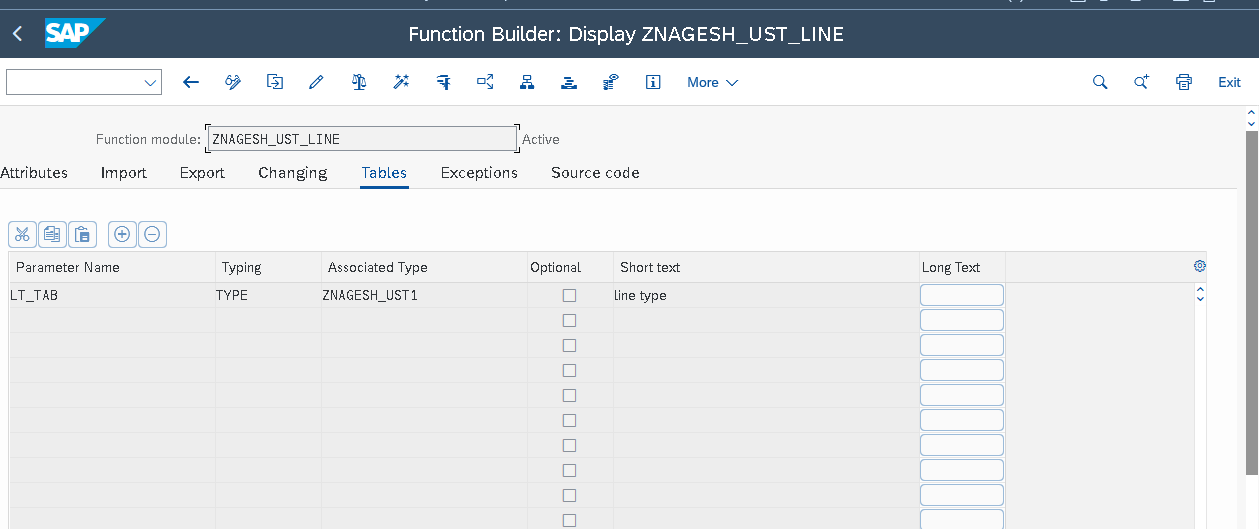
**2.Program**

REPORT ZNAGESH\_LINEPRO.  
 Data: ls\_tb type ZNAGESH\_UST1,  
       wa\_tb type ZNAGESH\_UST.  
  
 select-options s\_kunnr for wa\_tb-kunnr.  
 CALL FUNCTION 'ZNAGESH\_UST\_LINE'  
   EXPORTING  
     l1\_kunnr       = s\_kunnr-low  
     l2\_kunnr       = s\_kunnr-high  
   TABLES  
     lt\_tab         = ls\_tb.  
 loop at ls\_tb into wa\_tb.  
   write:/ wa\_tb-KUNNR,  
wa\_tb-LAND1,  
wa\_tb-NAME1,  
wa\_tb-NAME2,  
wa\_tb-ORT01.  
ENDloop.

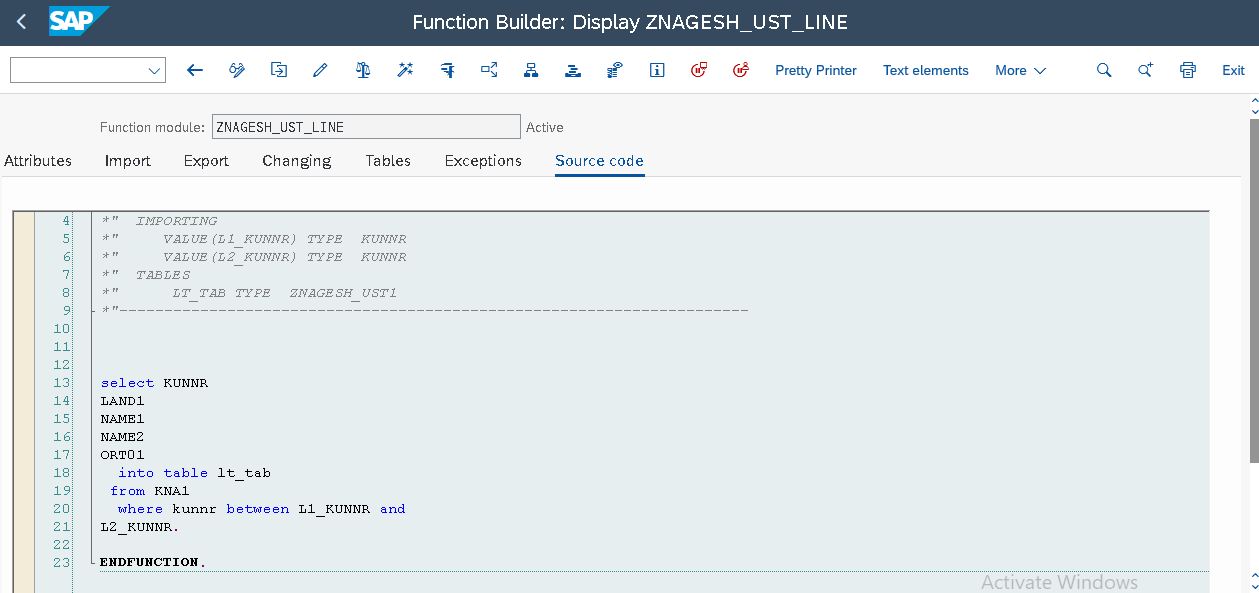
IMPORT :



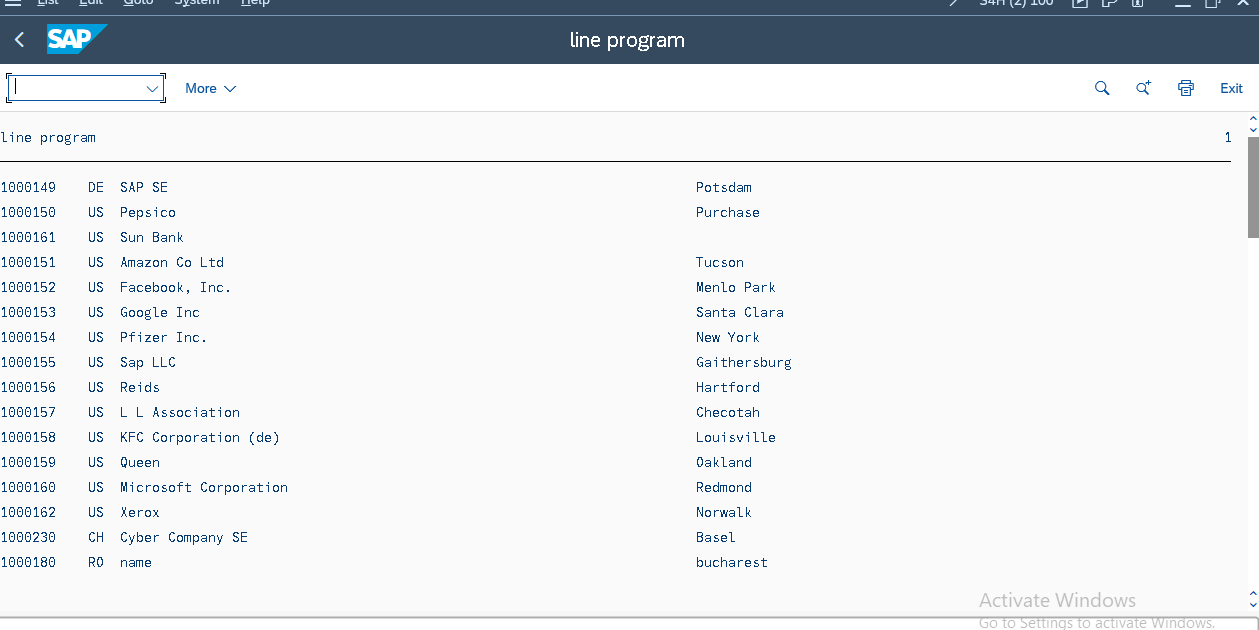
TABLES:



SOURCE CODE:



OUTPUT:

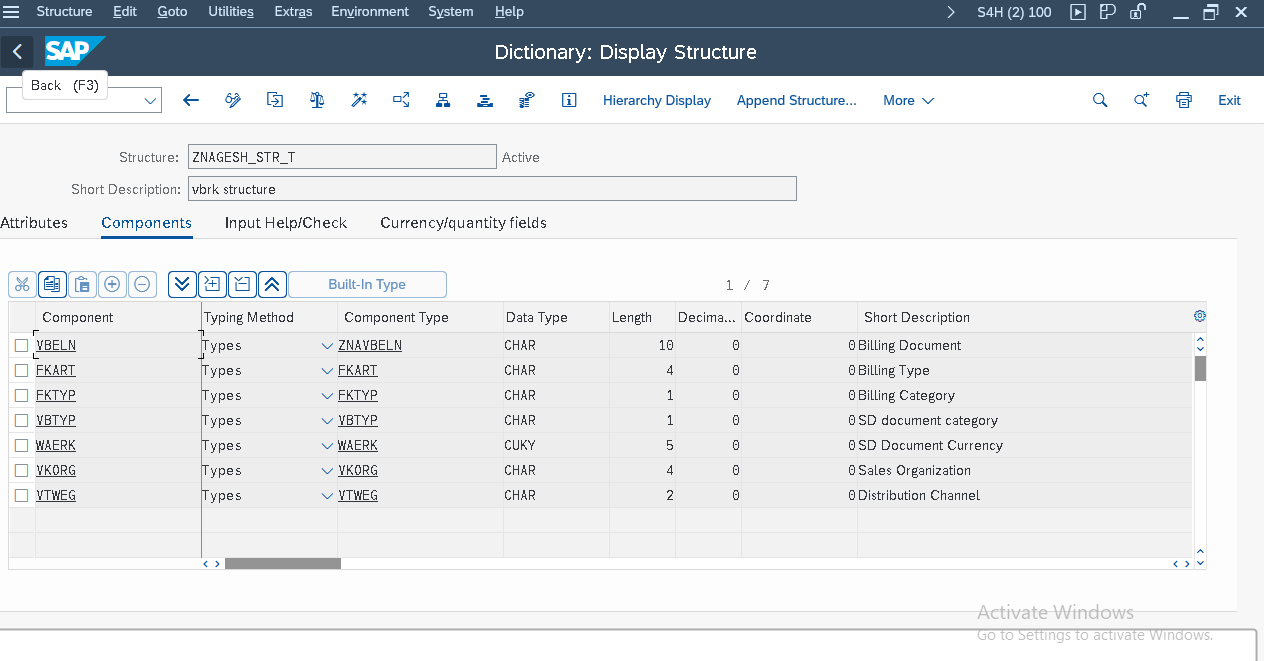


**3.PROGRAM:**

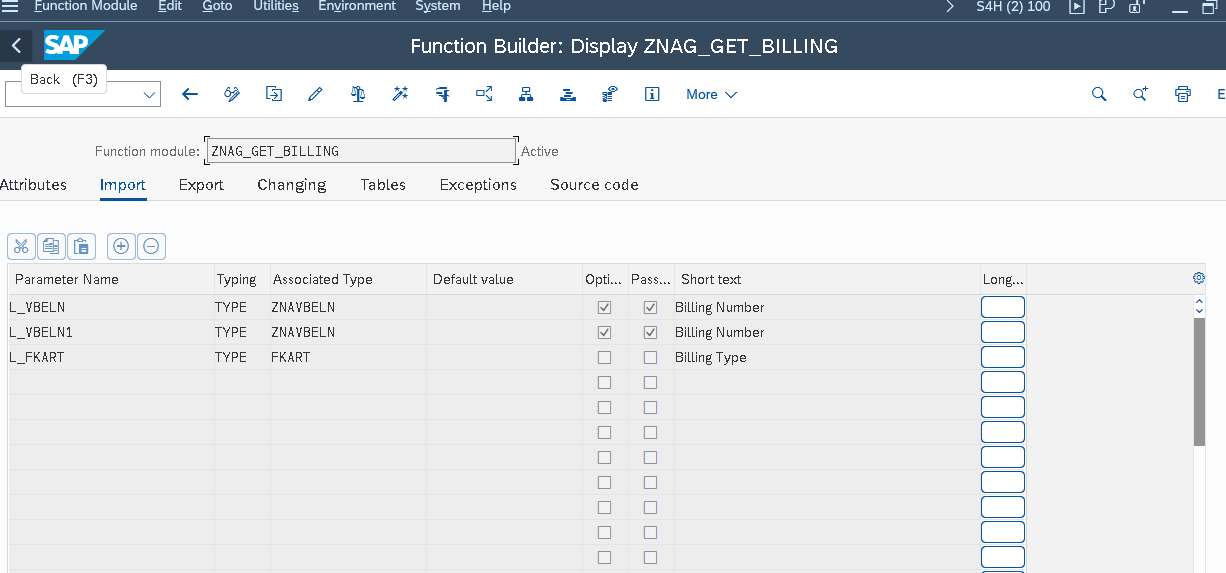
REPORT ZNAG\_SET1\_FUN.  
  
PARAMETERS p\_fkart type fkart.  
  
  data lv\_vbeln type ZNAVBELN. ""data element name  
  
  SELECT-OPTIONS s\_vbeln for lv\_vbeln

TOP-OF-PAGE.  
  
  write :/ 'VBELN',  
20'FKART',  ""spaces  
30'FKTYP',  
40'VBTYP',  
50'WAERK',  
60'VKORG',  
70'VTWEG'.  
  uline.  
  
 data : lt\_tab type STANDARD TABLE OF ZNAGESH\_STR\_T, ""structure name  
wa\_tab type ZNAGESH\_STR\_T.  ""line type name or structure name  
  
 START-OF-SELECTION.  
 CALL FUNCTION 'ZNAG\_GET\_BILLING'  
   EXPORTING  
   L\_VBELN        = s\_vbeln-low  
   L\_VBELN1       = s\_vbeln-high  
     l\_fkart        = p\_fkart     
   tables  
     gt\_tab         = lt\_tab.  
 End-of-selection.  
  
loop at lt\_tab into wa\_tab.  
  write :/ wa\_tab-VBELN,  
20 wa\_tab-FKART,  
30 wa\_tab-FKTYP,  
40 wa\_tab-VBTYP,  
50 wa\_tab-WAERK,  
60 wa\_tab-VKORG,  
70 wa\_tab-VTWEG.  
  ENDLOOP.

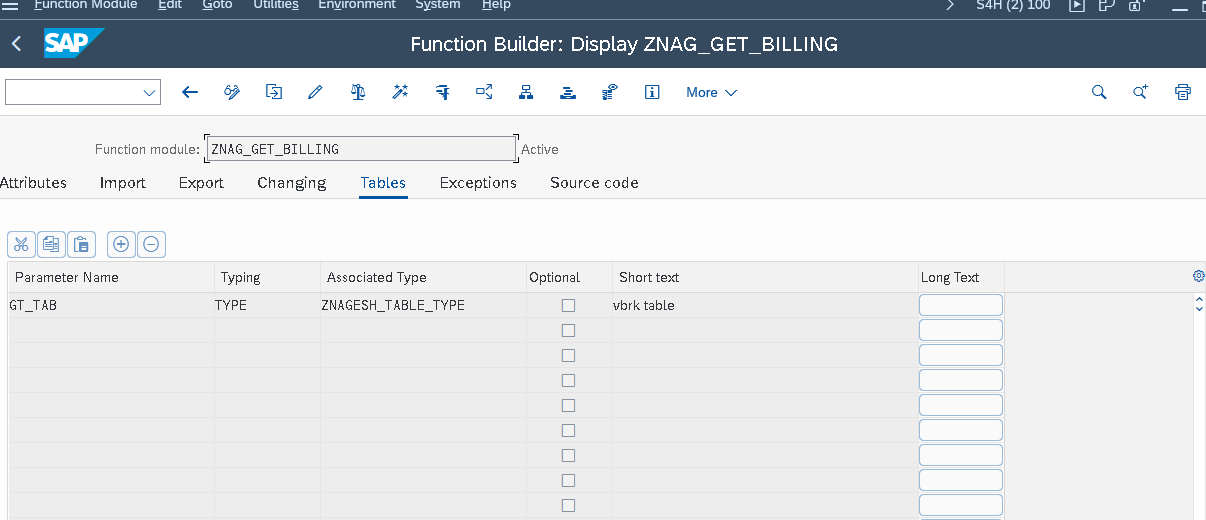
STRUCTURE:



IMPORT:



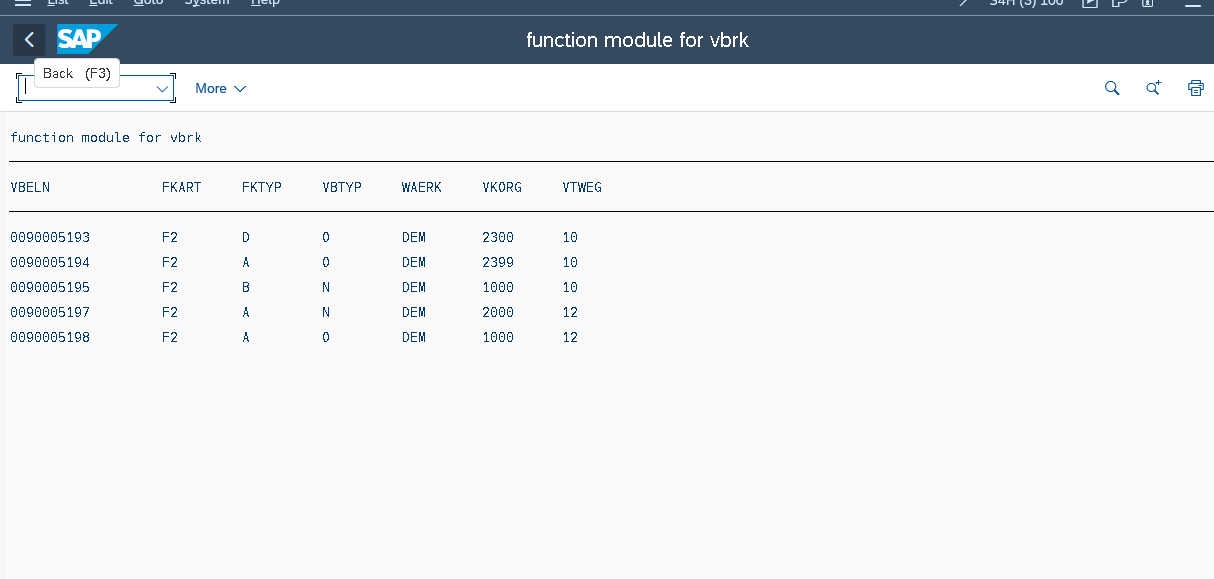
TABLES:



SOURCE CODE:

FUNCTION ZNAG\_GET\_BILLING.  
"----------------------------------------------------------------------  
"\*"Local Interface:  
"  IMPORTING  
"     VALUE(L\_VBELN) TYPE  ZNAVBELN OPTIONAL  
"     VALUE(L\_VBELN1) TYPE  ZNAVBELN OPTIONAL  
"     REFERENCE(L\_FKART) TYPE  FKART  
"  TABLES  
"      GT\_TAB TYPE  ZNAGESH\_TABLE\_TYPE  
"----------------------------------------------------------------------  
  
  
select VBELN  
FKART  
FKTYP  
VBTYP  
WAERK  
VKORG  
VTWEG  
  into table gt\_tab  
  from znvbrk  
  where vbeln between l\_vbeln and l\_vbeln1 and fkart eq l\_fkart. ""fkart eq l\_fkart is not required when there is no output in question  
  
  
ENDFUNCTION.

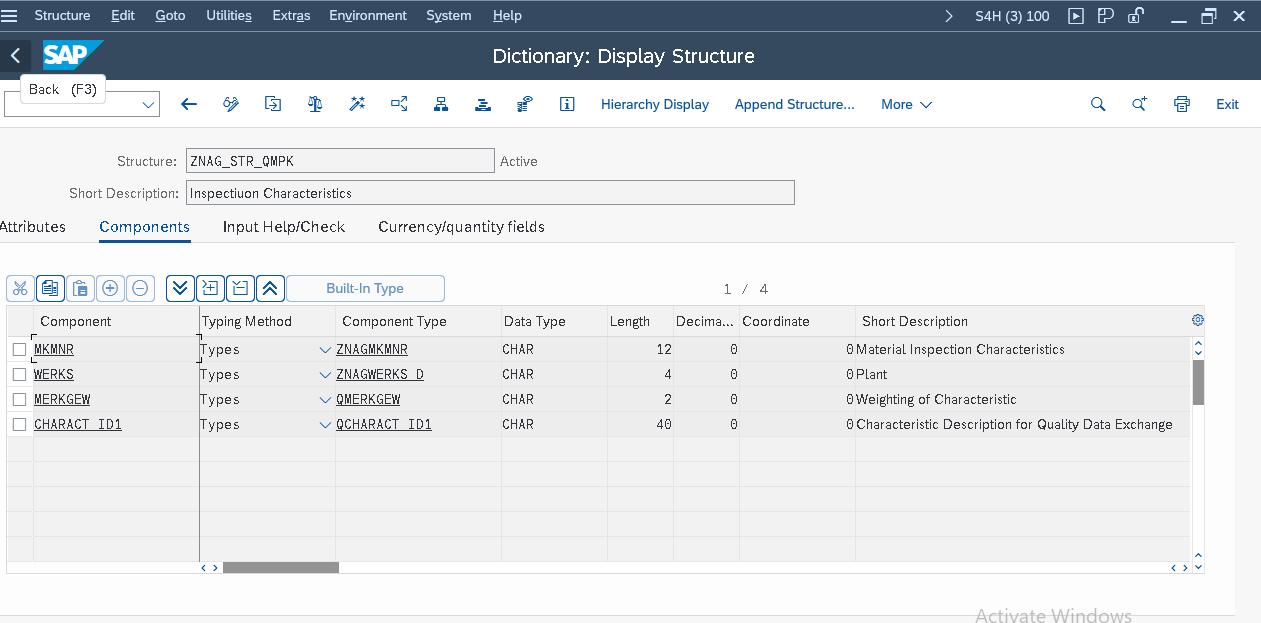
OUTPUT:



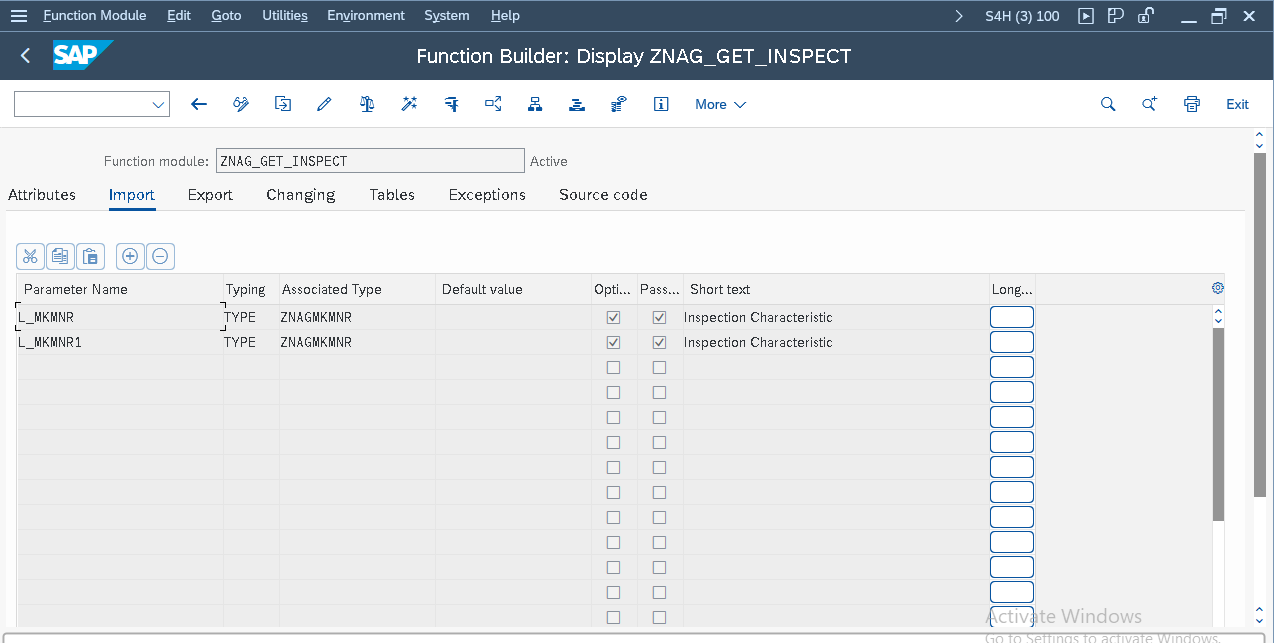
**4.Program**

REPORT znag\_exam\_fun.  
  
  
DATA lv\_mkmnr TYPE Znagmkmnr. ""data element name  
  
SELECT-OPTIONS s\_mkmnr FOR lv\_mkmnr.  
  
TOP-OF-PAGE.  
  
  WRITE :/ 'mkmnr',  
20'werks',  
30'werkgew',  
40'charact\_id1'.  
  ULINE.  
  
  DATA : lt\_tab TYPE STANDARD TABLE OF znag\_str\_qmpk, ""structure name  
         wa\_tab TYPE znag\_str\_qmpk.  ""line type name or structure name  
  
  START-OF-SELECTION.  
  
  CALL FUNCTION 'ZNAG\_GET\_INSPECT'  
EXPORTING  
  L\_MKMNR        = s\_mkmnr-low  
   L\_MKMNR1       = s\_mkmnr-high  
  TABLES  
    gt\_tab           = lt\_tab.  
  
END-OF-SELECTION.  
  
  
  
  
  LOOP AT lt\_tab INTO wa\_tab.  
    WRITE :/ wa\_tab-mkmnr,  
  20 wa\_tab-werks,  
  30 wa\_tab-merkgew,  
  40 wa\_tab-charact\_id1.  
  ENDLOOP.

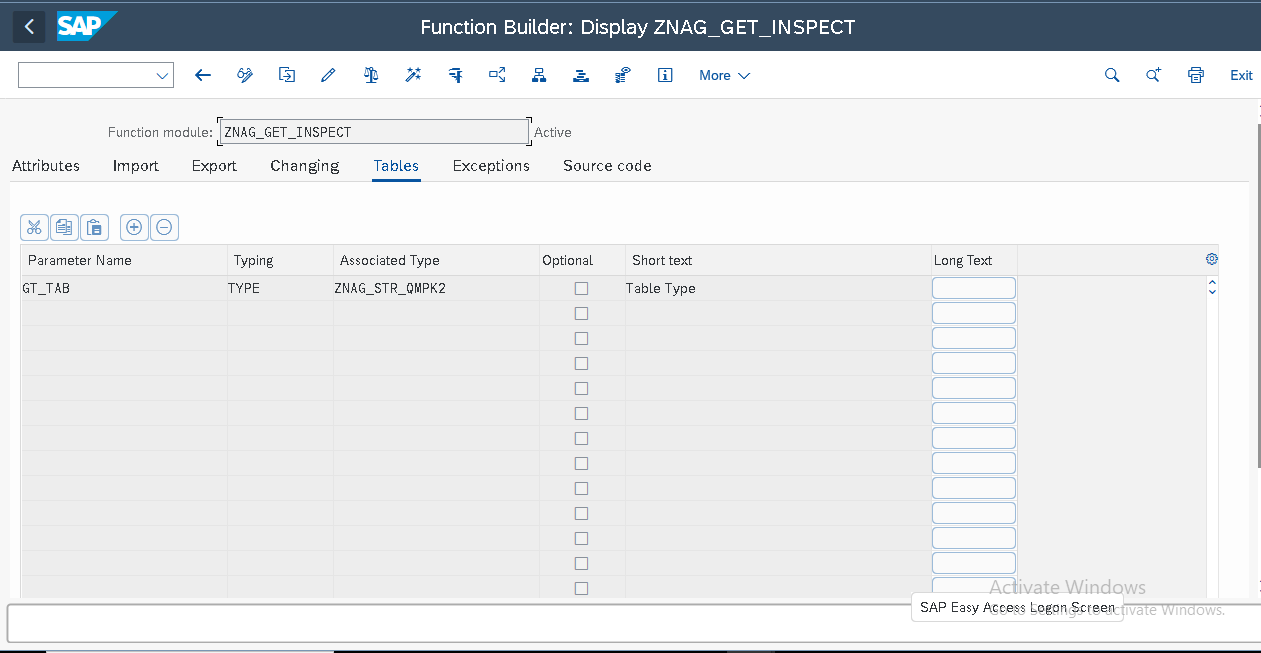
STRUCTURE:



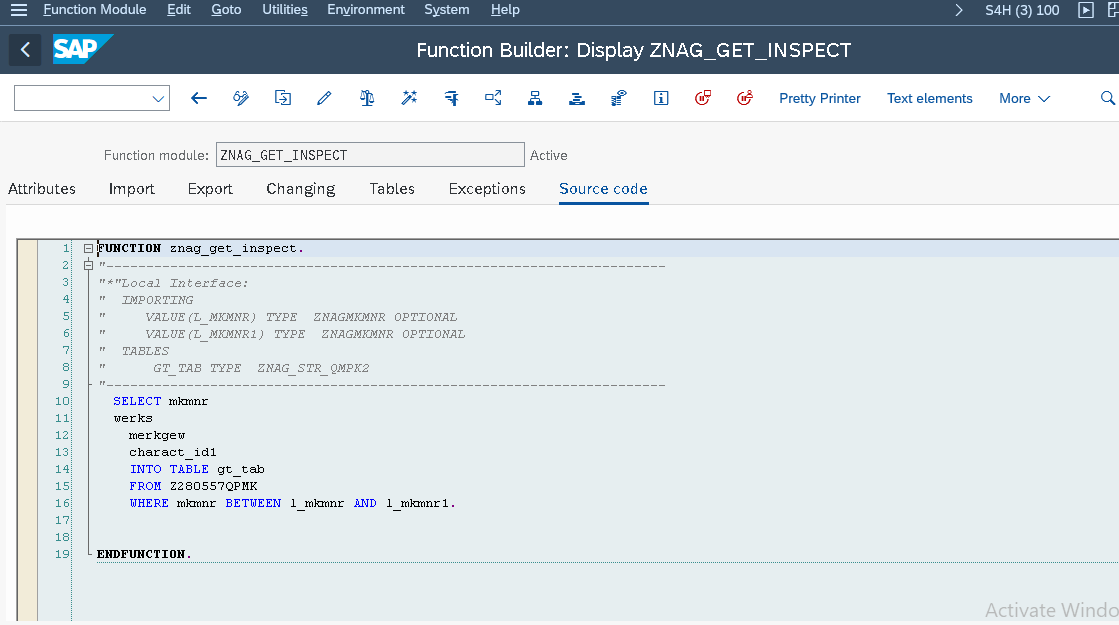
IMPORT:



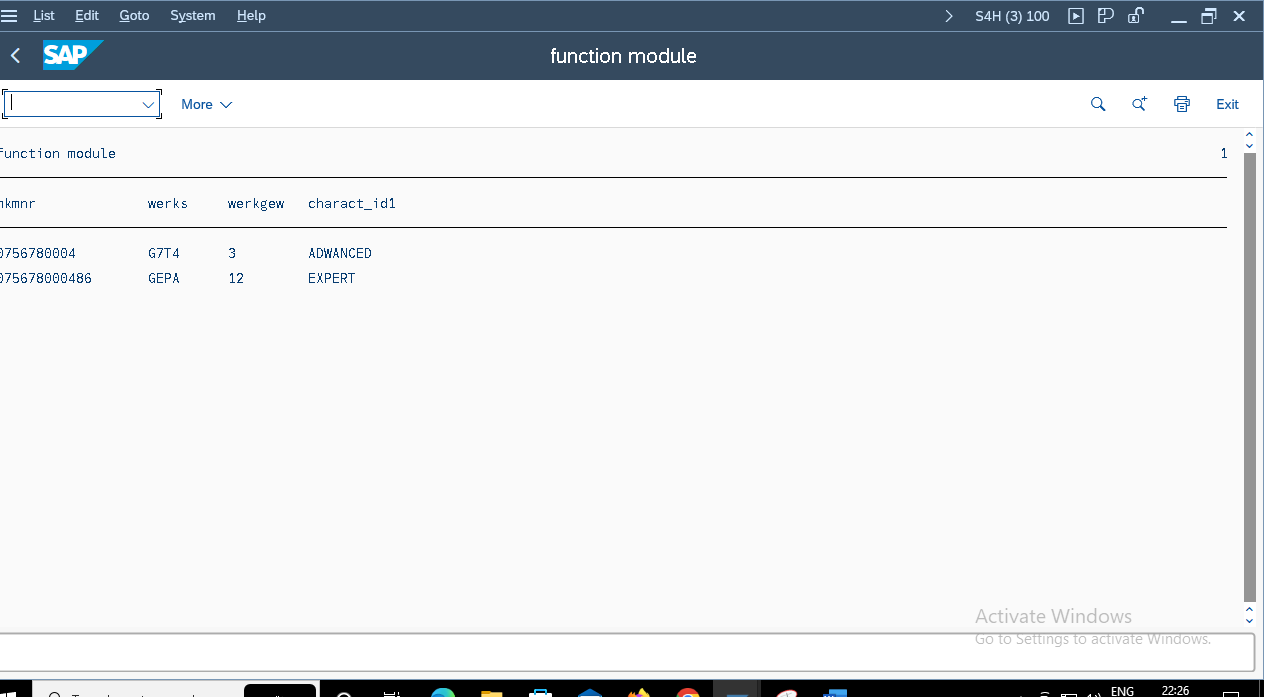
TABLES:



SOURCE CODE:



OUTPUT:



**CLASSICAL REPORTS/INTERACTIVE REPORTS**

**1.Program**

REPORT ZNAGESH\_CLASSIC\_R NO STANDARD PAGE HEADING LINE-COUNT 20(3)  
LINE-SIZE 350 MESSAGE-ID ZNAGESH\_CLASS\_R.  
  
  
LOAD-OF-PROGRAM.   ""1st event  
  
INCLUDE ZNAGESH\_INC\_I.  
  
SELECT-OPTIONS S\_VBELN FOR WA\_TAB-VBELN OBLIGATORY.  
  
INITIALIZATION.  
  
CLEAR: WA\_TAB, LT\_TAB.  
  
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 VBAP  
    WHERE VBELN IN S\_VBELN.  
  
IF SY-SUBRC EQ 0.  
  MESSAGE S000.   " SE-91 TCODE  
ELSE.  
  MESSAGE E001.   " SE-91 TCODE  
  
ENDIF.  
  
START-OF-SELECTION.  
   PERFORM GET\_VBAP\_DATA.  
  
END-OF-SELECTION.  
   PERFORM DISP\_VBAP\_DATA.  
  
   DATA LV\_COUNT TYPE I.  
   LV\_COUNT = SY-LINCT - SY-LINNO.  
   SKIP LV\_COUNT.  
  
   TOP-OF-PAGE.  
   ULINE.  
   WRITE :/ 'SALES DOC',  
          30 'MATERIAL NO',  
          50 'MAT ENTERED',  
          70 'ITEM CATEGORY',  
          85 'SALES ORDER ITEM'.  
   ULINE.  
  
   END-OF-PAGE.  
   WRITE :/ 'CURRENT LIST PAGE NO:' , SY-PAGNO,  
            'DATE: ', SY-DATUM,  
             'TIME: ', SY-UZEIT.  
  
   FORM GET\_VBAP\_DATA.  
  
     SELECT VBELN MATNR MATWA PSTYV ARKTX  
       INTO TABLE LT\_TAB  
       FROM VBAP  
       WHERE VBELN IN S\_VBELN.  
  
    IF SY-SUBRC EQ 0.  
      WRITE :/ 'RECORDS ARE FOUND IN VBAP TABLE'.  
    ELSE.  
      WRITE :/ 'RECORDS ARE NOT FOUND IN VBAP TABLE'.  
    ENDIF.  
  
   ENDFORM.  
  
  
   FORM DISP\_VBAP\_DATA.  
     LOOP AT LT\_TAB INTO WA\_TAB.  
       WRITE :/ WA\_TAB-VBELN,  
                30 WA\_TAB-MATNR,  
                50 WA\_TAB-MATWA,  
                70 WA\_TAB-PSTYV,  
                85 WA\_TAB-ARKTX.  
     ENDLOOP.  
   ENDFORM.

INCLUDE:

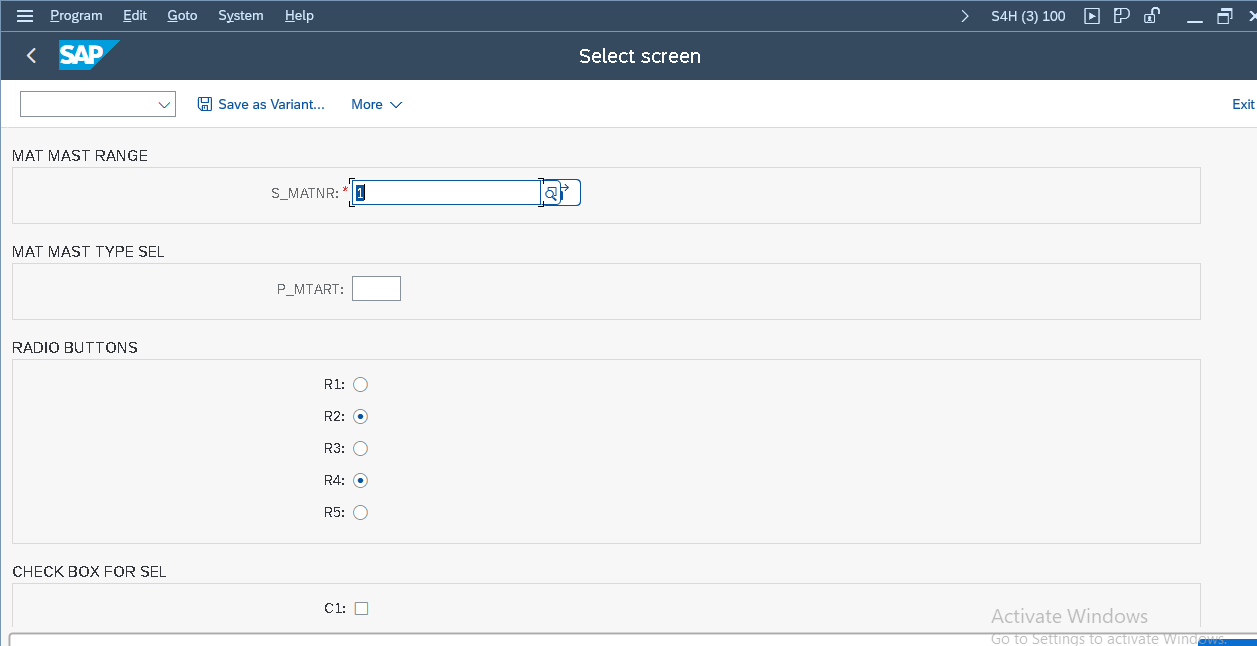
\*&---------------------------------------------------------------------\*  
\*& Include          ZNAGESH\_INC\_I  
\*&---------------------------------------------------------------------\*  
TYPES : BEGIN OF LS\_TAB,  " STRUCTURES  
        VBELN TYPE VBELN, " SALES DOCUMENT  
        MATNR TYPE MATNR, " MATERIAL NUMBER  
        MATWA TYPE MATWA, " MATERIAL ENTERED  
        PSTYV TYPE PSTYV, " SALES ITEM CATEGORY  
        ARKTX TYPE ARKTX, " SHORT TEXT FOR SALES ORDER ITEM  
      END OF LS\_TAB.  
  
  DATA : LT\_TAB TYPE STANDARD TABLE OF LS\_TAB, " INTERNAL TABLE  
        WA\_TAB TYPE LS\_TAB. "WORK AREA  
  
  DATA LV\_VBELN TYPE VBELN.  
  
  
  
  TYPES : BEGIN OF LS\_TAB2,  " STRUCTURES   ""MAKT TAble  
        MATNR type matnr,  
SPRAS type spras,  
MAKTX type maktx,  
MAKTG  type maktg," SHORT TEXT FOR SALES ORDER  
      END OF LS\_TAB2.  
  
  DATA : LT\_TAB2 TYPE STANDARD TABLE OF LS\_TAB2, " INTERNAL TABLE  
        WA\_TAB2 TYPE LS\_TAB2. "WORK AREA

**OUTPUT:**

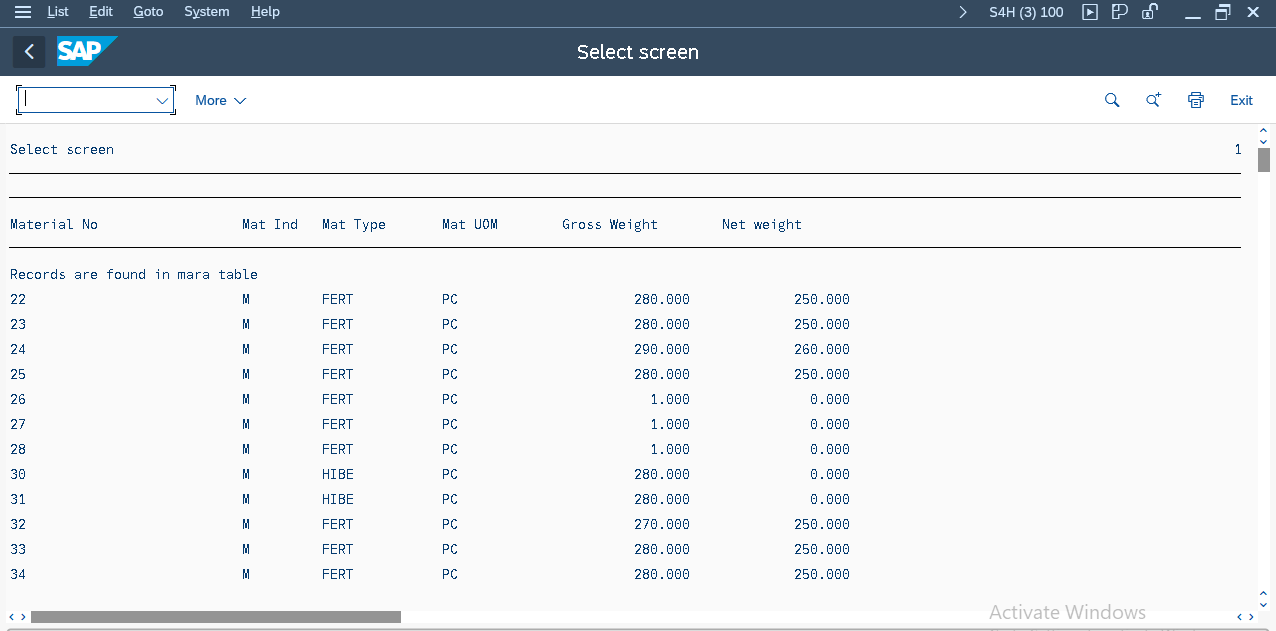


2.SELECTION SCREEN

\*&---------------------------------------------------------------------\*  
\*& Report ZNAGESH\_UST\_SEL\_SCREEN  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNAGESH\_UST\_SEL\_SCREEN  
MESSAGE-Id ZNAG\_MSG\_SELECT\_S LINE-COUNT 25(3) LINE-SIZE 500.  
  
TYPES : BEGIN OF ls\_tab,  " strs  
          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.  
  
DATA : lt\_tab TYPE STANDARD TABLE OF ls\_tab,  " itab  
       wa\_tab TYPE ls\_tab.   " work area  
  
DATA lv\_matnr TYPE matnr.  
  
SELECTION-SCREEN : BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-001.  
  
\*SELECT-OPTIONS s\_matnr FOR wa\_tab-matnr OBLIGATORY NO-EXTENSION.  
SELECT-OPTIONS s\_matnr FOR wa\_tab-matnr OBLIGATORY NO INTERVALS.  
  
  
SELECTION-SCREEN END OF BLOCK B1.  
  
  
SELECTION-SCREEN : BEGIN OF BLOCK B2 WITH FRAME TITLE TEXT-002.  
  
PARAMETERS p\_mtart TYPE mtart.  
  
SELECTION-SCREEN END OF BLOCK B2.  
  
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,  
  
  
             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 DEFAULT 'X'.  
SELECTION-SCREEN END OF BLOCK B4.  
  
  
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  
\*\*\*\*\* dummy sql code to validate the s\_matnt input  
  SELECT SINGLE matnr  
    INTO lv\_matnr  
     FROM mara  
     WHERE matnr IN s\_matnr.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s000.  " se91 - tcode.  
\*\*\* MESSAGE S000(ztrupen\_ust\_msg).  " se91 - tcode.  
\*\*\*\*\*\*\*\*\*\*\* MESSAGE S124(00).  " se91 - tcode.  
\*  
\*\*\*\*MESSAGE i002(ztrupen\_ust\_msg).  " se91 - tcode.  
  ELSE.  
    MESSAGE e001.  " se91 - tcode.  
\*\*\*MESSAGE E001(ztrupen\_ust\_msg).  " se91 - tcode.  
\*\*\*\*MESSAGE w003(ztrupen\_ust\_msg).  " se91 - tcode.  
\*\*\*\*MESSAGE x003(ztrupen\_ust\_msg).  " se91 - tcode.  
\*\*MESSAGE a003(ztrupen\_ust\_msg).  " se91 - tcode.  
  ENDIF.  
  
  
  
START-OF-SELECTION.  
  PERFORM get\_mat\_data.  
  
  
END-OF-SELECTION.  
  PERFORM disp\_mat\_data.  
  
  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.  
  
\*&---------------------------------------------------------------------\*  
\*& Form get\_mat\_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.  
    WRITE :/ 'Records are found in mara table'.  
  ELSE.  
    WRITE : / 'Records are not found in mara table'.  
  ENDIF.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form disp\_mat\_data  
\*&---------------------------------------------------------------------\*  
  
FORM disp\_mat\_data .  
  LOOP AT lt\_tab INTO wa\_tab.  
    WRITE : / wa\_tab-matnr,  
             30 wa\_tab-mbrsh,  
             40 wa\_tab-mtart,  
             55 wa\_tab-meins,  
             70 wa\_tab-brgew,  
             90 wa\_tab-ntgew.  
  ENDLOOP.  
ENDFORM.

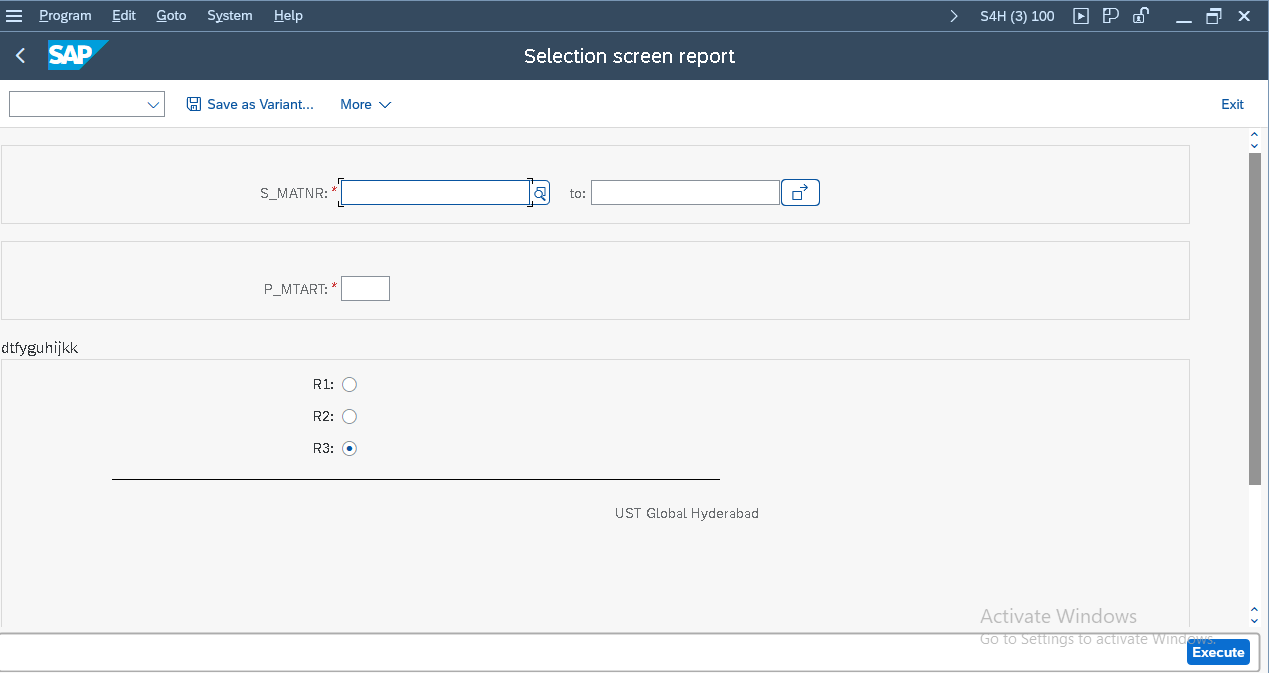


OUTPUT:

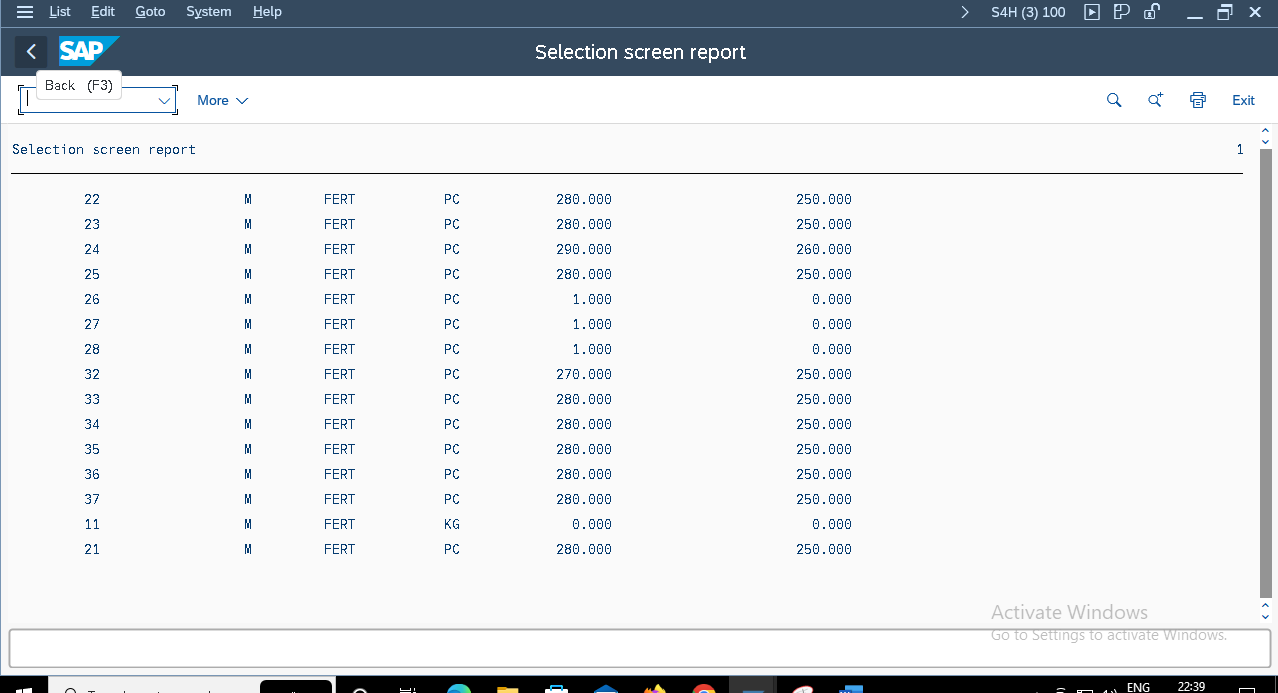


3.SELECTION SCREEN REPORT

REPORT ZNAGESH\_UST\_SEL\_SCREEN\_REPORT.  
  
DATA lv\_matnr TYPE matnr.   " F TYPE DTL  
  
TYPES : BEGIN OF ty\_tab,   " local strs  
  matnr TYPE matnr,  
  mbrsh TYPE mbrsh,  
  mtart TYPE mtart,  
  meins TYPE meins,  
  brgew TYPE brgew,  
  ntgew TYPE ntgew,  
END OF ty\_tab .  
  
DATA : it\_tab TYPE STANDARD TABLE OF ty\_tab,  " INTERNAL TABLE  
      wa\_tab TYPE ty\_tab. " WORK AREA  
  
SELECTION-SCREEN : BEGIN OF BLOCK B2 WITH FRAME TITLE TEXT-002.  ""1.block""  
  
  SELECT-options s\_matnr FOR lv\_matnr OBLIGATORY . "   # 1  
\*  SELECT-options s\_matnr FOR lv\_matnr OBLIGATORY NO-extension. " # 2.  
\*SELECT-options s\_matnr FOR lv\_matnr OBLIGATORY NO INTERVALS. " # 3.  
  
SELECTION-SCREEN : END OF BLOCK B2.  
  
  
SELECTION-SCREEN : BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-001. ""2. Block  
  
PARAMETERS P\_MTART TYPE MTART OBLIGATORY.  
  
SELECTION-SCREEN : END OF BLOCK B1 .  
  
  
SELECTION-SCREEN : BEGIN OF BLOCK B3 WITH FRAME TITLE TEXT-003.  ""3.Block For Radio Buttons  
  
PARAMETERS : R1 RADIOBUTTON GROUP RAD1,  
             R2 RADIOBUTTON GROUP RAD1,  
             R3 RADIOBUTTON GROUP RAD1 DEFAULT 'X'.  
  
SELECTION-SCREEN ULINE /10(60).  ""Under line  
SELECTION-SCREEN COMMENT /60(70) lv\_comm1. ""We cant write to print the statement, so here we have written this stmnt  
SELECTION-SCREEN SKIP 3.    ""After skip 3 lines , we can see another line.  
SELECTION-SCREEN ULINE.  
  
SELECTION-SCREEN : END OF BLOCK B3.  
  
INITIALIZATION.  ""Event 1 all declarations will be held in initialization  
\*data lv\_comm1 TYPE c VALUE 'UST Global Hyderabad'.  
  
lv\_comm1 = 'UST Global Hyderabad'.  
  
  
  
SELECTION-SCREEN : BEGIN OF BLOCK B4 WITH FRAME TITLE TEXT-004.    ""Event 2  
  
PARAMETERS : C1 AS CHECKBOX ,  
             C2 AS CHECKBOX DEFAULT 'X'.  
SELECTION-SCREEN : END OF BLOCK B4.  
  
LOOP AT s\_matnr.  
  ULINE.  
WRITE : 'Mat Low :' , s\_matnr-low,  
        'Mat High :' , s\_matnr-high,  
        'Range Sign :' , s\_matnr-sign,  
        'Range Options :' , s\_matnr-option.  
ULINE.  
  ENDLOOP.  
  
  
  start-OF-selection.   ""Event 3  
SELECT matnr mbrsh mtart meins brgew ntgew  
FROM mara  
INTO TABLE it\_tab  
WHERE matnr IN s\_matnr AND  
      MTART EQ P\_MTART.  
  
  
  END-OF-selection.    ""Event 4  
LOOP AT it\_tab INTO wa\_tab.  " B TO W  
  WRITE :/10 wa\_tab-matnr ,   "  58  
  30 wa\_tab-mbrsh,   " W-F       ""30 is the length for spaces  
  40 wa\_tab-mtart,  
  55 wa\_tab-meins,  
  60 wa\_tab-brgew,  
  90 wa\_tab-ntgew.  
ENDLOOP.



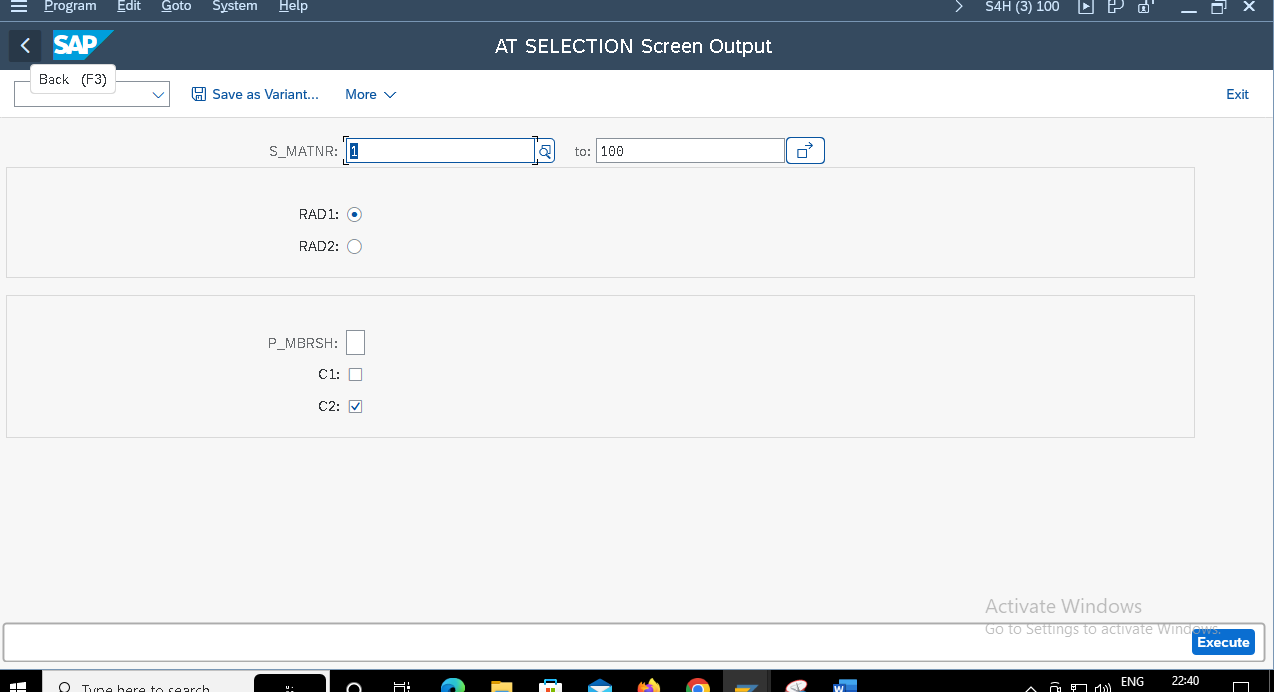
OUTPUT:



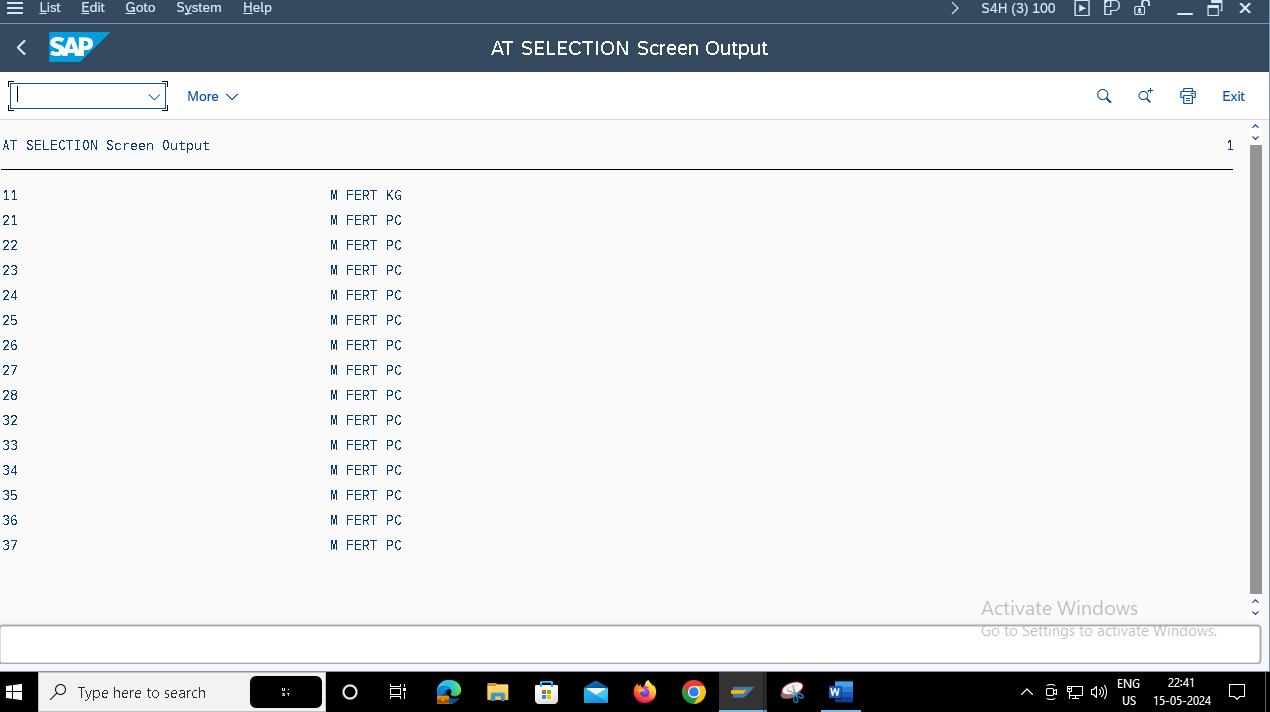
4.SELECTION SCREEN-OUTPUT

\*&---------------------------------------------------------------------\*  
\*& Report ZNAG\_AT\_SELS\_OUTPUT  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNAG\_AT\_SELS\_OUTPUT.  
  
INCLUDE ZNagesh\_ATSELSCR\_TOP.  
  
SELECT-OPTIONS S\_MATNR FOR LV\_MATNR default '1' TO '100'.  
  
  
SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-T01.  
  
  PARAMETERS:RAD1 RADIOBUTTON GROUP RAD   USER-COMMAND FLAG  DEFAULT 'X',  
            RAD2 RADIOBUTTON GROUP RAD ."DEFAULT 'X'.  
  
SELECTION-SCREEN END OF BLOCK B1.  
  
  
SELECTION-SCREEN BEGIN OF BLOCK B2 WITH FRAME TITLE TEXT-T02 .  
  PARAMETERS : P\_MBRSH TYPE MBRSH MODIF ID M1,  " INDustry SECTOR  
              P\_MTART TYPE MTART MODIF ID M2.  " MATerial  TYPE  
  
  PARAMETERS : c1 AS CHECKBOX ,  
              c2 AS CHECKBOX DEFAULT 'X' .  
  
SELECTION-SCREEN END OF BLOCK B2.  
  
  
AT SELECTION-SCREEN OUTPUT.  
  
LOOP AT SCREEN.  
  IF RAD1 = 'X' AND  
  SCREEN-GROUP1 = 'M2'.  
    SCREEN-ACTIVE = 0.  
\*    SCREEN-INVISIBLE = 1.  
ELSEIF RAD2 = 'X' AND  
  SCREEN-GROUP1 = 'M1'.  
    SCREEN-ACTIVE = 0.  
  ENDIF.  
  MODIFY SCREEN.  
ENDLOOP.  
  
START-OF-SELECTION.  
IF RAD1 = 'X'.  
  PERFORM GET\_MBRSH.  
ELSE.  
  PERFORM GET\_MTART.  
ENDIF.  
\*&---------------------------------------------------------------------\*  
\*&      Form  GET\_MBRSH  
\*&---------------------------------------------------------------------\*  
FORM GET\_MBRSH .  
  SELECT MATNR  
  MBRSH  
  MTART  
  MEINS  
  INTO TABLE IT\_TAB  
  FROM MARA  
  WHERE MATNR IN S\_MATNR AND MBRSH EQ P\_MBRSH .  
SORT IT\_TAB BY MATNR.  
  
  
  LOOP AT IT\_TAB INTO WA\_TAB.  
    WRITE : / WA\_TAB-MATNR,  
    WA\_TAB-MBRSH,  
    WA\_TAB-MTART,  
    WA\_TAB-MEINS.  
  
  ENDLOOP.  
  
ENDFORM.                    " GET\_MBRSH  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  GET\_MTART  
\*&---------------------------------------------------------------------\*  
  
FORM GET\_MTART .  
  SELECT MATNR  
  MBRSH  
  MTART  
  MEINS  
  INTO TABLE IT\_TAB  
  FROM MARA  
  WHERE MATNR IN S\_MATNR AND MTART EQ P\_MTART.

SORT  IT\_TAB BY MATNR .  
  LOOP AT IT\_TAB INTO WA\_TAB.  
    WRITE : / WA\_TAB-MATNR,  
    WA\_TAB-MBRSH,  
    WA\_TAB-MTART,  
    WA\_TAB-MEINS.  
  ENDLOOP.  
  
ENDFORM.



OUTPUT:



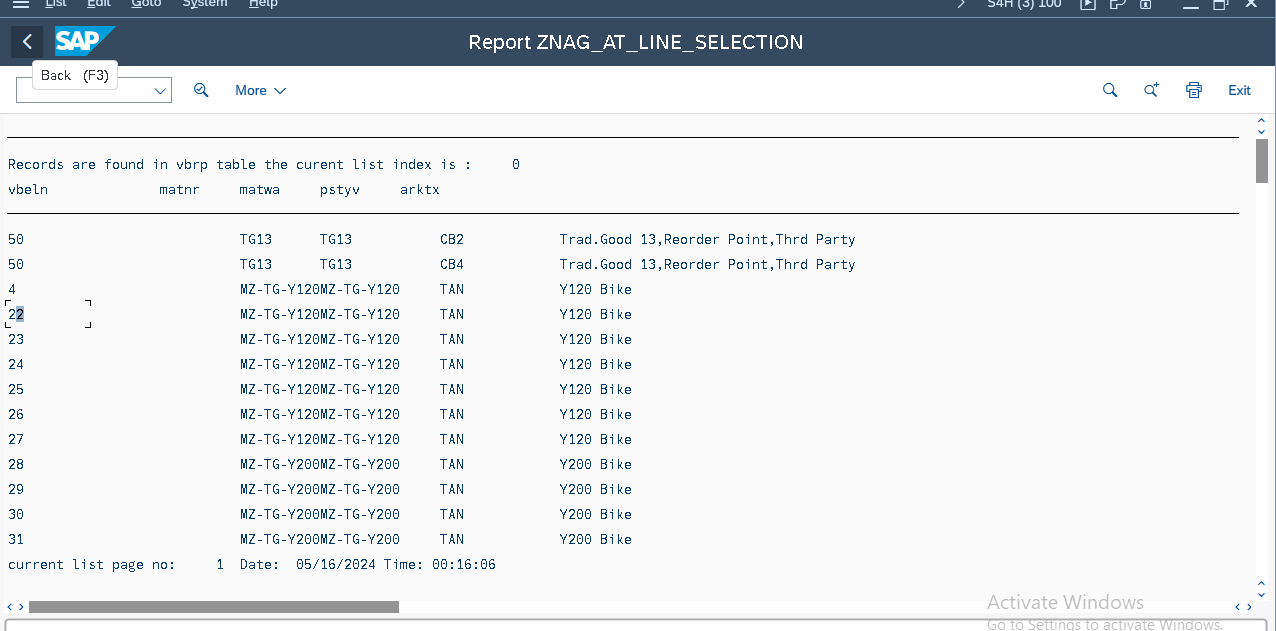
5.AT-LINE\_SELECTION:

REPORT znag\_at\_line\_selection NO STANDARD PAGE HEADING LINE-COUNT 20(3)  
LINE-SIZE 500 MESSAGE-ID znagesh\_class\_r.  
  
LOAD-OF-PROGRAM.   ""1  
  INCLUDE Znag\_at\_line\_inc.  
  
  SELECT-OPTIONS s\_vbeln FOR wa\_tab-vbeln OBLIGATORY.  
  
INITIALIZATION.     """2  
  CLEAR: wa\_tab, lt\_tab.  
  s\_vbeln-low = '90000001'.  
  s\_vbeln-high = '90000050'.  
  APPEND s\_vbeln.  
  
AT SELECTION-SCREEN ON s\_vbeln. ""3  
  
  SELECT SINGLE vbeln  
    INTO lv\_vbeln  
    FROM vbap  
    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\_data.  
  
END-OF-SELECTION.  
  
  PERFORM display\_data.  
  lv\_count = sy-linct - sy-linno.  
  SKIP lv\_count.  
  
TOP-OF-PAGE.  
  ULINE.  
  WRITE :/ 'Records are found in vbrp table', 'the curent list index is :  ', sy-lsind.  
  WRITE :/ 'vbeln',  
  20 'matnr',  
  30 'matwa',  
  40 'pstyv',  
  50 'arktx'.  
  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 vbeln posnr uepos fkimg vrkme  
        INTO TABLE lt\_tab2  
        FROM vbrp  
      WHERE vbeln EQ wa\_tab-vbeln.  ""Hide Stmt  
  
      LOOP AT lt\_tab2 INTO wa\_tab2.  
        WRITE : /3 wa\_tab2-vbeln HOTSPOT,  
                 20 wa\_tab2-posnr,  
                 30 wa\_tab2-uepos,  
                 40 wa\_tab2-fkimg,  
                 50 wa\_tab2-vrkme.  
  
        HIDE wa\_tab2-vbeln.  
      ENDLOOP.  
  
    WHEN 2.  
  
      SELECT vbeln fkart fktyp vbtyp waerk  
     INTO TABLE lt\_tab3  
     FROM vbrk  
      WHERE vbeln EQ wa\_tab2-vbeln.  ""Hide Stmt  
  
      LOOP AT lt\_tab3 INTO wa\_tab3.  
        WRITE : /3 wa\_tab3-vbeln,   """" hotspot,  
                 20 wa\_tab3-fkart,  
                 30 wa\_tab3-fktyp,  
                 40 wa\_tab3-vbtyp,  
                 50 wa\_tab3-waerk.  
      ENDLOOP.  
  ENDCASE.  
  
TOP-OF-PAGE DURING LINE-SELECTION.  
  
  CASE sy-lsind.  
    WHEN 1.  
      WRITE :/ 'Records are found in vbrp  table', 'the curent list index is :  ', sy-lsind.  
      ULINE.  
      WRITE :/3 'vbeln',  
             30 'posnr',  
             40 'uepos',  
             50 'fkimg',  
             60 'vrkme'.  
      ULINE.  
  
    WHEN 2.  
      WRITE : / 'Records are found in vbrk table', 'the curent list index is :  ', sy-lsind.  
      ULINE.  
      WRITE :/3 'vbeln',  
             20 'fkart',  
             30 'fktyp',  
             40 'vbtyp',  
             50 'waerk'.  
  
  
      ULINE.  
  ENDCASE.  
  
  
  
  
  
  
FORM get\_data .  
  SELECT vbeln matnr matwa pstyv arktx  
      INTO TABLE lt\_Tab  
      FROM vbap  
      WHERE vbeln IN  s\_vbeln.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s000.  
  ELSE.  
    MESSAGE e001.  
  ENDIF.  
  
ENDFORM.  
  
\*&---------------------------------------------------------------------\*  
FORM display\_data .  
  LOOP AT lt\_tab INTO wa\_tab.  
    WRITE : / wa\_tab-vbeln  HOTSPOT,  
             30 wa\_tab-matnr,  
             40 wa\_tab-matwa,  
             55 wa\_tab-pstyv,  
             70 wa\_tab-arktx.  
  
  
    HIDE wa\_tab-vbeln.  " for next where  
  
  ENDLOOP.  
  
ENDFORM.

INCLUDE:

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAG\_AT\_LINE\_INC  
\*&---------------------------------------------------------------------\*  
types : begin of ls\_tab,  ""structure for VBAP  
           vbeln type vbeln,  
  matnr type matnr,  
  matwa type matwa,  
  pstyv type pstyv,  
  arktx type arktx,  
  
        end of ls\_tab,  
  
        begin of ls\_tab2,   ""Structure for VBRP  
           VBELN type vbeln,  
POSNR type posnr,  
UEPOS type posnr,  
FKIMG type fkimg,  
VRKME type vrkme,  
  
        end of ls\_tab2,  
  
        begin of ls\_tab3,   ""structure of vbrk  
  
          VBELN type vbeln,  
FKART type fkart,  
FKTYP type fktyp,  
VBTYP  type  vbtyp,  
WAERK type waerk,  
  
        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\_vbeln type vbeln,  ""vbeln is the data element  
       lv\_count TYPE i.

OUTPUT:



6.AT\_USER\_COMMAND

\*&---------------------------------------------------------------------\*  
\*& Report ZNAG\_AT\_USER\_COMMAND  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT znag\_at\_user\_command no STANDARD PAGE HEADING line-count 20(3)  
line-size 500 MESSAGE-ID ZNAGESH\_CLASS\_R.  
INCLUDE Znag\_AT\_User\_Inc.  
  
SELECT-OPTIONS s\_vbeln FOR wa\_tab-vbeln OBLIGATORY.  
  
INITIALIZATION.  
  CLEAR: wa\_tab, lt\_tab.  
  
  s\_vbeln-low = '1'.  
  s\_vbeln-high = '100'.  
  APPEND s\_vbeln.  
  
AT SELECTION-SCREEN ON s\_vbeln. "120 to 200  
  
  SELECT SINGLE vbeln  
    INTO lv\_vbeln  
    FROM vbap  
    WHERE vbeln IN s\_vbeln.  
  
  
  IF sy-subrc EQ 0.  
    MESSAGE s000.   ""Tcode for message creation se91  
  ELSE.  
    MESSAGE e001.  
  ENDIF.  
  
START-OF-SELECTION.  
  SET PF-STATUS 'ZNAGESH\_PFS' EXCLUDING 'INFO'. ""se41 for menu painter  
  
  PERFORM get\_vbap\_data.  
END-OF-SELECTION.  
  
  PERFORM disp\_vbap\_data.  
  
  
\*&---------------------------------------------------------------------\*  
\*& Form disp\_vbap\_data  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
  
  
DATA lv\_count TYPE i.  
lv\_count = sy-linct - sy-linno. ""line count and line number  
SKIP lv\_count.  
  
TOP-OF-PAGE.  
  uline.  
  WRITE :/ 'Records are found in vbap table', 'the current list index is : ',sy-lsind.  
  WRITE :/ 'sales document',  
         30 'Material Number',  
         40 'Material entered',  
         50 'Item category',  
         60 'Sales Order Item'.  
  ULINE.  
  
END-OF-PAGE.  
  WRITE :/ 'current ist page no:' , sy-pagno,  
  'Date:',sy-datum,  
  'Time:',sy-uzeit.  
  
AT USER-COMMAND.  
  
  CASE  sy-ucomm. ""user command  
    WHEN 'DISP'.  
      WRITE 'UST INFO'.  
    WHEN 'INFO'.  
      WRITE /: 'the current date:',sy-datum,  
      'TimeL',sy-uzeit.  
  
    WHEN 'TCODE'.  
      CALL TRANSACTION 'ZNAG\_SALESORDER\_TCOD'.  
  
      WHEN 'sap'.  
      WRITE 'SAP LABS BNGL'.  
      WHEN 'ABAP'.  
      WRITE 'SAP ABAP HYD'.  
      ENDCASE.  
  
form get\_vbap\_data.  
  SELECT vbeln matnr matwa pstyv arktx  
      INTO TABLE lt\_Tab  
      FROM vbap  
      WHERE vbeln IN  s\_vbeln.  
  IF sy-subrc EQ 0.  
    WRITE :/ 'Records are found in vbap table'.  
  ELSE.  
    WRITE : / 'Records are not found in vbap table'.  
  ENDIF.  
  
ENDFORM.  
  
FORM disp\_vbap\_data .  
  LOOP AT lt\_tab INTO wa\_tab.  
    WRITE : / wa\_tab-vbeln  HOTSPOT,  
             30 wa\_tab-matnr,  
             40 wa\_tab-matwa,  
             55 wa\_tab-pstyv,  
             70 wa\_tab-arktx.  
  
  
    HIDE wa\_tab-vbeln.  " for next where  
endloop.  
ENDFORM.

INCLUDE:

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAG\_AT\_USER\_INC  
\*&---------------------------------------------------------------------\*  
  
types : begin of ls\_tab,      ""Columns from VBAP Table  
          vbeln type vbeln,     ""Primary key of vbap table  
          matnr type matnr,  
          matwa type matwa,  
          pstyv type pstyv,  
          arktx type arktx,  
  
   end of ls\_tab.  
  
data : lt\_tab type standard table of ls\_tab,  " itab  
       wa\_tab type ls\_tab.   " work area  
  
DATA LV\_vbeln TYPE vbeln.

OUTPUT:



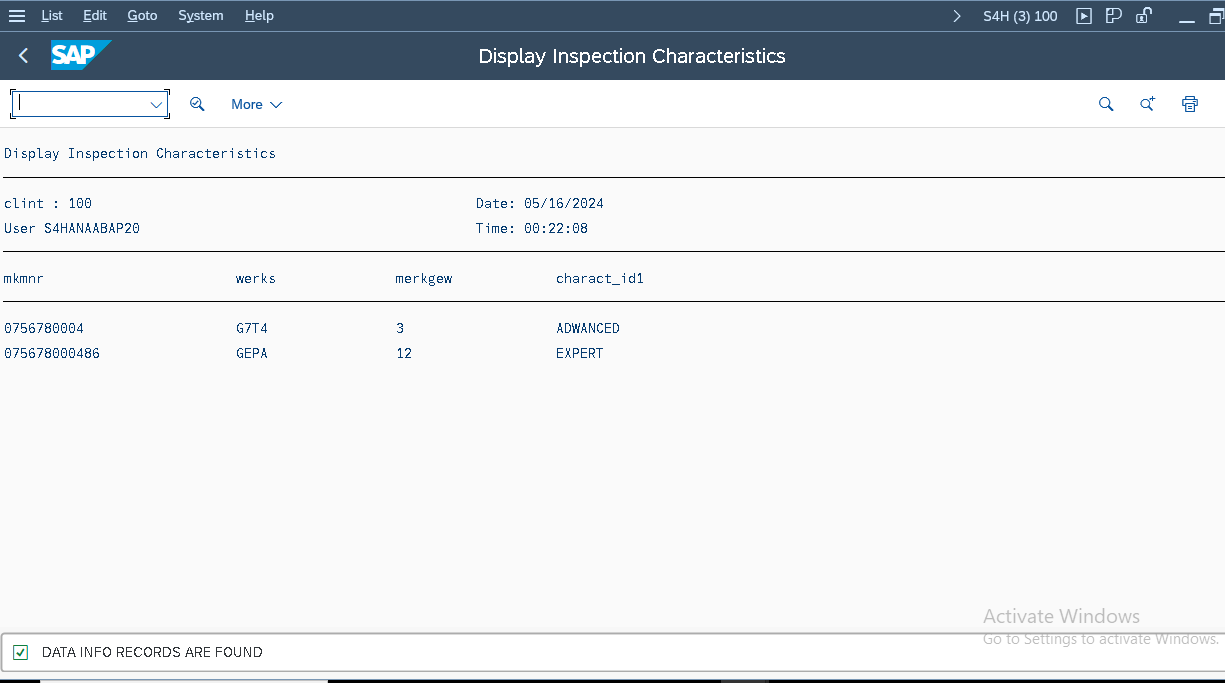
8.BASIC LIST/SECONDARY LIST (INTERACTIVE REPORT)

\*&---------------------------------------------------------------------\*  
\*& Report ZNAG\_UST\_INT  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNAG\_UST\_INT.  
  
Include Znag\_INT\_inc.  
Data l\_MKMNR1 type ZNAGMKMNR.  
  
INITIALIZATION .   ""Providing the default values  
  s\_mkmnr-low ='0756780004'.  
  s\_mkmnr-high = '075678000486'.  
  APPEND s\_MKMNR.  
  
  
  AT SELECTION-SCREEN ON s\_MKMNR.  
  
  SELECT SINGLE MKMNR  
    INTO l\_MKMNR1  
  FROM Z280557QPMK  
  WHERE MKMNR IN s\_MKMNR.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s000(ZNAGESH\_CLASS\_R).  
  ELSE .  
    MESSAGE e001(ZNAGESH\_CLASS\_R).  
  ENDIF.  
  
  
  START-OF-SELECTION.  
  SELECT MKMNR WERKS MERKGEW CHARACT\_ID1  
  INTO TABLE lt\_tab1  
  FROM Z280557QPMK WHERE MKMNR IN s\_MKMNR.  
  
    LOOP AT lt\_tab1 INTO wa\_tab1.  
    WRITE :/ wa\_tab1-MKMNR HOTSPOT ,30 wa\_tab1-WERKS,50 wa\_tab1-MERKGEW,70 wa\_tab1-CHARACT\_ID1. ""Hotspot for finger symbol  
    hide wa\_tab1-MKMNR.   "" To collect the data from one line selection to another selection  
  ENDLOOP.  
  
END-OF-SELECTION.  
  
  
TOP-OF-PAGE.  
  WRITE:/  'clint : 100' , 60  'Date:', sy-datum,/  
  'User',sy-uname, 60  'Time:', sy-uzeit.  
  
  ULINE.  
  WRITE :/  'mkmnr',  
         30 'werks',  
         50 'merkgew',  
         70 'charact\_id1'.  
  uline.   ""Basic list was done  
  
  
  
  AT LINE-SELECTION.   ""Secondary list   ""At line selection program  
     CASE sy-lsind.  ""list index  
    WHEN 1.  
  
      SELECT  VERSION PMETHODE AUSWMENGEI  
      INTO TABLE lt\_Tab2  
      FROM Z280557QPMZ  
      WHERE MKMNR EQ wa\_tab1-MKMNR.  
  
      LOOP AT lt\_tab2 INTO wa\_tab2.  
        WRITE : /   wa\_tab2-VERSION ,  
                 20 wa\_tab2-PMETHODE,  
                 30 wa\_tab2-AUSWMENGEI.  
  
      ENDLOOP.  
     endcase.  
  TOP-OF-PAGE DURING LINE-SELECTION.  
  WRITE:/  'clint : 100' , 60  'date:', sy-datum,/  
  'User',sy-uname, 60  'Time:', sy-uzeit,/  
  'mkmnr :',wa\_tab1-MKMNR.  
  
  ULINE.  
  WRITE :/ 'version',  ""Give short description  
        20 'pmethode',  
         30 'auswmenget'.  
  
  uline.

INCLUDE :

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAG\_INT\_INC  
\*&---------------------------------------------------------------------\*  
Types : begin of ls\_tab1,  "local structure 1 VBRK  
  MKMNR TYPE  ZNAGMKMNR,  
WERKS  TYPE ZNAGWERKS\_D,  
MERKGEW  TYPE QMERKGEW,  
CHARACT\_ID1  TYPE QCHARACT\_ID1,  
  End of ls\_tab1.  
  
  Types : begin of ls\_tab2,   ""local structure table 2  
    VERSION TYPE  ZNAGVERSION,  
PMETHODE  TYPE QPMETHODE,  
AUSWMENGEI TYPE  QCGRAUSW,  
  End of ls\_tab2.  
  
  Data : lt\_tab1 type STANDARD TABLE OF ls\_tab1,  ""lt\_tab1 and wa\_tab1 referred to ls\_tab1  
        wa\_tab1 type ls\_tab1,  
  
  lt\_tab2 type STANDARD TABLE OF ls\_tab2,  
  wa\_tab2 type ls\_tab2,  
  
  l\_MKMNR type ZNAGMKMNR.   ""for select options l is the variable  
  
  select-options s\_MKMNR for l\_MKMNR.""writing select options for variable

BASIC LIST OUTPUT:



SECONDARY LIST OUTPUT:

