INTRODUCTION TO DBMS

1. **What is SQL, and why is it essential in database management?**

**SQL (Structured Query Language)** is the standard programming language used for managing and manipulating relational databases.

SQL is essential in many ways such as:

1. **Data Management**: SQL allows precise control over the data in a relational database, including retrieval, insertion, deletion, and updates.
2. **Standardization**: It is an ANSI and ISO standard, used across all major relational database systems (MySQL, PostgreSQL, SQL Server, Oracle, etc.).
3. **Data Integrity**: Supports rules and constraints to maintain accuracy and consistency in the data.
4. **Efficient Data Access**: SQL provides tools (like JOIN, GROUP BY, and WHERE) for complex and efficient querying.
5. **Scalability**: SQL is used in small-scale applications as well as large enterprise systems handling millions of records.
6. **Automation and Reporting**: SQL queries are widely used in generating reports and automating data tasks.

**2. Explain the difference between DBMS and RDBMS**

|  |  |  |
| --- | --- | --- |
| **Feature** | **DBMS (Database Management System)** | **RDBMS (Relational Database Management System)** |
| **Data Structure** | Stores data as **files** or **collections** of data | Stores data in **tables (rows and columns)** |
| **Relationships** | No relationships between data | Maintains **relationships** using **foreign keys** |
| **Data Integrity** | Does not enforce data integrity | Enforces **data integrity and constraints** |
| **Normalization** | Not typically supported | Supports **data normalization** to reduce redundancy |
| **Multi-user Support** | Limited or none | Designed to support **multiple users** simultaneously |
| **Examples** | File systems, XML, JSON storage | MySQL, PostgreSQL, Oracle, SQL Server, SQLite |
| **Rules Enforcement** | No strict rules for data consistency | Follows **ACID properties** (Atomicity, Consistency, Isolation, Durability) |
| **Security** | Basic or no security | Strong **user access controls and permissions** |

**3. Describe the role of SQL in managing relational databases.**

**1. Data Definition (DDL – Data Definition Language)**

Used to define and manage database structure.

Commands: CREATE, ALTER, DROP, TRUNCATE

**Example:**CREATE TABLE employees (

id INT, name VARCHAR(100), salary DECIMAL(10,2));

**2. Data Manipulation (DML – Data Manipulation Language)**

Used to insert, update, delete, and retrieve data.

Commands: INSERT, UPDATE, DELETE, SELECT

**Example:**SELECT \* FROM employees WHERE salary > 50000;

**3. Data Control (DCL – Data Control Language)**

Manages user access and permissions.

Commands: GRANT, REVOKE

**Example:**GRANT SELECT ON employees TO analyst;

**4. Transaction Control (TCL – Transaction Control Language)**

Controls data consistency and integrity during transactions.

Commands: COMMIT, ROLLBACK, SAVEPOINT

**Example:**BEGIN;

UPDATE employees SET salary = salary + 5000 WHERE id = 101;

COMMIT;

**4. What are the key features of SQL?**

**1. Data Querying**

Retrieve specific data using SELECT statements.

Supports conditions, filtering, and sorting.

**Example:**

SELECT name FROM employees WHERE salary > 50000;

**2. Data Definition**

* Create and modify database structure (tables, views, indexes).
* Commands: CREATE, ALTER, DROP

**Example:**

CREATE TABLE customers (id INT, name VARCHAR(100));

**3. Data Manipulation**

* Insert, update, and delete data records.
* Commands: INSERT, UPDATE, DELETE

**Example:**

INSERT INTO customers VALUES (1, ‘Alice');

**4. Data Control**

* Manage access permissions and security.
* Commands: GRANT, REVOKE

**Example:**

GRANT SELECT ON employees TO hr\_manager;

**5. Transaction Control**

* Ensure data integrity and consistency during multi-step operations.
* Commands: BEGIN, COMMIT, ROLLBACK

Example:

BEGIN;

UPDATE accounts SET balance = balance - 100 WHERE id = 1;

COMMIT;

**6. Relational Data Handling**

* Supports relationships between tables using **primary keys** and **foreign keys**.
* Enables **JOINs** to combine data from multiple tables.

**7. High-Level Abstraction**

* SQL allows users to focus on **what** data they want, not **how** it's retrieved.

**Example:**

SELECT COUNT(\*) FROM orders WHERE status = ‘delivered';

**8. Portability & Standardization**

* SQL is a **standardized language** (ANSI/ISO) supported by most RDBMSs like MySQL, PostgreSQL, SQL Server, and Oracle.

**9. Scalability & Performance**

* Can handle small and large volumes of data with indexing, optimization, and partitioning techniques.

**10. Extensibility**

* Many RDBMSs extend SQL with additional functions and procedural features (e.g., PL/SQL in Oracle, T-SQL in SQL Server).