

Interview questions

EASY:

What is SQL? Explain its purpose and usage.
What are the different types of SQL statements? Explain each.
What is the difference between SQL and MySQL?
What is the difference between CHAR and VARCHAR data types?
Explain the differences between the primary key, foreign key, and unique key.
What is an index in SQL? How does it improve query performance?
What is the difference between a view and a table?
How do you handle NULL values in SQL?
What are the different types of constraints in SQL?
What is the difference between a database and a schema?

MEDIUM:

Explain the concept of data normalization and its importance in databases.
What are the ACID properties in SQL?
How can you prevent SQL injection attacks?
Explain the differences between a subquery and a join? When would you use each?
Explain the differences between UNION and UNION ALL in SQL.
What is the difference between the WHERE and HAVING clauses?
Explain the differences between the DELETE, TRUNCATE, and DROP commands.
How can you retrieve the top N records from a table?
What is the difference between a left join and an inner join?
How do you perform data import and export in SQL?

TOUGH:

1 - What are triggers in SQL? When and why would you use them?
2 - How do you perform advanced aggregation and grouping operations in SQL?
3 - Discuss different types of joins and when to use each one.

EASY:

1- What is SQL? Explain its purpose and usage.

Answer: SQL (Structured Query Language) is a programming language used to manage and manipulate relational databases. Its purpose is to interact with databases to create, retrieve, update, and delete data. It allows users to define and modify database structures, perform calculations, and retrieve information based on specific criteria.

2 - What are the different types of SQL statements? Explain each. Answer:

SELECT: Retrieves data from one or more tables.

INSERT: Inserts new data into a table.

UPDATE: Modifies existing data in a table.

DELETE: Removes data from a table.

CREATE: Creates a new table, view, or database. ALTER: Modifies the structure of a table or view. DROP: Deletes a table, view, or database.

TRUNCATE: Removes all data from a table.

GRANT: Provides user access privileges.

REVOKE: Revokes user access privileges.

3 - What is the difference between SQL and MySQL?

Answer: SQL is a programming language used to manage relational databases, while MySQL is a specific relational database management system (RDBMS) that uses SQL as its language for interacting with the database. SQL is a standardized language that is used by various RDBMSs, including MySQL, Oracle, SQL Server, and others.

4 - What is the difference between CHAR and VARCHAR data types? Answer:

CHAR: It is a fixed-length data type that stores a fixed number of characters. If the stored value is shorter than the specified length, it is padded with spaces. For example, a CHAR(10) column will always occupy 10 characters of storage.

VARCHAR: It is a variable-length data type that stores a variable number of characters. It only uses the necessary amount of storage based on the actual data length. For example, a VARCHAR(10) column will occupy as many characters as the stored value requires (up to 10 characters).

5 - Explain the differences between the primary key, foreign key, and unique key. Answer:

Primary Key: It is a column or combination of columns that uniquely identifies each row in a table. It ensures data integrity and enforces uniqueness.

Foreign Key: It establishes a relationship between two tables by referencing the primary key of another table. It maintains referential integrity and ensures data consistency across related tables.

Unique Key: It ensures the uniqueness of values in a column or a combination of columns. Unlike the primary key, it allows NULL values and allows multiple unique keys in a table.

6 -What is an index in SQL? How does it improve query performance?

Answer: An index in SQL is a data structure that improves the speed of data retrieval operations on database tables. It works like an index in a book, allowing the database to quickly locate the desired data based on the indexed column(s). By creating an index on frequently queried columns, the database can avoid scanning the entire table, leading to faster query performance.

7- What is the difference between a view and a table?

Answer: A table is a physical storage structure that holds data in rows and columns, whereas a view is a virtual table created as a saved query result. A table stores data permanently, while a view is a logical representation of data derived from one or more tables. Views provide a way to simplify complex queries, restrict access to sensitive data, and present a customized view of the underlying data.

8- How do you handle NULL values in SQL?

Answer: NULL represents the absence of a value in SQL. To handle NULL values, you can use the IS NULL and IS NOT NULL operators to check if a column contains NULL or not. Additionally, you can use the COALESCE function to replace NULL values with a specified default value in the result set.

9 -What are the different types of constraints in SQL? Answer:

NOT NULL: Ensures that a column does not contain NULL values.

UNIQUE: Enforces the uniqueness of values in a column or a combination of columns.

PRIMARY KEY: Uniquely identifies each row in a table.

FOREIGN KEY: Establishes a relationship between two tables based on the values of a column.

CHECK: Defines a condition that must be true for the column's values.

DEFAULT: Specifies a default value for a column when no value is provided.

INDEX: Improves the performance of data retrieval operations.

10 - What is the difference between a database and a schema?

Answer: In SQL, a database is a collection of related tables, views, and other database objects, along with the data stored in those objects. It is the overall container for organizing and managing data. On the other hand, a schema is a logical container within a database that allows you to group related database objects together. It provides a way to organize and separate objects, such as tables and views, into logical units.

MEDIUM:

1 - Explain the concept of data normalization and its importance in databases.

Answer: Data normalization is the process of organizing and structuring a database design to eliminate data redundancy and dependency issues. It involves breaking down a database into smaller, logical tables and defining relationships between them. Normalization reduces data duplication, improves data integrity, and ensures efficient data management by eliminating anomalies and inconsistencies.

2 - What are the ACID properties in SQL?

Answer: ACID stands for Atomicity, Consistency, Isolation, and Durability, which are essential properties that guarantee the reliability and integrity of database transactions.

Atomicity ensures that a transaction is treated as a single, indivisible unit of work. It either completes entirely or is rolled back if any part fails.

Consistency ensures that a transaction brings the database from one valid state to another. It enforces integrity constraints and rules defined on the database.

Isolation ensures that concurrent transactions do not interfere with each other. Each transaction operates independently, as if it were the only transaction executing.

Durability ensures that once a transaction is committed, its changes are permanently saved and will survive any subsequent failures.

3- How can you prevent SQL injection attacks?

Answer: SQL injection attacks occur when malicious SQL statements are inserted into an application's input, allowing an attacker to manipulate or extract data from the database. To prevent SQL injection attacks, you should:

Use parameterized queries or prepared statements to separate SQL code from user input.

Validate and sanitize user input to ensure it does not contain malicious characters or SQL statements.

Limit database user privileges to minimize the potential damage of an attack.

Regularly update and patch database systems to fix security vulnerabilities.

Implement strict input validation and output encoding practices to protect against cross-site scripting (XSS) attacks.

4 - Explain the differences between a subquery and a join? When would you use each? Answer:

Subquery: A subquery is a query nested inside another query. It is used to retrieve data based on the results of another query. Subqueries are handy when you need to perform complex calculations or filter data using information from another table. You would use a subquery when you want to perform a separate query and use its result as a condition or a value in the outer query.

Join: A join combines rows from two or more tables based on related columns. Joins are useful when you want to fetch data from different tables using a common column or when you want to combine data from related tables. You would use a join when you need to retrieve data from multiple tables simultaneously, merging the data based on a common column.

5 - Explain the differences between UNION and UNION ALL in SQL. Answer:

UNION: The UNION operator combines the result sets of two or more SELECT statements into a single result set. It removes duplicate rows from the final result set.

UNION ALL: The UNION ALL operator also combines the result sets of two or more SELECT statements into a single result set. However, it does not remove duplicate rows, meaning that all rows from all SELECT statements are included in the final result set.

6- What is the difference between the WHERE and HAVING clauses?

Answer:

WHERE: The WHERE clause is used in a SELECT, UPDATE, or DELETE statement to specify conditions for filtering rows from a table. It operates on individual rows and filters rows based on column values before the grouping and aggregation operations occur.

HAVING: The HAVING clause is used in a SELECT statement to specify conditions for filtering rows after the grouping and aggregation operations have taken place. It operates on grouped rows and filters rows based on aggregate function results or grouped column values.

7 - Explain the differences between the DELETE, TRUNCATE, and DROP commands. Answer:

DELETE: The DELETE command is used to remove specific rows from a table based on specified conditions. It is a DML (Data Manipulation Language) command and can be rolled back if used within a transaction.

TRUNCATE: The TRUNCATE command is used to remove all rows from a table. Unlike DELETE, it does not specify any conditions and is faster because it deallocates the storage space. It is also a DDL (Data Definition Language) command and cannot be rolled back.

DROP: The DROP command is used to remove an entire table, including its structure and data. It is a DDL command and cannot be rolled back.

8 -How can you retrieve the top N records from a table?

Answer: In SQL, you can retrieve the top N records from a table using the LIMIT clause (for MySQL and PostgreSQL) or the TOP clause (for SQL Server and Oracle). The syntax varies between database systems, but the general form is:

MySQL / PostgreSQL: `SELECT * FROM table_name LIMIT N;`

SQL Server: `SELECT TOP N * FROM table_name;`

Oracle: `SELECT * FROM (SELECT * FROM table_name) WHERE ROWNUM <= N;`

9 -What is the difference between a left join and an inner join? Answer:

Inner Join: An inner join returns only the matching rows from the joined tables based on the specified join condition. It combines rows from both tables that have matching values in the join column(s).

Left Join: A left join returns all rows from the left (or first) table and the matching rows from the right (or second) table based on the specified join condition. If there is no match in the right table, NULL values are returned.

10 -How do you perform data import and export in SQL?

Answer: In SQL, you can perform data import and export using various methods:

Import: You can use the SQL INSERT statement to insert data into a table by specifying the values directly. Additionally, many database management systems provide tools, such as LOAD DATA INFILE in MySQL or BULK INSERT in SQL Server, to import data from external files into database tables.

Export: You can use the SQL SELECT statement to retrieve data from tables and then export it to external files using database-specific tools or commands, such as SELECT ... INTO OUTFILE in MySQL or bcp command in SQL Server. Alternatively, you can use database administration tools or export functionality within integrated development environments (IDEs) to export data in different formats like CSV, Excel, or XML.

What are triggers in SQL? When and why would you use them?

Triggers in SQL are special stored procedures that are automatically executed when certain events occur, such as inserting, updating, or deleting data in a table. They are used to enforce rules and automate actions in the database.

For example, let's say you have a table for orders, and you want to update the inventory whenever a new order is placed. You can use a trigger to automatically perform this update whenever an insert operation happens on the orders table. Triggers are also useful for maintaining data integrity, auditing changes, or performing complex calculations based on specific events.

How do you perform advanced aggregation and grouping operations in SQL?

In SQL, you can perform advanced aggregation and grouping operations using the GROUP BY clause and aggregate functions. The GROUP BY clause groups rows based on one or more columns, and aggregate functions like SUM, COUNT, AVG, MAX, and MIN calculate summary values within each group.

For example, let's say you have a table of sales transactions with columns for product, category, and quantity. To calculate the total quantity sold for each product category, you can write a query like this:

```
SELECT category, SUM(quantity) AS total_quantity
FROM sales
GROUP BY category;
```

This query groups the sales rows by category and calculates the sum of the quantity for each category.

You can also use aggregate functions without the GROUP BY clause to perform calculations on the entire table, like finding the overall maximum or minimum value.

Discuss different types of joins and when to use each one.

Normalization

S

SHADOWING

Sharding in SQL:

- Definition: Sharding is a database architecture strategy in which a large database is divided into smaller, more manageable parts called shards. Each shard is an independent database that can be distributed across different servers or nodes.
- Purpose: The goal of sharding is to improve performance, scalability, and parallelism by distributing data across multiple servers, thereby reducing the load on a single server.

PARTITIONING

- Definition: Partitioning is a database design technique where large tables are divided into smaller, more manageable pieces called partitions. Each partition typically holds a subset of the table's data based on a defined criteria.
- Purpose: Partitioning is used to enhance query performance, simplify data management, and improve maintenance operations by allowing operations to be performed on specific partitions rather than the entire table.