

Note:

SAP is the name of the company founded in 1972 under German name.

CLASSMATE

Date \_\_\_\_\_  
Page \_\_\_\_\_

## BASIC OF SAP-ABAP

SAP = System Applications and products in data processing.

[ SAP is an ERP software that helps run the day to day operations of an enterprise]

- SAP also name of the ERP (Enterprise Resource Planning) Software as well as the name of the company.

### Advantage of SAP-

- SAP system eliminate duplicate data.
- Business processes are standardize.

ABAP - Advanced Business Application Programming, is a fourth-generation programming language used for development and customization purposes in the SAP software.

ECC - SAP ECC is SAP ERP central component.

It provides different businesses with excellent module to cover the different verticals of industry application related to finance, logistic, sales and marketing etc.

GTS - SAP-GTS (SAP Global Trade Services)

is a software that allows companies to support and define import + export trade processes in SAP-ERP.

ERP - Enterprise Resource Planning refers to a type of software that organizations use to manage day-to-day business activities such as accounting, project management, risk management etc.

Type ERP - SAP, Oracle, Peoplesoft, BAAN etc.

- Types of messages available in SAP?

→ 1. A	= ABEND (abnormal)	(Termination message)
2. I	= Information	
3. E	= Error	
4. S	= Status	
5. W	= Warning	
6. X	= Runtime Error.	

- 3 Layers of the SAP R/3? (SAP 3 tier Architecture)

→ 'R' stands for Real-time data processing.

### 3 Layers -

① Database Layer    ② Application Layer    ③ Presentation Layer

#### ① Database Layer -

SAP R/3 software components that specialized in the Management,

Storage and Retrieval of data from Database Layer.

- Mainly for Performance and security reason.

#### ② Application Layer -

SAP R/3 software components that specialized in processing Business Applications and form Application layer.

#### ③ Presentation Layer -

SAP R/3 software components that specialized in interacting with end-user from the presentation layer.

- Presentation server is any input device that communicates with end user.  
e.g mobile device.

- Why ABAP development needed -

→ We need ABAP development Applications or objects will helpful for end-users for easily access data from database and Reduce risk or effort of end-users.

- client dependent and client-independent.

→ client dependent data-

client-dependent data is defined as data specific to an individual client.

client-Independent Data-

client-Independent data can be defined as data contain across all clients in the system.

- ERP (Enterprise Resource Planning) -

Enterprise Resource planning refers to a type of software that software used by organization to handle their daily activities such as accounting, project management, Risk Management characteristics

① An integrated system    ② operates in real time.

③ common database that supports all the application.

Advantages

① As the integration of business processes saves time & expense.

② management can make decision faster & with fewer errors

SAP: System Applications and products in Data Processing  
ABAP: Advanced Business Application Programming Language.

## Topic 1: DDJC (Data Dictionary elements) [SE11]

- (a) Database Tables
- (b) Structure
- (c) Views
- (d) Data Element
- (e) Domain
- (f) search help
- (g) Lock Objects
- (h) Table Maintenance Generator (TMG)

### DDJC (Data dictionary elements)

ABAP dictionary is a central data management system. Its main function is to support the creation and management of data definitions, such as Data elements, structures, table types etc.

#### Data Dictionary Initial screen.

SAP Easy Access → Tools → ABAP Workbench → Development  
↓  
ABAP Dictionary

Transaction code = SE11

Note: Transaction code in SAP are Execution of program.

Primary key - Uniquely identifies the rows in table by one or more columns.

CLASSMATE  
Date \_\_\_\_\_  
Page \_\_\_\_\_

## ① Database Table-

- Using this field, we can create any custom table or display existing table by entering the name of the table.
- The Name of the custom table must be started with the letter 'Z' or 'Y'.

(i.e- An object which stores data in the form of rows & columns)

### Types of tables -

① Transparent Table

② Pooled Table

③ Cluster Table

Row called Record  
column called Field

## ① Transparent Table -

Transparent tables store the data directly and it exists same structure in database and dictionary with same data and fields. We can read table from database.

### Features

- Contain a single table used to store master data.
- It has a one-to-one relationship with a table in the database.
- There is only a single table.

### Use

They are used to hold master data.

Foreign key - Used to establish the relationship between the different tables present in ABAP dictionary.

### ① Pooled Table -

It is a special table in SAP ABAP dictionary. The data from the tables are stored together in a table pool.

#### Use -

Pooled Tables are used to store the internal controlling information.

#### Features -

- They are used to hold a large number of very small tables.
- It has a many-to-one relationship with a table in the database.
- It is stored with other pooled tables in a single table called table pool in the database.
- The database table has different name, different number of fields and fields have different names.
- Table pools contains than table cluster.

### ② Cluster Table -

Cluster tables are logical tables that are to be assigned to a table cluster they are terminated. Cluster tables helps to control the data, store as temporary data, texts etc.

#### Use -

- The cluster table used when the tables have primary key in common and data in these tables are all access simultaneously.

#### Features -

- They are used to hold data from a few number of large tables.
- It has a many-to-one relationship with table in the database.
- Many cluster tables are stored in a single table in the database called a table cluster.

Note:- Type Group - (Creates group of data types)  
type group allows for defining the non-predefined  
types and grouping them together.

CLASSMATE  
Page

Technical requirement to create the table-

- ① Name of the custom table start with 'Y' or 'Z'. (X reserve for system)
- ② Provide fields, data types and lengths.
- ③ Delivery class.
- ④ Technical settings (Data class + size category)

• Delivery class-

It defines the owner of the tables as well as it controls  
the transport of the data from one table to another table.

• Data class -

Physical area of the database in where table is logically stored

• size category

Probable space requirement of the table in the database.

Size category (0-4)

① APPL0 - (Master Data)

Data frequently accessed but rarely updated.

② APPL1 - (Transaction Data)

Data can access frequently as well as update frequently.

③ APPL2 - (Organization Data)

Data can access frequently and update rarely.

Note: Value Range - Which sets the amount of data values for the fields which refers to domain

classmate

Date

Basic object of the data dictionary -

- ① Tables
- ② Domain
- ③ Data elements
- ④ Structure
- ⑤ Foreign keys.

Payal

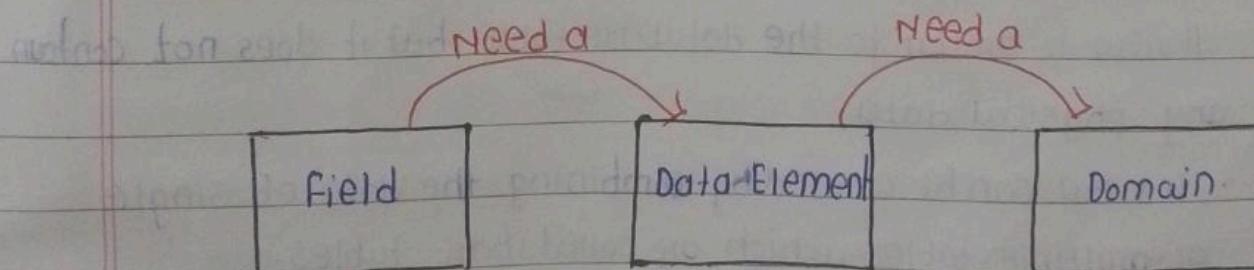
② Domain - (Describe technical characteristics of an attribute)

- Domain is the collection of data types and lengths.
- It is assigned to data element and can be reused in the field.
- It describe the value range of the field the field.

③ Data Element - (semantic definition of table field e.g. description of field)

- Data element is the collection of domain with short description.
- Data element used to provide semantic information of the Database Table.
- Describe how a field can be displayed to end-user.

Relationship bet<sup>n</sup> Field, Data element and Domain.



Field Table in data element.

The field table are used to provide the text information to the data element.

Inner Join - Only Matching key field be selected.

(Unmatch data will not be selected for the view)

CLASSMATE  
Date \_\_\_\_\_  
Page \_\_\_\_\_

### (b) Structure

is the data object that is made up of components of any data type stored one after another in the memory.

Note: A structure may have only a single record at run-time, but a table can have many records.

Steps to create:

SEII → Data type → Name (z....) → Create → Select Shunt  
↓

Provide field + Data ele. ← provide short Descr. ← Press Enter  
(already created)

↓

Save + check + Activate.

### (c) View - (View is like/Nothing but a snapshot)

A view is similar to the database table, but it does not contain any physical data.

- A view can be created by combining the data of single or multiple tables, which are called base tables.

Types of View -

- ① Database View
- ② Projection View
- ③ Help View
- ④ Maintenance View.

outerjoin - All the data of the first table will be displayed from another table only match record will be displayed.

classmate

Date \_\_\_\_\_  
Page \_\_\_\_\_

Note: IF there is no matching record in other tables, then the view will show the record with blank values.

### ① Database view

- If view is created on one or more tables by combining the fields using inner join such a view is called as database view.
- Can not perform any maintenance operation on table data we can just read the data.

~~data~~

### ② Projection view-

- If a view is created on a single table, then is called a projection view.
- This view allow us to read and also maintain the data.
- only required fields filtered out.

### ③ Help view-

- The Help View is created on two or more than two tables specifically for the 'Search Help' in DBFC.
- Combines the data by using an outer join.
- Allows only to read the data and cannot maintain the data.
- used as search help selection method.

~~Page~~

Database View	Maintenance View
① Can be used instead of innerjoin	cannot use ABAP open SQL statement
② Cannot update multiple tables at a time.	② Can be update multiple tables at a time.

### Maintenance View -

- The maintenance view is created on two or more tables, which is used to maintain the data of several tables together.
- It can combine several tables in a single unit, but the tables must have a foreign key relationship.
- It allows us to maintain and read the data of the table.
- By using maintenance view ,we can display, modify and maintain the data.

### joins to create view

- ① Inner Join
- ② Outer Join.

#### ① Inner Join -

- only matching key field selected from both tables.

#### ② Outer Join -

- All data from the first table will be displayed and from other tables, only matching record will be displayed.

## components of tables in ABAP DDIC-

### ① Delivery and Maintenance

② Delivery class -

③ Display / Maintenance

Allows the user to only necessary / required data display .

### ② Fields

④ Field -

Name which provide to each field of table its name as field name .

⑤ key - (Primary key, foreign key) .

Particular field belongs to key field or not .

⑥ Data element and Domain .

⑦

⑧ Technical settings -

⑨ Data class

⑩ size category .

### use of technical setting -

Technical settings of a table to define how the table is handled when it is created in the database . We can specify whether the table is buffered and whether changes to data records of table are logged .

- Optimize storage space requirement .

Primary Index: Primary index is primary fields without primary index we can't create the base table. We can create up to 16 primary index. Use for custom-table only.

CLASSMATE

Date \_\_\_\_\_  
Page \_\_\_\_\_

- Difference between Transparent Table, Pool Table, Cluster Table

Transparent Table

Pool Table

Cluster Table

①	Contain a single table used to store master data	They are used to hold large number of very small table.	Hold data from few numbers of large tables.
②	It has a one-to-one relationship with a table in database	Many-to-one relationship	Many-to-one relationship.
③	Transparent table there is one associate table in the database	Stored with other pool tables in a single table called table pool.	Many cluster tables stored in single table in database called cluster table.
④	Same name and same number of fields	Different	Different.
⑤	Single table	More table than cluster table less tables than pool	
⑥	Secondary index created	Secondary index cannot be created	Secondary index cannot be created

secondary index - secondary index is possible for other than primary fields.  
without secondary index we can't create the base table.  
we can create up to 9 secondary index.  
use for both standard and custom-table.

- Difference bet<sup>n</sup> value table and check table.



value Table

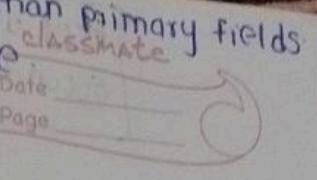
check Table

- |   |   |
|---|---|
| ① value table is validation at domain level.  | ① check table is validation at field level (table level).   |
| ② value table define at domain level and provide help for all fields which refers domain. |   |
| ③ used to extract data in the program.  | ④ can't be used to extract data.  |
| ⑤ search help is displayed from the check table on the foreign key table.                 | ⑥ The search help is displayed on each field which is linked with the same domain of the value table. |
| ⑦ it is not for validation of the data.   | ⑧ This is to validate the data.   |



⑦ it is not for validation of the data.

⑧ This is to validate the data.



Why we are not supposed to create many secondary indexes  $\Rightarrow$  Database Over Head.

classmate \_\_\_\_\_

Date \_\_\_\_\_  
Page \_\_\_\_\_

## Buffering and Buffering types-

### Buffering -

Whenever an open SQL statement is used to read a record from the database table, if the table's attributes indicate that the data should be buffered, the record is saved in RAM on the application server in data buffers. Later, if that record is read again, it is read from the buffer instead of the database.

### Types

#### ① Single Record buffered -

Fetch the one by one record from the table.

#### ② Generic Area buffered -

Depending on the Where condition of record.

#### ③ Fully Buffered -

Fetch the all records from the table.

Note: We are using Buffer for master data tables, because master data tables update rarely and used / accessed regularly

## structure and structure type-

Structure is a combination of different fields, which is having same data type or different type fields. This structure also called as "WorkArea". Structure is skeletal view of a table, it contains the definition of columns and don't have any contents.

### Types

- ① Append structures
- ② Include structures.

#### ① Append structure - (Add New field to a standard SAP Table)

- Append structure are used for enhancements that are not included in the standard.
- An Append structure is a structure that assigned to exactly one table or structure.
- More than one Append structure for a table or structure.
- Within this we implement required field along with its properties
- these are custom defined fields.
- It will be appended only at last.

#### ② Include structure - (Add new fields to custom tables).

- This structure can be include any number of times in any number of tables.
- These are SAP defined fields.
- It can be included anywhere in the table.

## F Search Help and Types

- Allows create help document for the user.

OR

- A search help is a search tool to help you find data in the SAP R/3 system.

### Types

- ① Elementry search Help
- ② Collective search Help.

### ① Elementry search Help (is for F4 Help)

- Helps describes a search path.

- Elementry search help must define where the data of the hit list should be read from, how the exchange of values between the screen template and selection method is implemented and how the online input help should be defined.

### Steps:

SE11 → Search help Name → create → choose Elementry search help

↓  
Activate ← Selection method ← Enter search help parameters

## ⑤ collective search help-

- collective search help combine several elementry search help.
- A collective search help can offer several alternative search paths.

### Steps

SEII → Enter search help name → create → choose collective search

Help



Assign parameters → include search help ← Enter search help  
to each include search help parameter.



click push button → Activate

Note:- Lock object automatically generate two function module

① DEQUEUE\_<Lock object Name>  $\Rightarrow$  To insert the object in queue  
② ENQUEUE\_<Lock object Name>  $\Rightarrow$  To remove object is being queued

### ④ Lock object

- Not allowed the access of multiple users simultaneously.
- Used to synchronize access to the same data by more than one program.
- Lock objects are used in SAP to avoid the inconsistency at the time of data is being insert / change into database.

### Types

① Read lock (shared lock) [S]

② Write lock (Exclusive lock) [E]

③ Enhanced write lock (Exclusive lock without cumulating) [X]

### Steps

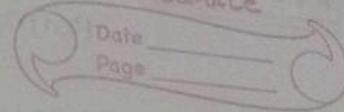
SE11  $\rightarrow$  Lockobject select  $\rightarrow$  Enter name [EZ ... ]

Enter table name in name field  $\leftarrow$  click Table Tab  $\leftarrow$  short description

select lock mode  $\rightarrow$  click lock parameter tab

S+A  
(Save + Activate)

- TMG events
- ① Before saving the data in the database.
  - ② After saving the data in the database.
  - ③ Before deleting the data display.
  - ④ After deleting the data display.
  - ⑤ Creating a new entry.



### (b) Table Maintenance Generator-

- The purpose of the table maintenance generator is to maintain the records or entries from the database table.
- TMG is a interface tool which is used to change the entry of the table or delete an entry from the table or create entry.

### Steps

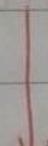
SEII → Enter TableName → click change button → go to Utilities



Provide Authorization Group ← Provide Function Group ← Select TMG



Provide Maintenance screen → Provide maintenance screen no.  
(single step or two step)



create + save.

(TMG check by SM30 or SM31)

#### Single Step Maintenance screen

Table Maintenance program will have only one screen where you can add, delete or edit records.

#### Two Step Maintenance screen

Two screen created the overview screen & single screen are created. The user can see the key fields in the first screen and can further go on to edit further details.

## Topic-2 Report Programming

### Report-

A Report is a presentation of data in an organized structure.

### Types

- ① Classical Reports
- ② Interactive Reports
- ③ ALV Reports

### \* Two types of program -

① Report

② Dynpros.

#### ① Report

Reports are just programs that generates list of data. Basically they supply data to the front end database.

#### ② Dynpros

Dynpros are dynamic programs. They allows the user to interact in the execution of the program.

- AT-selection-Screen output → is like process Before output Event. This would get triggered before screen is displayed.
- AT-selection-Screen - like a process AFTER input This event would get triggered after screen is displayed.

## ① classical Report

A classical report is created by using the output data in the Write statement inside a loop. They do not contain any sub-report.  
- These report consist of only one screen as an output.

### classical Report Event-

#### ① Load of program-

Upload or Load the program into memory for Execution.

#### ② Initialization-

Used to initialize the default values of program variable and selection-screen variable.

#### ③ AT-selection Screen output-

Event is used to modify the selection screen dynamically based on user action on the selection screen.

#### ④ AT-selection-Screen-

Event used to validate multiple fields of selection-screen.

#### ⑤ AT-selection-Screen on field-

Validate single input field on the selection-screen.

#### ⑥ AT-selection on value request-

This event is used to provide a search help for an input field.

### ④ AT-selection screen on help request -

Used to provide help information for an input field.

### ⑤ Start-of-selection

Event is used to write the original business logic statement.

### ⑥ End-of-selection -

Used to indicate the start of selection has been ended.

### ⑦ Top-of-page -

Display constant type page heading for all the pages in the output screen.

### ⑧ End-of-page -

Provide constant footer information across all the pages of the list screen.

## ② Interactive Report -

Interactive Report Programming used to actively control the data retrieval and display of data.

- Interactive report used to create a detailed list and the detailed data is written on a secondary list.

### Events in Interactive Report -

1. AT Line-selection.
2. AT USER command
3. Top-of-Page during Line Selection.
4. AT PFn.
5. SET PF-Status.

#### 1. AT Line-selection -

This event is used to generate an interactive list whenever particular in the out is double clicked.

#### 2. AT USER-command -

This is used to handle user action on the screen whenever standard menu bar and application tool are changed.

#### 3. Top-of-Page during Line selection -

This event is used to write something on top of every page of individual secondary list. Also Top-of-page event is used to write something on only basic list.

#### 4. AT PFn - To create PF Status (customize button on screen).

click F1 to F12 function keys.

⑤ AT PF-STATUS - At the time of attaching our own GUT to the program.

## ⑥ ALV Reports - (ABAP List viewer)

- ALV Report in SAP ABAP are very commonly used in many standard and custom SAP transaction across different SAP modules. ALV allows you to perform various functions dynamically such as sorting, arranging, filtering & retrieving of data.

### Types of ALV Report -

① Simple ALV Report - Normal ALV list.

② Blocked ALV Report -

This is used to display multiple list continuously.

③ ALV Hierarchical sequential Report -

Display in Hierarchical.

Note: ALV functions is used to enhance the readability and functionality any report output.

### Uses of ALV Reports

- Sorting of records.
- Total & sub-totals.
- changing the order of the columns in the report

### Function Modules in ALV Report

display result in a

① Reuse\_ALV\_Grid\_Display

② Reuse\_ALV\_List\_Display.

② Reuse\_ALV\_Fieldcatalog

④ Reuse\_ALV\_Fieldcatalog\_Merge.

⑤ Slist\_layout\_ALV

⑤ PP\_Status=> Slist\_Extab.

↓  
custom button

Prints the  
data. ↗

Populates field  
catalogue ↗

Layout ↗  
purpose ↗

ALV improve the functionality + Readability  
of any report output

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

#### ④ Control Level processing events - OR AT Events ?

Control level processing events are also called as AT Events.  
These events are used within the loop statement of an internal table.

Types -

##### ⓐ AT FIRST

The control level is defined by the first row of the internal table. The control break takes place when this row is read.

##### ⓑ AT NEW

The control level is continuous rows of the internal table that have the same content in an initial part.

##### ⓒ AT LAST -

The control level is defined by the last row of internal table. The control break takes place when this row is read.

##### ⓓ AT END OF -

Used to detect a change in the value of the field between the loop passes.

## Topic-2 Internal Table And Programming

CLASSMATE

Date \_\_\_\_\_  
Page \_\_\_\_\_

### Regular basis used Transaction codes -

SE11	= Data Dictionary
SE30	= Table Maintenance
SE38	= ABAP Editor
SE24	= Class Builder
SE51	= Menu Painter
SE67	= Function module
SEG3	= Custom Transaction code creation
SFP	= Adobe Form   Interface.

### Message type SAP-ABAP-

A	= Termination Message
E	= Error Message
I	= Information Message
S	= Status Message
W	= Warning Message
X	= Exit Message.

- SAP system fields-

- ① SY-Uline = Horizontal line
- SY-Uzeit = Current time
- SY-Ucomm = Function code that triggered PAF
- SY-Uname = Logon Name of User
- SY-tabix = Current Row index
- SY-Tcode = current transaction code.
- SY-Tfill = current number of row
- SY-Title = Text in header line.
- SY-SUBRC = Return value after ABAP statement.
- SY-SCOLS = Number of columns.
- SY-REPID = Current master program.
- SY-index = Current Loop pass.
- SY-DBCNT = Number of processed table rows.
- SY-DATUM = Current Date

Note: \$TMP is the default package which is provided by SAP.

Note: Parameter not accept any float datatype

Date  
Page

### Data Types SAP-ABAP-

Numeric Data type

I → Integer

F → Float

P → Packed Decimal

Character Data Type.

C → char

N → Numeric char

D → Date

T → Time.

Syntax :

DATA: <variable-name> Type <Data-Type>

### Parameter -

Parameter is the keyword which accept the input at runtime.

Syntax:

Parameter <Name> type <Data-type> .

Syntax for select -query

select <Field1> <Field2>

----- from <database-table>

into table <internal Table>

where Field = <Parameter> .

Varints - save your selection parameter  
in the input screen.

It minimize the need to enter selection parameters each time you run a SAP report.

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

- Select-option -

Select-options is the keyword which accepts the single value, multiple single value, single range and multiple range.

### Syntax

select-options <Name of select-options> for <Variable Name>

### Syntax for select-query .

select <Field1> <Field2>

--- from <Database Table>

into table <Internal Table>

where <Field> in <select-options>.

- Difference between parameter and select-options:

#### Parameter

#### Select-options.

- ① Parameter is the keyword which accept the single value at run time.
- ① Select-options is the keyword which accept the single value, multiple single values, single range and multiple ranges.

- ② Without provide input parameter we can't get data
- ② Without providing input select-options we can get the entire data from database.

Date \_\_\_\_\_  
Page \_\_\_\_\_

This is introducing types statement with the addition Begin of started with Types + ending with End of

## • Declaring Structure in ABAP editor

### Syntax

Data : Begin of <structure/table Name>

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

} List of fields

End of <structure/table Name>

Note: IWork area always holds only one record at a time.  
So we go for internal table.

## • Internal Table -

- Internal Table is the collection of records.
- Internal table is the collection of records.
- Internal tables are temporary tables.
- It not save any record in SAP.
- Internal tables are dynamic memory location.
- The scope of the internal table is up to that program.
- Inserting / Accessing record from internal table always record by record.

### Syntax of Internal Table

Data <internal Table> like table of <IWorkArea>

### Syntax of IWork Area

Data <IWorkArea> type <structure Name>

Database table and internal table

## Database Table

## Internal Table.

- |  |   |
|--|---|
| ① Database Tables are permanent storage location         | ① Internal tables are temporary storage location            |
| ② We can access the database table from anywhere in SAP. | ② We can access the internal table within the program only. |
| ③ Must to provide size of the database table             | ④ Dynamically size allocate to internal table.              |

## • Types of internal table.

## ① Standard Table    ② Sorted JT

## ③ Hashed Internal Table

- |   |   |
|---|---|
| - Accept duplicate Record - May or may not accept | - Not accept duplicate                      |
| - Fields are Non-unique                           | - At least one Unique field                 |
| - Insert data through <u>Append</u> keyword       | - Insert data through <u>Insert</u> keyword |
| - Searching record is <u>Linear search</u>        | - <u>Binary search</u>                      |

Syntax :

Data <JT> like standard table  
of <WA>

## Syntax

Data <IT> like sorted  
table of <WA> with  
Unique/Non-unique key  
(Field)

Syntax

Data <IT> like Hashed tab  
of <WA> with unique/  
Non-unique key <PI><P2>

## • Keyword Lines

Lines is the keyword which returns the number of records in internal table.

### Syntax

Describe Table <Internal Table> kind <variable> Lines <variable>

optional                                    optional

## • Declaring internal table by referring database table -

Data <IWA> like <DatabaseTable>

Data <IT> like table of <IWA / DatabaseTable>

## • Internal Table with Header Line

Header line creates one work area with the name of internal table. i.e name of Work Area and Internal Table is same.

### Syntax

Data: Begin of <typeName>

---

---

End of <typeName>

Data: <IT> type table of <typeName> with header line.

- Method use to pass record from Work Area to Internal Table body
- ① Append - Insert a new record at the end of table.

Syntax

Append <WA> TO <IT>

- ② Insert - Insert a new record at the given index.

Syntax

Insert <WA> into <IT> index n

- ③ Collect - This is used for summation purpose.

Syntax

Collect <WA> into <IT>

- Methods to use initialize structure and internal table.

- ① Clear - clears work area content / IT content.

Syntax-

clear <WA|IT>

- ② Refresh - Clears the content of internal Table.

Syntax

Refresh <IT>

- ③ Free - Free clears IT as well as Memory

Syntax-

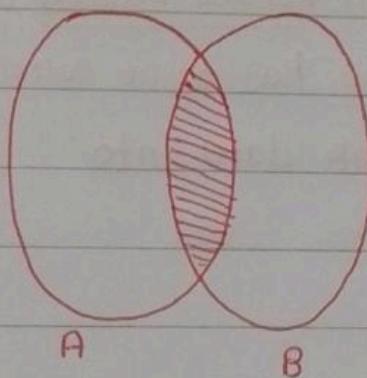
Free <IT>

## Joins in Internal Tables-

- ① Inner Join    ② Left outer join    ③ For All entries.

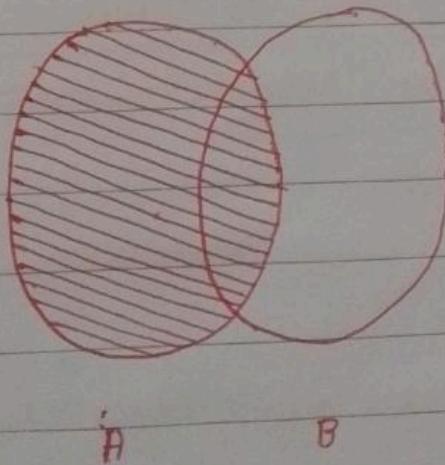
### ① Inner join-

only matching key field be selected. Retrive data based on 'with key' fields also depends on 'Where' condition.  
i.e Filtering data at Database level.



### ② Left outer join

All the data of the first table will be displayed, from another table only match record will be displayed.



③ For All entries

First we have to fetch data from 1<sup>st</sup> database table into 1<sup>st</sup> Internal table for specified condition.

Again we will write one more select query to fetch the values from 2<sup>nd</sup> database table for the corresponding entries in internal table, by internal table 1.

Note: The table is used in For All Entries, that is must having record or not initial and has some common fields with the table you want to fetch data into.

## Operations of Internal Table as well as Database Table -

① Append - To insert IWork area record at the last position of Internal Table.

(a) Single Record  $\Rightarrow$  Append <IWA> To <IT>

(b) Multiple Record  $\Rightarrow$  Append Lines of <IT1> to <IT2>

② Insert - To insert records into internal Table at specific location.

(a) Single Record  $\Rightarrow$  Insert <IWA> into <IT> index <Index\_No>

(b) Multiple Record  $\Rightarrow$  Insert lines of <IT1> From <indexNo1> to <indexNo5> into <IT2> index <Index\_No>

③ Delete

(a) Delete record from specific location -

Delete <IT> from <index-No>

(b) Delete the ranges of record from internal Table -

Delete <IT> from <N1> to <N2>

(c) Delete IT data based on condition -

Delete <IT> where condition.

④

Collect

Collect <IWA> from <IT>

⑤ Modify - Used to overwrite the records of a JT.

ⓐ Modify single Record-

Modify <JT> from <WA> transporting <Field\_Name> where <condition>

ⓑ Modify Multiple Record-

Modify <IT> from <WA> transporting <F1> <F2> where <condition>

⑥ Read

① To process single record-

Read Table <IT> into <WA> index <Index-No>

② Read the data from internal table into WA based on key-

Read Table <IT> into <WA> with key <condition>

⑦ Delete adjacent duplicates-

Delete adjacent duplicates from <JT> comparing <Field>

⑧ Sort = Sort <JT> by <Field> (Ascending / Descending)

⑨ Update - Update <JT> from <WA>

⑩ Describe => Describe Table <AT> Lines, Gr-lines.

- Difference between Select single and upto 1 rows.

Select single

Upto 1 rows.

① key fields pass in where condition.

If key fields are 1 then pass 1 field.

or

4 fields are there pass

4 fields.

① To Avoid passing too many fields in where condition then 'Upto 1 Rows' are used.

Syntax

Select

\_\_\_\_\_

Endselect.

② Only one time triggered

② Two time triggered.

③ This is used to fetch the exact record

③ This is used for validation.

## Topic-3 Modularization Technique.

CLASSMATE

Date \_\_\_\_\_  
Page \_\_\_\_\_

### Modularization -

Modularization means place source of code or statement in a module, instead of placing all statement in main program we just call the module.

### Types

① Use of include files

② Subroutines

③ Function Modules.

#### ① Use of include files -

Include programs are global repository objects used to modularize the source code. They allow us to use the same source code in different programs.

-Also be useful if having lengthy data declaration that you won't to use in different programs.

Rules - ① Include programs cannot call themselves

② Include programs must contain complete statements.

Syntax - **include <include program name>**

write: | \_\_\_\_\_ .

program \_\_\_\_\_

include <include program name>

## ② Subroutines - (subroutines are procedure)

- Procedure can be define in any ABAP program and call from any program.
- Subroutines generally called internally.
- Subroutines contain algorithm or code that used locally.

### Syntax -

Form <subroutine> [<pass>]

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

EndForm.

// where <subroutine> = Name of the subroutine

<pass> : Parameter being passed.

### Types

① Internal Subroutines

② External Subroutines.

## ① internal subroutines-

- subroutines defined in same program being called.
- Can access all the data objects declared in the main ABAP/4 program.

### Syntax

Form <subroutine> <pass>

\_\_\_\_\_

EndForm

### calling internal subroutine

Perform <subroutine> <pass>

[Note: ① Data Declared in main program is automatically available.

② When variable declared inside subroutine is called local variable]

② External subroutine

- External subroutine is perform statement is one program and form routine is in another program.
- Need to use the <pass> option or declare data objects in common part of memory.

Syntax

Perform <subroutine> <program> <pass>

OR

Perform <subroutine> <program> <pass> [if found]

OR

Perform <subroutine> in program <program> <pass> [if found]

OR

Perform <index> of <subroutine1><subroutine2> <pass>

Notes:

① Nested calls are allowed in subroutines.

[i.e Perform within a Form ----- Endform]

② Recursive calls are possible.

③ To define local data, use the data statement after Form

④ To define global data used within a subroutine, use local statement after Form

⑤ Perform and Form must contain same number of parameters.

Note - An subroutine : Using - Act as a importing parameter  
changing - Act as importing and <sup>Exisitance</sup> <sup>Date</sup> Exporting Parameter.

### Subroutine with parameter syntax -

Defination  
syntax

Form <Subroutine Name> [Using ..... value<parameter>  
type <datatype> | like <field>]  
[changing ..... value[]<parameter>  
type <datatype> | like <field>]

EndForm.

Calling  
syntax

Perform <Subroutine Name> Using ..... <parameters>  
changing ..... <parameters>

### Types of parameter -

① Actual Parameters.

Parameters that appear on the  
Perform statement are called  
actual parameters.

② Formal Parameters.

Parameters that appears on the  
Form statement are called  
formal parameters.

• Function Modules are also sub-programs which contains set of reusable statements for better readability & reusability.

## ② Function Module [SE37]

Function modules are ABAP routines that encapsulate program code and provide an interface for data exchange.

- Functional modules are reusable components that are defined in Function Library.

### Function Group-

Function modules are organized into Function Group that are collection of logically related functions.

- It is container of user-defined function modules.

### Syntax

Function <Function\_Module>

Endfunction

### Steps to create Function Group-

SE80 → Select function group from drop down list



Click Yes to create ← Enter ← Give Name start with Z or Y  
object



Short description → Save under package → Function group created.

→ Function group contains two include programs by default one for global declaration another one is for common subroutines

Types of Function module

① Regular Function Module

② Remote Enabled Function module

③ Update Function Module

CLASSMATE

Date \_\_\_\_\_

Page \_\_\_\_\_

Steps to create function module-

Firstly create Function Group (SE80)

SE80 → Name Give → click Create (Provide Function group name + enter + continue)

↓  
Include file 'LZCALU01'

[contain source code of pt FunModule]

↓

Include file 'LZCALTOP'

[contain Global Data]

↓

Main program 'SAPLZCAL'

←↓

Define interface parameters + Exceptions

↓

Write source code

↓

Activate function module.

① Regular Function Module - (Default option)

This kind of function module is executed immediately & synchronously on your current SAP system.

② Remote Enabled Function Module - can be called by other SAP or non-SAP system utilize RPC protocol.

③ Update Function Module - Not executed immediately. They are scheduled for execution in an update work process.

CLASSMATE  
Date \_\_\_\_\_  
Page \_\_\_\_\_

RPC = Special type of Function module which is used to communicate with remote server.  
Remote server can be SAP-SAP server or SAP to Non-SAP server.

### • Remote function call

- RFC stands for Remote function call. which is used to communicate with SAP to SAP system and it does not communicate with SAP to Non-SAP system.
- We can use the RFC interface for communication between SAP systems and between external system.
- SAP offers several interfaces that are based on RFC.

(a) ALE (Application Link Enabling)

(b) BAPI's (Business Application programming Interface)

### Normal Function Module

### RPC function module

- |  |  |
|--|--|
| ① Normal function module only called within the server.          | ① RFC function module called across the server.                  |
| ② We can't access data from Normal function module from non-SAP. | ② We can access data from RPC function module from non-SAP also. |
| ③ We can use pass by value or pass by reference                  | ③ we can use pass by value                                       |
| ④ Fast accessing using normal function module                    | ④ slow accessing using RPC.                                      |

## Types of RFC Function Module

- ① Synchronous RPC
- ② Asynchronous RPC
- ③ Transactional RPC
- ④ Queued RPC

### Transactional RFC-

Data can be transferred between two systems reliably and safely using transactional RFC.

- The called function module is executed exactly once in the called system.

### BAPI - (Business Application Programming Interface)

It is a library of functions that are released to the public as an interface into an existing SAP system from an external system.

- A BAPI are standard SAP function modules provided by SAP for remote access

- Technic

### BDC - (Batch Data Conversion or Communication)

which is used to collect the data from Non-SAP to SAP system.

Types - ① Direct Input method    ② Batch Input method

Technique of transferring data from legacy system into SAP system

### Difference between synchronous and Asynchronous method in BDC-

• Synchronous Method processing is waiting for acknowledgement of record updation.

• Asynchronous method processing does not waiting for acknowledgement of record updation.

### LSM14 - (Legacy system Migration Workbench)

is a R/3 based tool, which is used to transferring data from Non-SAP System to R/3 once

BoR - Business object Repository  
CLASSmate  
It is container of business object.  
(SIW01 T-code used BoR)

Date \_\_\_\_\_  
Page \_\_\_\_\_

User Exit is used for single implementation  
procedural approach.

BADA - they are used for multiple implementation  
& object oriented approaches.

## Enhancement-

Enhancement is ADD-ON new feature to the existing software.

### Types

- ① User Exits
- ② Form Routines
- ③ Customer Exits
- ④ BADA'S

#### ① User Exits

- User Exits is very old concept and first enhancement technique provided by SAP.
- User Exits are nothing but a subroutines, we are going to write the logic in Form-Endform.
- User Exits are not upgraded.
- User Exits are created in SAP Namespace.
- User Exits require access key to make changes.

#### ② Customer Exits-

- Customer Exits is pure enhancement technique provided by SAP
- customer exit having function module interface we have to implement.
- Customer exits are created in customer namespace.
- Customer exits are created in customer does not require access key.

- BADI are the enhancement which can be applied to the standard SAP program as per business requirement.

CLASSMATE

Date \_\_\_\_\_

Page \_\_\_\_\_

④ BADI'S - (Business Add-In) [SE18-BADI DEF^n, SE19-BADI Impl^n]

- BADI is an interface. It is newer version of user-exits.
- BADI's are available in customer level, company code level and partner level.
- BADI's support multiple implementation.
- BADI's are object oriented, implemented multiple times.

#### • Difference between RPC and BAPI

BAPI and RPC Remote enabled function modules. The difference between RPC and BAP are business object.

When we create business object and those are registered in our (Business object repository) which can be accessed outside the system system by using some other application

#### • Types of BADI

- ① Single Implementation BADI
- ② Multiple Implementation BADI.
- ③ Filter BADI
- ④ Custom BADI.

The term BADI stands for Business Add-in. It is a new SAP object oriented enhancement technique. BADI is used to implement our business functionality to the existing SAP standard functionality.

## Topic-4 Module Pool Programming / Screen Programming / Dialog Programming

SE51 - Screen Painter

SE41 - Menu Painter

SE38 - ABAP Editor

SEG3 - Transaction Code Creation.

### Events in MPP-

#### ① PAF (Process After Input)

e.g. checking values where there is no automatic check, checking/processing cursor position, processing connect entries, triggering an error dialog.

#### ② PBO (Process Before Output)

Processing before the screen is displayed (PBO)

e.g. initializing screen fields, inserting default values, positioning cursor, showing and hiding fields and changing field attributes dynamically.

#### ③ Process on Help Request (POH)

#### ④ Process on Value Request (POV).

### Syntax

#### ① PBO (Process Before Output)

Module <Module-Name>

### Implementation -

Module <module-name> output.

Endmodule.

Note: Every button has unique function code.

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

## ② PAI (PROCESS AFTER INPUT)

Module <Module-Name>

Implementation -

Module <Module-Name> Input

\_\_\_\_\_

Endmodule.

## ③ Process on value request (POV) -

Field <Field-Name> Module <Module-Name>

## ④ Process on Help Request (POH)

Field <Field-Name> Module <Module-Name>

### • Purpose of chain-Endchain -

If you want to validate a group of fields on a screen at that time we use chain-Endchain statement.

### • OK-Code

OK-Code is a variable which is used to collect the screen attributes related function codes.

OK-Code is type of system variable systcomm

- Purpose of Module AT Exit-command -  
Automatic field check can be avoided by AT Exit-command which works exactly the same way as cancel works on application tool bar.

- options to move out of screen -

① Call screen - Calling screen a special case of embedding a screen sequence. [Alive All screen call new screen]

② SET screen -

Statement processes the remaining statements of the current screen and then goes to the called screen.

③ Leave screen -

Statement ends the current screen and call the subsequent screen.

Normal screen

sub-screen.

- |  |   |
|--|---|
| ① Normal screen having ok-code                       | ① sub-screen does not have ok-code.                     |
| ② Normal screen having GUI status                    | ② sub-screen does not have GUI status.                  |
| ③ Normal screen calls another screen.                | ③ Subscreen does not call another screen or sub-screen. |
| ④ Normal screens having <u>Module Exit-command</u> . | ④ Subscreen does not have <u>Module Exit-command</u> -  |

- Strip of tabs
- Each tab must contain at least one sub-screen area.
- Each sub screen area can call only one sub screen at time.
- By default tabstrip contains 2 tabs. Only one tab is always activated.

Syntax for calling SUB-SCREEN in MPP -

Call subscreen <subscreen Area Name> including SY-REPTD 'subscreenId'

Controls available in MPP -

- ① Table Control
- ② Tab Strip Control
- ③ Custom Control.

### ① Table control -

- Table control is an area on the screen in which you can display data in tabular form.
- Table controls allow you to enter, display and modify tabular data easily on the screen.

### ② TAB STRIP control -

- A tabstrip control is a screen object consisting of two or more pages.
- Each tab page consists of a tab title and page area.
- Each tab of the tab strip control is attached with a subscreen.
- When user clicked on particular tab then appropriate subscreen is called and the screen attach to that subscreen is displayed.

### ③ Custom control -

A custom control is an area on the screen, created using the Screen painter. custom control are used to embed controls.

Note

Formal Parameters :- Parameter define during definition CLASSMATE of subroutine or subroutine with FORM statement.

Date \_\_\_\_\_  
Page \_\_\_\_\_

## Topic - 5 Adobe Forms

### Adobe Forms -

It is a universal file format developed by Adobe that preserves all the fonts, formatting, graphics and colors of any source document regardless of the application and platform used to create it.

### Interactive Adobe form -

as a form which needs user interaction to get data from it.

components involved in creation of SAP Adobe Forms -

- ① Layout
- ② Context
- ③ Interface.

### Advantages of Adobe forms -

- ① Flexible tool for designing the form.
- ② Graphics can be included in the form directly.

### Types of form layout -

- ① static Layout. (Form have fixed layout)
- ② Dynamic Layout (Designed to expand)

Actual Parameter - Parameter define during calling  
the subroutine or subroutine  
with perform statement.

CLASSMATE

Date \_\_\_\_\_  
Page \_\_\_\_\_

SubForm - subForm is nothing but group of elements

- subForm is used to organize the different elements like input fields, tables, text presents body pages.

### Form Interface -

In the form interface, you specify the data that is exchanged with the application program.

### MVC Architecture -

Web Dynpro ABAP follows MVC (Model View Controller) Architecture.

M - Model - Actual business logic

V - view - view is nothing but a screen with UI element that hold data.

C - Controller - This is responsible for communication between model and view.

### How to call Adobe Frame in program -

Using 3 functions

① FP\_JOB\_OPEN

② FP\_FUNCTION\_MODULE\_NAME.

③ FP\_JOB\_CLOSE.

### Component of layout sets -

Header Data, Page, Page window, widows, paragraph format, character format.

- Transfer logo to SAP content  $\Rightarrow$  SE78.  
(Logo must be in .BMP format)
  - After uploading the logo-
    - SAP  $\rightarrow$  Create interface  $\rightarrow$  Declare global variable  
(VI type XSTRING)
- ↓
- Then call method       $\leftarrow$  In code initialization  
`CL-SSF-&F-Utilities->`      (write select query to get  
`get_bds_graphic_as_bmp`      logo from table)
- ↓
- `PASS P-Object = 'Graphics'`
- `P-Name = 'Name_of_image'`  $\rightarrow$  This return XSTRING type  
`P-ID = 'BMAP'`      variable  
`P-type = 'BCOL'`
- ↓
- In the right side pane under       $\leftarrow$  Create form and assign  
 context tab, create graphic node      interface with form
- `select Graphic node in properties`       $\rightarrow$  Drag & Drop image field  
 section make the Graphic type  
 a graphic content      to layout
- ↓
- `Properties, continue with STA`      Check graphic binding

### Types of Form Layout -

① Static Layout - These forms have fixed layouts. When presented to the end user, the form retains its original layout, regardless of the amount of data available to fill the form.

② Dynamic Layout - A form with a dynamic layout is designed to expand or shrink according to the amount of data available to fill it.