Here are some common SQL interview questions along with their answers, explanations, and examples:

**1. What is SQL?**

**Answer:** SQL (Structured Query Language) is a standard language used to manage and manipulate relational databases. It allows users to interact with the data stored in a database, such as querying, inserting, updating, and deleting data.

**Explanation:** SQL is used by database administrators, developers, and data analysts to perform tasks like querying for information or changing the structure of the database.

**Example:**

SELECT \* FROM Employees;

**2. What are the different types of SQL statements?**

**Answer:** SQL statements are categorized into the following types:

1. **DDL (Data Definition Language):** Defines and manages the structure of the database.
   * CREATE, ALTER, DROP
2. **DML (Data Manipulation Language):** Used for manipulating data.
   * SELECT, INSERT, UPDATE, DELETE
3. **DCL (Data Control Language):** Deals with permissions and access control.
   * GRANT, REVOKE
4. **TCL (Transaction Control Language):** Manages the changes made by DML statements.
   * COMMIT, ROLLBACK, SAVEPOINT

**Explanation:**

* **DDL** is used for creating, modifying, and deleting database objects like tables.
* **DML** works with data itself, used for querying and manipulating data.
* **DCL** is used for managing permissions.
* **TCL** controls transactions and ensures data integrity.

**3. What is the difference between WHERE and HAVING clauses?**

**Answer:**

* WHERE is used to filter records before grouping (works on individual rows).
* HAVING is used to filter records after grouping (works on aggregated data).

**Explanation:**

* WHERE is used for filtering rows before they are grouped.
* HAVING is used to filter after data has been grouped using GROUP BY.

**Example:**

-- Using WHERE to filter rows

SELECT \* FROM Employees WHERE salary > 50000;

-- Using HAVING to filter grouped data

SELECT department, AVG(salary)

FROM Employees

GROUP BY department

HAVING AVG(salary) > 60000;

**4. What is a JOIN in SQL?**

**Answer:** A JOIN is used to combine rows from two or more tables based on a related column between them.

**Explanation:**

* **INNER JOIN**: Returns only the rows that have matching values in both tables.
* **LEFT JOIN**: Returns all rows from the left table and matching rows from the right table, or NULL if no match.
* **RIGHT JOIN**: Returns all rows from the right table and matching rows from the left table, or NULL if no match.
* **FULL JOIN**: Returns rows when there is a match in either left or right table.
* **CROSS JOIN**: Returns the Cartesian product of both tables (every combination of rows).

**Example:**

-- INNER JOIN

SELECT Employees.name, Departments.department\_name

FROM Employees

INNER JOIN Departments ON Employees.department\_id = Departments.department\_id;

-- LEFT JOIN

SELECT Employees.name, Departments.department\_name

FROM Employees

LEFT JOIN Departments ON Employees.department\_id = Departments.department\_id;

**5. What is a Primary Key?**

**Answer:** A **Primary Key** is a unique identifier for a record in a database table. It ensures that no two rows have the same value for the primary key column.

**Explanation:**

* It must be unique and not NULL for every record.
* Often used to establish relationships between tables using **foreign keys**.

**Example:**

CREATE TABLE Employees (

employee\_id INT PRIMARY KEY,

name VARCHAR(100),

salary DECIMAL(10, 2)

);

**6. What is the difference between DELETE and TRUNCATE?**

**Answer:**

* DELETE is used to remove rows from a table based on a condition. It is a **DML** operation and can be rolled back.
* TRUNCATE removes all rows from a table. It is a **DDL** operation and cannot be rolled back.

**Explanation:**

* **DELETE**: Can be used with a WHERE clause, allowing you to delete specific rows.
* **TRUNCATE**: Removes all rows and resets any auto-increment counter.

**Example:**

-- DELETE

DELETE FROM Employees WHERE employee\_id = 1;

-- TRUNCATE

TRUNCATE TABLE Employees;

**7. What is an Index in SQL?**

**Answer:** An **Index** is a database object used to speed up the retrieval of rows from a table. It works similarly to an index in a book.

**Explanation:**

* Indexes improve query performance by providing a quick lookup path to the data.
* However, they slow down INSERT, UPDATE, and DELETE operations because the index also needs to be updated.

**Example:**

CREATE INDEX idx\_employee\_name ON Employees(name);

**8. What is a Subquery?**

**Answer:** A **Subquery** is a query nested inside another query. It can be used in SELECT, INSERT, UPDATE, or DELETE statements.

**Explanation:**

* A subquery is often used to return a result set that is used by the outer query.

**Example:**

-- Subquery in SELECT

SELECT name

FROM Employees

WHERE salary > (SELECT AVG(salary) FROM Employees);

**9. What is the difference between UNION and UNION ALL?**

**Answer:**

* **UNION** combines the results of two queries, removing duplicates.
* **UNION ALL** combines the results of two queries, including duplicates.

**Explanation:**

* Use UNION when you want distinct results, and UNION ALL when you want to include all duplicates.

**Example:**

-- Using UNION (removes duplicates)

SELECT name FROM Employees

UNION

SELECT name FROM Contractors;

-- Using UNION ALL (includes duplicates)

SELECT name FROM Employees

UNION ALL

SELECT name FROM Contractors;

**10. What is a Foreign Key?**

**Answer:** A **Foreign Key** is a column or set of columns in one table that references the primary key in another table. It is used to enforce referential integrity.

**Explanation:**

* A foreign key ensures that the value in the child table corresponds to a valid value in the parent table.

**Example:**

CREATE TABLE Departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(100)

);

CREATE TABLE Employees (

employee\_id INT PRIMARY KEY,

name VARCHAR(100),

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES Departments(department\_id)

);

These are some common SQL interview questions with answers, explanations, and examples. Be prepared to practice these concepts and write queries based on real-world scenarios!