**­  LIT TRAINING**

**DAILY ASSIGNMENT SUBMISSIONS**

**DAY - 15**

**EMP ID : 46255263**

**EMP NAME : JYOTSNA SHREE RAIVADA**

**CDS-Assignment 1 – Create a Simple CDS view with Date Functions in ABAP CDS Views**

**Scenario –**

* **Create a CDS view with 7 fields from VBAK using date functions**

1. **VBELN (Sales Document),**
2. **AUART (Sales Document Type),**
3. **AUDAT (Document Date),**
4. **VDATU (Requested delivery date)**
5. **Days between Document Date & Requested delivery date as Processing days**
6. **Delivery date + 10 days as shipping date**
7. **Delivery Date + 2 months as billing date**

**Tables involved – VBAK**

**define** **view** Y\_CDS\_DAY14\_Q1 **as** **select** **from** vbak

**{**

vbeln **as** Sales\_Document**,**

auart **as** Sales\_Document\_Type**,**

audat **as** Document\_Date**,**

vdatu **as** Requested\_Delivery\_Date**,**

DATS\_DAYS\_BETWEEN**(**audat**,** vdatu**)** **as** Processing\_Days**,**

DATS\_ADD\_DAYS**(**vdatu**,** 10**,** 'NULL'**)** **as** Shipping\_Date**,**

DATS\_ADD\_MONTHS**(**vdatu**,** 2**,** 'NULL'**)** **as** Billing\_Date

**}**

Graphical user interface, table

Description automatically generated

**CDS-Assignment 2 – Create a Simple CDS view with String Functions in ABAP CDS Views**

**Scenario –**

* **Create a CDS view with 10 fields from KNA1 using string functions**

1. **KUNNR (Customer Number),**
2. **LAND1 (Country Key),**
3. **Concatenate NAME1 & NAME2**
4. **Concatenate STRAS, ORT01 & REGIO & PSTLZ with space in between.**
5. **Get first 2 characters of NAME1**
6. **Get last 3 characters of NAME2**
7. **No of characters of TELF1**
8. **NAME1+2(4). 4 characters starting from 2nd character.**
9. **Remove the leading zeros of KUNNR**
10. **Remove last 4 characters of TELF1.**

**Tables involved – KNA1**

**define** **view** Y\_CDS\_DAY14\_Q2 **as** **select** **from** kna1 **{**

kunnr **as** Customer\_Number**,**

land1 **as** Country\_Key**,**

CONCAT**(**name1**,** name2**)** **as** Full\_Name**,**

CONCAT\_WITH\_SPACE**(**stras**,** ort01**,** 4**)** **as** Vendor\_Name**,**

CONCAT\_WITH\_SPACE**(**regio**,** pstlz**,** 4 **)** **as** Modification**,**

left**(**name1**,** 2**)** **as** First\_2Chars\_Name1**,**

RIGHT**(**name1**,** 3**)** **as** Last\_3Chars\_Name2**,**

LENGTH**(**telf1**)** **as** Total\_Chars\_of\_TELF1**,**

SUBSTRING**(**name1**,** 2**,** 4**)** **as** Char2\_To\_Char5**,**

LTRIM**(**kunnr**,** '0'**)** **as** Removed\_Leading\_Zeros**,**

RTRIM**(**telf1**,** '4'**)** **as** Removed\_Last\_4Chars

**}**

Graphical user interface, application

Description automatically generated

**CDS-Assignment 3 – Create a Simple CDS view with Parameters**

**Scenario –**

* **Create a CDS view with GL currency (PSWSL) as parameter and get the entries form BSEG**

**Tables involved – BSEG**

**define** **view** Y\_CDS\_DAY14\_Q3

**with** **parameters** gl\_currency **:** *pswsl*

**as** **select** **from** bseg**{**

bukrs **as** Company\_Code**,**

belnr **as** Doc\_Num**,**

gjahr **as** Fiscal\_Year**,**

koart **as** Account\_Type**,**

shkzg **as** Card\_Type**,**

pswsl **as** Currency

**}**

**where** pswsl **=** **$parameters.**gl\_currency**;**

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

**AMDP-Assignment 1 – Procedures - create a basic procedure to fetch data with scalar variable**

**Scenario –**

* **Print Sales Order number, Material Number, Quantity from VBAP table using scaler variable**

**Tables involved – VBAP**

CLASS y\_amdp1 DEFINITION

PUBLIC

FINAL

CREATE PUBLIC .

PUBLIC SECTION.

TYPES : BEGIN OF TY\_TAB1,

VBELN TYPE VBAP-VBELN,

matnr TYPE VBAP-MATNR,

zmeng TYPE VBAP-ZMENG,

END OF TY\_TAB1.

TYPES : IT\_TAB1 TYPE STANDARD TABLE OF TY\_TAB1.

INTERFACES : IF\_AMDP\_MARKER\_HDB.

METHODS get\_sales\_data

IMPORTING VALUE(S\_DOCS) TYPE STRING

VALUE(IM\_MANDT) TYPE MANDT

EXPORTING VALUE(IT\_RES) TYPE IT\_TAB1.

PROTECTED SECTION.

PRIVATE SECTION.

ENDCLASS.

CLASS y\_amdp1 IMPLEMENTATION.

METHOD GET\_sales\_DATA BY DATABASE PROCEDURE FOR HDB LANGUAGE SQLSCRIPT OPTIONS READ-ONLY USING VBAP.

IT\_RES = SELECT VBELN, MATNR, ZMENG FROM VBAP;

ENDMETHOD.

ENDCLASS.

**REPORT PROGRAM:**

REPORT Y\_AMDP1\_DAY14.  
TABLES: vbap.  
SELECT-OPTIONS doc\_num FOR vbap-vbeln.  
DATA(o\_ref) = NEW  Y\_AMDP1( ). ##Needed  
  
DATA(s\_filter) = cl\_shdb\_seltab=>combine\_seltabs(  
it\_named\_seltabs = VALUE #( ( name = ' VBELN ' dref = REF #( DOC\_NUM[] ) ) ) ).  
o\_ref->get\_sales\_data(  
  EXPORTING     ##NEEDED  
    s\_docs = s\_filter  
    im\_mandt = sy-mandt  
  IMPORTING  
    it\_res      = DATA(it\_tab1) ).  
SORT it\_tab1.  
  
cl\_demo\_output=>display(  
  EXPORTING data = it\_tab1  
          name = 'Sales Documents Data'  
).

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application, Word

Description automatically generated**

**AMDP-Assignment 2 – Procedures - create a basic procedure with Table**

**Scenario –**

* **Select VBELN, VKORG, MATNR, MENGE from VBAK and VBAP and display the data using table based on material (MATNR).**
* **MATNR will be input field**
* **Do inner join**

**Tables involved – VBAK, VBAP**

CLASS y\_amdp2 DEFINITION

PUBLIC

FINAL

CREATE PUBLIC .

PUBLIC SECTION.

TYPES : BEGIN OF TY\_TAB1,

VBELN TYPE VBELN,

VKORG TYPE VKORG,

matnr TYPE MATNR,

zmeng TYPE DZMENG,

END OF TY\_TAB1.

TYPES : IT\_TAB1 TYPE STANDARD TABLE OF TY\_TAB1.

INTERFACES : IF\_AMDP\_MARKER\_HDB.

METHODS: get\_sales

IMPORTING VALUE(S\_MATNR) TYPE MATNR

EXPORTING VALUE(OT\_TAB) TYPE IT\_TAB1.

PROTECTED SECTION.

PRIVATE SECTION.

ENDCLASS.

CLASS Y\_AMDP2 IMPLEMENTATION.

METHOD GET\_SALES BY DATABASE PROCEDURE FOR HDB LANGUAGE SQLSCRIPT USING VBAK VBAP.

OT\_TAB = SELECT X.VBELN, X.VKORG, Y.MATNR, Y.ZMENG FROM VBAK X INNER JOIN VBAP AS Y ON

X.VBELN = Y.VBELN WHERE Y.MATNR= :S\_MATNR;

ENDMETHOD.

ENDCLASS.

REPORT Y\_AMDP2\_DAY14.  
PARAMETERS p\_matnr type vbap-matnr.  
DATA(OBJ) = NEW Y\_AMDP2(  ).  
OBJ->get\_sales(  
  EXPORTING  
    s\_matnr = p\_matnr  
  IMPORTING  
    ot\_tab = data(IT\_TAB1)).  
  
cl\_demo\_output=>display(  
  EXPORTING  
    data =   it\_tab1  " Text or Data  
    name = 'Sales Documents Data').

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, application, Word

Description automatically generated**