DBMS VIVA QUESTIONS

1. What is a DBMS, and how does it differ from a file management system?

A Database Management System is software designed to store, manage, and retrieve data efficiently, securely, and conveniently. It provides a systematic way to create, retrieve, update, and manage data.

The primary difference between a DBMS and a file management system lies in their approach to data management. Data is stored separately in a file management system, and data manipulation is application-specific. It lacks central control over data, leading to data redundancy, inconsistency, and lack of integrity and security controls. Conversely, a DBMS centralizes data management, reducing redundancy and inconsistency and providing controlled access and systematic data organization.

2. Explain the concept of a database schema.

A database schema is the skeleton structure representing the entire database's logical view. It defines how data is organized and how their relations are associated. It includes tables, views, indexes, relationships, stored procedures, and more, defining the entities and their relationships.

3. What is the difference between logical and physical database design?

Logical database design involves creating a detailed database model that includes entities, attributes, and relationships without considering how the data will be stored physically.

Physical database design, involves translating the logical design into a technical specification for storing data. It focuses on how to store and retrieve the data efficiently, considering the DBMS's specific features and limitations.

4. Describe the three levels of data abstraction in a DBMS.

The three levels of data abstraction in a DBMS are:

- Physical Level: The lowest level of abstraction. It describes how data is stored in the database, including the data structures and access methods, such as indexing and hashing.
- Logical Level: This is the next higher level of abstraction. It describes what data is stored in the database and the relationships among those data. The schema is defined at this level, but how data is stored is irrelevant.

- View Level: This is the highest level of abstraction. It involves the way data is seen and managed by end-users. This level provides a way to hide the complexity of the database, showing only data of interest to specific users through views.

5. What is an Entity-Relationship (ER) model?

The Entity-Relationship (ER) model is a high-level conceptual data model used in database design. It allows the representation of the data entities, the relationships between these entities, and the attributes of both entities and relationships.

6. Explain the difference between a primary key and a foreign key.

- Primary Key: A unique identifier for each record in a database table. It ensures that no two rows have the same primary key. It cannot accept null values and uniquely identifies a record in the table.
- Foreign Key: An attribute in one table that links to the primary key of another table. A foreign key can accept multiple null values unless specifically restricted.

7. What is a composite key?

A composite key is a type of key used in the database that consists of two or more columns in a table that can uniquely identify each row. The combination of columns guarantees uniqueness, whereas individual columns do not. Composite keys are used when no single column can uniquely identify each row.

8. Describe normalization and its importance.

Normalization refers to structuring data within a database to reduce redundancy and enhance data integrity. This entails breaking down extensive tables into smaller, more manageable ones and establishing connections among them. The importance of normalization includes:

- Reducing data redundancy which saves storage and improves data integrity.
- Ensuring data dependencies make sense to prevent anomalies in data modification.
- Making the database more flexible by facilitating easier updates, insertions, and deletions.

9. What are the different normal forms, and why are they used?

The different normal forms are a series of guidelines to ensure the database is designed to reduce redundancy and dependency. The main normal forms are:

- First Normal Form (1NF): Ensures each table cell contains a single value and entry per column is of a similar kind.
- Second Normal Form (2NF): Achieved when it is in 1NF and all non-key attributes are fully functional and dependent on the primary key.
- Third Normal Form (3NF): Achieved when it is in 2NF and all the columns depend only on the primary key, not other non-key attributes.
- Boyce-Codd Normal Form (BCNF): A stronger version of 3NF, ensuring every determinant is a candidate key.
- Additional forms include 4NF (eliminating multi-valued dependencies) and 5NF (eliminating join dependencies).
- They reduce redundancy, prevent update anomalies, and ensure data integrity.

10. How does denormalization differ from normalization?

Denormalization adds some redundancy to a relational database that was removed during normalization. The purpose of denormalization is often to improve the performance of read operations by reducing the number of joins needed between tables.

11. What is SQL, and what are its main components?

SQL(Structured Query language) is a standard programming language designed to manage and manipulate relational databases. The main components of SQL include:

- Data Definition Language (DDL): This includes commands such as CREATE, ALTER, and DROP, which define and modify the database structure or schema.
- Data Manipulation Language (DML): This includes commands like INSERT, UPDATE, DELETE, and SELECT, which allow users to manipulate the data stored in the database.
- Data Control Language (DCL): This includes commands like GRANT and REVOKE, which control access to the data.
- Transaction Control Commands: These commands, such as COMMIT and ROLLBACK, manage transaction processing in the database.

12. How do you write a SQL query to select all records from a table?

SELECT * FROM employees;

13. What is a join in SQL, and what are the different types of joins?

A join in SQL combines fields from two tables using values common to each. The main types of joins include:

- INNER JOIN: Returns rows when at least one match exists in both tables.
- LEFT JOIN (or LEFT OUTER JOIN): Returns all rows from the left table and the matched rows from the right table. If there's no match, NULL values are returned for columns of the right table.
- RIGHT JOIN (or RIGHT OUTER JOIN): This function returns all rows from the right table and the matched rows from the left table. If there's no match, NULL values are returned for the columns of the left table.
- FULL JOIN (or FULL OUTER JOIN): Returns rows when there is a match in one of the tables. It effectively combines the results of both LEFT JOIN and RIGHT JOIN.

14. Explain the use of indexes in a database.

Indexes in a database speed up data retrieval from a table. They work like an index in a book, allowing the database to find data without scanning the entire table.

15. Describe the difference between the HAVING and WHERE clause.

- WHERE Clause: This clause filters rows before groupings are made. It applies conditions to individual records in the table(s) involved in the SQL statement. The WHERE clause cannot be used with aggregate functions.
- HAVING Clause: This clause filters groups after applying the GROUP BY clause. It is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to filter the results of a GROUP BY operation.

16. What are stored procedures, and what are their advantages?

Stored procedures are precompiled SQL statements stored in the database. They can be executed with a single call, allowing complex operations to be encapsulated as public functions on the database server. The advantages of stored procedures include:

- Performance: Stored procedures are precompiled, running faster than dynamic SQL queries.
- Reduced Network Traffic: Only the call to the procedure is sent across the network, not the procedure code itself.

- Security: Stored procedures can provide an additional layer of security, allowing users to execute complex operations without granting them direct access to the underlying tables.
- Reusability and Maintainability: Stored procedures allow you to centralize logic in the database, making the code more reusable and easier to maintain.

17. What is a database transaction?

A database transaction is a logical work unit performed within a database management system (DBMS). It represents a sequence of one or more operations, such as inserts, updates, or deletions, treated as a single, indivisible unit. Transactions ensure data integrity and consistency by completing all the operations successfully (commit) or rolling back the changes if an error occurs (rollback).

18. Explain the ACID properties of a transaction.

ACID properties ensure that database transactions are reliable and maintain data integrity:

- Atomicity: A transaction is atomic, meaning it either completes in its entirety or not at all. If any part fails, the entire transaction returns to its original state.
- Consistency: The database remains consistent before and after the transaction. All constraints, rules, and relationships defined in the database are enforced during the transaction.
- Isolation: Each transaction is isolated from other transactions until it is completed. This ensures that the intermediate state of one transaction is invisible to other concurrent transactions.
- Durability: Once a transaction is committed, its changes are permanent and persist even in system failure. The changes are stored permanently in non-volatile memory (e.g., disk).

19. What is concurrency control, and why is it important?

Concurrency control manages simultaneous access to shared resources in a database by multiple users or transactions. It ensures that transactions execute correctly and maintain data consistency despite running concurrently. Concurrency control is essential because it prevents data corruption, maintains data integrity, and ensures the isolation of transactions from one another.

20. Describe the difference between optimistic and pessimistic locking.

- Optimistic Locking: In optimistic locking, the system does not acquire locks on data resources until the transaction is ready to commit. It assumes that conflicts between transactions are rare. Instead of locking, it checks whether any other transaction has modified

the data since it was last read. The transaction is aborted if a conflict is detected and the user is prompted to retry.

- Pessimistic Locking: In pessimistic locking, locks are acquired on data resources as soon as they are accessed. It assumes that conflicts between transactions are common. This approach ensures that other transactions cannot access the locked resources until the lock is released. Pessimistic locking can lead to reduced concurrency but guarantees data consistency.

21. 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.

22. How does a database maintain data integrity during transactions?

A database maintains data integrity during transactions by enforcing the ACID properties:

- Atomicity
- Consistency
- Isolation
- Durability

23. What is a two-phase commit?

A two-phase commit (2PC) protocol ensures the atomicity of distributed transactions involving multiple databases or resources. It consists of two phases:

- Prepare Phase: In this phase, the transaction coordinator (typically the database management system) asks all participants (databases or resources involved in the transaction) to prepare to commit the transaction.

- Commit Phase: If all participants are prepared to commit, the transaction coordinator sends a commit command to all participants. If any participant is not prepared to commit, the coordinator sends a rollback command to all participants to abort the transaction.
- The two-phase commit protocol ensures that all participants commit the transaction, or none do, preventing inconsistencies in distributed systems.

24. Describe the role of a transaction log in a DBMS.

A transaction log (also known as a redo log or audit trail) is a file that records all changes made to the database during transactions.

25. What are savepoints in a transaction?

Savepoints are markers within a transaction that allow you to define points you can roll back without rolling back the entire transaction. They provide a way to divide a transaction into smaller units and selectively undo parts of the transaction if necessary.

26.What is a distributed database and what are its advantages?

A distributed database is a system in which data is stored and managed across multiple computing devices or nodes, often in different geographical locations. The advantages of distributed databases include:

- Improved Scalability: Distributed databases can scale horizontally by adding more nodes, allowing them to handle larger volumes of data and higher numbers of users.
- Increased Availability: Data replication across multiple nodes improves fault tolerance and availability. If one node fails, other nodes can still access the data.
- Geographic Distribution: Distributed databases can store data closer to where needed, reducing latency and improving users' performance in different locations.
- Better Performance: Distributing data and processing across multiple nodes can improve query performance by parallelizing data retrieval and processing tasks.

27. Explain the concept of database replication.

Database replication is copying data from one database to another in real time or near real time. Its primary purpose is to improve data availability, fault tolerance, and disaster recovery.

28. What is a NoSQL database, and how does it differ from a relational database?

A NoSQL (Not Only SQL) database offers a method for storing and accessing data, diverging from the tabular structures employed in relational databases. Unlike relational databases, NoSQL databases do not strictly adhere to the ACID properties and use different data models, such as key-value, document, columnar, or graph-based models.

29. What is Big Data and how does DBMS handle it?

Big Data refers to large volumes of structured, semi-structured, and unstructured data that cannot be processed or analyzed using traditional database management tools and techniques. Specialized Big Data platforms and technologies, such as Hadoop, Spark, and NoSQL databases, are used to handle Big Data.

30. How do database triggers work?

Database triggers are special stored procedures automatically executed in response to certain events or actions in a database. Triggers are used to enforce business rules, maintain data integrity, and automate database tasks. Database triggers are commonly used with constraints, stored procedures, and other objects to enforce complex business logic and ensure data consistency.

31. What is SQL injection, and how can it be prevented?

SQL injection is a type of cyber attack in which malicious SQL code is injected into input fields or parameters of a web application to manipulate the database query and gain unauthorized access to the database.

32. How can data redundancy be managed in a DBMS?

Data redundancy can be managed in a DBMS through various techniques, including:

- Normalization: Organizing data into separate tables and eliminating redundant data by breaking it down into smaller, related tables.
- Denormalization: Introducing controlled redundancy by duplicating some data to improve query performance or simplify data retrieval.
- Use of Foreign Keys: Establishing relationships between tables using foreign keys to ensure data integrity and prevent redundant data.
- Data Deduplication: Identifying and removing duplicate records or data elements from the database to reduce redundancy.

- Data Compression: Using compression techniques to store data more efficiently and reduce storage requirements for redundant data.
- Regular Maintenance: Performing routine cleanup, data archiving, and data purging to remove outdated or unnecessary data from the database.

33. What is a database transaction?

A database transaction is a logical work unit performed within a database management system (DBMS). It represents a sequence of one or more operations, such as inserts, updates, or deletions, treated as a single, indivisible unit. Transactions ensure data integrity and consistency by completing all the operations successfully (commit) or rolling back the changes if an error occurs (rollback).

34. How does a database maintain data integrity during transactions?

A database maintains data integrity during transactions by enforcing the ACID properties:

- Atomicity
- Consistency
- Isolation
- Durability

35. What is a two-phase commit?

A two-phase commit (2PC) protocol ensures the atomicity of distributed transactions involving multiple databases or resources. It consists of two phases:

- Prepare Phase: In this phase, the transaction coordinator (typically the database management system) asks all participants (databases or resources involved in the transaction) to prepare to commit the transaction.
- Commit Phase: If all participants are prepared to commit, the transaction coordinator sends a commit command to all participants. If any participant is not prepared to commit, the coordinator sends a rollback command to all participants to abort the transaction.
- The two-phase commit protocol ensures that all participants commit the transaction, or none do, preventing inconsistencies in distributed systems.

36. 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

37.What is an Entity type?

An entity type is specified as a collection of entities, having the same attributes

38. What is an Entity set?

The entity set specifies the collection of all entities of a particular entity type in the database

example, a student has student_id, department, and course as its characteristics.

39. What is an Extension of entity type?

An extension of an entity type is specified as a collection of entities of a particular entity type that are grouped into an entity set.

40. 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

41.What is an attribute?

An attribute refers to a database component. It is used to describe the property of an entity

42.What are the integrity rules in DBMS?

Data integrity is one significant aspect while maintaining the database. So, data integrity is enforced in the database system by imposing a series of rules. Those set of integrity is known as the integrity rules.

There are two integrity rules in DBMS:

Entity Integrity : It specifies that "Primary key cannot have a NULL value."

Referential Integrity: It specifies that "Foreign Key can be either a NULL value or should be the Primary Key value of other relation

43. 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

44. Different 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.