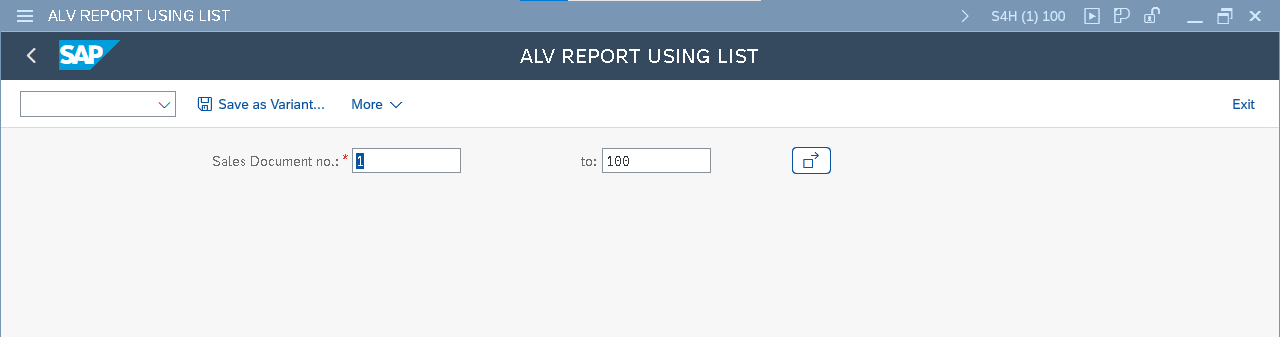
ALV REPORTS

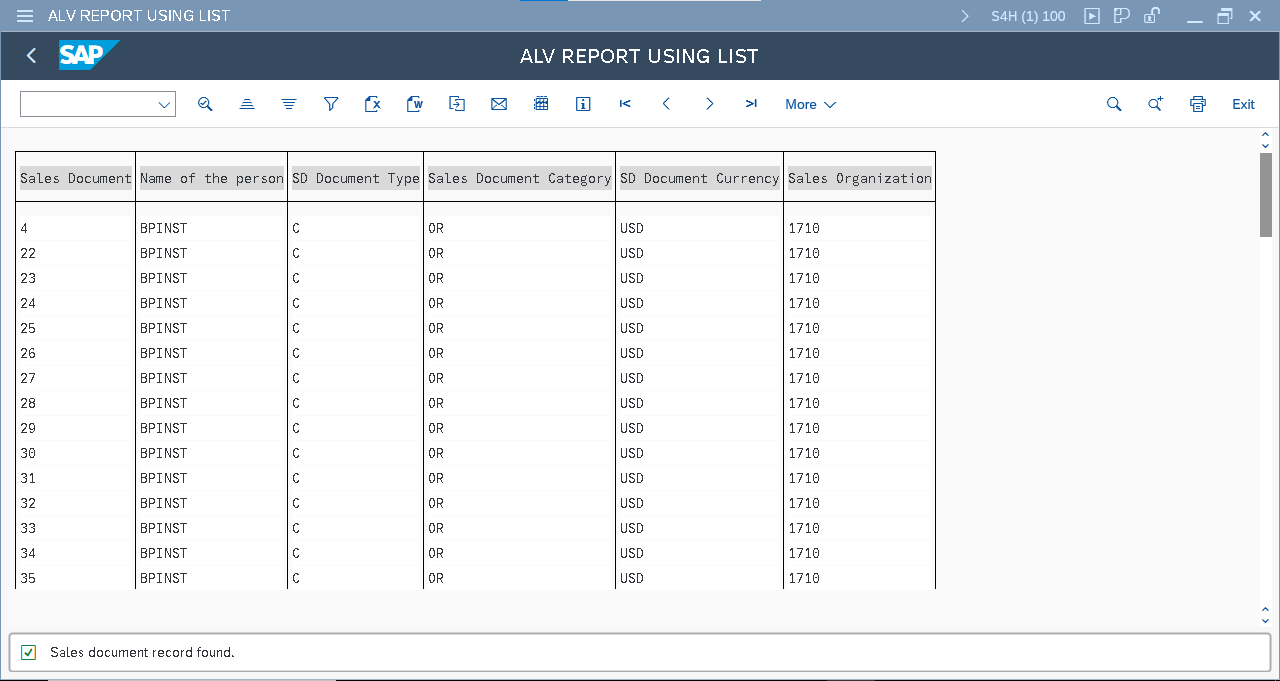
1. ALV Report using list

\*&---------------------------------------------------------------------\*  
\*& Report ZNIDH\_ALV\_LIST\_DISP  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNIDH\_ALV\_LIST\_DISP NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500 MESSAGE-ID ZNIDH\_MSG.  
  
INCLUDE ZNIDH\_ALV\_LIST\_DISP\_TOPINC.  
  
select-OPTIONS s\_vbeln for lv\_vbeln OBLIGATORY.  
  
INITIALIZATION.  
  
clear :  wa\_fieldcat , lt\_fieldcat.  
s\_vbeln-low = '0000000001'.  
s\_vbeln-high = '0000000100'.  
append s\_vbeln.  
  
at SELECTION-SCREEN on s\_vbeln.  
  select single vbeln  
    into lv\_vbeln  
    from vbak  
    where vbeln in s\_vbeln.  
  
    if sy-subrc eq 0.  
      MESSAGE s000.  
    else.  
      MESSAGE e001.  
    endif.  
  
START-OF-SELECTION.  
    perform get\_alv\_fieldcat.  
    perform get\_vbe\_data.  
end-of-SELECTION.  
    perform disp\_alv\_fieldcat.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_alv\_fieldcat  
\*&---------------------------------------------------------------------\*  
FORM get\_alv\_fieldcat .  
  "VBELN  
  wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'Sales Document'.  
  wa\_fieldcat-fieldname = 'VBELN'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "ERNAM  
  wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'Name of the person'.  
  wa\_fieldcat-fieldname = 'ERNAM'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "VBTYP  
  wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'SD Document Type'.  
  wa\_fieldcat-fieldname = 'VBTYP'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "AUART  
  wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'Sales Document Category'.  
  wa\_fieldcat-fieldname = 'AUART'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "WAERK  
  wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = 'SD Document Currency'.  
  wa\_fieldcat-fieldname = 'WAERK'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "VKORG  
  wa\_fieldcat-col\_pos = '6'.  
  wa\_fieldcat-seltext\_l = 'Sales Organization'.  
  wa\_fieldcat-fieldname = 'VKORG'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_vbe\_data  
\*&---------------------------------------------------------------------\*  
FORM get\_vbe\_data .  
  select vbeln ernam vbtyp auart waerk vkorg  
    into table lt\_tab  
    from vbak  
    where vbeln in s\_vbeln.  
  
  if sy-subrc eq 0.  
    MESSAGE s000.  
  else.  
    MESSAGE e001.  
  endif.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form disp\_alv\_fieldcat  
\*&---------------------------------------------------------------------\*  
FORM disp\_alv\_fieldcat .  
  CALL FUNCTION 'REUSE\_ALV\_LIST\_DISPLAY'  
  EXPORTING  
     I\_CALLBACK\_PROGRAM             = sy-repid  
     IT\_FIELDCAT                    =  lt\_fieldcat  
  
    TABLES  
      t\_outtab                       = lt\_tab  
   EXCEPTIONS  
     PROGRAM\_ERROR                  = 1  
     OTHERS                         = 2  
            .  
\*  IF sy-subrc <> 0.  
\*  " Implement suitable error handling here  
\*  
\*  ENDIF.  
  
  
ENDFORM.

INCLUDE PROGRAM FOR THIS

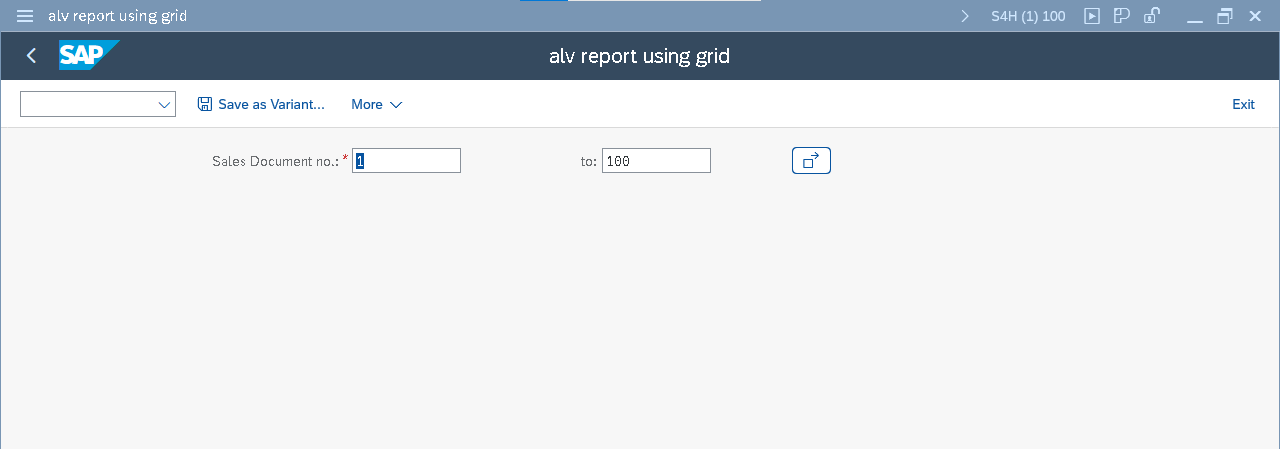
\*&---------------------------------------------------------------------\*  
\*& Include          ZNIDH\_ALV\_LIST\_DISP\_TOPINC  
\*&---------------------------------------------------------------------\*  
  
  
types : begin of ls\_tab,  
        vbeln type vbeln\_va, "Sales document  
        ernam type ernam, "Name of the person who created the object  
        vbtyp type vbtyp, "SD Document category  
        auart type auart, "Sales document type  
        waerk type waerk,"SD Socument currency  
        vkorg type vkorg, "Sales Organization  
  
       end of ls\_tab.  
  
data : lt\_tab type STANDARD TABLE OF ls\_tab,  
       lt\_fieldcat type slis\_t\_fieldcat\_alv,  
       wa\_fieldcat type slis\_fieldcat\_alv,   "work area fieldcat  
       lv\_vbeln type vbeln.

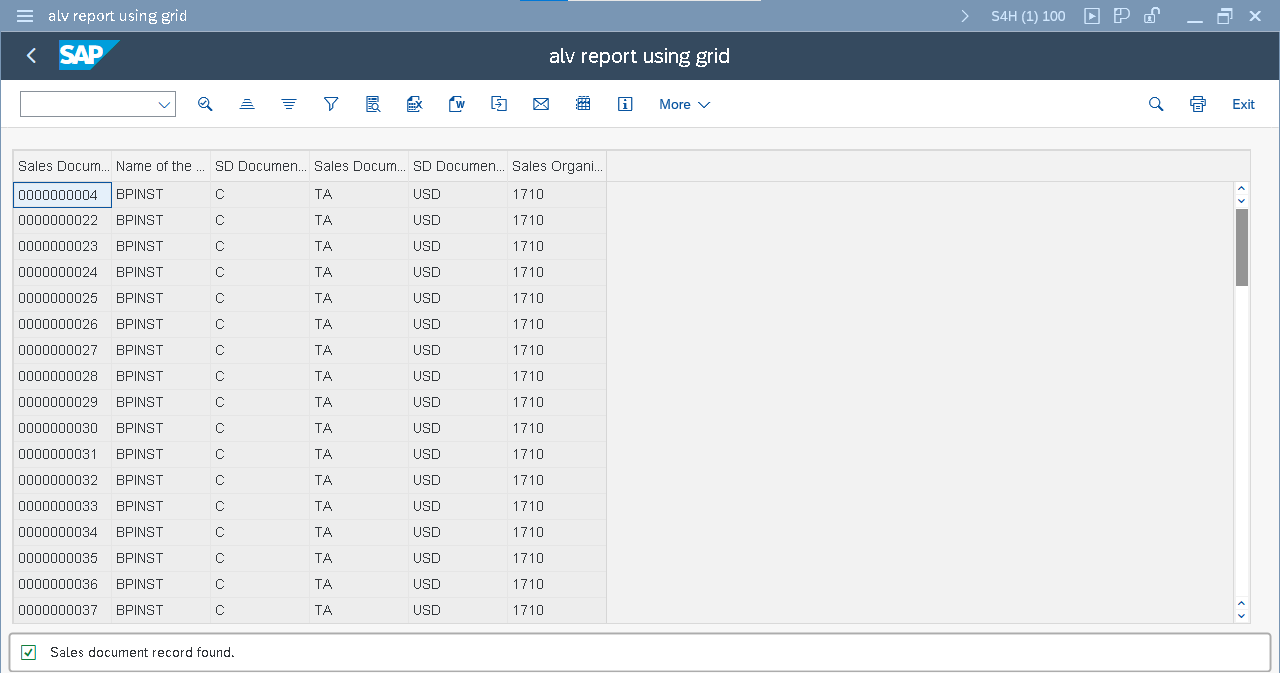




1. ALV REPORTS USING GRID

\*&---------------------------------------------------------------------\*  
\*& Report ZNIDH\_ALV\_GRID\_DISP  
\*&---------------------------------------------------------------------\*  
  
REPORT ZNIDH\_ALV\_GRID\_DISP NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500 MESSAGE-ID ZNIDH\_MSG.  
  
INCLUDE ZNIDH\_ALV\_LIST\_DISP\_TOPINC.  
  
select-OPTIONS s\_vbeln for lv\_vbeln OBLIGATORY.  
  
INITIALIZATION.  
  
clear :  wa\_fieldcat , lt\_fieldcat.  
s\_vbeln-low = '0000000001'.  
s\_vbeln-high = '0000000100'.  
append s\_vbeln.  
  
at SELECTION-SCREEN on s\_vbeln.  
  select single vbeln  
    into lv\_vbeln  
    from vbak  
    where vbeln in s\_vbeln.  
  
    if sy-subrc eq 0.  
      MESSAGE s000.  
    else.  
      MESSAGE e001.  
    endif.  
  
START-OF-SELECTION.  
    perform get\_alv\_fieldcat.  
    perform get\_vbe\_data.  
end-of-SELECTION.  
    perform disp\_alv\_fieldcat.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_alv\_fieldcat  
\*&---------------------------------------------------------------------\*  
FORM get\_alv\_fieldcat .  
  "VBELN  
  wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'Sales Document'.  
  wa\_fieldcat-fieldname = 'VBELN'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "ERNAM  
  wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'Name of the person'.  
  wa\_fieldcat-fieldname = 'ERNAM'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "VBTYP  
  wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'SD Document Type'.  
  wa\_fieldcat-fieldname = 'VBTYP'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "AUART  
  wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'Sales Document Category'.  
  wa\_fieldcat-fieldname = 'AUART'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "WAERK  
  wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = 'SD Document Currency'.  
  wa\_fieldcat-fieldname = 'WAERK'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "VKORG  
  wa\_fieldcat-col\_pos = '6'.  
  wa\_fieldcat-seltext\_l = 'Sales Organization'.  
  wa\_fieldcat-fieldname = 'VKORG'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_vbe\_data  
\*&---------------------------------------------------------------------\*  
FORM get\_vbe\_data .  
  select vbeln ernam vbtyp auart waerk vkorg  
    into table lt\_tab  
    from vbak  
    where vbeln in s\_vbeln.  
  
  if sy-subrc eq 0.  
    MESSAGE s000.  
  else.  
    MESSAGE e001.  
  endif.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form disp\_alv\_fieldcat  
\*&---------------------------------------------------------------------\*  
FORM disp\_alv\_fieldcat .  
  CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
  EXPORTING  
  
     I\_CALLBACK\_PROGRAM                = sy-repid  
     IT\_FIELDCAT                       = lt\_fieldcat  
  
    TABLES  
      t\_outtab                          = lt\_tab  
   EXCEPTIONS  
     PROGRAM\_ERROR                     = 1  
     OTHERS                            = 2  
            .  
  IF sy-subrc <> 0.  
\* Implement suitable error handling here  
  ENDIF.  
  
ENDFORM.





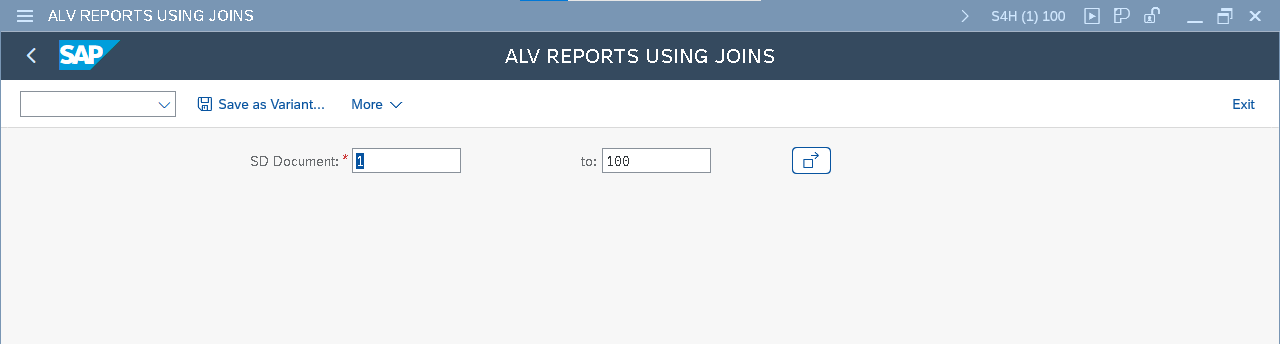
1. ALV REPORTS USING GRID JOINS

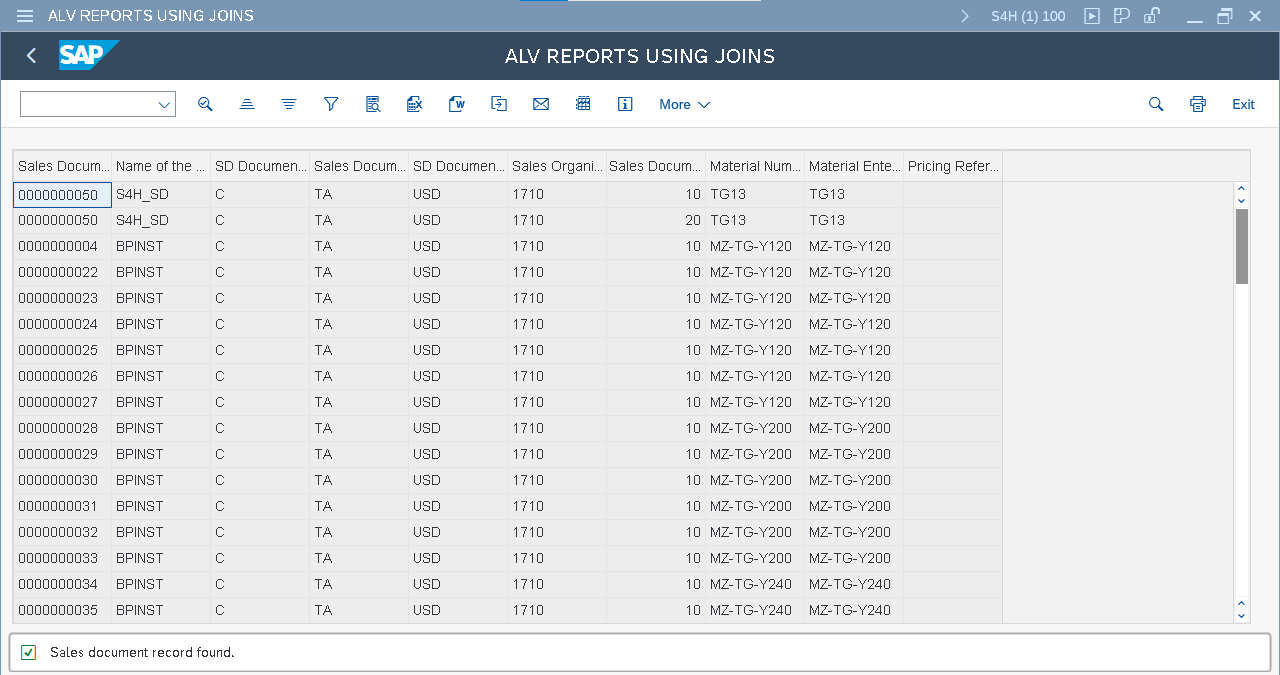
\*&---------------------------------------------------------------------\*  
\*& Report ZNIDH\_ALV\_JOINSGRID\_DISP  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNIDH\_ALV\_JOINSGRID\_DISP NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500 MESSAGE-ID ZNIDH\_MSG.  
  
INCLUDE ZNIDH\_ALV\_JOINS\_DISP\_TOPINC.  
  
select-OPTIONS s\_vbeln for lv\_vbeln OBLIGATORY.  
  
INITIALIZATION.  
  
clear :  wa\_fieldcat , lt\_fieldcat.  
s\_vbeln-low = '0000000001'.  
s\_vbeln-high = '0000000100'.  
append s\_vbeln.  
  
at SELECTION-SCREEN on s\_vbeln.  
  select single vbeln  
    into lv\_vbeln  
    from vbak  
    where vbeln in s\_vbeln.  
  
    if sy-subrc eq 0.  
      MESSAGE s000.  
    else.  
      MESSAGE e001.  
    endif.  
  
START-OF-SELECTION.  
    perform get\_alv\_fieldcat.  
    perform get\_vbe\_data.  
end-of-SELECTION.  
    perform disp\_alv\_fieldcat.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_alv\_fieldcat  
\*&---------------------------------------------------------------------\*  
FORM get\_alv\_fieldcat .  
  "VBELN  
  wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'Sales Document'.  
  wa\_fieldcat-fieldname = 'VBELN'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "ERNAM  
  wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'Name of the person'.  
  wa\_fieldcat-fieldname = 'ERNAM'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "VBTYP  
  wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'SD Document Type'.  
  wa\_fieldcat-fieldname = 'VBTYP'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "AUART  
  wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'Sales Document Category'.  
  wa\_fieldcat-fieldname = 'AUART'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "WAERK  
  wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = 'SD Document Currency'.  
  wa\_fieldcat-fieldname = 'WAERK'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "VKORG  
  wa\_fieldcat-col\_pos = '6'.  
  wa\_fieldcat-seltext\_l = 'Sales Organization'.  
  wa\_fieldcat-fieldname = 'VKORG'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "POSNR  
  wa\_fieldcat-col\_pos = '7'.  
  wa\_fieldcat-seltext\_l = 'Sales Document Item'.  
  wa\_fieldcat-fieldname = 'POSNR'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "MATNR  
  wa\_fieldcat-col\_pos = '8'.  
  wa\_fieldcat-seltext\_l = 'Material Number'.  
  wa\_fieldcat-fieldname = 'MATNR'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  
  "MATWA  
  wa\_fieldcat-col\_pos = '9'.  
  wa\_fieldcat-seltext\_l = 'Material Entered'.  
  wa\_fieldcat-fieldname = 'MATWA'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "PMATN  
  wa\_fieldcat-col\_pos = '10'.  
  wa\_fieldcat-seltext\_l = 'Pricing Reference Material'.  
  wa\_fieldcat-fieldname = 'PMATN'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_vbe\_data  
\*&---------------------------------------------------------------------\*  
FORM get\_vbe\_data .  
  select vbak~vbeln  
         vbak~ernam  
         vbak~vbtyp  
         vbak~auart  
         vbak~waerk  
         vbak~vkorg  
  
         vbap~posnr  
         vbap~matnr  
         vbap~matwa  
         vbap~pmatn  
    into table lt\_tab  
    from vbak inner join vbap on vbak~vbeln = vbap~vbeln  
    where vbak~vbeln in s\_vbeln.  
  
  if sy-subrc eq 0.  
    MESSAGE s000.  
  else.  
    MESSAGE e001.  
  endif.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form disp\_alv\_fieldcat  
\*&---------------------------------------------------------------------\*  
FORM disp\_alv\_fieldcat .  
  CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
  EXPORTING  
  
     I\_CALLBACK\_PROGRAM                = sy-repid  
     IT\_FIELDCAT                       = lt\_fieldcat  
  
    TABLES  
      t\_outtab                          = lt\_tab  
   EXCEPTIONS  
     PROGRAM\_ERROR                     = 1  
     OTHERS                            = 2  
            .  
  IF sy-subrc <> 0.  
\* Implement suitable error handling here  
  ENDIF.  
  
ENDFORM.

INCLUDE PROGRAM FOR THIS

\*&---------------------------------------------------------------------\*  
\*& Include          ZNIDH\_ALV\_JOINS\_DISP\_TOPINC  
\*&---------------------------------------------------------------------\*  
  
types : begin of ls\_tab,  
        vbeln type vbeln\_va, "Sales document  
        ernam type ernam, "Name of the person who created the object  
        vbtyp type vbtyp, "SD Document category  
        auart type auart, "Sales document type  
        waerk type waerk,"SD Socument currency  
        vkorg type vkorg, "Sales Organization  
  
        posnr TYPE posnr\_va, "Sales Document Item  
        matnr TYPE matnr, "Material Number  
        matwa TYPE matwa, "Material Entered  
        pmatn TYPE pmatn, "Pricing Reference Material  
  
       end of ls\_tab.  
  
data : lt\_tab type STANDARD TABLE OF ls\_tab,  
       lt\_fieldcat type slis\_t\_fieldcat\_alv,  
       wa\_fieldcat type slis\_fieldcat\_alv,  
       lv\_vbeln type vbeln.

OUTPUT





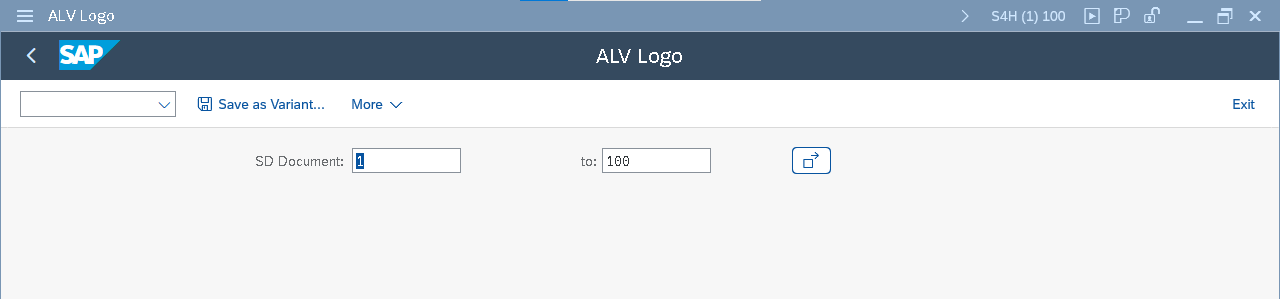
4)ALV REPORTS USING GRID CREATING LOGO

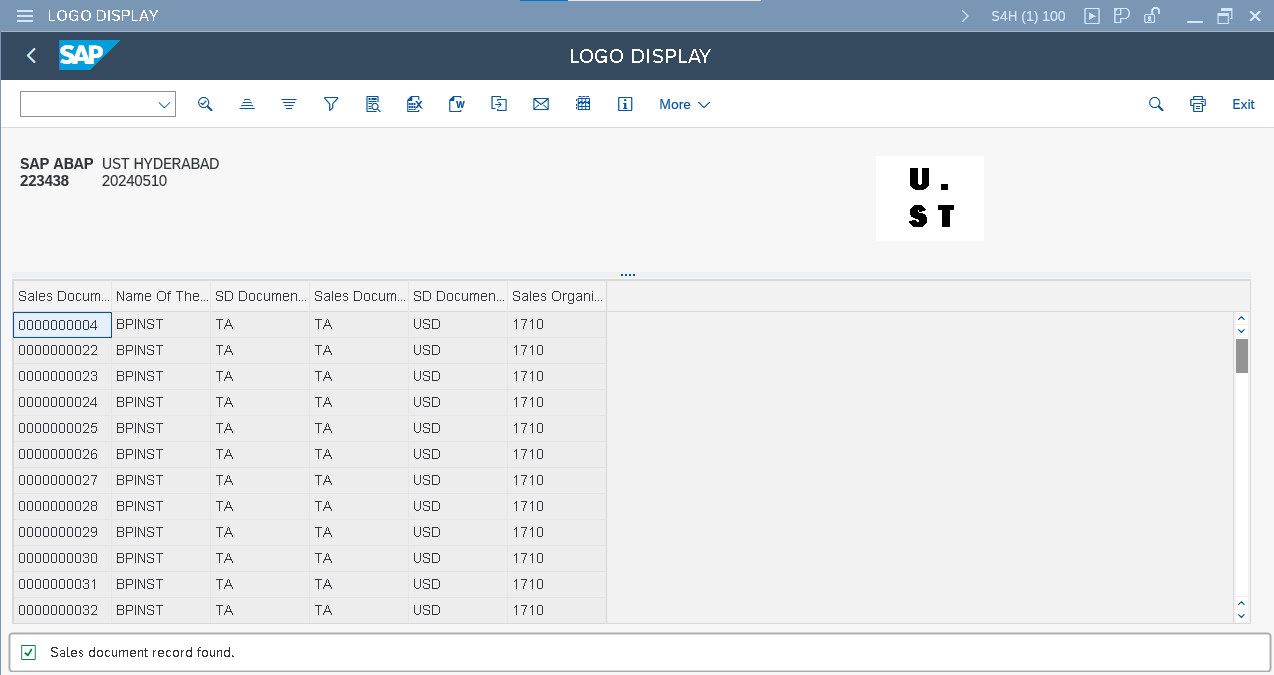
\*&---------------------------------------------------------------------\*  
\*& Report ZNIDH\_ALV\_LOGO  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNIDH\_ALV\_LOGO NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500 MESSAGE-ID ZNIDH\_MSG.  
  
INCLUDE ZNIDH\_ALV\_LOGO\_TOPINC.  
  
select-options s\_vbeln for lv\_vbeln.  
  
INITIALIZATION.  
clear : lt\_tab, lt\_fieldcat, wa\_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 vbak  
    where vbeln in s\_vbeln.  
  
  if sy-subrc eq 0.  
    MESSAGE s000.  
  else.  
    MESSAGE e001.  
  ENDIF.  
  
START-OF-SELECTION.  
  perform get\_alv\_fcat.  
  perform get\_vbe\_data.  
end-of-SELECTION.  
  perform disp\_alv\_fieldcat.  
  
\*&---------------------------------------------------------------------\*  
\*& Form get\_alv\_fcat  
\*&---------------------------------------------------------------------\*  
FORM get\_alv\_fcat .  
  "VBELN  
  wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = text-001.  
  wa\_fieldcat-fieldname = cs\_vbeln.  
  wa\_fieldcat-tabname = cs\_tab.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "ERNAM  
  wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = text-002.  
  wa\_fieldcat-fieldname = cs\_ernam.  
  wa\_fieldcat-tabname = cs\_tab.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "VBTYP  
  wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = text-003.  
  wa\_fieldcat-fieldname = cs\_auart.  
  wa\_fieldcat-tabname = cs\_tab.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "AUART  
  wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = text-004.  
  wa\_fieldcat-fieldname = cs\_auart.  
  wa\_fieldcat-tabname = cs\_tab.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "WAERK  
  wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = text-005.  
  wa\_fieldcat-fieldname = cs\_waerk.  
  wa\_fieldcat-tabname = cs\_tab.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "VKORG  
  wa\_fieldcat-col\_pos = '6'.  
  wa\_fieldcat-seltext\_l = text-006.  
  wa\_fieldcat-fieldname = cs\_vkorg.  
  wa\_fieldcat-tabname = cs\_tab.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_vbe\_data  
\*&---------------------------------------------------------------------\*  
FORM get\_vbe\_data .  
  select vbeln ernam vbtyp auart waerk vkorg  
    into table lt\_tab  
    from vbak  
    where vbeln in s\_vbeln.  
  
  if sy-subrc eq 0.  
    MESSAGE s000.  
  else.  
    MESSAGE e001.  
  endif.  
  
ENDFORM.  
  
FORM disp\_alv\_fieldcat.  
  
  ty\_layout-no\_colhead = 'X'.  
  ty\_layout-no\_hline = 'X'.  
  ty\_layout-no\_vline = 'X'.  
  
  CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
  EXPORTING  
  
     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                      = 'LOGO DISPLAY'  
\*     I\_GRID\_SETTINGS                   =  
\*     IS\_LAYOUT                         =  
     IT\_FIELDCAT                       =  lt\_fieldcat  
  
\*  
    TABLES  
      t\_outtab                          = lt\_tab  
   EXCEPTIONS  
     PROGRAM\_ERROR                     = 1  
     OTHERS                            = 2  
            .  
  IF sy-subrc <> 0.  
\* Implement suitable error handling here  
  ENDIF.  
  
  
ENDFORM.  
  
form ust\_logo.  
   data : lt\_header type slis\_t\_listheader,  
         wa\_header type slis\_listheader.  
  
   wa\_header-typ = 'S'.  
   wa\_header-info = 'UST HYDERABAD'.  
   wa\_header-key = 'SAP ABAP'.  
   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                   = 'ZNIDH\_LOGO'  " oaer object name  
\*      I\_END\_OF\_LIST\_GRID       =  
\*      I\_ALV\_FORM               =  
             .  
  
  
  
ENDFORM.

INCLUDE PROGRAM FOR THIS

\*&---------------------------------------------------------------------\*  
\*& Include          ZNIDH\_ALV\_LOGO\_TOPINC  
\*&---------------------------------------------------------------------\*  
  
  
types : begin of ls\_tab,  
        vbeln type vbeln\_va, "Sales document  
        ernam type ernam, "Name of the person who created the object  
        vbtyp type vbtyp, "SD Document category  
        auart type auart, "Sales document type  
        waerk type waerk,"SD Socument currency  
        vkorg type vkorg, "Sales Organization  
  
       end of ls\_tab.  
  
data : lt\_tab type STANDARD TABLE OF ls\_tab,  
       lt\_fieldcat type slis\_t\_fieldcat\_alv,  
       wa\_fieldcat type slis\_fieldcat\_alv,  
       lv\_vbeln type vbeln,  
       TY\_LAYOUT TYPE SLIS\_LAYOUT\_ALV.  
  
constants : cs\_tab(10) type c value 'lt\_tab',  
            cs\_vbeln(10) type c value 'VBELN',  
            cs\_ernam(10) type c value 'ERNAM',  
            cs\_vbtyp(10) type c value 'VBTYP',  
            cs\_auart(10) type c value 'AUART',  
            cs\_waerk(10) type c value 'WAERK',  
            cs\_vkorg(10) type c value 'VKORG'.

OUTPUT:



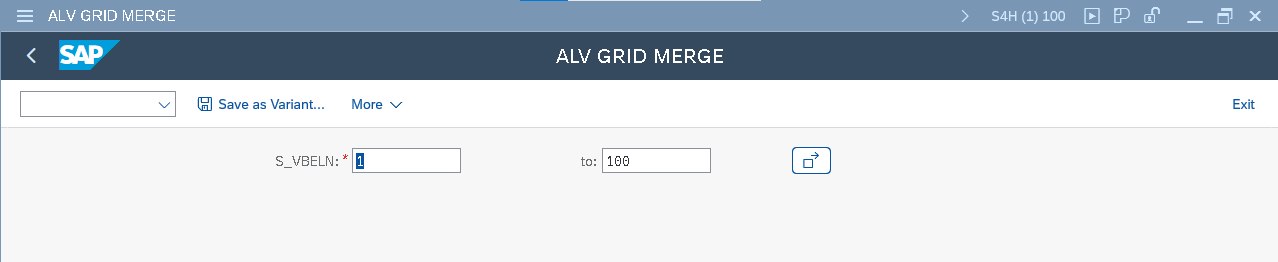


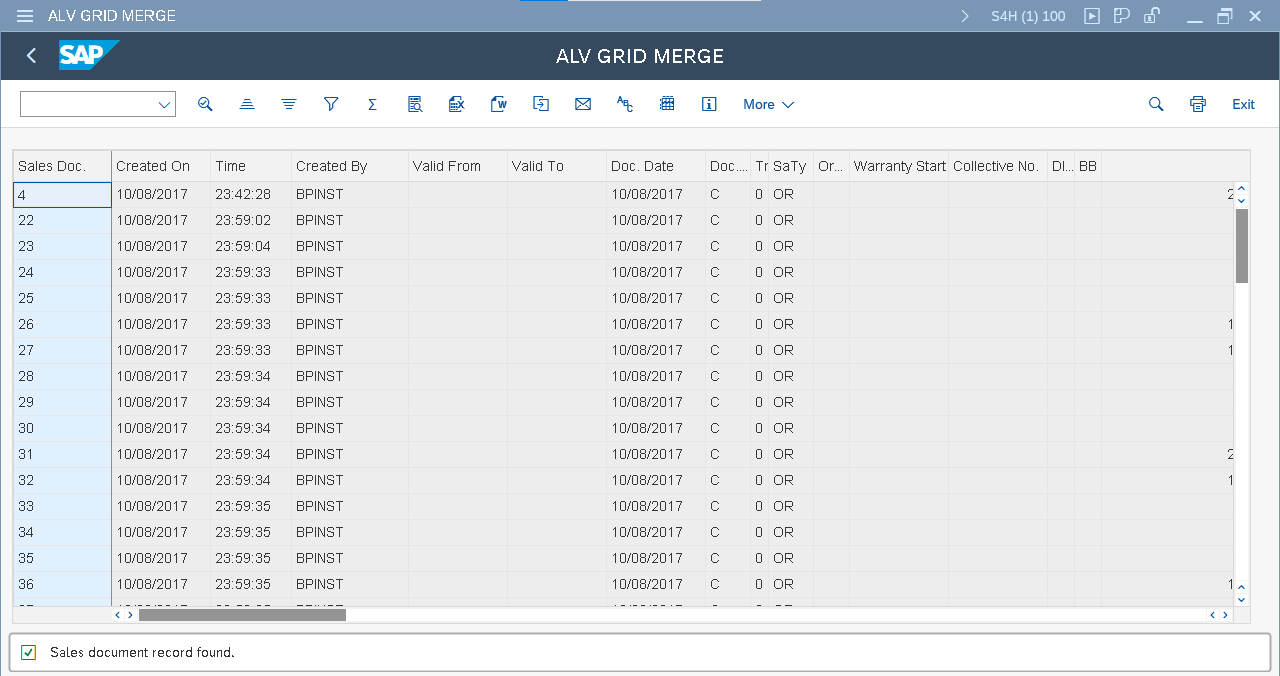
1. ALV REPORTS GRID MERGE

\*&---------------------------------------------------------------------\*  
\*& Report ZNIDH\_ALV\_GRID\_MERGE  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNIDH\_ALV\_GRID\_MERGE NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500 MESSAGE-ID ZNIDH\_MSG.  
  
INCLUDE ZNIDH\_ALV\_MERGE.  
  
select-OPTIONS s\_vbeln for lv\_vbeln OBLIGATORY.  
  
INITIALIZATION.  
  
clear :  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 vbak  
    where vbeln in s\_vbeln.  
  
    if sy-subrc eq 0.  
      MESSAGE s000.  
    else.  
      MESSAGE e001.  
    endif.  
  
START-OF-SELECTION.  
    perform get\_alv\_fieldcat.  
    perform get\_vbe\_data.  
end-of-SELECTION.  
    perform disp\_alv\_fieldcat.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_alv\_fieldcat  
\*&---------------------------------------------------------------------\*  
FORM get\_alv\_fieldcat .  
  CALL FUNCTION 'REUSE\_ALV\_FIELDCATALOG\_MERGE'  
   EXPORTING  
    I\_PROGRAM\_NAME               = sy-repid  
     I\_INTERNAL\_TABNAME           = 'lt\_tab'  
     I\_STRUCTURE\_NAME             = 'vbak'  
\*     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.  
  
\*  "VBELN  
\*  wa\_fieldcat-col\_pos = '1'.  
\*  wa\_fieldcat-seltext\_l = 'Sales Document'.  
\*  wa\_fieldcat-fieldname = 'VBELN'.  
\*  wa\_fieldcat-tabname = 'lt\_tab'.  
\*  
\*  append wa\_fieldcat to lt\_fieldcat.  
\*  clear wa\_fieldcat.  
\*  
\*  "ERNAM  
\*  wa\_fieldcat-col\_pos = '2'.  
\*  wa\_fieldcat-seltext\_l = 'Name of the person'.  
\*  wa\_fieldcat-fieldname = 'ERNAM'.  
\*  wa\_fieldcat-tabname = 'lt\_tab'.  
\*  
\*  
\*  append wa\_fieldcat to lt\_fieldcat.  
\*  clear wa\_fieldcat.  
\*  
\*  "VBTYP  
\*  wa\_fieldcat-col\_pos = '3'.  
\*  wa\_fieldcat-seltext\_l = 'SD Document Type'.  
\*  wa\_fieldcat-fieldname = 'VBTYP'.  
\*  wa\_fieldcat-tabname = 'lt\_tab'.  
\*  
\*  
\*  append wa\_fieldcat to lt\_fieldcat.  
\*  clear wa\_fieldcat.  
\*  
\*  "AUART  
\*  wa\_fieldcat-col\_pos = '4'.  
\*  wa\_fieldcat-seltext\_l = 'Sales Document Category'.  
\*  wa\_fieldcat-fieldname = 'AUART'.  
\*  wa\_fieldcat-tabname = 'lt\_tab'.  
\*  
\*  
\*  append wa\_fieldcat to lt\_fieldcat.  
\*  clear wa\_fieldcat.  
\*  
\*  "WAERK  
\*  wa\_fieldcat-col\_pos = '5'.  
\*  wa\_fieldcat-seltext\_l = 'SD Document Currency'.  
\*  wa\_fieldcat-fieldname = 'WAERK'.  
\*  wa\_fieldcat-tabname = 'lt\_tab'.  
\*  
\*  
\*  append wa\_fieldcat to lt\_fieldcat.  
\*  clear wa\_fieldcat.  
\*  
\*  "VKORG  
\*  wa\_fieldcat-col\_pos = '6'.  
\*  wa\_fieldcat-seltext\_l = 'Sales Organization'.  
\*  wa\_fieldcat-fieldname = 'VKORG'.  
\*  wa\_fieldcat-tabname = 'lt\_tab'.  
\*  
\*  
\*  append wa\_fieldcat to lt\_fieldcat.  
\*  clear wa\_fieldcat.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_vbe\_data  
\*&---------------------------------------------------------------------\*  
FORM get\_vbe\_data .  
  select \*  
    into table lt\_tab  
    from vbak  
    where vbeln in s\_vbeln.  
  
  if sy-subrc eq 0.  
    MESSAGE s000.  
  else.  
    MESSAGE e001.  
  endif.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form disp\_alv\_fieldcat  
\*&---------------------------------------------------------------------\*  
FORM disp\_alv\_fieldcat .  
 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                  = 'vbak'  
\*    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          ZNIDH\_ALV\_MERGE  
\*&---------------------------------------------------------------------\*  
  
data : lt\_fieldcat type slis\_t\_fieldcat\_alv,  
       wa\_fieldcat type slis\_fieldcat\_Alv,  
       lt\_tab type STANDARD TABLE OF vbak,  
       lv\_vbeln type vbeln,  
       ty\_layout type slis\_layout\_alv.

OUTPUT:



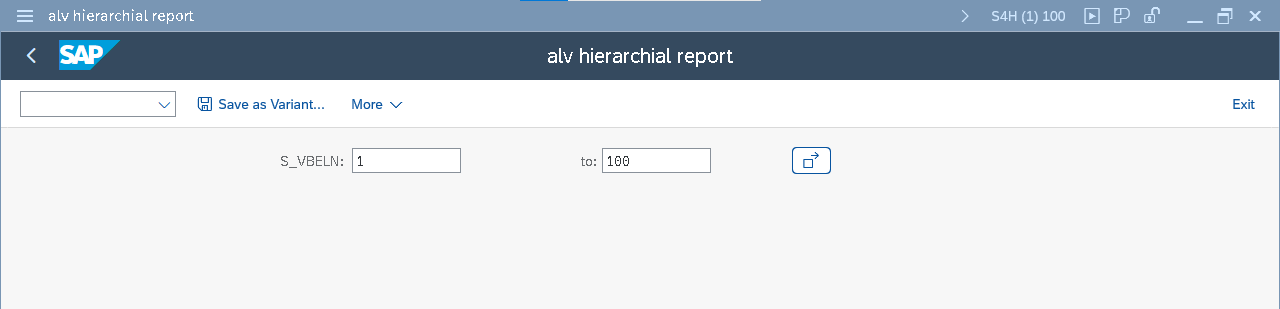


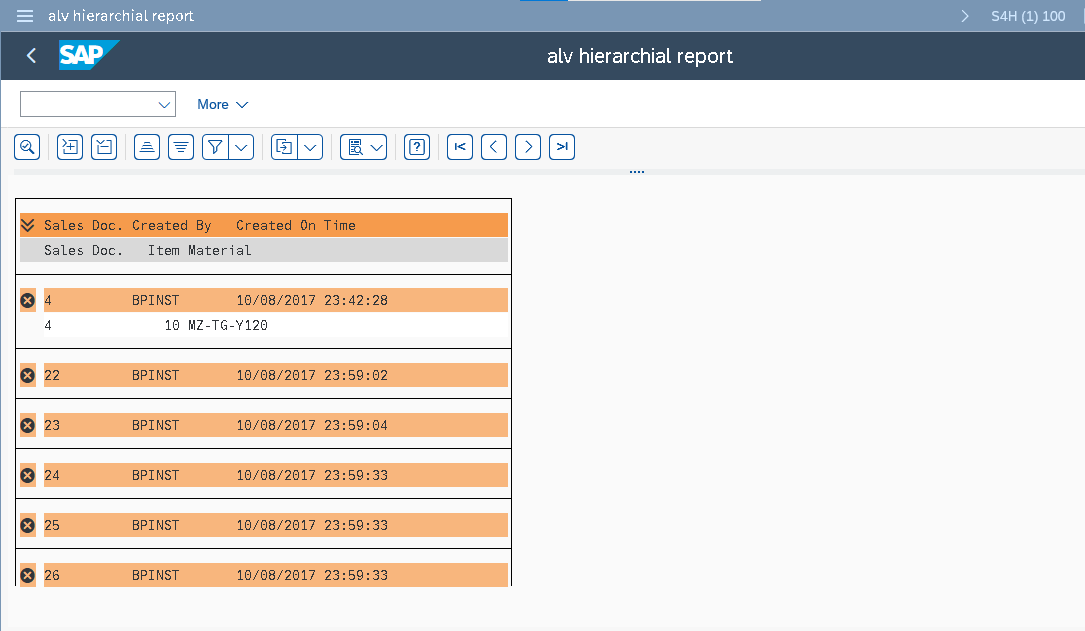
1. ALV REPORTS USING GRID POPUP

\*&---------------------------------------------------------------------\*  
\*& Report ZNIDH\_ALV\_GRID\_POPUP  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNIDH\_ALV\_GRID\_POPUP NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500 MESSAGE-ID ZNIDH\_MSG.  
  
INCLUDE ZNIDH\_ALV\_GRID\_MERGE.  
  
select-OPTIONS s\_vbeln for lv\_vbeln OBLIGATORY.  
  
INITIALIZATION.  
  
clear :  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 vbak  
    where vbeln in s\_vbeln.  
  
    if sy-subrc eq 0.  
      MESSAGE s000.  
    else.  
      MESSAGE e001.  
    endif.  
  
START-OF-SELECTION.  
    perform get\_alv\_fieldcat.  
    perform get\_vbe\_data.  
end-of-SELECTION.  
    perform disp\_alv\_fieldcat.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_alv\_fieldcat  
\*&---------------------------------------------------------------------\*  
FORM get\_alv\_fieldcat .  
  "VBELN  
  wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'Sales Document'.  
  wa\_fieldcat-fieldname = 'VBELN'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "ERNAM  
  wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'Name of the person'.  
  wa\_fieldcat-fieldname = 'ERNAM'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "VBTYP  
  wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'SD Document Type'.  
  wa\_fieldcat-fieldname = 'VBTYP'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "AUART  
  wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'Sales Document Category'.  
  wa\_fieldcat-fieldname = 'AUART'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "WAERK  
  wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = 'SD Document Currency'.  
  wa\_fieldcat-fieldname = 'WAERK'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  "VKORG  
  wa\_fieldcat-col\_pos = '6'.  
  wa\_fieldcat-seltext\_l = 'Sales Organization'.  
  wa\_fieldcat-fieldname = 'VKORG'.  
  wa\_fieldcat-tabname = 'lt\_tab'.  
  
  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form get\_vbe\_data  
\*&---------------------------------------------------------------------\*  
FORM get\_vbe\_data .  
  select vbeln ernam vbtyp auart waerk vkorg  
    into table lt\_tab  
    from vbak  
    where vbeln in s\_vbeln.  
  
  if sy-subrc eq 0.  
    MESSAGE s000.  
  else.  
    MESSAGE e001.  
  endif.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form disp\_alv\_fieldcat  
\*&---------------------------------------------------------------------\*  
  
FORM disp\_alv\_fieldcat .  
  CALL FUNCTION 'REUSE\_ALV\_POPUP\_TO\_SELECT'  
  EXPORTING  
   I\_TITLE                       = 'VBAK'  
\*   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.

7) ALV REPORT HIERARCHIAL

*\*&---------------------------------------------------------------------\**  
*\*& Report ZNIDH\_ALV\_HIERARCHIAL*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZNIDH\_ALV\_HIERARCHIAL.  
  
include ZNIDH\_ALV\_HIER\_INC.  
  
tables vbak.  
select-OPTIONS s\_vbeln for vbak-vbeln.  
  
START-OF-SELECTION.  
  
perform get\_data.  
perform create\_data.  
perform hierarchydisplay.  
  
FORM get\_data .  
  
  select vbeln ernam erdat erzet  
    from vbak  
    into CORRESPONDING FIELDS OF table lt\_tab1  
    where vbeln in s\_vbeln.  
  
  select vbeln posnr matnr  
    from vbap  
    into CORRESPONDING FIELDS OF table lt\_tab2  
    where vbeln in s\_vbeln.  
  
ENDFORM.  
  
FORM create\_data .  
  *"building fcat for vbak*  
  wa\_fcat-col\_pos = '1'.  
  wa\_fcat-fieldname = 'vbeln'.  
  wa\_fcat-tabname = 'lt\_tab1'.  
  wa\_fcat-ref\_tabname = 'vbak'.  
  wa\_fcat-ref\_fieldname = 'vbeln'.  
  append wa\_fcat to lt\_fcat.  
  clear wa\_fcat.  
  
  wa\_fcat-col\_pos = '2'.  
  wa\_fcat-fieldname = 'ernam'.  
  wa\_fcat-tabname = 'lt\_tab1'.  
  wa\_fcat-ref\_tabname = 'vbak'.  
  wa\_fcat-ref\_fieldname = 'ernam'.  
  append wa\_fcat to lt\_fcat.  
  clear wa\_fcat.  
  
  wa\_fcat-col\_pos = '3'.  
  wa\_fcat-fieldname = 'erdat'.  
  wa\_fcat-tabname = 'lt\_tab1'.  
  wa\_fcat-ref\_tabname = 'vbak'.  
  wa\_fcat-ref\_fieldname = 'erdat'.  
  append wa\_fcat to lt\_fcat.  
  clear wa\_fcat.  
  
  wa\_fcat-col\_pos = '4'.  
  wa\_fcat-fieldname = 'erzet'.  
  wa\_fcat-tabname = 'lt\_tab1'.  
  wa\_fcat-ref\_tabname = 'vbak'.  
  wa\_fcat-ref\_fieldname = 'erzet'.  
  append wa\_fcat to lt\_fcat.  
  clear wa\_fcat.  
  
  *"building fcat for vbap*  
  wa\_fcat-col\_pos = '1'.  
  wa\_fcat-fieldname = 'vbeln'.  
  wa\_fcat-tabname = 'lt\_tab2'.  
  wa\_fcat-ref\_tabname = 'vbap'.  
  wa\_fcat-ref\_fieldname = 'vbeln'.  
  append wa\_fcat to lt\_fcat.  
  clear wa\_fcat.  
  
  wa\_fcat-col\_pos = '2'.  
  wa\_fcat-fieldname = 'posnr'.  
  wa\_fcat-tabname = 'lt\_tab2'.  
  wa\_fcat-ref\_tabname = 'vbap'.  
  wa\_fcat-ref\_fieldname = 'posnr'.  
  append wa\_fcat to lt\_fcat.  
  clear wa\_fcat.  
  
  wa\_fcat-col\_pos = '3'.  
  wa\_fcat-fieldname = 'matnr'.  
  wa\_fcat-tabname = 'lt\_tab2'.  
  wa\_fcat-ref\_tabname = 'vbap'.  
  wa\_fcat-ref\_fieldname = 'matnr'.  
  append wa\_fcat to lt\_fcat.  
  clear wa\_fcat.  
  
ENDFORM.  
  
FORM hierarchydisplay .  
  if sy-subrc = 0.  
  lt\_key-header01 = 'vbeln'.  
  layout-expand\_fieldname = 'UST'.  
  
  CALL FUNCTION 'REUSE\_ALV\_HIERSEQ\_LIST\_DISPLAY'  
    EXPORTING  
  
      I\_CALLBACK\_PROGRAM             = sy-repid  
      IS\_LAYOUT                      = layout  
      IT\_FIELDCAT                    = lt\_fcat[]  
      i\_tabname\_header               = 'lt\_tab1'  
      i\_tabname\_item                 = 'lt\_tab2'  
      is\_keyinfo                     = lt\_key  
  
    TABLES  
      t\_outtab\_header                = lt\_tab1  
      t\_outtab\_item                  = lt\_tab2  
   EXCEPTIONS  
     PROGRAM\_ERROR                  = 1  
     OTHERS                         = 2  
            .  
  IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
  ENDIF.  
  else.  
    write :/ 'No Records Found'.  
  endif.  
  
  
ENDFORM.





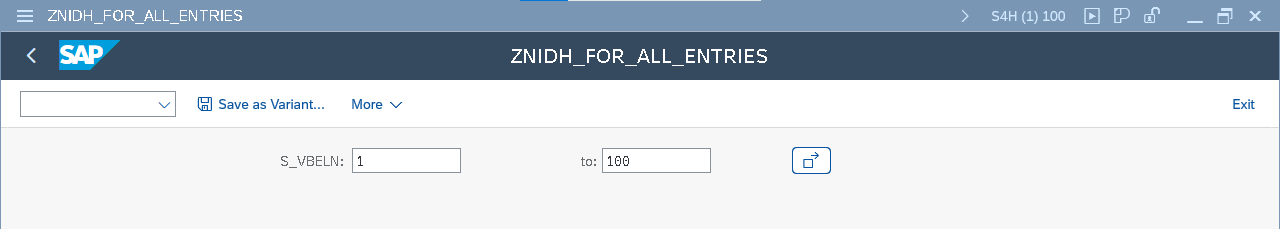
FOR ALL ENTRIES

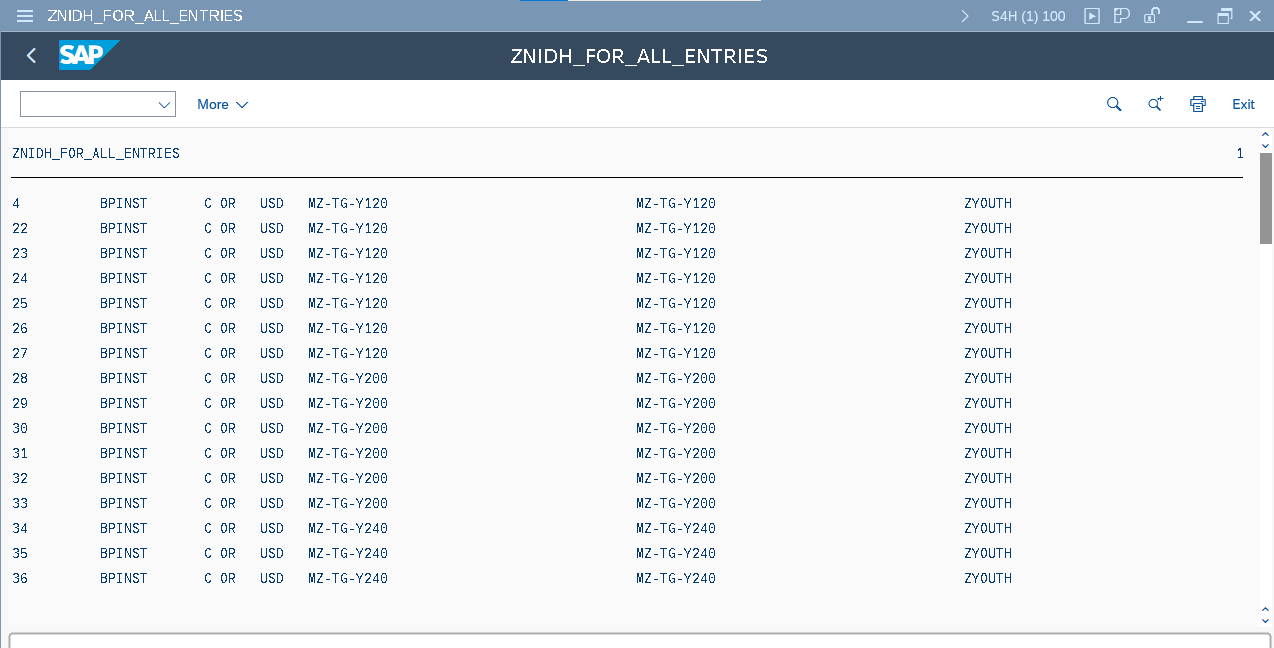
\*&---------------------------------------------------------------------\*  
\*& Report ZNIDH\_FOR\_ALL\_ENTRIES  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNIDH\_FOR\_ALL\_ENTRIES.  
  
INCLUDE ZNIDH\_FOR\_ALL\_ENTRIES\_INC.  
  
Tables vbak.  
  
select-options s\_vbeln for vbak-vbeln.  
  
START-OF-SELECTION.  
select vbeln ernam vbtyp auart waerk  
  into table lt\_tab1  
  from vbak  
  where vbeln in s\_vbeln.  
  
if not lt\_tab1[] is initial.  
  
 select vbeln matnr matwa matkl  
      into table lt\_tab2  
      from vbap  
      FOR ALL ENTRIES IN lt\_tab1  
      where vbeln eq lt\_tab1-vbeln.  
  
  delete ADJACENT DUPLICATES FROM lt\_tab1 COMPARING vbeln.  
  
 sort lt\_tab2 by vbeln.  
  
 loop at lt\_tab1 into wa\_tab1.  
  
read table lt\_tab2 into wa\_tab2 with key vbeln = wa\_tab1-vbeln BINARY SEARCH.  
  
 if sy-subrc = 0.  
  move: wa\_tab1-vbeln to wa\_tab3-vbeln,  
        wa\_tab1-ernam to wa\_tab3-ernam,  
        wa\_tab1-vbtyp to wa\_tab3-vbtyp,  
        wa\_tab1-auart to wa\_tab3-auart,  
        wa\_tab1-waerk to wa\_tab3-waerk,  
  
  
        wa\_tab2-matnr to wa\_tab3-matnr,  
        wa\_tab2-matwa to wa\_tab3-matwa,  
        wa\_tab2-matkl to wa\_tab3-matkl.  
  
  append wa\_tab3 to lt\_tab3.  
  
 else.  
    continue.  
 endif.  
  endloop.  
  
  
  
  loop at lt\_tab3 into wa\_tab3.  
    write : / wa\_tab3-vbeln,  
              wa\_tab3-ernam,  
              wa\_tab3-vbtyp,  
              wa\_tab3-auart,  
              wa\_tab3-waerk,  
  
              wa\_tab3-matnr,  
              wa\_tab3-matwa,  
              wa\_tab3-matkl.  
  
  endloop.  
  
  else.  
    write 'No records found'.  
  
  endif.

INCLUDE:

\*&---------------------------------------------------------------------\*  
\*& Include          ZNIDH\_FOR\_ALL\_ENTRIES\_INC  
\*&---------------------------------------------------------------------\*  
  
types :  begin of ty\_tab1,  
         vbeln type vbeln\_va,  "Sales document  
         ernam type ernam, "NAme of the person created the object  
         vbtyp type vbtyp, "SD Document Category  
         auart type auart, "Sales Document Type  
         waerk type waerk, "SD Document Currency  
         vkorg type vkorg, "Sales Organization  
         end of ty\_tab1.  
  
data : lt\_tab1 type STANDARD TABLE OF ty\_tab1,  
       wa\_tab1 type ty\_tab1.  
  
  
types :  begin of ty\_tab2,  
         vbeln type vbeln\_va,  "Sales document  
         matnr type matnr, "Material Number  
         matwa type matwa, "Material Entered  
         matkl type matkl, "Material group  
         end of ty\_tab2.  
  
data : lt\_tab2 type STANDARD TABLE OF ty\_tab2,  
       wa\_tab2 type ty\_tab2.  
  
  
types :  begin of ty\_tab3,  "final strs  
         vbeln type vbeln\_va,  "Sales document  
         ernam type ernam, "NAme of the person created the object  
         vbtyp type vbtyp, "SD Document Category  
         auart type auart, "Sales Document Type  
         waerk type waerk, "SD Document Currency  
         vkorg type vkorg, "Sales Organization  
  
         matnr type matnr, "Material Number  
         matwa type matwa, "Material Entered  
         matkl type matkl, "Material group  
  
         end of ty\_tab3.  
  
data : lt\_tab3 type STANDARD TABLE OF ty\_tab3,  
       wa\_tab3 type ty\_tab3.

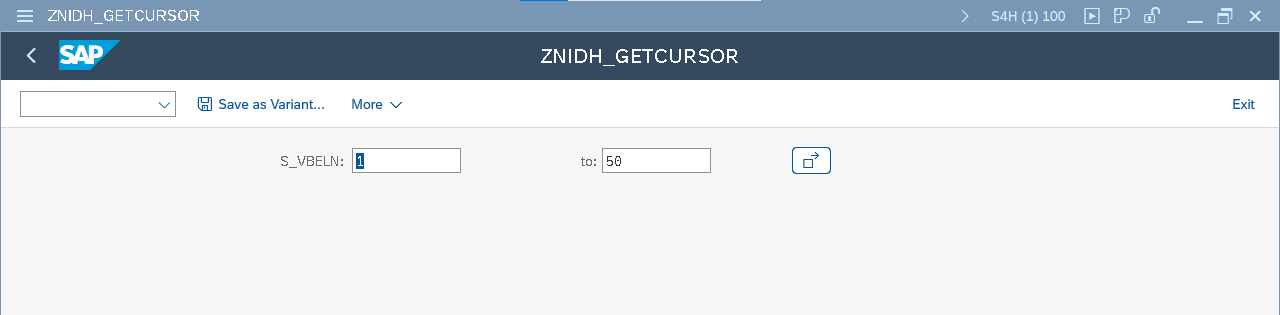
OUTPUT:

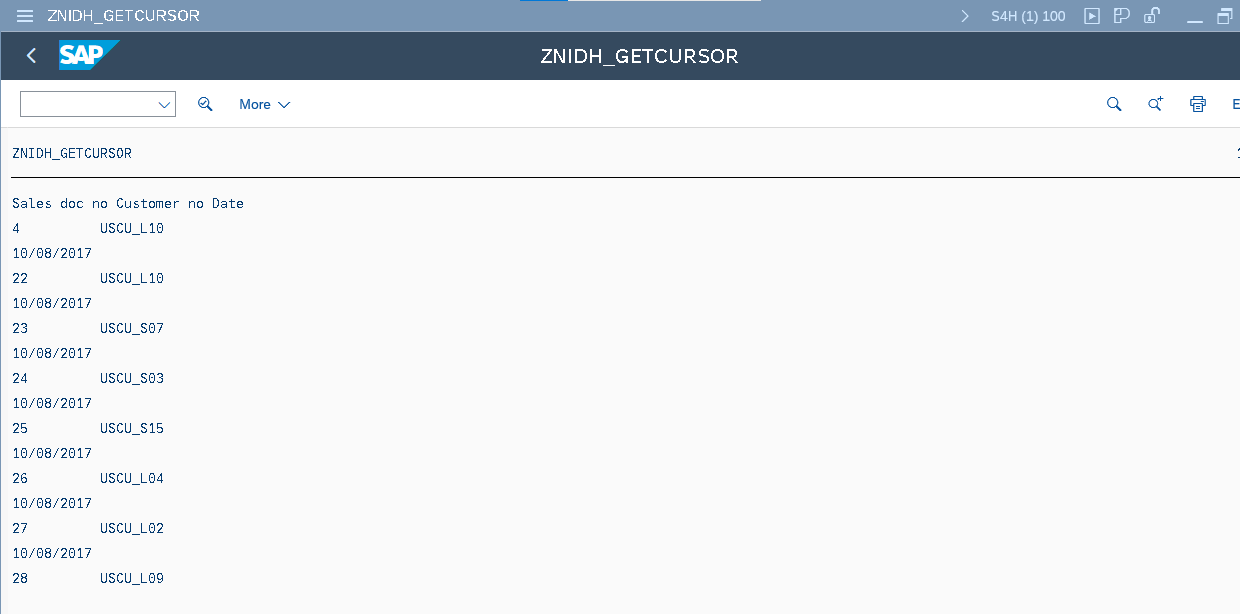




GET CURSOR METHOD

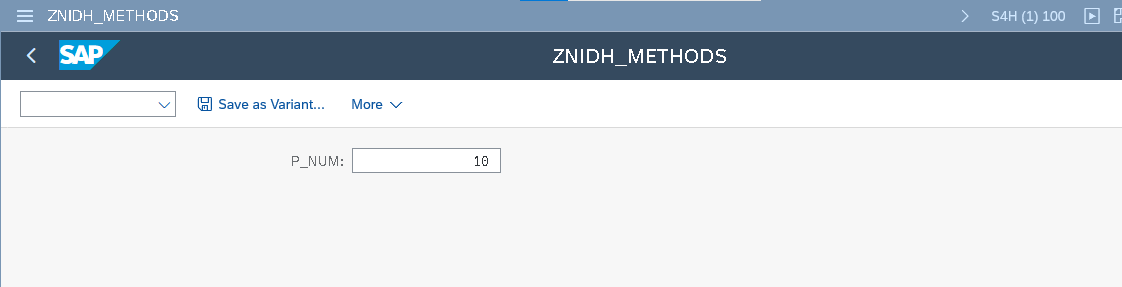
*\*&---------------------------------------------------------------------\**  
*\*& Report ZNIDH\_GETCURSOR*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZNIDH\_GETCURSOR.  
  
*\*\*\*\*\*\*\*\*\*    1   \*\*\*\*\*\*\*\*\*\*\**  
  
*\*data: var1(10) type c value 'Field 1',*  
*\*      var2(10) type c value 'Field 2',*  
*\*      var3(10) type c value 'Field 3',*  
*\*      fname(10) type c.*  
*\**  
*\*START-OF-SELECTION.*  
*\*FORMAT color 7.*  
*\*write :/ 'Basic List'.*  
*\**  
*\*"As we have seen in the interactive reports that even if we click*  
*\*" any field we were redirecting to the next list(secondary list)*  
*\*"to avoid that we are turning the hotspot off*  
*\**  
*\*format hotspot on.*  
*\*write :/ var1, var2, var3.*  
*\*format HOTSPOT off.*  
*\**  
*\*skip 2.*  
*\*write :/ 'UST SAP ABAP'.*  
*\**  
*\*at line-selection.*  
*\*  get CURSOR field fname.*  
*\**  
*\*if fname = 'var1'.*  
*\*  write :/ 'User interacted on Field1'.*  
*\*elseif fname = 'var2'.*  
*\*  write :/ 'User interacted on Field2'.*  
*\*elseif fname = 'var3'.*  
*\*  write :/ 'User interacted on Field3'.*  
*\*endif.*  
  
  
*\*\*\*\*\*\*\*\*      2     \*\*\*\*\*\*\*\**  
  
data lv\_vbeln type vbeln\_va.  
  
types : begin of ty\_tab,   *"vbak table*  
  vbeln type vbeln\_va,  
  kunnr type kunnr,  
  erdat type erdat,  
  end of ty\_tab.  
  
data : lt\_tab TYPE table of ty\_tab,  
      wa\_tab type ty\_tab.  
  
select-options s\_vbeln for lv\_vbeln.  
  
START-OF-SELECTION.  
select vbeln kunnr erdat  
  into table lt\_tab  
  from vbak  
  where vbeln in s\_vbeln.  
  
end-of-selection.  
  
loop at lt\_tab into wa\_tab.  
  
  format hotspot on.  
  write :/ wa\_tab-vbeln,  
           wa\_tab-kunnr.  
  format hotspot off.  
  write :/ wa\_tab-erdat.  
  
endloop.  
  
top-of-page.  
write :/ 'Sales doc no',  
         'Customer no',  
         'Date'.  
  
AT LINE-SELECTION.  
  
  data : fnam(15), fval(15).  
  
  GET CURSOR field fnam value fval.  *"vbeln 5000*  
  if fnam = 'wa\_tab-vbeln'.  
    set PARAMETER ID 'AUN' field fval.  
    call TRANSACTION 'VA03' and skip first screen.  
  
  elseif fnam = 'wa\_tab-kunnr'.   *"kunnr 123*  
    set PARAMETER ID 'KUN' field fval.  
    call transaction 'XD02' and skip FIRST SCREEN.  
  endif.

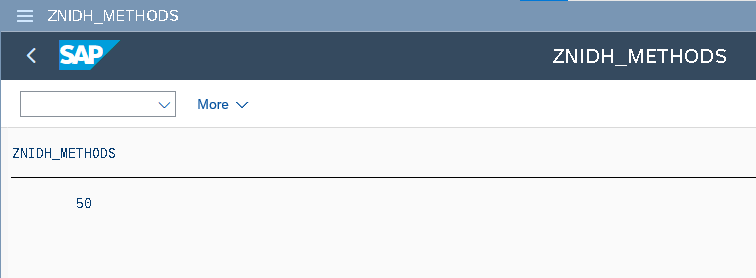




METHODS

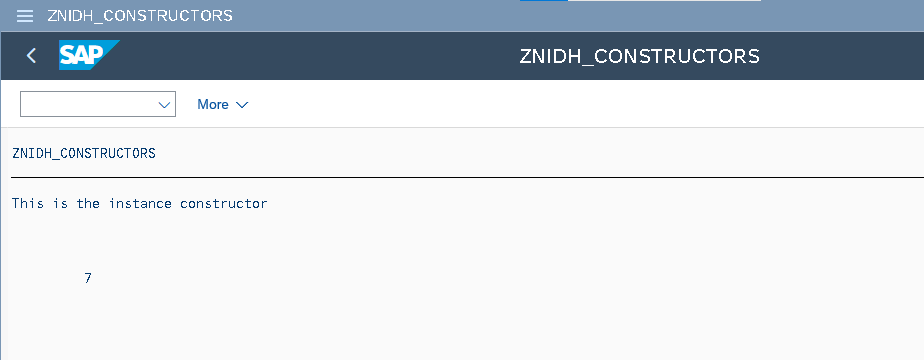
*\*&---------------------------------------------------------------------\**  
*\*& Report ZNIDH\_METHODS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZNIDH\_METHODS.  
  
*\*\*\*\*\*\*\*\*\*\*\*  1  USE OF THE STATIC ATTRIBUTES  \*\*\*\*\*\*\*\*\*\*\**  
*\*class c1 DEFINITION.*  
*\*  PUBLIC SECTION.*  
*\*  class-DATA num type i.   "static attribute*  
*\*  METHODS M1.*  
*\*ENDCLASS.*  
*\**  
*\*class c1 IMPLEMENTATION.*  
*\*  METHOD m1.*  
*\*  num = num + 1.*  
*\*  write num.*  
*\*  endmethod.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\**  
*\*c1=>num = 3.*  
*\*write c1=>num.*  
*\*data : obj1 type REF TO c1,*  
*\*      obj2 type REF TO c1.*  
*\*create object: obj1, obj2.*  
*\**  
*\*call METHOD: obj1->m1, obj2->m1.*  
  
  
*\*\*\*\*\*\*\*\* 2  different ways for calling method using one import parameter*  
*\*class c1 DEFINITION.*  
*\*  PUBLIC SECTION.*  
*\*  data num type i value 10.*  
*\*  METHODS M1 importing i\_value type i.*  
*\*ENDCLASS.*  
*\**  
*\*class c1 IMPLEMENTATION.*  
*\*  METHOD M1.*  
*\*    num = num \* i\_value.*  
*\*    write / num.*  
*\*    num = 10.*  
*\*  ENDMETHOD.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\**  
*\*data obj1 type REF TO c1.*  
*\*create OBJECT obj1.*  
*\**  
*\*call method obj1->m1 EXPORTING i\_value = 5.*  
*\*call method obj1->m1( i\_value = 6 ).*  
*\*call method obj1->m1( 7 ).*  
*\*obj1->m1( 8 ).*  
  
  
*\*\*\*\*\*\*\* 3 import parameters passed by reference, can't change inside the method*  
  
*\*parameters p\_num type i.*  
*\*class c1 DEFINITION.*  
*\*  PUBLIC SECTION.*  
*\**  
*\*  data num type i.*  
*\*  methods m1 IMPORTING input\_a type i*  
*\*            value(input\_b) type i.*  
*\**  
*\*endclass.*  
*\**  
*\*class c1 IMPLEMENTATION.*  
*\*  method m1.*  
*\*    input\_b = 6.*  
*\*    write / input\_b.*  
*\*  ENDMETHOD.*  
*\*endclass.*  
*\**  
*\*START-OF-SELECTION.*  
*\*data obj1 TYPE REF TO c1.*  
*\*create object obj1.*  
*\**  
*\*call method: obj1->m1 EXPORTING input\_a = 2 input\_b = p\_num.*  
  
  
*\*\*\*\*\*\* 4 export and change parameters*  
  
*\*PARAMETERS p\_a(10) type c.*  
*\**  
*\*data : lv\_tax type p DECIMALS 2,*  
*\*       lv\_sal type p DECIMALS 2.*  
*\**  
*\*class c1 DEFINITION.*  
*\*  PUBLIC SECTION.*  
*\*  methods m1: IMPORTING grade type c*  
*\*             EXPORTING tax type p*  
*\*             changing sal type p.*  
*\*ENDCLASS.*  
*\**  
*\*class c1 IMPLEMENTATION.*  
*\*  METHOD m1.*  
*\*    CASE grade.*  
*\*      WHEN 'A01'.*  
*\*        tax = sal \* '0.2'.*  
*\*      WHEN 'A02'.*  
*\*        tax = sal \* '0.1'.*  
*\*      WHEN OTHERS.*  
*\*        tax = sal \* '0.15'.*  
*\*    ENDCASE.*  
*\**  
*\*    sal = sal - tax.*  
*\*  ENDMETHOD.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\*data obj1 type REF TO c1.*  
*\*create object obj1.*  
*\**  
*\*lv\_tax = 0.*  
*\*lv\_sal = 19000.*  
*\**  
*\*write:/ 'Before calling method the salary and the tax are :',lv\_sal,lv\_tax.*  
*\**  
*\*call method obj1->m1 exporting grade = p\_a*  
*\*                     importing tax = lv\_tax*  
*\*                     changing sal = lv\_sal.*  
*\**  
*\*write :/ 'After calling method the salary and the tax are :',lv\_sal,lv\_tax.*  
  
  
*\*\*\*\*\*\*\*\*\*\*\* 5 use of returning parameters in method(functional method)*  
  
*\*data num type i.*  
*\**  
*\*class c1 DEFINITION.*  
*\*  PUBLIC SECTION.*  
*\*  methods m1 importing input1 type i*  
*\*                       input2 type i*  
*\*             RETURNING VALUE(result) type i.*  
*\*ENDCLASS.*  
*\**  
*\*class c1 IMPLEMENTATION.*  
*\*  method m1.*  
*\*  result = input1 \* 2 - input2.*  
*\*  ENDMETHOD.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\*data obj1 TYPE REF TO c1.*  
*\*create object obj1.*  
*\**  
*\*call method: obj1->m1 EXPORTING input1 = 2*  
*\*                                input2 = 3*  
*\*                      RECEIVING result = num. "1 syntax*  
*\*write num.*  
*\**  
*\*num = obj1->m1( input1 = 3  input2 = 4 ). "2 syntax*  
*\*write / num.*  
*\**  
*\*move obj1->m1( input1 = 4  input2 = 5 ) to num. "3 syntax*  
*\*write / num.*  
  
  
  
*\*\*\*\*\*\*\*\*\*\*  6 static methods can call only static attributes. instance can call both*  
  
*\*class c1 DEFINITION.*  
*\*  PUBLIC SECTION.*  
*\*  class-data: stnum type i value 5.*  
*\*  data : instnum type i value 6.*  
*\**  
*\*  class-methods: stmeth.*  
*\*  methods: instmeth.*  
*\*ENDCLASS.*  
*\**  
*\**  
*\*class c1 IMPLEMENTATION.*  
*\*  method stmeth.*  
*\*    write :/ stnum. "instnum.*  
*\*  ENDMETHOD.*  
*\*  method instmeth.*  
*\*    write :/ stnum, instnum.*  
*\*  ENDMETHOD.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\*data obj1 TYPE REF TO c1.*  
*\*create OBJECT obj1.*  
*\**  
*\*call method c1=>stmeth.*  
*\*call method obj1->instmeth.*  
  
  
*\*\*\*\*\*\*\*\*\* 7 message exceptions*  
  
class c1 DEFINITION.  
PUBLIC SECTION.  
methods m1 : importing a type i  
             EXPORTING b type i  
  
             EXCEPTIONS e1.  
endclass.  
  
  
class c1 IMPLEMENTATION.  
  method m1.  
    if a lt 5.  
      message i398(00) with 'should>=5' RAISING e1.  
    else.  
      b = a \* 5.  
    endif.  
  ENDMETHOD.  
endclass.  
  
parameters p\_num type i.  
  
START-OF-SELECTION.  
data obj1 TYPE REF TO c1.  
create OBJECT obj1.  
  
call method obj1->m1 EXPORTING a = p\_num  
                IMPORTING b = p\_num  
                EXCEPTIONS e1 = 4.  
  
if sy-subrc = 4.  
  MESSAGE id sy-msgid type sy-msgty number sy-msgno  
  WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.  
else.  
  write :/ p\_num.  
  
endif.





CONSTRUCTORS

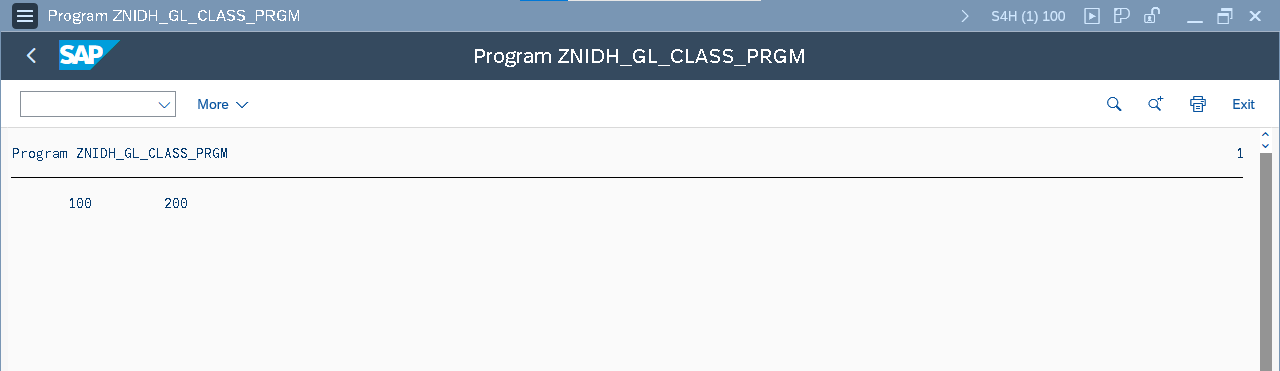
*\*&---------------------------------------------------------------------\**  
*\*& Report ZNIDH\_CONSTRUCTORS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZNIDH\_CONSTRUCTORS.  
  
*\*\*\*\*\*\*\*\*\*\*\*     1 Instance constructor called when object is created    \*\*\*\*\*\*\*\*\*\*\**  
  
*\*class c1 DEFINITION.*  
*\*  PUBLIC SECTION.*  
*\*  methods: constructor.*  
*\*ENDCLASS.*  
*\**  
*\*class c1 IMPLEMENTATION.*  
*\*  method : constructor.*  
*\*  write : / 'I am the calling constructor'.*  
*\*  skip 1.*  
*\*  endmethod.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\*data obj1 TYPE REF TO c1.*  
*\*create OBJECT obj1.*  
  
*\*\*\*\*call METHOD . "no need for calling the method for constructor as the instance constructor*  
                  *"is invoked automatically when we create the object*  
  
  
  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*  2 Instance constructor can have import parameters    \*\*\*\*\*\*\*\*\*\*\**  
  
*\*class c1 DEFINITION.*  
*\*  PUBLIC SECTION.*  
*\*  methods: constructor importing i\_today type d.*  
*\*ENDCLASS.*  
*\**  
*\*class c1 IMPLEMENTATION.*  
*\*  method constructor.*  
*\*  write:/ 'Today, date is',i\_today DD/MM/YYYY.*  
*\*  endmethod.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\*data obj1 type REF TO c1.*  
*\*create object obj1 EXPORTING i\_today = sy-datum .*  
  
  
  
  
*\*\*\*\*\*\*\*\*\*  3 Instance Constructors can raise exceptions   \*\*\*\*\*\*\*\*\*\*\**  
  
*\*class c1 DEFINITION.*  
*\* public SECTION.*  
*\* methods constructor importing num type i*  
*\*                                   exceptions e1.*  
*\*ENDCLASS.*  
*\**  
*\*class c1 IMPLEMENTATION.*  
*\*  method constructor.*  
*\*    if num lt 10.*  
*\*      raise e1.*  
*\*    else.*  
*\*      exit.*  
*\*    endif.*  
*\*  endmethod.*  
*\*endclass.*  
*\**  
*\*PARAMETERS: p\_num type i.*  
*\**  
*\*START-OF-SELECTION.*  
*\*data obj1 TYPE REF TO c1.*  
*\*create object obj1 exporting num = p\_num*  
*\*                             EXCEPTIONS e1 = 4.*  
*\**  
*\*if sy-subrc = 4.*  
*\*  write :/ 'Cannot take the number less than 10'.*  
*\*else.*  
*\*  exit.*  
*\*endif.*  
  
  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*   4 Static constructor*  
  
class c1 DEFINITION.  
  PUBLIC SECTION.  
  class-data num type i value 7.  
  
  methods : constructor.  
  class-METHODS : st\_constructor.  
  
ENDCLASS.  
  
class c1 IMPLEMENTATION.  
  
  method  constructor.  
    write :/ 'This is the instance constructor'.  
    skip 2.  
  ENDMETHOD.  
  
  
  
  method : st\_constructor.  
    write :/ 'This is the static constructor'.  
  ENDMETHOD.  
  
ENDCLASS.  
  
START-OF-SELECTION.  
data obj1 type ref to c1.  
create object obj1.  
  
write c1=>num.



GL CLASS PROGRAMS

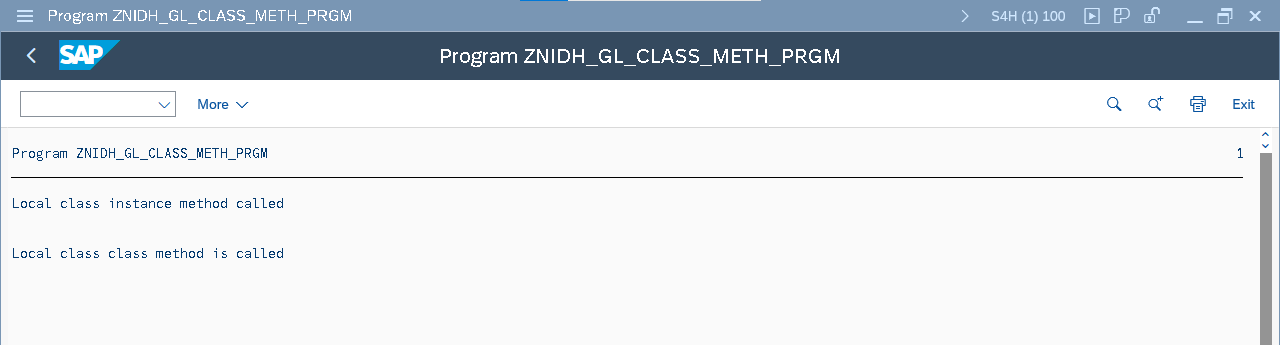
1)

\*&---------------------------------------------------------------------\*  
\*& Report ZNIDH\_GL\_CLASS\_PRGM  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNIDH\_GL\_CLASS\_PRGM.  
  
\*data i\_obj type REF TO ZCL\_NIDH\_GL\_CLASS.  
\*create OBJECT i\_obj.  
\*  
\*START-OF-SELECTION.  
\*write :/ i\_obj->I\_A,  
\*         ZCL\_NIDH\_GL\_CLASS=>S\_B.  
  
  
\*\*\*\*\*\*\*\*  USING LOCAL CLASS  \*\*\*\*\*\*\*\*  
class CL\_LOCAL DEFINITION.  
  PUBLIC SECTION.  
  data I\_A type i value 100.  "assigning value to instance attribute  
  class-data S\_B type i value 200.  "assigning value to static attribute  
  ENDCLASS.  
  
data i\_obj type REF TO CL\_LOCAL.  "declaring object  
  create object i\_obj.  "creating object  
  
  START-OF-SELECTION.  
  write :/ i\_obj->I\_A,  "calling instance attribute through object  
           CL\_LOCAL=>S\_B. "calling static attribute through class



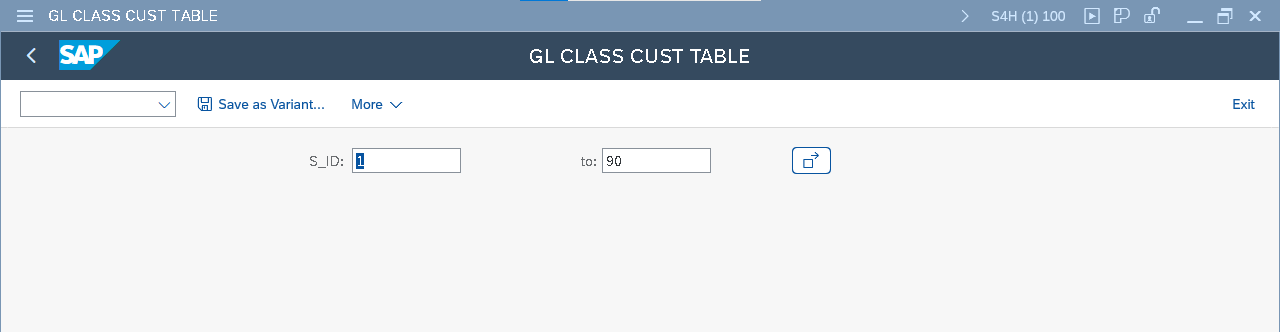
2)

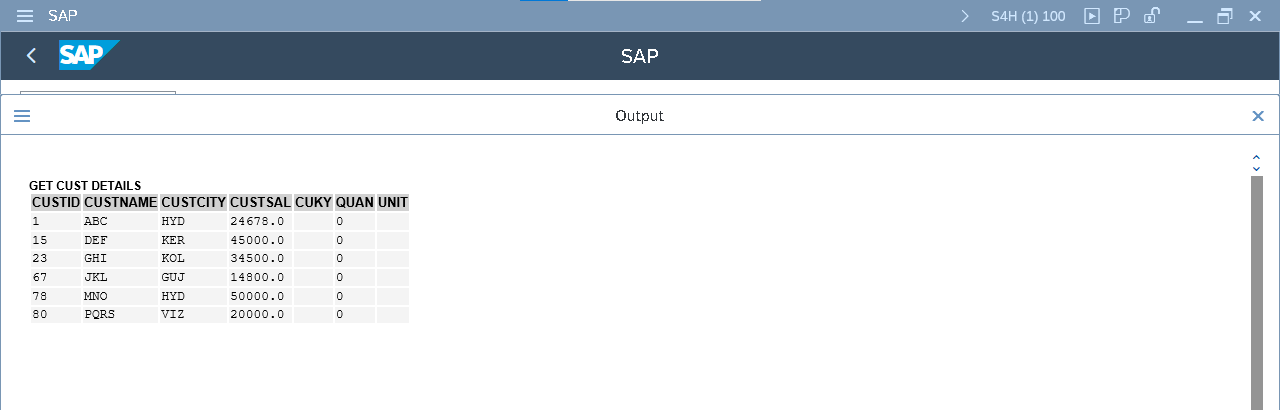
\*&---------------------------------------------------------------------\*  
\*& Report ZNIDH\_GL\_CLASS\_METH\_PRGM  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNIDH\_GL\_CLASS\_METH\_PRGM.  
  
\*data i\_obj TYPE REF TO ZCL\_NIDH\_GL\_CLASS\_METHOD. "declaring object  
\*create object i\_obj.  "creating object  
\*  
\*START-OF-SELECTION.  
\*call method i\_obj->i\_meth1.  
\*skip 1.  
\*call method ZCL\_NIDH\_GL\_CLASS\_METHOD=>s\_meth2.  
  
\*i\_obj->i\_meth1( ).  
\*skip 1.  
\*ZCL\_NIDH\_GL\_CLASS\_METHOD=>s\_meth2( ).  
  
  
class CL\_LOCALCLASS DEFINITION.  
  public section.  
  methods I\_METH1.  
  class-methods S\_METH2.  
  ENDCLASS.  
  
class CL\_LOCALCLASS IMPLEMENTATION.  
  METHOD I\_METH1.  
      write 'Local class instance method called'.  
    ENDMETHOD.  
  
  METHOD S\_METH2.  
    write 'Local class class method is called'.  
    ENDMETHOD.  
  ENDCLASS.  
  
  START-OF-SELECTION.  
  data i\_obj type REF TO CL\_LOCALCLASS.  
  create OBJECT i\_obj.  
  
  i\_obj->I\_METH1( ).  
  skip 1.  
  CL\_LOCALCLASS=>S\_METH2( ).



3)

\*&---------------------------------------------------------------------\*  
\*& Report ZNIDH\_GL\_CLASS\_CUST\_TABLE  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNIDH\_GL\_CLASS\_CUST\_TABLE.  
  
data lt\_tab type ZNIDH\_CUST\_TT.  
  
data i\_obj type REF TO ZCL\_NIDH\_GL\_CUST\_METHOD.  
create OBJECT i\_obj.  
  
data lv\_id type custid.  
select-options S\_id for lv\_id.  
  
START-OF-SELECTION.  
i\_obj->get\_cust\_method(  
  EXPORTING  
    i\_custid1        = s\_id-low  
    i\_custid2        = s\_id-high  
  IMPORTING  
    e\_tab            = lt\_tab )  
\*  EXCEPTIONS  
\*    custid\_not\_found = 1  
\*    others           = 2  
        .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
  
CL\_DEMO\_OUTPUT=>DISPLAY(  
DATA = LT\_TAB  
NAME = 'GET CUST DETAILS'  
).





OOPS METHODS

\*&---------------------------------------------------------------------\*  
\*& Report ZNIDH\_METHODS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZNIDH\_METHODS.  
  
\*\*\*\*\*\*\*\*\*\*\*  1  USE OF THE STATIC ATTRIBUTES  \*\*\*\*\*\*\*\*\*\*\*  
\*class c1 DEFINITION.  
\*  PUBLIC SECTION.  
\*  class-DATA num type i.   "static attribute  
\*  METHODS M1.  
\*ENDCLASS.  
\*  
\*class c1 IMPLEMENTATION.  
\*  METHOD m1.  
\*  num = num + 1.  
\*  write num.  
\*  endmethod.  
\*ENDCLASS.  
\*  
\*START-OF-SELECTION.  
\*  
\*c1=>num = 3.  
\*write c1=>num.  
\*data : obj1 type REF TO c1,  
\*      obj2 type REF TO c1.  
\*create object: obj1, obj2.  
\*  
\*call METHOD: obj1->m1, obj2->m1.  
  
  
\*\*\*\*\*\*\*\* 2  different ways for calling method using one import parameter  
\*class c1 DEFINITION.  
\*  PUBLIC SECTION.  
\*  data num type i value 10.  
\*  METHODS M1 importing i\_value type i.  
\*ENDCLASS.  
\*  
\*class c1 IMPLEMENTATION.  
\*  METHOD M1.  
\*    num = num \* i\_value.  
\*    write / num.  
\*    num = 10.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*START-OF-SELECTION.  
\*  
\*data obj1 type REF TO c1.  
\*create OBJECT obj1.  
\*  
\*call method obj1->m1 EXPORTING i\_value = 5.  
\*call method obj1->m1( i\_value = 6 ).  
\*call method obj1->m1( 7 ).  
\*obj1->m1( 8 ).  
  
  
\*\*\*\*\*\*\* 3 import parameters passed by reference, can't change inside the method  
  
\*parameters p\_num type i.  
\*class c1 DEFINITION.  
\*  PUBLIC SECTION.  
\*  
\*  data num type i.  
\*  methods m1 IMPORTING input\_a type i  
\*            value(input\_b) type i.  
\*  
\*endclass.  
\*  
\*class c1 IMPLEMENTATION.  
\*  method m1.  
\*    input\_b = 6.  
\*    write / input\_b.  
\*  ENDMETHOD.  
\*endclass.  
\*  
\*START-OF-SELECTION.  
\*data obj1 TYPE REF TO c1.  
\*create object obj1.  
\*  
\*call method: obj1->m1 EXPORTING input\_a = 2 input\_b = p\_num.  
  
  
\*\*\*\*\*\* 4 export and change parameters  
  
\*PARAMETERS p\_a(10) type c.  
\*  
\*data : lv\_tax type p DECIMALS 2,  
\*       lv\_sal type p DECIMALS 2.  
\*  
\*class c1 DEFINITION.  
\*  PUBLIC SECTION.  
\*  methods m1: IMPORTING grade type c  
\*             EXPORTING tax type p  
\*             changing sal type p.  
\*ENDCLASS.  
\*  
\*class c1 IMPLEMENTATION.  
\*  METHOD m1.  
\*    CASE grade.  
\*      WHEN 'A01'.  
\*        tax = sal \* '0.2'.  
\*      WHEN 'A02'.  
\*        tax = sal \* '0.1'.  
\*      WHEN OTHERS.  
\*        tax = sal \* '0.15'.  
\*    ENDCASE.  
\*  
\*    sal = sal - tax.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*START-OF-SELECTION.  
\*data obj1 type REF TO c1.  
\*create object obj1.  
\*  
\*lv\_tax = 0.  
\*lv\_sal = 19000.  
\*  
\*write:/ 'Before calling method the salary and the tax are :',lv\_sal,lv\_tax.  
\*  
\*call method obj1->m1 exporting grade = p\_a  
\*                     importing tax = lv\_tax  
\*                     changing sal = lv\_sal.  
\*  
\*write :/ 'After calling method the salary and the tax are :',lv\_sal,lv\_tax.  
  
  
\*\*\*\*\*\*\*\*\*\*\* 5 use of returning parameters in method(functional method)  
  
\*data num type i.  
\*  
\*class c1 DEFINITION.  
\*  PUBLIC SECTION.  
\*  methods m1 importing input1 type i  
\*                       input2 type i  
\*             RETURNING VALUE(result) type i.  
\*ENDCLASS.  
\*  
\*class c1 IMPLEMENTATION.  
\*  method m1.  
\*  result = input1 \* 2 - input2.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*START-OF-SELECTION.  
\*data obj1 TYPE REF TO c1.  
\*create object obj1.  
\*  
\*call method: obj1->m1 EXPORTING input1 = 2  
\*                                input2 = 3  
\*                      RECEIVING result = num. "1 syntax  
\*write num.  
\*  
\*num = obj1->m1( input1 = 3  input2 = 4 ). "2 syntax  
\*write / num.  
\*  
\*move obj1->m1( input1 = 4  input2 = 5 ) to num. "3 syntax  
\*write / num.  
  
  
  
\*\*\*\*\*\*\*\*\*\*  6 static methods can call only static attributes. instance can call both  
  
\*class c1 DEFINITION.  
\*  PUBLIC SECTION.  
\*  class-data: stnum type i value 5.  
\*  data : instnum type i value 6.  
\*  
\*  class-methods: stmeth.  
\*  methods: instmeth.  
\*ENDCLASS.  
\*  
\*  
\*class c1 IMPLEMENTATION.  
\*  method stmeth.  
\*    write :/ stnum. "instnum.  
\*  ENDMETHOD.  
\*  method instmeth.  
\*    write :/ stnum, instnum.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*START-OF-SELECTION.  
\*data obj1 TYPE REF TO c1.  
\*create OBJECT obj1.  
\*  
\*call method c1=>stmeth.  
\*call method obj1->instmeth.  
  
  
\*\*\*\*\*\*\*\*\* 7 message exceptions  
  
class c1 DEFINITION.  
PUBLIC SECTION.  
methods m1 : importing a type i  
             EXPORTING b type i  
  
             EXCEPTIONS e1.  
endclass.  
  
  
class c1 IMPLEMENTATION.  
  method m1.  
    if a lt 5.  
      message i398(00) with 'should>=5' RAISING e1.  
    else.  
      b = a \* 5.  
    endif.  
  ENDMETHOD.  
endclass.  
  
parameters p\_num type i.  
  
START-OF-SELECTION.  
data obj1 TYPE REF TO c1.  
create OBJECT obj1.  
  
call method obj1->m1 EXPORTING a = p\_num  
                IMPORTING b = p\_num  
                EXCEPTIONS e1 = 4.  
  
if sy-subrc = 4.  
  MESSAGE id sy-msgid type sy-msgty number sy-msgno  
  WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.  
else.  
  write :/ p\_num.  
  
endif.

