Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

ABAP/4
Module - 3 LAB BOOK

Table of Contents

Getting Started

1.1 Overview

This lab book is a guided tour for learning SAP ABAP. It comprises of assignments to be done. Refer the demos and work out the assignments given by referring the case studies which will expose you to work with Java applications.

1.2 Setup Checklist for SAP ABAP

Here is what is expected on your machine in order to work with lab assignment.

Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 2010 or higher.
- Memory: (8GB or more recommended)

Please ensure that the following is done:

- SAP GUI is installed
- Connection to the SAP Server is present

Lab 1-1 Modularization Techniques

Goals	• How to create the RFC connection and Call the RFC functions by using the RFC DESTINATION. • How to Create the Function Group and Function Modules. • How to Create and Call the Subroutines.
Time	120 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting to SAP server

Function Modules:

Assignment # 1:

Spell out numbers based on the user input value.

SPELL_AMOUNT This function module converts an amount or number into words based on the Language.

- Convert a number into words.

  To do this, the transfer parameters LANGUAGE and AMOUNT have to be entered.

Create an executable program to call the predefined function module **SPELL_AMOUNT**.

Eg: If the user input is **1000** (One Thousand) as showed in the below screen.

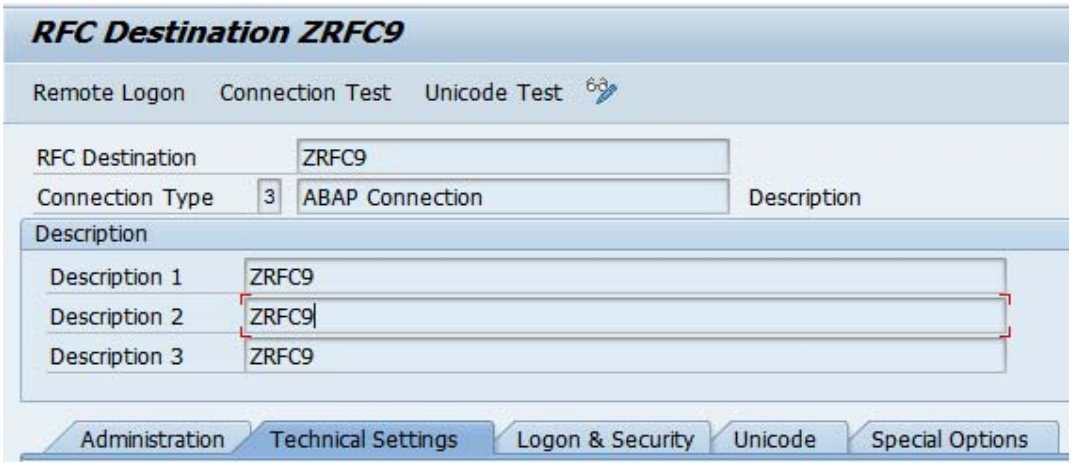
Expected output should be **ONE THOUSAND** in words for the Language key EN.

Assignment # 2:

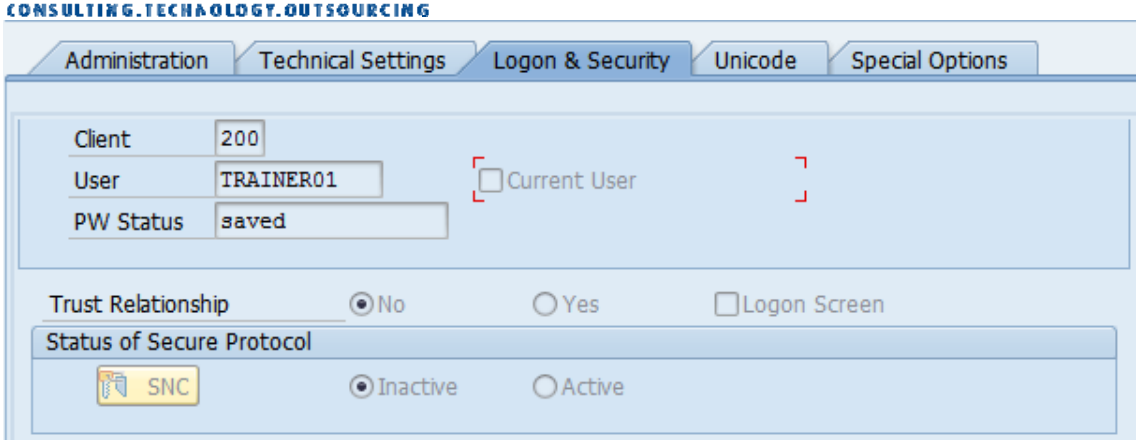
Create a RFC Connection between the two clients in the same or different application system.

Step # 1: Go to SM59 T-code and select the ABAP (Type- 3) connections and create the RFC.

Step # 2: Provide the RFC destination name, short description and click on the tab **Logon & Security.**



Step # 3: Provide the client number, user name and password for which client you want to connect the same or other application server.

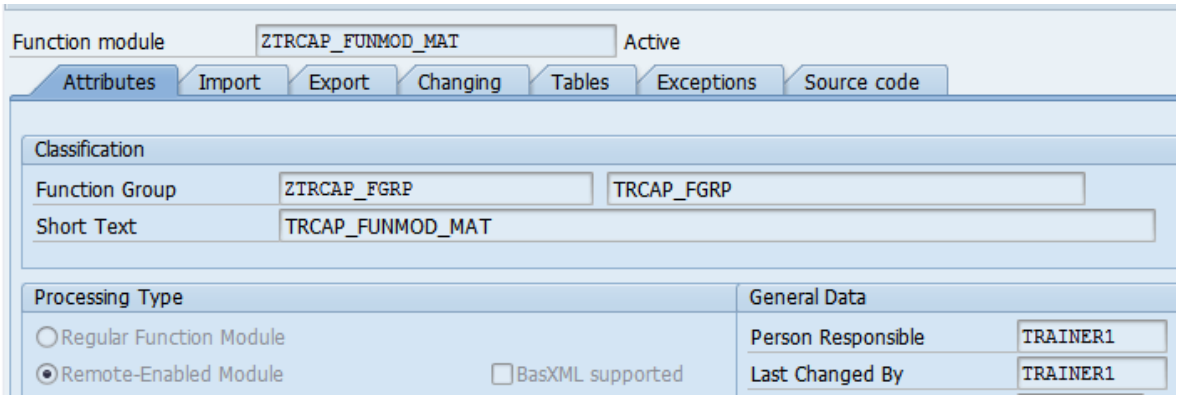


Step # 4: Save the details and Click on the Remote Logon. It will connect to other client.

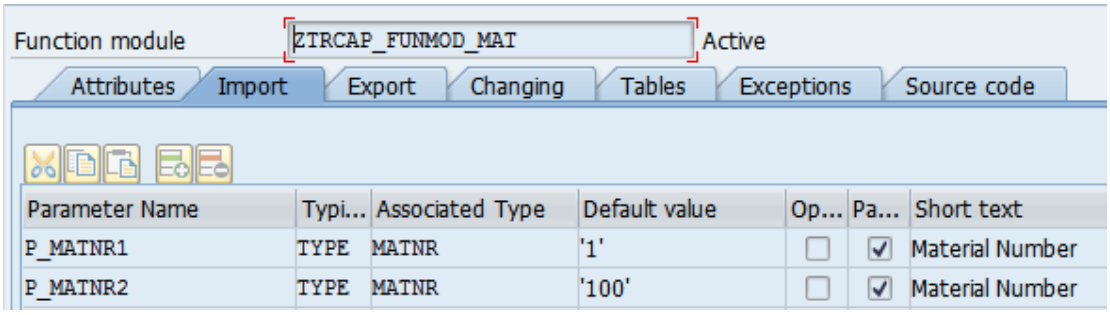
Assignment # 3:

1) Create a Function Group and RFC Function Module in Client 100 Outbound.
2) In Client 100 write an abap program to call the RFC function module to get the data remotely from other client 200.

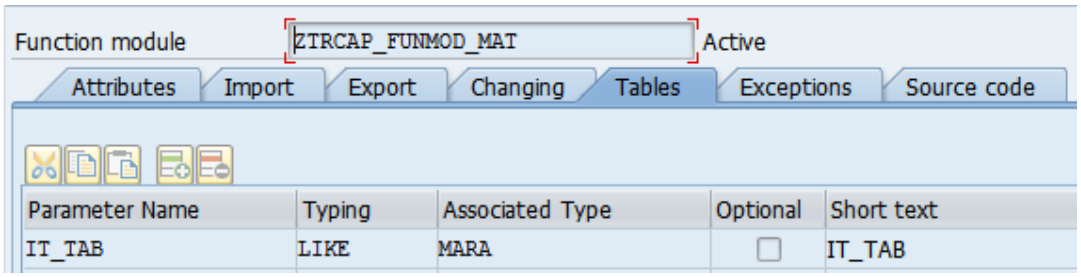
Step # 1: Go to SM37 T-Code and provide the function name, description and select the Remote-Enabled Module radio button in the **Attributes Tab.**



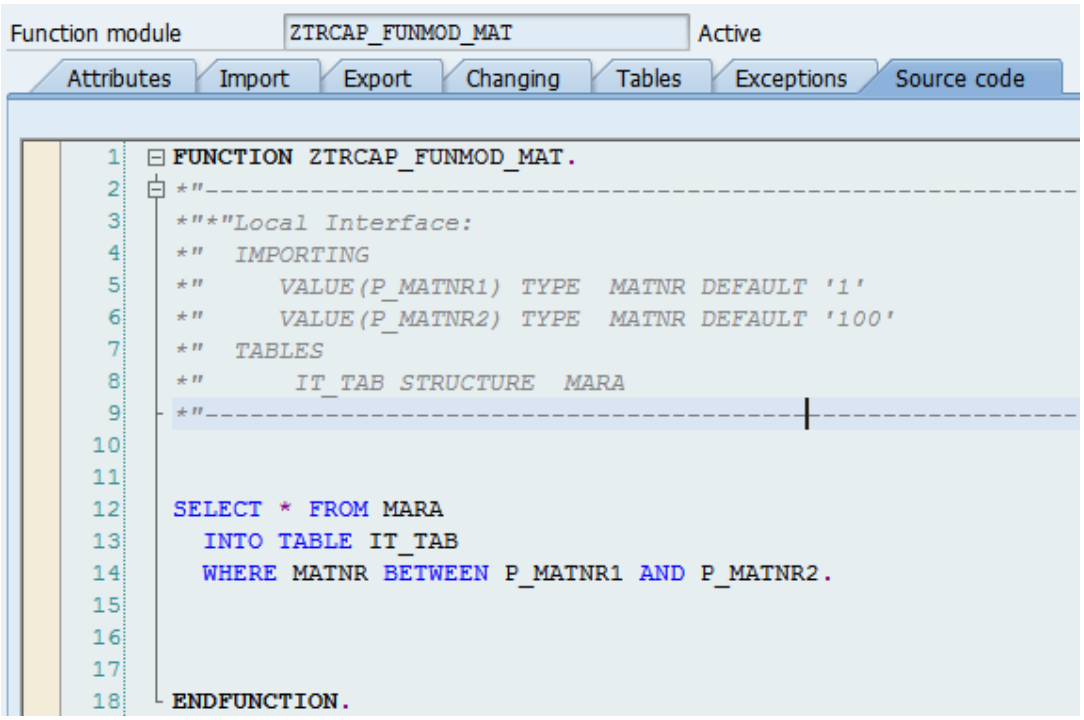
Step # 2: Create the two import parameters to accept the range of materials from the user.



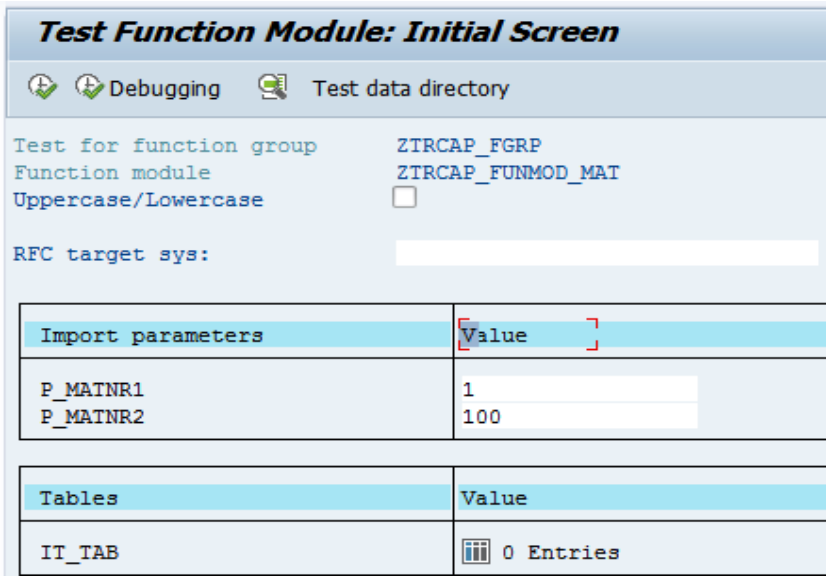
Step # 3: Create an internal table under the tables parameter based on the requirement.



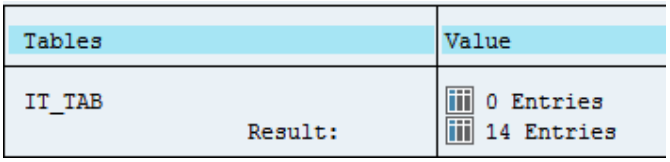
Step # 4: write the function module **source code** logic based on the import parameters and tables.



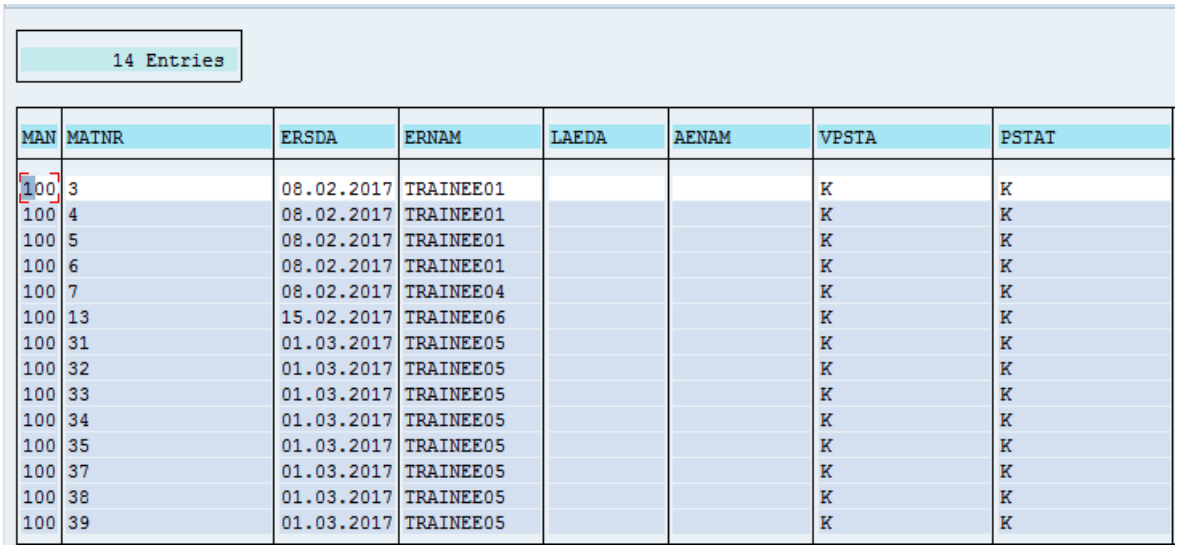
Step # 5: Execute the function module and provide the material value range in import parameters.



Step # 6: Click on the table entries icon.



Step # 7: you will get the data from the login client eg: client no 100.



MAN	MATNR	ERSDA	ERNAM	LAEDA	AENAM	VPSTA	PSTAT
100	3	08.02.2017	TRAINEE01			K	K
100	4	08.02.2017	TRAINEE01			K	K
100	5	08.02.2017	TRAINEE01			K	K
100	6	08.02.2017	TRAINEE01			K	K
100	7	08.02.2017	TRAINEE04			K	K
100	13	15.02.2017	TRAINEE06			K	K
100	31	01.03.2017	TRAINEE05			K	K
100	32	01.03.2017	TRAINEE05			K	K
100	33	01.03.2017	TRAINEE05			K	K
100	34	01.03.2017	TRAINEE05			K	K
100	35	01.03.2017	TRAINEE05			K	K
100	37	01.03.2017	TRAINEE05			K	K
100	38	01.03.2017	TRAINEE05			K	K
100	39	01.03.2017	TRAINEE05			K	K

Step # 8: Write an executable program in client 100 and execute it to get the data from the remote client 200 by using the RFC destination.

CALL FUNCTION 'ZTRCAP_FUNMOD_MAT' DESTINATION 'ZRFC9'(SM59 RFC Name)

Note: Afer call function with **DESTINATION <RFC NAME>** you will get the 200 client data, without that you will get the login clietn 100 data only.

```
*&---------------------------------------------------------------------*
*& Report ZTRCAP_FUNMOD1_MAT_PRG
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
REPORT ZTRCAP_FUNMOD1_MAT_PRG.
TABLES MARA.

DATA IT_TAB2 TYPE MARA OCCURS 0 WITH HEADER LINE.

SELECT-OPTIONS S_MATNR FOR MARA-MATNR.

CALL FUNCTION 'ZTRCAP_FUNMOD_MAT'  DESTINATION 'ZRFC9'
 EXPORTING
   P_MATNR1       = S_MATNR-LOW
   P_MATNR2       = S_MATNR-HIGH
  TABLES
    IT_TAB         = IT_TAB2
          .

LOOP AT IT_TAB2 INTO IT_TAB2.
  WRITE : / IT_TAB2-MANDT,
            IT_TAB2-MATNR,
            IT_TAB2-MBRSH,
            IT_TAB2-MTART,
            IT_TAB2-MEINS.
  ENDLOOP.
```


SUBROUTINES:

Assignment # 1:

Passing Parameters by Vaule
Create a simple program which accepts a material number. Write a subroutine which passes the material number by value and displays the following details regarding in the subroutine:
Material Number
Industry Sector
Material Type
Base UOM
Gross weight
Net Weight

Assignment # 2:

Passing Parameters by Reference
Make a copy of the above program and pass the material number by reference. Change the material number in the subroutine and display the details in the main program.

Assignment # 3:

Passing Structures
Create a simple program which accepts a material number. Write down a select query in the program which retrieves details of the Material Number in the structure.
Create a subroutine which receives the structure and displays the data.
Hint: Use Select Single to retrieve a single record

Assignment # 4:

Passing Internal Tables
Create a simple program which accepts a material number. Write down a select query in the program which retrieves details of the Material Number in the internal table.
Create a subroutine which receives the internal table and displays the data.
Note: Do various options of declaring internal table with/without header line.

Lab 2-1 Interactive List Events

Goals	• How to use an interactive list events.
Time	2 Hours
Lab Setup	• Connectivity to SAP server • Login details for connecting to SAP server

Assignment # 1:

Create an Interactive list event report with Hide statement for maximum 2 secondary list levels.

Create an executable program to prepare the range of materials in the basic list from the MARA table based on the user selection of materials, prepare at least two secondary list reports accordingly.

Program Logic Hints:

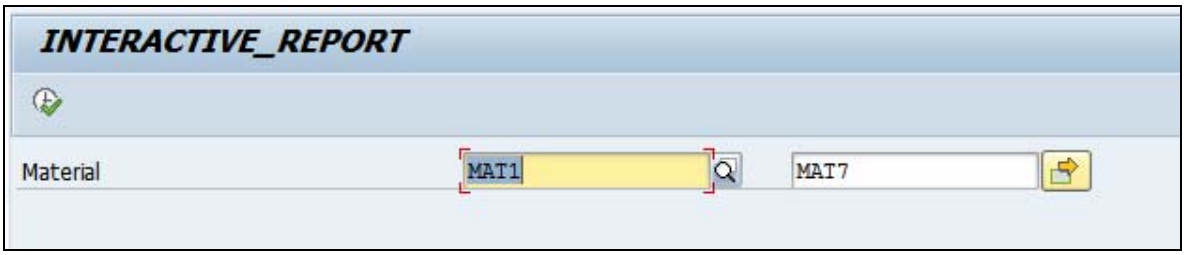
- Declare the events in the report.
  START-OF-SELECTION. END-OF-SELECTION.
  TOP-OF-PAGE. END-OF-PAGE. AT LINE-SELECTION.
  TOP-OF-PAGE DURING LINE-SELECTION.

- Use the HIDE Statement Inside the loop for each list.

Reference T-Codes and Tables:

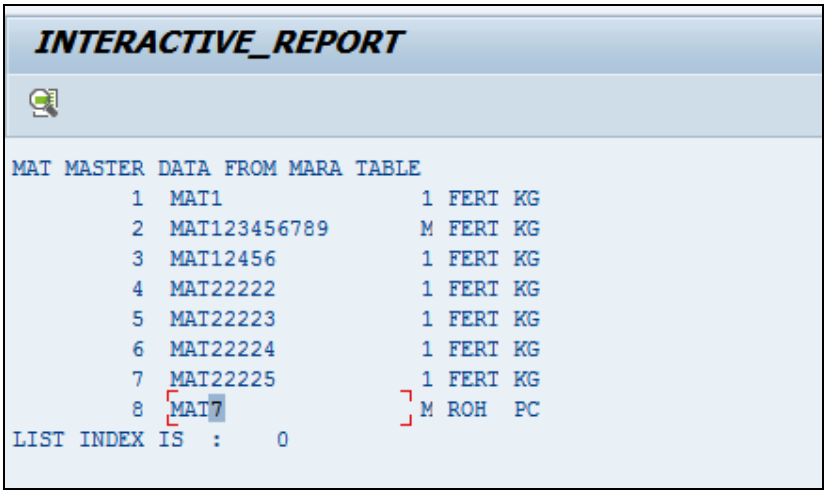
T-Codes: SE38 and MM03. **Tables:** MARA, MARC and MAKT.

Step # 1. Go SE38 T-Code to create an executable program and the Input should be Materials range and it should be an obligatory.

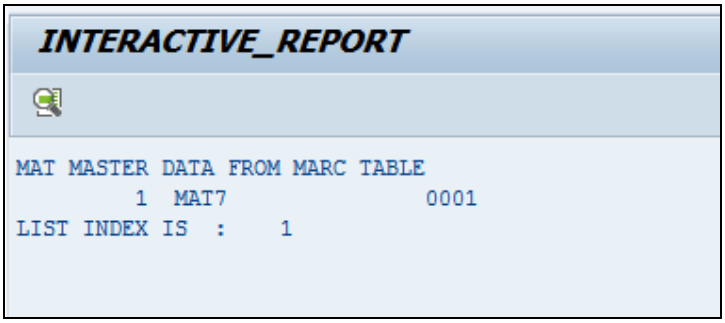


Step # 2. Prepare the Basic List (INDEX 0) fields from the MARA table.

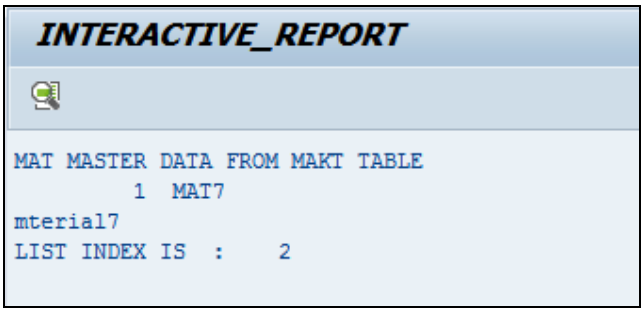
Field names: MATNR, MBRSH, MTART and MEINS.



Step # 2. Prepare the First Secondary list (INDEX 1) fields from the MARC Table.
Field Names: MATNR and WERKS.



Step # 2. Prepare the Second Secondary list (INDEX 2) fields from the MAKT Table.
Field Names: MATNR, MAKTX and SPRAS.



Assignment # 2:

Create an At User-Command Interactive report.

Create an executable program to prepare the basic list and user command by using the menu painter.

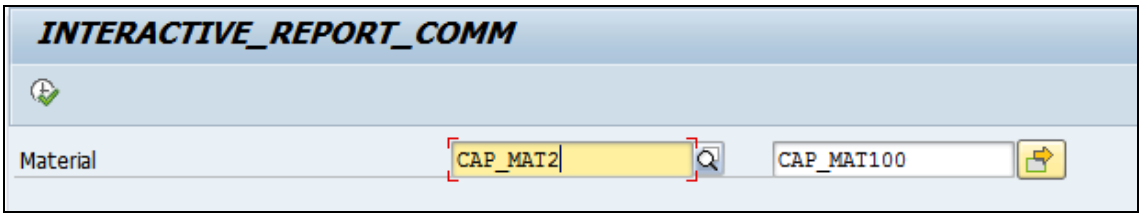
Program Logic Hints:

- Declare the events in the report.
  START-OF-SELECTION. END-OF-SELECTION. TOP-OF-PAGE. AT USER-COMMAND.
- Use the SET PF-STATUS to design the menu painter.

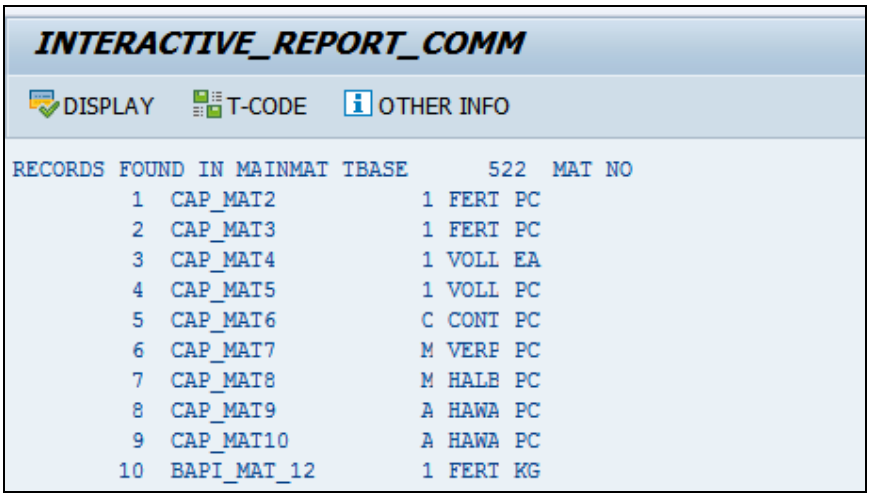
Reference T-Codes and Tables:

T-Codes: SE38 and MM03 **Tables:** MARA, MARC and MAKT.

Step # 1. Go SE38 T-Code and create an executable program and the Input should be Materials range and it should be an obligatory.



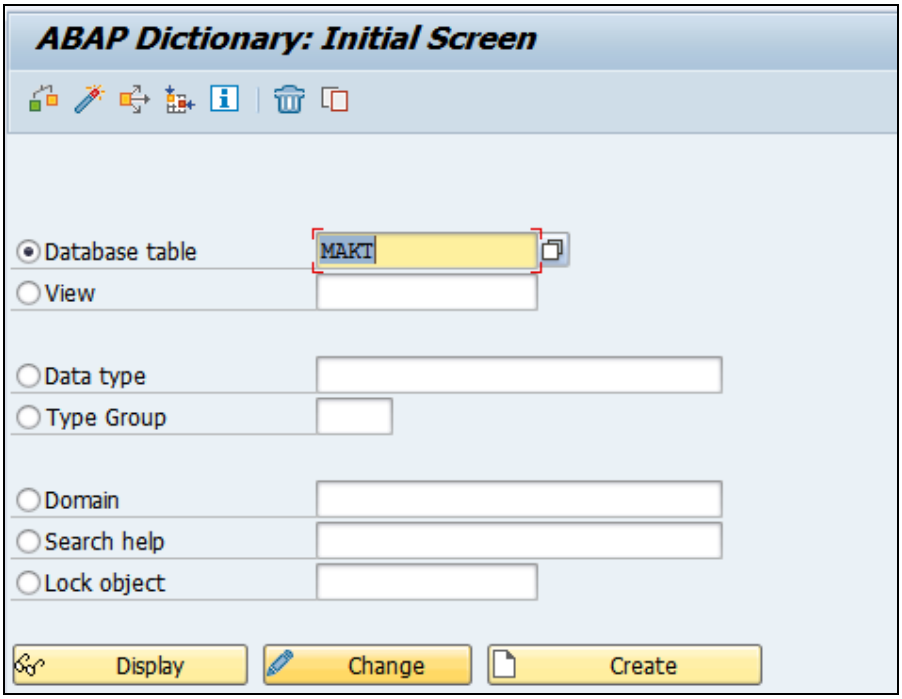
Step # 2. Display the PUSHBUTTONS , T-CODE and OTHER INFO by using the Application Tool bar (T-Code SE41 Menu Painter) in the report output and prepare the basic list data from MARA Table for MATNR, MBRSH , MTART and MEINS fields based on the select options.



Step # 3. When user selects the **DISPLAY Pushbutton**, Get the data from MARC and MAKT tables based on the select options range by using the for all Entries Concept.

Display the following fields output:
MARC Table: MATNR, WERKS, PSTAT, LVORM, BSTMI and BSTMA.
MAKT Table: MATNR, SPRAS, MAKTX and MAKTG.

Step # 4. When user selects the **T-Code Pushbutton**, Call the transaction SE11.



Assignment # 3:

Create an Interactive report by using the GET CURSOR Technique.

Reference T-Codes and Tables:

T-Codes: SE38 and ME23N and MK03 **Tables:** EKKO, EKPO and LFA1.

Go SE38 T-Code and create an executable program with simple interactive report by using the

GET CURSOR FIELD FNAM VALUE FVAL.

Display the basic list report with Vendor no(LFA1-LIFNR) and Name (LFA1-NAME1) and Purchase order (EKKO-EBELN).

Vendor No	Vendor Name	Purchase Order No
Eg:		
V-101	V-100ABC	4500000001
V-102	V-101PQR	4500000002
V-103	V-102XYZ	4500000003

```
IF FNAM = 'WA_TAB-VENDOR'. " If user clicked on the VENDOR NO.
```

Display the Vendor Master details by using the LFA1 table and display the fields LIFNR AND NAME1 and any other 5 Fields output.

```
ELSEIF FNAM = 'WA_TAB-EBELN'. " If user clicked on the PURCHASE ORDER NO.
```

Display the Purchase order details by using the EKKO table and display the Fields EBELN and any other 5 Fields output.

```
ENDIF.
```

Assignment # 4:

Create an Interactive report by using the model dialog box.

Go SE38 T-Code and create an executable program with model dialog box for both the basic list and secondary lists.

Consider the sales order header and line items tables i.e VBAK and VBAP. Display the output for any 5 Fields from each tables.

Lab 3-1 Module Pool Programming

Goals	• Screen designing by using subscreens, tab strips, table controls and user defined transaction codes to update the data in the ztables.
Time	4 Hrs.
Lab Setup	• Connectivity to SAP server • Login details for connecting SAP server

Assignment # 1:

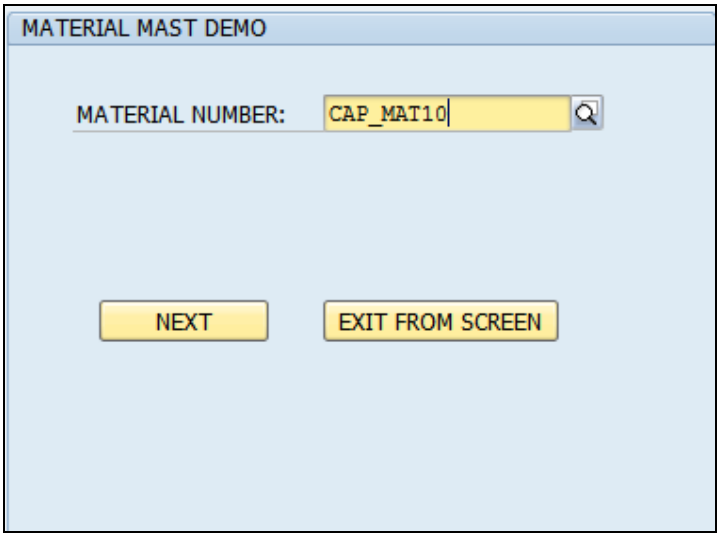
Create a simple module pool transaction to display the single material output.

Reference T-Codes and Tables:

T-Codes: SE38, SE51, SE93 and MM03 **Tables:** MARA.

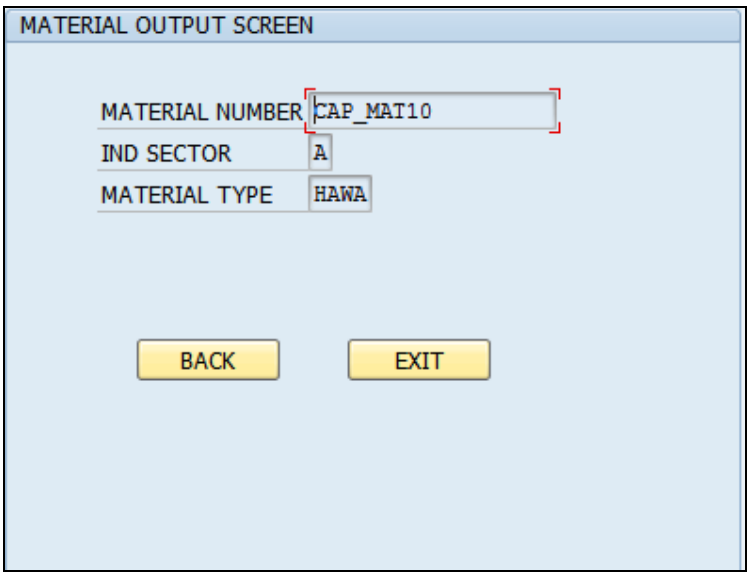
Go to SE38 and SE51 T-Codes to create module pool program.

Step # 1: In the first screen accept the single material number from the user and when you click on the **NEXT** button the output should be displayed in the second screen and when you click on the **EXIT FROM SCREEN** button **L**eave from the screen.



Step # 2: In the second screen display the single material output based on the input provided in the first screen and when you click on the **BACK** Button control should be back to first screen to modify the input of the material number and when you click on **EXIT** Button leave from the program.

MATERIAL NUMBER, IND SECTOR and MATERIAL TYPE Fields should be in display mode (Output fields) and user cannot edit the fields at runtime.



Assignment # 2:

Create a module pool program to display the table control output.

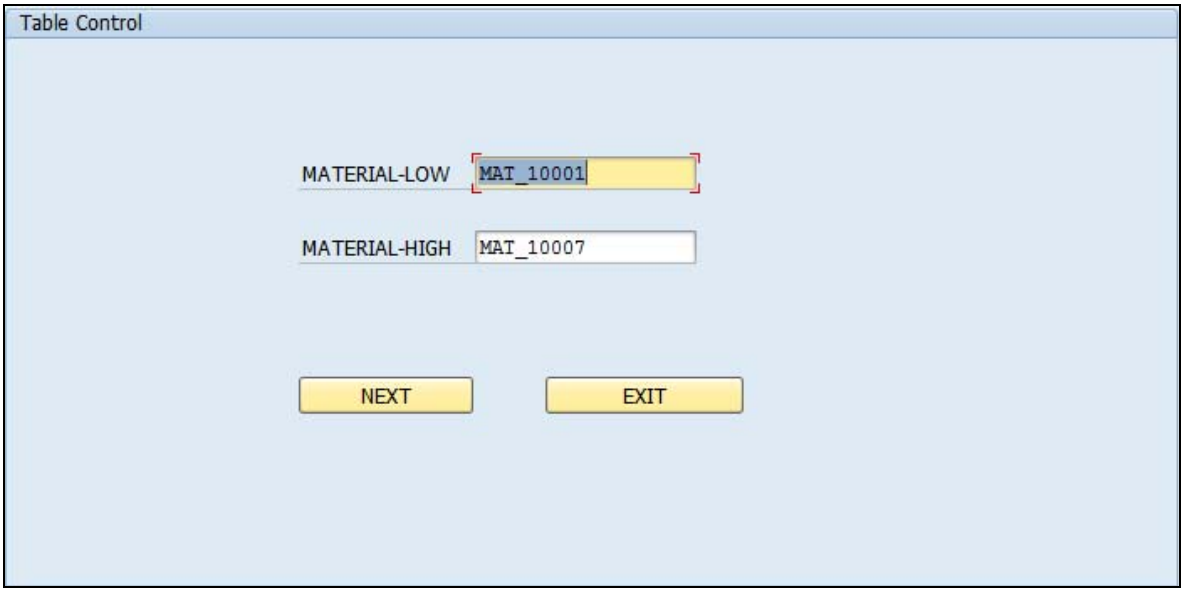
Reference T-Codes and Tables:

T-Codes: SE38, SE51, SE93 and MM03 **Tables:** MARA, MARC And MAKT.

Go to SE38 and SE51 T-Codes to create module pool program.

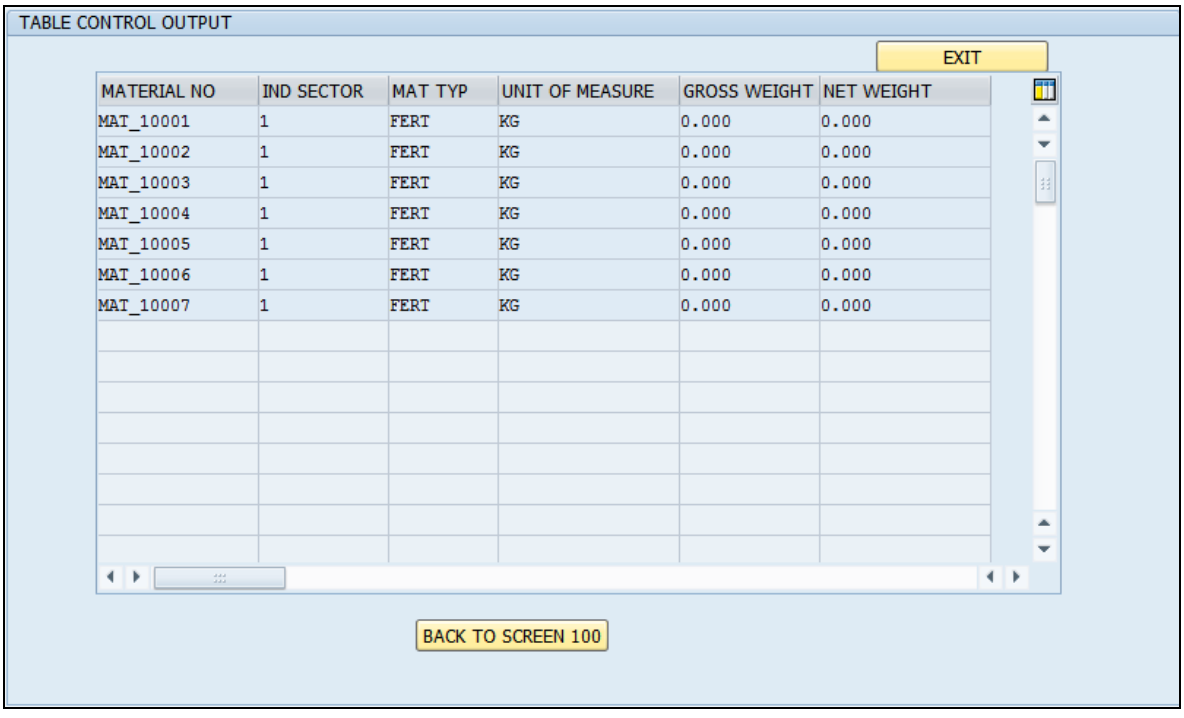
In the First Screen accept material range from the user and when you click on the NEXT pushbutton the output should be displayed in the second screen as a table control.

When you click on the EXIT button Leave from the screen.



In the Second Screen When you click on the **BACK TO SCREEN 100** push button, go back to the initial screen to modify the input of the material range and when you click on the **EXIT** push button leave from the program.

Table Control output should be in display mode user cannot be edit the fields at runtime.



MATERIAL NO	IND SECTOR	MAT TYP	UNIT OF MEASURE	GROSS WEIGHT	NET WEIGHT
MAT_10001	1	FERT	KG	0.000	0.000
MAT_10002	1	FERT	KG	0.000	0.000
MAT_10003	1	FERT	KG	0.000	0.000
MAT_10004	1	FERT	KG	0.000	0.000
MAT_10005	1	FERT	KG	0.000	0.000
MAT_10006	1	FERT	KG	0.000	0.000
MAT_10007	1	FERT	KG	0.000	0.000

Assignment # 3:

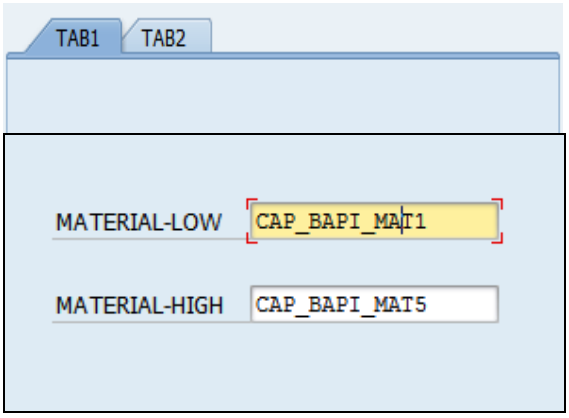
Create a module pool program to display the tabstrip control output.

Go to SE38 and SE51 T-Codes to create module pool program.

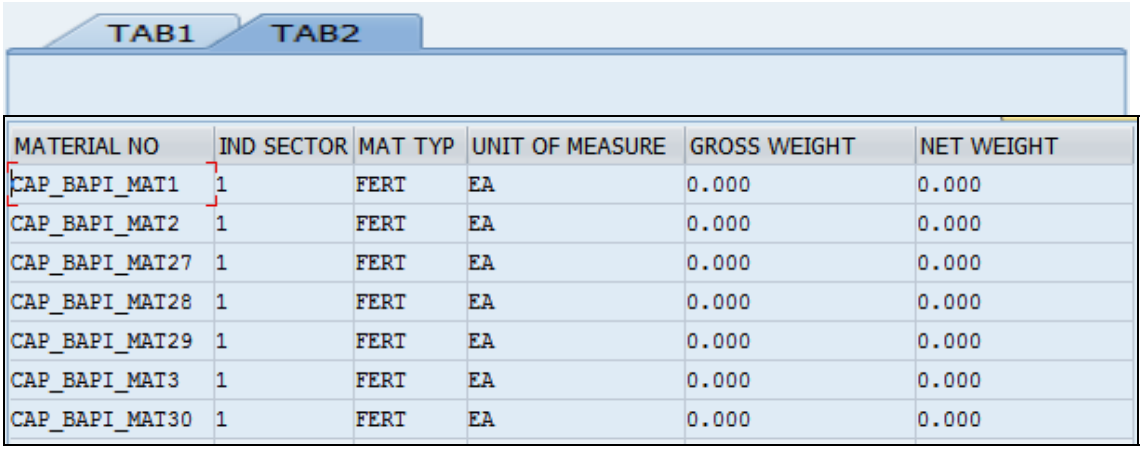
Reference T-Codes and Tables:

T-Codes: SE38, SE51, SE93 and MM03 **Tables:** MARA.

Step # 1: In the first screen create the tabstrip control in **TAB1** to accept the material range from the user and when you click on the **TAB2** pushbutton the output should be displayed in the second screen as a table control format.



Step # 2: In the second screen create table control and get the data from mara table based on the material range provided in TAB1.



MATERIAL NO	IND SECTOR	MAT TYP	UNIT OF MEASURE	GROSS WEIGHT	NET WEIGHT
CAP_BAPI_MAT1	1	FERT	EA	0.000	0.000
CAP_BAPI_MAT2	1	FERT	EA	0.000	0.000
CAP_BAPI_MAT27	1	FERT	EA	0.000	0.000
CAP_BAPI_MAT28	1	FERT	EA	0.000	0.000
CAP_BAPI_MAT29	1	FERT	EA	0.000	0.000
CAP_BAPI_MAT3	1	FERT	EA	0.000	0.000
CAP_BAPI_MAT30	1	FERT	EA	0.000	0.000

Assignment # 4:

Create a module pool program to work with ztable DML operations.

Go to SE38 and SE51 T-codes to create module pool program.

Reference T-Codes and Tables:

T-Codes: SE38, SE51 and SE93. **Tables:** zemp.

Design the module pool screen to update the zemployee table information.

Note: Select the **Display/Maintenance Allowed with restrictions** option under **Delivery and Maintenance** tab in the table (SE11) to prevent the data Load/Insert directly on to the table**.**