

# Interview -01

## Interview Questions

### (Practice Project)



## Easy

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

**Ans:** A Database Management System (DBMS) is specialized software that manages, stores, and retrieves data efficiently, securely, and systematically. Unlike a file management system, which stores data in separate files and is application-specific, a DBMS centralized data management. This centralization minimizes redundancy, ensures consistency, and provides robust data integrity and security.

### 2. Explain the concept of a database schema.

**Ans:** A database schema is the logical framework or blueprint of a database. It outlines the structure of the database, including tables, views, indexes, and relationships, and defines how data is organized and interconnected. The schema is established during the database design phase and typically remains stable over time.

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

**Ans:** Logical database design focuses on defining the data model at a conceptual level, detailing entities, attributes, and relationships without considering the physical storage. Physical database design, on the other hand, translates the logical model into a technical specification for actual data storage, addressing storage structures, indexing, and data retrieval efficiency.

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

**Ans:**

- **Physical Level:** The lowest abstraction level, detailing how data is physically stored, including data structures and access methods.
- **Logical Level:** The middle abstraction level, defining what data is stored and the relationships among data, with the schema providing a view of the database.
- **View Level:** The highest abstraction level, presenting data to users in a simplified manner, often through views that hide the complexity of the underlying database.

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

**Ans:** The Entity-Relationship (ER) model is a conceptual framework used to design databases. It represents data entities, their attributes, and relationships among entities, providing a visual and conceptual map of the database's structure.

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

**Ans:**

- **Primary Key:** A unique identifier for each record in a table, ensuring uniqueness and non-nullability within the table.
- **Foreign Key:** An attribute in one table that references the primary key of another table, establishing a relationship between the two tables and allowing for multiple null values unless otherwise restricted.

## 7. What is a composite key?

**Ans:** A composite key is a key that consists of two or more columns in a table, combined to uniquely identify each row. It is used when a single column cannot uniquely identify a row on its own.

## 8.What are views in SQL?

**Ans:** A view is a virtual table which consists of a subset of data contained in a table. Views are not virtually present, and it takes less space to store. View can have data of one or more tables combined, and it is depending on the relationship

## 9. Difference between DELETE and TRUNCATE Commands.

**Ans:** DELETE command is used to remove rows from the table, and WHERE clause can be used for a conditional set of parameters. Commit and Rollback can be performed after the delete statement.

**TRUNCATE** removes all rows from the table. Truncate operation cannot be rolled back.

## 10. What are aggregate and scalar functions?

**Ans:**

Functions are methods used to perform data operations. SQL has many in-built functions used to perform string concatenations, mathematical calculations etc. SQL functions are categorized into the following two categories: Aggregate Functions and Scalar Functions.

**The Aggregate Functions in SQL perform calculations on a group of values and then return a single value. Following are a few of the most commonly used Aggregate Functions:**

Function	Description
SUM()	Used to return the sum of a group of values.
COUNT()	Returns the number of rows either based on a condition, or without a condition.
AVG()	Used to calculate the average value of a numeric column.
MIN()	This function returns the minimum value of a column.
MAX()	Returns a maximum value of a column.
FIRST()	Used to return the first value of the column.
LAST()	This function returns the last value of the column.

**The Scalar Functions in SQL are used to return a single value from the given input value. Following are a few of the most commonly used Scalar Functions:**

Function	Description
LCASE()	Used to convert string column values to lowercase
UCASE()	This function is used to convert a string column values to Uppercase.
LEN()	Returns the length of the text values in the column.
MID()	Extracts substrings in SQL from column values having String data type.
ROUND()	Rounds off a numeric value to the nearest integer.
NOW()	This function is used to return the current system date and time.
FORMAT()	Used to format how a field must be displayed.

## 11. What is ETL in SQL?

**Ans:** ETL stands for Extract, Transform and Load. It is a three step process, where we would have to start off by extracting the data from sources. Once we collate the data from different sources, we have our raw data. This raw data has to be transformed into a tidy format, which will come in the second phase. Finally, we would have to load this tidy data into tools which would help us to find insights.

## 12. What is the difference between CHAR and VARCHAR2 data types in SQL server?

**Ans:** When stored in a database, varchar2 uses only the allocated space. E.g. if you have a varchar2(1999) and put 50 bytes in the table, it will use 52 bytes.  
But when stored in a database, char always uses the maximum length and is blank-padded. E.g. if you have char(1999) and put 50 bytes in the table, it will consume 2000 bytes.

## 13. What is the difference between SQL and MySQL?

**Ans:** SQL is a Structured Query Language that is used for manipulating and accessing the relational database. On the other hand, MySQL itself is a relational database that uses SQL as the standard database language.

## 14. What is a CLAUSE?

**Ans:**

SQL clause is defined to limit the result set by providing conditions to the query. This usually filters some rows from the whole set of records. Example – Query that has WHERE condition

## 15. Explain UNION, MINUS and INTERACT commands?

**Ans:**

- UNION operator is used to combine the results of two tables, and it eliminates duplicate rows from the tables.
- MINUS operator is used to return rows from the first query but not from the second query. Matching records of first and second query and other rows from the first query will be displayed as a result set.
- INTERSECT operator is used to return rows returned by both the queries.

## 16. What is the main difference in the BETWEEN and IN condition operators?

**Ans:**

BETWEEN operator is used to display rows based on a range of values in a row whereas the IN condition operator is used to check for values contained in a specific set of values.

- Example of BETWEEN: SELECT \* FROM Students where ROLL\_NO BETWEEN 10 AND 50;
- Example of IN: SELECT \* FROM students where ROLL\_NO IN (8,15,25);

## 17. What is the difference between JOIN and UNION?

**Ans:**

- JOIN in SQL is used to combine data from many tables based on a matched condition between them. The data combined using the JOIN statement results into new columns.
- UNION in SQL is used to combine the result-set of two or more SELECT statements. The data combined using UNION statement results into new distinct rows

## Medium

### 18. Describe normalization and its importance.

**Ans:** Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. It involves dividing large tables into smaller, related ones and defining appropriate relationships. This process prevents anomalies, ensures data consistency, and enhances database flexibility.

### 19. What are the different normal forms, and why are they used? And what is denormalization?

**Ans:**

- **First Normal Form (1NF):** Ensures that each cell in a table contains a single value and each column has homogeneous data.
- **Second Normal Form (2NF):** Achieved when the table is in 1NF, and all non-key attributes are fully functionally dependent on the primary key.
- **Third Normal Form (3NF):** Achieved when the table is in 2NF, and all columns depend solely on the primary key, with no transitive dependencies.
- **Boyce-Codd Normal Form (BCNF):** A stricter version of 3NF where every determinant is a candidate key.
- **Fourth Normal Form (4NF):** Eliminates multivalued dependencies.
- **Fifth Normal Form (5NF):** Eliminates join dependencies.

Denormalization is a technique used to access the data from higher to lower normal forms of database. It is also a process of introducing redundancy into a table by incorporating data from the related tables.

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

**Ans:**

SQL (Structured Query Language) is a standard language used to manage and manipulate relational databases. Its main components include:

- **Data Definition Language (DDL):** Commands like CREATE, ALTER, and DROP for defining and modifying database structures.
- **Data Manipulation Language (DML):** Commands like INSERT, UPDATE, DELETE, and SELECT for managing data.
- **Data Control Language (DCL):** Commands like GRANT and REVOKE for managing permissions.
- **Transaction Control Commands:** Commands such as COMMIT and ROLLBACK for managing transactions.

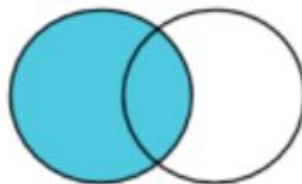
### 21. Explain the difference between DDL, DML, and DCL in SQL.

**Ans:**

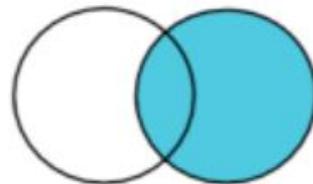
- - **DDL (Data Definition Language):** Defines or modifies database structures, such as creating or altering tables.
- - **DML (Data Manipulation Language):** Manages data within schema objects, including inserting, updating, and retrieving data.
- - **DCL (Data Control Language):** Manages database access and permissions, including granting and revoking user rights.

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

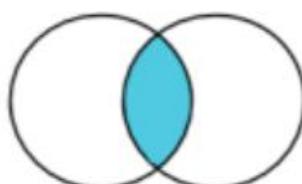
**Ans:**



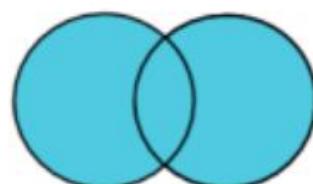
**Left Join**



**Right Join**



**Inner Join**



**Full Outer Join**

A join in SQL combines rows from two or more tables based on a related column. Types of joins include:

- **INNER JOIN:** Returns rows with matching values in both tables.
- **LEFT JOIN (LEFT OUTER JOIN):** Returns all rows from the left table and matched rows from the right table; NULL for unmatched rows in the right table.
- **RIGHT JOIN (RIGHT OUTER JOIN):** Returns all rows from the right table and matched rows from the left table; NULL for unmatched rows in the left table.
- **FULL JOIN (FULL OUTER JOIN):** Returns rows with matches in either table, combining results from both LEFT JOIN and RIGHT JOIN.

## 23. How can you optimize a slow-running SQL query?

**Ans:** To optimize a slow-running SQL query, consider:

- **Using indexes:** Ensure indexes are applied to frequently queried columns.
- **Optimizing joins:** Reduce the number of joins and ensure efficient join order.
- **Limiting data:** Use WHERE clauses to filter results early and limit the amount of data returned.
- **Avoiding SELECT:** Specify only the columns needed.
- **Using query optimization tools:** Analyze queries with built-in tools and explain plans.

## 24. Describe the difference between the HAVING and WHERE clause.

**Ans:**

- **WHERE Clause:** Filters records before grouping occurs; cannot be used with aggregate functions.
- **HAVING Clause:** Filters groups after aggregation and the GROUP BY clause; used with aggregate functions to filter grouped results.

## Hard

### 25. Explain the ACID properties of a transaction.

**Ans:** 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).

### 26. What are the common security threats to a database?

**Ans:** Common security threats to a database include:

**Unauthorized Access:** Unauthorized users gain access to sensitive data or database resources.

**SQL Injection:** Attackers inject malicious SQL code into input fields to manipulate database queries and gain unauthorized access.

**Data Breaches:** Unauthorized access or disclosure of sensitive data, often due to inadequate access controls or encryption.

**Data Manipulation:** Malicious users modify or delete data, leading to data loss or corruption.

**Denial of Service (DoS):** Attackers flooding the database server with requests to overload and disrupt its normal operation.

**Insider Threats:** Malicious or negligent actions by employees or trusted users, such as stealing data or leaking sensitive information.

### 27. How does encryption protect database data?

**Ans:** Encryption protects database data by converting it into a ciphertext that can only be decrypted with the appropriate decryption key. Encrypted data is unreadable and unintelligible to unauthorized users or attackers who gain unauthorized access to the database.

Encryption helps ensure data confidentiality by preventing unauthorized access to sensitive information, even if the database is compromised. Common encryption techniques used in database systems include column-level encryption, transparent data encryption (TDE), and data encryption in transit using SSL/TLS protocols.

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

**Ans:** 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. SQL injection attacks can result in data leakage, data loss, unauthorized access, and database corruption. SQL injection attacks can be prevented by:

Using parameterized queries or prepared statements to sanitize user input and prevent injection of malicious SQL code.

Implementing input validation and data sanitization ensures user input conforms to expected formats and does not contain malicious characters.

Escaping special characters in user input before including them in database queries.

Limiting database privileges and access rights to minimize the impact of a successful SQL injection attack.

Regularly updating and patching web applications and database systems to fix vulnerabilities that attackers could exploit.

## **29. How can data redundancy be managed in a DBMS?**

**Ans:** 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.

## **30. What is a NoSQL database, and how does it differ from a relational database?**

**Ans:** A NoSQL (Not Only SQL) database offers a method for storing and accessing data, diverging from the tabular structures employed in relational databases. It is designed to handle large volumes of structured, semi-structured, and unstructured data and is optimized for horizontal scalability and distributed data architectures. 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.