**LIT TRAINING**

**Batch Name : SAP ABAP ON HANA** **DAY 15 ASSIGNMENT**

**EMPLOYEE ID – 46255245**

Assignment 1 – Create a Simple program to fetch data and display ALV using Std. SAP class-methods.

**Scenario** –

* Create a Custom program using ECLIPSE. Select data from table SNWD\_PD.
* The output should consist three columns – Category, Max price and Average price.
* The data should consist of all categories from SNWD\_PD, with maximum price for each category and average price for each category.
* There should be only 1 select statement used to achieve this using In-line Declaration, CASE, Aggregate, HAVING and GROUP BY clauses as needed.
* Selection screen can contain CATEGORY
* Output should be in ALV Grid format.

**Tables involved** – SNWD\_PD / Columns – CATEGORY & PRICE

CODE:-

\*&---------------------------------------------------------------------\*

\*& Report z45alvidalit\_1

\*&---------------------------------------------------------------------\*

\* Description: \*

\* \*

\* Author: \*

\* \*

\* Create date: \*

\* \*

\*&---------------------------------------------------------------------\*

\* Modification log: \*

\* -------------------------------------------------------------------- \*

\* Date User name ID Change Request Case ref. \*

\* Description \*

\*&---------------------------------------------------------------------\*

REPORT z45alvidalit\_1.

TABLES: SNWD\_PD.

SELECT-OPTIONS: S\_CAT FOR SNWD\_PD-CATEGORY.

SELECT CATEGORY,

MAX( PRICE ) AS MAX\_PRICE,

AVG( PRICE ) AS AVG\_PRICE

FROM SNWD\_PD INTO TABLE @DATA(ITAB)

GROUP BY CATEGORY

HAVING CATEGORY IN @S\_CAT

ORDER BY CATEGORY.

CL\_SALV\_table=>factory(

EXPORTING

list\_display = IF\_SALV\_C\_BOOL\_SAP=>FALSE " ALV Displayed in List Mode

\* r\_container = " Abstract Container for GUI Controls

\* container\_name =

IMPORTING

r\_salv\_table = data(alv) " Basis Class Simple ALV Tables

CHANGING

t\_table = itab

).

\* CATCH cx\_salv\_msg.

ALV->get\_columns( )->get\_column( columnname = 'MAX\_PRICE' )->set\_medium\_text( value = 'Max Price' ).

\* CATCH cx\_salv\_not\_found. "

ALV->get\_columns( )->get\_column( columnname = 'AVG\_PRICE' )->set\_medium\_text( value = 'Average Price' ).

alv->display( ).

INPUT & OUTPUT SCREENS:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Assignment 2 – ALV with IDA

**Scenario** –

* Create one CDS view by selecting fields VBELN,POSNR,  VKORG, MATNR, MENGE from VBAK and VBAP table
* Perform below operations using IDA framework

1. Create select-options based on MATNR for selection screen
2. Fetch records from CDS view in ABAP program
3. Pass user input (select-option entries) to filter record accordingly
4. Hide POSNR field from display
5. Change label of VBELN as SOrder, VKORG as SOrganization, and MENGE as TotalQuantity
6. Disable standard function SORT for MATNR field
7. Hide standard toolbar FILTER
8. Set title as “ALV report with IDA, CDS consumption”
9. Enable ZEBRA pattern
10. Multiple row selection should be enabled

**Tables involved** – VBAP, VBAK

\*&---------------------------------------------------------------------\*

\*& Report z45alvidalit\_2

\*&---------------------------------------------------------------------\*

\* Description: \*

\* \*

\* Author: \*

\* \*

\* Create date: \*

\* \*

\*&---------------------------------------------------------------------\*

\* Modification log: \*

\* -------------------------------------------------------------------- \*

\* Date User name ID Change Request Case ref. \*

\* Description \*

\*&---------------------------------------------------------------------\*

REPORT z45alvidalit\_2.

tables: vbap,vbak.

select-options s\_matnr for vbap-matnr.

DATA(OBJ) = CL\_SALV\_GUI\_TABLE\_IDA=>create\_for\_cds\_view(

iv\_cds\_view\_name = 'Z45CDSALV\_2').

DATA(LV\_RANGE) = NEW CL\_SALV\_RANGE\_TAB\_COLLECTOR( ).

LV\_RANGE->add\_ranges\_for\_name(

EXPORTING

iv\_name = 'MATNR'

it\_ranges = s\_matnr[]

).

lv\_range->get\_collected\_ranges(

IMPORTING

et\_named\_ranges = DATA(LO\_NAMED\_RANGES)

).

OBJ->set\_select\_options(

EXPORTING

it\_ranges = LO\_NAMED\_RANGES

\* io\_condition =

).

\* CATCH cx\_salv\_ida\_associate\_invalid. "

\* CATCH cx\_salv\_db\_connection. "

\* CATCH cx\_salv\_ida\_condition\_invalid. "

\* CATCH cx\_salv\_ida\_unknown\_name. "

OBJ->field\_catalog( )->get\_available\_fields(

IMPORTING

ets\_field\_names = DATA(ITAB)

).

DELETE ITAB WHERE TABLE\_LINE = 'POSNR'.

OBJ->field\_catalog( )->set\_field\_header\_texts(

EXPORTING

iv\_field\_name = 'VBELN'

iv\_header\_text = 'SOrder'

iv\_tooltip\_text = 'Sales Order'

\* iv\_tooltip\_text\_long =

).

\* CATCH cx\_salv\_ida\_unknown\_name. "

\* CATCH cx\_salv\_call\_after\_1st\_display. "

obj->field\_catalog( )->set\_field\_header\_texts(

EXPORTING

iv\_field\_name = 'VKORG'

iv\_header\_text = 'SOrganization'

iv\_tooltip\_text = 'Sales Organization'

\* iv\_tooltip\_text\_long =

).

\* CATCH cx\_salv\_ida\_unknown\_name. "

\* CATCH cx\_salv\_call\_after\_1st\_display. "

obj->field\_catalog( )->set\_field\_header\_texts(

EXPORTING

iv\_field\_name = 'ZMENG'

iv\_header\_text = 'POrder'

iv\_tooltip\_text = 'Purchase Order Details'

\* iv\_tooltip\_text\_long =

).

\* CATCH cx\_salv\_ida\_unknown\_name. "

\* CATCH cx\_salv\_call\_after\_1st\_display. "

obj->field\_catalog( )->disable\_sort( iv\_field\_name = 'MATNR' ).

\* CATCH cx\_salv\_ida\_unknown\_name. "

\* CATCH cx\_salv\_call\_after\_1st\_display. "

OBJ->toolbar( )->set\_visibility\_std\_functions(

EXPORTING

\* iv\_details = ABAP\_UNDEFINED

\* iv\_sort = ABAP\_UNDEFINED

\* iv\_group = ABAP\_UNDEFINED

iv\_filter = '' "ABAP\_UNDEFINED

\* iv\_aggregation = ABAP\_UNDEFINED

\* iv\_print = ABAP\_UNDEFINED

\* iv\_export = ABAP\_UNDEFINED

\* iv\_settings\_dialog = ABAP\_UNDEFINED

).

OBJ->display\_options( )->set\_title( iv\_title = 'ALV report with IDA, CDS consumption' ).

OBJ->display\_options( )->enable\_alternating\_row\_pattern( ).

OBJ->selection( )->set\_selection\_mode(

iv\_mode = 'MULTI'

).

\* CATCH cx\_salv\_ida\_contract\_violation. "

OBJ->fullscreen( )->display( ).

INPUT & OUTPUT:-

Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface, text, application

Description automatically generated