ABAP/4
ABAP Part I LAB BOOK

Table of Contents

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 7 or higher.
- Memory: (1GB or more recommended)

Please ensure that the following is done:

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

Goals	• Create simple programs in ABAP editor
Time	60 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting to SAP server

1. Write a program to display the Empid, EmpName, Emp_addr in chain statement using write.

Hint: All the variable should be in declared using DATA statement.

2. Write a program to display the System Date, System Time, Current User ID,Current report name by using single write statement.

3. Write a program to display the first 10 passengers from sbook table. Consider any 10 fields to display the output.

4. Write a program to display the data from sbook table. The data should be filtered by airline and ordered by customid.
The airline should be accepted through parameters or select-options.
Consider any 10 fields to display the output.

Use Select .. EndSelect in 3 and 4.

After that, use internal tables also in 3 and 4.

Goals	• Create Tables, Views and Search Helps
Time	60 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting to SAP server

1. Create the tables with the following structure. Name of the table as z_xxxxxx_emp. Where xxxxx is your empcode.

z_xxxxxx_emp

Field Name	Data Element	Domain	Domain Data Type and length
EMPNO (PK)	Z_XX_DEEMPNO	Z_XX_DEMPNO	NUMC 4
ENAME	Z_XX_DEENAME	Z_XX_DENAME	CHAR 10
JOB	Z_XX_DEJOB	Z_XX_DJOB	CHAR 9
MGR	Z_XX_DEMGR	Z_XX_DMGR	NUMC 4
HIREDATE	Z_XX_DEHIREDATE	Z_XX_DHIREDATE	DATS
SAL	Z_XX_DESAL	Z_XX_DSAL	DECIMAL 7 2
COMM	Z_XX_DECOMM	Z_XX_DCOMM	DECIMAL 7 2
DEPTNO (FK)	Z_XX_DEDEPTNO	Z_XX_DDEPTNO	NUMC 2

z_xxxxxx_dept

Field Name	Data Element	Domain	Domain Data Type and length
DEPTNO(PK)	Z_XX_DEDEPTNO	Z_XX_DEDEPTNO	NUMC 2
DNAME	Z_XX_DEDNAME	Z_XX_DEDNAME	CHAR 14
LOC	Z_XX_DELOC	Z_XX_DELOC	CHAR 13

Note: Data Element and Domains for deptno field is the same in both tables.

z_xxxxxx_salgrade

Field Name	Data Element	Domain	Domain Data Type and length
GRADE	Z_XX_DEGRADE	Z_XX_DGRADE	NUMC 2
LOSAL	Z_XX_DELOSAL	Z_XX_DLOSAL	DECIMAL 7 2
HISAL	Z_XX_DEHISAL	Z_XX_DHISAL	DECIMAL 7 2

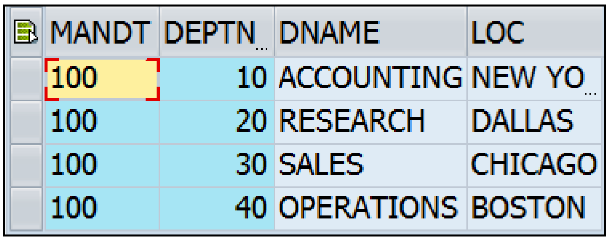
Make all three fieldsas PK.

Where:

1) XXXXXX is your empcode

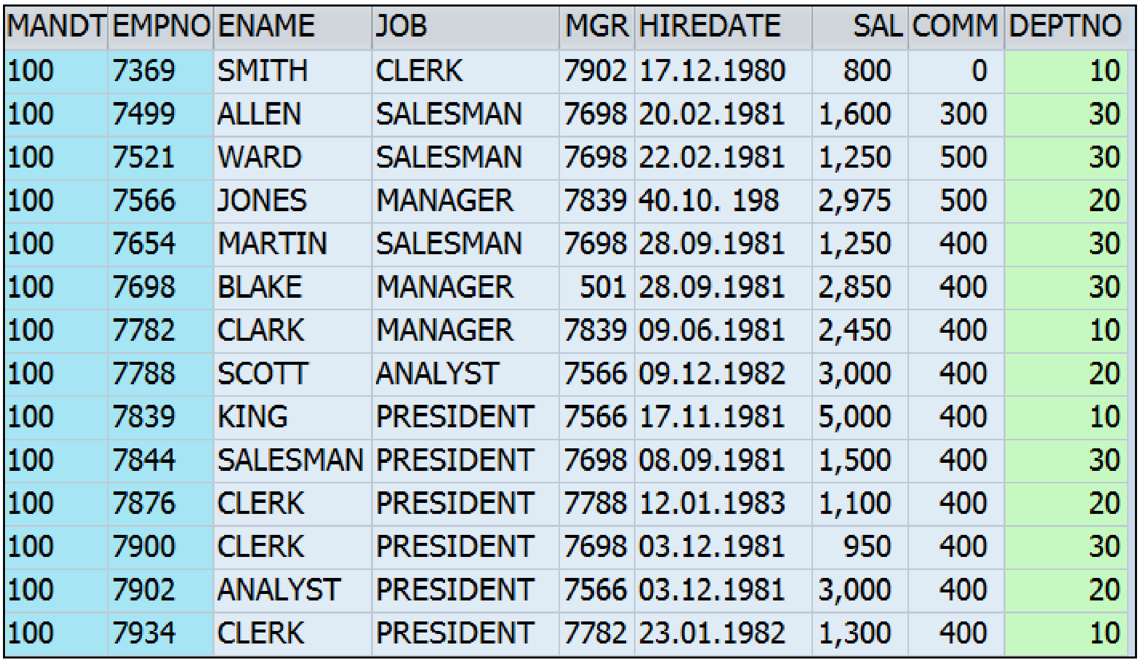
This is to ensure uniqueness of your table name, domain and data element name.

2. Write down an ABAP program to insert the following records in the Dept table.
   Use Internal Table to insert the records.



MANDT	DEPTN…	DNAME	LOC
100	10	ACCOUNTING	NEW YO…
100	20	RESEARCH	DALLAS
100	30	SALES	CHICAGO
100	40	OPERATIONS	BOSTON

3. Write down an ABAP program to insert the following records in the Emp table.
   OR Use Table Maintainance generator/Internal table to insert records.



MANDT	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
100	7369	SMITH	CLERK	7902	17.12.1980	800	0	10
100	7499	ALLEN	SALESMAN	7698	20.02.1981	1,600	300	30
100	7521	WARD	SALESMAN	7698	22.02.1981	1,250	500	30
100	7566	JONES	MANAGER	7839	40.10. 198	2,975	500	20
100	7654	MARTIN	SALESMAN	7698	28.09.1981	1,250	400	30
100	7698	BLAKE	MANAGER	501	28.09.1981	2,850	400	30
100	7782	CLARK	MANAGER	7839	09.06.1981	2,450	400	10
100	7788	SCOTT	ANALYST	7566	09.12.1982	3,000	400	20
100	7839	KING	PRESIDENT	7566	17.11.1981	5,000	400	10
100	7844	SALESMAN	PRESIDENT	7698	08.09.1981	1,500	400	30
100	7876	CLERK	PRESIDENT	7788	12.01.1983	1,100	400	20
100	7900	CLERK	PRESIDENT	7698	03.12.1981	950	400	30
100	7902	ANALYST	PRESIDENT	7566	03.12.1981	3,000	400	20
100	7934	CLERK	PRESIDENT	7782	23.01.1982	1,300	400	10

4. Create an ABAP program as per the following specifcations:

ABAP Inputs:

The ABAP program will have parameters for employee number , employee name and salary with the definitions as per the database table fields.

Screen Validations

Mandatory input for all parameters.
Employee name should not be blank and Salary cannot be zero.

Employee number can contain only numbers.

Main Processing Logic :

Read the Emp table created above using the Employee number, which has been input.

If matching entry exists, then check the input name and salary. If at least one of them is different than db record, then record should be modified with changed values.

If matching entry for the accepted Employee number does not exist, new entry should be created in table with input values. You may hardcode the values for remaining fields(Job, Mgr, Hiredate, Comm, Deptno) here.

Output Format:

If record has been modified, display 'Record Updated'.

If record has been added, display 'Record added'.

Test Conditions

Report should be run with values for which entries exists in the db and also new values for which entries do not exist, so that both addition and modification of records can be tested.

Error checking: If employee code contains characters other then numbers, then suitable error message should be displayed. (Define Text elements for message text).

Hints

For the above program ABAP make use of following:

SELECT, INSERT, UPDATE.

5. Create a copy of the above assignment. Use only MODIFY instead of INSERT/UPDATE.

6. Create two ztables (Eg: ZXXX_SCARR, ZXXX_SPFLI) by using predefined table fields of SCARR AND SPFLI. Select at least 5 fields from each table and load some data into that two ztables.

Note: To enter the data/entries to ZTables Ref: SCARR and SPFLI tables content/entries.

Step # 2: Maintain foreign key between the two ztables.

Step # 3: Create the data base view for the above two ztables.

Step # 4: Write an ABAP code and get the data from the database view (from the step # 3) and display the output.

7. Create help view and search help.

Step # 1: Create two ztables (Eg: ZXXX_SFLIGHT, ZXXXX_SBOOK) by using predefined table fields of SFLIGHT AND SBOOK. Select at least 5 fields from each table and load some data into that two ztables.

Step # 2: Maintain foreign key between the two ztables.

Step # 3: Create the help view for the above two ztables.

Step # 4: Create the elementary search help and provide the help view name in the selection method of search help.

Step # 5: Attach the search help to the primary keys of the both the ztables.

8. Create a maintainance view based on ZXXX_SFLIGHT, ZXXXX_SBOOK tables created above.

Use the Table Maintainance generator to see if records can be inserted in the base table ZXXX_SFLIGHT, ZXXXX_SBOOK though the maintainance view.

Lab 3-1 Internal tables

Goals	• Retrive records from Internal Table
Time	60 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting to SAP server

1. Write an ABAP program which will retrieve 20 records of the DB table SBOOK and insert in it a Standard Internal table. Name the internal table as itsbook. Do not define any key for the internal table.
   Retrieve only the following fields carrid, connid,fldate,bookid,customid,Airline currency, booking date,passenger name
   Write separate subroutines to :
   1. Display all records from the internal table
   2. Display only records from 2 to 5
   3. Display the 5th Record
   4. Display the record where name of passenger is Adam Heller
   5. Modify the ename of the 5th Record as Steve
   6. Delete the record where ename is Adam Heller
   7. Delete the 10th Record
   8. Delete records from 12 to 16

   In all the above cases display all the records after perform each of the above tasks.

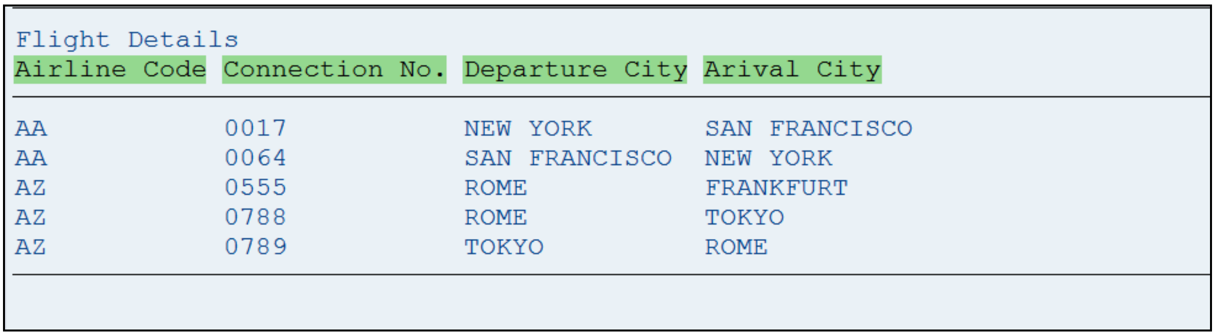
2. Create a copy of the above assignment. Modify the above internal table to define the key as passenger name.
   Retrieve the record with passenger name is Adam Heller.
   Hint: Use the table key clause

3. Create a copy of Assignment1. Declare the table as a sorted table and perform all the operations as mentioned in it.

4. Create a copy of the above assignment1. Declare the table as a hashed table and perform all the operations as mentioned in it. Note your observations.

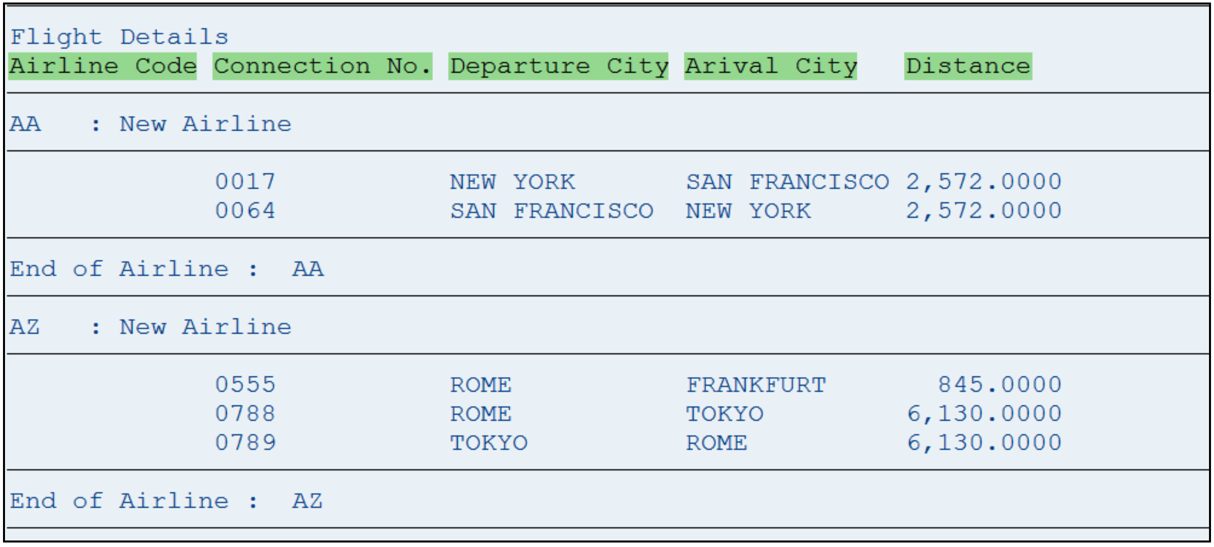
Goals	• Use control break logic to display data from Internal table and other commands. To work with an Internal table commands INSERT, INSERT Multiple, UPDATE, MODIFY and DELETE.
Time	60 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting to SAP server

1. Create an ABAP program to display output as shown below.
Hint: Use At First, At Last. Retreive records from DB table SPFLI.



```
Flight Details
Airline Code Connection No. Departure City Arival City

AA           0017           NEW YORK       SAN FRANCISCO
AA           0064           SAN FRANCISCO  NEW YORK
AZ           0555           ROME           FRANKFURT
AZ           0788           ROME           TOKYO
AZ           0789           TOKYO          ROME
```

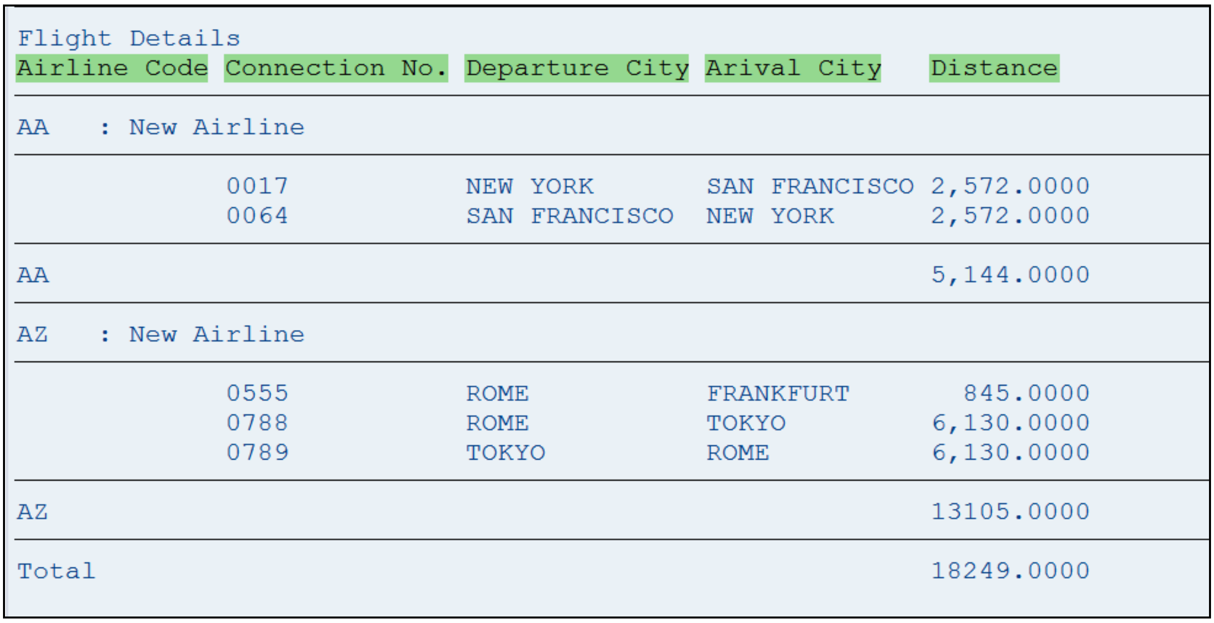
2. Create an ABAP program to display output as shown below.
Hint: Use At First, At Last, At New. Retreive records from DB table SPFLI.



```
Flight Details
Airline Code Connection No. Departure City Arival City    Distance

AA    : New Airline

             0017           NEW YORK       SAN FRANCISCO 2,572.0000
             0064           SAN FRANCISCO  NEW YORK      2,572.0000

End of Airline :  AA

AZ    : New Airline

             0555           ROME           FRANKFURT       845.0000
             0788           ROME           TOKYO         6,130.0000
             0789           TOKYO          ROME          6,130.0000

End of Airline :  AZ
```

3. Create an ABAP program to display output as shown below.

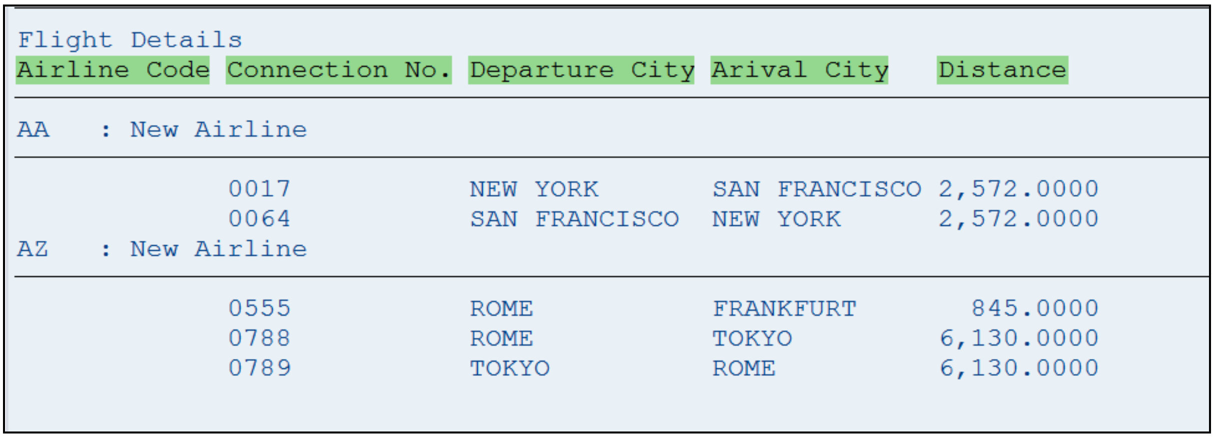
   Hint: Use At First, At Last, At New, Sum. Retreive records from DB table SPFLI.



```
Flight Details
Airline Code Connection No. Departure City Arival City    Distance

AA    : New Airline

             0017           NEW YORK       SAN FRANCISCO 2,572.0000
             0064           SAN FRANCISCO  NEW YORK      2,572.0000

AA                                                       5,144.0000

AZ    : New Airline

             0555           ROME           FRANKFURT       845.0000
             0788           ROME           TOKYO         6,130.0000
             0789           TOKYO          ROME          6,130.0000

AZ                                                       13105.0000

Total                                                    18249.0000
```

4. Create an ABAP program to display output as shown below.

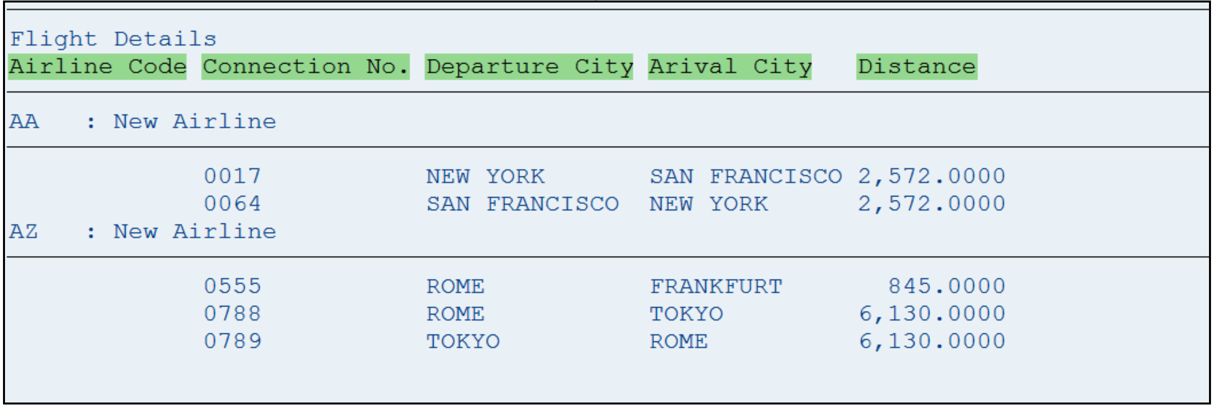
   Hint: Use At First, At Last, At New, On Change. Retreive records from DB table SPFLI.



```
Flight Details
Airline Code Connection No. Departure City Arival City    Distance

AA    : New Airline

             0017           NEW YORK       SAN FRANCISCO 2,572.0000
             0064           SAN FRANCISCO  NEW YORK      2,572.0000
AZ    : New Airline

             0555           ROME           FRANKFURT       845.0000
             0788           ROME           TOKYO         6,130.0000
             0789           TOKYO          ROME          6,130.0000
```

5. Create an ABAP program to display output as shown below.

   Hint: Use At First, At Last, At New, On Change. Retreive records from DB table SPFLI



```
Flight Details
Airline Code Connection No. Departure City Arival City    Distance

AA    : New Airline

             0017           NEW YORK       SAN FRANCISCO 2,572.0000
             0064           SAN FRANCISCO  NEW YORK      2,572.0000
AZ    : New Airline

             0555           ROME           FRANKFURT       845.0000
             0788           ROME           TOKYO         6,130.0000
             0789           TOKYO          ROME          6,130.0000
```

6. Create an ABAP program to display data from SPFLI(carrid, connid) and SFLIGHT(carrid, connid, fldate) table.
   The program should accept the cityfrom from the SPFLI table and display the corresponding records from the SFLIGHT table. Use joins.

7. Create an ABAP program to display data from SPFLI(carrid, connid) and SFLIGHT(carrid, connid, fldate) table.
   The program should accept the cityfrom from the SPFLI table and display the corresponding records from the SFLIGHT table. Use FOR ALL ENTRIES.

Goals	• Create an ABAP Program for String Operations
Time	90 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting SAP server

1. Create an ABAP Program with the following parameters.

P_STRING1 - input string. The input should be taken in LOWERCASE.
FLAGS should be displayed as radiobutton. Depending upon the value of the selected FLAG, corresponding function will be performed on the input string.

T_FLAG – to convert the string to UPPERCASE
L_FLAG – to return the length of the string
S_FLAG – to remove leading zeroes in the string
O_FLAG – to return subset of string starting from offset P_OFFSET and having length P_LENGTH.

P_OFFSET and P_LENGTH are parameters to take in the offset and length of the substring.

Screen Validations

Offset and Length should be numeric.

In case O_Flag not set, both offset and length must be 0.
O_Flag set, at least length must be non-zero positive value.
O_Flag set, then offset + length should be <= length of string.
S_Flag set, then string must contain leading zeroes.

Main Processing Logic:

Depending upon the FLAG, corresponding operation should be performed on the string.

Output Format:

If there is an error during validations, suitable error message should be displayed.
Or If the validations are successful, then the resultant string should be displayed after the operation is performed.

Test Conditions

Testing should be performed for all values of flag.

Error checking: Check out all scenarios where error will occur and check whether appropriate error message is displayed or not (Define text elements for message text).

Hints:

For this ABAP make use of following:
RADIOBUTTON GROUP, TRANSLATE, STRLEN, SHIFT.

Use subroutines for performing each of the operations.

2. Create an ABAP program with the following parameters.

Define P_STRING – input string default value .
‘000000000075001234, Material No: 0000000000014634566, 53’.

Screen-validations:

Check if atleast 1 comma exists in the input string.

Main Processing Logic:

Split the input string at comma into 3 different variables.

Search 2nd of these variables for string ‘Material No:’.

Move the Material no. into another variable.

Output Format:

Write the extracted material no. to the screen.

Hints:

Use SPLIT, SEARCH, STRLEN, commands

3. Create an ABAP Program.

It will have following parameters.

Provide String_1 and String_2 value as a default in the program and user can change the values at runtime with own input values.

Screen-validations:

Check atleast one space should be exists in the input value of string_1.
String_1 and String_2 should be obligatory.

Main Processing Logic:

Accept two input strings (String_1 and String_2) from the user at rutime by using parameters and display these two string values at the list output as a heading.

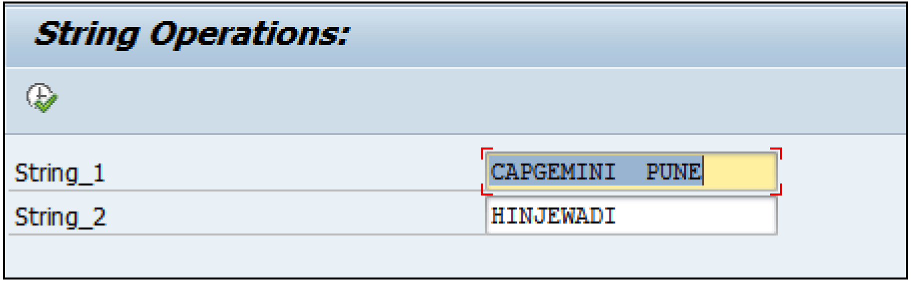
Hints:

# 1. Use CONCATENATE, TRANSLATE, CONDENSE, NO-GAPS, STRING LENGTH, SHIFT AND SHIFT BY PLACES commands to display the main logic.

# 2. Use split command by using Internal tables .

Input Format: Format of the input screen like below with proper text and title by using the text elements and list headings.

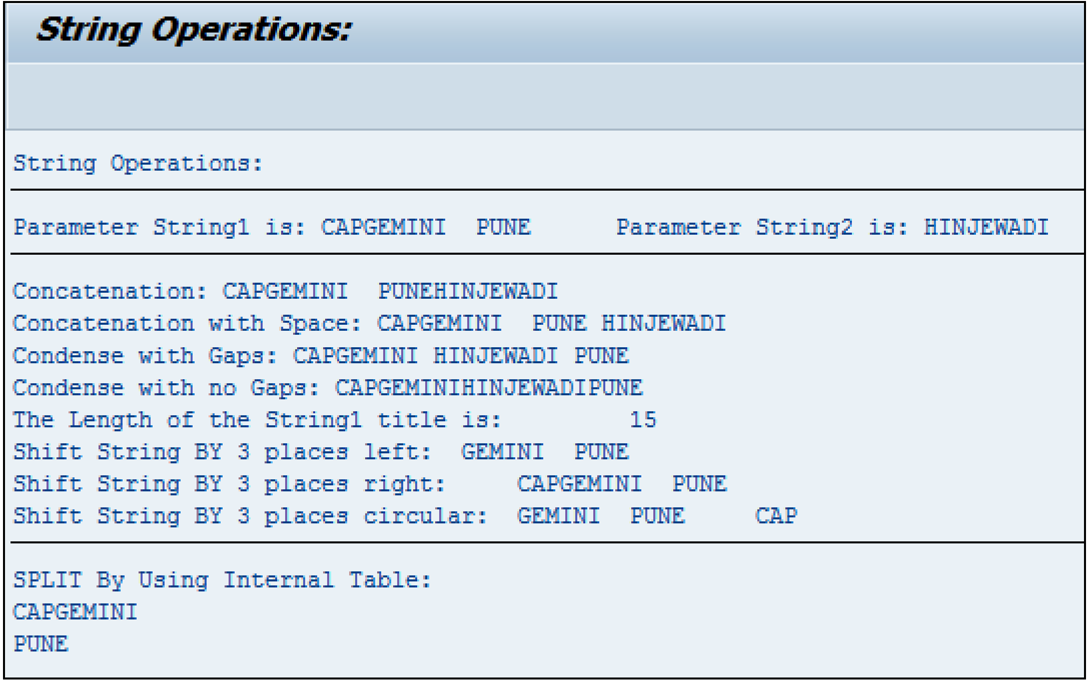
Input Screen:



String Operations:

String_1 CAPGEMINI PUNE
String_2 HINJEWADI

Output Format: the output format should be like as shown below.
Output Screen:



String Operations:

```
String Operations:

Parameter String1 is: CAPGEMINI  PUNE      Parameter String2 is: HINJEWADI

Concatenation: CAPGEMINI  PUNEHINJEWADI
Concatenation with Space: CAPGEMINI  PUNE HINJEWADI
Condense with Gaps: CAPGEMINI HINJEWADI PUNE
Condense with no Gaps: CAPGEMINIHINJEWADIPUNE
The Length of the String1 title is:         15
Shift String BY 3 places left:  GEMINI  PUNE
Shift String BY 3 places right:     CAPGEMINI  PUNE
Shift String BY 3 places circular:  GEMINI  PUNE     CAP

SPLIT By Using Internal Table:
CAPGEMINI
PUNE
```

Lab 6-1 Common Control Statements

Goals	• Use common control statements
Time	60 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting to SAP server

1. Write an ABAP program to calculate the Grade of the students, based on the input marks.

A. If the input Marks are > 90 - Grade 5.
B. If the input Marks are between 80 and 90 – grade 4.
C. If the input Marks are between 70 and 80 – grade 3 .
D. If the input Marks are between 60 and 70 – grade 2
E. If the input Marks are < 60 grade 0.

Goals	• How to Create the Function Group and Function Modules. • How to Create and Call the Subroutines.
Time	60 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting to SAP server

1. 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

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.

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

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.

5. Do assignment1 using fucntion Module. An ABAP program accepts a Material number and passes it to a function module. The function module returns the below details to the ABAP program.

   Industry Sector ,Material Type, Base UOM , Gross weight, Net Weight

   The ABAP program then displays the details.

   Declare variable in the Top Include in the Function Group.

6. Modify the above program to use subroutines in the Function Group.

Lab 8-1 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

1. 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.

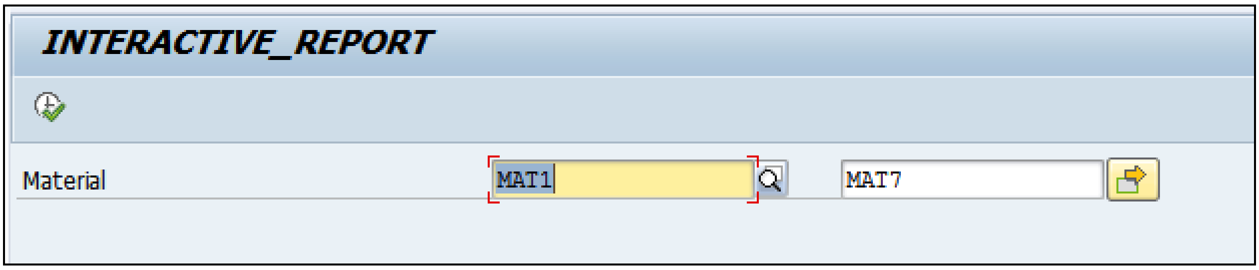
Program Logic Hints:

- 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.

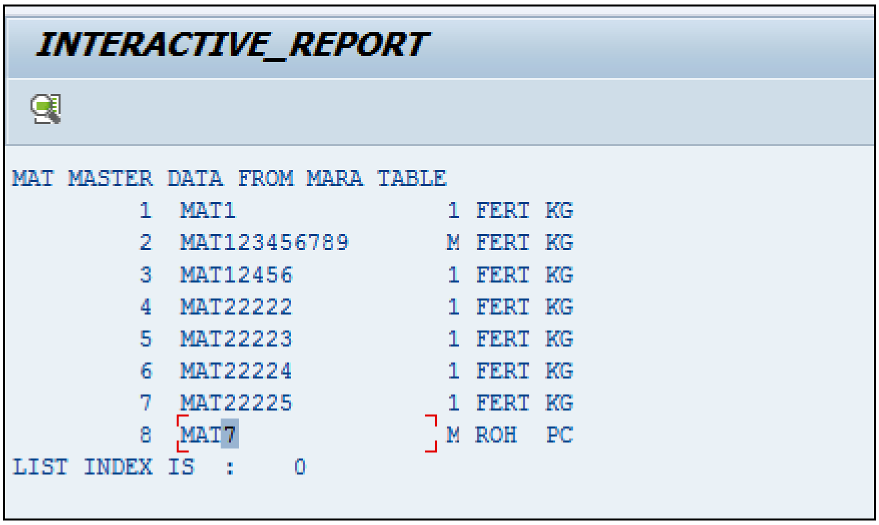
Step # 1. Go SE38 T-Code to create an executable program and the Input should be Materials range and it should be an obligatory.



INTERACTIVE_REPORT

Material MAT1 MAT7

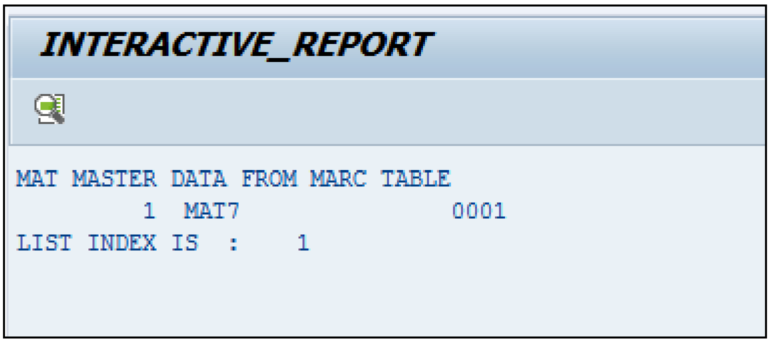
Step # 2. Prepare the Basic List (INDEX 0) fields from the MARA table.
Field names: MATNR, MBRSH, MTART and MEINS.



INTERACTIVE_REPORT

```
MAT MASTER DATA FROM MARA TABLE
         1  MAT1                 1 FERT KG
         2  MAT123456789         M FERT KG
         3  MAT12456             1 FERT KG
         4  MAT22222             1 FERT KG
         5  MAT22223             1 FERT KG
         6  MAT22224             1 FERT KG
         7  MAT22225             1 FERT KG
         8  MAT7                 M ROH  PC
LIST INDEX IS  :    0
```

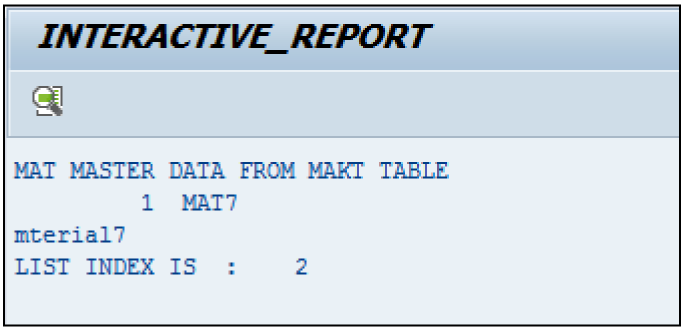
Step # 2. Prepare the First Secondary list (INDEX 1) fields from the MARC Table.
Field Names: MATNR and WERKS.



INTERACTIVE_REPORT

```
MAT MASTER DATA FROM MARC TABLE
         1  MAT7                 0001
LIST INDEX IS  :    1
```

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



INTERACTIVE_REPORT

```
MAT MASTER DATA FROM MAKT TABLE
         1  MAT7
mterial7
LIST INDEX IS  :    2
```

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

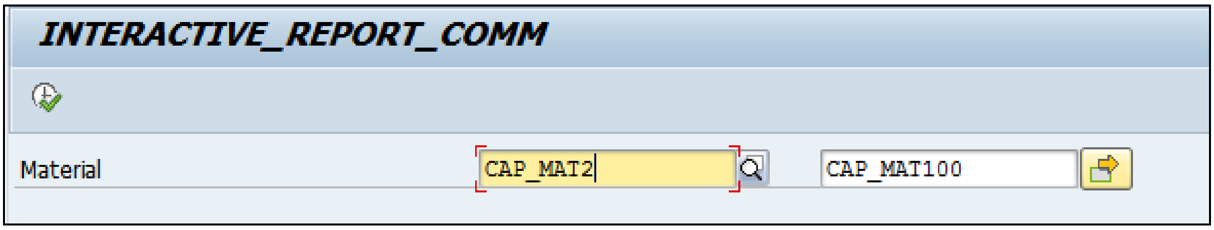
Program Logic Hints:

- 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.

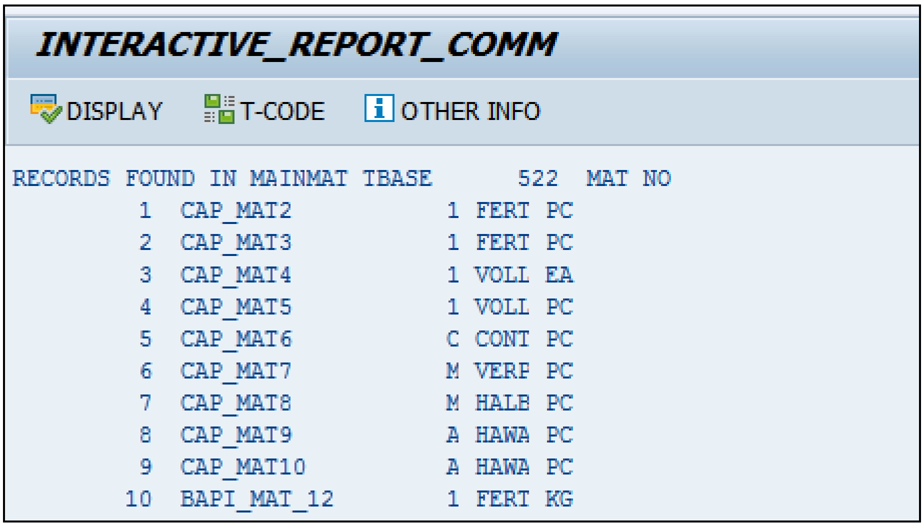
Step # 1. Go SE38 T-Code and create an executable program. Input should be Materials range and it should be an obligatory.



INTERACTIVE_REPORT_COMM

Material CAP_MAT2 to CAP_MAT100

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.
Prepare the basic list data from MARA Table for MATNR, MBRSH , MTART and MEINS fields based on the select options.



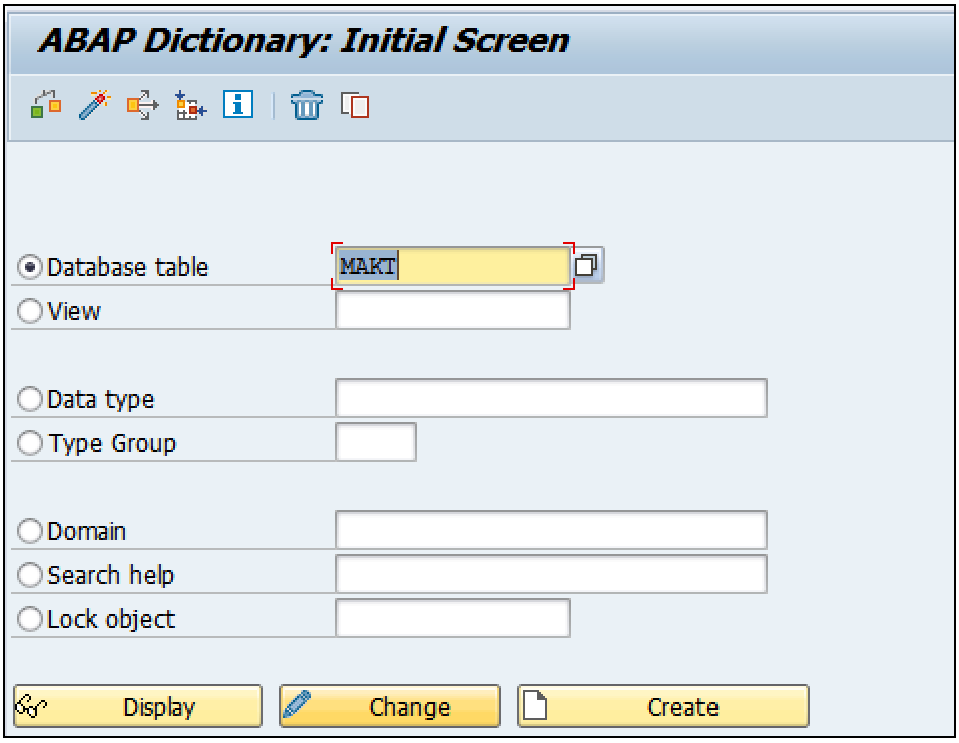
INTERACTIVE_REPORT_COMM

```
RECORDS FOUND IN MAINMAT TBASE      522  MAT NO
         1  CAP_MAT2           1  FERT  PC
         2  CAP_MAT3           1  FERT  PC
         3  CAP_MAT4           1  VOLL  EA
         4  CAP_MAT5           1  VOLL  PC
         5  CAP_MAT6           C  CONT  PC
         6  CAP_MAT7           M  VERP  PC
         7  CAP_MAT8           M  HALB  PC
         8  CAP_MAT9           A  HAWA  PC
         9  CAP_MAT10          A  HAWA  PC
        10  BAPI_MAT_12        1  FERT  KG
```

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.



ABAP Dictionary: Initial Screen

● Database table MAKT
○ View

○ Data type
○ Type Group

○ Domain
○ Search help
○ Lock object

3. Create a copy of the above assignment. Use the GET CURSOR Technique.
4. Create an Interactive report by using the GET CURSOR Technique.

<u>Reference T-Codes and Tables:</u>

T-Codes: SE38 **Tables:** SCARR,SPFLI and SFLIGHT.

Display the basic list report with the following details from the SCARR,SPFLI and SFLIGHT table.
Display only 25 records.
CARRID, CARRNAME, CONNID,COUNTRYFR, FLDATE, PRICE
Use For ALL entries for doing the above task.

If user clicks on CARRID, display the the following details SCARR TABLE.
CARRNAME, CURRCODE, URL

If user clicks on CONNID, display the the following details from SPFLI TABLE.
COUNTRYFR, CITYFROM, AIRPFROM