SAP ABAP
Interview
Questions & Answers

SAP R/3 ARCHITECTURE

1. What guarantees the integration of all application modules?

The R/3 basis system guarantees the integration of all application modules. The R/3 basis s/w provides the run time environment for the R/3 applications ensures optimal integration, defines a stable architectural frame for system enhancements, and contains the administration tools for the entire system.

One of the main tasks of the basis system is to guarantee the portability of the complete system.

2. What are the central interfaces of the R/3 system?

Presentation Interface.

Database Interface.

Operating system Interface.

3. Which interface controls what is shown on the p.c.?

Presentation Interface.

4. Which interface converts SQL requirements in the SAP development system to those of the database?

Database Interface.

5. What is SAP dispatcher?

SAP dispatcher is the control agent that manages the resources for the R/3 applications.

6. What are the functions of dispatcher?

Equal distribution of transaction load to the work processes.

Management of buffer areas in main memory.

Integration of the presentation levels.

Organization of communication activities.

7. What is a work process?

A work process is where individual dialog steps are actually processed and the work is done. Each work process handles one type of request.

8. Name various work processes of R/3 system?

Dialog or Online (processes only one request at a time).

Background (Started at a specific time)

Update (primary or secondary)

Enque (Lock mechanism).

Spool (generated online or during back ground processing for printing).

9. Explain about the two services that are used to deal with communication.

Message Service: Used by the application servers to exchange short internal messages, all system communications.

Gateway Service: Enables communication between R/3 and external applications using CPI-C protocol.

10. Which work process triggers database changes?

Update work process.

11. Define service (within R/3)?

A service is a process or group of processes that perform a specific system function and often provide an application-programming interface for other processes to call.

12. What are the roll and page areas?

Roll and page areas are SAP R/3 buffers used to store user contexts (process requests). The SAP dispatcher assigns process requests to work processes as they are queued in the roll and page areas.

Paging area holds data from the application programs.

Roll area holds data from previous dialog steps and data that characterize the user.

13. What are the different layers in R/3 system?

Presentation Layer.

Application Layer.

Database Layer.

14. What are the phases of background processing?

Job Scheduling.

Job Processing.

Job Overview.

15. What components of the R/e system initiate the start of background jobs at the specified time?

The batch scheduler initiates the start of background job. The dispatcher then sends this request to an available background work process for processing.

16. Define Instance.

An instance is an administrative unit in which components of an R/3 systems providing one or more services are grouped together. The services offered by an instance are started and stopped at random. All components are parameterized using a joint instance profile. A central R/3 system consists of a single instance in which all-necessary SAP services are offered. Each instance uses separate buffer areas.

17. From hardware perspective, every information system can be divided into three task areas Presentation, Application Logic and Data Storage.

The R/3 Basis software is highly suitable for use in multi-level client/server architectures.

18. What are R/3 Basis configurations?

A central system with centrally installed presentation software.

Two-level client/server system with rolled out presentation software.

Two-level client/server system. Presentation and Application run on the same computer.

Three-level client/server system. Presentation, Application and database each run on separate computers.

19. What is a Service in SAP terminology?

A service refers to something offered by a s/w component.

20. What is Server in SAP terminology?

A component can consist of one process or a group and is then called the server for the respective service.

21. What is a client in SAP terminology?

A S/W component that uses the service (offered by a s/w component) is called a Client. At the same time these clients may also be servers for other services.

22.What is a SAP system?

The union of all s/w components that are assigned to the same databases is called as a SAP system.

23. What is the means of communications between R/3 and external applications?

The means of communication between R/2,R/3 and external applications is via the CPI-C handler or SAP Gateway, using the CPI-C Protocol.

24. What is the protocol used by SAP Gateway process?

The SAP Gateway process communicates with the clients based on the TCP/IP Protocol.

25. Expand CPI-C.

Common Program Interface Communication.

26. What is a Spool request?

Spool requests are generated during dialog or background processing and placed in the spool database with information about the printer and print format. The actual data is places in the Tem Se (Temporary Sequential objects).

27. What are different types of Log records?

V1 and V2. V1 must be processed before V2. But, we can have more than one V2 logs.

28. What are the types of Update requests?

An update request can be divided into one primary (V1) and several Secondary update components (V2). Time-critical operations are placed in V1 component and those whose timing is less critical are placed in V2 components. If a V1 update fails, V2 components will not be processed.

29. Dialog work processes perform only one dialog step and then available for the next request.

30. Explain what is a transaction in SAP terminology.

In SAP terminology, a transaction is series of logically connected dialog steps.

31. Explain how SAP GUI handles output screen for the user.

The SAP front-end s/w can either run on the same computer or on different computers provided for that purpose. User terminal input is accepted by the SAP terminal program SAP GUI, converted to SAP proprietary format and sent to the SAP dispatcher. The dispatcher coordinates the information exchange between the SAP GUIs and the work processes. The dispatcher first places the processing request in request queues, which it then processes. The dispatcher dispatches the requests one after another, to the available work process. The actual processing takes place in the work process. When processing is complete, the result of a work process is returned via the dispatcher to the SAP GUI. The SAP GUI interprets the received data and generates the output screen for the user.

DATA DICTIONARY

1. What are the layers of data description in R/3?

- The external layer.
- The ABAP/4 layer.
- The database layer.

2. Define external layer?

The external layer is the plane at which the user sees and interacts with the data, that is, the data format in the user interface. This data format is independent of the database system used.

3. Define ABAP/4 layer?

The ABAP/4 layer describes the data formats used by the ABAP/4 processor.

4. Define Database layer?

The database layer describes the data formats used in the database.

5. What is a Data Class?

The Data class determines in which table space the table is stored when it is created in the database.

6. What is a Size Category?

The Size category describes the probable space requirement of the table in the database.

7. How many types of size categories and data classes are there?

There are five size categories (0-4) and 11 data classes only three of which are appropriate for application tables:

- APPL0- Master data (data frequently accessed but rarely updated).
- APPL1- Transaction data (data that is changed frequently).
- APPL2- Organizational data (customizing data that is entered when system is configured and then rarely changed).

The other two types are:

- USR
- USR1 – Intended for customer's own developments.

8. What are control tables?

The values specified for the size category and data class are mapped to database-specific values via control tables.

9. What is the function of the transport system and workbench organizer?

The function of the transport system and the Workbench Organizer is to manage any changes made to objects of the ABAP/4 Development Workbench and to transport these changes between different SAP systems.

10. What is a table pool?

A table pool (or pool) is used to combine several logical tables in the ABAP/4 Dictionary. The definition of a pool consists of at least two key fields and a long argument field (VARDATA).

11. What are pooled tables?

These are logical tables, which must be assigned to a table pool when they are defined. Pooled tables can be used to store control data (such as screen sequences or program parameters).

12. What is a table cluster?

A table cluster combines several logical tables in the ABAP/4 Dictionary. Several logical rows from different cluster tables are brought together in a single physical record. The records from the cluster tables assigned to a cluster are thus stored in a single common table in the database.

13. How can we access the correction and transport system?

Each time you create a new object or change an existing object in the ABAP/4 Dictionary, you branch automatically to the Workbench Organizer or correction and transport system.

14. Which objects are independent transport objects?

Domains, Data elements, Tables, Technical settings for tables, Secondary indexes for transparent tables, Structures, Views, Matchcode objects, Matchcode Ids, Lock objects.

15. How is conversion of data types done between ABAP/4 & DB layer?

Conversion between ABAP/4 data types and the database layer is done within the database interface.

16. How is conversion of data types done between ABAP/4 & external level?

Conversion between the external layer and the ABAP/4 layer is done in the SAP dialog manager DYNP.

17. What are the Data types of the external layer?

ACCP, Char, CLNT, CUKY, CURR, DATS, DESC, FLTP, INT1, INT2, INT4, LANG, LCHR, LRAW, NUMC, PREC, QUAN, RAW, TIMS, UNIT,VARC.

18. What are the Data types of the ABAP/4 layer?

Possible ABAP/4 data types:

C: Character.

D: Date, format YYYYMMDD.

F: Floating-point number in DOUBLE PRECISION (8 bytes).

I: Integer.

N: Numerical character string of arbitrary length.

P: Amount of counter field (packed; implementation depends on h/w platform).

S: Time Stamp YYYYMMDDHHMMSS.

V: Character string of variable length, length is given in the first two bytes.

X: Hexadecimal (binary) storage.

19. How can we set the table spaces and extent sizes?

You can specify the extent sizes and the table space (physical storage area in the database) in which a transparent table is to be stored by setting the size category and data class.

20. What is the function of the correction system?

The correction system manages changes to internal system components. Such as objects of the ABAP/4 Dictionary.

21. What are local objects?

Local objects (Dev class$TMP) are independent of correction and transport system.

22. What is a Development class?

Related objects from the ABAP/4 repository are assigned to the same development class. This enables you to correct and transport related objects as a unit.

23. What is a data dictionary?

Data Dictionary is a central source of data in a data management system. Its main function is to support the creation and management of data definitions. It has details about

- What data is contained?
- What are the attributes of the data?
- What is the relationship existing between the various data elements?

24. What functions does a data dictionary perform?

In a data management system, the principal functions performed by the data dictionary are

- Management of data definitions.
- Provision of information for evaluation.
- Support for s/w development.
- Support form documentation.
- Ensuring that the data definitions are flexible and up-to-date.

25. What are the features of ABAP/4 Dictionary?

The most important features are:

- Integrated to aABAP/4 Development Workbench.
- Active in the runtime environment.

26. What are the uses of the information in the Data dictionary?

The following information is directly taken from the Data dictionary:

- Information on fields displayed with F1 help.
- Possible entries for fields displayed with F4 help.
- Matchcode and help views search utilities.

27. What are the basic objects of the data dictionary?

- Tables
- Domains
- Data elements

- Structures
- Foreign Keys

28. What are the aggregate objects in the data dictionary?

- Views
- Match codes
- Lock objects.

29. In the ABAP/4 Dictionary Tables can be defined independent of the underlying database (T/F).

True.

30. ABAP/4 Dictionary contains the Logical definition of the table.

31. A field containing currency amounts (data type CURR) must be assigned to a reference table and a reference field. Explain.

As a reference table, a system containing all the valid currencies is assigned or any other table, which contains a field with the currency key format. This field is called as reference field. The assignment of the field containing currency amounts to the reference field is made at runtime. The value in the reference field determines the currency of the amount.

32. A field containing quantity amounts (data type QUAN) must be assigned to a reference table and a reference field. Explain?

As a reference table, a system table containing all the valid quantity units is assigned or any other table, which contains a field with the format or quantity units (data type UNIT). This field is called as reference field.

The assignment of the field containing quantity amounts to the reference field is made at runtime. The value in the reference field determines the quantity unit of the amount.

33. What is the significance of Technical settings (specified while creating a table in the data dictionary)?

By specifying technical settings we can control how database tables are created in the database. The technical settings allows us to

- Optimize storage space requirements.
- Table access behavior.
- Buffering required.
- Changes to entries logged.

34. What is a Table attribute?

The table's attributes determine who is responsible for maintaining a table and which types of access are allowed for the table. The most important table attributes are:

- Delivery class.
- Table maintenance allowed.
- Activation type.

35. What is the significance of Delivery Class?

- The delivery class controls the degree to which the SAP or the customer is responsible for table maintenance.
- Whether SAP provides the table with or without contents.
- Determines the table type.
- Determines how the table behaves when it is first installed, at upgrade, when it is transported, and when a client copy is performed.

36. What is the max. no. Of structures that can be included in a table or structure.
Nine.

37. What are two methods of modifying SAP standard tables?

- Append Structures and
- Customizing Includes.

38. What is the difference between a Substructure and an Append Structure?

- In case of a substructure, the reference originates in the table itself, in the form of a statement include….
- In case of an append structure, the table itself remains unchanged and the reference originates in the append structure.

39. To how many tables can an append structure be assigned.
One.

40. If a table that is to be extended contains a long field, we cannot use append structures why?
Long fields in a table must always be located in the end, as the last field of the table. If a table has an append structure the append line must also be on the last field of the table.

41. Can we include customizing include or an append structure with Pooled or Cluster tables?
No.

42. What are the two ways for restricting the value range for a domain?

- By specifying fixed values.
- By stipulating a value table.

43. Structures can contain data only during the runtime of a program (T/F)
True.

44. What are the aggregate objects in the Dictionary?

- Views
- Match Code.
- Lock Object.

45. What are base tables of an aggregate object?
The tables making up an aggregate object (primary and secondary) are called aggregate object.

46. The data of a view is not physically stored, but derived from one or more tables (t/f)
True.

47. What are the 2 other types of Views, which are not allowed in Release 3.0?

- Structure Views.
- Entity Views.

48. What is a Match Code?

Match code is a tool to help us to search for data records in the system. Match Codes are an efficient and user-friendly search aid where key of a record is unknown.

49. What are the two levels in defining a Match Code?

- Match Code Object.
- Match Code Id.

50. What is the max no of match code Id's that can be defined for one Match code object?

A match code Id is a one character ID that can be a letter or a number.

51. Can we define our own Match Code ID's for SAP Matchcodes?

Yes, the number 0 to 9 are reserved for us to create our own Match Code Ids for a SAP defined Matchcode object.

52. What is an Update type with reference to a Match code ID?

If the data in one of the base tables of a matchcode ID changes, the matchcode data has to be updated. The update type stipulates when the matchcode is to be updated and how it is to be done. The update type also specifies which method is to be used for Building matchcodes. You must specify the update type when you define a matchcode ID.

53. Can matchcode object contain Ids with different update types?

Yes.

54. What are the update types possible?

The following update types are possible:

- Update type A: The matchcode data is updated asynchronously to database changes.
- Update type S: The matchcode data is updated synchronously to database changes.
- Update type P: The matchcode data is updated by the application program.
- Update type I: Access to the matchcode data is managed using a database view.
- Update type L: Access to the matchcode is achieved by calling a function module.

55. What are the two different ways of building a match code object?

A match code can be built in two different ways:

- Logical structure: The matchcode data is set up temporarily at the moment when the match code is accessed. (Update type I, k).
- Physical Structure: The match code data is physically stored in a separate table in the database. (Update type A, S, P).

56. What are the differences between a Database index and a match code?

- Match code can contain fields from several tables whereas an index can contain fields from only one table.
- Match code objects can be built on transparent tables and pooled and cluster tables.

57. What is the function of a Domain?

- A domain describes the technical settings of a table field.
- A domain defines a value range, which sets the permissible data values for the fields, which refers to this domain.
- A single domain can be used as basis for any number of fields that are identical in structure.

58. Can you delete a domain, which is being used by data elements?

No.

59. What are conversion routines?

- Non-standard conversions from display format to sap internal format and vice-versa are implemented with so called conversion routines.

60. What is the function of a data element?

A data element describes the role played by a domain in a technical context. A data element contains semantic information.

61. Can a domain, assigned to a data element be changed?

Yes. We can do so by just overwriting the entry in the field domain.

62. Can you delete data element, which is being used by table fields.

No.

63. Can you define a field without a data element?

Yes. If you want to specify no data element and therefore no domain for a field, you can enter data type and field length and a short text directly in the table maintenance.

64. What are null values?

If the value of a field in a table is undefined or unknown, it is called a null value.

65. What is the difference between a structure and a table?

Structures are constructed the almost the same way as tables, the only difference using that no database table is generated from them.

66. What is a view?

A view is a logical view on one or more tables. A view on one or more tables i.e., the data from a view is not actually physically stored instead being derived from one or more tables.

67. How many types of Views are there?

- Database View
- Help View
- Projection View
- Maintenance View

68. What is Locking?

When two users simultaneously attempt to access the same data record, this is synchronized by a lock mechanism.

69. What is database utility?

Database utility is the interface between the ABAP/4 Dictionary and the underlying the SAP system.

70. What are the basic functions of Database utility?

The basic functions of database utility are:

- Create database objects.
- Delete database objects.
- Adjust database objects to changed ABAP/4 dictionary definition.

71. What is Repository Info. Systems?

It is a tool with which you can make data stored in the ABAP/4 Dictionary available.

MODULARIZATION

1. Does every ABAP/4 have a modular structure?
Yes.

2. What is Modularization and its benefits?
If the program contains the same or similar blocks of statements or it is required to process the same function several times, we can avoid redundancy by using modularization techniques. By modularizing the ABAP/4 programs we make them easy to read and improve their structure. Modularized programs are also easier to maintain and to update.

3. Name the ABAP/4 Modularization techniques.

- Source code module.
- Subroutines.
- Functions.

4. How can we create callable modules of program code within one ABAP/4 program?

- By defining Macros.
- By creating include programs in the library.

5. M is the attribute type of the module program.

6. Is it possible to pass data to and from include programs explicitly?
No. If it is required to pass data to and from modules it is required to use subroutines or function modules.

7. What are subroutines?
Subroutines are program modules, which can be called from other ABAP/4 programs or within the same program.

8. What are the types of Subroutines?

- Internal Subroutines: The source code of the internal subroutines will be in the same ABAP/4 program as the calling procedure (internal call).
- External Subroutines: The source code of the external subroutines will be in an ABAP/4 program other than the calling procedure.

9. It is not possible to create an ABAP/4 program, which contains only Subroutines (T/F).
False.

10. A subroutine can contain nested form and endform blocks. (T/F)
False.

11. Data can be passed between calling programs and the subroutines using Parameters.

12. What are the different types of parameters?
Formal Parameters: Parameters, which are defined during the definition of subroutine with the FORM statement.

Actual Parameters: Parameters which are specified during the call of a subroutine with the PERFORM statement.

13. **How can one distinguish between different kinds of parameters?**
    - Input parameters are used to pass data to subroutines.
    - Output parameters are used to pass data from subroutines.

14. **What are the different methods of passing data?**
    - Calling by reference: During a subroutine call, only the address of the actual parameter is transferred to the formal parameters. The formal parameter has no memory of its own, and we work with the field of the calling program within the subroutine. If we change the formal parameter, the field contents in the calling program also changes.
    - Calling by value: During a subroutine call, the formal parameters are created as copies of the actual parameters. The formal parameters have memory of their own. Changes to the formal parameters have no effect on the actual parameters.
    - Calling by value and result: During a subroutine call, the formal parameters are created as copies of the actual parameters. The formal parameters have their own memory space. Changes to the formal parameters are copied to the actual parameters at the end of the subroutine.

15. **The method by which internal tables are passed is By Reference.**

16. **How can an internal table with Header line and one without header line be distinguished when passed to a subroutine?**
    Itab[] is used in the form and endform if the internal table is passed with a header line.

17. **What should be declared explicitly in the corresponding ABAP/4 Statements to access internal tables without header lines & why?**
    Work Area. This is required as the Work Area is the interface for transferring data to and from the table.

18. **A subroutine can be terminated unconditionally using EXIT. (T/F)**
True.

19. **A subroutine can be terminated upon a condition using CHECK Statement.**

20. **Function Modules are also external Subroutines. (T/F).**
    True.

21. **What is the difference between the function module and a normal ABAP/4 subroutine?**
    In contrast to normal subroutines function modules have uniquely defined interface. Declaring data as common parts is not possible for function modules. Function modules are stored in a central library.

22. **What is a function group?**
    A function group is a collection of logically related modules that share global data with each other. All the modules in the group are included in the same main program. When an ABAP/4 program contains a CALL FUNCTION statement, the system loads the entire function group in with the program code at runtime. Every function module belongs to a function group.

23. **What is the disadvantage of a call by reference?**
During a call by reference damage or loss of data is not restricted to the subroutine, but will instantly lead to changes to the original data objects.

24. **A function module can be called from a transaction screen outside an ABAP/4 program. (T/F).**
True.

25. **What is an update task?**
It is an SAP provided procedure for updating a database.

26. **What happens if a function module runs in an update task?**
The system performs the module processing asynchronously. Instead of carrying out the call immediately, the system waits until the next database update is triggered with the 'COMMIT WORK' command.

27. **The function modules are created and stored in the <u>Function Library.</u>**

28. **When a function module is activated syntax checking is performed automatically. (Y/N)**
True.

29. **What is the use of the RAISING exception?**
The raising exception determines whether the calling program will handle the exception itself or leave the exception to the system.

30. **What is the difference between internal tables and extract datasets?**
    - The lines of an internal table always have the same structure. By using extract datasets, you can handle groups of data with different structure and get statistical figures from the grouped data.
    - You have to define the structure of the internal table at the beginning. You need not define the structure of the extract dataset.
    - In contrast to internal tables, the system partly compresses exact datasets when storing them. This reduces the storage space required.
    - Internal tables require special work area for interface whereas extract datasets do not need a special work area for interface.

31. **It is possible to assign a local data object defined in a subroutine or function module to a field group. (T/F).**
False.

32. **What is the difference between field-group header and other field groups?**
The header field group is a special field group for the sort criteria. The system automatically prefixes any other field groups with the header field group.

33. **Can a filed occur in several field groups.**
Yes. But it leads to unnecessary data redundancy.

34. **When sorting the extract dataset the fields used as default sort key lie in the <u>Header field group.</u>**

35. **What does the insert statement in extract datasets do?**
    It defines the fields of a field group.

36. **What does the extract statement do in extract datasets?**
    The data is written to virtual memory by extract commands.

37. **A field-groups statement or an insert statement reverses storage space and transfers values. (T/F).**
    False.

38. **While using extract datasets it is required to have a special workarea for interface (T/F)**
    False.

39. **The LOOP-ENDLOOP on extract datasets can be used without any kind of errors (T/F)**
    False. It causes runtime errors.

40. **The Maximum no of key fields that can be used in a header is 50.**

41. **While sorting field groups we cannot use more than one key field (T/F).**
    False.

42. **While sorting, if the main storage available is not enough, the system writes data to an external help file. The SAP profile parameter, which determines this help file, is DIR_SORTTMP.**

43. **The extract statements in field groups can be used before or after processing the sort statements. (T/F)**
    FALSE.

LOGICAL DATABASE

1. **Preparation of the data records by the L.D.B and reading of the data records in the actual report are accomplished with the command pair Put and Get.**

2. **The three main elements of LDB are Structure, Selections, and Database Program.**

3. **In LDB what determines hierarchy of the tables?**
   Structure.

4. **In general what are the two ways in which one can retrieve data from tables?**
   Using Select statements, Database Program.

5. **With LDB one can modify the pre-generated selection screen to their needs (T/F).**
   Yes.

6. **Logical databases are programs that read data from Database tables (Dictionary Structures).**

7. **The event Get<table name> LATE process all tables that are hierarchically superior to the <table name>. (True/False)**
   False. It processes all tables that are hierarchically inferior to the <table name>.

8. The Database Program of LDB is a collection of SUBROUTINES, which selects data and passes it to the report.

9. **The layout of the Database program is determined by both Structure and Selections.**

10. **The order in which data is made available to the report depends on Structure of the LDB.**

11. **Apart from the structure and selections of the LDB the GET statements in the report determines the behavior of the database at runtime.**

12. **Node at the highest level in the structure is known as Root.**

13. **There can be more than one node at the highest level in the structure. (T/F)**
    False. One can define only one node at the highest level in the structure on LDB.

14. **All nodes in the structure of LDB need not be defined in the ABAP/4 Dictionary (T/F).**
    False. One has to define all nodes in the Dictionary or one has to select all nodes that are defined in the Dictionary.

15. **It is not possible to use ABAP/4 Dictionary Structures without an underlying database using LDB. (T/F)**
    True. One can use additionally related tables, along with the tables defined in the structure of LDB.

16. **Dynamic selections allow the user to define further selections for database access in addition to the selection criteria already defined in the LDB selections.**

17. **Check statement can be used along with the event GET in the report for checking the selections, which are not table-specific values.**

18. **In sense of Database Management System (DBMS) LOGICAL DATABASE is a database Structure. (T/F).**
    False.

19. **It is not necessary to maintain the Parent-Child relationship between the tables in Logical Database Structure. (T/F)**
    False. One has to maintain the Parent-Child relationship.

20. **Is it possible to extract data from tables without using the event 'GET' in the report with an appropriate LDB. (T/F).**
    False. One can extract data from tables using Select statements in a report, though the report is having a LDB attribute.

21. **What sorts of tables one can se in designing the hierarchy of a LDB?**
    Tables, which are having Foreign Key relations.

22. **A report program, which uses only SELECT statements, is called SQL Report.**

23. **One cannot use SELECT statements in a report program linked to a Logical Database (T/F).**
    False.

24. **Is it true that the Logical Database reads the data from the database tables using Select Statements (T/F).**
    Yes. We are coding that in Database part of LDB.

25. **In a report with an LDB attribute, you do not have to define how the information should be retrieved from the database tables, but only how the data should be represented on the screen. (T/F).**
    True.

26. **One can use the event GET in a report without LDB attribute. (T/F).**
    False.

27. **The last character of the LDB name denotes Application.**

28. **The structure of Logical Databases reflects the Foreign Key dependencies of hierarchical tables in the SAP system.**

29. **It is mandatory that for each table in the LDB hierarchy there should exist one GET statement in the report. (T/F).**
    False. It is not necessary.

30. **What happens if a report does not contain a GET statement for a particular node of a Logical Database.**
    Process will transfer to the next event.

31. **In a Logical Database, one can define input fields on the selection screen with Select-Options and Parameters statements.**

32. **Suppose a logical database program contains the following lines:**
SELECT-OPTIONS CONNID FOR SPFLI-CONNID.

PARAMETERS CARRID LIKE SFLIGHT-CARRID FOR TABLE SFLIGHT.

What will be the output, for the above two statements?

Only select-options connid for spfli-carrid will be displayed on the screen.

33. **Consider a report with F1S attribute, what will be the output for the following code. Whether you will get the data from spfli and sflight or not, with corresponding tables statement,**
GET SPFLI.

GET SFLIGHT.

Write:/ spfli-carrid, spfli-connid, sflight-fldate, sbook-bookid.

Yes, you will get the data from spfli and sflight.

34. **Consider a report with F1S attribute, what will be the output of the following code. Whether you will get the data from sbook or not, with corresponding tables statement.**
GET SPFLI.

GET SFLIGHT.

Write:/ spfli-carrid, spfli-connid, sflight-fldate, sbook-bookid.

You cannot extract data from sbook.

35. **Identify any errors in the following code and correct the same, and what will be the output. If there exists corresponding tables statement, for spfli, sflight, sbook.**
GET SPFLI.

GET SBOOK.

Write:/ spfli-carrid, spfli-connid, sflight-fldate, sbook-bookid, sbook-class.

No syntax errors. You will get data from all the three tables.

36. **Does the following two statements do the same task? If so which one takes less time and which one is recommended.**
Select * from spfli where spfli-carrid = 'LH' and spfli-connid = '400'.

Endselect.

Select * from spfli. Check: spfli-carrid = 'LH' and spflid-connid = '400'.

Endselect.

Yes they will do the same task. Second Select statement takes less time and is recommended.

37. **If you want to improve the response time (time to access data) Logical Databases permits you to achieve this using VIEWS.**

38. **Is there any upper limit (max) to the possible number of nodes in a logical database structure? If so what is the limit?**
    Yes, there is an upper limit for defining number of nodes in a Logical Database Structure.

    Maximum nodes = 1200 / length where length = max. Length of name in the structure.

39. **In the structure of Logical Database nodes at different levels are displayed in the same columns. (T/F) If false what types of nodes are displayed in the same columns. If true what type of nodes are not displayed in the same columns.**
    False. Nodes at same levels are displayed in the same columns.

40. **What are the advantages of Logical Databases?**
    It offers an easy-to-use selection screens. You can modify the pre-generated selection screen to your needs. It offers check functions to check whether user input is complete, correct, and plausible. It offers reasonable data selections. It contains central authorization checks for data base accesses. Enhancements such as improved performance immediately apply to all report programs that use the logical database.

41. **Though all the ABAP/4 Dictionary Structures that exists in the structure of the LDB, being defined in Database Program, we are defining the Dictionary Structures in the Report. What is the reason for such declaration?**
    By declaring so we are providing work areas for data passing between Logical Database and Report. In addition, the logical database configures the selection screen depending on the selection of database tables.

42. **Is it mandatory to declare all the tables in Report by the key word tables for all the tables that exist in the structure of LDB, and are being defined in the Database part of LDB.**
    No, It is not mandatory to declare all tables in report.

43. **If one wants to access data using Logical Database, the use of events is unavoidable. (T/F)**.
    True.

REPORT GENERATION – FORMATTING

1. **The alignment of a type ‘c’ field in a report is left Aligned.**

2. **In the statement Write:/15(10) Ofal-lifnr. what do the number 15 and 10 stand for**
   15 stand for the offset on the screen and 10 stands for the field length displayed.

3. **Specify the default alignment for the following field types:**
   ‘D’ – Left, ‘F’-Right, ‘N’-Left, ‘I’-Right, ‘T’-Left.

4. **If s_time has the value ‘123456’ how would you get an output of 12:34:56 with a single ‘Write:’ statement.**
   Write: s_time using edit mask’--:--:--‘.

5. **In order to suppress the leading zeroes of a number field the keywords used are NO-ZERO.**

6. **The total no of date formats that can be used to display a date during output is MM/DD/YY, DD/MM/YY, DD/MM/YYYY, MM/DD/YYYY, MMDDYY, DDMMYY, YYMMDD.**

7. **The UNDER Command allows for vertical alignment of fields one below the other.**

8. **In order to concatenate strings only for output purposes the command NO-GAP can be used in conjunction with the ‘Write’ statement.**

9. **The no of decimal places for output can be defines within a write statement. (T/F).**
   TRUE. Write:/<F> decimals 2.

10. **Data can be moved from one field to another using a ‘Write:’ Statement and stored in the desired format. (T/F).**
    TRUE. Write: Date_1 to Date_2 format DD/MM/YY.

11. **In the statement Write:/15(10) lfa1-lifnr. The values 15 and 11 can also be defined by variables (T/F).**
    False.

12. **Differentiate between the following two statements if any.**
    ULINE.

    Write: sy-uline.

    No-difference. Except that uline is used outside the ‘Write’ Statement.

13. **In order to skip a single line the number of lines need not be given as an assignment (T/F)**
    TRUE.

14. **The “SKIP TO LINE line number” is dependent on the LINE-COUNT statement included in the report statement of the program.**

15. **In order to skip columns the command used is POSITION <n>.**

16. **In order to have boldfaced text as output the command used is Write:<f>INTENSIFIED.**

17. **Background and foreground colors can be interchanged using the command Format Inverse.**

18. **In order to restore the system defaults for all changes made with the format statement is Format Reset.**

19. **Like ULINE the statement VLINE is used to insert vertical lines. (T/F).**
    False.

20. **Suppressing the number signs (+/-) is carried out using the addition NO-SIGNS to the Write statement. (T/F).**
    False.

21. **If SY-UZEIT has the value 6:34:45 it can be displayed as 063445 using No Edit Mask.**

22. **If the variable "Text" has the value 'ABCDEF' the output for the statement "Write:/Text+2(3)" will be "CDE"**

23. **The fields specified by select-options and parameters statement cannot be grouped together in the selection screen. (T/F).**
    False.

24. **When calling an external report the parameters or select-options specified in the external report cannot be called. (T/F)**
    FALSE.

25. **Selection Texts in the text elements of the program helps in changing the displayed names of variables in the parameters statement.**

26. **Type F datatype cannot be used to define parameters.**

27. **Rounding off of values can be carried out using the write statement. (T/F).** TRUE

28. **How would you define the exponents for a type 'f' field?**
    Exponent <e>.

29. **How would you format the output as left, centered or right-justified using the write statement.**
    Left-justified, Centered, Right-justified.

30. **If the same formatting options were used for a WRITE statement that follows the FORMAT statement, which settings would take precedence.**
    The settings in the Write Statement.

31. **For each new event, the system resets all formatting options to their default values (T/F)**
    TRUE.

32. **All formatting options have the default value OFF. (T/F).**
    TRUE.

33. **How would you set the formatting options statically and dynamically within a report?**
    Statically: FORMAT <option1>[ON|OFF]….

Dynamically: FORMAT <option1> = <var1><option2>=<var2>….

34. **The page footer is defined using the statement <u>END-OF-PAGE.</u>**

35. **The processing block following END-OF-PAGE is processed only if you reserve lines for the footer in the LINE-COUNT option of the REPORT statement. (T/F)**
TRUE.

36. **To execute a page break under the condition that less than a certain number of lines is left on a page is achieved by <u>RESERVE n lines.</u>**

37. **The RESERVE statement only takes effect if output is written to the subsequent page. No blank pages are created and it defines a block of lines that must be output as a whole. (T/F).** TRUE.

38. **To set the next output line to the first line of a block of lines defined with the RESERVE statement the statement <u>BACK</u> is used.**

39. **What is the limit for the length of a page if the page length is not specified in the report statement. <u>60,000 lines.</u>**

40. **How would you start the printing process from within the program while creating a list?**
NEW-PAGE PRINT ON.

41. **You can change the width of pages within list levels triggered by page breaks. (T/F).**
FALSE.

42. **Hotspots are special areas of an output list used to trigger events. (T/F**) <u>TRUE.</u>

43. **To designate fields as hotspots at runtime, use <u>FORMAT HOTSPOT = <h>.</u>**

44. **Horizontal lines created with ULINE and blank lines created with SKIP can be formatted as hotspots. (T/F).** <u>FALSE.</u>

45. **How would you suppress the display of a parameter on the selection screen?**
Parameters <p> ………..No-Display.

46. **Can you assign a matchcode object to a parameter? If so how?**
Yes. PARAMETERS <p>……..MATCHCODE OBJECT <obj>……..

47. **For each SELECT-OPTIONS statement, the system creates a selection table. (T/F)**
TRUE.

48. **To position a set of parameters or comments on a single line on the selection screen, you must declare the elements in a block enclosed by**
<u>SELECTION-SCREEN BEGIN OF LINE.</u>

……..

<u>SELECTION-SCREEN END OF LINE.</u>

49. **How can Symbols or R/3 icons be output on the screen?**
WRITE <symbol-name>AS SYMBOL.

```
WRITE <icon-name> AS ICON.
```

50. **In the standard setting, you cannot create empty lines with the WRITE statement alone.** (T/F). TRUE.

REPORTING – GENERAL

1. **The system field, which indicates success or failure of a SQL operation, is SY-SUBRC.**

2. **What is the syntax for specifying database table name at runtime in SELECT statement.**

NAME = ‘SPFL1’.

```
SELECT * FROM (NAME).

………………

………………

ENDSELECT.
```

3. **How do you read selected lines of database table into an internal table in packages of predefined size.**
   SELECT * FROM <SPFLI>INTO TABLE <ITAB>PACKAGE SIZE<N>.

   Where n is variable.

4. **Name the WILDCARD characters which are used for comparisons with character strings & numeric strings. ‘%’ and ‘-‘.**

5. **In SELECT statements can you specify a variable in WHERE condition or a part of the condition, if so what is the syntax.**
   SELECT * FROM <table>WHERE <var1><condition><var or const>.

6. **Name the ABAP/4 key words, which are used to change the contents of database table.**

UPDATE or MODIFY.

7. How to specify a client for database table processing.

```
TABLES SPFLI.

SELECT * FROM SPFLI CLIENT SPECIFIED WHERE MANDT BETWEEN ‘001’ AND ‘003’.

……..

ENDSELECT.
```

8. **How do you write a DATA object from ABAP/4 program to ABAP/4 memory and restore the same from memory to program.**
   EXPORT <f1>[FROM <g1>]<f2>[FROM <g2>]…. TO MEMORY ID <key>.

   The ID <key>, which can be up to 32 characters long, identifies the data in memory.

9. **What are DATA CLUSTERS?**
   You can group any complex internal data objects of an ABAP/4 program together in data clusters and store them temporarily in ABAP/4 memory or for longer periods in databases. You can store data clusters in special databases of the ABAP/4 Dictionary. These databases are known as ABAP/4 cluster databases and have a predefined structure. Storing a data cluster is specific to ABAP/4. Although you can also access cluster databases using SQL statements, only ABAP/4 statements are able to decode the structure of the stored data cluster.

10. **Statements used to delete data objects in ABAP/4 memory FREE MEMORY [ID <key>].**

11. **How will you create a file on application server.**
    Open dataset <dsn> for output.

12. **ABAP/4 statement for opening a file on application server for reading Open dataset <dsn> for input.**

13. **How will you transfer data into a file in application server?**
    Data fname(60) value 'mYFILE'.

    Data num type i.

    Open dataset fname for output.

    Do 10 times.

    Num = Num +1.

    Transfer num to fname.

    Enddo.

    …….etc.

14. **Name the function modules to write data from an Internal Table to the Presentation Server.**
    DOWNLOAD and WS_DOWNLOAD.

15. **Name the function module that can be used to give information about files on Presentation Server and about its Operating System.**
    WS_QUERY.

16. **Name the ABAP/4 key word, which is used to clear the Headerline of an Internal Table.**
    CLEAR<itab>.

17. **Name the function modules to read data from Presentation Server into an Internal Table.**
    UPLOAD and WS_UPLOAD.

18. **Name the ABAP/4 keywords to initialize an Internal Table with and without headerline.**
    REFRESH <itab>.

19. **How to determine the attributes of an internal table?**
DESCRIBE TABLE <itab>[LINES <lin>] [OCCURS <occ>].

20. **Name the ABAP/4 key word for searching a string in an Internal Table.**
SEARCH <itab> FOR <str><options>.

The different options (<options>) for the search in an internal table are:

ABBREVIATED

Searches table<itab>for a word containing the character string specified in <str>, where other characters might separate the characters. The first letter of the word and the string <str> must be the same.

STARTING AT<lin1>

Searches table<itab> for <str>, starting at line <line1>. <\lin1> can be a variable.

ENDING AT<n2>

Searches table <itab>for <str>upto line<lin2>. <lin2>can be a variable.

AND MARK

If the search string is found, all the characters in the search string (and all the characters in between when using ABBREVIATED) are converted to upper case.

21. **What are the different attributes that can be assigned to a variant?**
The different attributes that can be assigned to a variant are....

Description

Enter a short, meaningful description of the variant. This may be upto 30 characters long.

Background only

Specify whether you want to use the variant in background processing only, or in online environment as well.

Protected variant.

Mark the field if you want to protect your variant against being changed by other users.

Do not display variant.

Mark this field if you want the variant name to be displayed in the catalog only, but not in the F4 value list.

For the selections you cover in a variant, you can enter the following attributes:

Type

The system displays whether the field is a parameter or a select option.

Protected

Mark this field for each field on the selection screen you want to protect from being overwritten. Values that you mark this way are displayed to the users, but they cannot change them, that are they are not ready to accept input.

Invisible

If you mark this column, the system will not display the corresponding field on the selection screen the user sees when starting the report program.

Variable

Mark this column if you want to set the value for this field at runtime.

22. **Is it possible to create new dynamic programs during runtime of an ABAP/4 program? If so how?**

To create new dynamic programs during the runtime of an ABAP/4 program, you must use an internal table. For this purpose, you should create this internal table with one character type column and a line width of 72. You can use any method you like from Filling Internal Tables to write the code of your new program into the internal table. Especially, you can use internal fields in which contents are dependent on the flow of the program that you use to create a new one, to influence the coding of the new program dynamically. The following example shows how to proceed in principal:

DATA CODE (72) OCCURS 10.

APPEND ‘REPORT ZDYN1.’

TO CODE.

APPEND ‘WRITE/”Hello, I am dynamically created!”.’

TO CODE.

Two lines of a very simple program are written into the internal table CODE.

In the next step you have to put the new module, in the above example it is a report, into the library. For this purpose you can use the following statement:

Syntax

INSERT REPORT <prog>FROM <itab>.

The program <prog> is inserted in your present development class in the R/3 Repository. If a program with this name does not already exists, it is newly created with the following attributes:

Title: none,

Type: 1 (Reporting),

Application: S (Basis).

You can specify the name of the program <prog> explicitly within single quotation marks or you can write the name of a character field, which contains the program name. The name of the program must not necessarily be the same as given in the coding, but it is recommended to do so. <itab> is the internal table containing the source code. For the above example you could write:

INSERT REPORT ‘ZDYN1’ FROM CODE.

Or

DATA REP (8).

REP = ‘ZDYN1’

INSERT REPORT REP FROM CODE.

23. **Data types can be elementary or structured (T/F).**
    TRUE.

24. **The amount of memory associated with a data type is <u>ZERO.</u>**

25. **Data objects are the physical units a program uses at runtime. (T/F).**
    TRUE.

26. **The data object does not occupy any space in memory. (T/F)**
    FALSE.

27. **What are the three hierarchical levels of data types and objects?**
    Program-independent data, defined in the ABAP/4 Dictionary.

    Internal data used globally in one program.

    Data used locally in a procedure (subroutine, function module)

28. **How would you find the attributes of a data type or data object?**
    DESCRIBE FIELD <f> [LENGTH <l.] [TYPE <t> [COMPONENTS <n>]]

    [OUTPUT-LENGTH <o>] [DECIMALS <d>]

    [EDIT MASK <m>].

29. **The components of a field string cannot have different data types. (T/F).**
    FALSE.

30. **Field strings are also called as <u>Record or Structures.</u>**

31. **If a field string is aligned (Left, centered, right justified etc.), the filler fields are also added to the length of the type C field. (T/F).**
    TRUE.

32. **You cannot assign a local data object defined in a subroutine or function module to a field group. (T/F)**
TRUE.

33. **A field group reserves storage space for the fields, and does not contain pointers to existing fields (T/F).**
False.

34. **Defining a field group as 'HEADER' is optional (T/F)**
FALSE.

35. **How would you define a field symbol?**
FIELD-SYMBOLS<FS>.

36. **Which function module would you use to check the user's authorization to access files before opening a file?**
AUTHORITY_CHECK_DATASET

37. **Name the function module used to convert logical file names to physical file names in ABAP/4 programs.**
FILE_GET_NAME.

38. **Parameters, which are defined during the definition of a subroutine with the FORM statement, are called Formal Parameters.**

39. **Parameters which are specified during the call of a subroutine with the PERFORM statement are called Actual Parameters.**

40. **In subroutines internal tables that are passed by TABLES, are always called by value and result. (T/F)**
FALSE. They are called by reference.

INTERACTIVE REPORTING

1. **What is interactive reporting?**
   It helps you to create easy-to-read lists. You can display an overview list first that contains general information and provide the user with the possibility of choosing detailed information that you display on further lists.

2. **What are the uses of interactive reporting?**
   The user can actively control data retrieval and display during the session. Instead of an extensive and detailed list, you create a basic list with condensed information from which the user can switch to detailed displays by positioning the cursor and entering commands. The detailed information appears in secondary lists.

3. **What are the event key words in interactive reporting?**

Event Keyword	Event
AT LINE-SELECTION	Moment at which the user selects a line by double clicking on it or by positioning the cursor on it and pressing F2.
AT USER-COMMAND	Moment at which the user presses a function key.
TOP-OF-PAGE DURING LINE-SELECTION	Moment during list processing of a secondary list at which a new page starts.

4. **What is secondary list?**
   It allows you to enhance the information presented in the basic list. The user can, for example, select a line of the basic list for which he wants to see more detailed information. You display these details on a secondary list. Secondary lists may either overlay the basic list completely or you can display them in an extra window on the screen. The secondary lists can themselves be interactive again.

5. **How to select valid lines for secondary list?**
   To prevent the user from selecting invalid lines, ABAP/4 offers several possibilities. At the end of the processing block END-OF-SELECTION, delete the contents of one or more fields you previously stored for valid lines using the HIDE statement. At the event AT LINE-SELECTION, check whether the work area is initial or whether the HIDE statement stored field contents there. After processing the secondary list, clear the work area again. This prevents the user from trying to create further secondary lists from the secondary list displayed.

6. **How to create user interfaces for lists?**
   The R/3 system automatically, generates a graphical user interface (GUI) for your lists that offers the basic functions for list processing, such as saving or printing the list. If you want to include additional functionality, such as pushbuttons, you must define your own interface status. To create a new status, the Development Workbench offers the Menu Painter. With

the Menu Painter, you can create menus and application toolbars. And you can assign Function Keys to certain functions. At the beginning of the statement block of AT END-OF-SELECTION, active the status of the basic list using the statement: SET PF-STATUS 'STATUS'.

7. **What is interactive reporting?**
   A classical non-interactive report consists of one program that creates a single list. Instead of one extensive and detailed list, with interactive reporting you create basic list from which the user can call detailed information by positioning the cursor and entering commands. Interactive reporting thus reduces information retrieval to the data actually required.

8. **Can we call reports and transactions from interactive reporting lists?**
   Yes. It also allows you to call transactions or other reports from lists. These programs then use values displayed in the list as input values. The user can, for example, call a transaction from within a list of change the database table whose data is displayed in the list.

9. **What are system fields for secondary lists?**

   SY-LSIND Index of the list created during the current event (basic list = 0)

   SY-LIST1 Index of the list level from which the event was triggered.

   SY-LILL1 Absolute number of the line from which the event was triggered.

   SY-LISEL Contents of the line from which the event was triggered.

   SY-CUROW Position of the line in the window from which the event was triggered (counting starts with 1)

   SY-CUCOL Position of the column in the window from which the event was triggered (counting starts with 2).

   SY-CPAGE Page number of the first displayed page of the list from which the event was triggered.

   SY-STARO Number of the first line of the first page displayed of the list from which the event was triggered (counting starts with 1). Possibly, a page header occupies this line.

   SY-STACO Number of the first column displayed in the list from which the event was triggered (counting starts with 1).

   SY-UCOMM Function code that triggered the event.

   SY-PFKEY Status of the displayed list.

10. **How to maintain lists?**
    To return from a high list level to the next-lower level (SY-LSIND), the user chooses Back on a secondary list. The system then releases the currently displayed list and activates the list created one step earlier. The system deletes the contents of the released list. To explicitly specify the list level, into which you want to place output, set the SY-lsind field. The system accepts only index values, which correspond to existing list levels. It then deletes all existing

list levels whose index is greater or equal to the index specify. For example, if you set SY-LSIND to 0, the system deletes all secondary lists and overwrites the basic list with the current secondary list.

11. What are the page headers for secondary lists?

On secondary lists, the system does not display a standard page header and it does not trigger the event. TOP-OF-PAGE. To create page headers for secondary list, you must enhance TOP-OF-PAGE: Syntax TOP-OF-PAGE DURING LINE-SELECTION. The system triggers this event for each secondary list. If you want to create different page headers for different list levels, you must program the processing block of this event accordingly, for example by using system fields such as SY-LSIND or SY-PFKEY in control statements (IF, CASE).

12. How to use messages in lists?

ABAP/4 allows you to react to incorrect or doubtful user input by displaying messages that influence the program flow depending on how serious the error was. Handling messages is mainly a topic of dialog programming. You store and maintain messages in Table T100. Messages are sorted by language, by a two-character ID, and by a three-digit number. You can assign different message types to each message you output. The influence of a message on the program flow depends on the message type. In our program, use the MESSAGE statement to output messages statically or dynamically and to determine the message type.

Syntax:REPORT <rep> MESSAGE-ID <id>.

13. What are the types of messages?

A message can have five different types. These message types have the following effects during list processing:

.A (=Abend):

.E (=Error) or W (=Warning):

.I (=Information):

.S (=Success):

14. What are the user interfaces of interactive lists?

If you want the user to communicate with the system during list display, the list must be interactive. You can define specific interactive possibilities in the status of the list's user interface (GUI). To define the statuses of interfaces in the R/3 system, use the Menu Painter tool. In the Menu Painter, assign function codes to certain interactive functions. After an user action occurs on the completed interface, the ABAP/4 processor checks the function code and, if valid, triggers the corresponding event.

15. What are the drill-down features provided by ABAP/4 in interactive lists?

ABAP/4 provides some interactive events on lists such as AT LINE-SELECTION (double click) or AT USER-COMMAND (pressing a button). You can use these events to move through layers of information about individual items in a list.

16. **What is meant by stacked list?**
    A stacked list is nothing but secondary list and is displayed on a full-size screen unless you have specified its coordinates using the window command.

17. **Is the basic list deleted when the new list is created?**
    No. It is not deleted and you can return back to it using one of the standard navigation functions like clicking on the back button or the cancel button.

18. **What is meant by hotspots?**
    A Hotspot is a list area where the mouse pointer appears as an upright hand symbol. When a user points to that area (and the hand cursor is active), a single click does the same thing as a double-click. Hotspots are supported from R/3 release 3.0c.

19. **What is the length of function code at user-command?**
    Each menu function, push button, or function key has an associated function code of length FOUR (for example, FREE), which is available in the system field SYUCOMM after the user action.

20. **Can we create a gui status in a program from the object browser?**
    Yes. You can create a GUI STATUS in a program using SET PF-STATUS.

21. **In which system field does the name of current gui status is there?**
    The name of the current GUI STATUS is available in the system field SY-PFKEY.

22. **Can we display a list in a pop-up screen other than full-size stacked list?**
    Yes, we can display a list in a pop-up screen using the command WINDOW with the additions starting at X1 Y1 and ending at X2 Y2 to set the upper-left and the lower-right corners where x1 y1 and x2 y2 are the coordinates.

23. **What is meant by hide area?**
    The hide command temporarily stores the contents of the field at the current line in a system-controlled memory called the HIDE AREA. At an interactive event, the contents of the field are restored from the HIDE AREA.

24. **When the get cursor command used in interactive lists?**
    If the hidden information is not sufficient to uniquely identify the selected line, the command GET CURSOR is used. The GET CURSOR command returns the name of the field at the cursor position in a field specified after the addition field, and the value of the selected field in a field specified after value.

25. **How can you display frames (horizontal and vertical lines) in lists?**
    You can display tabular lists with horizontal and vertical lines (FRAMES) using the ULINE command and the system field SY-VLINE. The corners arising at the intersection of horizontal and vertical lines are automatically drawn by the system.

26. **What are the events used for page headers and footers?**
    The events TOP-OF-PAGE and END-OF-PAGE are used for pager headers and footers.

27. How can you access the function code from menu painter?

From within the program, you can use the SY-UCOMM system field to access the function code. You can define individual interfaces for your report and assign them in the report to any list level. If you do not specify self-defined interfaces in the report but use at least one of the three interactive event keywords. AT LINE-SELECTION, AT PF<nn>, OR AT USER-COMMAND in the program, the system automatically uses appropriate predefined standard interfaces. These standard interfaces provide the same functions as the standard list described under the standard list.

28. How the at-user command serves mainly in lists?

The AT USER-COMMAND event serves mainly to handle own function codes. In this case, you should create an individual interface with the Menu Painter and define such function codes.

29. How to pass data from list to report?

ABAP/4 provides three ways of passing data:

---Passing data automatically using system fields

---Using statements in the program to fetch data

---Passing list attributes

30. How can you manipulate the presentation and attributes of interactive lists?

---Scrolling through Interactive Lists.

---Setting the Cursor from within the Program.

---Modifying List Lines.

31. How to call other programs?

	Report	Transaction
Call and return	SUBMIT AND RETURN	CALL TRANSACTION
Call without return	SUBMIT	LEAVE TO TRANSACTION

You can use these statements in any ABAP/4 program.

32. What will exactly the hide statement do?

For displaying the details on secondary lists requires that you have previously stored the contents of the selected line from within the program. To do this, ABAP/4 provides the HIDE statement. This statement stores the current field contents for the current list line. When calling a secondary list from a list line for which the HIDE fields are stored, the system fills the stored values back into the variables in the program. In the program code, insert the HIDE statement directly after the WRITE statement for the current line. Interactive lists provide the user with the so-called '<u>INTERACTIVE REPORTING'</u> facility. For background processing the only possible method of picking the relevant data is through '<u>NON INTERACTIVE REPORT'</u> . After starting a background job, there is no way of influencing the program. But whereas for dialog sessions there are no such restrictions.

33. **How many lists can a program can produce?**

Each program can produce up to 21 lists: one basic list and 20 secondary lists. If the user creates a list on the next level (that is, SY-LSIND increases), the system stores the previous list and displays the new one. Only one list is active, and that is always the most recently created list.

TRANSACTIONS

1. **What is a transaction?**

- A transaction is dialog program that change data objects in a consistant way.

2. **What are the requirements a dialog program must fulfill?**

A dialog program must fulfil the following requirements

- a user friendly user interface.
- Format and consistancey checks for the data entered by the user.
- Easy correction of input errors.
- Access to data by storing it in the data bases.

3.**What are the basic components of dialog program?**

- Screens (Dynpros)
- Each dialog in an SAP system is controlled by dynpros.A dynpros consists of a screen And its flow logic and controls exactly one dialog step.

- ABAP/4 module Pool.

Each dynpro refers to exactly one ABAP/4 dialog program .Such a dialog program is also called a module pool ,since it consists of interactive modules.

4.**What is PBO and PAI events?**

PBO- Process Before Output-It determines the flow logic before displaying the screen.

PAI-Process After Input-It determines the flowlogic after the display of the screen and after receiving inputs from the User.

5. **What is dynpro?What are its components ?**

- A dynpro (Dynamic Program) consists of a screen and its flow logic and controls exactly one dialog steps.
- The different components of the dynpro are :
  Flow Logic: calls of the ABAP/4 modules for a screen .

  Screen layout:Positions of the text ,fields,pushbuttons and so on for a screen

  Screen Attributes:Number of the screen,number of the subsequent screen,and others

  Fields attributes :Definition of the attributes of the individual fields on a screen.

6. **What is a ABAP/4 module pool?**

-Each dynpro refers to exactly one ABAP/4 dialog program.Such a dialog program is also called

a module pool ,since it consists on interactive modules.

7..**Can we use WRITE statements in screen fields?if not how is data transferred from field data to screen fields?**

-We cannot write field data to the screen using the WRITE statement.The system instead transfers data by comparing screen fields names with ABAP/4 variable names.If both names are the same,it

transfers screen fields values to ABAP/4 programs fields and Vice Versa.This happens immediately after displaying the screen.

8.**Can we use flow logic control key words in ABAP/4 and vice-versa?**

- The flow control of a dynpro consists os a few statements that syntactically ressemble ABAP/4 statements .However ,We cannot use flow control keywords in ABAP/4 and vice-versa.

9.**What is GUI status? How to create /Edit GUI status?**

-A GUI status is a subset of the interface elements used for a certain screen.The status comprises

those elements that are currently needed by the transaction .The GUI status for a transaction may be composed of the following elements:

-Title bar.

-Mneu bar.

-Application tool bar

-Push buttons.

To create and edit GUI status and GUI title,we use the Menu Painter.

10. **How does the interection between the Dynpro and the ABAP/4 Modules takes place?**

-A transaction is a collection os screens and ABAP/4 routines ,controlled and executed by a Dialog processor.The Dilaog processor processes screen after the screen ,thereby triggering the appropriate

ABAP/4 processing of each screen .For each screen,the system executes the flow logic that contains the corresponding ABAP/4 processing.The controls passes from screen flow logic to ABAP/4 code and back.

11. **How does the Dialog handle user requests?**

- when an action is performed ,the system triggers the PROCESS AFTER INPUT event.The data passed includes field screen data data entered by the user and a function code. A functioncode is a technical name that has been allocated in a screen Painter or Menu Painter to a meny entry,a push button,the ENTER key or a function Key of a screen.An internal work field(ok-code)in the PAI module evaluates the function code,and the appropriate action is taken.

12. What is to be defined for a push button fields in the screen attributes?

- A function code has to be defined in the screen attributes for the push buttons in a screen.

13. How are the function code handles in Flow Logic?

- When the User selects a function in a transaction ,the system copies the function code into a specially designated work field called OK_CODE.This field is Global in ABAP/4 Module Pool.The OK_CODE can then be evaluated in the corresponding PAI module. The function code is always passed in Exactly the same way , regardless of Whether it comes from a screen's pushbutton,a menu

option ,function key or other GUI element.

14. What controls the screen flow?

- The SET SCREEN and LEAVE SCREEN statements controls screen flow.

15. The Function code currently active is ascertained by what Variable?

- The function code currently active in a Program can be ascertained from the SY-UCOMM Variable.

16. The function code currently active is ascertained by what variable ?

- By SY-UCOMM Variable.

17. What are the "field" and "chain" Statements?

- The FIELD and CHAIN flow logic statements let you Program Your own checks.FIELD and CHAIN tell the system Which fields you are checking and Whether the System should Perform Checks in the flow logic or call an ABAP/4 Module.

18. What is an "on input filed" statements?

- ON INPUT
  The ABAP/4 module is called only if a field contains the Value other than the initial Value.This initial Value is determined by the filed's Dta Type: blanks for character Fields

  ,Zeroes for numerics. If the user changes the Fields Value back t o its initial value,ON INPUT does not trigger a call.

19. What is an "on request Field" statement?

- ON REQUEST

The ABAP/4 Module is called only if the user has entered the value in the field value since the last screen display .The Value counts as changed Even if the User simply types in the value that was already there .In general ,the ON REQUEST condition is triggered through any

Form of" MANUAL INPUT'.

20. What is an on"*-input filed" statement?

ON *-INPUT

- The ABAP/4 module is called if the user has entered the "*" in the first character of the field, and the field has the attribute *-entry in the screen Painter.You can use this option in Exceptionla cases where you want to check only fields with certain Kinds of Input.

21. What are conditional chain statement?

ON CHAIN-INPUT similar to ON INPUT.

The ABAP/4 module is called if any one of the fields in the chain contains a value other than its initial value(blank or nulls).

ON CHAIN-REQUEST

This condition functions just like ON REQUEST, but the ABAP/4 module is called if any one of the fields in the chain changes value.

22. What is "at exit-command:?

The flowlogic Keyword at EXIT-COMMAND is a special addition to the MODULE statement in the Flow Logic .AT EXIT-COMMAND lets you call a module before the system executes the automatic fields checks.

23. Which Function type has to be used for using "at exit-command" ?

- To Use AT EXIT – COMMAND ,We must assign a function Type "E" to the relevant function in the MENU Painter OR Screen Painter .

24. What are the different message types available in the ABAP/4 ?

- There are 5 types of message types available.
- E: ERROR
- W-WARNING
- I –INFORMATION
- A-ABNORMAL TERMINATION.
- S-SUCCESS

25. Of the two " next screen " attributes the attributes that has more priority is -------------------.
Dynamic.

26. Navigation to a subsequent screen can be specified statically/dynamically .(TRUE/FALSE).

TRUE.

27. Dynamic screen sequence for a screen can be set using ------------ and ---------------- commands

Set Screen, Call screen.

27. The commands through Which an ABAP/4 Module can "branch to " or "call" the next screen are

1.------------,2--------------,3---------------,4------------.

- Set screen<scr no>,Call screen<scr no> ,Leave screen, Leave to screen <scr no>.

28**. What is difference between SET SCREEN and CALL SCREEN ?**

- With SET SCREEN the current screen simply specifies the next screen in the chain , control branches to this next screen as sonn as th e current screen has been processed .Return from next screen to current screen is not automatic .It does not interrupt processing of the current screen.If we want to branch to the next screen without finishing the current one ,use LEAVE SCREEN.
- With CALL SCREEN , the current (calling) chain is suspended , and a next screen (screen chain) is called .The called can then return to the suspended chain with the statement LEAVE SCREEN TO SCREEN 0 .Sometime we might want to let an user call a pop up screen from the main application screen to let him enter secondary information.After they have completed their enteries, the users should be able to close the popup and return directly to the place where they left off in the main screen.Here comes CALL SCREEN into picture .This statement lets us insert such a sequence intp the current one.

29. Can we specify the next screen number with a variable (*Yes/No)?

- Yes

30. The field SY-DYNR refers to--------------

Number of the current screen.

31. What is dialog Module?

- A dialog Module is a callable sequence of screens that does not belong to a particular transaction.Dialog modules have their module pools , and can be called by any transaction.

32. The Syntex used to call a screen as dialog box (pop up)is---------

CALL SCREEN <screen number.>

STARTING AT <start column><start line>

ENDING AT <end column> <end line>

33. What is "call mode"?

- In the ABAP/4 WORLD each stackable sequence of screens is a "call mode", This is IMP because of the way u return from the given sequence .To terminate a call mode and return to a suspended chain set the "next screen" to 0 and leave to it:

LEAVE TO SCREEN 0 or (SET SCREEN 0 and LEAVE SCREEN) .When u return to the suspended chain execution resumes with the statement directly following the original CALL SCREEN statement.The original sequence of screens in a transaction (that is , without having stacked any additional call modes),you returned from the transaction altogether.

34. The max number of calling modes stacked at one time is?

- NINE

35. What is LUW or Data base Transaction ?

- A "LUW"(logical unit of work) is the span of time during which any database updates must be performed in an "all or nothing" manner .Either they are all performed (committed),or they are all thrown away (rolled back).In the ABAP/4 world , LUWs and
- Transactions can have several meanings:
  LUW (or "database LUW" or "database transaction")

  This is the set of updates terminated by a database commit. A LUW lasts ,at most , from one screen change to the next (because the SAP system triggers database commits automatically at every screen change).

36. What is SAP LUW or Update Transaction?

Update transaction (or "SAP LUW")

This is a set of updates terminated by an ABAP/4 commit. A SAP LUW may last much longer than a database LUW, since most update processing extends over multiple transaction screens. The programmer terminates an update transaction by issuing a COMMIT WORK statement.

37. What happens if only one of the commands SET SCREEN and LEAVE SCREEN is used without using the other?

If we use SET SCREEN without LEAVE SCREEN, the program finishes processing for the current screen before branching to <scr no>. If we use LEAVE SCREEN without a SET SCREEN before it, the current screen process will be terminated and branch directly to the screen specified as the default next-screen in the screen attributes.

38. What is the significance of the screen number '0'?

In "calling mode", the special screen number 0 (LEAVE TO SCREEN 0) causes the system to jump back to the previous call level. That is, if you have called a screen sequence with CALL SCREEN leaving to screen 0 terminates the sequence and returns to the calling screen. If you have not called a screen sequence, LEAVE TO SCREEN 0 terminates the transaction.

39. What does the 'SUPPRESS DIALOG' do?

Suppressing of entire screens is possible with this command. This command allows us to perform screen processing "in the background". Suppresing screens is useful when we are branching to list-mode from a transaction dialog step.

40. What is the significance of the memory table 'SCREEN'?

At runtime, attributes for each screen field are stored in the memory table called 'SCREEN'. We need not declare this table in our program. The system maintains the table for us internally and updates it with every screen change.

41. What are the fields in the memory table 'SCREEN'?

Name	Length	Description

NAME	30	Name of the screen field
GROUP1	3	Field belongs to field group 1
GROUP2	3	Field belongs to field group 2
GROUP3	3	Field belongs to field group 3
GROUP4	3	Field belongs to field group4
ACTIVE	1	Field is visible and ready for input.
REQUIRED	1	Field input is mandatory.
INPUT	1	Field is ready for input.
OUTPUT	1	Field is display only.
INTENSIFIED	1	Field is highlighted
INVISIBLE	1	Field is suppressed.
LENGTH	1	Field output length is reduced.
DISPLAY_3D	1	Field is displayed with 3D frames.
VALUE_HELP	1	Field is displayed with value help.

42. Why grouping of fields is required? What is the max no of modification groups for each field?

If the same attribute need to be changed for several fields at the same time these fields can be grouped together. We can specify up to four modification groups for each field.

43. What are the attributes of a field that can be activated or deactivated during runtime?

Input, Output, Mandatory, Active, Highlighted, Invisible.

44. What is a screen group? How it is useful?

Screen group is a field in the Screen Attributes of a screen. Here we can define a string of up to four characters which is available at the screen runtime in the SY-DNGR field. Rather than maintaining field selection separately for each screen of a program, we can combine logically associated screens together in a screen group.

45. What is a Subscreen? How can we use a Subscreen?

A subscreen is an independent screen that is displayed in a n area of another ("main") screen. To use a subscreen we must call it in the flow logic (both PBO and PAI) of the main screen. The CALL SUBSCREEN stratement tells the system to execute the PBO and PAI events for the subscreen as part of the PBO or PAI events of the main screen. The flow logic of your main program should look as follows:

PROCESS BEFORE OUTPUT.

CALL SUBSCREEN <area> INCLUDING '<program>' '<screen>'.

PROCESS AFTER INPUT.

CALL SUBSCREEN <area>.

Area is the name of the subscreen area you defined in your main screen. This name can have up to ten characters. Program is the name of the program to which the subscreen belongs and screen is the subscreen's number.

46. What are the restrictions on Subscreens?

Subscreens have several restrictions. They cannot:

- Set their own GUI status
- Have a named OK code
- Call another screen
- Contain an AT EXIT-COMMAND module
- Support positioning of the cursor.

47. How can we use / display table in a screen?

ABAP/4 offers two mechanisms for displaying and using table data in a screen. These mechanisms are TABLE CONTROLS and STEP LOOPS.

48. What are the differences between TABLE CONTROLS and STEP LOOPS?

TABLE CONTROLS are simply enhanced STEP LOOPS that display with the look and feel of a table widget in a desktop application. But from a programming standpoint, TABLE CONTROLS and STEP LOOPS are almost exactly the same. One major difference between STEP LOOPS and TABLE CONTROLS is in STEP LOOPS their table rows can span more than one time on the screen. By contrast the rows in a TABLE CONTROLS are always single lines, but can be very long. (Table control rows are scrollable). The structure of table control is different from step loops. A step loop, as a screen object, is simply a series of field rows that appear as a repeating block. A table control, as a screen object consists of: I) table fields (displayed in the screen) ii) a control structure that governs the table display and what the user can do with it.

49. What are the dynapro keywords?

FIELD, MODULE, SELECT, VALUES and CHAIN are the dynapro keywords.

50. Why do we need to code a LOOP statement in both the PBO and PAI events for each table in the screen?

We need to code a LOOP statement in both PBO and PAI events for each table in the screen. This is because the LOOP statement causes the screen fields to be copied back and forth between the ABAP/4 program and the screen field. For this reason, at least an empty LOOP….ENDLOOP must be there.

51. The field SY-STEPL refers to the index of the screen table row that is currently being processed. The system variable SY-stepl only has a meaning within the confines of LOOP….ENDLOOP processing. Outside the loop, it has no valid value.

52. How can we declare a table control in the ABAP/4 program?

Using the syntax controls <table control name> type tableview using screen <scr no>.

53. **Differentiate between static and dynamic step loops.**
Step loops fall into two classes: Static and Dynamic. Static step loops have a fixed size that cannot be changed at runtime. Dynamic step loops are variable in size. If the user re-sizes the window the system automatically increases or decreases the number of step loop blocks displayed. In any given screen you can define any number of static step loops but only a single dynamic one.

54. **What are the two ways of producing a list within a transaction?**
By submitting a separate report.

By using leave to list-processing.

55. **What is the use of the statement Leave to List-processing?**
Leave to List-processing statement is used to produce a list from a module pool. Leave to list processing statement allows to switch from dialog-mode to list-mode within a dialog program.

56. **When will the current screen processing terminates?**
A current screen processing terminates when control reaches either a Leave-screen or the end of PAI.

57. **How is the command Suppress-Dialog useful?**
Suppressing entire screens is possible using this command. This command allows us to perform screen processing "in the background". The system carries out all PBO and PAI logic, but does not display the screen to the user. Suppressing screens is useful when we are branching to list-mode from a transaction dialog step.

58. **What happens if we use Leave to list-processing without using Suppress-Dialog?**
If we don't use Suppress-Dialog to next screen will be displayed but as empty, when the user presses ENTER, the standard list output is displayed.

59. **How the transaction that are programmed by the user can be protected?**
By implementing an authority check.

60. **What are the modes in which any update tasks work?**
Synchronous and Asynchronous.

61. **What is the difference between Synchronous and Asynchronous updates?**
A program asks the system to perform a certain task, and then either waits or doesn't wait for the task to finish. In synchronous processing, the program waits: control returns to the program only when the task has been completed. In asynchronous processing, the program does not wait: the system returns control after merely logging the request for execution.

62. **SAP system configuration incluedes Dialog tasks and Update tasks.**

63. **Dialog-task updates are Synchronous updates.**

64. Update –task updates are Asynchronous updates.

65. What is the difference between Commit-work and Rollback-Work tasks?

Commit-Work statement “performs” many functions relevant to synchronized execution of tasks. Rollback-work statement “cancels: all reuests relevant to synchronized execution of tasks.

66. What are the different database integrities?

- Semantic Integrity.
- Relational Integrity.
- Primary Key Integrity.
- Value Set Integrity.
- Foreign Key integrity and
- Operational integrity.

67. All SAP Databases are Relational Databases.

68. What is SAP locking?

It is a mechanism for defining and applying logical locks to database objects.

69. What does a lock object involve?

The tables.

The lock argument.

70. What are the different kinds of lock modes?

Shared lock

Exclusive lock.

Extended exclusive list.

71. How can a lock object be called in the transaction?

By calling Enqueue<lock object> and Dequeue<lock object> in the transaction.

72. What are the events by which we can program “help texts” and display “possible value lists”?

-PROCESS ON HELP-REQUEST (POH).

-PROCESS ON VALUE-REQUEST (POV).

73. What is a matchcode?

A matchcode is an aid to finding records stored in the system whenever an object key is required in an input field but the user only knows other (non-key) information about the object.

74. In what ways we can get the context sensitive F1 help on a field?

- Data element documentation.
- Data element additional text in screen painter.
- Using the process on help request event.

75. **What is roll area?**

A roll area contains the program's runtime context. In addition to the runtime stack and other structures, all local variables and any data known to the program are stored here.

76. **How does the system handle roll areas for external program components?**

- Transactions run in their own roll areas.
- Reports run in their own roll areas.
- Dialog modules run in their own roll areas
- Function modules run in the roll area of their callers.

77. **Does the external program run in the same SAP LUW as the caller, or in a separate one?**

- Transactions run with a separate SAP LUW
- Reports run with a separate SAP LUW.
- Dialog modules run in the same SAP LUW as the caller
- Function modules run in the same SAP LUW as the caller.

The only exceptions to the above rules are function modules called with IN UPDATE TASK (V2 function only) or IN BACKGROUND TASK (ALE applications). These always run in their own (separate) update transactions.

78. **What are function modules?**

Function modules are general-purpose library routines that are available system-wide.

79. **What are the types of parameters in the function modules?**

In general, function module can have four types of parameters:

- EXPORTING: for passing data to the called function.
- IMPORTING: for receiving data returned from the function module.
- TABLES: for passing internal tables only, by reference (that is, by address).
- CHANGING: for passing parameters to and from the function.

80. **What is the difference between Leave Transaction and Call Transaction?**

In contrast to LEAVE TO TRANSACTION, the CALL TRANSACTION statement causes the system to start a new SAP LUW. This second SAP LUW runs parallel to the SAP LUW for the calling transaction.

81. **How can we pass selection and parameter data to a report?**

There are three options for passing selection and parameter data to the report.

- Using SUBMIT…WITH
- Using a report variant.
- Using a range table.

82. **How to send a report to the printer instead of displaying it on the screen?**

We can send a report to the printer instead of diplaying it on the screen. To do this, use the keywords TO SAP-SPOOL:

SUBMIT RSFLFIND…TO SAP-SPOOL DESTINATION 'LT50'.

83. **How can we send data to external programs?**

Using SPA/GPA parameters(SAP memory).

Using EXPORT/IMPORT data (ABAP/4 memory)

84. **What are SPA/GPA parameters (SAP memory)**

SPA/GPA parameters are field values saved globally in memory. There are two ways to use SPA/GPA parmeters:

By setting field attributes in the Screen Painter.

By using the SET PARAMETER or GET PARAMETER statements.

BDC

1. **What is full form of BDC Session?**
   Batch Data Communication Session.

2. **What are the steps in a BDC session?**
   The first step in a BDC session is to identify the screens of the transaction that the program will process. Next step is to write a program to build the BDC table that will be used to submit the data to SAP. The final step is to submit the BDC table to the system in the batch mode or as a single transaction by the CALL TRANSACTION command.

3. **How do you find the information on the current screen?**
   The information on the current screen can be found by SYSTEM → STATUS command from any menu.

4. **How do you save data in BDC tables?**
   The data in BDC tables is saved by using the field name ‘BDC_OKCODE’ and field value of ‘/11’.

5. **What is the last entry in all BDC tables?**
   In all BDC tables the last entry is to save the data by using the field name BDC_OKCODE and a field value of ‘/11’.

6. **What is a multiple line field?**
   A multiple line field is a special kind of field which allows the user to enter multiple lines of data into it.

7. **How do you populate data into a multiple line field?**
   To populate data into a multiple line field, an index is added to the field name to indicate which line is to be populated by the BDC session (Line index).

8. **Write the BDC table structure.**
   BDC table structure

FIELD	TYPE	DESCRIPTION
Program	CHAR (8)	Program name of transaction.
DynPro	CHAR (4)	Screen number of transaction.
DynBegin	CHAR (1)	Indicator for new screen.
Fnam	CHAR (35)	Name of database field from screen.
Fval	CHAR (80)	Value to submit to field.

9. **Does the CALL TRANSACTION method allow multiple transactions to be processed by SAP?**
   No. The CALL TRANSACTION method allows only a single transaction to be processed by SAP.

10. **Does the BDC-INSERT function allow multiple transactions to be processed by SAP?**
    Yes.

11. **What is the syntax for ‘CALL TRANSACTION’?**
    CALL TRANSACTION trans [using bdctab MODE mode].

Three possible entries are there for MODE.

A	-	Show all screens.
E	-	Show only screens with errors.
N	-	Show no screens.