Database Management Systems
Part A (Two mark questions)

1. **List the purpose of Database System (or) List the drawback of normal File Processing System.**

   Problems with File Processing System:
   1. **Data redundancy and inconsistency**
   2. **Difficulty in accessing data**
   3. **Difficulty in data isolation**
   4. **Integrity problems**
   5. **Atomicity problems**
   6. **Concurrent-access anomalies**
   7. **Security problems**

   We can solve the above problems using Database System.

2. **Define Data Abstraction and list the levels of Data Abstraction.**

A major purpose of a database system is to provide users with an abstract view of the data. That is, the system hides certain details of how the data are stored and maintained. Since many database systems users are not computer trained, developers hide the complexity from users through several levels of abstraction, to simplify users' interaction with the System: Physical level, Logical Level, View Level.

3. **Define DBMS.**

A Database-management system consists of a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information about one particular enterprise. The primary goal of a DBMS is to provide an environment that is both convenient and efficient to use in retrieving and storing database information.

4. **Define Data Independence.**

The ability to modify a schema definition in one level without affecting a schema definition in the next higher level is called data independence. There are two levels of data independence: Physical data independence, and Logical data independence.

5. **Define Data Models and list the types of Data Model.**

Underlying the structure of a database is the data model: a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints. The various data models that have been proposed fall into three different groups: object-based logical models, record-based logical models, and physical models.

6. **Discuss about Object-Based Logical Models.**

Object-based logical models are used in describing data at the logical and view levels. They provide fairly flexible structuring capabilities and allow data constraints to be specified explicitly. There are many different models: entity-relationship model, object-oriented model, semantic data model, and functional data model.

7. **Define E-R model.**

The entity-relationship data modal is based on perception of a real world that consists of a collection of basic objects, called entities, and of relationships among these objects. The overall logical structure of a database can be expressed graphically by an E-R diagram, which is built up from the following components:

- **Rectangles, which represent entity sets.**
- **Ellipses, which represent attributes**
- **Diamonds, which represent relationships among entity sets**
- **Lines, which link attributes to entity sets and entity sets to relationships.**

E.g.)

8. **Define entity and entity set.**

An entity is a "thing" or "object" in the real world that is distinguishable from other objects. For example, each person is an entity, and bank accounts can be considered to be entities. The set of all entities of the same type are termed an entity set.

Refer example in question no: 7

9. **Define relationship and relationship set.**

A relationship is an association among several entities. For example, a Depositor relationship associates a customer with each account that she has. The set of all relationships of the same type, are termed a relationship set.

Refer example in question no: 7

10. **Define Object-Oriented Model.**

The object-oriented model is based on a collection of objects. An object contains values stored in instance variables within the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods. Objects that contain the same types of values and the same methods are grouped together into classes. The only way in which one object can access the data of another object is by invoking a method of that other object. This action is called sending a message to the object.

11. **Define Record-Based Logical Models.**

Record-based logical models are used in describing data at the logical and view levels. They are used both to specify the overall structure of the database and to provide a higher-level description of the implementation. Record-based models are so named because the database is structured in fixed-format records of several types. Each record type defines a fixed number of fields, or attributes, and each field is usually of fixed length. The three most widely accepted record-based data models are the relational, network, and hierarchical models.

12. **Define Relational Model.**

The relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name.

13. **Define Network Model.**

Data in the network model are represented by collections of records, and relationships among data are represented by links, which can be viewed as pointers. The records in the database are organized as collections of arbitrary graphs.

14. **Define Hierarchical Model.**

The hierarchical model is similar to the network model in the sense that data and relationships among data are represented by records and links, respectively. It differs from the network model in that the records are organized as collection of trees rather than arbitrary graphs.

15. **Define DDL.**

A database schema is specified by a set of definitions expressed by a special language called a data-definition language. The result of compilation of DDL statements is a set of tables that is stored in a special file called data dictionary. A data dictionary is a file that contains metadata-that is, data about data. The storage structure and access methods used by the database system are specified by a set of definitions in a special type of DDL called a data storage and definition language.

16. **Define DML.**

By data manipulation, we mean

- **The retrieval of information stored in the database.**
- **The insertion of new information into the database**
- **The deletion of information from the database**
- **The modification of information stored in the database.**

A DML is a language that enables users to access or manipulate data as organized by the appropriate data model. There are two types: Procedural DMLs and Nonprocedural DMLs.

17. **Define Query and Query language.**

A query is a statement requesting the retrieval of information. The portion of a DML that involves information retrieval is called query language.

18. **List the role of DBA.**

The person who has central control over the system is called the database administrator. The functions of the DBA include the following:

- **Schema definition**
- **Storage structure and access-method definition**
- **Schema and physical-organization modification**
- **Granting of authorization for data access**
- **Integrity-constraint specification**

19. **List the different types of database-system users.**

There are four different types pf database-system users, differentiated by the way that they expect to interact with the system.

- **Application programmers**
- **Sophisticated Users**
- **Specialized users**
- **Naïve users.**

20. **Write about the role of Transaction manager.**

TM is responsible for ensuring that the database remains in a consistent state despite system failures. The TM also ensures that concurrent transaction executions proceed without conflicting.

21. **Write about the role of Storage manager.**

A SM is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system. The SM is responsible for interaction with the data stored on disk.

22. **Define Attributes.**

Entities are described in a database by a set of attributes. For example, the attributes account-number and balance describe one particular account in a bank.

An attribute, as used in the E-R model, can be characterized by the following attribute types.

- **Simple and composite attributes**
- **Single-valued and multi valued attributes**

- **Null attributes**
- **Derived attributes.**

23. **Define Mapping Constraints.**

An E-R enterprise schema may define certain constraints to which the contents of a database must conform. Two of the most important types of constraints are
Mapping Cardinalities: express the number of entities to which another entity can be associated via a relationship set.
Existence Dependencies: If the existence of entity x depends on the existence y, then x is said to be existence dependent on y.

24. **Define Super key.**

A super key is a set of one or more attributes that, taken collectively, allow us to identify uniquely an entity in the entity set. For example, the social-security attribute of the entity set customer is sufficient to distinguish one customer entity from another. Similarly, the combination of customer-name and social security is a super key for the entity set customer.

25. Define Primary key.

Superkeys for which no proper subset is a super key. Such minimal superkeys are called candidate keys or primary keys. For example, the social-security attribute of the entity set customer is sufficient to distinguish one customer entity from another.

25. **Define Weak Entity Sets.**

An entity set may not have sufficient attributes to form a primary key. Such an entity set is termed a Weak Entity Set. As an illustration, consider the entity set payment, which has the three attributes: payment-number, payment-date, and payment-amount. Although each payment entity is distinct, payments for different loans may share the same payment number. Thus, this entity set does not have a primary key; it is a weak entity set.

26. **Define Strong Entity Set.**

An entity set that has a primary key is termed Strong Entity set. E.g.) Customer entity set.

27. **Define Relational Algebra.**

A general expression in the relational algebra is constructed out of smaller sub expressions. Let E1 and E2 be relational algebra expressions. Then, the following are all relational algebra expressions:

- **E1 U E2**
- **E1 – E2**
- **E1 * E2**
- **(E1), where P is a predicate on attribute in E1.**
- **(E1), where S is a list consisting of some of the attributes in E1**
- **(E1), where x is the new name for the result of E1.**

28. **List the possible operations is Relational Algebra.**

- **Select operation**
- **Project operation**
- **Union operation**
- **Set Difference operation**
- **Cartesian Product operation**
- **Rename operation**
- **Set-Intersection operation**
- **Natural-join operation**

- **Division**
- **Assignment operation**

29. **Define Aggregate Functions.**

Aggregate functions are functions that take a collection of values as input and return a single value. SQL offers five built-in aggregate functions:

- **Average: avg**
- **Minimum: min**
- **Maximum: max**
- **Total: sum**
- **Count: count**

30. **Define Null Values.**

SQL allows the use of null values to indicate absence of information about the value of an attribute.

31. **Define Nested Sub queries.**

SQL provides a mechanism for the nesting of sub queries. A sub query is a select-from-where expression that is nested within another query. A common use of sub queries is to perform tests for set membership, set comparisons, and set cardinality.

32. **Define Embedded SQL.**

The SQL standard defines embeddings of SQL in a variety of programming languages, such as Pascal, PL/I, Fortran, C, and COBOL. A language in which SQL queries are embedded is referred to as a host language, and the SQL structures permitted in the host language constitute embedded SQL.

33. **Define Integrity Constraints.**

Integrity constraints provide a means of ensuring that changes made to the database by authorized users do not result in a loss of data consistency. Thus Integrity Constraints guard against accidental damage to the database. The constraints were in the following forms: Key declarations, and Form of a relationship.

34. **Define Referential Integrity.**

Often, we wish to ensure that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation. This condition is called referential integrity.

35. **Define Assertions.**

An assertion is a predicate expressing a condition that we wish the database always satisfied. E.g.) create assertion <assertion-name> check <predicate>

36. **Define Triggers.**

A trigger is a statement that is executed automatically by the system as a side effect of a modification to the database. To design a trigger mechanism, we must meet two requirements:

1. **Specify the conditions under which the trigger is to be executed.**
2. **Specify the actions to be taken when the trigger executes.**

37. **Define Functional Dependency.**

Functional dependencies are constraints on the set of legal relations. They allow us to express facts about the enterprise that we are modeling with our database. Syntax: A -> B e.g.) account no -> balance for account table.

38. **List the pitfalls in Relational Database Design.**

1. **Repetition of information**
2. **Inability to represent certain information**

39. **Define normalization.**

By decomposition technique we can avoid the Pitfalls in Relational Database Design. This process is termed as normalization.

40. **List the properties of decomposition.**

1. **Lossless join**
2. **Dependency Preservation**
3. **No repetition of information**

41. **Define First Normal Form.**

If the Relation R contains only the atomic fields then that Relation R is in first normal form.

E.g.) R = (account no, balance) – first normal form.

42. **Define Second Normal Form.**

A relation schema R is in 2 NF with respect to a set F of FD's if for all FD's of the form A -> B, where A is contained in R and B is contained in R, and A is a superkey for schema R.

43. **Define BCNF.**

A relation schema R is in BCNF with respect to a set F of FD's if for all FD's of the form A -> B, where A is contained in R and B is contained in R, at least one of the following holds:

1. **A -> B is a trivial FD**
2. **A is a superkey for schema R.**

44. **Define 3 Normal Form.**

A relation schema R is in 3 NF with respect to a set F of FD's if for all FD's of the form A -> B, where A is contained in R and B is contained in R, at least one of the following holds:

1. **A -> B is a trivial FD**
2. **A is a superkey for schema R.**
3. **Each attribute in B – A is contained in a candidate key for R.**

45. **Define Fourth Normal Form.**

A relation schema R is in 4NF with respect to a set F of FD's if for all FD's of the form A ->> B (Multi valued Dependency), where A is contained in R and B is contained in R, at least one of the following holds:

1. **A ->> B is a trivial MD**
2. **A is a superkey for schema R.**

3. **Define 5NF or Join Dependencies.**

Let R be a relation schema and R1, R2, …, Rn be a decomposition of R. The join dependency *(R1, R2, …Rn) is used to restrict the set of legal relations to those for which R1,R2,…Rn is a lossless-join decomposition of R. Formally, if R= R1 U R2U …U Rn, we say that a relation r® satisfies the join dependency *(R1, R2, …Rn) if
R =
A join dependency is trivial if one of the Ri is R itself.

47. Define Cache?

The cache is the fastest and most costly form of storage. Cache memory is small; its use is managed by the operating system.

48. Explain Optical Storage Device?

The most popular form of optical storage is the compact disk read-only memory, can be read by a laser. Optical storage is the write-once, read-many disk, which allows data to be written once, but does not allow them to be erased and rewritten.

49. Define disk controller?

It is an interface between the computer system and the actual hardware of the disk drive. Accept high-level command to read or write a sector. It attaches checksums to each sector that is written. It also performs remapping of bad sectors.

50. Define RAID.

It is collectively called redundant arrays of inexpensive disk, have been proposed to address the performance and reliability issues. Raids are used for their higher reliability and higher data transfer rate. RAID stands for independent, instead of inexpensive.

51. Define file organization

A file is organized logically as a sequence of records. These records are mapped onto disk blocks. Files are provided as a basic construct in operating system.

52. Define Hash indices?

Indices are based on the values being distributed uniformly across a range of buckets. The bucket to which a value is assigned is determined by a function, called a hash function.

53. Define dense index?

An index record appears for every search-key value in the file. The index record contains the search-key value and pointer to the first data record with that search-key value.

54. Define sparse index?

An index record is created for only some of the values. Each index record contains a search-key value and a pointer to the first data record with that search-key value. To locate a record we find the index entry with the largest search-key value that is less than or equal to the search-key value.

55. Explain B+ -tree index structure?

The B+ -tree index structure is the most widely used of several index structures that maintain their efficiency despite insertion and deletion of data. A B+ -tree index takes the form of a balanced tree in which every path from the root of the tree to a leaf of
The tree is the same length.

56. Define Static Hashing?

File organization based on the technique of hashing allow us to avoid accessing an index structure. Hashing also provides a way of constructing indices.

57. Define Query processing?

Query processing refers to the range of activities involved in extracting data form a database. These activities include translation of queries expressed in high-level database language into expression that can be implemented at the physical level of the file system.

58. Define Merge-join?

The merge-join algorithm can be used to compute natural joins and equi-joins.

59. Explain Hybrid Hash-join?

The hybrid hash-join algorithm performs another optimization; it is useful when memory size is relatively large, but not all the build relation fits in memory. The partitioning phase of the hash-join algorithm needs one block of memory as a buffer for each partition that is created, and one block of memory as an input buffer.

60. Define hash-table overflow?

Hash-table overflow occurs in partition i of the build relation s if the hash index on H is larger than main memory. Hash-table overflow can occur if there are many tuples in the build relation with the same values for the join attributes.

61. **What is transaction?**

A transaction is a unit of program execution that accesses and possibly updates various data items. A transaction usually results from the execution of a user program written in a high-level data-manipulation language or programming language, and is delimited by statements of the form begin transaction and end

transaction. The transaction consists of all operations executed between the begin and end of the transaction.

62. **List the properties of transaction.**

    1. **Atomicity**
    2. **Consistency**
    3. **Isolation**
    4. **Durability**

63. **List the possible transaction states.**

    1. **Active**
    2. **Partially committed**
    3. **Aborted**
    4. **Committed**