**ALV REPORTS**

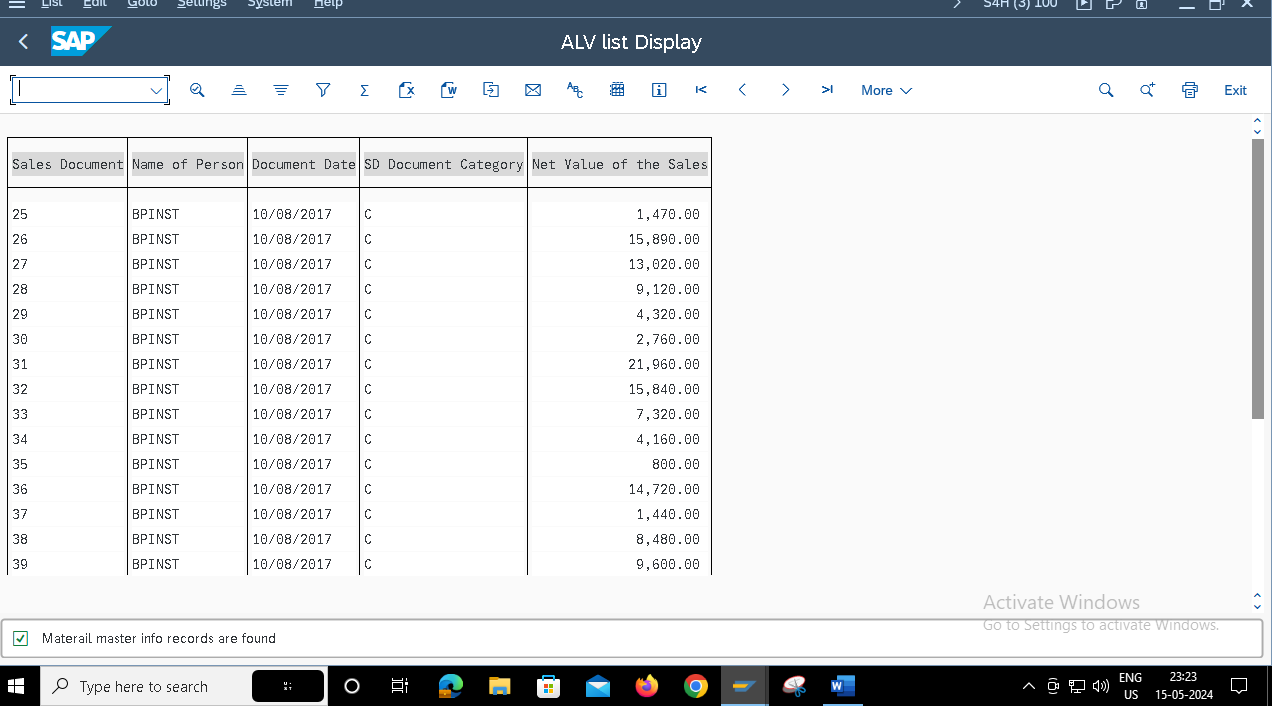
**1.ALV LIST DISPLAY**

\*&---------------------------------------------------------------------\*  
\*& Report ZNAGESH\_ALV\_LIST\_DISPLAY  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNAGESH\_ALV\_LIST\_DISPLAY no standard page heading line-count 20(3)  
line-size 500 message-id ZNAG\_MSG\_SELECT\_S.  
  
INclude ZNAGESH\_ALV\_LIST\_TOP.  
  
select-options s\_vbeln for l\_vbeln obligatory.  
data lv\_vbeln type vbeln. ""variable creation  
  
initialization.  
clear: l\_tab, LT\_FIELDCAT, wa\_FIELDCAT. """clear the previous data to enter new data  
s\_vbeln-low = '0000000025'.  
s\_vbeln-high = '0000000050'.  
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.  
  select vbeln ernam audat vbtyp netwr into table l\_tab from vbak where vbeln in s\_vbeln.  
  
    if sy-subrc eq 0.  
    message s000.  
  else.  
    message e001.  
  endif.  
  
  
  wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'Sales Document'.  
  wa\_fieldcat-fieldname = 'vbeln'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
   wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'Name of Person'.  
  wa\_fieldcat-fieldname = 'ernam'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
   wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'Document Date'.  
  wa\_fieldcat-fieldname = 'audat'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  
   wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'SD Document Category'.  
  wa\_fieldcat-fieldname = 'vbtyp'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
   wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = 'Net Value of the Sales'.  
  wa\_fieldcat-fieldname = 'netwr'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
 End-of-selection.  
  
 CALL FUNCTION 'REUSE\_ALV\_LIST\_DISPLAY'  
  EXPORTING  
\*    I\_INTERFACE\_CHECK              = ' '  
\*    I\_BYPASSING\_BUFFER             =  
\*    I\_BUFFER\_ACTIVE                = ' '  
     I\_CALLBACK\_PROGRAM             = SY-REPID ""or report name  
\*    I\_CALLBACK\_PF\_STATUS\_SET       = ' '  
\*    I\_CALLBACK\_USER\_COMMAND        = ' '  
\*    I\_STRUCTURE\_NAME               =  
\*    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  
\*    IR\_SALV\_LIST\_ADAPTER           =  
\*    IT\_EXCEPT\_QINFO                =  
\*    I\_SUPPRESS\_EMPTY\_DATA          = ABAP\_FALSE  
\*    IO\_SALV\_ADAPTER                =  
\*    O\_COMMON\_HUB                   =  
\*  IMPORTING  
\*    E\_EXIT\_CAUSED\_BY\_CALLER        =  
\*    ES\_EXIT\_CAUSED\_BY\_USER         =  
   TABLES  
     t\_outtab                       = l\_tab  
  EXCEPTIONS  
   PROGRAM\_ERROR                  = 1  
   OTHERS                         = 2  
           .  
 IF sy-subrc <> 0.  
\* Implement suitable error handling here  
 ENDIF.

INCLUDE:

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAGESH\_ALV\_LIST\_TOP  
\*&---------------------------------------------------------------------\*  
types : begin of ls\_1,       ""local structure - types     ""field string - Data  
  VBELN type vbeln,  
  ERNAM type ERNAM,  
  AUDAT type audat,  
  vbtyp type vbtyp,  
  netwr type netwr,  
  End of ls\_1.  
  
  Data : l\_tab type STANDARD TABLE OF ls\_1,  ""i\_tab  
  l\_vbeln type vbeln,  
  
  LT\_FIELDCAT TYPE SLIS\_T\_FIELDCAT\_ALV,  
  WA\_FIELDCAT TYPE SLIS\_FIELDCAT\_ALV.

OUTPUT:



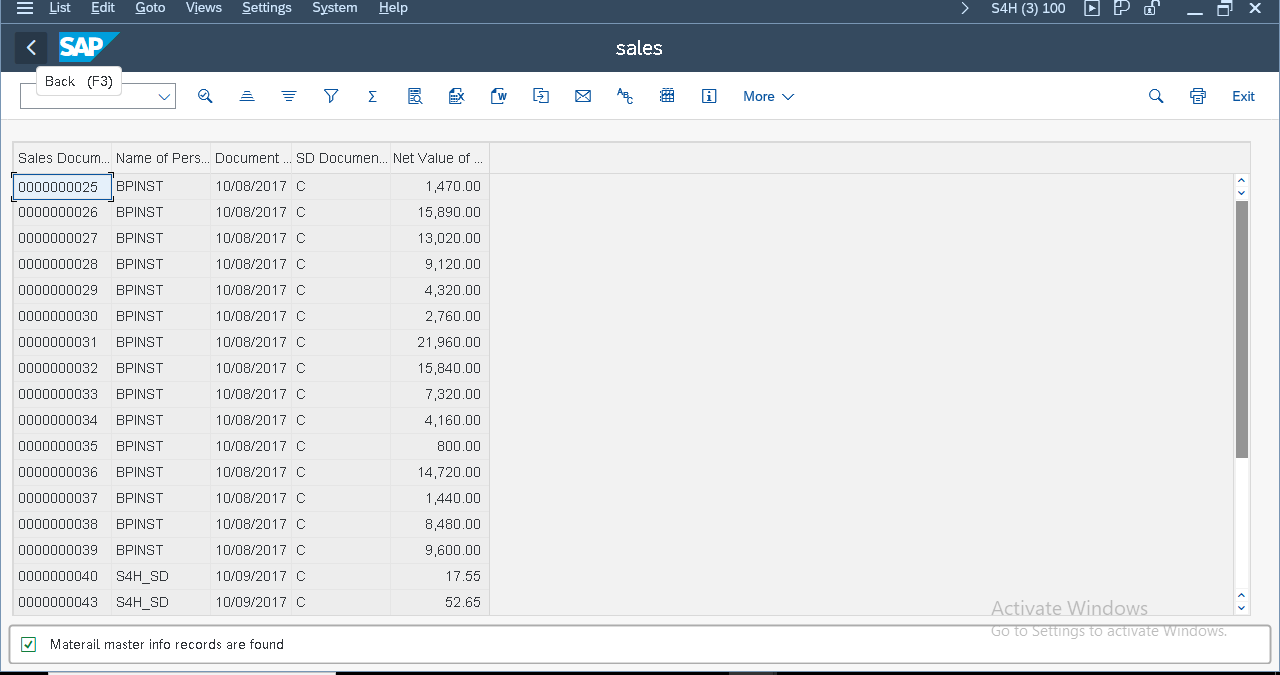
**2.ALV GRID DISPLAY:**

\*&---------------------------------------------------------------------\*  
\*& Report ZNAGESH\_ALV\_GRID\_DISPLAY  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT ZNAGESH\_ALV\_GRID\_DISPLAY no standard page heading line-count 20(3)  
line-size 500 message-id ZNAG\_MSG\_SELECT\_S.  
  
INclude ZNAGESH\_ALV\_LIST\_TOP.  
  
select-options s\_vbeln for l\_vbeln obligatory.  
data lv\_vbeln type vbeln. ""variable creation  
  
initialization.  
clear: l\_tab, LT\_FIELDCAT, wa\_FIELDCAT. """clear the previous data to enter new data  
s\_vbeln-low = '0000000025'.  
s\_vbeln-high = '0000000050'.  
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 Znagesh\_grid\_call.   ""we can write the perform manually and after double click on that it asks to create form and display form.In that we can write the code.  
  perform Znagesh\_grid\_call2.  
  
 End-of-selection.  
 perform Znagesh\_grid\_call3.  
FORM Znagesh\_grid\_call .    ""select the data  
  select vbeln ernam audat vbtyp netwr into table l\_tab from vbak where vbeln in s\_vbeln.  
  
    if sy-subrc eq 0.  
    message s000.  
  else.  
    message e001.  
  endif.  
  
ENDFORM.  
  
FORM Znagesh\_grid\_call2 .   "" giving header details  
   wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'Sales Document'.  
  wa\_fieldcat-fieldname = 'vbeln'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
   wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'Name of Person'.  
  wa\_fieldcat-fieldname = 'ernam'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
   wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'Document Date'.  
  wa\_fieldcat-fieldname = 'audat'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  
   wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'SD Document Category'.  
  wa\_fieldcat-fieldname = 'vbtyp'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
   wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = 'Net Value of the Sales'.  
  wa\_fieldcat-fieldname = 'netwr'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
ENDFORM.  
  
FORM Znagesh\_grid\_call3 .   "" Calling fubction module  
 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            = ' '  
\*    I\_CALLBACK\_HTML\_TOP\_OF\_PAGE       = ' '  
\*    I\_CALLBACK\_HTML\_END\_OF\_LIST       = ' '  
\*    I\_STRUCTURE\_NAME                  =  
\*    I\_BACKGROUND\_ID                   = ' '  
    I\_GRID\_TITLE                       = 'sales'  
\*    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                          = l\_tab  
  EXCEPTIONS  
    PROGRAM\_ERROR                     = 1  
    OTHERS                            = 2  
           .  
 IF sy-subrc <> 0.  
\* Implement suitable error handling here  
 ENDIF.  
  
ENDFORM.

INCLUDE:

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAGESH\_ALV\_LIST\_TOP  
\*&---------------------------------------------------------------------\*  
types : begin of ls\_1,       ""local structure - types     ""field string - Data  
  VBELN type vbeln,  
  ERNAM type ERNAM,  
  AUDAT type audat,  
  vbtyp type vbtyp,  
  netwr type netwr,  
  End of ls\_1.  
  
  Data : l\_tab type STANDARD TABLE OF ls\_1,  ""i\_tab  
  l\_vbeln type vbeln,  
  
  LT\_FIELDCAT TYPE SLIS\_T\_FIELDCAT\_ALV,  
  WA\_FIELDCAT TYPE SLIS\_FIELDCAT\_ALV.

OUTPUT :



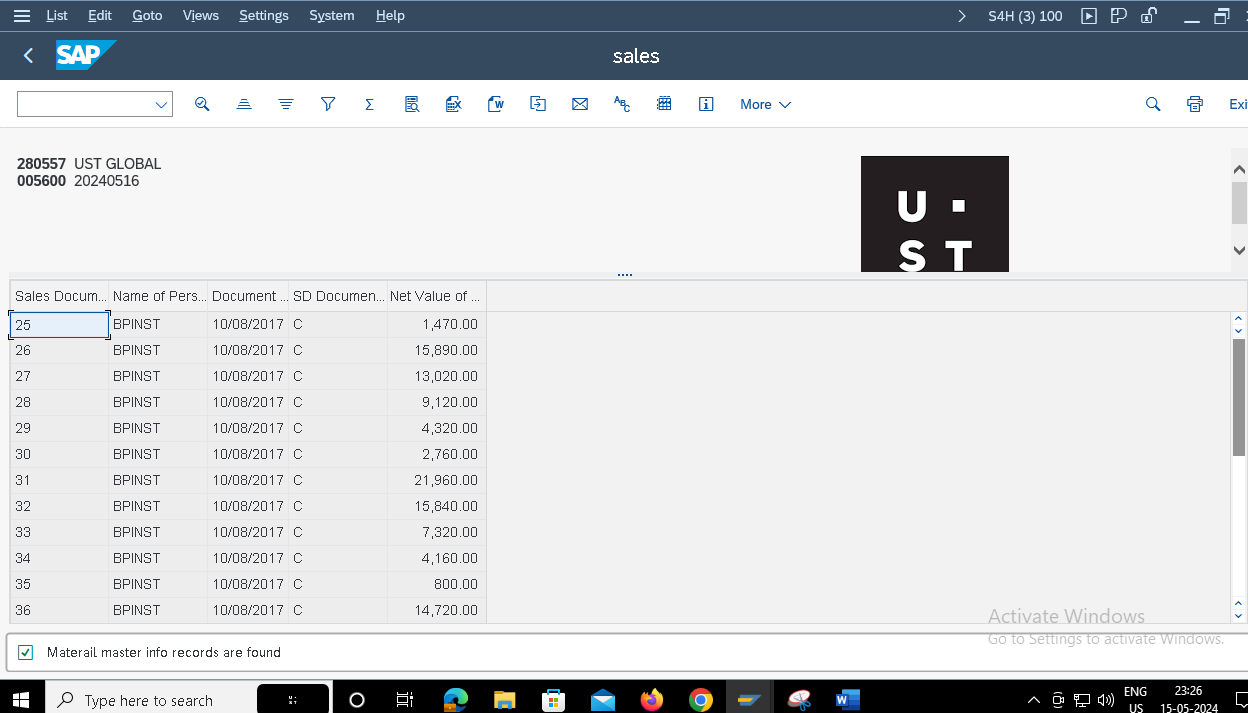
**3.ALV GRID LOGO:**

\*&---------------------------------------------------------------------\*  
\*& Report ZNAGESH\_GRID\_LOGO  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
  
  
REPORT ZNAGESH\_GRID\_LOGO no standard page heading line-count 20(3)  
line-size 500 message-id ZNAG\_MSG\_SELECT\_S.  
  
INCLUDE ZNAGESH\_ALV\_LOGO\_TOP.  
  
select-options s\_vbeln for l\_vbeln obligatory.  
data lv\_vbeln type vbeln. ""variable creation  
  
initialization.  
clear: l\_tab, LT\_FIELDCAT, wa\_FIELDCAT. """clear the previous data to enter new data  
s\_vbeln-low = '0000000025'.  
s\_vbeln-high = '0000000050'.  
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 Znagesh\_grid\_call.   ""we can write the perform manually and after double click on that it asks to create form and display form.In that we can write the code.  
  perform Znagesh\_grid\_call2.  
  
 End-of-selection.  
 perform Znagesh\_grid\_call3.  
FORM Znagesh\_grid\_call .    ""select the data  
  select vbeln ernam audat vbtyp netwr into table l\_tab from vbak where vbeln in s\_vbeln.  
  
    if sy-subrc eq 0.  
    message s000.  
  else.  
    message e001.  
  endif.  
  
ENDFORM.  
  
FORM Znagesh\_grid\_call2 .   "" giving header details  
   wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'Sales Document'.  
  wa\_fieldcat-fieldname = 'vbeln'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  wa\_fieldcat-no\_zero = 'X'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
   wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'Name of Person'.  
  wa\_fieldcat-fieldname = 'ernam'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
   wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'Document Date'.  
  wa\_fieldcat-fieldname = 'audat'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  
   wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'SD Document Category'.  
  wa\_fieldcat-fieldname = 'vbtyp'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
   wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = 'Net Value of the Sales'.  
  wa\_fieldcat-fieldname = 'netwr'.  
  wa\_fieldcat-tabname  = 'l\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
ENDFORM.  
  
FORM Znagesh\_grid\_call3 .   "" Calling fubction module  
 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                       ='sales'  
\*    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                          = l\_tab  
  EXCEPTIONS  
    PROGRAM\_ERROR                     = 1  
    OTHERS                            = 2  
           .  
 IF sy-subrc <> 0.  
\* Implement suitable error handling here  
 ENDIF.  
  
ENDFORM.  
form ust\_logo.   " SUBROUTINES  
  data : lt\_header type slis\_t\_listheader,  
         wa\_header type slis\_listheader.  
  
\*  wa\_header-typ = 'H'.  
  wa\_header-typ = 'S'.  
\*  wa\_header-typ = 'A'.  
  
  wa\_header-info = 'UST GLOBAL'.  
  wa\_header-key = '280557'.  
  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             = 'ZNAGESH\_UST\_LOGO'   " OAER OBJECT KEY NAME  
\*     I\_END\_OF\_LIST\_GRID =  
\*     I\_ALV\_FORM         =  
    .  
endform.

INCLUDE:

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAGESH\_ALV\_LOGO\_TOP  
\*&---------------------------------------------------------------------\*  
types : begin of ls\_1,       ""local structure - types     ""field string - Data  
  VBELN type vbeln,  
  ERNAM type ERNAM,  
  AUDAT type audat,  
  vbtyp type vbtyp,  
  netwr type netwr,  
  End of ls\_1.  
  
  Data : l\_tab type STANDARD TABLE OF ls\_1,  ""i\_tab  
  l\_vbeln type vbeln,  
  LT\_FIELDCAT TYPE SLIS\_T\_FIELDCAT\_ALV,  
  WA\_FIELDCAT TYPE SLIS\_FIELDCAT\_ALV,  
  
  TY\_layout type slis\_Layout\_alv.  
  
  CONSTANTS : CS\_TAB(10) TYPE C VALUE 'l\_tab',  
              CS\_VBELN(10) TYPE C VALUE 'VBELN',  
              CS\_ERNAM(10) TYPE C VALUE 'ERNAM'.

OUTPUT:



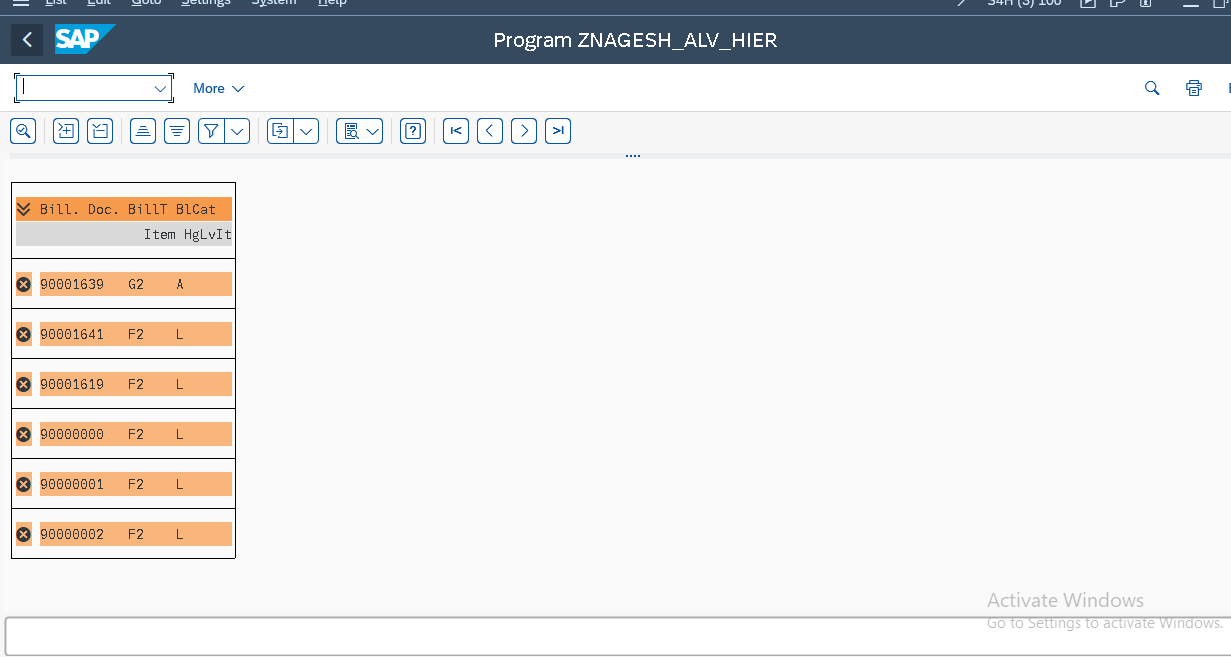
**4. ALV\_HIER**

\*&---------------------------------------------------------------------\*  
\*& Report ZNAGESH\_ALV\_HIER  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNAGESH\_ALV\_HIER NO STANDARD PAGE HEADING line-count 23(3) line-size 300.  
Include ZNAG\_HIER\_INC.  
TABLEs VBRK.  
  
  SELECT-OPTIONS S\_VBELN FOR VBRK-VBELN.  
  
START-OF-SELECTION.  
  
PERFORM GET\_DATA.  
  
PERFORM CREATE\_FIELDCAT.  
  
PERFORM HIER\_DISPLAY.  
  
FORM get\_data.  
  
  SELECT VBELN FKART FKTYP FROM VBRK  " TABLE 1  
    INTO CORRESPONDING FIELDS OF TABLE IT\_TAB1  
    WHERE VBELN IN  
    S\_VBELN.  
  
  
  SELECT   VBELN  POSNR  UEPOS  
    FROM VBRP  " TABLE 2  
    INTO  TABLE IT\_TAB2  
    WHERE VBELN IN S\_VBELN.  
  
ENDFORM.  
  
FORM create\_fieldcat .  
  
  WA\_FIELDCAT-col\_pos = '1'.  " COLUM POSITION  
  WA\_FIELDCAT-fieldname = 'VBELN'. " COLUMN NAME  
  WA\_FIELDCAT-tabname = 'IT\_TAB1'. " INTERNAL TABLE  
  WA\_FIELDCAT-ref\_tabname = 'VBRK'. " TABLE NAME  
  WA\_FIELDCAT-ref\_fieldname = 'VBELN'. "REFERENCE FILED  
  APPEND WA\_FIELDCAT TO IT\_FIELDCAT.  
  CLEAR WA\_FIELDCAT.  
  
  
   WA\_FIELDCAT-col\_pos = '2'.  " COLUM POSITION  
  WA\_FIELDCAT-fieldname = 'FKART'. " COLUMN NAME  
  WA\_FIELDCAT-tabname = 'IT\_TAB1'. " INTERNAL TABLE  
  WA\_FIELDCAT-ref\_tabname = 'VBRK'. " TABLE NAME  
  WA\_FIELDCAT-ref\_fieldname = 'FKART'. "REFERENCE FILED  
  APPEND WA\_FIELDCAT TO IT\_FIELDCAT.  
  CLEAR WA\_FIELDCAT.  
  
  
   WA\_FIELDCAT-col\_pos = '3'.  " COLUM POSITION  
  WA\_FIELDCAT-fieldname = 'FKTYP'. " COLUMN NAME  
  WA\_FIELDCAT-tabname = 'IT\_TAB1'. " INTERNAL TABLE  
  WA\_FIELDCAT-ref\_tabname = 'VBRK'. " TABLE NAME  
  WA\_FIELDCAT-ref\_fieldname = 'FKTYP'. "REFERENCE FILED  
  APPEND WA\_FIELDCAT TO IT\_FIELDCAT.  
  CLEAR WA\_FIELDCAT.  
  
  
  
   WA\_FIELDCAT-col\_pos = '1'.  " COLUM POSITION  
  WA\_FIELDCAT-fieldname = 'VBELN'. " COLUMN NAME  
  WA\_FIELDCAT-tabname = 'IT\_TAB2'. " INTERNAL TABLE  
  WA\_FIELDCAT-ref\_tabname = 'VBRP'. " TABLE NAME  
  WA\_FIELDCAT-ref\_fieldname = 'VEBLN'. "REFERENCE FILED  
  APPEND WA\_FIELDCAT TO IT\_FIELDCAT.  
  CLEAR WA\_FIELDCAT.  
  
   WA\_FIELDCAT-col\_pos = '2'.  " COLUM POSITION  
  WA\_FIELDCAT-fieldname = 'POSNR'. " COLUMN NAME  
  WA\_FIELDCAT-tabname = 'IT\_TAB2'. " INTERNAL TABLE  
  WA\_FIELDCAT-ref\_tabname = 'VBRP'. " TABLE NAME  
  WA\_FIELDCAT-ref\_fieldname = 'POSNR'. "REFERENCE FILED  
  APPEND WA\_FIELDCAT TO IT\_FIELDCAT.  
  CLEAR WA\_FIELDCAT.  
  
  
   WA\_FIELDCAT-col\_pos = '3'.  " COLUM POSITION  
  WA\_FIELDCAT-fieldname = 'UEPOS'. " COLUMN NAME  
  WA\_FIELDCAT-tabname = 'IT\_TAB2'. " INTERNAL TABLE  
  WA\_FIELDCAT-ref\_tabname = 'VBRP'. " TABLE NAME  
  WA\_FIELDCAT-ref\_fieldname = 'UEPOS'. "REFERENCE FILED  
  APPEND WA\_FIELDCAT TO IT\_FIELDCAT.  
  CLEAR WA\_FIELDCAT.  
  
  
  
ENDFORM.  
  
FORM hier\_display .  
  
  
  IT\_KEY-header01 = 'VBELN'.  
  LAYOUT-EXPAND\_FIELDNAME = 'UST'.  
  
  IF SY-SUBRC EQ 0.  
CALL FUNCTION 'REUSE\_ALV\_HIERSEQ\_LIST\_DISPLAY'  
  EXPORTING  
\*   I\_INTERFACE\_CHECK              = ' '  
   I\_CALLBACK\_PROGRAM             = SY-REPID  
\*   I\_CALLBACK\_PF\_STATUS\_SET       = ' '  
\*   I\_CALLBACK\_USER\_COMMAND        = ' '  
  IS\_LAYOUT                      = LAYOUT  
   IT\_FIELDCAT                    = IT\_FIELDCAT[]  
\*   IT\_EXCLUDING                   =  
\*   IT\_SPECIAL\_GROUPS              =  
\*   IT\_SORT                        =  
\*   IT\_FILTER                      =  
\*   IS\_SEL\_HIDE                    =  
\*   I\_SCREEN\_START\_COLUMN          = 0  
\*   I\_SCREEN\_START\_LINE            = 0  
\*   I\_SCREEN\_END\_COLUMN            = 0  
\*   I\_SCREEN\_END\_LINE              = 0  
\*  I\_DEFAULT                      = X  
\*   I\_SAVE                         = ' '  
\*   IS\_VARIANT                     =  
\*   IT\_EVENTS                      =  
\*   IT\_EVENT\_EXIT                  =  
    i\_tabname\_header               = 'IT\_TAB1'  
    i\_tabname\_item                 = 'IT\_TAB2'  
\*   I\_STRUCTURE\_NAME\_HEADER        =  
\*   I\_STRUCTURE\_NAME\_ITEM          =  
    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\_TAB1 "table 1  
    t\_outtab\_item                  = IT\_TAB2  ""table 2  
 EXCEPTIONS  
   PROGRAM\_ERROR                  = 1  
   OTHERS                         = 2  
          .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
ENDIF.  
  
ENDFORM..

INCLUDE :

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAG\_HIER\_INC  
\*&---------------------------------------------------------------------\*  
TYPES: Begin of ls\_tab1, "Structure for table 1 vbak  
       UST(1),  
       VBELN type VBELN\_VF,  
       FKART type FKART,  
        FKTYP type FKTYP,  
        END of ls\_tab1,  
  
  BEGIN of ls\_tab2,  ""Structure for table 2 VBRK  
    VBELN TYPE VBELN\_VF,  
    POSNR TYPE POSNR\_VF,  
    UEPOS TYPE UEPOS,  
    end of ls\_tab2.  
  
  
    data: IT\_TAB1 TYPE Table of ls\_tab1, ""internal table for table 1 VBAP  
          WA\_tab1 type ls\_tab1,        ""Work area table for table 1 VBAP  
  
          IT\_FIELDCAT type SLIS\_T\_FIELDCAT\_ALV, ""Field catalog for internal table  
          WA\_FIELDCAT like line of IT\_FIELDCAT, ""FIEld catalog for work area  
  
          IT\_KEY type SLIS\_KEYINFO\_ALV, ""Key Element for  SLIS\_Layout  
          LAYOUT TYPE SLIS\_LAYOUT\_ALV,  ""Layout for layout field  
          LV\_VBELN TYPE VBELN\_VF,    ""Data type for taking range value. "Local data type to store range in selectuon screen  
  
          IT\_tab2 type table of ls\_tab2,""internal table for table 2 VBRK  
          wa\_tab2 type ls\_tab2.    ""WEork area table for table 2 VBRK

OUTPUT:



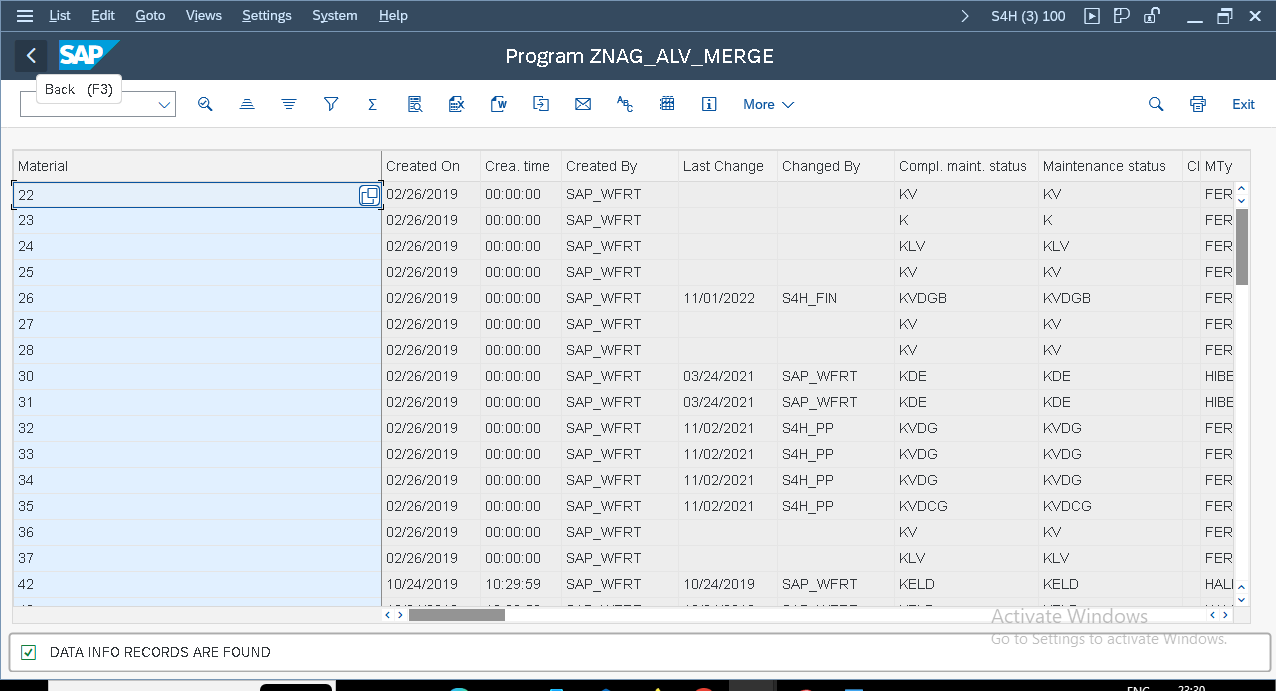
**4.GRID MEGRGE**

\*&---------------------------------------------------------------------\*  
\*& Report ZNAG\_ALV\_MERGE  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNAG\_ALV\_MERGE no standard page heading line-count 20(3)  
line-size 500 message-id ZNAGESH\_CLASS\_R .  
  
include Znag\_ALV\_merge\_top.  
  
select-options s\_matnr for lv\_matnr obligatory.  
  
INITIALIZATION.  
clear: lt\_Tab, wa\_fieldcat, lt\_fieldcat.  
  s\_matnr-low  = '1'.  
  s\_matnr-high  = '100'.  
  append s\_matnr.  
  
  at selection-screen on s\_matnr.   " 1 to 100  
  
  select single matnr  
    into lv\_matnr  
     from mara  
     where matnr in s\_matnr.""matnr value kept in s\_matnr in screen selection  
  
    if sy-subrc eq 0.  
    message s000.  " se91 - tcode.  
  else.  
    message e001.  " se91 - tcode.  
  endif.  
  
  START-OF-SELECTION.  
   perform get\_mat\_data.  
   perform get\_alv\_fcat.  
   end-OF-SELECTION.  
  
   perform display\_mat\_alvout.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_mat\_data  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM get\_mat\_data .  
  select \*  ""we can use \*  in only merge to get all fields of data.  
      into table lt\_Tab  
      from mara  
      where matnr in  s\_matnr.  
  
  if sy-subrc eq 0.  
    message s000.  
  else.  
    message e001.  
  endif.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_alv\_fcat  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM get\_alv\_fcat .  
CALL FUNCTION 'REUSE\_ALV\_FIELDCATALOG\_MERGE'  
 EXPORTING  
   I\_PROGRAM\_NAME               = sy-repid  
 I\_INTERNAL\_TABNAME           = 'lt\_tab'  
  I\_STRUCTURE\_NAME             = 'mara'  
\*   I\_CLIENT\_NEVER\_DISPLAY       = 'X'  
\*   I\_INCLNAME                   =  
\*   I\_BYPASSING\_BUFFER           =  
\*   I\_BUFFER\_ACTIVE              =  
  CHANGING  
    ct\_fieldcat                  = lt\_fieldcat  
 EXCEPTIONS  
\*   INCONSISTENT\_INTERFACE       = 1  
  PROGRAM\_ERROR                = 2  
   OTHERS                       = 3  
          .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form display\_mat\_alvout  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM display\_mat\_alvout .  
  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            = ' '  
\*     I\_CALLBACK\_HTML\_TOP\_OF\_PAGE       = ' '  
\*     I\_CALLBACK\_HTML\_END\_OF\_LIST       = ' '  
\*     I\_STRUCTURE\_NAME                  =  
\*     I\_BACKGROUND\_ID                   = ' '  
\*     I\_GRID\_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.

INCLUDE:

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAG\_ALV\_MERGE\_TOP  
\*&---------------------------------------------------------------------\*  
  
 data : lt\_tab type standard table of MARA,  " itab  
           lv\_matnr type matnr,  
           WA\_FIELDCAT TYPE SLIS\_FIELDCAT\_ALV,  " ALV FCAT FOR WA  
           LT\_FIELDCAT TYPE SLIS\_T\_FIELDCAT\_ALV,  
           TY\_LAYOUT TYPE SLIS\_LAYOUT\_ALV.    """layout

OUTPUT:



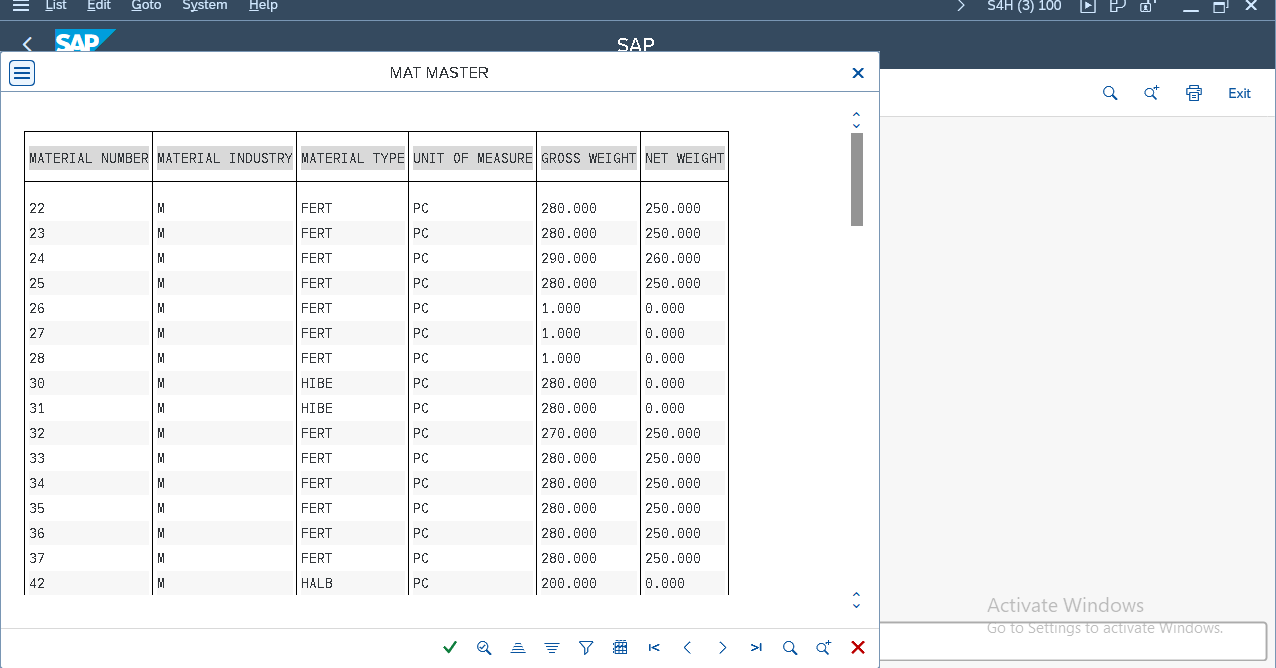
**5.GRID POPUP:**

\*&---------------------------------------------------------------------\*  
\*& Report ZNAG\_ALV\_GRID\_POPUP  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNAG\_ALV\_GRID\_POPUP NO STANDARD PAGE HEADING LINE-COUNT 20(3)  
LINE-SIZE 500 MESSAGE-ID znag\_msg\_select\_s.  
  
INCLUDE Znag\_alv\_popup.  
  
select-options s\_matnr for lv\_matnr obligatory.  
  
initialization.  
  clear: lt\_Tab, wa\_fieldcat, lt\_fieldcat.  
  s\_matnr-low  = '1'.  
  s\_matnr-high  = '100'.  
  append s\_matnr.  
  
at selection-screen on s\_matnr.   " 1 to 100  
  
  select single 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.  
  
start-of-selection.  
  
  perform get\_alv\_fcat.  
  perform get\_mat\_data.  
  
end-of-selection.  
  perform disp\_mat\_alvout.  
  
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 GET\_ALV\_FCAT  
\*&---------------------------------------------------------------------\*  
form get\_alv\_fcat .  
\*\*\*\*\*  MATNR  
\*\* LV\_COUNT = LV\_COUNT + 1.  
  wa\_fieldcat-col\_pos =  '1'.  
  wa\_fieldcat-seltext\_l = text-001. "   'MAT NUMBER'.  
  wa\_fieldcat-fieldname = cs\_matnr. "  'MATNR'.  
  wa\_fieldcat-tabname  = cs\_tab .  "   'LT\_TAB'.  
\*WA\_FIELDCAT-emphasize = 'C101'.  
  wa\_fieldcat-no\_zero = 'X'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
\*\*\*\*\*  MBRSH  
  wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = text-002 ."  'MAT IND SECTOR'.  
  wa\_fieldcat-fieldname = cs\_mbrsh . "  'MBRSH'.  
  wa\_fieldcat-tabname  = cs\_tab .  " 'LT\_TAB'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
\*\*\*\*\*  MTART  
  wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = text-003 . "'MAT TYPE'.  
  wa\_fieldcat-fieldname = 'MTART'.  
  wa\_fieldcat-tabname  = cs\_tab."   'LT\_TAB'.  
\*WA\_FIELDCAT-emphasize = 'C200'.  
  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  
\*\*\*\*\*  MEINS  
  wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = text-004 ."'MAT BASE UOM'.  
  wa\_fieldcat-fieldname = 'MEINS'.  
  wa\_fieldcat-tabname  = 'LT\_TAB'.  
\*WA\_FIELDCAT-emphasize = 'C611'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  
  
\*\*\*\*\*  BRGEW  
  wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = text-005. "'MAT GROSS WEIGHT'.  
  wa\_fieldcat-fieldname = 'BRGEW'.  
  wa\_fieldcat-tabname  = cs\_tab . "  'LT\_TAB'.  
\*WA\_FIELDCAT-emphasize = 'C711'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
\*\*\*\*\*  NTGEW  
  wa\_fieldcat-col\_pos = '6'.  
  wa\_fieldcat-seltext\_s = text-006 ."  'MAT NET WEIGHT'.  
  
  wa\_fieldcat-fieldname = 'NTGEW'.  
  wa\_fieldcat-tabname  = cs\_tab ."   'LT\_TAB'.  
\*WA\_FIELDCAT-emphasize = 'C411'.   ""for highlight the data with colours  
  wa\_fieldcat-do\_sum = 'X'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
endform.  
\*&---------------------------------------------------------------------\*  
\*& Form DISP\_MAT\_ALVOUT  
\*&---------------------------------------------------------------------\*  
\*& text  
form disp\_mat\_alvout .  
  
  
call function 'REUSE\_ALV\_POPUP\_TO\_SELECT'  
  exporting  
   I\_TITLE                       = 'MAT MASTER'  
\*   I\_SELECTION                   = 'X'  
\*   I\_ALLOW\_NO\_SELECTION          =  
   I\_ZEBRA                       = 'X'  
\*   I\_SCREEN\_START\_COLUMN         = 0  
\*   I\_SCREEN\_START\_LINE           = 0  
\*   I\_SCREEN\_END\_COLUMN           = 0  
\*   I\_SCREEN\_END\_LINE             = 0  
\*   I\_CHECKBOX\_FIELDNAME          =  
\*   I\_LINEMARK\_FIELDNAME          =  
\*   I\_SCROLL\_TO\_SEL\_LINE          = 'X'  
    i\_tabname                     =  'LT\_TAB'  
   I\_STRUCTURE\_NAME              = 'LS\_TAB'  
   IT\_FIELDCAT                   = LT\_FIELDCAT  
\*   IT\_EXCLUDING                  =  
   I\_CALLBACK\_PROGRAM            =  SY-REPID  
\*   I\_CALLBACK\_USER\_COMMAND       =  
\*   IS\_PRIVATE                    =  
\* IMPORTING  
\*   ES\_SELFIELD                   =  
\*   E\_EXIT                        =  
  tables  
    t\_outtab                      = LT\_TAB  
\* EXCEPTIONS  
\*   PROGRAM\_ERROR                 = 1  
\*   OTHERS                        = 2  
          .  
if sy-subrc <> 0.  
\* Implement suitable error handling here  
endif.  
  
  
  
  
  
  
endform.

INCLUDE:

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAG\_ALV\_POPUP  
\*&---------------------------------------------------------------------\*  
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.  
  
    data : lt\_tab type standard table of ls\_tab,  " itab  
\*\*\*\*\*\*           wa\_tab type ls\_tab,   " work area  
           lv\_matnr type matnr,  
          WA\_FIELDCAT TYPE SLIS\_FIELDCAT\_ALV,  " ALV FCAT FOR WA  
           LT\_FIELDCAT TYPE SLIS\_T\_FIELDCAT\_ALV,  
  
           TY\_LAYOUT TYPE SLIS\_LAYOUT\_ALV.  
  
          CONSTANTS : CS\_TAB(10) TYPE C VALUE 'LT\_TAB',   "to replace the hardcoded constraints for better code performance"  
                      CS\_MATNR(10) TYPE C VALUE 'MATNR',  
                      CS\_MBRSH(10) TYPE C VALUE 'MBRSH'.

OUTPUT:



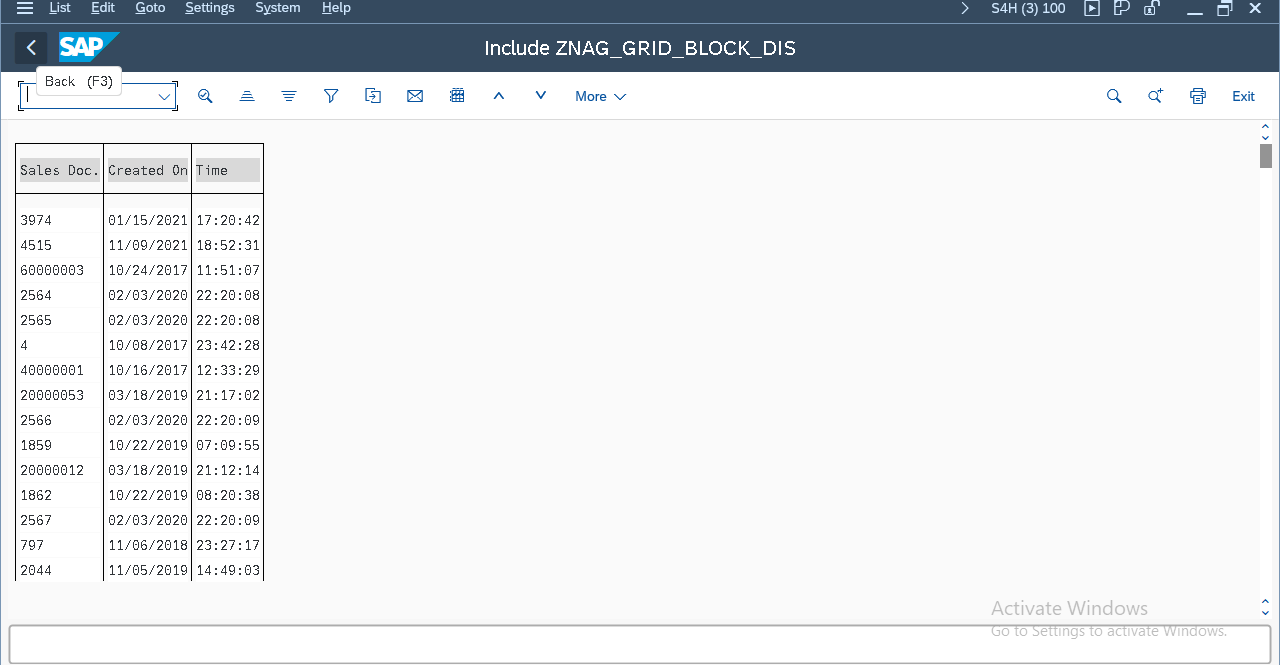
**6.GRID BLOCK**

\*&---------------------------------------------------------------------\*  
\*& Report ZNAG\_GRID\_BLOCK\_DIS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT znag\_grid\_block\_dis.  
  
INCLUDE znagesh\_alv\_block\_\_top.  
  
SELECT-OPTIONS s\_vbeln FOR lv\_vbeln. ""variable creation  vbeln\_VA data element name  
  
START-OF-SELECTION.  
  
  PERFORM Znagesh\_get\_data.   ""we can write the perform manually and after double click on that it asks to create form and display form.In that we can write the code.  
  
  PERFORM Znag\_fcat\_vbak.  
  PERFORM Znag\_fact\_vbap.  
  
  perform block\_list\_initial.  
  perform block\_list\_append\_vbak.  
  perform block\_list\_append\_vbap.  
  perform block\_list\_display.  
  
  
  
  
  
  
\*  
\*&---------------------------------------------------------------------\*  
\*& Form Znagesh\_get\_data  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM Znagesh\_get\_data .  
  SELECT vbeln  
         erdat  
          erzet  
           FROM vbak  
  INTO TABLE it\_vbak  WHERE vbeln IN s\_vbeln.  
  
  IF NOT it\_vbak[] IS INITIAL .  
  
    SELECT  vbeln pOsnr matnr  
    FROM vbap  
      INTO TABLE it\_vbap "get vbap data  
  
    "for all entries in it\_vbak  
       WHERE vbeln IN s\_vbeln.  
  
  ENDIF.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form Znag\_fcat\_vbap  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM Znag\_fcat\_vbak .  
  
  w\_fcat\_vbak-col\_pos       = '1'. "coloum position  
  w\_fcat\_vbak-fieldname     = 'VBELN'. "column name  
  w\_fcat\_vbak-ref\_tabname   = 'VBAK'. "table  VBAK  
  w\_fcat\_vbak-ref\_fieldname = 'VBELN'. "reference field, it will show descriptions automatically  
  APPEND w\_fcat\_vbak TO t\_fcat\_vbak.  
  CLEAR w\_fcat\_vbak.  
  
  w\_fcat\_vbak-col\_pos       = '2'. "coloum position  
  w\_fcat\_vbak-fieldname     = 'ERDAT'. "column name  
  w\_fcat\_vbak-ref\_tabname   = 'VBAK'. "table  VBAK  
  w\_fcat\_vbak-ref\_fieldname = 'ERDAT'. "reference field, it will show descriptions automatically  
  APPEND w\_fcat\_VBAK TO t\_fcat\_VBAK.  
  CLEAR w\_fcat\_VBAK.  
  
  w\_fcat\_vbak-col\_pos       = '3'. "coloum position  
  w\_fcat\_vbak-fieldname     = 'ERZET'. "column name  
  w\_fcat\_vbak-ref\_tabname   = 'VBAK'. "table  VBAK  
  w\_fcat\_vbak-ref\_fieldname = 'ERZET'. "reference field, it will show descriptions automatically  
  APPEND w\_fcat\_VBAK TO t\_fcat\_VBAK.  
  CLEAR w\_fcat\_VBAK.  
  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form Znag\_fact\_vbap  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM Znag\_fact\_vbap .  
  
  w\_fcat\_vbap-col\_pos       = '1'. "coloum position  
  w\_fcat\_vbap-fieldname     = 'VBELN'. "column name  
  w\_fcat\_vbap-ref\_tabname   = 'VBAP'. "table  VBAP  
  w\_fcat\_vbap-ref\_fieldname = 'VBELN'. "reference field, it will show descriptions automatically  
  APPEND w\_fcat\_vbap TO t\_fcat\_vbap.  
  CLEAR w\_fcat\_vbap.  
  
  w\_fcat\_vbap-col\_pos       = '2'. "coloum position  
  w\_fcat\_vbap-fieldname     = 'POSNR'. "column name  
  w\_fcat\_vbap-ref\_tabname   = 'VBAP'. "table  VBAP  
  w\_fcat\_vbap-ref\_fieldname = 'POSNR'. "reference field, it will show descriptions automatically  
  APPEND w\_fcat\_vbap TO t\_fcat\_vbap.  
  CLEAR w\_fcat\_vbap.  
  
  w\_fcat\_vbap-col\_pos       = '3'. "coloum position  
  w\_fcat\_vbap-fieldname     = 'MATNR'. "column name  
  w\_fcat\_vbap-ref\_tabname   = 'VBAP'. "table  VBAP  
  w\_fcat\_vbap-ref\_fieldname = 'MATNR'. "reference field, it will show descriptions automatically  
  APPEND w\_fcat\_vbap TO t\_fcat\_vbap.  
  CLEAR w\_fcat\_vbap.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form block\_list\_initial  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM block\_list\_initial .  
  call function 'REUSE\_ALV\_BLOCK\_LIST\_INIT' "initialize Block List ALV  
    exporting  
      i\_callback\_program = sy-repid.  
  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form block\_list\_append\_vbak  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM block\_list\_append\_vbak .  
  call function 'REUSE\_ALV\_BLOCK\_LIST\_APPEND' "append ALV lists  
    exporting  
      is\_layout   = s\_layout "set layout  
      it\_fieldcat = t\_fcat\_vbak "set field catalog  
      i\_tabname   = 'IT\_VBAK' "table  
      it\_events   = s\_events "events  
    tables  
      t\_outtab    = it\_VBAK. "out put table  
  
  if sy-subrc <> 0.  
\* Implement suitable error handling here  
  endif.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form block\_list\_append\_vbap  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM block\_list\_append\_vbap .  
  call function 'REUSE\_ALV\_BLOCK\_LIST\_APPEND' "append ALV lists  
    exporting  
      is\_layout   = s\_layout "set layout  
      it\_fieldcat = t\_fcat\_VBAP "set field catalog  
      i\_tabname   = 'IT\_VBAP' "table  
      it\_events   = s\_events "events  
    tables  
      t\_outtab    = it\_VBAP. "out put table  
  
  if sy-subrc <> 0.  
\* Implement suitable error handling here  
  endif.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form block\_list\_display  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM block\_list\_display .  
   call function 'REUSE\_ALV\_BLOCK\_LIST\_DISPLAY'. "display blocked list  
  
  
ENDFORM.

INCLUDE:

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAGESH\_ALV\_BLOCK\_\_TOP  
\*&---------------------------------------------------------------------\*  
TYPES : BEGIN OF ls\_vbak,  "User defined TYPES     VBAK TABLE  
      VBELN    TYPE VBELN\_VA,  
      ERDAT    TYPE ERDAT,  
      ERZET TYPE ERZET,  
  
END OF ls\_vbak.  
  
\*-------------------------------------  
TYPES : BEGIN OF ls\_vbap,       ""VBAP TABLE  
         vbeln TYPE vbeln\_va,  
         posnr TYPE posnr,  
         matnr TYPE matnr,  
END OF ls\_vbap.  
  
DATA : it\_vbak TYPE TABLE OF ls\_vbak, "vbak internal table  
      wa\_vbak TYPE ls\_vbak, "work area  
      lv\_vbeln type vbeln,  
  
     it\_vbap TYPE TABLE OF ls\_vbap, "vbap internal table  
      wa\_vbap TYPE ls\_vbap,  
  
     t\_fcat\_VBAK TYPE slis\_t\_fieldcat\_alv, "field catalog for VBAK table  
     w\_fcat\_VBAK type slis\_fieldcat\_alv,  
  
  
     t\_fcat\_vbap TYPE slis\_t\_fieldcat\_alv, " vbap field catalog  
     w\_fcat\_vbap type slis\_fieldcat\_alv,  
  
     s\_events TYPE slis\_t\_event,  
  
    s\_layout TYPE slis\_layout\_alv.

OUTPUT:



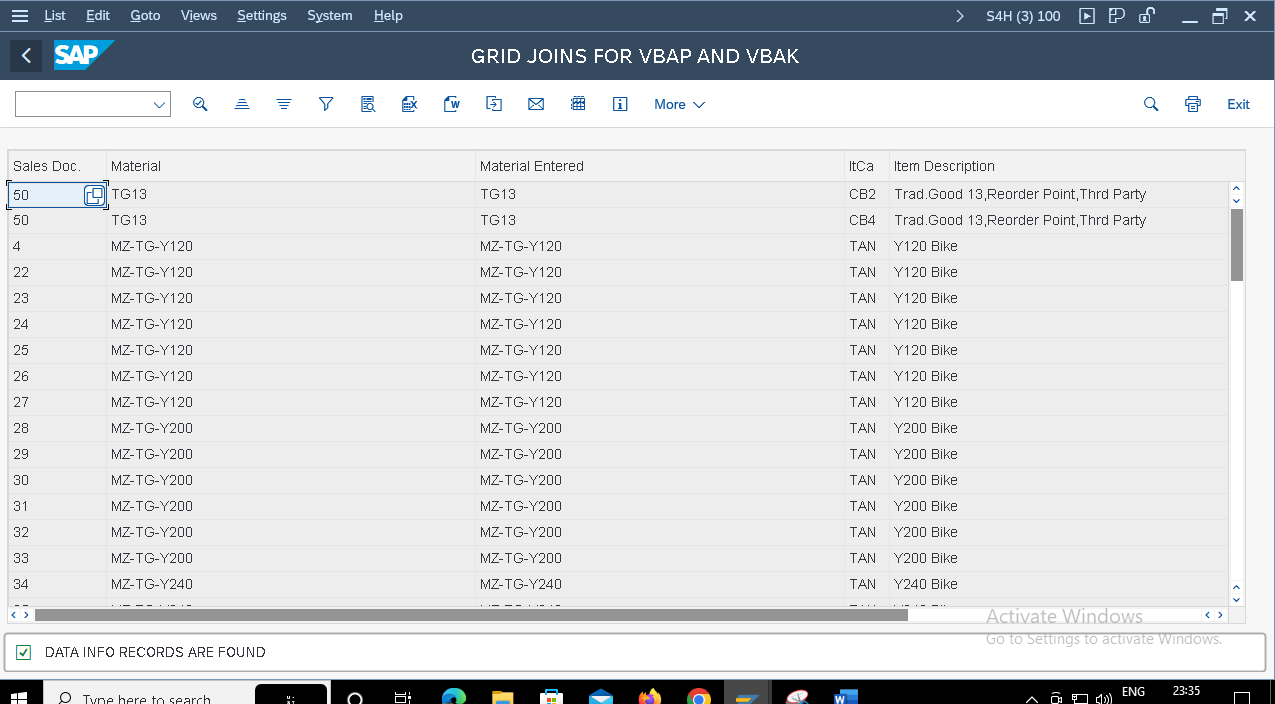
**7.GRID JOINS**

\*&---------------------------------------------------------------------\*  
\*& Report ZNAG\_GRID\_DIS\_JOINS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNAG\_GRID\_DIS\_JOINS no standard page heading line-count 20(3)  
line-size 500 message-id ZNAGESH\_CLASS\_R.  
  
Include Znag\_Grid\_joins.  
  
SELECT-OPTIONS S\_VBELN FOR LV\_VBELN OBLIGATORY.  
  
INITIALIZATION.  
CLEAR: LT\_FINAL, WA\_FIELDCAT, LT\_FIELDCAT.  
S\_VBELN-LOW = '1'.  
S\_VBELN-HIGH = '100'.  
APPEND S\_VBELN.  
  
AT SELECTION-SCREEN ON S\_VBELN.  
  
  SELECT SINGLE VBELN  
    INTO LV\_VBELN  
    FROM VBAP  
    WHERE VBELN IN S\_VBELN.  
  
    IF SY-SUBRC EQ 0.  
      MESSAGE S000.  
    ELSE.  
      MESSAGE E001.  
    ENDIF.  
  
  START-OF-SELECTION.  
  
  PERFORM GET\_GRID\_JOIN\_DATA.  
  
    PERFORM GET\_FIELDS USING TEXT-003 TEXT-001 TEXT-012.  
    PERFORM GET\_FIELDS USING TEXT-004 TEXT-001 TEXT-012.  
    PERFORM GET\_FIELDS USING TEXT-005 TEXT-001 TEXT-012.  
    PERFORM GET\_FIELDS USING TEXT-006 TEXT-001 TEXT-012.  
    PERFORM GET\_FIELDS USING TEXT-007 TEXT-001 TEXT-012.  
  
    PERFORM GET\_FIELDS USING TEXT-003 TEXT-002 TEXT-013.  
    PERFORM GET\_FIELDS USING TEXT-008 TEXT-002 TEXT-013.  
    PERFORM GET\_FIELDS USING TEXT-011 TEXT-002 TEXT-013.  
  
    END-OF-SELECTION.  
    PERFORM DISP\_GRID\_JOINS\_DATA.  
  
\*&---------------------------------------------------------------------\*  
\*& Form GET\_GRID\_JOIN\_DATA  
\*&---------------------------------------------------------------------\*  
  
FORM get\_grid\_join\_data .  
  
SELECT  
  VBAP~VBELN MATNR MATWA PSTYV ARKTX  
  VBAK~ERDAT ANGDT  
  INTO TABLE LT\_FINAL  
  FROM VBAP  
  INNER JOIN VBAK  
  ON VBAP~VBELN EQ VBAK~VBELN  
  WHERE VBAP~VBELN IN S\_VBELN.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form GET\_FIELDS  
  
FORM get\_fields  USING P\_A P\_B P\_C.  
STATICS INDEX TYPE I VALUE 1.  
  
WA\_FIELDCAT-col\_pos = index. "COLUMN POSITION  
  WA\_FIELDCAT-FIELDNAME = P\_A. "COLUMN NAME  
  WA\_FIELDCAT-TABNAME = P\_B.   "TABLE 1 EKKO  
  WA\_FIELDCAT-REF\_TABNAME = P\_C.  
  WA\_FIELDCAT-REF\_FIELDNAME = P\_A.  
  
  INDEX = INDEX + 1.  
  
  APPEND WA\_FIELDCAT TO LT\_FIELDCAT.  
  CLEAR WA\_FIELDCAT.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form DISP\_GRID\_JOINS\_DATA  
  
FORM disp\_grid\_joins\_data .  
CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
 EXPORTING  
   I\_CALLBACK\_PROGRAM                = SY-REPID  
   I\_GRID\_TITLE                      = 'GRID JOINS FOR VBAP AND VBAK'  
\*   I\_GRID\_SETTINGS                   =  
\*   IS\_LAYOUT                         =  
   IT\_FIELDCAT                       = LT\_FIELDCAT  
  TABLES  
    t\_outtab                          = LT\_FINAL  
\* EXCEPTIONS  
\*   PROGRAM\_ERROR                     = 1  
\*   OTHERS                            = 2  
.  
  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
  
ENDFORM.

INCLUDE:

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAG\_GRID\_JOINS  
\*&---------------------------------------------------------------------\*  
  
  
\*&---------------------------------------------------------------------\*  
  
\*&---------------------------------------------------------------------\*  
  
types : begin of ls\_tab,  
          VBELN type VBELN, " SALES DOCUMENT  
          MATNR type MATNR, " MATERIAL NUMBER  
          MATWA type MATWA, " Material ENTERED  
          PSTYV type PSTYV, " ITEM CATEGORY  
          ARKTX type ARKTX, " SALES ORDER ITEM  
  
        end of ls\_tab,  
  
        BEGIN OF LS\_TAB2,  
          VBELN TYPE VBELN,  
          ERDAT TYPE ERDAT,  
          ANGDT TYPE ANGDT\_V,  
  
        END OF LS\_TAB2,  
  
        BEGIN OF LS\_FINAL,  
          VBELN TYPE VBELN,  
          MATNR TYPE MATNR,  
          MATWA TYPE MATWA,  
          PSTYV TYPE PSTYV,  
          ARKTX TYPE ARKTX,  
          ERDAT TYPE ERDAT,  
          ANGDT TYPE ANGDT,  
        END OF LS\_FINAL.  
  
  
  
    data : lt\_VBAP type standard table of ls\_tab,  " itab  
  
           lv\_VBELN type VBELN,  
  
           LT\_VBAK TYPE STANDARD TABLE OF LS\_TAB2,  
  
           LT\_FINAL TYPE STANDARD TABLE OF LS\_FINAL,  
  
  
           WA\_FIELDCAT TYPE SLIS\_FIELDCAT\_ALV,  " ALV FIELD CAT FOR WA  
           LT\_FIELDCAT TYPE SLIS\_T\_FIELDCAT\_ALV.

OUTPUT:



**8.FOR ALL ENTRIES**

\*&---------------------------------------------------------------------\*  
\*& Report ZNAGESH\_FORALL  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNAGESH\_FORALL.  
INCLUDE ZNAGESH\_FORALL\_ENTRIES\_TOP.  
  
TABLES VBAK.  
  
SELECT-OPTIONS s\_vbeln FOR vbak-vbeln.  
  
START-OF-SELECTION.  
  
  SELECT  
    VBELN  
    ERDAT  
    ERZET  
    ERNAM  
    ANGDT  
    BNDDT  
    AUDAT  
     INTO TABLE it\_tab1  
     FROM VBAK  
    WHERE vbeln IN s\_vbeln.  
  
    If not it\_tab1 is initial.  
      select  
        vbeln  
        POSNR  
        MATNR  
        MATWA  
        PMATN  
        into table it\_tab2  
        from vbap  
  
      FOR ALL ENTRIES IN it\_tab1   "  # 2 \*\*\*\*\*  
WHERE vbeln eq it\_tab1-vbeln.  
  
  
 DELETE ADJACENT DUPLICATES FROM it\_tab1 COMPARING vbeln.  "3rd  
  
sort  it\_tab2 by vbeln .  " 4th  
  
loop at it\_tab1 into wa\_tab1.  
  
READ TABLE it\_tab2 INTO wa\_tab2 WITH KEY vbeln = wa\_tab1-vbeln BINARY SEARCH. "5th  
  
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-ernam to wa\_final-ernam,  
      wa\_tab1-angdt to wa\_final-angdt,  
      wa\_tab1-bnddt to wa\_final-bnddt,  
      wa\_tab1-audat to wa\_final-audat,  
  
      wa\_tab2-posnr to wa\_final-posnr,  
      wa\_tab2-matnr to wa\_final-matnr,  
      wa\_tab2-matwa to wa\_final-matwa,  
      wa\_tab2-pmatn to wa\_final-pmatn.  
  
      Append wa\_final to it\_final.  
  
      else.  
        continue.  
        endif.  
        endloop.  
  
        LOOP AT it\_final INTO wa\_final.  
    WRITE:/ wa\_final-vbeln,  
            wa\_final-erdat,  
            wa\_final-erzet,  
            wa\_final-ernam,  
            wa\_final-angdt,  
            wa\_final-bnddt,  
            wa\_final-audat,  
            wa\_final-posnr,  
            wa\_final-matnr,  
            wa\_final-matwa,  
            wa\_final-pmatn.  
    endloop.  
  
    else.  
  
WRITE 'sales details records are not found'.    ""Common colums from the two tables will display within the given range 22 to 40(Example).  
  
  endif.

INCLUDE

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAGESH\_FORALL\_ENTRIES\_TOP  
\*&---------------------------------------------------------------------\*  
TYPES : BEGIN OF ls\_tab1,  " local strs  
VBELN  TYPE VBELN\_VA,         ""Fields of VBAK Table  
ERDAT  TYPE ERDAT,  
ERZET TYPE  ERZET,  
ERNAM TYPE  ERNAM,  
ANGDT  TYPE ANGDT\_V,  
BNDDT  TYPE   BNDDT,  
AUDAT  TYPE  AUDAT,  
  
  END OF ls\_tab1.  
  
  DATA :it\_tab1 TYPE STANDARD TABLE OF ls\_tab1, " int table  
      wa\_tab1 TYPE ls\_tab1.   " work area.  
  
TYPES : Begin of ls\_tab2,  
  VBELN  TYPE VBELN\_VA,      ""Fields of VBAP Table  
POSNR   TYPE POSNR\_VA,  
MATNR TYPE  MATNR,  
MATWA  TYPE MATWA,  
PMATN  TYPE PMATN,  
  
  END OF ls\_tab2.  
  Data : it\_tab2 type STANDARD TABLE OF ls\_tab2,  
        wa\_tab2 type ls\_tab2.  
  
  
  TYPES : Begin of ls\_final,  
  
VBELN  TYPE VBELN\_VA,         ""Fields of VBAK Table  
ERDAT  TYPE ERDAT,  
ERZET TYPE  ERZET,  
ERNAM TYPE  ERNAM,  
ANGDT  TYPE ANGDT\_V,  
BNDDT  TYPE   BNDDT,  
AUDAT  TYPE  AUDAT,  
  
POSNR   TYPE POSNR\_VA,  
MATNR TYPE  MATNR,  
MATWA  TYPE MATWA,  
PMATN  TYPE PMATN,  
    END OF ls\_final.  
  
    data : it\_final type STANDARD TABLE OF ls\_final,  
          wa\_final type ls\_final.

**OUTPUT:**



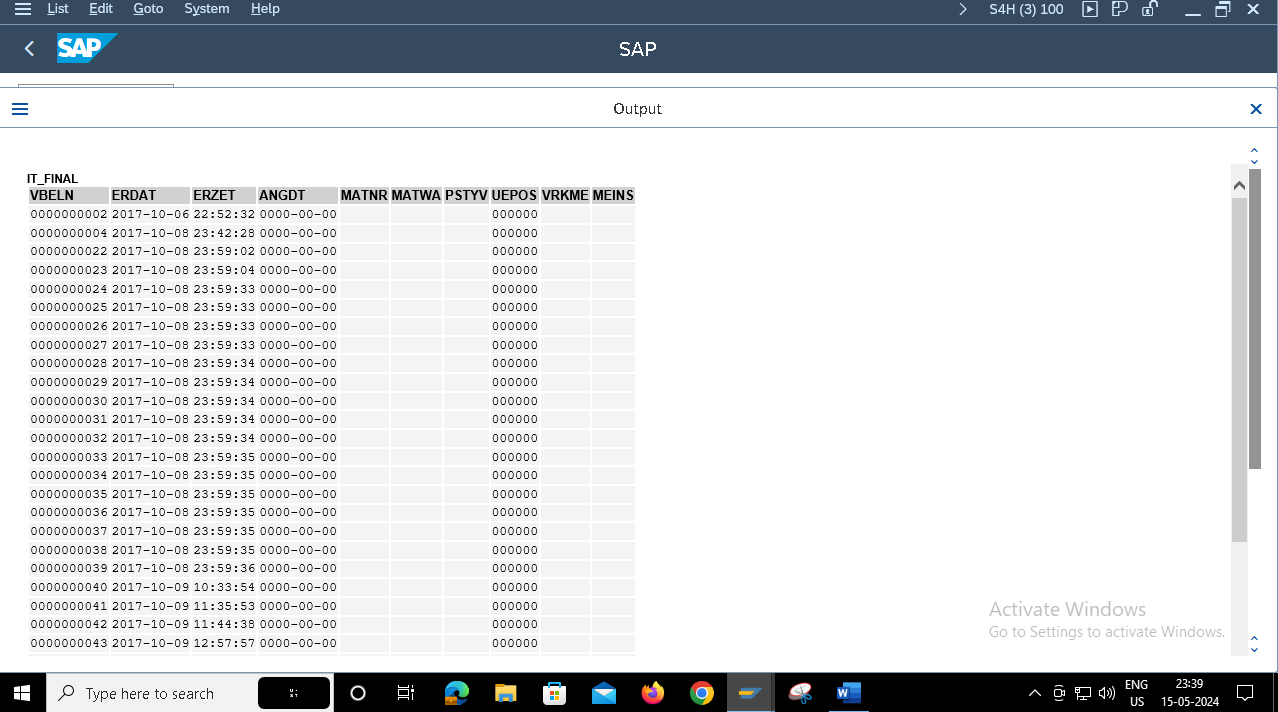
**9.FOR ALL ENTRIES WITH 3 TABLES:**

\*&---------------------------------------------------------------------\*  
\*& Report ZNAG\_FORALL\_WITH\_3TAB  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNAG\_FORALL\_WITH\_3TAB NO STANDARD PAGE HEADING LINE-COUNT 25(3)  
LINE-SIZE 500 MESSAGE-ID ZNAGESH\_CLASS\_R .  
  
Include Znag\_forall\_inc\_3tab.  
  
  
  
SELECT-OPTIONS S\_VBELN FOR WA\_VBAK-VBELN.  
  
INITIALIZATION.  
S\_VBELN-LOW = '1'.  
S\_VBELN-HIGH = '100'.  
  
APPEND S\_VBELN.  
  
START-OF-SELECTION.  
  
PERFORM GET\_FORALLENTRIES.  
  
END-OF-SELECTION.  
PERFORM DISPLAY\_FORALL.  
\*&---------------------------------------------------------------------\*  
\*& Form GET\_FORALLENTRIES  
  
FORM get\_forallentries .  
  
SELECT VBELN ERDAT ERZET ANGDT  
  INTO TABLE IT\_VBAK  
  FROM VBAK  
  WHERE VBELN IN S\_VBELN.  
  
  IF IT\_VBAK[] IS NOT INITIAL. "CHECKS IT\_VBAK HAS VALUES OR NOT  
\* WRITE 'ENTRIES FOUND'.  
    SELECT VBELN MATNR MATWA PSTYV  
      INTO TABLE IT\_VBAP  
      FROM VBAP  
      FOR ALL ENTRIES IN IT\_VBAK  
      WHERE VBELN = IT\_VBAK-VBELN.  
  
   ENDIF.  
  
  IF IT\_VBAP[] IS NOT INITIAL.  
  
    SELECT VBELN UEPOS VRKME MEINS  
      INTO TABLE IT\_VBRP  
      FROM VBRP  
      FOR ALL ENTRIES IN IT\_VBAP  
      WHERE VBELN = IT\_VBAP-VBELN.  
  ENDIF.  
  
  DELETE ADJACENT DUPLICATES FROM IT\_VBAK COMPARING VBELN.  
  
  SORT IT\_VBAK BY VBELN.  
  
  LOOP AT IT\_VBAK INTO WA\_VBAK.  
    READ TABLE IT\_VBAP INTO WA\_VBAP WITH KEY VBELN = WA\_VBAK-VBELN BINARY SEARCH.  
    READ TABLE IT\_VBRP INTO WA\_VBRP WITH KEY VBELN = WA\_VBAP-VBELN BINARY SEARCH.  
  
   MOVE : WA\_VBAK-VBELN TO WA\_FINAL-VBELN,  
          WA\_VBAK-ERDAT TO WA\_FINAL-ERDAT,  
          WA\_VBAK-ERZET TO WA\_FINAL-ERZET,  
          WA\_VBAK-ANGDT TO WA\_FINAL-ANGDT,  
  
          WA\_VBAP-MATNR TO WA\_FINAL-MATNR,  
          WA\_VBAP-MATWA TO WA\_FINAL-MATWA,  
          WA\_VBAP-PSTYV TO WA\_FINAL-PSTYV,  
  
          WA\_VBRP-UEPOS TO WA\_FINAL-UEPOS,  
          WA\_VBRP-VRKME TO WA\_FINAL-VRKME,  
          WA\_VBRP-MEINS TO WA\_FINAL-MEINS.  
  
   APPEND WA\_FINAL TO IT\_FINAL.  
   ENDLOOP.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form DISPLAY\_FORALL  
  
FORM display\_forall .  
CL\_DEMO\_OUTPUT=>display( IT\_FINAL ).  
ENDFORM.

**INCLUDE:**

\*&---------------------------------------------------------------------\*  
\*& Include          ZNAG\_FORALL\_INC\_3TAB  
\*&---------------------------------------------------------------------\*  
\*&---------------------------------------------------------------------\*  
\*& Include          ZRAS\_FORALLENT\_3TABS  
\*&---------------------------------------------------------------------\*  
  
TYPES : BEGIN OF ST\_VBAK,  
        VBELN TYPE VBELN\_VA,  
        ERDAT TYPE ERDAT,  
        ERZET TYPE ERZET,  
        ANGDT TYPE ANGDT\_V,  
        END OF ST\_VBAK.  
  
DATA : IT\_VBAK TYPE STANDARD TABLE OF ST\_VBAK,  
       WA\_VBAK TYPE ST\_VBAK.  
  
TYPES : BEGIN OF ST\_VBAP,  
        VBELN TYPE VBELN\_VA,  
        MATNR TYPE MATNR,  
        MATWA TYPE MATWA,  
        PSTYV TYPE PSTYV,  
        END OF ST\_VBAP.  
  
DATA : IT\_VBAP TYPE STANDARD TABLE OF ST\_VBAP,  
       WA\_VBAP TYPE ST\_VBAP.  
  
TYPES : BEGIN OF ST\_VBRP,  
        VBELN TYPE VBELN\_VF,  
        UEPOS TYPE UEPOS,  
        VRKME TYPE VRKME,  
        MEINS TYPE MEINS,  
        END OF ST\_VBRP.  
  
DATA : IT\_VBRP TYPE STANDARD TABLE OF ST\_VBRP,  
       WA\_VBRP TYPE ST\_VBRP.  
  
TYPES : BEGIN OF ST\_FINAL,  
        VBELN TYPE VBELN\_VA,  
        ERDAT TYPE ERDAT,  
        ERZET TYPE ERZET,  
        ANGDT TYPE ANGDT\_V,  
        MATNR TYPE MATNR,  
        MATWA TYPE MATWA,  
        PSTYV TYPE PSTYV,  
        UEPOS TYPE UEPOS,  
        VRKME TYPE VRKME,  
        MEINS TYPE MEINS,  
        END OF ST\_FINAL.  
  
DATA : IT\_FINAL TYPE STANDARD TABLE OF ST\_FINAL,  
       WA\_FINAL TYPE ST\_FINAL.

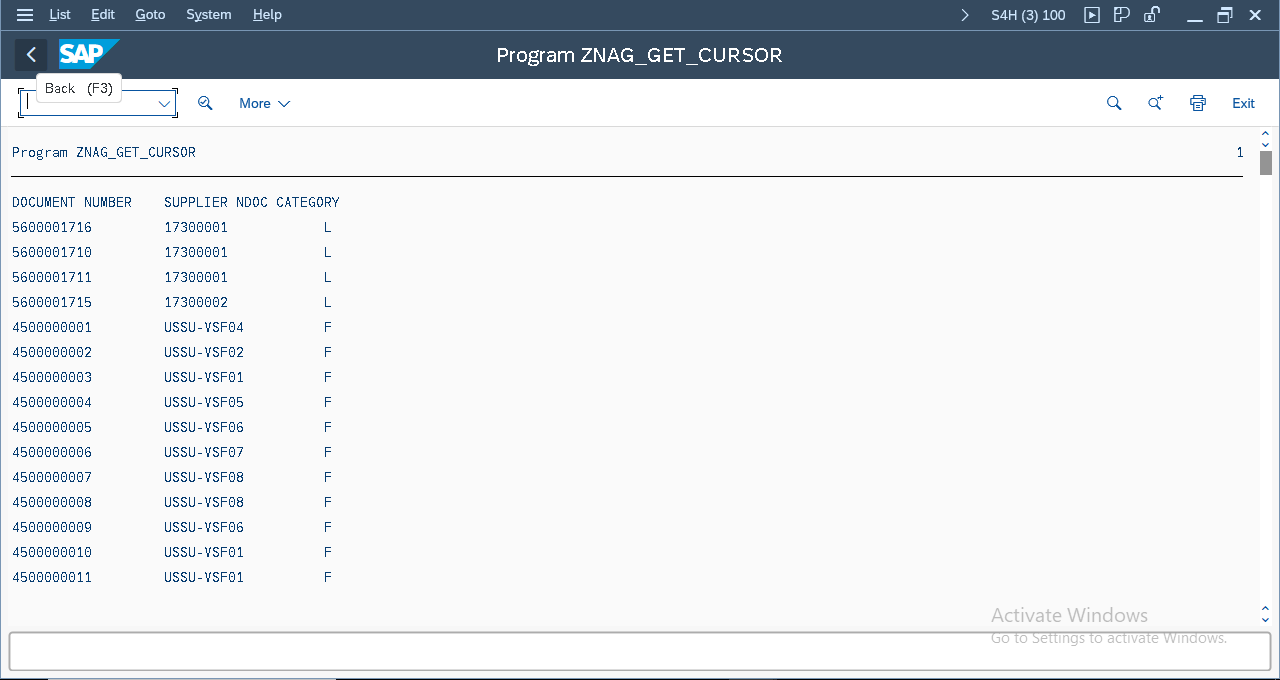
**OUTPUT:**

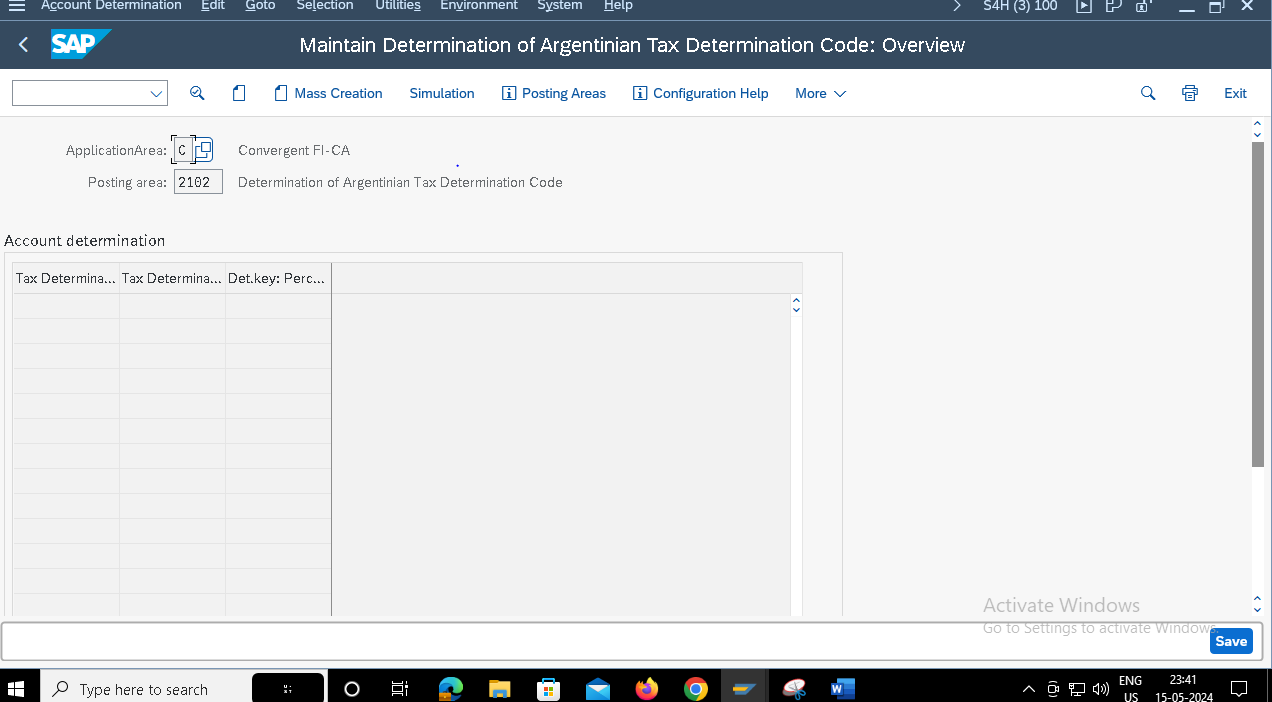


**10.GET CURSOR**

\*&---------------------------------------------------------------------\*  
\*& Report ZNAG\_GET\_CURSOR  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT znag\_get\_cursor.  
  
DATA lv\_ebeln TYPE ebeln. ""EKKO TABLE  
  
TYPES: BEGIN OF ls\_ekko,  
         ebeln TYPE ebeln,  
         lifnr TYPE lifnr,  
         bstyp TYPE ebstyp,  
       END OF ls\_ekko.  
  
DATA : it\_ekko TYPE TABLE OF ls\_ekko,  
       wa\_ekko TYPE ls\_ekko.  
  
SELECT-OPTIONS s\_ebeln FOR lv\_ebeln.  
  
START-OF-SELECTION.  
  
  SELECT ebeln lifnr bstyp  
    INTO TABLE it\_ekko  
    FROM ekko  
    WHERE ebeln IN s\_ebeln.  
  
END-OF-SELECTION.  
  
  LOOP AT it\_ekko INTO wa\_ekko.  
  
    FORMAT HOTSPOT ON.  
    WRITE : / wa\_ekko-ebeln,  ""ebeln from ekko table  
             20 wa\_ekko-lifnr.  ""lifnr from lAF1 table  
  
    FORMAT HOTSPOT OFF.  
    WRITE     40 wa\_ekko-bstyp.  
  ENDLOOP.  
  
  TOP-OF-PAGE.  
WRITE : /'DOCUMENT NUMBER',  
          20 'SUPPLIER NUMBER',  
          30 'DOC CATEGORY'.  
  
AT LINE-SELECTION.  
  
DATA : FNAME(15),  
       FVALUE(15).  
  
GET CURSOR FIELD FNAME VALUE FVALUE.  
  
IF FNAME = 'WA\_EKKO-EBELN'.  
  SET PARAMETER ID 'BES' FIELD FVALUE. ""BES is the parameter id of ekko-ebeln  
  CALL TRANSACTION 'EK03' AND SKIP FIRST SCREEN.  ""EKO3 is the T code for ekko table to display the data  
  
ELSEIF FNAME = 'WA\_EKKO-LIFNR'.  
  SET PARAMETER ID  'LIF' FIELD FVALUE.  ""LIF is the p id of LFA1 table  
  CALL TRANSACTION 'MK02' AND SKIP FIRST SCREEN. "" MK02 is the T code for LFA1 to do change in  vendor table LFA1  
ENDIF.  
\*\*\*\*\*\*The vendor for EKKO is the LFA1.

**OUTPUT:**





**11.SPELL AMOUNT:**

\*&---------------------------------------------------------------------\*  
\*& Report ZNAG\_SPELL\_AMOUNT\_PRO  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNAG\_SPELL\_AMOUNT\_PRO.  
  
data lv\_spell type spell.  
parameters p\_amt type i.  
  
call function 'SPELL\_AMOUNT'  
  exporting  
    amount   = p\_amt  
    language = sy-langu  
  importing  
    in\_words = lv\_spell  
\* EXCEPTIONS  
\*   NOT\_FOUND       = 1  
\*   TOO\_LARGE       = 2  
\*   OTHERS   = 3  
  .  
if sy-subrc <> 0.  
\* Implement suitable error handling here  
endif.  
  
write : lv\_spell-word.

**OUTPUT:**

