SQL Fundamentals for Interview Preparation

1. Introduction to Databases

- A database is an organized collection of data that can be easily accessed, managed, and updated.
- Relational Databases: Data is organized into tables. Examples: MySQL, PostgreSQL.
- Non-Relational Databases: No fixed schema, used for large-scale and unstructured data. Examples: MongoDB.

2. SQL (Structured Query Language)

- SQL is a standardized programming language used to manage relational databases.
- SQL allows operations like querying data, inserting, updating, and deleting records.

Example:

SELECT * FROM employees;

3. SQL Keywords

- CREATE, DATABASE, TABLE, USE, SELECT, FROM, WHERE, INSERT INTO, UPDATE, DELETE, and more.

Example:

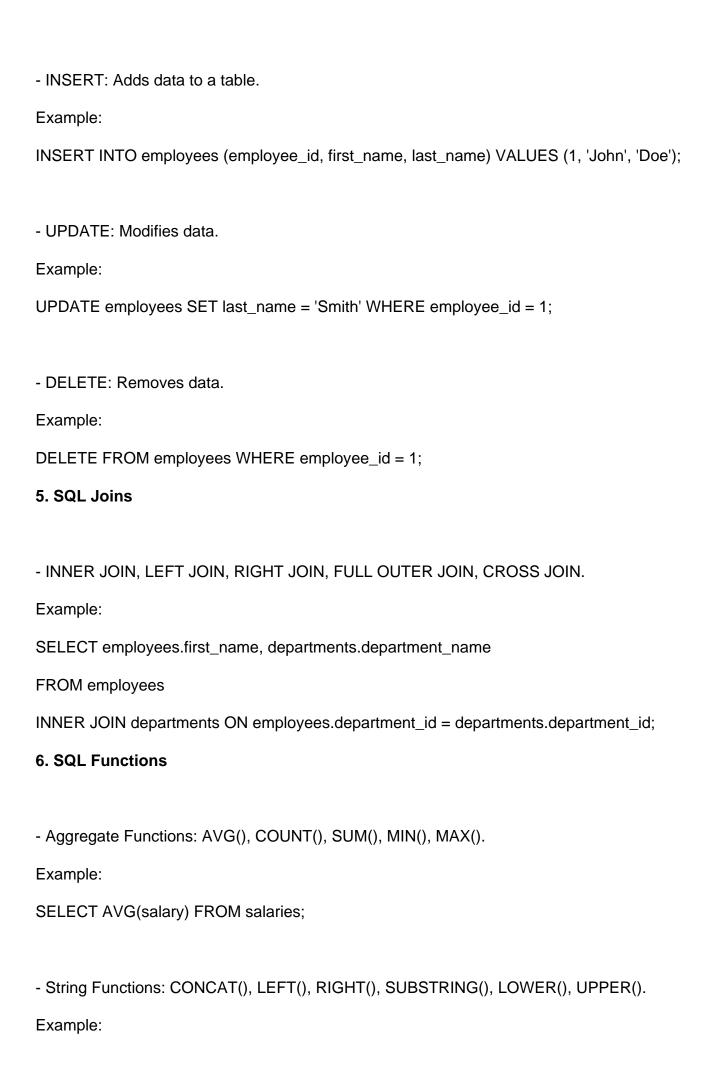
CREATE TABLE employees (employee_id INT PRIMARY KEY, first_name VARCHAR(50), last_name VARCHAR(50));

4. SQL Basic Operations

- SELECT: Retrieves data from a database.

Example:

SELECT first_name, last_name FROM employees;



SELECT CONCAT(first_name, ' ', last_name) AS full_name FROM employees;

7. SQL Keys

- Primary Key, Foreign Key, Unique Key, Super Key, Alternate Key, Candidate Key, Composite Key. Example:

CREATE TABLE orders (order_id INT PRIMARY KEY, employee_id INT, FOREIGN KEY (employee_id) REFERENCES employees(employee_id));

8. SQL Data Types

- CHAR, VARCHAR, INT, FLOAT, DATE, TIMESTAMP.

Example:

CREATE TABLE products (product_id INT, product_name VARCHAR(100), price FLOAT, manufacture_date DATE);

9. Relational vs. Non-Relational Databases

- Relational: Structured data, tables, SQL-based. Examples: MySQL, PostgreSQL.
- Non-Relational: Schema-less, handles unstructured data. Examples: MongoDB, Cassandra.

Example:

SQL (Relational):

SELECT * FROM customers;

NoSQL (Non-Relational - MongoDB-like syntax):

{ "_id": ObjectId("507f1f77bcf86cd799439011"), "firstName": "John", "orders": [{ "order_id": 1, "product": "Laptop"}] }