THEOREY ASSIGNMENT MODULE - 4

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

Ans:- SQL (Structured Query Language) is a standard programming language specifically designed for managing and manipulating relational databases. It allows users to interact with the data stored in databases by performing a variety of operations such as:

● Querying data (e.g., SELECT statements)

● Inserting new records (e.g., INSERT INTO)

● Updating existing records (e.g., UPDATE)

● Deleting records (e.g., DELETE)

● Creating and modifying database structures (e.g., CREATE TABLE, ALTER, DROP)

Why SQL is Essential in Database Management:

1. Data Retrieval:- SQL enables users to retrieve specific data efficiently using powerful querying capabilities like filtering, sorting, and joining multiple tables.

2. Data Manipulation:- It allows you to modify data as needed, making it easy to update or remove outdated or incorrect information.

3. Data Definition:- SQL lets you define and structure databases, tables, and relationships between data elements (Data Definition Language - DDL).

4. Data Control:- With SQL, you can manage permissions and control access to sensitive data (Data Control Language - DCL), ensuring security and compliance.

5. Standardization:- SQL is a standardized language used across many database systems (e.g., MySQL, PostgreSQL, Microsoft SQL Server, Oracle), making your skills transferable across platforms.

6. Support for Large-Scale Data Operations:- SQL can handle complex queries on large datasets, which is crucial for data analytics, reporting, and business intelligence.

7. Integration with Other Tools:- SQL is often integrated with programming languages (like Python, Java, PHP) and data analysis tools, enabling seamless application development and data-driven decision-making.

2. Explain the difference between DBMS and RDBMS

Ans:-

DBMS (Database Management System)

A DBMS is software that allows users to create, store, and manage databases. It handles data as files and supports basic operations like insert, update, delete, and retrieve.

Key Features:

- ● Data is stored in a file format, not in tables.
- ● No relationships between data.
- ● May not enforce data integrity or ACID properties strictly.
- ● Suitable for small datasets and simple applications.

- Examples: Microsoft Access, File System, older versions of FoxPro.

- RDBMS (Relational Database Management System)

An RDBMS is an advanced type of DBMS based on the relational model. It stores data in tables (relations) and supports relationships between different data entities.

Key Features:

- Data is stored in tabular form (rows and columns).
- Supports relationships using foreign keys.
- Enforces data integrity (e.g., constraints like primary key, unique).
- Follows ACID properties (Atomicity, Consistency, Isolation, Durability) for transaction management.
- Supports SQL as the standard query language.
- Suitable for complex and large-scale applications.
- **Examples: MySQL, PostgreSQL, Oracle, SQL Server, SQLite.**

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

Ans:-

Role of SQL in Managing Relational Databases

SQL (Structured Query Language) plays a central role in managing relational databases. It provides the tools needed to create, read, update, and delete data (often abbreviated as CRUD) as well as manage database structures and access controls.

Key Roles of SQL:

1. Data Querying

● SQL allows users to retrieve specific information from one or more tables.

Example:

SELECT name, age FROM employees WHERE department = 'Sales';

2. Data Manipulation (DML)

● Modify the contents of the database using: ○ **INSERT – Add new data**

○ **UPDATE – Modify existing data**

○ **DELETE – Remove data**

Example:

UPDATE employees SET salary = 60000 WHERE id = 101;

3. Data Definition (DDL)

● Define and modify the database structure using: ○ **CREATE – Create tables, views, indexes**

○ **ALTER – Modify table structure**

○ **DROP – Delete tables or databases**

Example:

CREATE TABLE employees (

id INT PRIMARY KEY,

name VARCHAR(50),

department VARCHAR(50)

);

4. Data Control (DCL)

● Manage access to data using:

○ **GRANT – Give users access rights**

○ **REVOKE – Remove access rights**

Example:

GRANT SELECT ON employees TO john; 5. Transaction Control (TCL)

● Manage transactions to ensure data consistency and reliability: ○ **COMMIT – Save changes**

○ **ROLLBACK – Undo changes**

○ **SAVEPOINT – Set a point within a transaction**

Example: BEGIN;

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

UPDATE accounts SET balance = balance + 100 WHERE id = 2;

COMMIT;

4. What are the key features of SQL?

Ans:-

Key Features of SQL

1. Data Querying

● Allows users to retrieve specific data from one or more tables using the SELECT statement.

● Supports powerful filtering, sorting, and joining.

Example:

SELECT name, salary FROM employees WHERE department = 'HR'; 2. Data Manipulation (DML)

● Modify the contents of the database: ○ INSERT – Add new records

○ UPDATE – Modify existing records

○ DELETE – Remove records

Example:

INSERT INTO employees (id, name, department) VALUES (101, 'Alice', 'Sales'); 3. Data Definition (DDL)

● Define and manage database structure: ○ CREATE – Create tables, indexes, schemas

○ ALTER – Modify existing database objects

○ DROP – Delete objects

Example:

CREATE TABLE departments (

id INT PRIMARY KEY,

name VARCHAR(50)

); 4. Data Control (DCL)

● Manage permissions and access control: ○ GRANT – Give privileges to users

○ REVOKE – Remove privileges

Example:

```
GRANT SELECT, INSERT ON employees TO user1;
```

5. Transaction Control (TCL)

● Ensure data integrity and consistency:

● BEGIN, COMMIT, ROLLBACK, SAVEPOINT

Example:

```
BEGIN;
UPDATE accounts SET balance = balance - 500 WHERE id = 1;
UPDATE accounts SET balance = balance + 500 WHERE id = 2;
COMMIT;
```

6. Relational Data Management

● SQL manages data in tables with relationships via foreign keys, enabling structured and normalized storage.

7. Standardized Language

● SQL is governed by ANSI/ISO standards, meaning it's supported (with minor variations) by all major RDBMS systems: MySQL, PostgreSQL, Oracle, SQL Server, etc.

8. Scalability and Flexibility

● Handles small to enterprise-scale databases.

● Can be integrated into applications via APIs or embedded within other programming languages (like Python, Java, PHP).

9. Functions and Expressions

● Supports built-in functions for string manipulation, mathematics, dates, aggregates (SUM(),AVG(), etc.).

10. Security

● SQL provides user authentication, role-based access, and row-level security mechanisms.

5. What are the basic components of SQL syntax?

Ans:-

1. Statements

SQL is made up of statements that perform specific tasks, such as retrieving or modifying data.

● Example statements:

○ SELECT – retrieves data

○ INSERT – adds data

○ UPDATE – changes data

○ DELETE – removes data

2. Clauses

Clauses are the building blocks of SQL statements. Common clauses	Purpose	Example

include: Clause		
SELECT	Specifies columns to retrieve	SELECT name, age
FROM	Indicates the table to query from	FROM employees
WHERE	Filters rows based on a condition	WHERE age > 30