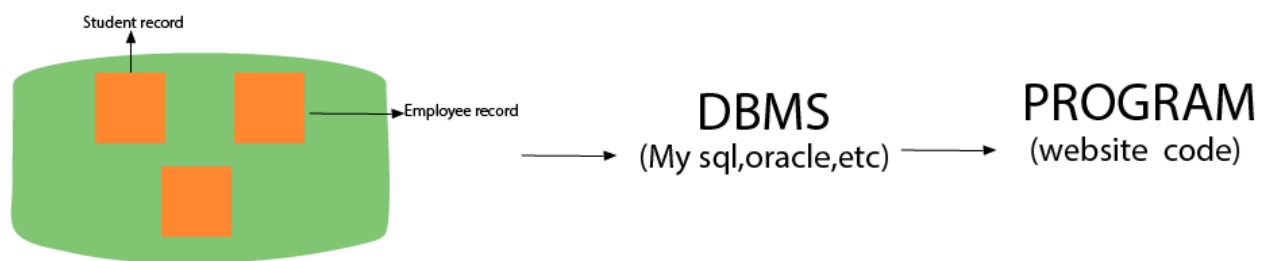


(41 questions)

1) What is DBMS?

DBMS is a collection of programs that facilitates users to create and maintain a database. In other words, DBMS provides us an interface or tool for performing different operations such as the creation of a database, inserting data into it, deleting data from it, updating the data, etc. DBMS is a software in which data is stored in a more secure way as compared to the file-based system. Using DBMS, we can overcome many problems such as- data redundancy, data inconsistency, easy access, more organized and understandable, and so on. There is the name of some popular Database Management System- MySQL, Oracle, SQL Server, Amazon simple DB (Cloud-based), etc.

Working of DBMS is defined in the figure below.



2) What is a database?

A Database is a logical, consistent and organized collection of data that it can easily be accessed, managed and updated.

3) What is a database system?

The collection of database and DBMS software together is known as a database system. Through the database system, we can perform many activities such as-

The data can be stored in the database with ease, and there are no issues of data redundancy and data inconsistency.

The data will be extracted from the database using DBMS software whenever required. So, the combination of database and DBMS software enables one to store, retrieve and access data with considerate accuracy and security.

4) What are the advantages of DBMS?

- Redundancy control
 - Restriction for unauthorized access
 - Provides multiple user interfaces
 - Provides backup and recovery
 - Enforces integrity constraints
 - Ensure data consistency
 - Easy accessibility
 - Easy data extraction and data processing due to the use of queries
-

5) What is a checkpoint in DBMS?

The Checkpoint is a type of mechanism where all the previous logs are removed from the system and permanently stored in the storage disk.

There are two ways which can help the DBMS in recovering and maintaining the ACID properties, and they are- maintaining the log of each transaction and maintaining shadow pages. So, when it comes to log based recovery system, checkpoints come into existence. Checkpoints are those points to which the database engine can recover after a crash as a specified minimal point from where the transaction log record can be used to recover all the committed data up to the point of the crash.

6) When does checkpoint occur in DBMS?

A checkpoint is like a snapshot of the DBMS state. Using checkpoints, the DBMS can reduce the amount of work to be done during a restart in the event of subsequent crashes. Checkpoints are used for the recovery of the database after the system crash. Checkpoints are used in the log-based recovery system. When due to a system crash we need to restart the system then at that point we use checkpoints. So that, we don't have to perform the transactions from the very starting.

7) What do you mean by transparent DBMS?

The transparent DBMS is a type of DBMS which keeps its physical structure hidden from users. Physical structure or physical storage structure implies to the memory manager of the DBMS, and it describes how the data stored on disk.

8) What are the unary operations in Relational Algebra?

PROJECTION and SELECTION are the unary operations in relational algebra. Unary operations are those operations which use single operands. Unary operations are SELECTION, PROJECTION, and RENAME.

As in SELECTION relational operators are used for example - =, <=, >=, etc.

9) What is RDBMS?

RDBMS stands for Relational Database Management Systems. It is used to maintain the data records and indices in tables. RDBMS is the form of DBMS which uses the structure to identify and access data concerning the other piece of data in the database. RDBMS is the system that enables you to perform different operations such as- update, insert, delete, manipulate and administer a relational database with minimal difficulties. Most of the time RDBMS use SQL language because it is easily understandable and is used for often.

10) How many types of database languages are?

There are four types of database languages:

- **Data Definition Language (DDL)** e.g., CREATE, ALTER, DROP, TRUNCATE, RENAME, etc. All these commands are used for updating the data that's why they are known as Data Definition Language.
- **Data Manipulation Language (DML)** e.g., SELECT, UPDATE, INSERT, DELETE, etc. These commands are used for the manipulation of already updated data that's why they are the part of Data Manipulation Language.
- **DATA Control Language (DCL)** e.g., GRANT and REVOKE. These commands are used for giving and removing the user access on the database. So, they are the part of Data Control Language.
- **Transaction Control Language (TCL)** e.g., COMMIT, ROLLBACK, and SAVEPOINT. These are the commands used for managing transactions in the database. TCL is used for managing the changes made by DML.

Database language implies the queries that are used for the update, modify and manipulate the data.

11) What do you understand by Data Model?

The Data model is specified as a **collection of conceptual tools for describing data, data relationships, data semantics and constraints**. These models are used to describe the relationship between the entities and their attributes.

There is the number of data models:

- Hierarchical data model
- network model
- relational model
- Entity-Relationship model and so on.

12) Define a Relation Schema and a Relation.

A Relation Schema is specified as a set of attributes. It is also known as table schema. It defines what the name of the table is. Relation schema is known as the blueprint with the help of which we can explain that how the data is organized into tables. This blueprint contains no data.

A relation is specified as a set of tuples. A relation is the set of related attributes with identifying key attributes

See this example:

Let r be the relation which contains set tuples $(t_1, t_2, t_3, \dots, t_n)$. Each tuple is an ordered list of n -values $t = (v_1, v_2, \dots, v_n)$.

13) What is the Relationship?

The Relationship is defined as an association among two or more entities. There are three type of relationships in DBMS-

One-To-One: Here one record of any object can be related to one record of another object.

One-To-Many (many-to-one): Here one record of any object can be related to many records of other object and vice versa.

Many-to-many: Here more than one records of an object can be related to n number of records of another object.

14) What are the disadvantages of file processing systems?

- Inconsistent
- Not secure
- Data redundancy

- Difficult in accessing data
 - Data isolation
 - Data integrity
 - Concurrent access is not possible
 - Limited data sharing
 - Atomicity problem
-

15) What is data abstraction in DBMS?

Data abstraction in DBMS is a process of hiding irrelevant details from users. Because database systems are made of complex data structures so, it makes accessible the user interaction with the database.

For example: We know that most of the users prefer those systems which have a simple GUI that means no complex processing. So, to keep the user tuned and for making the access to the data easy, it is necessary to do data abstraction. In addition to it, data abstraction divides the system in different layers to make the work specified and well defined.

16) What are the three levels of data abstraction?

Following are three levels of data abstraction:

Physical level: It is the lowest level of abstraction. It describes how data are stored.

Logical level: It is the next higher level of abstraction. It describes what data are stored in the database and what the relationship among those data is.

View level: It is the highest level of data abstraction. It describes only part of the entire database.

For example- User interacts with the system using the GUI and fill the required details, but the user doesn't have any idea how the data is being used. So, the abstraction level is entirely high in VIEW LEVEL.

Then, the next level is for PROGRAMMERS as in this level the fields and records are visible and the programmers have the knowledge of this layer. So, the level of abstraction here is a little low in VIEW LEVEL.

And lastly, physical level in which storage blocks are described.

17) What is DDL (Data Definition Language)?

Data Definition Language (DDL) is a standard for commands which defines the different structures in a database. Most commonly DDL statements are CREATE, ALTER, and DROP. These commands are used for updating data into the database.

18) What is DML (Data Manipulation Language)?

Data Manipulation Language (DML) is a language that enables the user to access or manipulate data as organized by the appropriate data model. For example- SELECT, UPDATE, INSERT, DELETE.

There is two type of DML:

Procedural DML or Low level DML: It requires a user to specify what data are needed and how to get those data.

Non-Procedural DML or High level DML: It requires a user to specify what data are needed without specifying how to get those data.

19) What is Relational Algebra?

Relational Algebra is a Procedural Query Language which contains a set of operations that take one or two relations as input and produce a new relationship. Relational

algebra is the basic set of operations for the relational model. The decisive point of relational algebra is that it is similar to the algebra which operates on the number.

There are few fundamental operations of relational algebra:

- select
 - project
 - set difference
 - union
 - rename,etc.
-

20) What do you mean by durability in DBMS?

Once the DBMS informs the user that a transaction has completed successfully, its effect should persist even if the system crashes before all its changes are reflected on disk. This property is called durability. Durability ensures that once the transaction is committed into the database, it will be stored in the non-volatile memory and after that system failure cannot affect that data anymore.

21) What is normalization?

Normalization is a process of analysing the given relation schemas according to their functional dependencies. It is used to minimize redundancy and also used to minimize insertion, deletion and update distractions. Normalization is considered as an essential process as it is used to avoid data redundancy, insertion anomaly, updation anomaly, deletion anomaly.

There most commonly used normal forms are:

- First Normal Form(1NF)
- Second Normal Form(2NF)
- Third Normal Form(3NF)

- Boyce & Codd Normal Form(BCNF)
-

22) What is Denormalization?

Denormalization is the process of boosting up database performance and adding of redundant data which helps to get rid of complex data. Denormalization is a part of database optimization technique. This process is used to avoid the use of complex and costly joins. Denormalization doesn't refer to the thought of not to normalize instead of that denormalization takes place after normalization. In this process, firstly the redundancy of the data will be removed using normalization process than through denormalization process we will add redundant data as per the requirement so that we can easily avoid the costly joins.

23) What is functional Dependency?

Functional Dependency is the starting point of normalization. It exists when a relation between two attributes allow you to determine the corresponding attribute's value uniquely. The functional dependency is also known as database dependency and defines as the relationship which occurs when one attribute in a relation uniquely determines another attribute. It is written as $A \rightarrow B$ which means B is functionally dependent on A.

24) What is the E-R model?

E-R model is a short name for the Entity-Relationship model. This model is based on the real world. It contains necessary objects (known as entities) and the relationship among these objects. Here the primary objects are the entity, attribute of that entity, relationship set, an attribute of that relationship set can be mapped in the form of E-R diagram.

In E-R diagram, entities are represented by rectangles, relationships are represented by diamonds, attributes are the characteristics of entities and represented by ellipses, and data flow is represented through a straight line.

25) What is an entity?

The Entity is a set of attributes in a database. An entity can be a real-world object which physically exists in this world. All the entities have their attribute which in the real world considered as the characteristics of the object.

For example: In the employee database of a company, the employee, department, and the designation can be considered as the entities. These entities have some characteristics which will be the attributes of the corresponding entity.

26) What is Weak Entity set?

An entity set that doesn't have sufficient attributes to form a primary key is referred to as a weak entity set. The member of a weak entity set is known as a subordinate entity. Weak entity set does not have a primary key, but we need a mean to differentiate among all those entries in the entity set that depend on one particular strong entity set.

27) What is an attribute?

An attribute refers to a database component. It is used to describe the property of an entity. An attribute can be defined as the characteristics of the entity. Entities can be uniquely identified using the attributes. Attributes represent the instances in the row of the database.

For example: If a student is an entity in the table then age will be the attribute of that student.

28) What is Data Independence?

Data independence specifies that "the application is independent of the storage structure and access strategy of data." It makes you able to modify the schema definition at one level without altering the schema definition in the next higher level.

It makes you able to modify the schema definition in one level should not affect the schema definition in the next higher level.

There are two types of Data Independence:

Physical Data Independence: Physical data is the data stored in the database. It is in the bit-format. Modification in physical level should not affect the logical level.

For example: If we want to manipulate the data inside any table that should not change the format of the table.

Logical Data Independence: Logical data is the data about the database. It basically defines the structure. Such as tables stored in the database. Modification in logical level should not affect the view level.

For example: If we need to modify the format of any table, that modification should not affect the data inside it.

29) What are the three levels of data abstraction?

Following are three levels of data abstraction:

Physical level: It is the lowest level of abstraction. It describes how data are stored.

Logical level: It is the next higher level of abstraction. It describes what data are stored in the database and what relationship among those data.

View level: It is the highest level of data abstraction. It describes only part of the entire database.

For example- User interact with the system using the GUI and fill the required details, but the user doesn't have any idea how the data is being used. So, the abstraction level is absolutely high in VIEW LEVEL.

Then, the next level is for PROGRAMMERS as in this level the fields and records are visible and the programmer has the knowledge of this layer. So, the level of abstraction here is a little low in VIEW LEVEL.

And lastly, physical level in which storage blocks are described.

30) What is Join?

The Join operation is one of the most useful activities in relational algebra. It is most commonly used way to combine information from two or more relations. A Join is always performed on the basis of the same or related column. Most complex queries of SQL involve JOIN command.

There are following types of join:

- Inner joins: Inner join is of 3 categories. They are:
 - Theta join
 - Natural join
 - Equi join
 - Outer joins: Outer join have three types. They are:
 - Left outer join
 - Right outer join
 - Full outer join
-

32) What is 1NF?

1NF is the **First Normal Form**. It is the simplest type of normalization that you can implement in a database. The primary objectives of 1NF are to:

- Every column must have atomic (single value)
 - To Remove duplicate columns from the same table
 - Create separate tables for each group of related data and identify each row with a unique column
-

33) What is 2NF?

2NF is the **Second Normal Form**. A table is said to be 2NF if it follows the following conditions:

- The table is in 1NF, i.e., firstly it is necessary that the table should follow the rules of 1NF.
 - Every non-prime attribute is fully functionally dependent on the primary key, i.e., every non-key attribute should be dependent on the primary key in such a way that if any key element is deleted, then even the non_key element will still be saved in the database.
-

34) What is 3NF?

3NF stands for **Third Normal Form**. A database is called in 3NF if it satisfies the following conditions:

- It is in second normal form.
- There is no transitive functional dependency.
- For example: $X \rightarrow Z$

Where:

$X \rightarrow Y$

Y does not \rightarrow X

$Y \rightarrow Z$ so, $X \rightarrow Z$

35) What is BCNF?

BCMF stands for **Boyce-Codd Normal Form**. It is an advanced version of 3NF, so it is also referred to as 3.5NF. BCNF is stricter than 3NF.

A table complies with BCNF if it satisfies the following conditions:

- It is in 3NF.
 - For every functional dependency $X \rightarrow Y$, X should be the super key of the table. It merely means that X cannot be a non-prime attribute if Y is a prime attribute.
-

36) Explain ACID properties

ACID properties are some basic rules, which has to be satisfied by every transaction to preserve the integrity. These properties and rules are:

ATOMICITY: Atomicity is more generally known as 'all or nothing rule.' Which implies all are considered as one unit, and they either run to completion or not executed at all.

CONSISTENCY: This property refers to the uniformity of the data. Consistency implies that the database is consistent before and after the transaction.

ISOLATION: This property states that the number of the transaction can be executed concurrently without leading to the inconsistency of the database state.

DURABILITY: This property ensures that once the transaction is committed it will be stored in the non-volatile memory and system crash can also not affect it anymore.

37) What is the difference between a DELETE command and TRUNCATE command?

DELETE command: DELETE command is used to delete rows from a table based on the condition that we provide in a WHERE clause.

- DELETE command delete only those rows which are specified with the WHERE clause.
- DELETE command can be rolled back.
- DELETE command maintain a log, that's why it is slow.
- DELETE use row lock while performing DELETE function.

TRUNCATE command: TRUNCATE command is used to remove all rows (complete data) from a table. It is similar to the DELETE command with no WHERE clause.

- The TRUNCATE command removes all the rows from the table.
 - The TRUNCATE command cannot be rolled back.
 - The TRUNCATE command doesn't maintain a log. That's why it is fast.
 - TRUNCATE use table log while performing the TRUNCATE function.
-

38) What is 2-Tier architecture?

The **2-Tier architecture** is the same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side.

39) What is the 3-Tier architecture?

The 3-Tier architecture contains another layer between the client and server. Introduction of 3-tier architecture is for the ease of the users as it provides the GUI, which, make the system secure and much more accessible. In this architecture, the application on the client-end interacts with an application on the server which further communicates with the database system.

40) Describe the types of keys?

There are following types of keys:

Primary key: The Primary key is an attribute in a table that can uniquely identify each record in a table. It is compulsory for every table.

Candidate key: The Candidate key is an attribute or set of an attribute which can uniquely identify a tuple. The Primary key can be selected from these attributes.

Super key: The Super key is a set of attributes which can uniquely identify a tuple. Super key is a superset of the candidate key.

Foreign key: The Foreign key is a primary key from one table, which has a relationship with another table. It acts as a cross-reference between tables.

Basic DBMS Interview Questions

41. Are NULL values in a database the same as that of blank space or zero?

No, a NULL value is very different from that of zero and blank space as it represents a value that is assigned, unknown, unavailable, or not applicable as compared to blank space which represents a character and zero represents a number.

Example: NULL value in “number_of_courses” taken by a student represents that its value is unknown whereas 0 in it means that the student hasn’t taken any courses.