**1.What is the difference between Centralized and Decentralised system?**

Ans: In a **centralized system**, all data and processing are managed in one central location, whereas in a **decentralized system**, data and processing are distributed across multiple independent systems.

**2.What is SAP?**

Ans: SAP is an integrated ERP (Enterprise Resource Planning) software that helps businesses manage their processes efficiently across various departments like SD, MM, FICO & HR.

**3. What is the different product of SAP?﻿**

Ans: The main SAP products include **SAP S/4HANA**, **SAP ECC**, **SAP BW/4HANA**, **SAP CRM**, **SAP SRM**, **SAP SCM**, and **SAP SuccessFactors**, each designed for specific business functions.

﻿

**4. Different types of SAP projects?﻿**

Ans: The different types of SAP projects are **Implementation**, **Support**, **Rollout**, **Upgrade**, **Migration, Green filed implementation and Brown Filed implementation** projects.

**Greenfield implementation** is setting up a completely new SAP system from scratch.

while **Brownfield implementation** is upgrading or transforming an existing SAP system to a newer version like S/4HANA.

**5. What is SAP project landscape?**

Ans: SAP project landscape refers to the **tiered environment structure** consisting of **Development (DEV)**, **Quality Assurance (QA)**, and **Production (PRD)** systems used to manage, test, and deploy SAP applications.

**6. What is SAP project approach?**

Ans: SAP project approach refers to **Accelerated SAP (ASAP) Methodology** used to implement SAP or **Agile** methodologies to ensure structured and successful project execution.

**ASAP Methodology** is a structured to **speed up SAP project delivery** through predefined phases: **Project Preparation, Business Blueprint, Realization, Final Preparation, and Go-Live & Support**.

**7. What is the ABAP?**

Ans: **ABAP (Advanced Business Application Programming)** is programming language used to develop applications, reports, interfaces, and enhancements within the SAP environment.

Its is 4th Generation language and not case sensitive

**8. Explain about types of applications in ABAP?**

Ans: The main types of applications in ABAP are **Reports (Classical and Interactive)**, **Module Pool (Dialog Programming)**, **Forms (Smart Forms, SAP script, Adobe Forms)**, **Interfaces (RFC, BAPI, IDoc, OData)**, and **Enhancements (Exits, BADI)**.

**RICEFW** components are:

* **Reports** – it is used to display the Business info of one/more dept in list/grid format on single/multi screens,
* **Interfaces** – Is used to transfer the data from one server to another i.e. SAP to SAP or SAP to Non-SAP, Flat file to SAP **(RFC, BAPI, IDoc, OData)**,
* **Conversions** – It is used to migrate data from system to flat file or flat file to SAP system **(BDC or LSMV),**
* **Enhancements** – Custom code added to standard SAP without modifying it,
* **Forms** – Output documents like invoices or POs using Smart Forms/SAP script/Adobe Forms,
* **Workflows** – Automated process flows for tasks like approvals and notifications.

**9. Tell me about SD flow, T-codes and DB tables?**

Ans: The **SD (Sales and Distribution) flow** includes:

**Company →Customer (Central – XD01, Sales – VD01, finance – FD01) →Inquiry (VA11) → Quotation (VA21) → Sales Order (VA01) → Delivery (VL01N) → Billing (VF01) → Payment (F-28).**  
**T-codes**: VA01 (Sales Order), VL01N (Delivery), VF01 (Billing), VA05 (Order List),  
**DB tables**: VBAK (Sales Document Header), VBAP (Item), LIKP (Delivery Header), LIPS (Delivery Item), VBRK (Billing Header), VBRP (Billing Item).

**10. Tell me about MM flow, T-codes and DB tables?**

Ans: **MM (Material Management)**

**Flow**: **PR → PO → GR → IR → Payment**

* **PR (Purchase Requisition)** – ME51N
* **PO (Purchase Order)** – ME21N
* **GR (Goods Receipt)** – MIGO
* **IR (Invoice Receipt)** – MIRO
* **Payment** – F-53

**Key Tables**:

* **EKKO** – PO Header
* **EKPO** – PO Item
* **MKPF** – GR Header
* **MSEG** – GR Line
* **RBKP** – Invoice Header
* **RSEG** – Invoice Item
* **EBAN** – Purchase Requisition.

**11. What is the difference between Data and Types keywords?**

Ans: **TYPES** is used to **define a data type**, while **DATA** is used to **declare a variable** of that data type in ABAP.

**12. What is the difference between Type and Like keywords?**

Ans: **TYPE** refers to a **data type definition** (e.g., TYPE I), while **LIKE** copies the **data type of an existing variable or database field** (e.g., LIKE mara-matnr).

**13. What is Constant?**

Ans: A **constant** in ABAP is a **fixed value** declared using the CONSTANTS keyword that **cannot be changed** during program execution.

**14. What is the difference between Parameters and Select-options?**

|  |  |
| --- | --- |
| **Parameters** | **Select-options** |
| 1. It is a single input field  2. It is a single step process to declare  3. It will not accept only float data type  4. We can work with Checkbox, Radio button and As List box  5. No intervals and no extension are not possible here  6. It cannot be converted as Select-options  7. Without value no results  8. It does not have any components like select-options | 1. It is a multiple input field  2. It is a 2-step process to declare, first we need to declare variable or DB table then refer it to select-options  3. It will not accept float and string data types  4. We cannot work with Checkbox, Radio button and As List Box  5. No intervals and no-extension are possible  6. It can be converted as parameters  7. Without value it fetches entire data  8. It has 4 components like SIGN, OPTION, LOW AND HIGH |

**15. What are operations and different types of operations?**

Ans: **Operations:** These are used to perform certain arithmetic calculations, compare variables and to compare certain conditions.

Different types of operations are: Arithmetic operations, Relational operations and Logical operations.

Arithmetic operations: It is used to perform mathematical calculations like Addition (+), Subtraction (-), Multiplication (\*), Division (/), Modulus (MOD) and Exponentials (\*\*)

Relational operations: It is used to compare 2 variables or fields like GT (>)/GE (>=)/LT (<)/LE (<=)/EQ (=)/NE (<>/><)

Logical operations: It is used to compare 2 statements or conditions like AND/OR.

**16. What are string operations and different types of them?**

Ans: **String operations:** These are used to perform certain operations on string or character data types variables and fields.

Different types of string operations are: CONCATENATE, SPLIT, REPLACE, TRANSLATE, CONDENSE, SHIFT, STRLEN, REVERSE etc...

**17. What are Program controls?**

Ans: Program Controls: It is used to control the flow of execution of statements in an applications

Different techniques are:

1. Branching Technique and

2. Looping Technique

**18. Difference between Branching and looping techniques?**

Ans: **1. Branching Technique:** It is used to process or execute only 1 satisfied condition at one time

Different types of branching techniques are: IF conditions and Case condition.

**2.Looping Technique:** It is used to process or execute 1/more/all the satisfied condition at 1 time

Different types of looping conditions are:

1. Unconditional Loop (DO....ENDDO)

2. Conditional Loop (While...End while)

3. Simple Loop (LOOP...ENDLOOP)

4. Query (Select queries).

**19. What is use of Exit, Check and Continue?**

Ans: **EXIT:** It is used within the looping condition to come output of the loop, if the condition is satisfied.

**CHECK:** It is used within the looping condition to process the satisfied condition.

**CONTINUE**: It is used within the looping condition to skip the satisfied condition.

**20. What is an internal table and different types of internal tables?**

Ans: It is a temporary table.

Scope of an internal table is up to that application level itself.

Data can be filled or displayed through work area.

Memory will be allocated or de-allocated dynamically during the runtime of an application.

|  |  |  |
| --- | --- | --- |
| **Standard ITAB** | **Sorted ITAB** | **Hashed ITAB** |
| 1. Default type of an itab is standard.  2. Kind of itab is 'T'.  3. We can hold duplicate records in standard itab.  4. Data can be transferred using Append from work area to itab.  5. It fetch data using linear search from DB table.  6. It is very slow in process.  Syntax:  Data: <IT> LIKE TABLE OF <WA>.  DATA <IT> LIKE STANDARD TABLE OF <WA>. | 1. We need to specify the type of itab.  2. Kind of itab is 'S'.  3. We may or may not hold duplicate records in sorted itab.  4. Data can be transferred using Insert from work area to itab.  5. It fetch data using Binary search from DB table.  6. It is moderate in process.  Syntax:  Data: <IT> LIKE SORTED TABLE OF <WA> WITH KEY <UNI/NON-UNI>. | 1. We need to specify the type of itab.  2. Kind of itab is 'H'.  3. It never holds any duplicate records.  4. Data can be transferred using Collect from work area to itab.  5. It fetch data using Hashed algorithm from DB table.  6. It is fast in process.  Syntax:  Data: <IT> LIKE HASHED TABLE OF <WA> WITH KEY <UNIQUE>. |

Different Types of internal table operations:

1. Data Transfer Technique

2. Data Initialization Technique

3. Data Change Technique

4. Data Deletion Technique

5. Data Access Technique and

6. Attributes of an internal table.

**21. What is field-symbols?**

Ans: **Field-symbols** in ABAP are like **pointers** that refer to memory locations of variables or internal table rows at runtime, declared using the FIELD-SYMBOLS keyword.

**22. Difference between select single and select up to 1 row?**

|  |  |
| --- | --- |
| **SELECT SINGLE** | **SELECT UP TO 1 ROWS** |
| 1. It is used to fetch single record from DB  2. It is used when the table has more than 1 key fields  3. We need to pass all the key field values in where condition  4. It is used to fetch exact record from DB | 1. It is also used to fetch single record from DB  2. It fetch first record among multiple records  3. We need to pass only partial key fields in where condition  4. It ends with ENDSELECT |

**23. Difference between Joins and for all entries?**

|  |  |
| --- | --- |
| **JOINS** | **FOR ALL ENTRIES** |
| **1.** It fetch the data from 2 or more DB tables.  2. Through one select query we can fetch multiple table data into 1 itab.  3. It will not specify under which table data does not exist.  4. Suggested to go for 3-4 transactional DB tables. | 1. It is used to fetch data from each DB table separately.  2. Through multiple select query, we will fetch data in different itab.  3. It specify which table does not hold any data.  4. Suggested to go for any no. of DB tables  5. Sy-subrc check is mandatory here after each select query, otherwise it fetches entire data from current select query if data does not exist. |

**24. What are the different internal table operations?**

1. **APPEND** – Add a new row at the end
2. **INSERT** – Add a row at a specific position
3. **READ TABLE** – Fetch a specific row based on a condition
4. **MODIFY** – Update an existing row
5. **DELETE** – Remove a row
6. **LOOP AT** – Iterate over the table
7. **SORT** – Arrange data in ascending/descending order
8. **CLEAR** – Reset the table or a row
9. **REFRESH** – Clear all rows of the table
10. **COLLECT** – Add values based on key fields (for summation).

**25. How to transfer data to final internal table, when we fetched the data in separate internal tables using FOR ALL ENTRIES?**

* First, we need to loop the table which has more no. of key fields.
* If there are multiple tables with same no. of key fields, then we need to check the item level table.
* If there are multiple item level tables then we need to check the least level in business.
* That table we need to loop and all other tables, we need to perform READ table operation.

**26. What are the modularisation techniques?**

* IT IS USED TO DIVIDE THE BUSINESS LOGIC INTO REUSEABLE BLOCK OF STATEMENTS.
* IT IS USED TO AVOID REDUNDANCY OR DUPLICATE LINES OF CODE.
* IT CONSIST OF 2 PARTS. DEFINE THE REUSEABLE BLOCK AND CALL THE REUSEABLE BLOCK.
* IT IMPROVES THE READABILITY OF AN APPLICATION.
* IT IS EASY TO HANDLE AN ERROR.

**27.** **Difference between Subroutines and Function Modules?**

Ans:

|  |  |
| --- | --- |
| **SUBROUTINES** | **FUNCTION MODULE** |
| Subroutines are Internal and External.  It has 3 parameters: USING, CHANGING and TABLES.  It cannot be tested independently without calling.  External Subroutines are defined in SE38 with Type of application as 'S'. | Function module is also global approach  It has 5 parameters: IMPORT, EXPORT, CHANGING, TABLES & EXCEPTION  Function modules are defined in Function Builder SE37  It can be tested independently. |

**28. Different types of Function Modules?**

Ans: Different types of **Function Modules** in ABAP are:

1. **Normal Function Module** – Standard reusable module for specific tasks.
2. **Remote-Enabled Function Module (RFC)** – Used for communication between different SAP or external systems.
3. **Update Function Module** – Executes in a separate LUW (Logical Unit of Work) for database updates (IN UPDATE TASK).

**29. Tell me some of the standard function modules?**

1. REUSE\_ALV\_GRID\_DISPLAY – Display ALV grid reports
2. REUSE\_ALV\_LIST\_DISPLAY – Display ALV list output
3. REUSE\_ALV\_BLOCK\_LIST\_INIT – Initialize block list for multiple ALVs
4. REUSE\_ALV\_BLOCK\_LIST\_APPEND – Add multiple ALVs in one output
5. REUSE\_ALV\_BLOCK\_LIST\_DISPLAY – Display the complete block ALV
6. REUSE\_ALV\_HIERSEQ\_LIST\_DISPLAY – Display hierarchical-sequential ALV (like header-item format)
7. CONVERSION\_EXIT\_ALPHA\_INPUT – Converts external to internal (adds leading zeros)
8. CONVERSION\_EXIT\_ALPHA\_OUTPUT – Converts internal to external (removes leading zeros)
9. BDC\_OPEN\_GROUP – Open a BDC session
10. BDC\_INSERT – Insert a transaction into the session
11. BDC\_CLOSE\_GROUP – Close the BDC session
12. CALL\_TRANSACTION – Execute BDC in foreground/background
13. GUI\_UPLOAD – Upload flat file for BDC input
14. F4\_FILENAME – File path selection popup
15. OPEN\_FORM, WRITE\_FORM, CLOSE\_FORM – For SAP Scripts
16. ALSM\_EXCEL\_TO\_INTERNAL\_TABLE – Upload Excel (basic, cell-wise)
17. OPEN DATASET – Open a file (for read/write)
18. READ DATASET – Read data line by line
19. TRANSFER – Write data to a file
20. CLOSE DATASET – Close file
21. DELETE DATASET – Delete file from app server

**30. What is the parallel cursor technique?**

Ans: **Parallel Cursor Technique** is an ABAP optimization method used to **improve performance during nested loops** by reducing the number of loop iterations.

In traditional nested loops, looping over large internal tables causes **high CPU usage and slow performance**.

Instead of looping through the inner table from the beginning each time, **you use a pointer/index to continue from where the last match was found**, improving efficiency.

**31. Difference between ABAP memory and SAP memory?**

**ABAP Memory:**

1. Used to share data within the same internal session (between programs).
2. Uses keywords: EXPORT TO MEMORY and IMPORT FROM MEMORY.
3. Requires a Memory ID to store and retrieve values.
4. Private to the user session — not accessible across transactions.
5. Data is cleared when the internal session ends.
6. Good for passing data between SUBMIT, CALL TRANSACTION, or CALL FUNCTION.

**SAP Memory:**

1. Used to share data across multiple sessions or transactions.
2. Uses keywords: SET PARAMETER ID and GET PARAMETER ID.
3. Requires a Parameter ID (from data element definition).
4. Global memory – shared across all SAP GUI sessions for the user.
5. Data stays until the user logs off or it’s overwritten.
6. Often used for auto-filling screen fields (e.g., material number, vendor).

**32.** **What is ABAP dictionary?**

Ans: IT IS A CENTRAL COMPONENT OF SAP SYSTEM.

IT SPECIFIES THE GLOBAL DEFINITION OF DB TABLES AND FIELDS.

COMPONENTS DEFINED IN ABAP DDIC CAN BE ACCESS FROM ANY WHERE IN SAP SYSTEM.

TCODE OF ABAP DIC IS SE11.

**33. Components of ABAP dictionary?**

Ans: DIFFERENT COMPONENTS OF DDIC ARE:

* DB TABLES
* VIEWS
* DATA TYPE
* TYPE GROUP
* DOMAIN
* SEARCH HELP
* LOCK OBJECTS

**34. What are DB tables?**

Ans: IT IS A 2-DIMENSIONAL MATRIX THAT CONSIST OF ROWS AND COLUMNS.

ROWS ARE RECORDS AND COLUMNS ARE FIELDS.

**35. Different types of DB tables?**

**TRANSPARENT TABLES**: IT HAS 1 - 1 RELATIONSHIP.

WHEN WE CREATE 1 TABLE IN ABAP DIC, IT WILL CREATE A COPY OF THAT TABLE IN DB LAYER.

TABLE NAME, FIELD NAME AND DE NAME WILL BE SAME IN DDIC AND DB LAYER.

WE CAN WORK WITH JOINS FOR THESE TYPES OF TABLES.

WE CAN MAINTAIN BUFFERING FOR THESE TABLES.

**POOLED TABLES**: IT HAS MANY - 1 RELATIONSHIP.

WHEN WE CREATE MULTIPLE SMALL TABLES IN ABAP DIC, IT WILL CREATE 1 SINGLE LARGE TABLE IN DB LAYER.

TABLE NAME, FIELD NAME AND DE NAME WILL BE DIFFERENT IN DDIC AND DB LAYER.

WE CANNOT WORK WITH JOINS FOR THESE TYPES OF TABLES.

WE CANNOT MAINTAIN BUFFERING FOR THESE TABLES.

**CLUSTER TABLES:** IT HAS MANY - 1 RELATIONSHIP.

WHEN WE CREATE MULTIPLE LARGE TABLES IN ABAP DIC, IT WILL CREATE 1 SINGLE VERY LARGE TABLE IN DB LAYER.

TABLE NAME, FIELD NAME AND DE NAME WILL BE DIFFERENT IN DDIC AND DB LAYER.

WE CANNOT WORK WITH JOINS FOR THESE TYPES OF TABLES.

WE CANNOT MAINTAIN BUFFERING FOR THESE TABLES.

**36. Difference between Append and. include structure?**

|  |  |
| --- | --- |
| **APPEND STRUCTURE** | **. INCLUDE STRUCTURE** |
| 1. IT CAN BE CREATED IN STD AND CUSTOM TABLE.  2. IT WILL BE ADDED IN THE BOTTOM OF TABLE.  3. APPEND STRUCTURE NAME CANNOT BE SAME IN MULTIPLE TABLES.  4. WITHIN THIS STRUCTURE WE CANNOT ADD ANOTHER APPEND STRUCTURE. | 1. IT CAN BE CREATED IN CUSTOM TABLE ONLY.  2. IT CAN BE ADDED ANY WHERE IN DB TABLE.  3. IT CAN BE SAME NAME IN MULTIPLE TABLES.  4. WITHIN THIS WE CAN ADD ANOTHER APPEND STRUCTURE. |

**37. What are Indexes and different types of them?**

Ans: INDEXES: IT IS USED TO IMPROVE THE PERFORMANCE OF SELECT QUERY WHILE FETCHING THE DATA FROM DB TABLES.

DIFFERENT TYPES OF INDEXES ARE: PRIMARY INDEX AND SECONDARY INDEX

DIFFERENCES BETQWEEN PRIMARY AND SECONDARY INDEXES:

|  |  |
| --- | --- |
| **PRIMARY INDEXES** | **SECONDARY INDEXES** |
| 1. IT IS A PRIMARY KEY FIELD.  2. IT IS MANDATORY TO CREATE IN DB TABLE.  3. IT SHOULD BE AT THE TOP OF THE TABLE.  4. PRIMARY INDEX CANNOT BE SECONDARY INDEX.  5. THERE CAN BE MAXIMUM 16 PRIMARY INDEX.  6. IT CAN BE PASSED REFERING TO DB | 1. IT IS A SECONDARY KEY FIELD.  2. IT IS OPTIONAL.  3. IT CAN BE ANY WHERE IN DB TABLE.  4. SECONDARY INDEX CANNOT BE PRIMARY INDEX.  5. THERE CAN BE MAXIMUM 9 SECONDARY INDEX.  6. IT CAN BE RESTRICTED TO ANY DB |

**38. What is Delivery Class?**

Ans: DELIVERY CLASS SPECIFIES THE ACCESS BEHAVIOR OF TABLE WITH TYPE OF TABLE AS APPLICATION TABLE, SYSTEM TABLE, CONTROL TABLE, CUSTOMIZING TABLE, ETC.

**39. What are technical settings?**

Ans: **TECHNICAL SETTINGS OF DB TABLE:**

IT SPECIFIES **THE LOGICAL STORAGE PARAMETER, BUFFERING AND LOGGING OF DB TABLES.**

**LOGICAL STORAGE PARAMETER:** IT SPECIFES DATA CLASS AND SIZE CATEGORY.

**40. What is Data class and different types of Data we can maintain?**

**DATA CLASS:** IT SPECIFIES THE TYPE OF DATA THAT WE CAN MAINTAIN IN DB TABLE.

**DIFFERENT TYPES OF DATA ARE:**

* MASTER DATA
* TRANSACTIONAL DATA
* ORGANIZATIONAL DATA
* SYSTEM DATA
* USER DATA.

**MASTER DATA(APPL0):** DATA THAT CAN BE ACCESSED FREQUENTLY AND MODFIED RARELY IS KNOWN AS MASTER DATA.

EXAMPLE: CUSTOMER DATA, VENDOR DATA, EMPLOYEE DATA, MATERIAL DATA ETC...

**TRANSACTIONAL DATA(APPL1**): DATA THAT CAN BE ACCESSED FREQUENTLY AND MODIFIED FREQUENTLY IS KNOWN AS TRANSACTIONAL DATA.

EXAMPLE: SALES, DELIVERY, INVOICE, PO. PR, ETC...

**SIZE CATEGORY:** IT SPECIFIES THE NO. OF RECORDS WE CAN MAINTAIN IN DB TABLE.

THIS CAN BE CHANGED IF REQUIRED EVEN FOR STANDARD/CUSTOM DB TABLES.

**BUFFERING:** IT IS USED TO MAINTAIN AND ACCESS THE DATA FROM SHARED BUFFERED MEMORY. IT IS TEMPORARY STORAGE LOCATION TO ACCESS THE DATA.

BY DEFAULT, DB TABLES ARE MAINTAINED AS BUFFERING NOT ALLOWED.

**DIFFERENT TYPES OF BUFFERING ARE:** SINGLE RECORD, GENERIC AND FULLY BUFFERED.

**SINGLE RCORD:** IT HOLDS ONLY SINGLE RECORD IN SHARE BUFFERED AREA THAT CAN BE ACCESSED FROM ANY APPLICATION.

**GENERIC BUFFERRED**: IT HOLDS ONLY KEY FIELD INFORMATION IN SHARED AREA

**FULLY BUFFERED:** IT HOLDS ENTIRE DATA IN SHARED AREA.

**LOGGING:** IT IS USED TO ACCESS FROM WINDOW, JAVA, .NET ETC SYSTEMS....

**41. Difference between Check table and foreign key table?**

* **CHECK TABLE:** IT IS MAIN TABLE TO VALIDATE DATA IN THE SUB LEVEL TABLES
* **FOREIGN KEY TABLE:** IT IS A FOREIGN KEY FIELD TABLE, THAT VALIDATE THE ENTRIES WITH CHECK TABLE.

**42. What is Value Table?**

Ans: A **Value Table** is the **reference table** assigned to a **Domain**, used to suggest **possible allowed values** for fields using that domain.

**43. What is a Table Maintenance Generator?**

Ans: **TABLE MAINETNANCE GENERATOR (TMG):**

* WHEN WE CREATE TMG, SAP PROVIDES FUNCTION GROUP WITH FM'S AND LOGIC RELATED TO CRUD OPERATION. SYSTEM GENERATE CODE REALTED TO CREATE, CHANGE, DELETE THE RECORDS IN DB TABLE.
* HERE WE CAN PASS WITH OR WITHOUT AUTHORIZATIONS.
* 1 STEP PROCESS IS USED TO DISPLAY DB TABLE WITH OUTPUT SCREEN IN TABLE MAINTENANCE TOOL (SM30).
* 2 STEP PROCESS IS USED TO DISPLAY TABLE AND THEN OUTPUT SCREEN IN SM30
* SYSTEM WILL PROPOSE DEFAULT SCREEN NO. 0001.

**44. What are the events of TMG?**

Ans: TMG ALSO HAS THEIR EVENT WHILE CREATE, CHANGE, BEFORE SAVE, AFTER SAVE, BEFORE DELETE, AFTER DELETE ETC... IT STARTS FROM 01, 02, 03, 04, 05, ETC.

**45. What is the purpose DB object utility?**

Ans: **SE14 – Database Object Utility** is used to perform **administrative and repair tasks** on **database tables and views**, especially when changes are made in the ABAP Dictionary.

**46.** **What are Views and different types of Views?**

VIEWS: VIEWS ARE VIRTUAL/ IMAGINARY TABLES.

VIEWS CONSIST THE FIELDS OF 1 OR MORE DB TABLES.

DIFFERENT TYPES OF VIEWS ARE:

* PROJECTION VIEW
* DATABASE VIEW
* MAINTENANCE VIEW
* HELP VIEW

**PROJECTION VIEW:** IT CONSIST THE REQUIRED FIELDS OF 1 TABLE.

HERE WE CAN MAKE THE VIEW AS READ OR READ AND CHANGE.

**DATABASE VIEW:** IT CONSIST THE REQUIRED FIELDS OF 1 OR MORE DB TABLES.

FOR 1 TABLE WE CAN MAKE THE VIEW AS READ OR READ AND CHANGE.

WHEREAS FOR MORE THAN 1 DB TABLE WE CANNOT CHANGE THE VIEW.

**47. Difference between Projection and DB view?**

**Projection View:**

1. Based on a single table only.
2. Used to restrict access to certain fields of a table.
3. No join is possible.
4. Mainly used for security and simplification.
5. Cannot be used in table maintenance (SM30).
6. Faster since no join is involved.
7. Defined in SE11.

**Database View:**

1. Based on multiple tables using INNER JOIN.
2. Used to combine related data from different tables.
3. Requires foreign key relationships.
4. Can be used in read-only or updatable mode (if only one table is updatable).
5. Can be used in table maintenance (SM30) if based on one table.
6. Slightly slower due to join operations.
7. Also defined in SE11.

**48. Difference between Data Element and Domain?**

**Data Element:**

* IT SPECIFIES THE SEMANTIC ATTRIBUTES OF A FIELD.
* SEMANTIC ATTRIBTE SPECIFIES THE BUSINESS MEANING OF A FIELD IN BUSINESS CONTEXT.
* IT SPECIFIES DOMAIN AND FIELD LABELS.

**Domain:**

* IT SPECIFIES THE TECHNICAL ATTRIBUTES OF A FIELD.
* TECHNICAL ATTRIBUTES SPECIFIES THE DATA TYPE, SIZE CATEGORY AND OUTPUT CHARACTERISTICS OF A FIELD.

**49. What is field label?**

Ans: **FIELDS LABELS** SPECIFIES THE DESCRIPTION OF A FIELD.

**50. Difference between Structure and Table Type?**

**STRUCTURE**:

* IT IS A TABLE WITHOUT VALUES IN IT.
* IT HOLDS THE VALUE DURING RUNTIME IN WORK AREA.
* WE CAN USE THESE STRUCTURE IN ANY DB TABLE OR IN APPLICATION TO DECLARE WORK AREAS.

**TABLE TYPE:**

* IT IS ALSO A TABLE WITHOUT DATA IN IT.
* IT IS ALSO USED TO REFER IN APPLICATION TO DECLARE ITABS.

**51. What are Type Groups? Tell me some standard Type-Groups?**

* **TYPE GROUP**: TYPE-POOLS IS A COLLECTION OF TYPE-GROUP.
* **TYPE-GROUP** IS A COLLECTION OF TYPES VARIABLES, TYPES STRUCTURE AND CONSTANTS**.**

1. **SLIS -** Used for ALV reports (contains structures like SLIS\_LAYOUT\_ALV, SLIS\_T\_FIELDCAT\_ALV)
2. **ICON -** Contains constants for using standard SAP icons.
3. **VRM** - Used for dropdown list boxes in selection screens.
4. **SYST -** Contains system field names and types.
5. **ABAP -** Contains basic ABAP constants and types.

**52. What are Search Help and different types of search help?**

**SEARCH HELP:** IT IS USED TO PROVIDE F4 HELP FOR A TABLE FIELDS**.**

**DIFFERENT TYPES OF SEARCH HELP ARE:**

* ELEMENTARY SEARCH
* COLLECTIVE SEARCH HELP

**ELEMENTARY SEARCH**: IT IS USED TO PROVIDE F4 HELP WITH THE FIELD OF 1 DB TABLE.

IT PROVIDES SINGLE TAB IN F4 HELP.

**LPOS AND SPOS:** LPOS SPECICIES THE LIST POSITION AND SPOS SPECIFIES THE SCREEN POSITION OF A FIELDS IN F4 HELP.

**COLLECTIVE SEARCH HELP**: IT IS A COLLECTION OF ELEMENTARY AND COLLECTIVE SEARCH HELP.

IT PROVIDE MULTIPLE TABS IN F4 HELP.

**53. Difference between LPOS and SPOS?**

Ans: **LPOS AND SPOS:** LPOS SPECICIES THE LIST POSITION AND SPOS SPECIFIES THE SCREEN POSITION OF A FIELDS IN F4 HELP.

**54. What is the Purpose of Hotkey?**

Ans: **HOT KEY**: IT IS USED TO TRIGGER SPECIFIC F4 HELP IN OUR APPLICATION.

**55. What is lock object?**

**LOCK OBEJCT:**

* IT IS USED TO RESTRICT THE MULTIPLE USER ACCESSING OF SAME TABLE AT 1 TIME.
* IT DOES NOT ALLOW MULTIPLE USER TO ACCESS THE SAME DB TABLE AT 1 TIME.
* WE CAN LOCK RELEVANT MULTIPLE DB TABLES ALSO AT 1 TIME.
* CUSTOM LOCK OBJECT STARTS WITH EY/EZ.

**56.** **What function module generated by lock object?**

Ans: LOCK OBJECT GENERATES 2 FUNCTION MODULES. THEY ARE:

* ENQUEUE FM AND
* DEQUEUE FM.

**57. Different types of lock modes?**

Ans: DIFFERENT TYPES OF **LOCK MODES** ARE WRITE LOCK(DEFAULT), READ LOCK, CUMULATIVE LOCK, OPTIMISTIC LOCK, COLLISION LOCK ETC.

**58. Can we lock multiple tables in 1 lock object?**

Ans: You can lock multiple tables in one lock object.

**59. Few Examples of pool and cluster table?**

**Pool Tables – Examples**

**Table Name Purpose**

* AINDX Stores indexes for archived documents.
* EDIDC Contains IDoc control records.
* TADIR Stores metadata of repository objects (programs, function groups, etc.).
* TRDIR Stores program directory (ABAP reports, includes, etc.).

**Cluster Tables – Examples**

**Table Name** **Purpose**

* BSEG Stores accounting document line items (part of FI documents).
* KOCLU Stores internal orders (CO).
* PCL1 HR -related payroll results (cluster data).
* PCL2 HR time management data clusters.

**60. What is SD flow?**

**SD PROCESS:**

* SD PROCESS HAPPENS BETWEEN COMPANY AND CUSTOMER.
* FIRST CUSTOMER WILL GO FOR ENQUIRY OF REQUIRED PRODUCTS/MATERIALS.
* BASED ON ENQUIRY, COMPANY WILL SEND A QUOTATION TO CUSTOMER.
* ONCE CUSTOMER IS SATISFIED WITH QUOTATION, CUSTOMER WILL RAISE AN ORDER.
* THIS ORDER WILL BE PURCHASE ORDER FOR CUSTOMER AND SALES ORDER FOR COMPANY.
* BASED ON ORDER, COMPANY WILL GO FOR PICKING AND PACKING OF MATERIALS.
* THIS IS KNOWN AS DELIVERY PROCESS.
* IF COMPANY TRANSPORTS THE MATERIALS, THEN THE SHIPMENT WILL BE DONE.
* COMPANY WILL GENERATE AN INVOICE.
* CUSTOMER WILL MAKE A PAYMENT.
* ONCE THE PAYMENT IS DONE, COMPANY WILL CONFIRM ACKNOWLEDGMENT AS POD.

**61.** **What is the T-codes of SD dept?**

**T-CODES RELATED TO SD DEPARTMENT ARE:**

1. XD01 - CREATE CENTRAL CUSTOMER
2. VD01 - CREATE SALES CUSTOMER
3. FD01 - CREATE FINANCE CUSTOMER
4. VA11 - CREATE ENQUIRY
5. VA21 - CREATE QUOTATION
6. VA01 - CREATE SALES ORDER
7. VA31 - CREATE SCHEUDLE LINE
8. VA41 - CREATE CONTRACT
9. VL36 - PICKING
10. VL01/VL01N - CREATE DELIVERY
11. VF01 - CREATE INVOICE
12. VT01 - CREATE SHIPMENT
13. MM01 - CREATE MATERIAL

**62. What are the tables of SD depts?**

**SD DEPARTMENT DB TABLES:**

* KNA1, KNB1, KNB4, KNBK, KNB5, KNVV, KNVH, KNVI, KNVS, KNVK, KNVP, KNKA,
* VBFA, VBUK, VBUP, VBAK, VBAP, VBEP, VEDA, VBKD,
* LIKP, LIPS, VTTK, VTTP, VBRK, VBRP,
* MARA, MAKT, MARC, MARD, MVER, MBEW, T000, T001, T001W, CDHDR, CDPOS etc.

**63. What is MM and Fi Flow?**

**MM PROCESS:**

* MM PROCESS HAPPENS BETWEEN COMPANY AND VENDOR.
* FIRST COMAPNY WILL PREPARE THE LIST OF PRODUCTS/MATERIALS THAT IS KNOWN AS PURCAHSE REQUISITION(PR).
* BASED ON PR, COMPANY WILL GO FOR REQUEST FOR QUOTATION(RFQ) WITH VENDOR.
* ONCE VENDOR SENDS THE QUOTATION, COMPANY WILL RAISE AN PURCAHSE ORDER(PO).
* VENDOR DELIVERY THE PRODUCTS TO THE COMPANY, THIS IS KNOWN AS GOOD RECEIPT.
* VENDOR GENERATES AN INVOICE AND COMPANY WILL MAKE A PAYMENT

**64. Tables of MM and Fi depts?**

**MM DEPARTMENT DB TABLES:**

* LFA1, LFB1, LFBK, LFB4,
* EBAN, EBKN, EKKO, EKPO, EKET, EINA, EKKN, EKBE,
* MARA, MAKT, MARC, MARD, MVER, MBEW, T000, T001, T001W, CDHDR, CDPOS etc

**65. T-codes of MM depts?**

**T-CODES RELATED TO MM DEPARTMENT ARE**:

1. XK01 - CREATE CENTRAL VENDOR
2. MK01 - CREATE MATERIAL VENDOR
3. FK01 - CREATE FINANCE VENDOR
4. ME51/ME51N - CREATE PR
5. ME41 - CREATE RFQ
6. ME31 - CREATE SCHEDULE LINE
7. ME21/ME21N - CREATE PO
8. MIGO - CREATE GOOD RECEIPT
9. MIRO - CREATE INCOMING INVOICE
10. MM01 - CREATE MATERIAL

**66. What are Classical report and their events**?

**Classical Report**: It is used to display the business information of one or more depts. on a single screen in list i.e. sequential lines.

**Events of Classical reports are:**

1.Load program/Program Constructors - Declarations

2. INITIALIZATION

3.AT SELECTION-SCREEN

4. AT SELECTION-SCREEN ON FIELD

5. AT SELECTION-SCREEN ON VALUE-REQUEST FOR

6. AT SELECTION-SCREEN ON HELP-REQUEST FOR

7. AT SELECTION-SCREEN OUTPUT

8.START-OF-SELECTION

9. TOP-OF-PAGE

10. END-OF-PAGE

11. END-OF-SELECTION

12. SET PF-STATUS

13. AT USER-COMMAND.

**67. Exclusive events of Interactive reports?**

**INTERACTIVE Report**: It is used to display the business information of two or more depts. on a multiple screen in list format i.e. sequential lines.

* AT LINE-SELECTION
* TOP-OF-PAGE DURING LINE SELECTION.

**68. Different techniques in Interactive reports?**

Ans: In **interactive report** we can navigate based on 2 techniques. They are:

* Hide technique
* Get Cursor techniques

**Hide Technique:** It is used to navigate to the secondary list based on record selection. It holds the content of selected record and that is used fetch the data of next level based on certain common field

**Get cursor technique:** It is used to navigate to the secondary list based on field selection. It holds the selected field name and field value.

**69. What is the purpose system field sy-lsind?**

**SY-LSIND:** It is system field that helps to capture the secondary list index no.

**70. What are ALV reports?**

**ALV REPORTS:**

IT IS USED TO DISPLAY BUSINESS INFORMATION OF 1 OR MORE DEPARTMENTS ON A SINGLE OR MULTIPLE SCREENS EITHER IN LIST OR GRID **FORMAT WITH CERTAIN PREDEFINE FUNCTIONALITIES AS BELOW:**

* SORT IN ASCENDING/DESCENDING ORDER
* FILTER
* SELECT ALL
* DESELECT ALL
* SUM
* DOWNLOAD DATA IN TEXT/XLS/DOC FILE IN PC
* SEND AS AN EMAIL ATTACHMENT and CHANGE LAYOUT ETC.

**TYPICAL STRUCTURE TO DEVELOP AN ALV REPORT:**

* DECLARATION PART
* SELECTION SCREEN DESIGN PART
* FILL/FETCH DATA IN DATA ITAB/FCAT ITAB/EVENT ITAB/LAYOUT WA/SORT ITAB/KEYINFO WA/LISTHEADER ITAB ETC
* DISPLAY OUTPUT
* SPECIAL OPERATIONS.

**71. What the different SLIS structure that you used in ALV reports?**

**SOME OF THE SLIS STRUCTURES THAT ARE FREQUENTLY USED WHILE WORKING WITH ALV REPORTS ARE:**

1. **SLIS\_FIELDCAT\_ALV/SLIS\_T\_FIELDCAT\_ALV** - TO FILL FCAT ITAB FOR TABLE FORMAT
2. **SLIS\_T\_EVENT/SLIS\_ALV\_EVENT** - TO FILL EVENT ITAB TO WORK WITH 6 EVENTS
3. **SLIS\_LAYOUT\_ALV** - TO COMPRESS OR TO PROVIDE ZEBRA STRIPS
4. **SLIS\_LISTHEADER/SLIS\_T\_LISTHEADER** - TO PROVIDE HEADER/FOOTER IN GRID ALV
5. **SLIS\_SELFIELD** - TO NAVIGATE BASED ON FIELD/RECORD SELECTION
6. **SLIS\_T\_EXTAB** - IT HOLDS GUI BUTTONS FUNCTION CODE
7. **SLIS\_SORTINFO\_ALV** - TO SORT IN ASC/DESC ORDER AND TO SUB TOTAL
8. **SLIS\_KEYINFO** - TO PROVIDE HEADER AND ITEM TABLE COMMON FIELDS FOR HIERARCHIAL ALV.

**72. Events in ALV reports?**

**EVENTS OF ALV REPORTS ARE:**

1. Load program/Program Constructors - Declarations.
2. INITIALIZATION
3. AT SELECTION-SCREEN
4. AT SELECTION-SCREEN ON FIELD
5. AT SELECTION-SCREEN ON VALUE-REQUEST FOR
6. AT SELECTION-SCREEN ON HELP-REQUEST FOR
7. AT SELECTION-SCREEN OUTPUT
8. START-OF-SELECTION
9. END-OF-SELECTION

**OTHER 6 EVENTS ARE HANDLED USING SLIS STRUCTURE EVENT INTERNAL TABLE:**

1. TOP\_OF\_PAGE
2. END\_OF\_PAGE
3. TOP\_OF\_LIST
4. END\_OF\_LIST
5. PF\_STATUS\_SET
6. USER\_COMMAND

**73. Different function modules of ALV reports?**

**SOME OF FUNCTION MODULES THAT ARE FREQUENTLY USED WHILE WORKING ALV REPORTS ARE:**

1. REUSE\_ALV\_LIST\_DISPLAY - TO DISPLAY IN LIST FORMAT
2. REUSE\_ALV\_GRID\_DISPLAY - TO DISPLAY IN GRID FORMAT
3. REUSE\_ALV\_FIELDCATALOG\_MERGE - TO BUILD FCAT ITAB WHILE WORKING WITH FEW FIELDS
4. REUSE\_ALV\_COMMENTARY\_WRITE - TO PROVIDE HEADER/FOOTER IN TOP/EOL EVENTS
5. REUSE\_ALV\_POPUP\_TO\_SELECT - TO DISPLAY SECONDARY LIST DATA IN POPUP
6. REUSE\_ALV\_BLOCK\_LIST\_INIT - TO INITIATE BLOCK LIST PROCESS
7. REUSE\_ALV\_BLOCK\_LIST\_APPEND - TO TRANSFER DATA TO REQUIRED BLOCKS
8. REUSE\_ALV\_BLOCK\_LIST\_DISPLAY - TO DISPLAY BLOCK ALV REPORTS
9. REUSE\_ALV\_HIERSEQ\_LIST\_DISPLAY - TO WORK WITH HIERARCHIAL ALV
10. RS\_TREE\_CONSTRUCT - TO BUILD TREE ALV DATA
11. RS\_TREE\_LIST\_DISPLAY - TO DISPLAY TREE ALV

**74. What is Tree ALV, structure and FM used in Tree ALV?**

Ans. Tree ALV is a special form of ALV (ABAP List Viewer) used to display **hierarchical (parent-child)** data in a **tree structure**, where nodes can be **expanded/collapsed**.

**Main Function Modules Used in Tree ALV:**

**FM** **Purpose**

**RS\_TREE\_LIST\_DISPLAY** Display simple tree (basic version).

**RS\_TREE\_CONSTRUCT** To construct a tree structure manually.

**75. How to pass header and footer in list report?**

**Use the event:**

* END-OF-PAGE.
* END-OF-PAGE.

76. How to pass header and footer in grid report?

Ans: **Use the SLIS structure event Internal table:**

* **SLIS\_LISTHEADER** as Work Area.
* **SLIS\_T\_LISTHEADER** as Internal Table.

**77. How to provide logo or images in ALV reports?**

Ans**: OAER** - TCODE IS BUSINESS DOCUMENT TOOL USED TO UPLOAD LOGOS OR IMAGES IN ORDER TO DISPLAY IN ALV HEADER OR FOOTER.

**78. How to perform subtotal?**

Ans:

* SLIS\_SORTINFO\_ALV - TO SORT IN ASC/DESC ORDER AND TO SUB TOTAL

**Step-by-Step:**

1. Define your work area and internal table.

* it\_sort TYPE slis\_t\_sortinfo\_alv,
* wa\_sort TYPE slis\_sortinfo\_alv.

1. Prepare field catalog.
2. Specify SUBTOTAL field(s) in the catalog with DO\_SUM = 'X'.
3. Use the SORT table with SUBTOT = 'X'.

**79. How to provide row colour?**

Ans: Using fields of **SLIS\_FIELDCAT\_ALV** (EMPHASIZE).

**80. Without events, how to provide header info, GUI status and logic related to guide buttons?**

Ans: You can define **Header and Footer** using **REUSE\_ALV\_COMMENTARY\_WRITE**. Also, by passing Header info in **i\_grid\_title** of **REUSE\_ALV\_GRID\_DISPLAY** Functional Module.

**81. How to make ALV column as an editable?**

Ans: To make an ALV column editable in ABAP, set the `**EDIT = 'X'**` flag in the field catalog and enable `EDIT = 'X'` in the layout structure during ALV grid display.

Set **ls\_layout-edit = 'X'** to allow overall editing.

**82. What is the performance tuning done while developing applications? (Explain about Extended program checks, ATC, Runtime analysis, SQL trace, etc)**

Ans**: PERFORMANCE ANALYSIS OF ABAP APPLICATIONS:**

* EXTENDED PROGRAM CHECKS - T-CODE: SLIN - TO CHECK UNWANTED ERRORS, WARNINGS AND MESSAGES
* WE CAN AVOID IT BY CHANGING THE CODE OR BY-PASSING PRAGMA'S
* ATC (ABAP TEST COCKPIT) - TO CHECK UNWANTED ERRORS AS PRIORITY 1,2 & 3
* WE CAN AVOID IT BY CHANGING THE CODE OR BY-PASSING PRAGMA'S.
* PERFORMANCE ANALYSIS: T-CODE: SE30 - TO CHECK ABAP CODE TIME, DB TIME AND SYSTEM TIME.
* IT ALSO PROVIDES CERTAIN TIPS & TRICKS to check better performance of certain code like:
* IF STMT & CASE STMT, SELECT WITH JOIN AND FOR ALL ENTRIES, SELECT SINGLE & SELECT UP TO 1 ROWS etc.
* T-CODE: SAT- TO CHECK DB TABLE TIME, LOOPING TIME, ABAP CODE TIME, ETC... IT DISPLAYS DETAILED TIME TAKEN BY ENTIRE CODE
* PERMONACE TRACE (SQL TRACE) - T-CODE: ST05 - TO CHECK SELECT QUERY PERFORMANCE
* Developed the application using TYPES structures and referred this structure to declare work area and internal tables
* Developed the application using Modularization technique concept i.e. include, subroutines and FM
* Used joins for 2-3 tables and more than 3 tables, I prefer to use for all entries
* While working with for all entries, always passed sy-subrc check before fetching another table data
* Always preferred to pass Binary search with Read table operations
* Developed all the reports using appropriate/required Events.

**83. What is Batch job process and how can we schedule batch job?**

Ans: A **Batch Job** is a **background processing job** in SAP used to execute ABAP programs or reports **without user interaction**, usually for **large volumes of data**, **scheduled execution**, or **periodic tasks** (e.g., nightly data loads, updates, etc.).

**Types of Jobs in SAP**

1. **Immediate Jobs** – Start right away.
2. **Scheduled Jobs** – Run at a defined time/date.
3. **Periodic Jobs** – Run repeatedly (daily, weekly, etc.).
4. **Event-Triggered Jobs** – Triggered by SAP events.

* You can schedule a batch job using **transaction code SM36**.
* Use **T-Code SM37** to monitor, check status, logs, etc.

**Job Statuses in SM37:**

**Status** **Meaning**

S Scheduled

R Released

F Finished Successfully

A Active (Running)

C Cancelled/Error.

**84. How to debug Batch job?**

Ans: Debugging a batch job is slightly different from debugging a foreground program. SAP provides several methods to debug a job running in background mode.

**Using JDBG (Most Common & Easy):**

**Steps:**

* Go to transaction SM37.
* Find the job you want to debug (set job status = "Active" or "Finished").
* Select the job and click on "Job -> Details" to get the program name and variant.
* Go back, select the job again, and click "Job -> Debugging" or just enter transaction JDBG.
* The job starts in debug mode, allowing you to step through the code as if it's running in background.
* This works only if the job has already started or completed (status = Finished or Active).

**85. What are SAP scripts?**

Ans: IT IS USED TO DESIGN THE BUSINESS DOCUMENTS TO PROVIDE THE HARD COPIES

**BUSINESS DOCUMENT LIKE:** SALES ORDER COPY, DELIVERY ORDER COPY, INVOICE ORDER, COPY, PURCHASE ORDER COPY, PURCHASE REQUISATION COPY.

IT IS ALSO USED TO DESIGN THE BUSINESS LABELS LIKE: PRODUCT INFORMATION, PRICE TAGS, WARNINGS, CAUTIONS, BARCODES, RECEIPTS ETC.

**86. What are the components of SAP scripts?**

**COMPONENTS OF SAP SCRIPT FORMS (LAYOUT)**

* HEADER
* PAGES
* WINDOWS
* PAGE WINDOWS
* PARAGRAPH FORMAT
* CHARACTER FORMATS
* DOCUMENTATION.

**87. What are the different types of windows?**

**DIFFERENT TYPES OF WINDOWS ARE:**

* MAIN WINDOW
* VARIABLE WINDOW
* CONSTANT WINDOW
* GRAPHICAL WINDOW.

**88. Exclusive FMs of SAP scripts?**

**SOME OF THE EXECLUSIVE FUNCTION MODULES OF SAP SCRIPTS LAYOUT**

* **OPEN\_FORM** -> TO INITIATE THE DATA TRANSFER PROCESS FROM DRIVER PROG TO LAYOUT
* **START\_FORM** -> TO INITIATE THE DATA TRANSFER PROCESS TO A SPECIFIC FORM
  + - IT IS MANDATORY WHEN WE ARE WORKING WITH MULTIPLE FORM AND
    - OPTIONAL WHEN WE ARE WORKING WITH SINGLE FORM
* **WRITE\_FORM** -> IT IS USED TO TRANSFER THE DATA TO A SPECIFIC WINDOW
  + - IT IS REPEATED MULTIPLE TIMES FOR MULTIPLE WINDOW DATA TRANSFER
    - PROCESS
* **END\_FORM** -> IT IS USED TO COMPLETE THE DATA TRANSFER PROCESS OF THE SEPCIFIC FORM
* **CLOSE\_FORM** -> IT IS USED TO COMPLETE THE ENTIRE TRANSFER PROCESS
* **READ\_TEXT** -> IT IS USED TO READ THE CONTENT OF STANDARD TEXT
* **CONTROL\_FORM** -> IT IS USED TO CONTROL THE “TDFORMAT” FROM THE DRIVER PROGRAM
  + - INSTEAD OF SCRIPT LAYOUT.

**89. What are Symbols and different types of them?**

**SYMBOLS:**

* DATA CANNOT BE PRINTED DIRECTLY ON A FORM WINDOWS THAT WAS PASSED FROM A DRIVER PROGRAM
* IT CAN BE PRINTED THROUGH SYMBOLS ONLY
* IT STARTS AND ENDS WITH (&)

**DIFFERENT TYPES OF SYMBOLS ARE:** PROGRAM SYMBOLS, STANDERD SYMBOLS, SYSTEM SYMBOLS AND TEXT SYMBOLS

**PROGRAM SYMBOLS:**

THEASE ARE USED TO DEFINED OUR OWN VARIABLE FROM THE SCREEN LAYOUT AND PRINT THE DATA OF IT (EX: “DEFINE SC\_VAR = ‘1000’)

IT IS ALSO USED TO PRINT THE VALUES OF A VARIABLE, WORK AREA FIELDS

(EX: &WA\_FIELD1&, &WA\_FIELD2&, &WA\_FIELD3&, & LV\_VAR&

**STANDARD SYMBOLS:**

* IT IS USED TO PRINT THE FIELD VALUES OF A STANDERD DB STRCUTURE “SAPSCRIPT”

(EX: FORM PAGES -&SAPSCRIPT – FORMPAGES&

TELEPHONE NO. -&SAPSCRIPT - TELENUM& ETC.)

**SYSTEM SYMBOL:**

* IT IS USED TO PRINT THE VALUES OF A SYSTEM DEFINE VARIABLES DIRECTLY LIKE DATE, TIME, DAY, MONTH, YEAR, USER ETC.

(Ex: &DATE&, &TIME&, &DAY& ETC.)

**TEXT SYMBOL:**

* IT IS USED TO PRINT THE SYMBOLS OF (TIDIG STRUCTURE) (ex: &MR&, & MRS&, &DEAR&, &DR & ETC.)

**90. What is the purpose Text elements?**

* WITH OUT TEXT ELENENTS FIRST RECORD IN THE MAIN WINDOW WILL BE PRINTED TWICE.
* IT IS USED TO PRINT THE CONTINEOUS TEXT ON WINDOW.

**91. What are control commands and name few of them?**

* These are used to control the flow of statements in a page window
* It is also used to specify the format of a window.

**Some of the control commands are:**

1. Include
2. Define
3. Address…. Endaddress
4. Protect……Endprotect
5. Top…………EndTop
6. Bottom……Endbottom
7. If……Endif
8. Case…EndCase
9. Set Date Mask
10. Set Time Mask
11. BOX
12. PERFORM…. Endperform.

**92. What are format options?**

Ans: It is used to change the format of an output in the layout.

**Some of the format options are:**

1. Offset

2. Output length

3. Omit leading Sign(s)

4. Omit leading zero(z)

5. Leading sign at left (<)

6. Leading sign at right (>)

7. Remove separators from Thousands (T)

8. Compress the text (c)

9. No. of Decimals (N) etc.,

**93. How to work with multiple forms?**

Ans: In SAP script, it’s possible to **call and use multiple forms** (layout sets) in a single print program using **control commands** or **function modules**.

**Important Function Modules**

| **Function Module** | **Purpose** |
| --- | --- |
| **OPEN\_FORM** | Starts printing with a form |
| **WRITE\_FORM** | Writes data to a window/element |
| **START\_FORM** | Switch to another form |
| **END\_FORM** | Ends current form |
| **CLOSE\_FORM** | Ends the print output |

**94. How can we design table in scripts?**

**Using Box:**

It is a command that is used to provide horizontal line, vertical line or a box to the window.

For this we need to provide the Coordinates as

**XYWHIF:**

* X - X-axis
* Y - Y-axis
* W - Width
* H - Height
* I - Intensity
* H – Frame

**95. How can we convert script to smart forms?**

Ans: **Steps to Convert Script to Smart form**

1. Go to transaction: SMARTFORMS
2. In the menu, click:
3. Utilities → Migration → From SAP script Form
4. Enter:

* SAP script form name
* New Smart Form name

1. Click Execute (F8).

**96. How to change package of scripts?**

* CHANGE THE PACKAGE OF SAP SCRIPT LAYOUT/FORM:
* RUN SM30 - PROVIDE TADIR AND MAINTAIN
* RSTRFORM->Name of the form-> Execute -> Double click change.

**97. How to debug scripts?**

* RUN SE71, ACTIVATE DEBUGGER OR
* RUN SE38 -> EXECUTE RSTXDBUG.

**98. How to download or upload scripts?**

Ans: RSTXSCRP -> RUN -> Object name -> Form Name -> Execute -> Define path location -> Export.

**99. How to maintain same text in multiple forms?**

Ans: To use the same reusable text in multiple forms, create a Standard Text using transaction **SO10**, and include it wherever needed.

**Steps:**

* Go to T-code: SO10
* Create Standard Text (e.g., ZCOMMON\_TEXT)
* Save it.

**100. How to upload images or logos in scripts?**

To upload and use a logo/image in SAP script, you store the image as a graphics object in SAP SE78 and include it in your SAP script form.

**Step-by-Step Process:**

**Upload the Image**

Use transaction SE78 (SAP Graphic Management):

* Go to T-Code: SE78

**Choose:**

* Graphic → BMAP → Import Graphic

**Upload:**

* Select image from your PC (must be in .bmp format for SAP script)
* Set Graphic Name and Description

**Save:**

* The image will be stored under GRAPHICS → BMAP → SAPSCRIPT.

**101. How to create TR for standard text?**

Ans: Standard texts created via SO10 are **not automatically transportable**, but you can transport them using **Report RSTXTRAN**.

**Steps to Create Transport Request for SO10 Text**

1. **Go to T-code:** SE38
2. Run the report: **RSTXTRAN**
3. In the selection screen:

**Text Name**: e.g., ZMYTEXT

**Text ID**: usually ST

**Text Object**: TEXT

**Language**: e.g., EN

1. Click **Execute (F8).**

**102.** **How to copy scripts from 1 client to another client?**

You can copy an SAP script form (layout set) from one client to another using **program RSTXSCRP**.

**Steps to Copy Script Using RSTXSCRP**

1. **Go to T-code:** SE38
2. Run the program: **RSTXSCRP**
3. Choose **Export** in the selection screen
4. Enter:
   * **Form name** (e.g., ZFORM\_NAME)
   * **Object type**: FORM
   * **Language**: e.g., EN
   * **File name**: Provide a name (this will be stored in SAP's application server)
5. Click **Execute (F8)** → These exports the script to a file

**Then in Target Client:**

1. Log in to the **target client**
2. Go to T-code: SE38
3. Run **RSTXSCRP** again
4. Choose **Import**
5. Enter:
   * Same **file name** used during export
   * Object type = FORM
6. Execute → The script will be imported into the new client.

**103. How to design labels in scripts?**

Ans: To design labels in SAP script, create a compact form in SE71 with small-sized windows matching the label dimensions, and print it using a label-compatible device type like ZEBRA configured via SPAD.

**104. What are the steps to work with standard forms?**

Ans: To work with standard forms in SAP, you identify the standard form and its print program.

We need to copy Standard from Golden client only i.e. 000.

**Step:**

Utilities -> Copy from client -> Form Name-> Golden client 000 -> Target From.

Note: Original Language only.

**105. Explain about any standard form on which you worked on?**

Ans: **MODIFY STANDARD FORMS:**1. IDENTIFY THE STANDARD FORMS IN **'NACE'** TCODE  
RUN NACE TCODE -> SELECT THE REQUIRED APPLICATION I.E. EF FOR PURCHASE ORDER   
-> SELECT THE BUTTON OUTPUT TYPES FROM APPLICATION TOOL BAR -> SELECT THE OUTPUT TYPES I.E. NEU AND PERFORM PROCESSING ROUTINES -> IDENTIFY THE STANDARD DRIVER PROGRAM AND FORM NAME I.E **MEDRUCK** HERE FOR PURCHASE ORDER

**Standard SAP Forms and their Driver Programs:**

|  |  |  |
| --- | --- | --- |
| **DEPARTMENT** | **FORM** | **DRIVER PROGRAM** |
| SALES  DELIVERY  INVOICE  PURCHASE ORDER | RVORDER01  RVDELNOTE  RVINVOICE01  MEDRUCK | RVADOR01  RVADDN01  RVADIN01  SAPFM06P |

2. COPY STANDARD FORM **SCC1/COPY FORM FROM SE71** TCODE  
 RUN SE71 -> PROVIDE STANDARD FORM NAME I.E. MEDRUCK HERE, PROVIDE GOLDEN CLIENT I.E. 000 AND PROVIDE CUSTOM FORM NAME I.E. ZMEDRUCK AND EXECUTE. NOW THE FORM IS COPIED

3. CHANGING LANGUAUE IS MANDATORY IN **SE76** TCODE  
RUN SE76/SE71 PROVIDE FORM NAME AND CHANGE LANGUAGE TO DE  
CREATE REQUIRED WINDOW/PARAGRAPH FORMAT/CHARACTER FORMAT  
CHANGE AGAIN TO OUR REQUIRED LANGUAGE I.E. EN

4. MODIFY THE STANDARD FORM AS THE REQUIREMENT  
MAKE THE CHANGES AS PER THE REQUIREMENT IN CUSTOM FORM AND DEVELOP A EXTERNAL SUBROUTINE POOL PROGRAM

5. DEVELOP AN EXTERNAL SUBROUTINE POOL PROGRAM FOR NEW FIELDS DATA FETCHING

6. PLACE COPIED FORM NAME IN **NACE** TCODE.

**106. Difference between scripts and smart forms?**

|  |  |
| --- | --- |
| **SCRIPTS** | **SMARTFORMS** |
| \* SCRIPTS ARE CLIENT DEPENDENT  \*MULTIPLE PAGE FORMAT & PARAGRAPH FORMAT ARE NOT POSSIBLE  \* PAGES AND PARAGRAPH FORMATS CAN’T BE REUSED  \*COLOURING IS NOT POSSIBLE  \*BACKGROUND IMAGE IS NOT POSSIBLE  \*DRAG AND DROP FACILITY IS NOT AVAILABLE  \*CODING CANNOT BE IMPLEMENTED IN SCRIPTS  \*BASED ON DRIVER PROGRAM LOGIC, WE CAN GET THE PRINTOUT  \* CONTROL COMMANDS ARE USED TO CHANGE THE LAYOUT  \* FOR COMPLEX DESIGN WE USE SCRIPTS  \*SCRIPTS CAN BE CONVERTED TO SMARTFORMS  \*SCRIPTS CAN BE DEBUGGED DYNAMICALLY | \* SMARTFORMS ARE CLIENT INDIPENDENT  \* MULTIPLE PAGE FORMAT & PARAGRAPH FORMAT ARE POSSIBLE  \* PARAGRAPH AND CHARACTOR FORMATS CAN BE REUSED AS THEY ARE MAINTAINED IN SMART STYLES  \*COLOURING IS POSSIBLE  \*BACKGROUND IMAGE IS NOT POSSIBLE  \*DRAG AND DROP FACILITY IS NOT AVAILABLE  \*CODING CAN BE IMPLEMENTED IN SMARTFORMS  \*BASED ON GENERATED FM, PRINT PROGRAM, WE PROVIDE THE PRINTOUT  \* CONTRL COMMANDS ARE NOT POSSIBLE  \* FOR COMPLEX CODING WE USE SMARTFORMS  \* SMARTFORMS CANNOT BE CONVERTED TO SCRIPTS  \* SMARTFORMS CAN BE DEBUGGED BY USING STATIC BREAK-POINT |

**107. What are smart forms and their components**?

* It is used to provide the business document hard copies, like Sales order copy, Deliver copy, Invoice copy, PO copy etc.
* Through smart forms also, we can design the labels like Sample labels, Test labels, accepted labels, rejected label, Product info, Price tags, Warnings, Cautions etc.

• Components of Smart Forms

A. Global Settings

B. Pages and Windows.

**108. Different types of Global Settings and windows?**

**A. GLOBAL SETTINGS:**

It is a collection of form attributes

1. Form Attributes

2. Form Interface

3. Global definitions

**Form Attributes:** It specified the administration info, Page style etc.,

**Form Interface:** It is used to declare the variables, parameters, internal tables, Was, Select-options and to handle the errors under the components Import, Export, Tables and Definitions

**Global Definitions:** It is used to declare Types Structures using red, folds, Was, Its and the logic related to the Smart form under the components Types, Global Data, Initialization, form Routines etc.,

**B. WONDOWS AND PAGES:**

These are used to design the pages with the regd. Windows

**Components of Pages are:**

* General Attributes
* Output Options
* Background Picture

**Different Types of Windows in Smart Forms:**

* Main Window
* Secondary Window
* Copies Window
* Final Window.

**109. Exclusive FM of smart forms?**

Ans: Function modules like **SSF\_FUNCTION\_MODULE\_NAME, SSF\_OPEN,** and **SSF\_CLOSE** are exclusive to Smart Forms and manage their runtime execution, unlike SAP script which uses **OPEN\_FORM, WRITE\_FORM,** and **CLOSE\_FORM**.

**110. Difference between Table and Template?**

**Differences between Table & Template:**

* Table is used to print the Main window (i.e., the data will be of dynamic length)
* Template is used to print the static length data of Secondary Windows.

**111. Difference between Alternatives and Folders?**

**Difference between Alternative & Folder:**

* Alternative: It is used to pass True/False conditions
* Folder: It is used to print the Static length data without truncating through secondary windows.

**112. What are the different events in smart forms?**

**Events in Smart forms:**

* **Event on Sort Begin**: It acts like At New field event of sort field.
* **Event on Sort End**: It acts like At the End of event at the sort field.

**113. How to debug smart forms?**

Ans: Debugging the Smart form i.e. It can be debugged only by the **Static Break-Point** key.

**114. How to design tables in smart forms?**

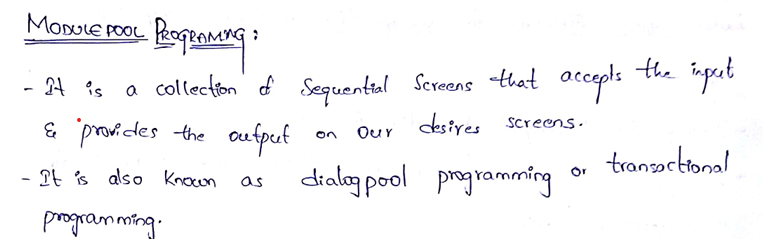
Ans: Use the **Table** node in Smart Forms, define the **line type** using a looped internal table, and design the header, main area (data rows), and footer within the node.

**115.** **How to download and upload smart forms and what is the file format?**

Ans: Use transaction **SMARTFORMS** to export/import via **Utilities > Download/Upload**, and the file format is **.xml**.

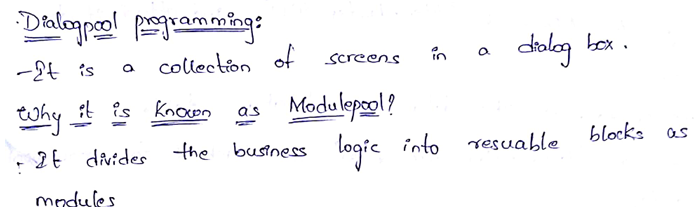
**116. What is module pool programming?**

Ans:



**117. Why MPP is known as Dialog programming?﻿**

Ans:



**118. Why MPP is known as Transactional programming?﻿**

Ans: Without Transactional Code the MPP cannot be executed.

**119. Components of Screen Painter?﻿**

Ans: **Components of Screen Painter:**

* Attributes,
* Element List,
* Flow logic,
* Layout.

**119. Components of Menu Painter?﻿**

Ans: The components of **Menu Painter** are

* Menus,
* Menu Bars,
* Standard Toolbar,
* Application Toolbar,
* Function Keys.

**120.** **What are different screen elements?﻿**

Ans: **Some of the screen – Element are:**

* Reset
* Text
* Input/output fields
* Checkbox
* Radio Button
* Push Button
* Box
* Table Control
* Table Control with Wizard
* Tab strip
* Tab Strip with wizard
* Sub screen
* Status
* Customer Container.

**121. What are screen attributes?**

Ans. Screen attributes are properties of a screen that control its behaviour, such as **screen number, description, screen type (normal, modal, sub screen), next screen, and hold data option** also **table type, resizing, separators, line selection and column selection**.

**122. How to make screen field as password protected?﻿**

Ans. Set the **screen field's display attribute** to **"Password"** (type = 3) in the **Screen Painter** to mask input with asterisks.

**123. How to make screen as disable?﻿**

Ans: To make a screen **disabled** (non-editable), set the **Screen Type** to **"Modal dialog box"** and use ABAP logic in PBO to set all fields’ screen-input = 0 in a LOOP AT SCREEN block.

**124. How to make screen field as output only?**

Ans: Set the screen field's attribute **"Output only"** in **Screen Painter**, or set screen-input = 0 for that field in the **PBO** using ABAP code.

**125. What are the events of MPP?**

The main events of Module Pool Programming (MPP) are:

1. **PBO (Process Before Output)** – Executes before screen display.
2. **PAI (Process After Input)** – Executes after user input.
3. **POH (Process on Help-Request)** – For F1 help.
4. **POV (Process on Value-Request)** – For F4 help.

**126. Difference between Table control and Tab strips?﻿**

Ans: **Table Control** displays data in a tabular (grid-like) format with scrollable rows, while **Tab strips** organize screens into multiple tabs to group related input fields or controls.

**127. Declaration related to Table control, Tab strip and Sub screen in ABAP editor, PBO and PAI?**

Here’s a single-sentence answer for each:

* **Table Control**: Declare using CONTROLS <name> TYPE TABLEVIEW USING SCREEN <num> in ABAP Editor; handle data flow in **PBO** and user actions in **PAI**.
  + **CONTROLS:** tc\_data TYPE TABLEVIEW USING SCREEN 100.
  + **LOOP AT** it\_data **INTO** wa\_data **WITH CONTROL** tc\_data **CURSOR** tc\_data-current\_line.

**MODULE** fill\_data.

**ENDLOOP.**

* **Tab strip**: Declare using CONTROLS <name> TYPE TABSTRIP in ABAP Editor; manage active tab logic in **PBO/PAI**.

CONTROLS: ts\_tab TYPE TABSTRIP.

DATA: tab\_active TYPE syucomm.

**PBO:**

ts\_tab-activetab = tab\_active.

**CASE** ts\_tab-activetab.

**WHEN** 'TAB1'. " Load tab 1 subscreen

tab\_active = 'TAB1'.

**WHEN** 'TAB2'. " Load tab 2 subscreen

tab\_active = 'TAB2'.

**ENDCASE.**

**PAI:**

**ts\_tab-activetab = tab\_active.**

**CASE** sy-ucomm.

**WHEN** 'TAB1' OR 'TAB2'.

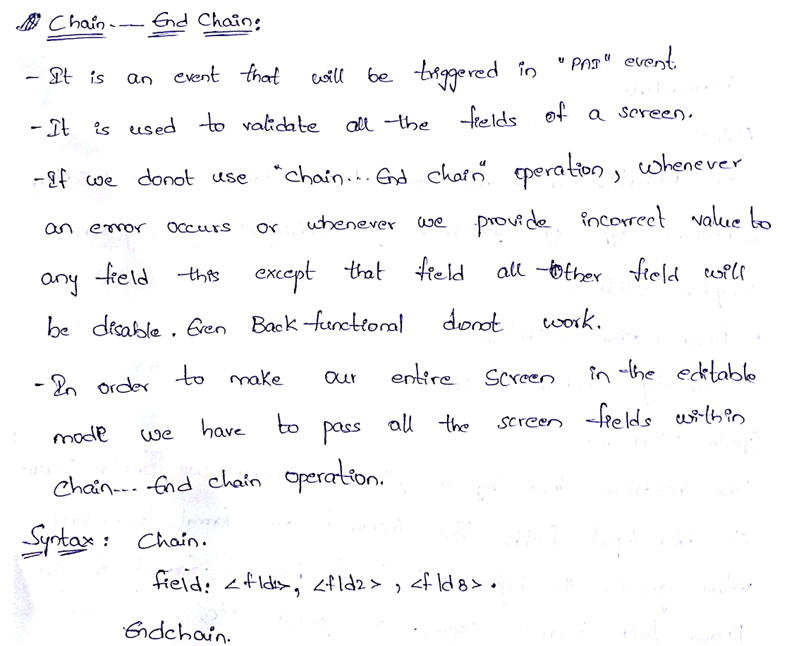
tab\_active = sy-ucomm**.**

**ENDCASE.**

* **Sub screen**: Declare sub screen area in Screen Painter; call it in **PBO** using CALL SUBSCREEN <area> INCLUDING <prog> <screen> and in **PAI** using CALL SUBSCREEN <area>.
  + **CALL SUBSCREEN** sub\_area**.**

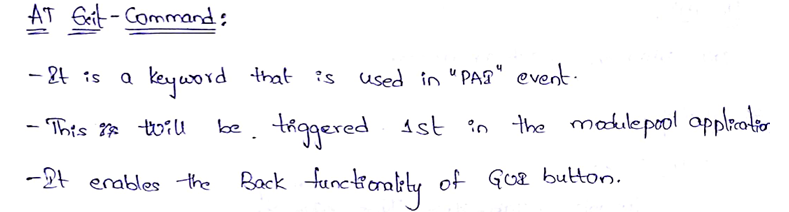
**128. What is the Purpose of Chain...Enchain?﻿**

Ans:



**129. What is At-exit command and where we use it?﻿**

Ans:



**130. What are the different types of validations in MPP?﻿**

Ans: **Different types of validations in MPP:**

* System validation
* ABAP Editor Validation
* Flow logic Editor Validation.

**131. Syntax of flow logic editor validation?**

Ans:

1. **Flow Logic (PAI) Syntax for Validation:**

PROCESS AFTER INPUT.

**MODULE** user\_command\_0100.

**MODULE** validate\_input\_field.

1. **In ABAP Module (e.g., validate\_input\_field):**

**MODULE** validate\_input\_field INPUT.

**IF** field\_name IS INITIAL.

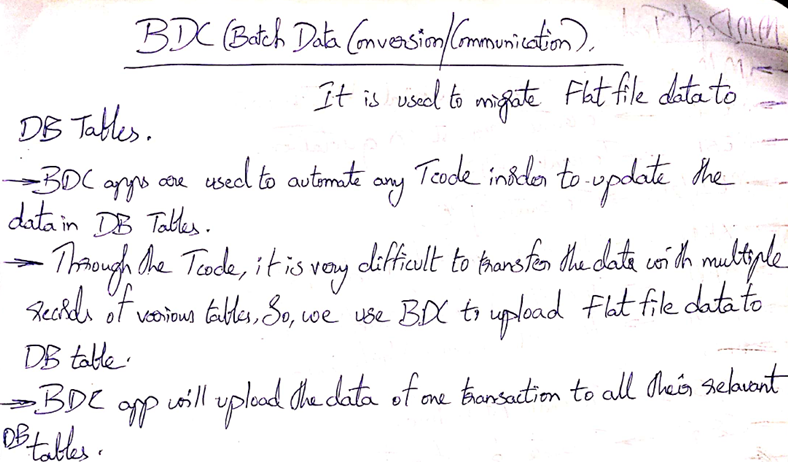
**MESSAGE** 'Field cannot be empty' TYPE 'E'.

**ENDIF.**

**ENDMODULE.**

**132.** **What is BDC?﻿**

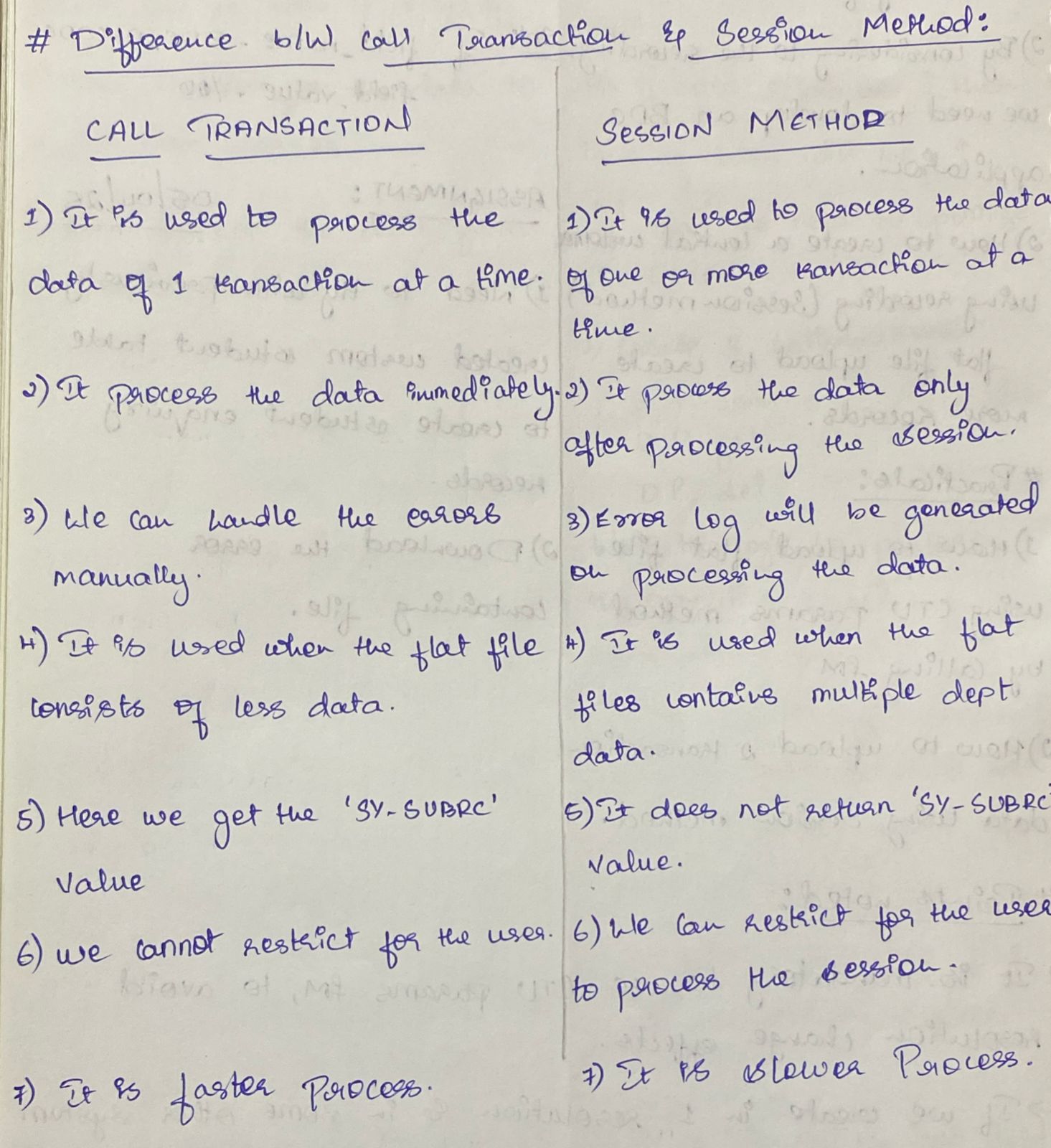
Ans:



**133. Different approaches in BDC to upload data and difference between them?**

Ans: **Approaches in BDC:**

1. **Call Transaction Method**
2. **Session Method.**



**134. What are DDIC structures that are used while working with BDC applications?**

The main **DDIC structures used in BDC applications** are:

* **BDCDATA** – Holds screen field values for BDC recording.
* **BDCMSGCOLL** – Stores messages returned during BDC processing.

These structures are used in programs that implement CALL TRANSACTION or BDC\_INSERT logic.

**135.Complete syntax of call transaction method?﻿**

Ans: **complete syntax of the CALL TRANSACTION method** in BDC:

**DATA**: lt\_bdcdata TYPE TABLE OF bdcdata,

lt\_msg TYPE TABLE OF bdcmsgcoll,

lv\_trans TYPE sy-tcode.

lv\_trans = 'T-CODE'. "Example: 'XD01', 'FB01', etc.

**CALL TRANSACTION** lv\_trans USING lt\_bdcdata

**MODE**  'A' " A = All screens, N = No screens, E = Errors only

**UPDATE** 'S' " S = Synchronous, A = Asynchronous, L = Local

**MESSAGES** INTO lt\_msg.

**136. What is the purpose of ctu\_paramas structure?﻿**

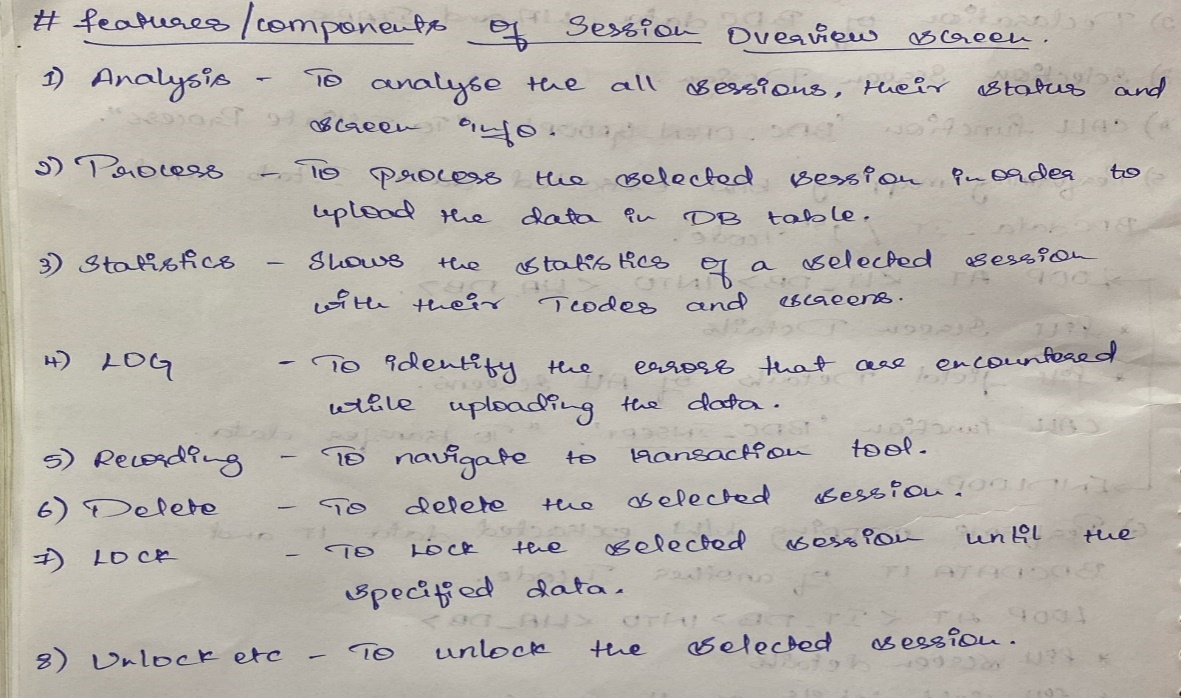
Ans: The **CTU\_PARAMS** structure is used to control runtime options for **BDC transactions** when calling CALL TRANSACTION USING with more flexibility.

It allows you to set parameters like **mode, update type, messages, and others** in a **single structure**, instead of passing them individually.

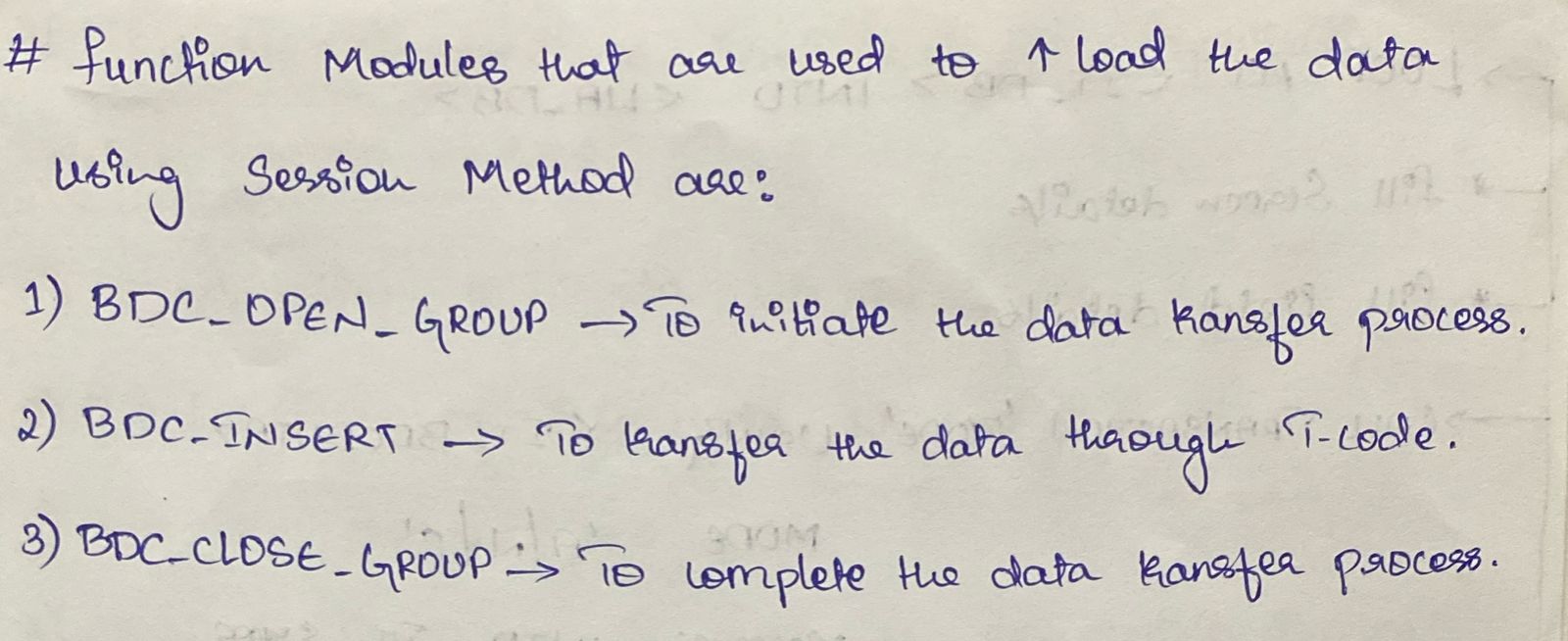
**137. What is recording and why we need it?**

Ans: **Recording** is the process of capturing user actions (screen navigation and field inputs) in a transaction to generate the required **screen flow and field mapping** for a **BDC or LSMW** upload.

**138.What is the Session overview and their components?**

 **﻿**

**139. Exclusive FM of BDC?﻿**



**140.How to import and export recording?﻿**

Ans: Use transaction **SHDB → Utilities → Download/Upload** to export or import recordings as **.txt files** between systems.

**141.How to identify standard BAPIs?﻿**

Ans: Use transaction **BAPI** or search in **SE37/SE80** with prefix **BAPI\*** to identify standard BAPIs provided by SAP.

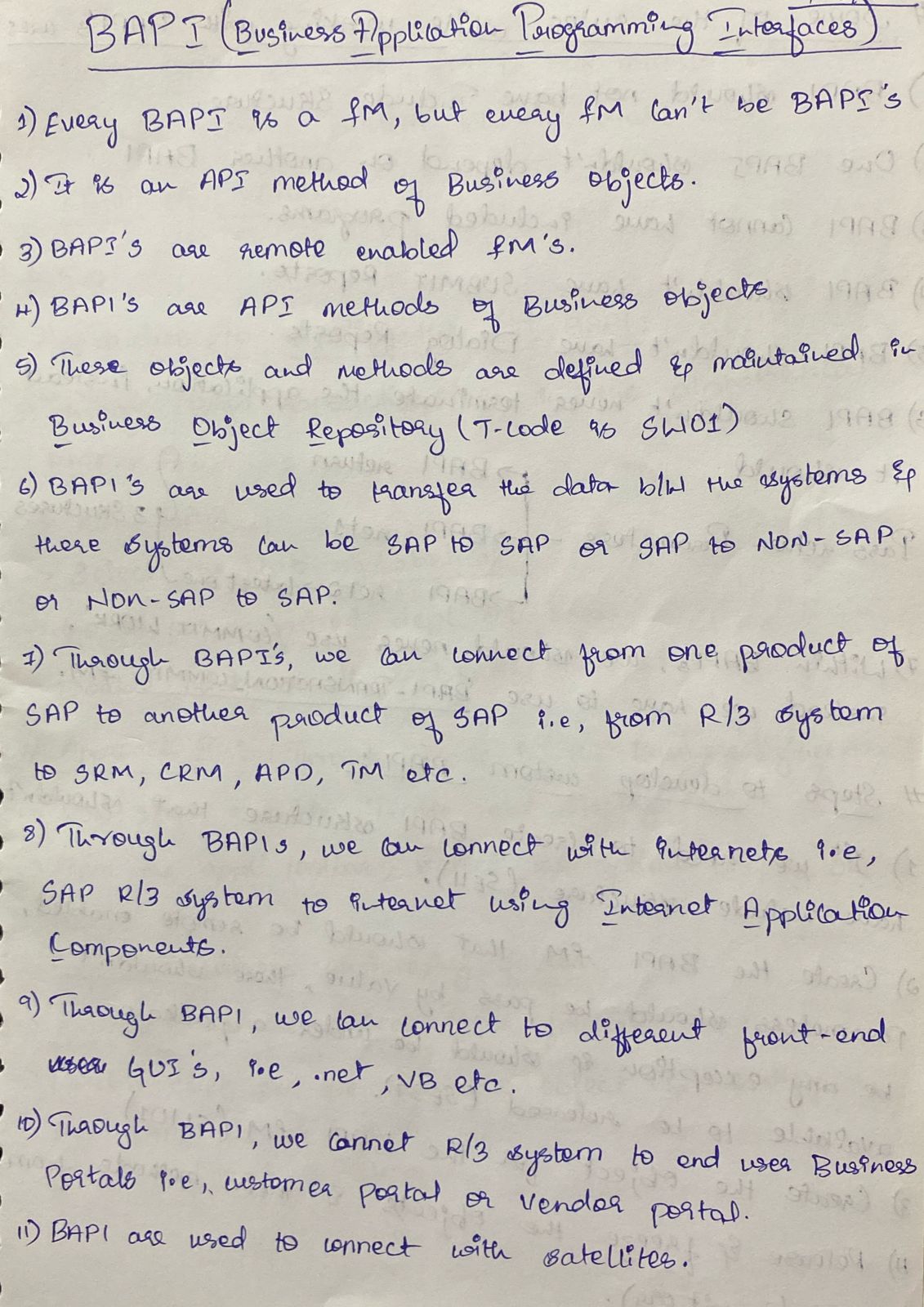
**142. Tell me some of standard BAPIs which you have come a crossed?**

* **BAPI\_CUSTOMER\_CREATEFROMDATA1** – Create customer master
* **BAPI\_CUSTOMER\_CHANGEFROMDATA1** – Change customer master
* **BAPI\_CUSTOMER\_EXISTENCECHECK** – To Check he existence customer master
* **BAPI\_CUSTOMER\_DELETE** – Delete customer master
* **BAPI\_SALESORDER\_CREATEFROMDAT2** – Create a sales order
* **BAPI\_SALESORDER\_CHANGE** – Change an existing sales order
* **BAPI\_SALESORDER\_GETDETAIL** – Get detailed data of a sales order
* **BAPI\_CUSTOMER\_GETDETAIL1** – Get customer master data

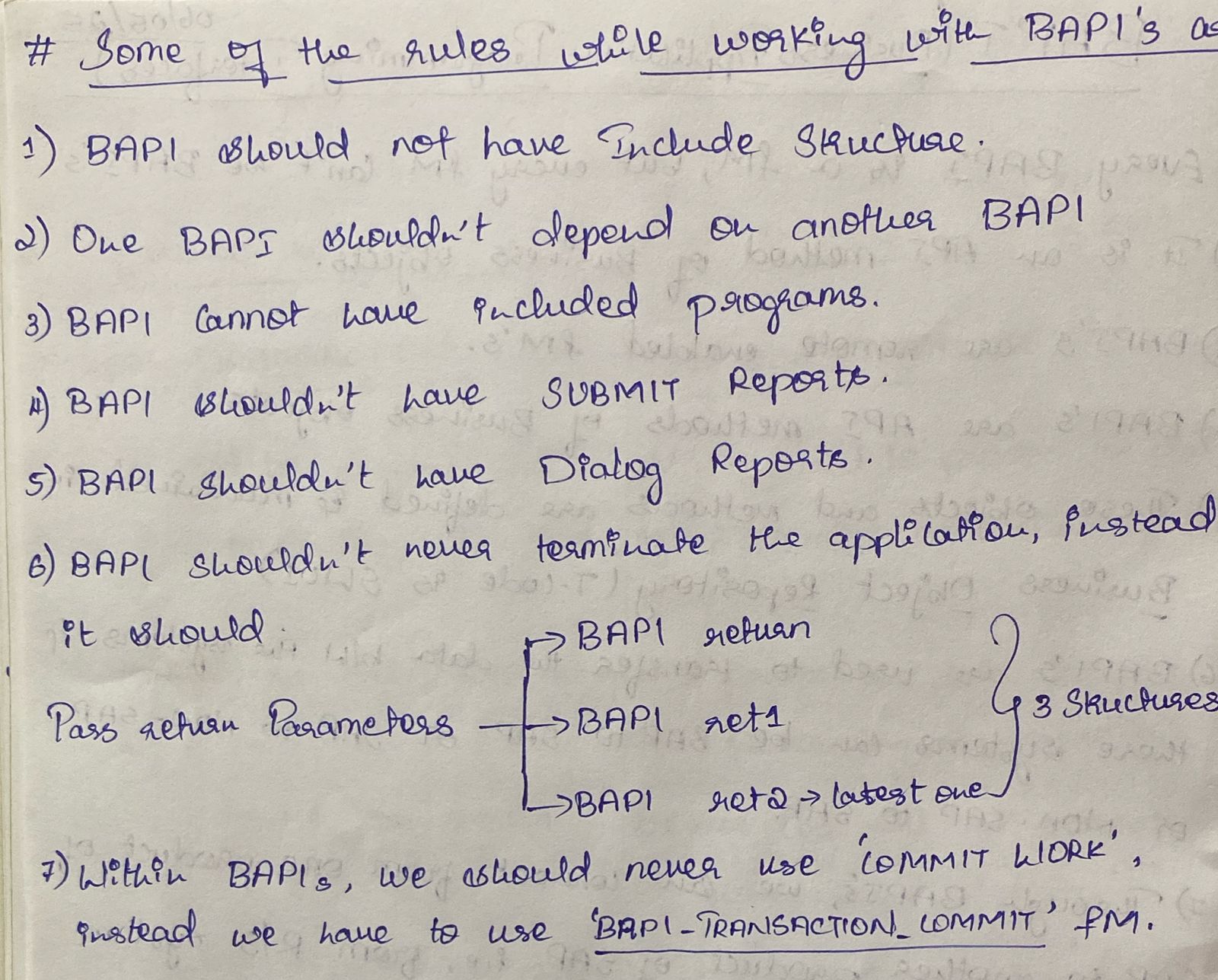
**143. What are BAPI methods?﻿**

Ans: **BAPI methods** are **functions of business objects** in the **Business Object Repository (BOR)** that define specific operations (like create, change, read) which can be accessed externally via **function modules**.

**144. What is BAPI?﻿**



**145.What are the rules in developing BAPIs?﻿**



**146.How to work with custom fields of standard tables using BAPIs?﻿**

Ans: To work with **custom fields in standard tables using BAPIs**, use the **BAPI extension parameter** (EXTENSIONIN/ EXTENSIONOUT) to pass the custom field values mapped to the correct structure (like BAPE\_\* and BAPI\_TE\_\*).

**147.What are Business Object Repository?﻿**

Ans: The **Business Object Repository (BOR)** is a centralized SAP repository that contains **business objects** (like Customer, Sales Order) and their **methods, events, and attributes**, used mainly for **BAPIs** and **workflow integration**.

**148.Components of BOR?﻿**

The main **components of the Business Object Repository (BOR)** are:

1. **Business Objects** – Represent real-world entities (e.g., BUS2032 for Sales Order).
2. **Methods** – Actions that can be performed (e.g., Create, Change).
3. **Attributes** – Data fields describing the object (e.g., Order Number).
4. **Events** – Triggers for workflows (e.g., Created, Changed).
5. **Interfaces** – Group reusable methods and attributes.
6. **Key Fields** – Unique identifiers for each object instance.

**149.Tell some of the standard BOR?﻿**

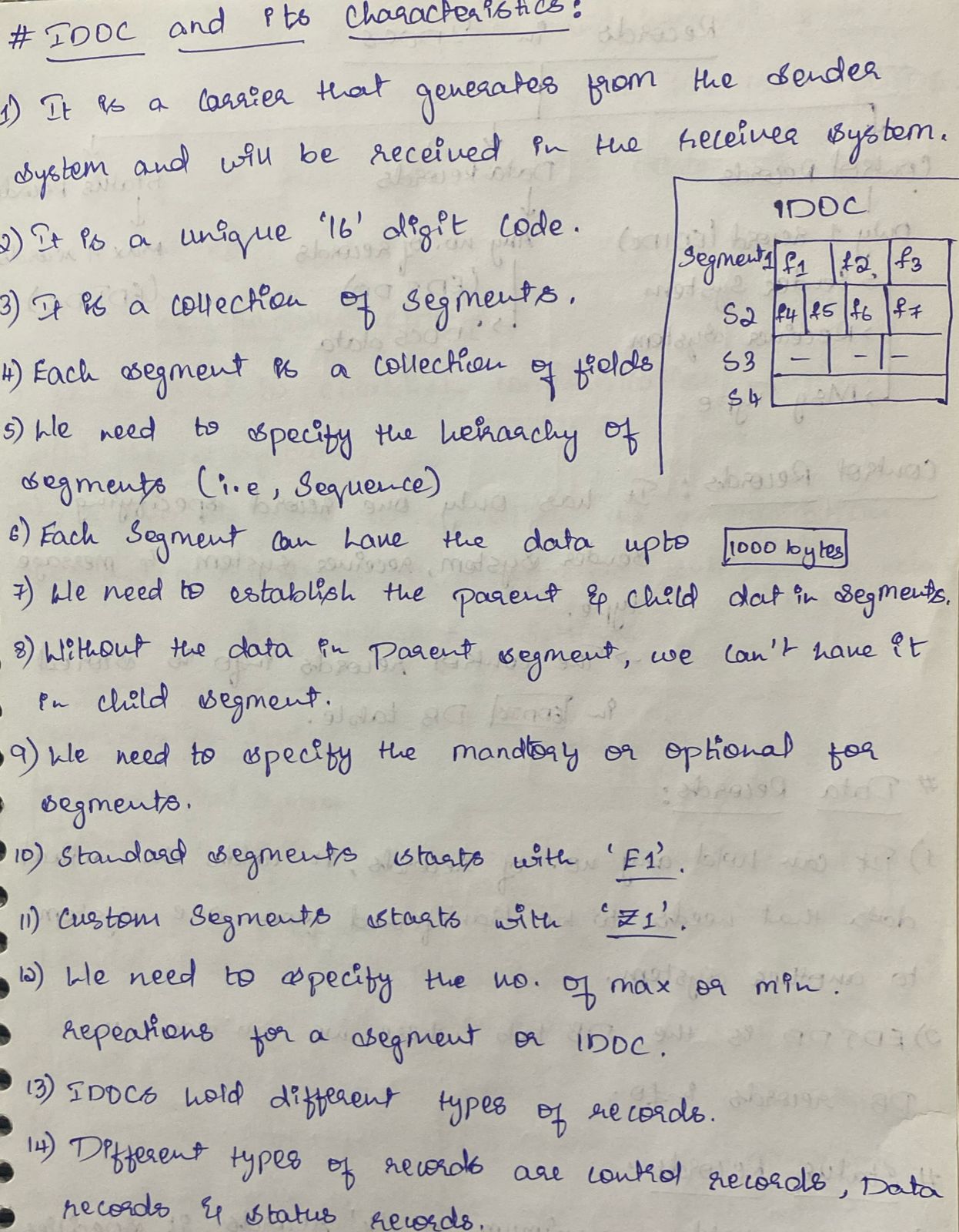
**Here are some commonly used standard Business Objects (BOR objects) in SAP:**

1. **BUS1001** – Material
2. **BUS1006** – Vendor
3. **BUS1007** – Customer
4. **BUS2012** – Purchase Order
5. **BUS2032** – Sales Order
6. **BUS2081** – Billing Document
7. **BUS1065** – Equipment
8. **BUS2001** – Project Definition
9. **BUS6035** – Service Notification
10. **BUS2010** – Purchase Requisition.

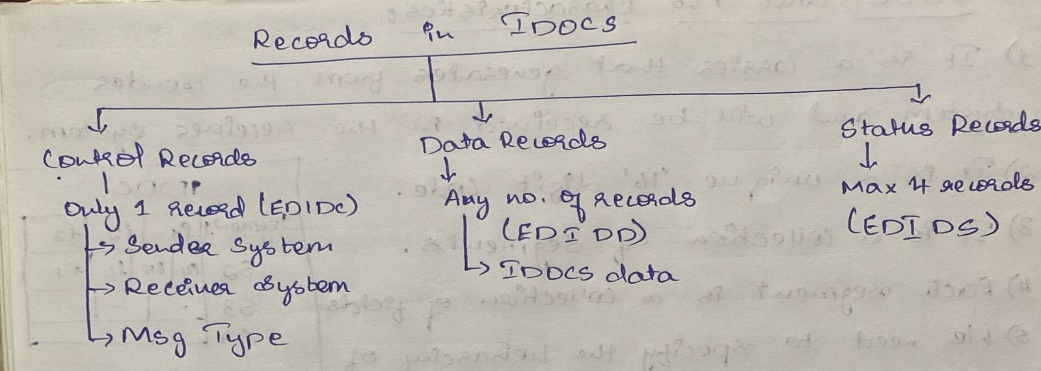
**150. What is ALE/IDocs?﻿**

Ans. **ALE (Application Link Enabling)** is a technology in SAP used for **distributing data across different SAP systems**, while **IDocs (Intermediate Documents)** are the **data containers** used to transfer information between systems in ALE or EDI scenarios.

**151. What are IDocs and their characteristics?﻿**



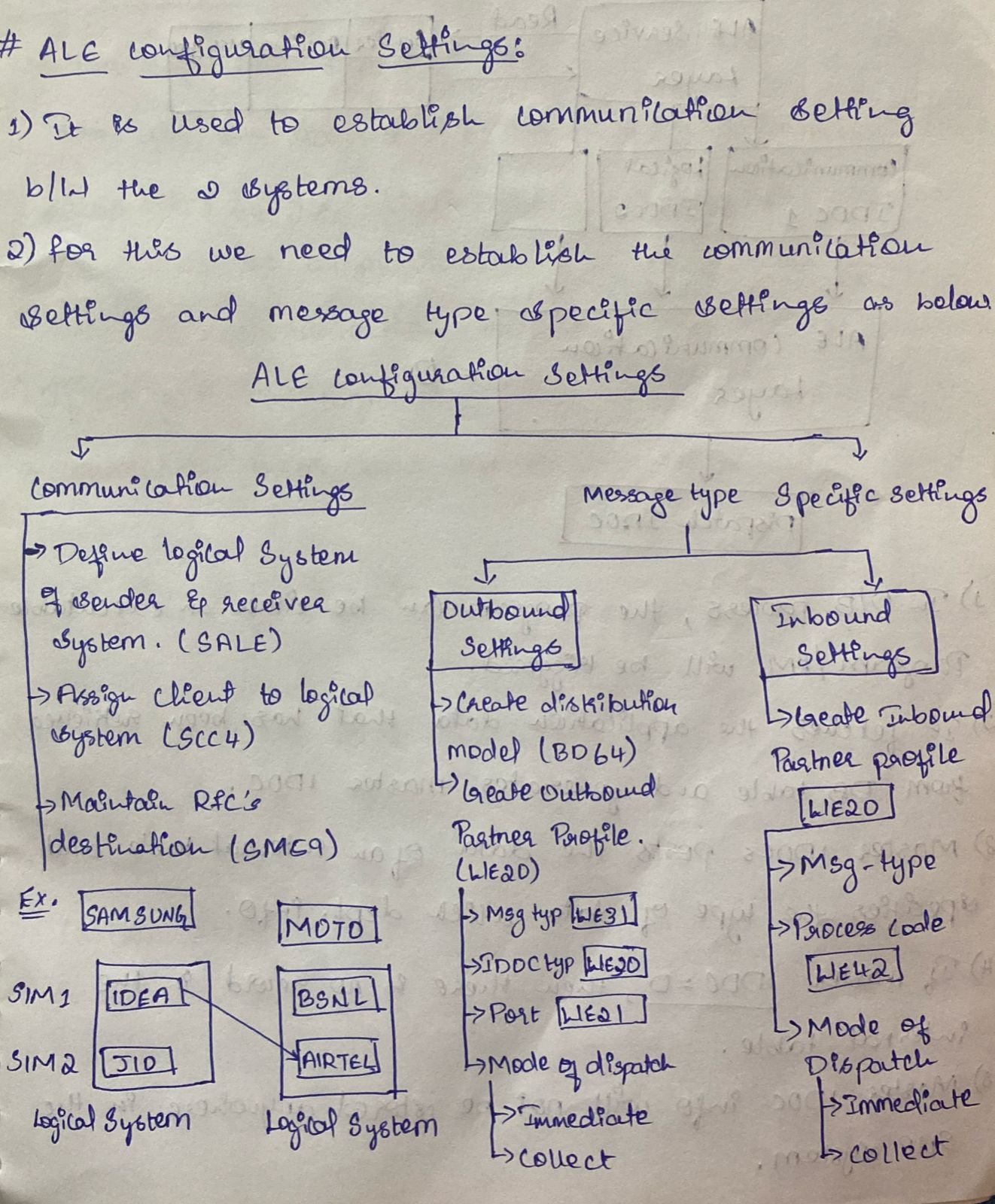
**152.Different types of records in IDocs and their DB tables?﻿**



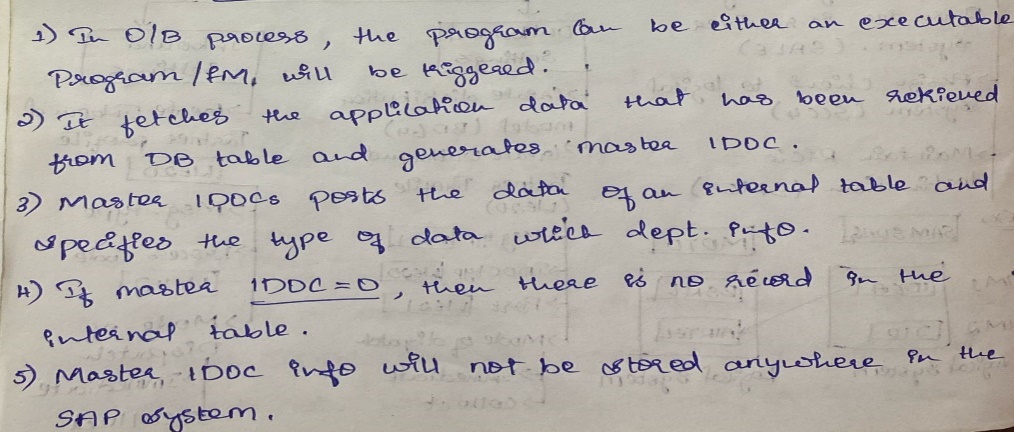
**153.Different types of IDocs?﻿**

* Master IDcos -> Holds master table data
* Custom IDocs -> Holds Custom IDoc data
* Extended Data -> Holds standard and Custom IDocs
* Standard IDoc -> Transfer Master table from sender system to receiver system.

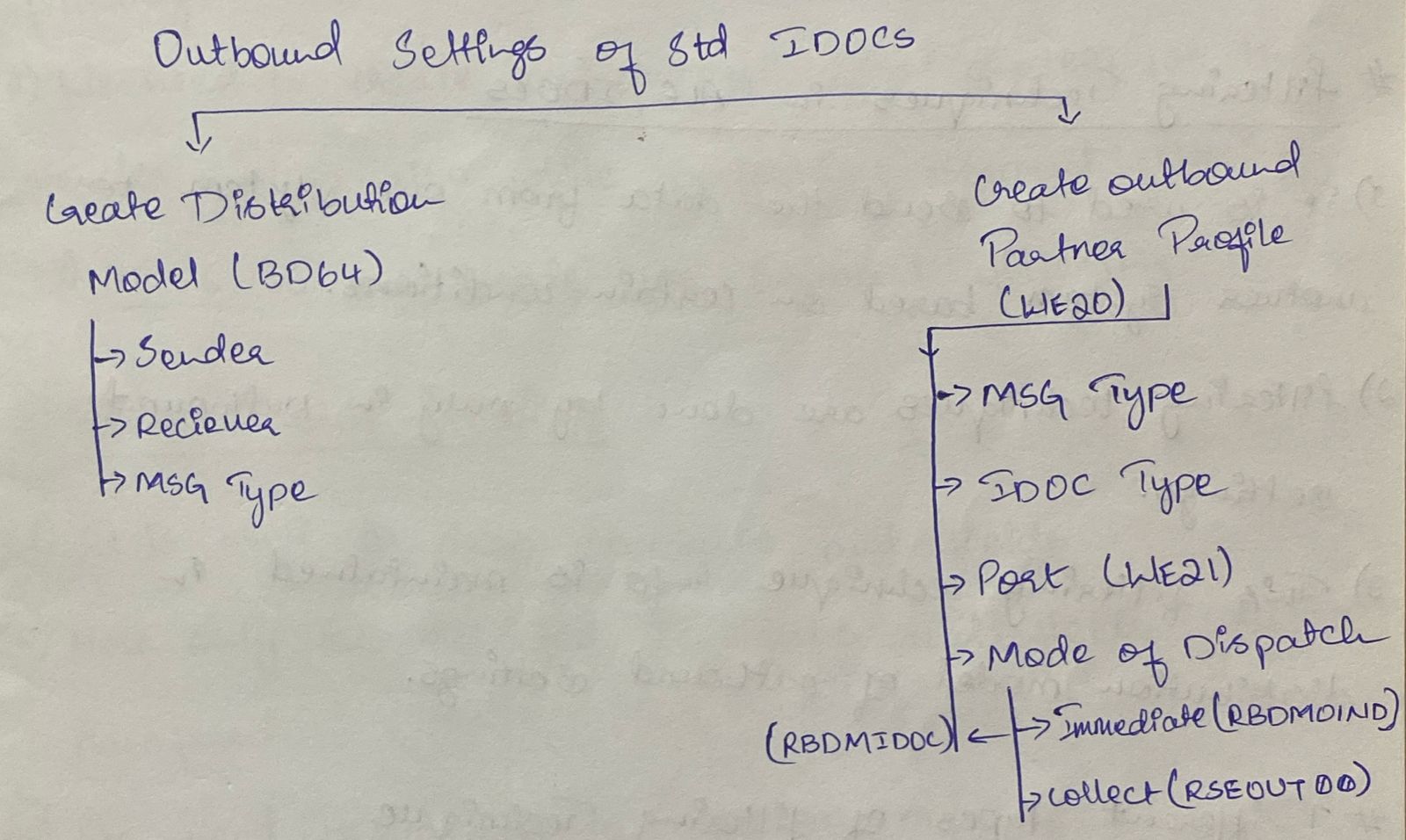
**154. ALE configuration settings for standard master IDocs?﻿**



**155.What is an outbound process?﻿**



**156.Outbound settings for master IDocs?**



**157.Tell me some of the standard IDocs applications?﻿**

* **Material Master** MATMAS
* **Customer Master** DEBMAS
* **Vendor Master** CREMAS.

**158.What is distribution model?﻿**

* **Sender logical system**
* **Receiver logical system**
* **Message type (e.g., MATMAS, DEBMAS).**

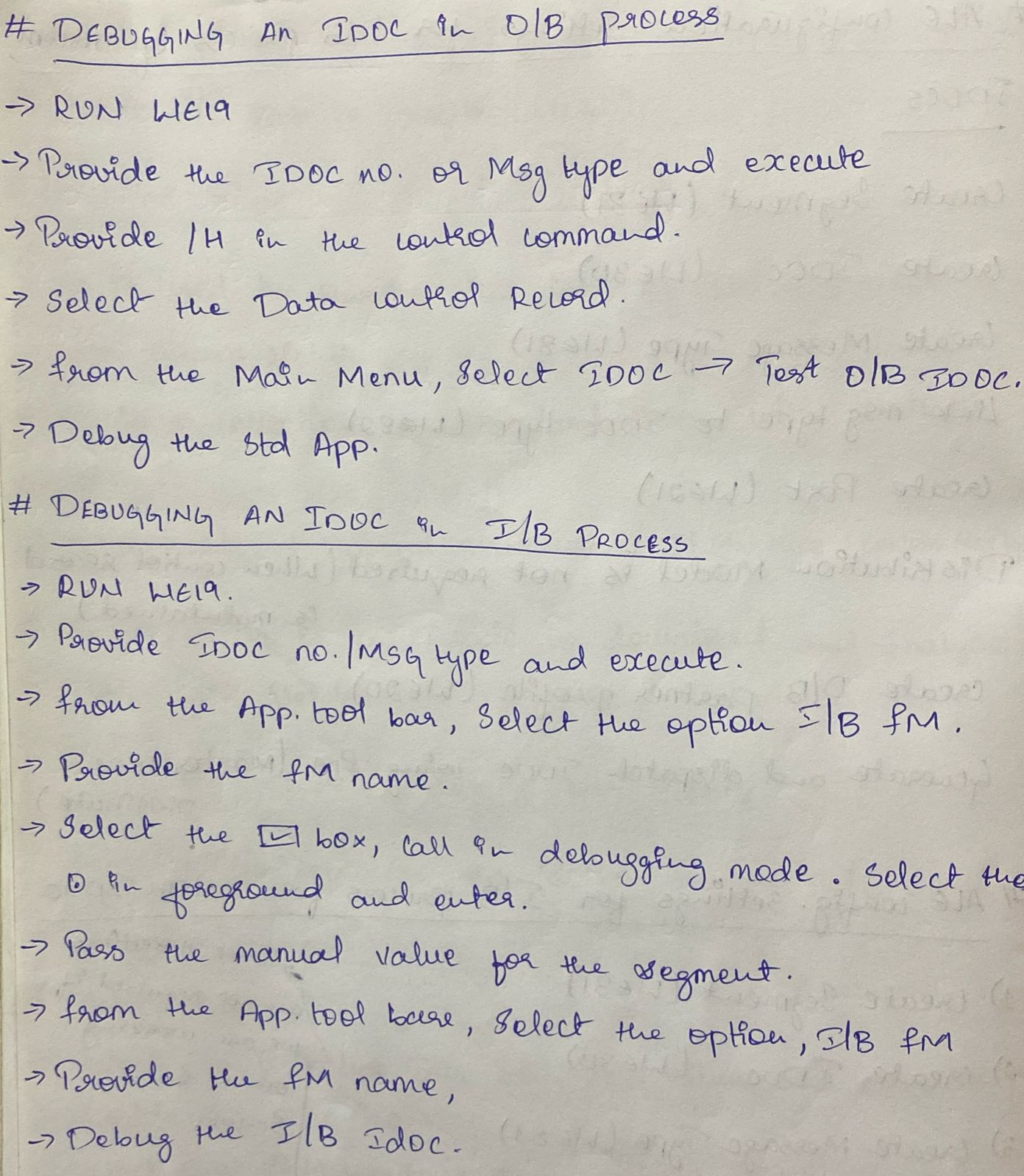
**159.What is Port and different types of port?﻿**

* **Transactional RFC ->** B/w SAP To SAP.
* **File RFC ->** B/w SAP TO NON-SAP
* **XML ->** B/w SAP and JAVA Etc.

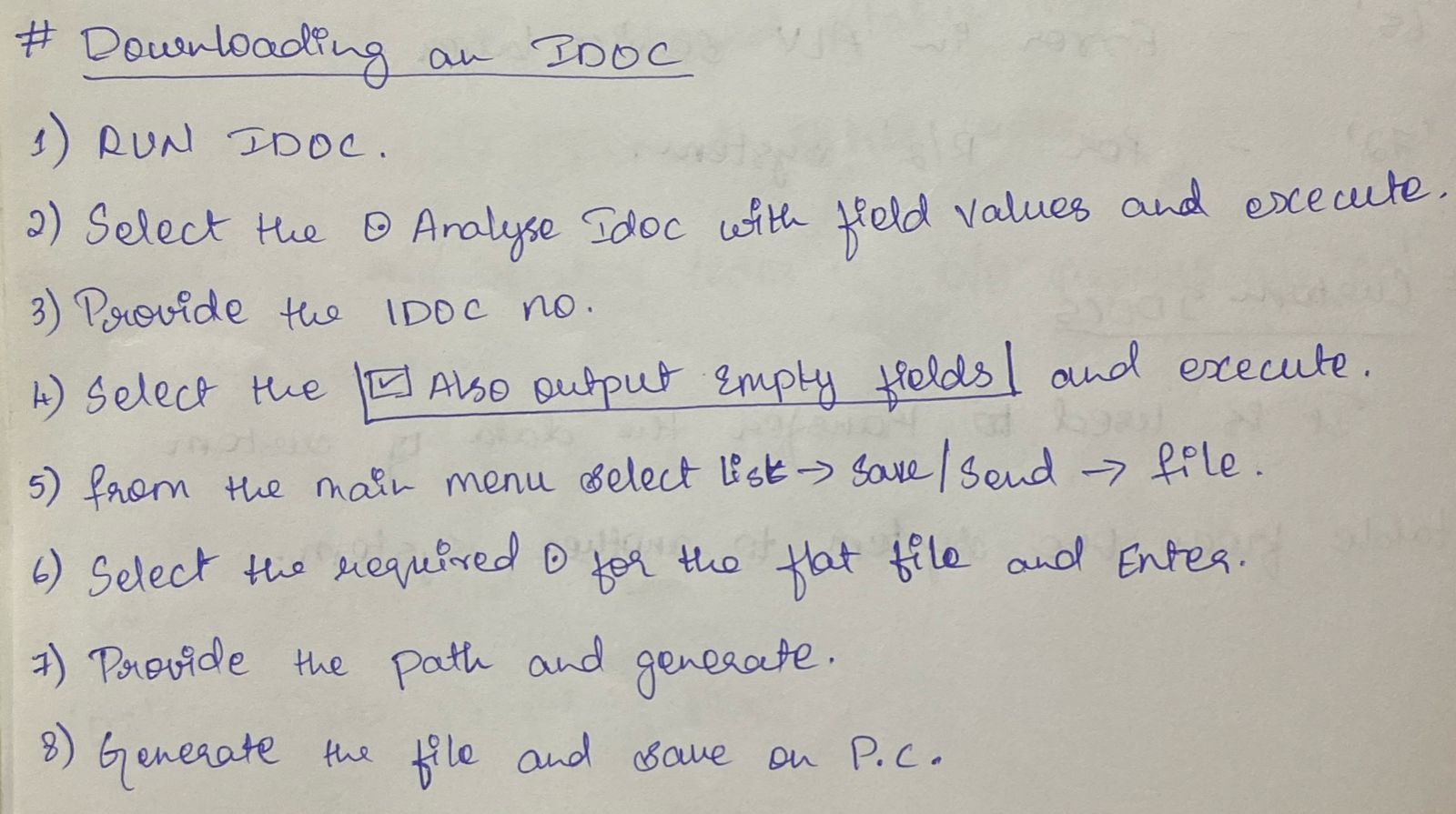
**160. How to test IDocs?**

Ans: Use **WE19** for testing IDocs manually, **BD10/BD12** for triggering outbound IDocs, and **WE02/WE05** to monitor their status.

**161. How to debug IDoc?**



**162.** **How to download IDoc data?**



**163.** **Tell me some of the status codes of outbound and inbound process?**

Here are some common **IDoc status codes** for **outbound and inbound processes**:

**Outbound IDoc Status Codes:**

| **Code** | **Meaning** |
| --- | --- |
| **01** | IDoc generated |
| **03** | Data passed to port OK |
| **12** | Error in IDoc processing |
| **18** | Reprocessing was successful |

**Inbound IDoc Status Codes:**

| **Code** | **Meaning** |
| --- | --- |
| **50** | IDoc received |
| **51** | Application document not posted (error) |
| **53** | Application document posted successfully |
| **64** | IDoc ready to be transferred to application |
| **68** | IDoc marked for deletion |

**164.** **Tell me standard applications used while outbound process to process IDoc? Or**

**165.** **Tell me standard application used while inbound process to process IDoc?**



**166.** **Configuration settings for Change pointer techniques?**



**167.What are the steps of custom IDoc outbound process with t-codes? AND**

**168.What are the steps of custom IDoc inbound process with t-codes?﻿**



**169.** **What is mean by Archiving?**

Ans: **Archiving** in SAP means **removing unused or outdated data from the active database** and storing it in a separate archive file for long-term retention, to improve system performance and reduce database size.

**170. What is Serialisation?**

Ans: **Serialization** in SAP IDocs ensures that **related IDocs are processed in a specific sequence** based on a defined message flow, to maintain **data consistency** (e.g., posting header before items).

**171. How to work with transactional data using IDocs?**

Ans: Transactional data is processed using IDocs by configuring message types, partner profiles, and process codes to send or receive documents like sales orders, invoices, and deliveries automatically or via test tools like WE19.

**172. What is object-oriented ABAP?**



**173. Difference between class and object?**

Ans: A **class** is a blueprint that defines properties and behaviours, while an **object** is a specific instance created based on that class.

**174. Components of class?**

Ans: The main components of a class in ABAP are:

* **Attributes,**
* **Methods,**
* **Events,**
* **Constructors,**
* **Interfaces.**

**175.** **Different types of classes?**

Ans: There are **two main types of classes** in ABAP:

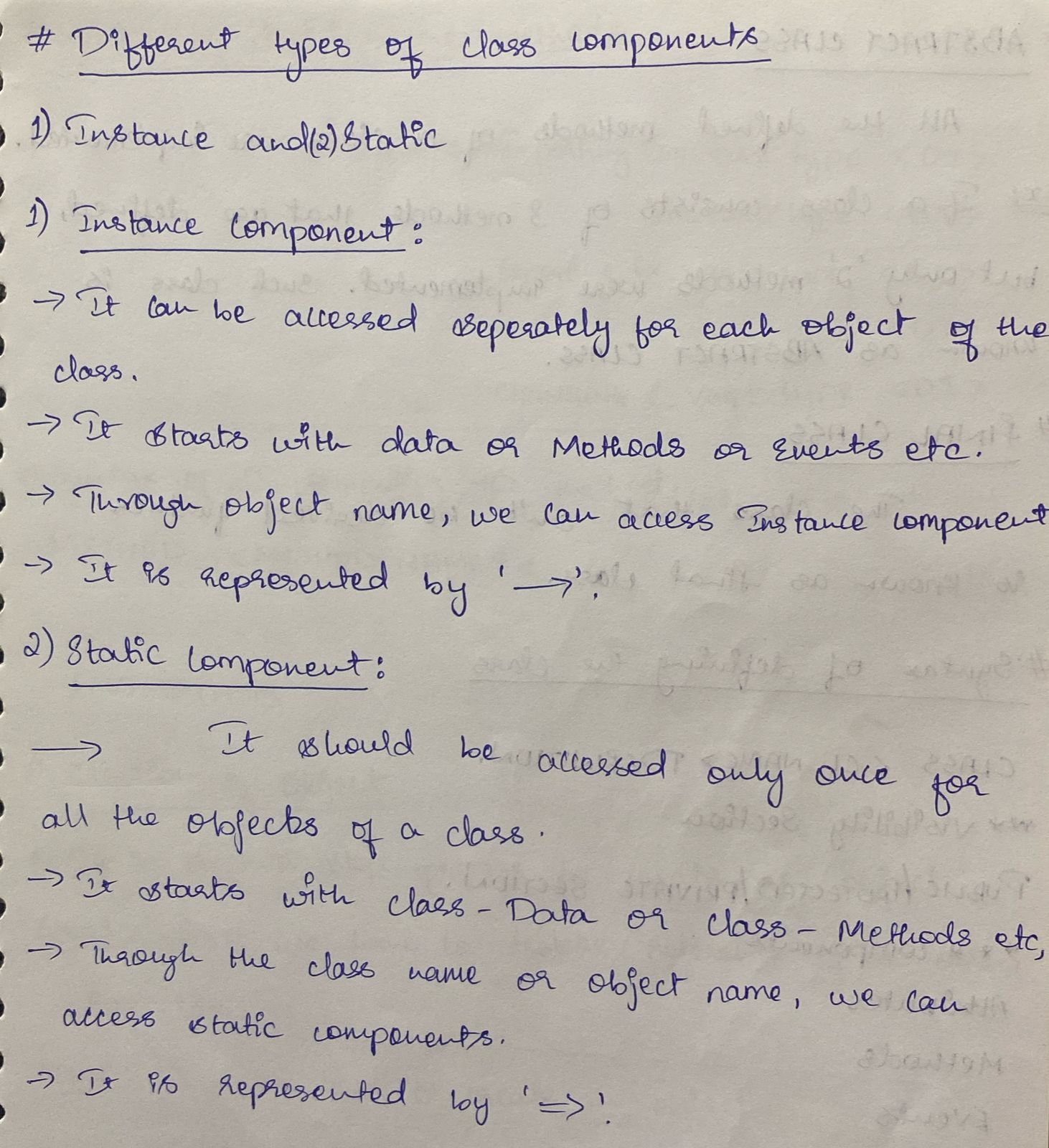
1. **Global Classes** – Defined in the Class Builder (SE24), reusable across programs.
2. **Local Classes** – Defined within ABAP programs, used only in that program.

**176. What are the different visibility sections?**

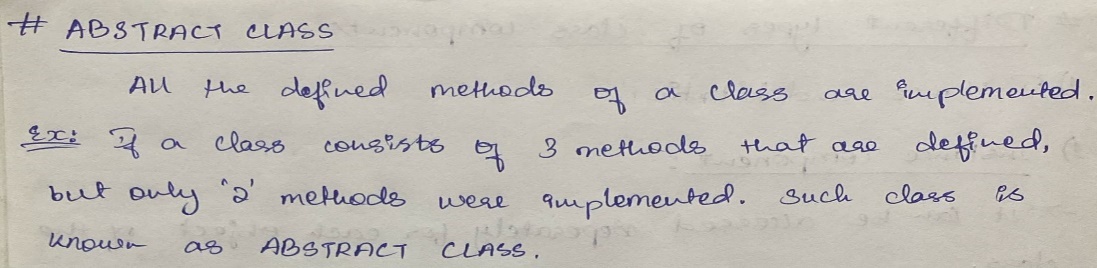
Ans: The different visibility sections in ABAP classes are:

1. **PUBLIC** – Accessible from anywhere.
2. **PROTECTED** – Accessible only within the class and its subclasses.
3. **PRIVATE** – Accessible only within the class itself.

**177.Difference between Instance and Static Components?﻿**



**178.What is an Abstract Class?**



**179.** **What is Final Class?**

Ans: A class that can’t be derived further is known as final Class.

**180. What is Constructor?**

Ans: It is a default method, when ever we create an object by default it will call certain methods.

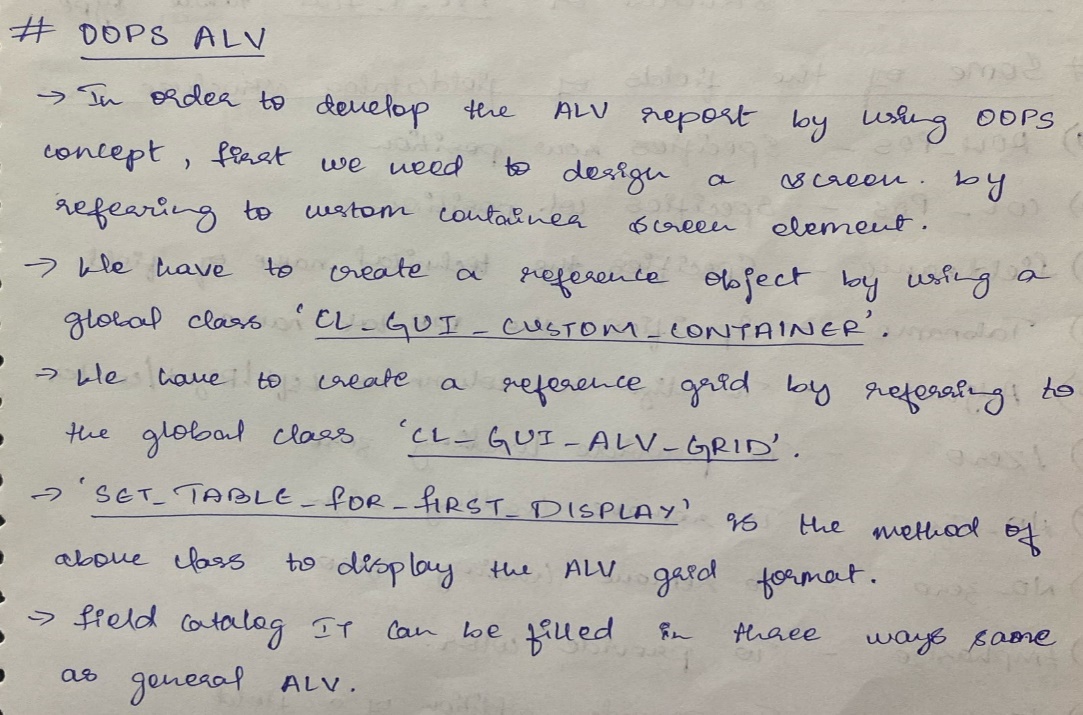
**181. Difference between Instance and Static Constructors?**

And: An **instance constructor** (CONSTRUCTOR) is called when an object is created, while a **static constructor** (CLASS\_CONSTRUCTOR) is called automatically once per class when any static component is accessed for the first time.

**182. Tell me some of the standard classes?**

* **CL\_GUI\_ALV\_GRID** – Used for displaying ALV in a custom container (e.g., docking container or screen).
* **CL\_SALV\_TABLE** – Simplified ALV for quick output (no need for screen programming).
* **CL\_GUI\_CUSTOM\_CONTAINER** – Required to host CL\_GUI\_ALV\_GRID on the screen.
* **CL\_ALV\_TABLE\_CREATE** – Helps generate dynamic internal tables for ALV.

**183. How to develop ALV reports using OABAP?**



**184. What is an Inheritance?**

Ans: Deriving Sub-class from super class is known as **Inheritance**.

**185. What is a singleton class?**

Ans: A **singleton class** in ABAP is a class designed to **allow only one instance** to be created.

**186. What is a persistence class?**

Ans: A **persistence class** in ABAP is a special global class that is used to **map database table rows to objects**, enabling you to perform **object-oriented data access** using the **Persistence Framework** instead of direct SQL.

**187. What is the difference between me and super reference?**

The difference between me and super references in ABAP is:

* **me** refers to the **current instance** of the class (like this in other languages).
* **super** is used to **call methods of the superclass** from within a subclass.

**188. How can we access the methods of other class?**

Ans: We can access the methods of another class by **creating its object** (for instance methods) or by **calling directly using the class name** (for static methods).

**189. What are events and Syntax to register the events?**

Ans: Events in ABAP allow a class to **trigger actions** that can be handled by **other objects**, promoting loose coupling and dynamic behaviour.

* **SET HANDLER lo\_handler->on\_evt\_done FOR lo\_event.**

**190.** **How can we handle the errors in classes?**

Ans: We can handle errors in classes using **exception handling** with TRY...CATCH blocks and by **raising exceptions** using RAISE EXCEPTION.

**191. What is an Interface?**

Ans: An **interface** in ABAP is a collection of **method, attribute, and constant declarations** that defines a **contract** which implementing classes must follow, **without providing implementation**.

**192. What should be the visibility section for Abstract Class and Interface?**

* For an **abstract class**, the visibility section can be **PUBLIC**, **PROTECTED**, or **PRIVATE** depending on the design.
* For an **interface**, the visibility section is always **PUBLIC** by default and **cannot be changed**.

**193. What are Enhancements and different types of enhancements?**

Ans: **Enhancements** in ABAP are techniques to modify or extend the standard SAP functionality **without changing the original code**.

**Types of Enhancements:**

1. **User Exits** – Predefined FORM routines in SAP that can be enhanced using includes (mostly in SD/MM modules).
2. **Customer Exits** – Function modules (e.g., EXIT\_…) that allow custom code via projects in CMOD.
3. **BADI (Business Add-Ins)** – Object-oriented enhancements using interfaces, allowing multiple implementations.
4. **Explicit Enhancements** – Custom code placed at predefined ENHANCEMENT-POINT or ENHANCEMENT-SECTION in the SAP code.
5. **Implicit Enhancements** – SAP provides implicit spots at method start/end, function modules, etc., for enhancement without needing explicit definition.
6. **Modification (Not Recommended)** – Direct changes to standard SAP code (not considered a true enhancement technique).

**194. Difference between Enhancement point and Enhancement section?**

* Use **Enhancement Point** when you want to **add logic** and
* Use **Enhancement Section** when you want to **replace logic**.

**195. What are Exits and different types of Exits?**

Ans: It is use to provide additional feature in the standard application where SAP has given the slot.

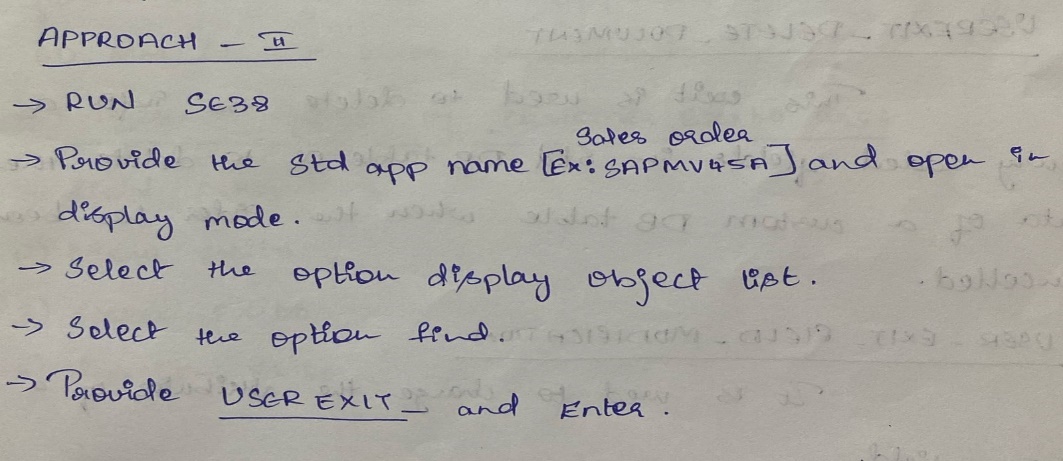
Different types:

* User Exit.
* Customer Exit.

**196. Difference between User and Customer Exits?**

| **Point** | **User Exits** | **Customer Exits** |
| --- | --- | --- |
| 1 | Older enhancement technique, mainly used in SD module. | Newer enhancement technique introduced later to standardize enhancements. |
| 2 | Implemented as subroutines (FORM...ENDFORM) in include programs. | Implemented using function modules with naming convention EXIT\_.... |
| 3 | Cannot be managed via transaction SMOD/CMOD. | Managed using transactions SMOD (define) and CMOD (implement). |
| 4 | Not explicitly provided for all business scenarios. | Predefined by SAP at specific enhancement points, allowing standardized use. |
| 5 | Modifications are overwritten during SAP upgrades. | Upgrades are safer, as code is added in customer namespace (Z...). |
|  |  |  |

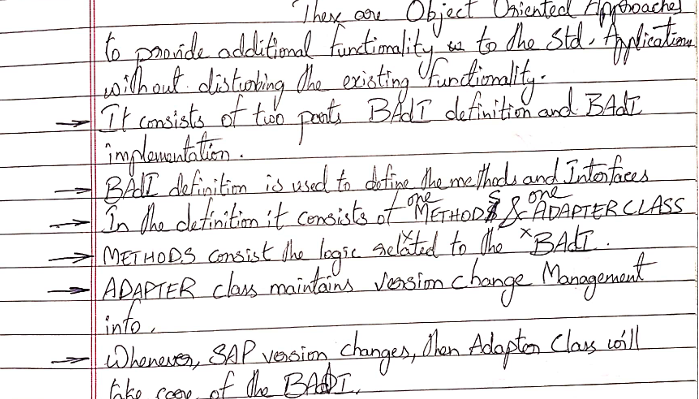
**197. How to Identify User exits and where to implement it?**



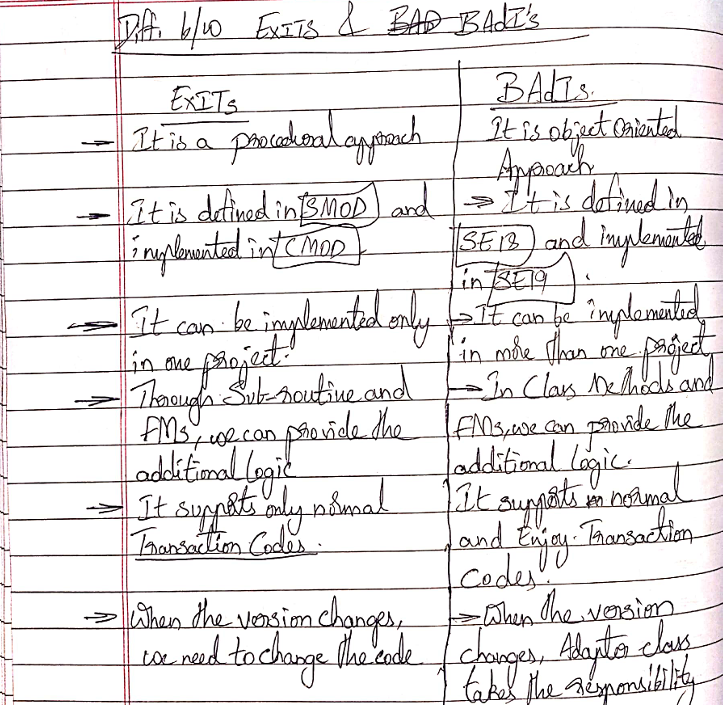
**198. How to identify Customer Exits and where to implement it?**

Ans: You can identify **Customer Exits** using transaction **SMOD** or via the package of a transaction, and implement them by creating a **project in CMOD** and writing code in the corresponding **EXIT\_… function modules**.

**199. What are BAdIs?**



**200. Difference between Exits and BAdIs?**



**201. Difference between Classical and Kernel BAdI?**

| **Point** | **Classical BAdI** | **Kernel BAdI (New BAdI)** |
| --- | --- | --- |
| 1 | Introduced in **SAP NetWeaver 6.40** and older versions. | Introduced in **SAP NetWeaver 7.0 (EhP2)** and higher. |
| 2 | Based on **ABAP Objects** but runs on **ABAP runtime**. | Runs directly on the **Kernel**, hence **faster performance**. |
| 3 | Defined using **SE18**, implemented using **SE19**. | Also defined using **SE18**, but with **new framework** options. |
| 4 | Supports **single or multiple use**, but not filter-dependent by default. | Supports **filter-dependent** and **context-dependent** implementations. |
| 5 | Can be **enhanced** but with more limitations. | Offers **advanced enhancement features** (like fallback, filters). |

**202. How to work with multiple enhancements is BAdIs in single statement?**

Ans: To work with **multiple BAdI implementations** in a **single statement**, use the **GET BADI and CALL BADI** approach inside a **LOOP** or **use the FILTER or MULTIPLE USE feature** depending on the BAdI design.

**203. What are Filters and Sorters in BAdIs?**

Ans: **Filters** in BAdIs allow you to execute specific BAdI implementations based on a filter value (like company code or country), while **Sorters** (used internally or via BADI\_SORTER) help control the **execution order** of multiple BAdI implementations.

**204. What is Fallback class in BAdIs?**

Ans: A **Fallback Class** in BAdIs is a **default implementation** that gets executed **when no active BAdI implementation is found**, ensuring the program still runs with standard logic.

**205. Tell me some of the user exits you worked on?**

* **USEREXIT\_SAVE\_DOCUMENT.**
* **USEREXIT\_DELETE\_DOCUMENT.**
* **USEREXIT\_FIELD\_MODIFICATION.**
* **USEREXIT\_NUMBER\_RANGE.**
* **USEREXIT\_MOVEFILED\_TO\_VBAK(OR)VBAP.**

**206.** **Tell me some of the customer exits you worked on?**

* **SAPMF02D** USER EXIT: CUSTOMER MASTER DATA
* **M06B0003** Number Range and Document number.
* **MGA00003** Material Master (Industrial and Retail): Number Display.

**207.Tell me some of classical BAdIs you worked on?﻿**

* **Sales and Distribution (SD)**

1. **BADI\_SD\_SALES**

➤ Used to enhance sales order processing in VA01/VA02 (e.g., custom validation, pricing logic).

1. **BADI\_LE\_SHP\_DELIVERY\_PROC**

➤ Used in delivery processing (VL01N/VL02N) to influence delivery header/item data.

* **Materials Management (MM)**

1. **ME\_PROCESS\_PO\_CUST**  
   ➤ Enhances Purchase Order processing (ME21N/ME22N) — e.g., defaulting custom fields, validation before save.
2. **BADI\_MATERIAL\_CHECK**  
   ➤ Validates material master data before saving (MM01/MM02).

**208. Tell me some of kernel BAdIs you worked on?**

* **Sales and Distribution (SD)**

1. **BADI\_SD\_DOCUMENTFLOW**  
   ➤ Used to modify or influence how document flow is displayed (e.g., filter certain documents in VA03).
2. **BADI\_SD\_PRICING**  
   ➤ Enhances pricing logic in sales orders — often used to apply special discounts or conditions.

* **Materials Management (MM)**

1. **ME\_PROCESS\_PO\_CUST**   
   ➤ Enhances PO processing — used to fill custom fields or apply validation before PO is saved.
2. **BADI\_MM\_PUR\_S4\_PO**  
   ➤ Used to enhance the Purchase Order Fiori app logic and validations.
3. **BADI\_MM\_INVOICE\_UPDATE**  
   ➤ Enhances the invoice processing (MIRO) — used to validate or enrich invoice data.