**Module-4) Se - Introduction To Dbms**

**1. What is SQL, and why is it essential in database management?**  
**Answer:**  
SQL (Structured Query Language) is a standardized programming language used to manage, manipulate, and query relational databases. It is essential in database management because it provides a systematic approach for accessing and managing data efficiently, performing CRUD operations (Create, Read, Update, Delete), and enforcing data integrity through constraints.

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

* **DBMS (Database Management System):**  
  A software system used to store, retrieve, and manage data in databases. It does not necessarily enforce relationships between tables and lacks advanced features like constraints. Examples: MS Access, XML.
* **RDBMS (Relational Database Management System):**  
  A type of DBMS that uses a relational model, storing data in tables with rows and columns. It enforces relationships between tables using primary and foreign keys and supports SQL. Examples: MySQL, PostgreSQL, Oracle.

**3. Describe the role of SQL in managing relational databases.**  
**Answer:**  
SQL is the primary tool for interacting with relational databases. Its roles include:

* Defining database structures with commands like CREATE, ALTER, and DROP.
* Manipulating data with INSERT, UPDATE, DELETE, and SELECT.
* Querying data efficiently using WHERE, GROUP BY, ORDER BY, and joins.
* Enforcing data integrity through constraints like PRIMARY KEY, FOREIGN KEY, and UNIQUE.
* Controlling database access with GRANT and REVOKE.

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

* **Data Querying:** Retrieve data using SELECT.
* **Data Manipulation:** Perform operations like INSERT, UPDATE, and DELETE.

**Data Definition:** Create and modify database structures using CREATE, ALTER, and DROP.

* **Data Integrity:** Enforce rules with constraints like NOT NULL, UNIQUE, and CHECK.
* **Transaction Control:** Manage transactions with COMMIT, ROLLBACK, and SAVEPOINT.
* **Flexibility:** Supports complex queries and joins.
* **Standardization:** Compatible with most RDBMS systems.

**LAB EXERCISES**

**Lab 1:**  
**Task:** Create a new database named school\_db and a table called students with the following columns: student\_id, student\_name, age, class, and address.  
**SQL Code:**

sql

Copy code

-- Create the database

CREATE DATABASE school\_db;

-- Use the created database

USE school\_db;

-- Create the students table

CREATE TABLE students (

student\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_name VARCHAR(100) NOT NULL,

age INT,

class VARCHAR(50),

address TEXT

);

**Lab 2:**  
**Task:** Insert five records into the students table and retrieve all records using the SELECT statement.  
**SQL Code:**

sql

Copy code

-- Insert records into the students table

INSERT INTO students (student\_name, age, class, address) VALUES

('John Doe', 14, '8th Grade', '123 Elm Street'),

('Jane Smith', 13, '7th Grade', '456 Oak Avenue'),

('Robert Brown', 15, '9th Grade', '789 Pine Road'),

('Emily Davis', 14, '8th Grade', '321 Maple Lane'),

('Michael Johnson', 16, '10th Grade', '654 Cedar Drive');

-- Retrieve all records from the students table

SELECT \* FROM students;