**INTERVIEW QUESTIONS – 01**

**01. What are the different types of databases, and how do they differ from each other?**

Ans 🡪

**Relational Databases**: Data is organized in tables (e.g., MySQL, PostgreSQL).  
**NoSQL Databases**: Non-relational databases that store data in various formats (e.g., MongoDB).  
**Object-oriented Databases**: Data is represented as objects (e.g., db4o).  
**Distributed Databases**: Data is distributed across multiple locations (e.g., Google Bigtable).

**02. What are SQL clauses, and can you provide examples of commonly used clauses?**

Ans 🡪

SQL clauses are commands used to filter, group, or sort data. Examples include:

**WHERE**: Filters rows (SELECT \* FROM students WHERE age > 18;).

**GROUP BY**: Groups data based on a column (SELECT department, COUNT(\*) FROM employees GROUP BY department;).

**HAVING**: Filters grouped data (SELECT department, COUNT(\*) FROM employees GROUP BY department HAVING COUNT(\*) > 5;).

**ORDER BY**: Sorts data (SELECT \* FROM students ORDER BY marks DESC;).

**03. What is the difference between SQL commands and SQL clauses?**

Ans 🡪

**SQL Commands**: Instructions to perform database operations like creating tables, inserting, or modifying data. Categories include:

**DDL (Data Definition Language)**: CREATE, ALTER, DROP.

**DML (Data Manipulation Language)**: SELECT, INSERT, UPDATE, DELETE.

**DCL (Data Control Language)**: GRANT, REVOKE.

**TCL (Transaction Control Language)**: COMMIT, ROLLBACK.

**SQL Clauses**: Subparts of SQL queries that refine operations (e.g., WHERE, GROUP BY, ORDER BY). Clauses enhance how commands operate.

**04. Can you explain the different types of SQL operators and provide examples?**

Ans 🡪

SQL operators are symbols or keywords used in queries. Types include:

**Arithmetic Operators**: +, -, \*, / (e.g., SELECT salary \* 1.1 FROM employees;).

**Comparison Operators**: =, >, <, != (e.g., SELECT \* FROM products WHERE price > 100;).

**Logical Operators**: AND, OR, NOT (e.g., SELECT \* FROM students WHERE age > 18 AND marks > 75;).

**LIKE Operator**: Matches patterns (e.g., SELECT \* FROM customers WHERE name LIKE 'A%';).

**IN Operator**: Checks within a list (e.g., SELECT \* FROM products WHERE category IN ('Electronics', 'Books');).

**05. What are aggregate functions in SQL, and can you provide examples?**

Ans 🡪

Aggregate functions perform calculations on a set of values:

**COUNT**: Counts rows (SELECT COUNT(\*) FROM orders;).

**SUM**: Sums values (SELECT SUM(salary) FROM employees;).

**AVG**: Calculates average (SELECT AVG(age) FROM students;).

**MAX**/**MIN**: Finds maximum/minimum (SELECT MAX(price) FROM products;).

**06. What is normalization, and why is it important in database design?**

Ans 🡪

Normalization is the process of structuring database tables to reduce redundancy and improve data integrity. **Benefits**:

* Reduces duplication.
* Improves query performance.
* Enhances data consistency.
* Ensures scalability.

**Types**:

* **1NF**: Eliminate duplicate columns and ensure atomic data.
* **2NF**: Ensure no partial dependency on composite keys.
* **3NF**: Remove transitive dependencies.

**07.What are the different types of joins in SQL, and how do they differ?**

Ans 🡪

Joins combine data from multiple tables:

**INNER JOIN**: Returns matching rows (SELECT \* FROM A INNER JOIN B ON A.id = B.id;).

**LEFT JOIN**: All rows from the left table, matching rows from the right.

**RIGHT JOIN**: All rows from the right table, matching rows from the left.

**FULL OUTER JOIN**: All rows from both tables, matching or not.

**CROSS JOIN**: Cartesian product of two tables.

**08. What is the purpose of the GROUP BY clause in SQL?**

Ans 🡪

The GROUP BY clause aggregates rows with the same values in specified columns.

**Example:**

**SELECT department, COUNT(\*)**

**FROM employees**

**GROUP BY department;**

**Purpose**:

* Summarize data.
* Combine with aggregate functions (e.g., COUNT, SUM).

**09. Can you explain the difference between primary key and foreign key?**

Ans 🡪

* **Primary Key**: Uniquely identifies a row in a table. Cannot be NULL or duplicate.
* **Foreign Key**: Establishes a relationship between two tables by referencing a primary key in another table.

**Example:**

**CREATE TABLE orders (**

**order\_id INT PRIMARY KEY,**

**customer\_id INT,**

**FOREIGN KEY (customer\_id) REFERENCES customers(id)**

**);**

**10. What are some common commands used in MySQL?**

Ans 🡪

**CREATE DATABASE**: Creates a new database.

**USE DATABASE**: Selects a database for operations.

**CREATE TABLE**: Creates a table.

**INSERT INTO**: Inserts data (INSERT INTO users (name, age) VALUES ('John', 25);).

**SELECT**: Fetches data (SELECT \* FROM employees;).

**UPDATE**: Modifies data (UPDATE users SET age = 30 WHERE name = 'John';).

**DELETE**: Removes data (DELETE FROM users WHERE age < 18;).