**LIT TRAINING**

Name: Pataparla Thanusha

Employee ID: 46255251

Q1. **CDS-Assignment 1 – Create a Simple CDS view with Date Functions in ABAP CDS Views**

Code:

@AbapCatalog.sqlViewName: 'Z5251\_SIMPLE\_SQL'

@AbapCatalog.compiler.compareFilter: true

@AbapCatalog.preserveKey: true

@AccessControl.authorizationCheck: #NOT\_REQUIRED

@EndUserText.label: 'SIMPLE CDS WITH SELECT'

**define** **view** Z5251\_SIMPLE\_CDS **as** **select** **from** vbak **{**

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

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

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

vdatu **as** Requested\_delivery\_date**,**

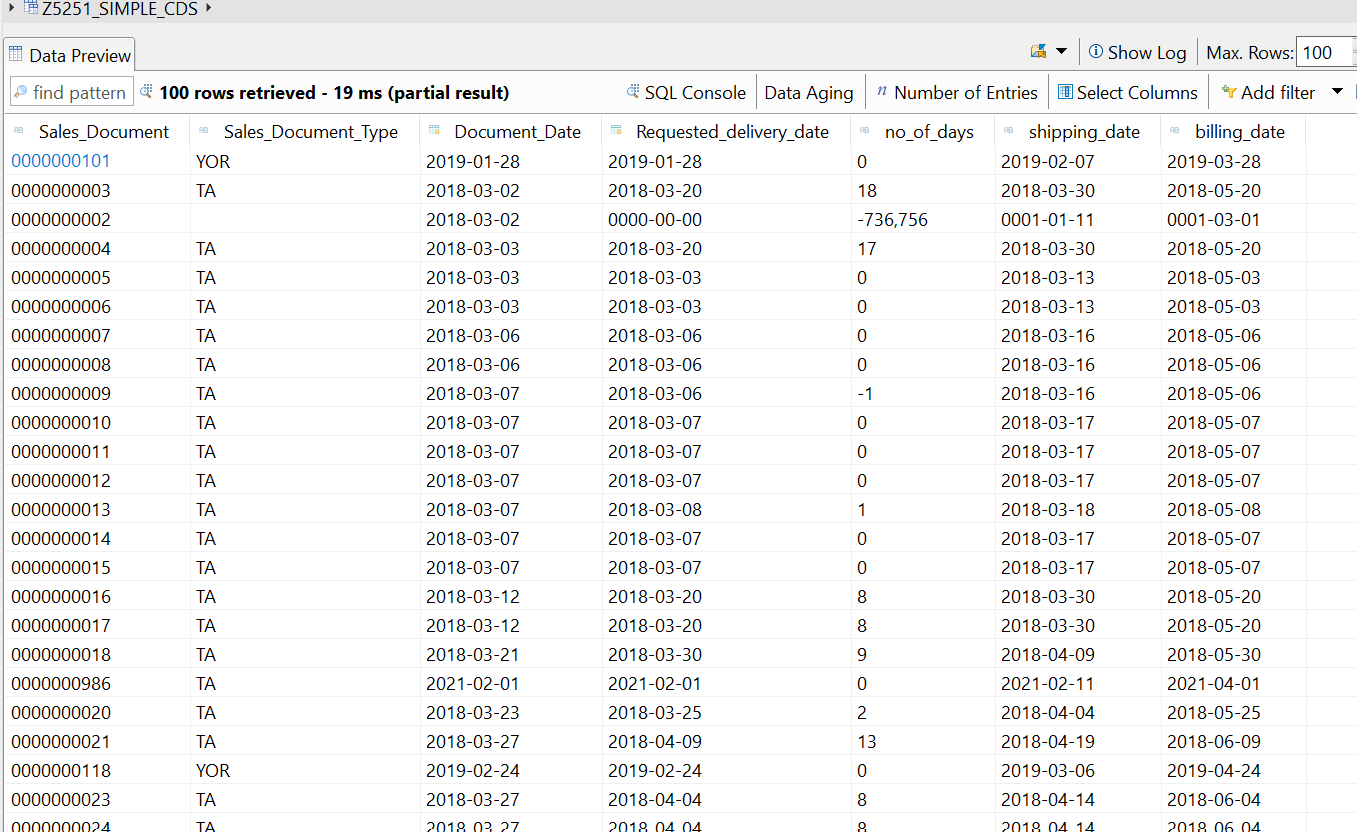
DATS\_DAYS\_BETWEEN**(**audat**,** vdatu**)** **as** no\_of\_days**,**

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

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

**}**

OUTPUT:



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

Code:

@AbapCatalog.sqlViewName: 'Z5251\_STR\_CDS'

@AbapCatalog.compiler.compareFilter: true

@AbapCatalog.preserveKey: true

@AccessControl.authorizationCheck: #NOT\_REQUIRED

@EndUserText.label: 'Simple cds with string fns'

**define** **view** Z5251\_STRING\_CDS **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** COMBO**,**

left**(**name1**,** 2**)** **as** F2C\_NAME1**,**

RIGHT**(**name1**,** 3**)** **as** L3C\_NAME2**,**

LENGTH**(**telf1**)** **as** NAME\_LENGTH**,**

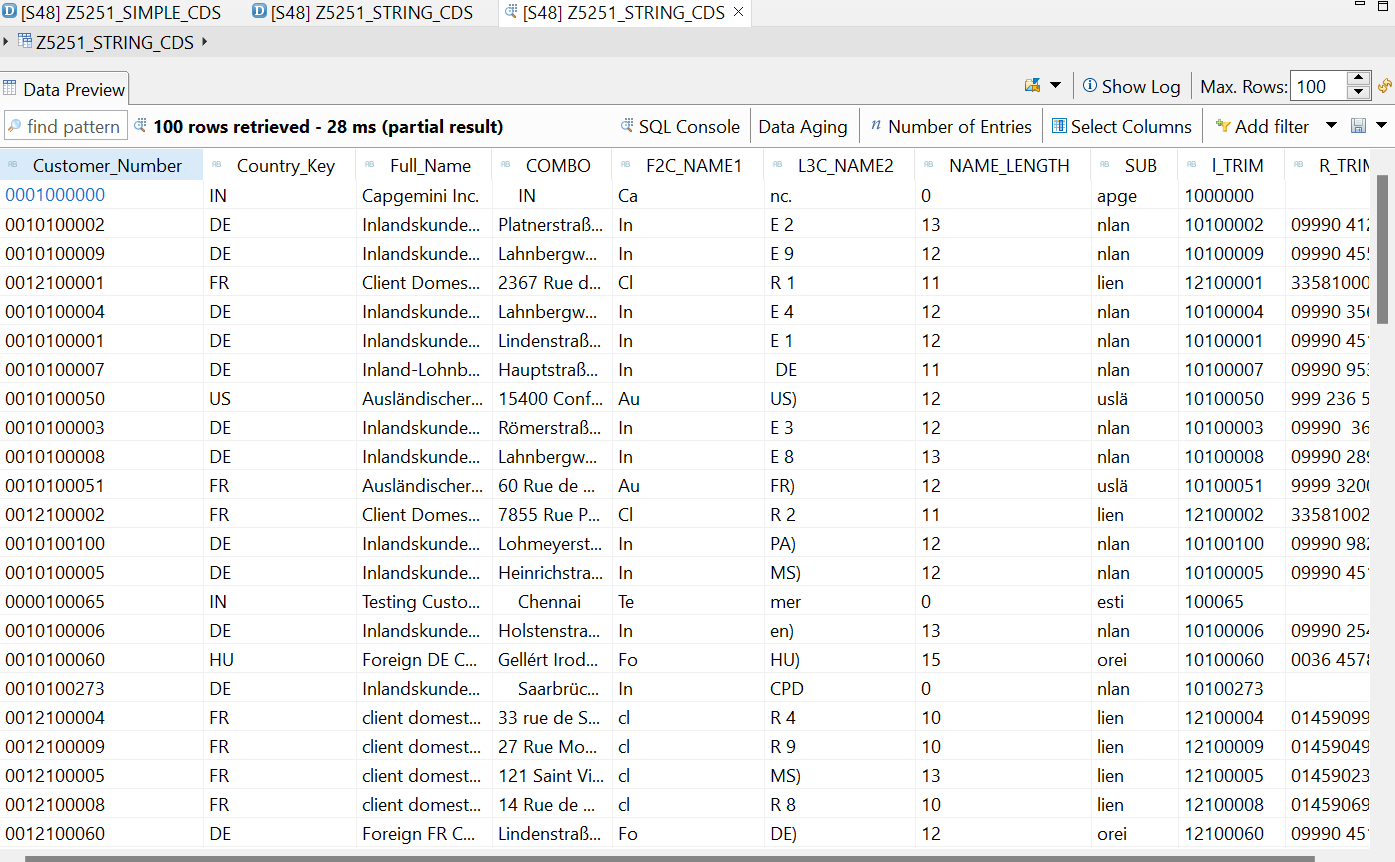
SUBSTRING**(**name1**,** 2**,** 4**)** **as** SUB**,**

LTRIM**(**kunnr**,** '0'**)** **as** l\_TRIM**,**

RTRIM**(**telf1**,** '4'**)** **as** R\_TRIM

**}**

OUTPUT:



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

Code:

@AbapCatalog.sqlViewName: 'Z5251\_PAR\_SQL'

@AbapCatalog.compiler.compareFilter: true

@AbapCatalog.preserveKey: true

@AccessControl.authorizationCheck: #NOT\_REQUIRED

@EndUserText.label: 'VIEW WITH PAR'

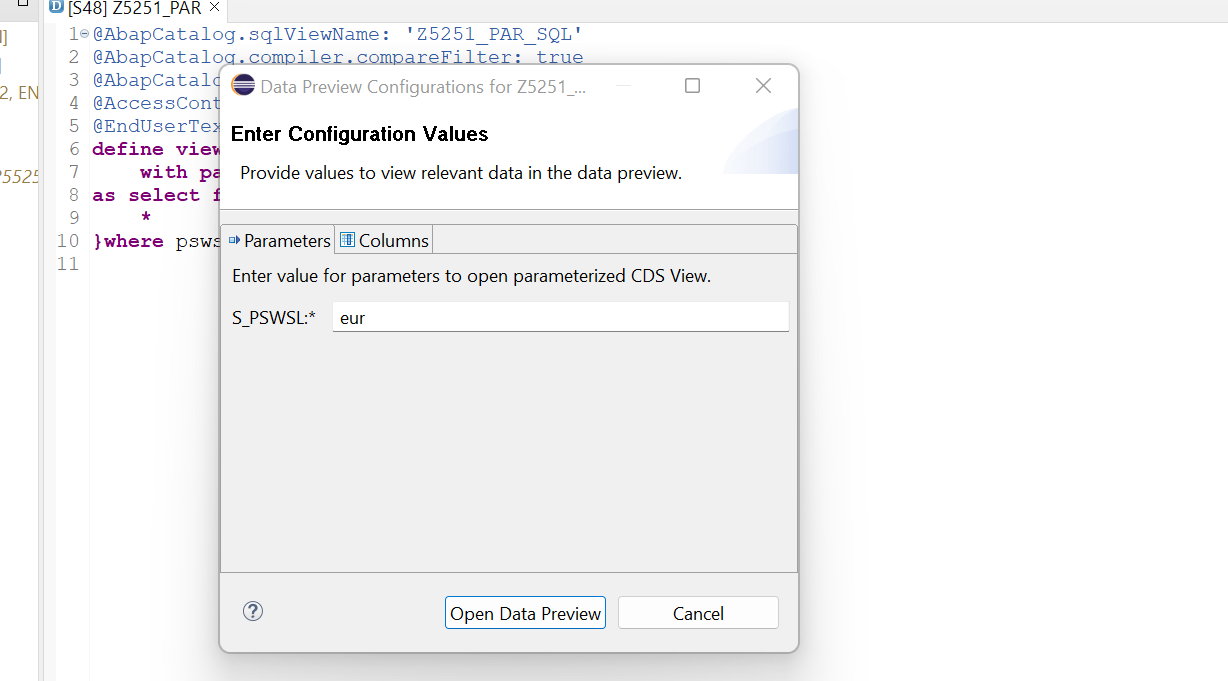
**define** **view** Z5251\_PAR

**with** **parameters** S\_PSWSL **:** *abap***.***cuky***(**5**)**

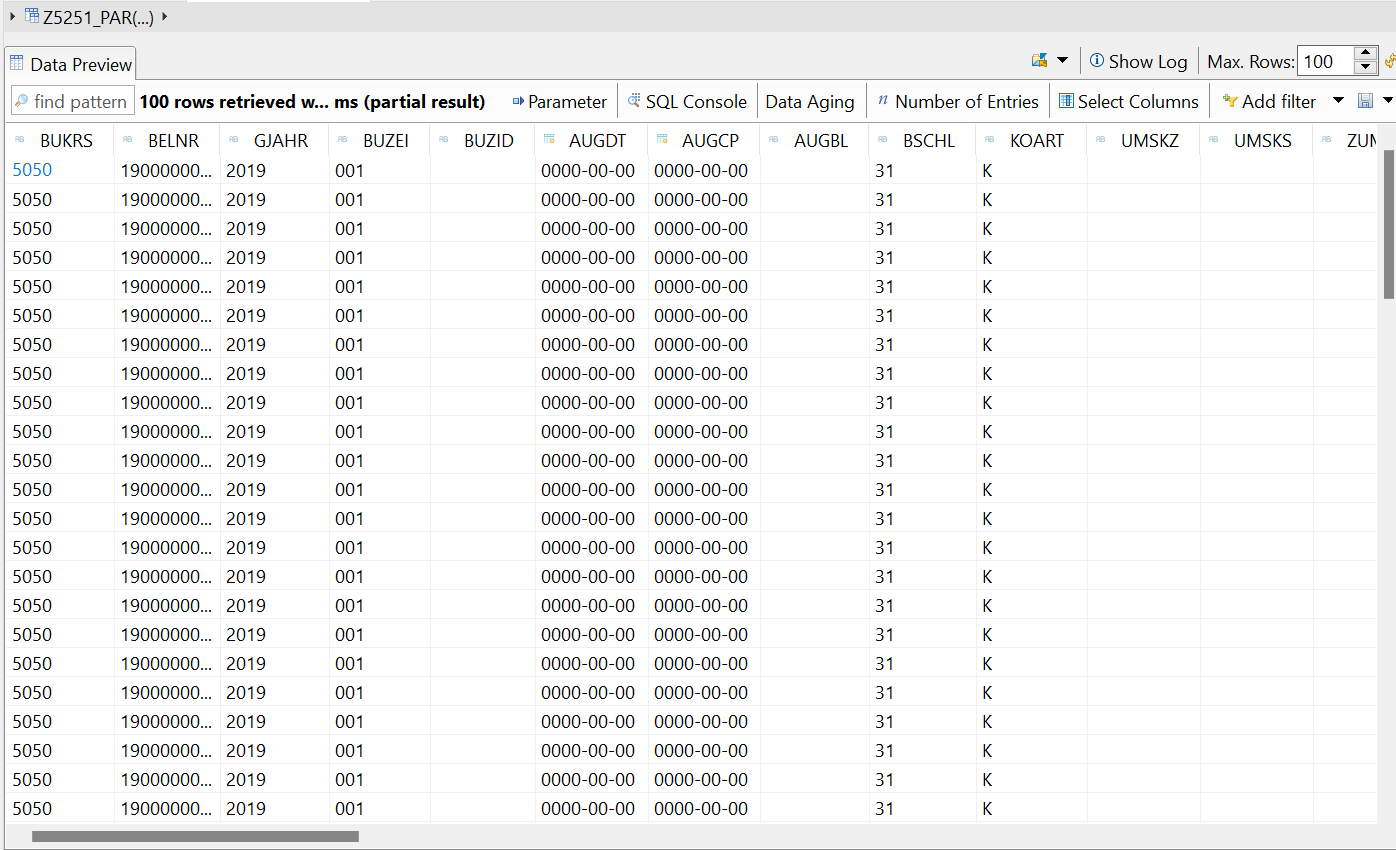
**as** **select** **from** bseg **{**

**\***

**}where** pswsl **=** **$parameters.**S\_PSWSL**;**



Output:



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

CLASS CODE:

CLASS Z5251\_CLASS1\_D14 DEFINITION

PUBLIC

FINAL

CREATE PUBLIC .

PUBLIC SECTION.

INTERFACES :IF\_AMDP\_MARKER\_HDB.

CLASS-METHODS: get\_vbap\_data

EXPORTING VALUE(lt\_VBAP) TYPE Z5236TTSTU\_VBAP.

PROTECTED SECTION.

PRIVATE SECTION.

ENDCLASS.

CLASS Z5251\_CLASS1\_D14 IMPLEMENTATION.

METHOD get\_vbap\_data BY DATABASE PROCEDURE FOR HDB LANGUAGE

SQLSCRIPT OPTIONS READ-ONLY

USING VBAP.

lt\_vbap = select vbeln, matnr, ZMENG from vbap;

ENDMETHOD.

ENDCLASS.Report:

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

\*& Report z5251\_amdp1

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

\* Description: \*

REPORT Z5251\_AMDP1.

Z5251\_CLASS1\_D14=>get\_vbap\_data( IMPORTING lt\_vbap = DATA(ex\_data) ).

TRY.

CALL METHOD cl\_salv\_table=>factory

IMPORTING

r\_salv\_table = DATA(lr\_salv)

changing

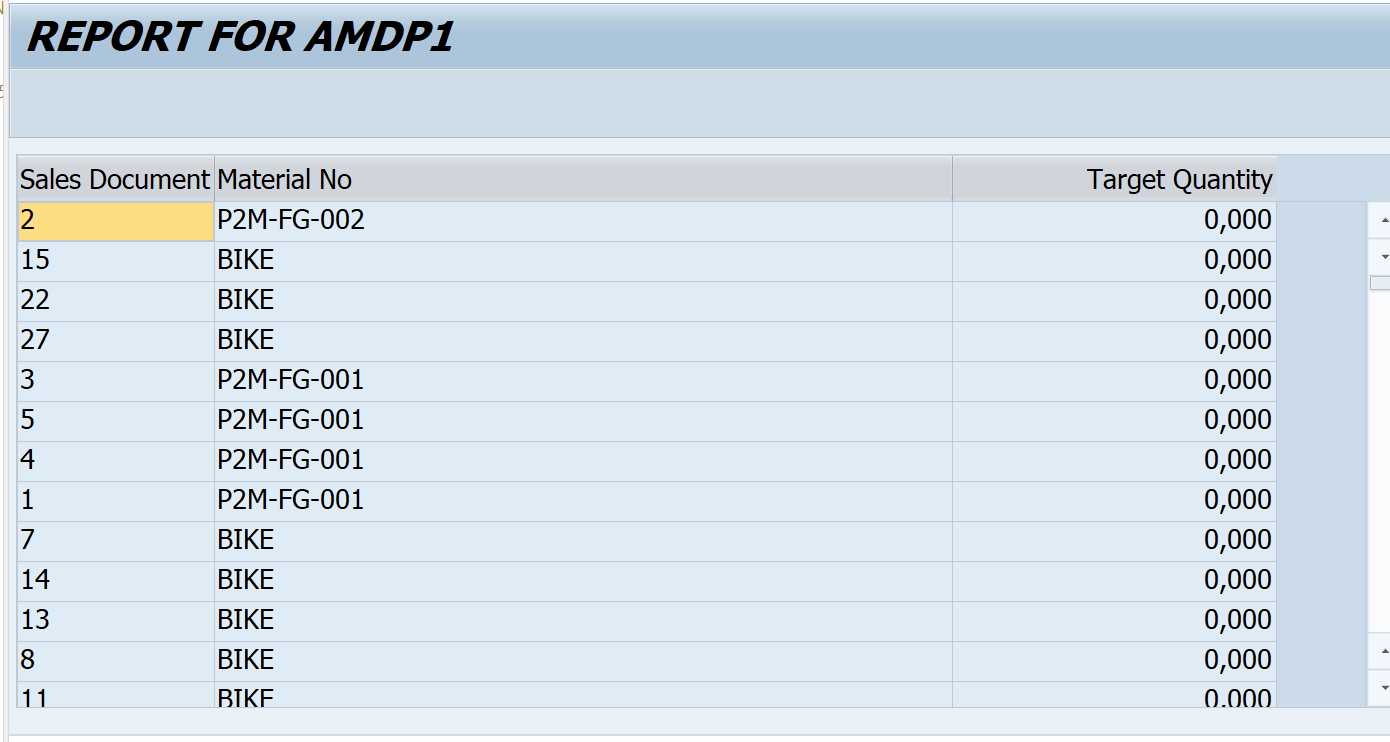
t\_table = ex\_data.

CATCH cx\_salv\_msg.

ENDTRY.

lr\_salv->display( ).

Output:



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

Class code:

CLASS Z5251\_CLASS2\_D14 DEFINITION

PUBLIC

FINAL

CREATE PUBLIC .

PUBLIC SECTION.

INTERFACES : IF\_AMDP\_MARKER\_HDB.

CLASS-METHODS:GET\_DATA

IMPORTING VALUE(IM\_MATNR) TYPE VBAP-MATNR

EXPORTING VALUE(ET\_RES) TYPE Z5236TTSRU\_IN.

PROTECTED SECTION.

PRIVATE SECTION.

ENDCLASS.

CLASS Z5251\_CLASS2\_D14 IMPLEMENTATION.

METHOD get\_data BY DATABASE PROCEDURE FOR HDB LANGUAGE

SQLSCRIPT OPTIONS READ-ONLY

USING VBAP VBAK.

ET\_RES = 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.

Report:

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

\*& Report z5251\_amdp2

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

REPORT Z5251\_AMDP2.

PARAMETERS: S\_MATNR TYPE VBAP-MATNR.

Z5251\_CLASS2\_D14=>get\_data( EXPORTING IM\_MATNR = S\_MATNR

IMPORTING ET\_RES = DATA(EX\_RES) ).

TRY.

CALL METHOD cl\_salv\_table=>factory

IMPORTING

r\_salv\_table = DATA(lr\_salv)

changing

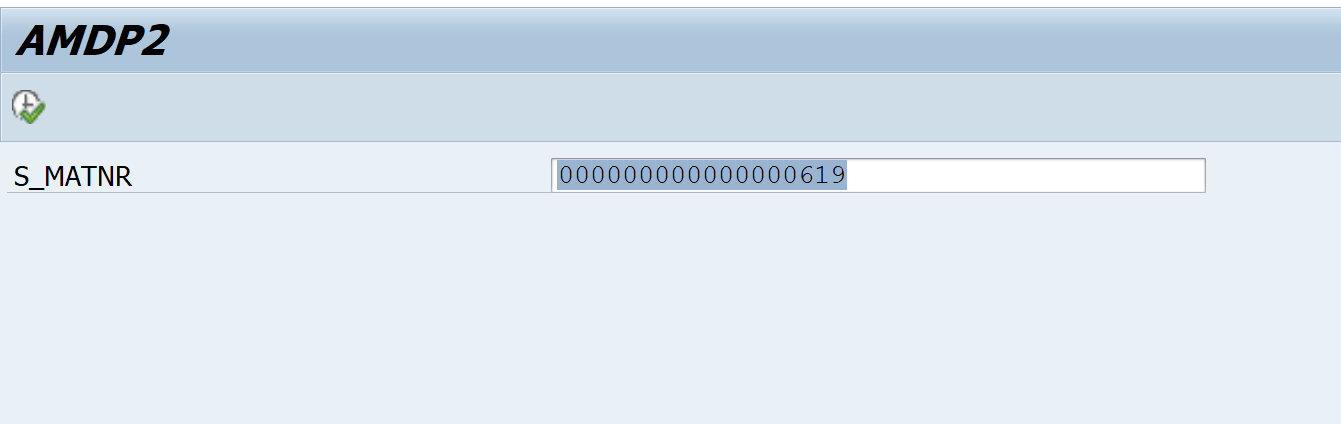
t\_table = ex\_res.

CATCH cx\_salv\_msg.

ENDTRY.

lr\_salv->display( ).

Selection screen:



Output:

