**LIT \_DAY\_14 \_ASSIGNMENT**

EMPLOYEE\_ID:46255235

NAME: JYOTHI NUNAVATU

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

CODE:

@AbapCatalog.sqlViewName: 'Z235\_Q1\_VIEW'

@AbapCatalog.compiler.compareFilter: true

@AbapCatalog.preserveKey: true

@AccessControl.authorizationCheck: #NOT\_REQUIRED

@EndUserText.label: 'CDS VIEW WITH DATE FUNTIONS'

**define** **view** Z235\_Q1\_CDS **as** **select** **from** vbak **{**

vbeln **as** SalesDocument**,**

auart **as** DocumentType**,**

audat **as** DocDate**,**

vdatu **as** DeliveryDate**,**

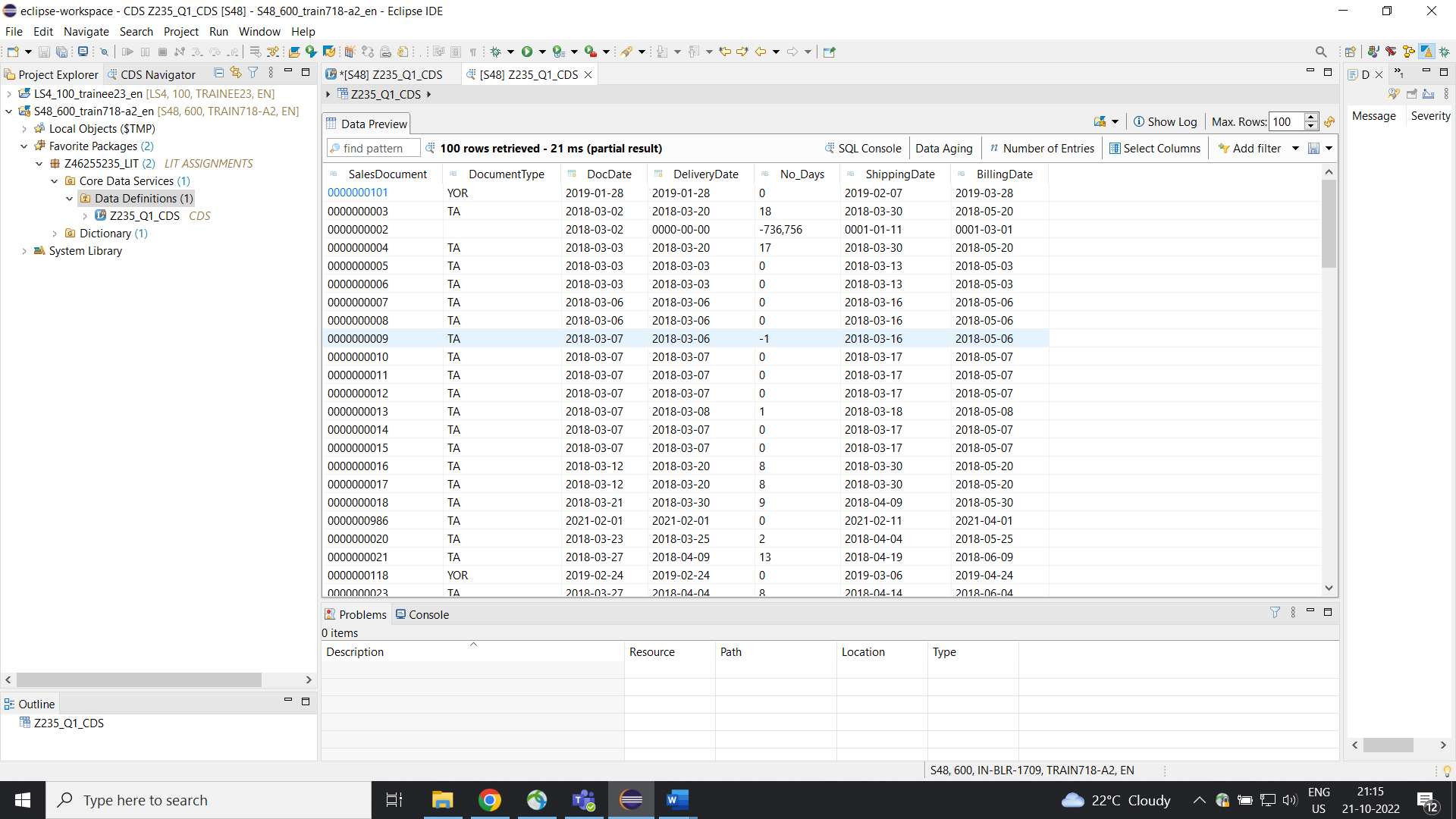
DATS\_DAYS\_BETWEEN**(**audat**,** vdatu**)** **as** No\_Days**,**

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

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

**}**

OUTPUT:



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.

CODE:

@AbapCatalog.sqlViewName: 'Z235\_Q2\_VIEW'

@AbapCatalog.compiler.compareFilter: true

@AbapCatalog.preserveKey: true

@AccessControl.authorizationCheck: #NOT\_REQUIRED

@EndUserText.label: 'CDS view with String Functions'

**define** **view** Z235\_Q2\_CDS **as** **select** **from** kna1 **{**

kunnr **as** CustomerNO**,**

land1 **as** CountryKey**,**

CONCAT**(**kna1**.**name1**,**kna1**.**name2**)** **as** FieldsConcatination**,**

CONCAT\_WITH\_SPACE**(**stras**,**ort01**,**2**)** **as** ConcatingWithSpace**,**

RIGHT**(**kna1**.**name1**,**3**)** **as** NoRight**,**

LENGTH**(**telf1**)** **as** CharNo**,**

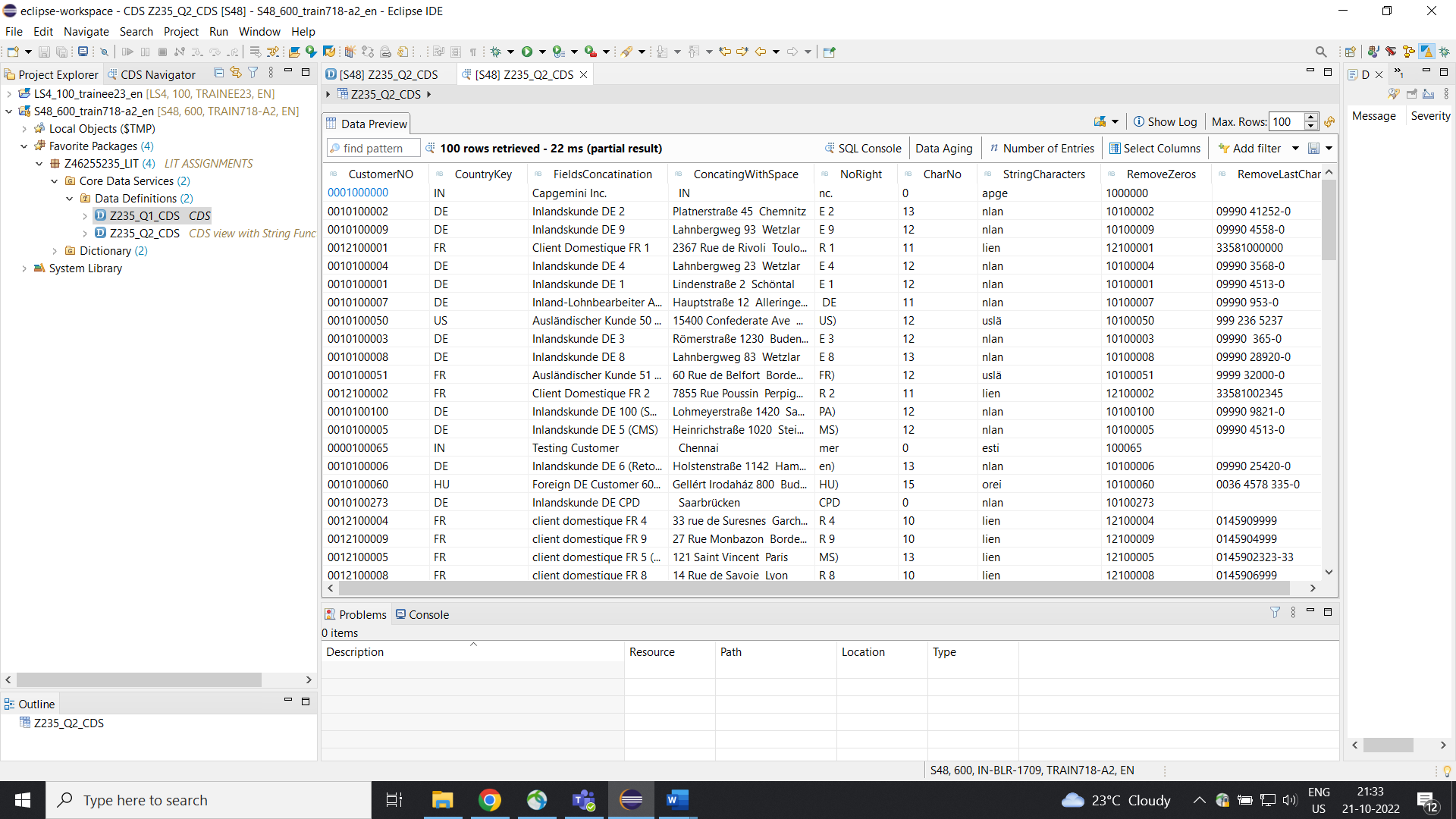
SUBSTRING**(**name1**,**2**,**4**)** **as** StringCharacters**,**

LTRIM**(**kunnr**,** '0'**)** **as** RemoveZeros**,**

RTRIM**(**telf1**,**'4'**)** **as** RemoveLastChar

**}**

OUTPUT:



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

CODE

@AbapCatalog.sqlViewName: 'Z235\_Q3\_VIEW'

@AbapCatalog.compiler.compareFilter: true

@AbapCatalog.preserveKey: true

@AccessControl.authorizationCheck: #NOT\_REQUIRED

@EndUserText.label: 'CDS view with Parameters'

**define** **view** Z235\_Q3\_CDS

**with** **parameters** GL\_CURRENCY **:** *abap***.***cuky***(**5**)**

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

bukrs **as** CompnayCode**,**

belnr **as** AccountingNo**,**

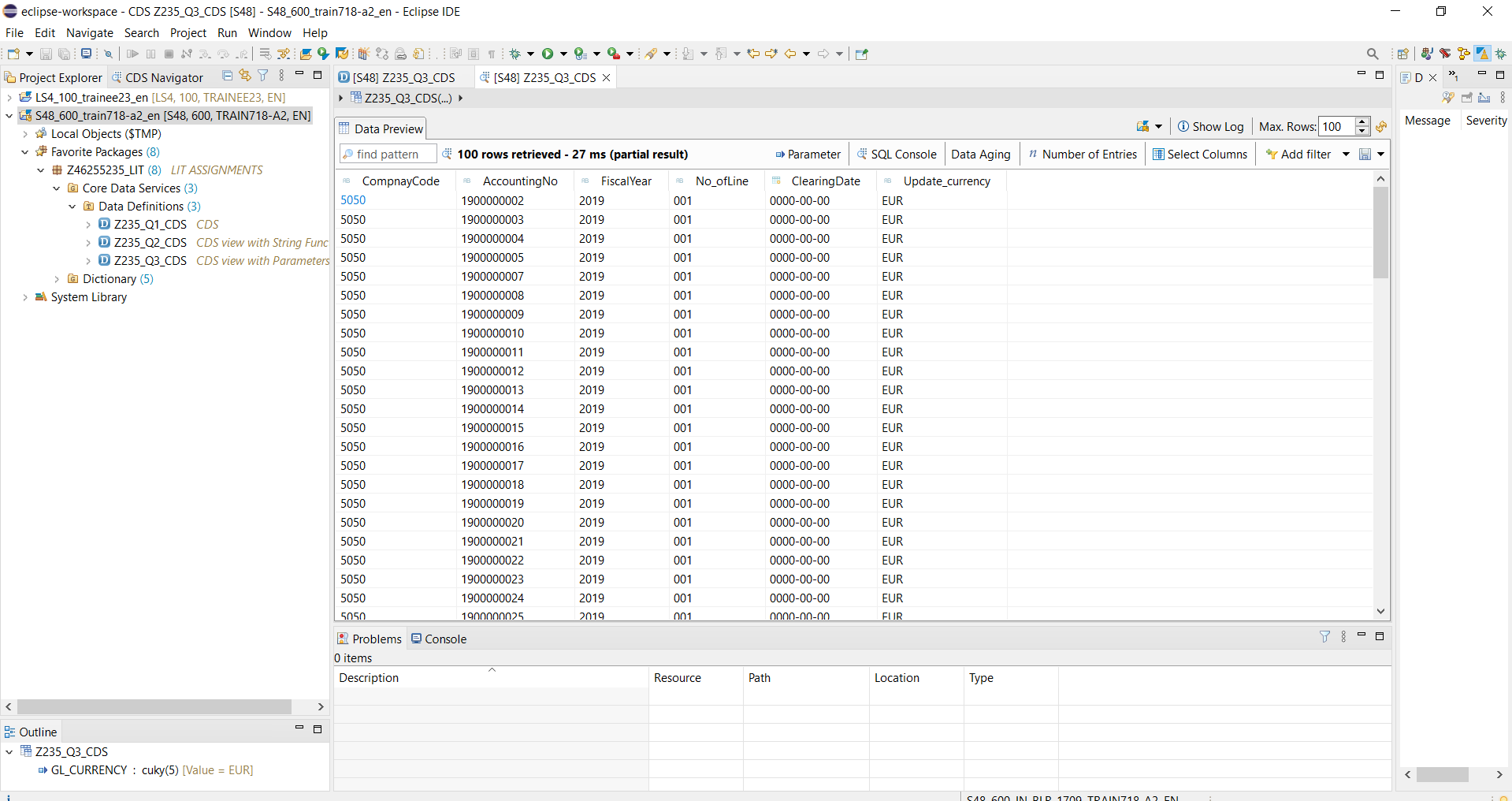
gjahr **as** FiscalYear**,**

buzei **as** No\_ofLine**,**

augdt **as** ClearingDate**,**

pswsl **as** Update\_currency

**}** **where** pswsl **=** **$parameters.**GL\_CURRENCY

OUTPUT:

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

CODE

* AMDP\_CLASS:

class z235\_q4\_amdp definition

public

final

create public .

public section.

TYPES: IT\_VBAP TYPE TABLE OF Z235\_TVBAPSTRU.

INTERFACES : IF\_AMDP\_MARKER\_HDB.

METHODS: METHOD\_VBAP

EXPORTING VALUE(OUT\_VBAP) TYPE Z235\_TVBAPSTRU.

endclass.

class z235\_q4\_amdp implementation.

METHOD METHOD\_VBAP BY DATABASE PROCEDURE FOR HDB LANGUAGE

SQLSCRIPT OPTIONS READ-only

USING VBAP.

OUT\_VBAP = SELECT VBELN,MATNR,ZMENG FROM VBAP;

ENDMETHOD.

endclass.

* AMDP PROGRAM

report z235\_q4\_amdp.

data(o\_amdp) = new z235\_q4\_amdp( ).

o\_amdp->method\_vbap(

importing

out\_vbap = data(it\_vbap)

).

cl\_demo\_output=>display(

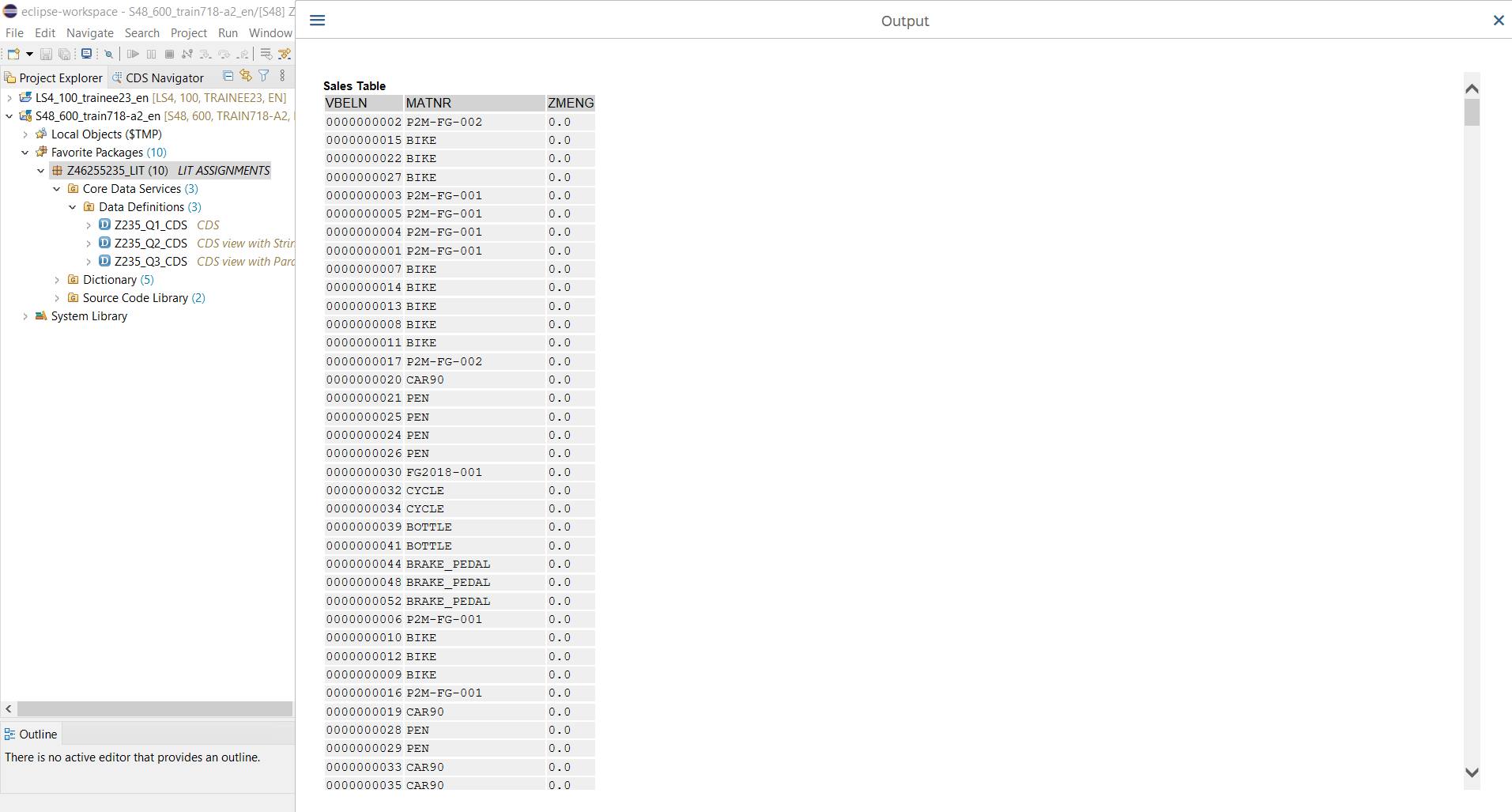
exporting

data = it\_vbap " Text or Data

name = 'Sales Table'

).

* OUTPUT



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

CODE:

* AMDP CLASS

class z235\_q5\_amdp definition

public final create public .

public section.

TYPES : BEGIN OF TY\_TABLE,

VBELN TYPE VBELN\_VA,

VKORG TYPE VKORG,

MATNR TYPE MATNR,

ZMENG TYPE DZMENG,

END OF TY\_TABLE.

TYPES: IT\_FINAL TYPE TABLE OF TY\_TABLE.

INTERFACES: IF\_AMDP\_MARKER\_HDB.

CLASS-METHODS:GET\_FINAL\_TABLE

IMPORTING VALUE(S\_MATNR) TYPE VBAP-MATNR

EXPORTING VALUE(OUT\_FINAL) TYPE IT\_FINAL.

protected section.

private section.

endclass.

class z235\_q5\_amdp implementation.

METHOD GET\_FINAL\_TABLE BY DATABASE PROCEDURE FOR HDB LANGUAGE

SQLSCRIPT OPTIONS READ-ONLY

USING VBAP VBAK.

OUT\_FINAL = SELECT A.VBELN, A.VKORG,

B.MATNR, B.ZMENG FROM VBAK AS A

INNER JOIN VBAP AS B

ON A.VBELN = B.VBELN;

ENDMETHOD.

endclass.

* AMDP PROGRAM

report z235\_q5\_amdp.

PARAMETERS : P\_MATNR TYPE VBAP-MATNR.

DATA(O\_AMDP) = NEW z235\_q5\_amdp( ).

O\_AMDP->get\_final\_table(

exporting

s\_matnr = P\_MATNR

importing

out\_final = DATA(IT\_FINAL)

).

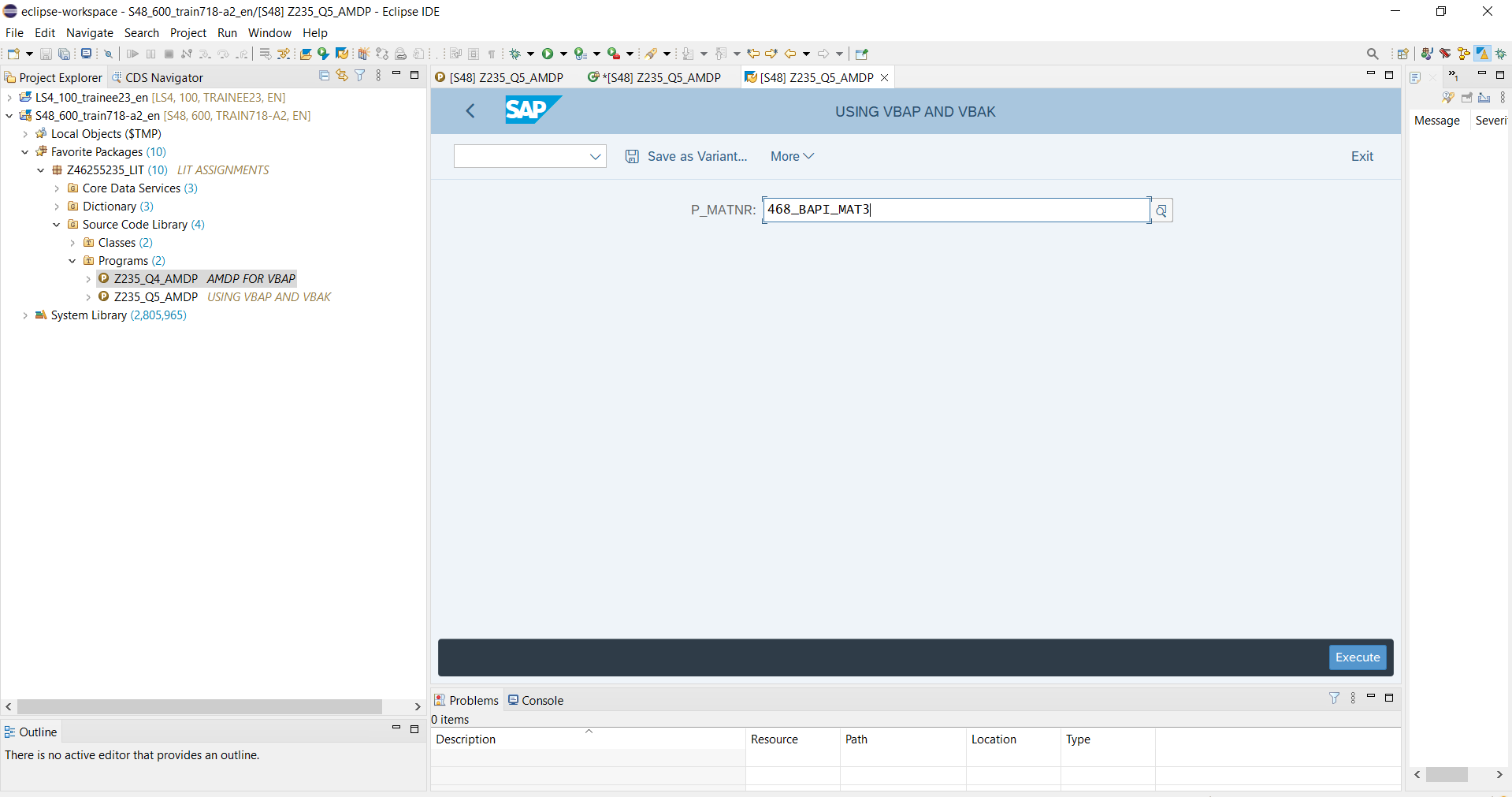
CL\_DEMO\_OUTPUT=>display(

exporting

data = IT\_FINAL " Text or Data

name = 'JOINING VBAP AND VBAK'

).

* SELECTION SCREEN
* OUTPUT

