SALES_REGISTER_REPORT.

To Create an Alv Interactive report.

Goto SE38.

Give the Program name and Shor Description.

Give the Type as Executable program.

Click on Save.

Write logic for Basic as well as Interactive list.

```
*&----------------------------------------------------------------------*
*& Report   ZSD_SALES_REGISTER_REPORT
*&
*&----------------------------------------------------------------------*
*&
*&
*&----------------------------------------------------------------------*

REPORT ZSD_SALES_REGISTER_REPORT.
type-pools:slis.

TABLES:KNA1, VBRK, VBRP.

*------------Declaration for selection screen---------*

SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE
TEXT-000.

SELECT-OPTIONS:s_kunnr for kna1-kunnr,
               s_matnr for vbrp-matnr,
               s_werks for vbrp-werks,
               s_fkart for vbrk-fkart,
               s_fkdat for vbrk-fkdat.

SELECTION-SCREEN END OF BLOCK B1.

*-------Structure declarations--------------*
```

```abap
types:BEGIN OF ty_vbrk,
       kunrg TYPE kunnr,      "Customer code
       vbeln TYPE vbeln_vf,   "Invoice nO
       fkdat TYPE fkdat,      "Invoice date
       spart TYPE spart,      "Division
       matnr TYPE matnr,      "Material
       arktx TYPE arktx,      "Item description
       fkimg TYPE fkimg,      "Invoice quntity
       netwr TYPE netwr,      "Invoice value
       aubel TYPE aubel,      "Sales Order
       vgbel TYPE vgbel,      "Delivary No
       name1 TYPE name1,      "Customer name
       ort01 TYPE ort01,      " Location
       werks TYPE werks_d,    "Plant
       vtext TYPE vtext,      "Division description
       end of ty_vbrk.

types:BEGIN OF ty_kna1,
       kunnr TYPE kunnr,
       name1 TYPE name1,
       ort01 TYPE ort01,
       end of ty_kna1.

types:BEGIN OF ty_tspat,
       spart TYPE spart,
       vtext TYPE vtext,
       end of ty_tspat.

data:wa_kna1 TYPE ty_kna1,
     wa_vbrk TYPE ty_vbrk,
     wa_tspat TYPE ty_tspat,
     wa_final TYPE ty_VBRK,
     it_kna1 TYPE TABLE OF ty_kna1,
     it_vbrk TYPE TABLE OF ty_vbrk,
     it_tspat TYPE TABLE OF ty_tspat,
     it_final TYPE TABLE OF ty_vbrk.

*-----------Maintaning fieldcatlogs.-----------*

data:wa_layo TYPE slis_layout_ALV,
     wa_fcat TYPE slis_fieldcat_alv,
     wa_event TYPE slis_alv_event,
```

```abap
        wa_header TYPE slis_listheader,
        it_event TYPE slis_t_event,
        it_header TYPE slis_t_listheader,
        wa_sort TYPE slis_sortinfo_alv,
        it_sort TYPE slis_t_sortinfo_alv,
        it_fcat TYPE slis_t_fieldcat_alv,
        it_fieldcat TYPE slis_t_fieldcat_Alv.

CONSTANTs:P_vbeln TYPE slis_fieldname VALUE 'vbeln',
          P_KUNRG TYPE slis_fieldname VALUE 'kunrg',
          P_name1 TYPE slis_fieldname VALUE 'name1',
          P_ort01 TYPE slis_fieldname VALUE 'ort01',
          P_fkdat TYPE slis_fieldname VALUE 'fkdat',
          P_matnr TYPE slis_fieldname VALUE 'matnr',
          P_arktx TYPE slis_fieldname VALUE 'arktx',
          P_werks TYPE slis_fieldname VALUE 'werks',
          P_spart TYPE slis_fieldname VALUE 'spart',
          P_vtext TYPE slis_fieldname VALUE 'vtext',
          P_fkimg TYPE slis_fieldname VALUE 'fkimg',
          P_netwr TYPE slis_fieldname VALUE 'netwr',
          P_aubel TYPE slis_fieldname VALUE 'aubel',
          P_vgbel TYPE slis_fieldname VALUE 'vgbel'.

data:v_flag(100) TYPE c VALUE 'X'.  "v_flag for
hotspot for interactive field and sum

PERFORM GET_VBRK_VBRP.
PERFORM GET_KNA1.
PERFORM GET_TSPAT.
PERFORM FILL_CAT USING:
p_kunrg 'Customer Number'    v_flag  space 'C510',
"C-COLOR 5-TYPE COLOR  1 -FOR INTENCIFIED ON , 0 -FOR
INTENCIFIED OFF 0-DEFAULT
p_name1 'Customer name'       space   space 'C500',
p_ort01 'Location'            space   space 'C200',
p_fkimg 'Invoice Qty'         space  v_flag 'C310',
p_netwr 'Invoice Value'       space  v_flag 'C410'.
PERFORM FILL_CAT1 USING:
p_vbeln 'Invoice No'          space space  'C110',
p_fkdat 'Invoice Date'        space space  'C200',
p_matnr 'Material'            space space  'C300',
p_arktx 'Item Description'    space space  'C400',
```

```abap
          p_werks 'Plant'                space space  'C500',
          p_spart 'Division'             space space  'C600',
          p_vtext 'Division Description' space space 'C700',
          p_fkimg 'Invoice Qty'          space v_flag 'C310',
          p_netwr 'Invoice Value'        space v_flag 'C410',
          p_aubel 'Sales Order'          space space  'C110',
          p_vgbel 'Delivary No'          space space  'C210'.
PERFORM layout.
PERFORM sort.
PERFORM process_data.
PERFORM fill_events.
PERFORM read_data.
PERFORM display_alv.

*------------Getting data from vbrk and vbrp table
using inner join------------*
form get_vbrk_vbrp.

  select vbrk~kunrg
         vbrk~vbeln
         vbrk~fkdat
         vbrp~spart
         vbrp~matnr
         vbrp~arktx
         vbrp~fkimg
         vbrp~netwr
         vbrp~aubel
         vbrp~vgbel INTO TABLE it_vbrk
         FROM vbrk INNER JOIN vbrp
         ON vbrk~vbeln = vbrp~vbeln
         WHERE VBRK~kunrg IN S_KUNNR
         AND   Vbrp~matnr in s_matnr
         AND   vbrk~fkdat in s_fkdat
         AND   vbrk~fkart in s_fkart.

endform.

*-----------------Getting data from kna1 table.-----
-------*
form get_kna1.

  if NOT it_vbrk IS INITIAL.
```

```abap
      select kunnr name1 ort01 FROM kna1 INTO TABLE it_kna1
        FOR ALL ENTRIES IN it_vbrk
        WHERE kunnr = it_vbrk-kunrg.
    endif.

endform.

*-------------------------Getting data from tspat table.----------*

form get_tspat.

  if NOT it_vbrk IS INITIAL.

    select spart vtext FROM tspat INTO TABLE it_tspat
      FOR ALL ENTRIES IN it_vbrk
      WHERE spart = it_vbrk-spart.

  endif.

endform.

form fill_cat USING A B C D E .

  wa_fcat-fieldname = A.
  wa_fcat-seltext_m = B.
  wa_fcat-hotspot   = C.
  wa_fcat-do_sum    = D.
  wa_fcat-emphasize = E.
  Append wa_fcat to it_fcat.
  clear wa_fcat.

endform.

form fill_cat1 USING A B C D E .

  wa_fcat-fieldname = A.
  wa_fcat-seltext_m = B.
  wa_fcat-hotspot   = C.
  wa_fcat-do_sum    = D.
```

```abap
    wa_fcat-emphasize = E.
    Append wa_fcat to it_fieldcat.
    clear wa_fcat.

endform.

form layout.

    wa_layo-zebra = 'X'.
    wa_layo-colwidth_optimize = 'X'.

endform.

form sort.

    wa_sort-fieldname = 'kunrg'.
    wa_sort-tabname   = 'it_vbrk'.
    wa_sort-up        = 'X'.
    wa_sort-subtot    = 'X'.
    Append wa_sort to it_sort.
    Clear wa_sort.

endform.

form fill_events.
    data:lv_header(100) TYPE c,
         N(4) TYPE c.

    wa_event-name = 'TOP_OF_PAGE'.
    wa_event-form = 'F_HEADER'.
    Append wa_event to it_event.
    clear wa_event.

*--------------- Mantaining headers-----------------
--*

    wa_header-typ   = 'H'.      " H used for Big
heading
    wa_header-info  = 'AAVI INOVATIONS PRIVATE LTD'.
    Append wa_header to it_header.
    clear wa_header.
```

```abap
describe table it_VBRK lines N.
concatenate 'Total Records' N INTO lv_header
SEPARATED BY ':'.

  wa_header-typ   = 'S'.      " S used for small
heading
  wa_header-info  = 'Sales Registored Reports'.
  Append wa_header to it_header.
  clear wa_header.

  wa_header-typ   = 'A'.    " A used for Action header
  wa_header-info  = lv_header.
  Append wa_header to it_header.
  clear wa_header.

endform.

form read_data.

  loop at it_vbrk INTO wa_vbrk.

    read TABLE it_kna1 INTO wa_kna1 WITH KEY kunnr =
wa_vbrk-kunrg.
    if sy-subrc eq 0.
      wa_vbrk-name1 = wa_kna1-name1.
      wa_vbrk-ort01 = wa_kna1-ort01.
    endif.
    read TABLE it_tspat INTO wa_tspat WITH KEY spart
= wa_vbrk-spart.
    if sy-subrc eq 0.
      wa_vbrk-vtext = wa_tspat-vtext.
    endif.
    modify it_vbrk FROM wa_vbrk TRANSPORTING name1
ort01 vtext.
  endloop.

endform.

form f_header.

  CALL FUNCTION 'REUSE_ALV_COMMENTARY_WRITE'
    EXPORTING
```

```abap
        IT_LIST_COMMENTARY        = it_header.
*       I_LOGO                    =
                 .
endform.

form process_data.

  sort it_vbrk by kunrg.

  loop at it_vbrk INTO wa_vbrk.
  At end of kunrg.
  sum.
  wa_final-kunrg = wa_vbrk-kunrg.
  wa_final-fkimg = wa_vbrk-fkimg.
  wa_final-netwr = wa_vbrk-netwr.

  read TABLE it_kna1 INTO wa_kna1 WITH KEY kunnr =
wa_vbrk-kunrg.
  if sy-subrc eq 0.
    wa_vbrk-name1 = wa_kna1-name1.
    wa_vbrk-ort01 = wa_kna1-ort01.
  endif.

  read TABLE it_tspat INTO wa_tspat WITH KEY spart =
wa_vbrk-spart.
  if sy-subrc eq 0.
    wa_vbrk-vtext = wa_tspat-spart.
  endif.

  endat.

  endloop.

endform.

form display_alv.

  data lv_repid TYPE sy-repid.
      lv_repid = sy-repid.

  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
```

```abap
      I_CALLBACK_PROGRAM                     = lv_repid
      I_CALLBACK_USER_COMMAND            =
'F_INTERACTIVE'
      IS_LAYOUT                              = wa_layo
      IT_FIELDCAT                            = it_fcat
      I_SAVE                                 = 'X'
      IT_EVENTS                              = It_event
    TABLES
      T_OUTTAB                               = it_vbrk
    EXCEPTIONS
      PROGRAM_ERROR                          = 1
      OTHERS                                 = 2.

endform.

*----------here is the logic for interactive
reports.------------*

form F_INTERACTIVE USING ucomm TYPE sy-ucomm
      sel TYPE slis_selfield.

  refresh it_final.

  if sel-fieldname = 'Kunrg'.

    loop at it_vbrk INTO wa_vbrk WHERE kunrg = sel-
value.
      Append wa_vbrk to it_final.
    endloop.

  endif.

      data lv_repid TYPE sy-repid.
        lv_repid = sy-repid.

  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
      I_CALLBACK_PROGRAM                     = lv_repid
       IS_LAYOUT                             = wa_layo
      IT_FIELDCAT                            = it_fieldcat
      I_SAVE                                 = 'X'
    TABLES
```
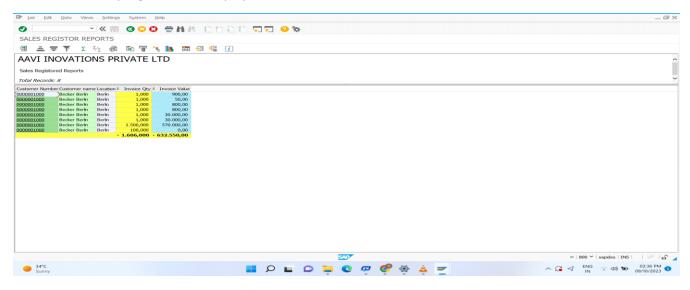
```
          T_OUTTAB                              = it_vbrk
       EXCEPTIONS
          PROGRAM_ERROR                         = 1
          OTHERS                                = 2.

endform.
```

Then Save check and activate the program.

Now Execute the program.it will display the Basic list.

Now if you click on 1000 you will get the Interactive list.