Module 4 – Introduction to DBMS

1. Introduction to SQL

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

SQL (Structured Query Language) is the standard language for relational databases. It is essential because it allows defining structures, inserting, updating, deleting, and querying data.

Explain the difference between DBMS and RDBMS.

DBMS manages data as files without enforced relationships. RDBMS stores data in relational tables, supports keys, and enforces integrity rules.

Describe the role of SQL in managing relational databases.

SQL defines schemas, manipulates data, queries information, and manages transactions in relational databases.

What are the key features of SQL?

Data definition, data manipulation, querying, transaction control, constraints, joins, security, and portability.

Lab Exercises

```
CREATE DATABASE school_db;
```

```
CREATE TABLE students(student_id INT PRIMARY KEY, student_name
VARCHAR(50), age INT, class VARCHAR(20), address VARCHAR(100));
```

```
INSERT INTO students VALUES (1,'Amit',12,'6th','Delhi'), (2,'Priya',11,'5th','Mumbai'),
(3,'Rahul',13,'7th','Chennai'), (4,'Sneha',10,'4th','Kolkata'), (5,'Vikram',14,'8th','Pune');
```

```
SELECT * FROM students;
```

2. SQL Syntax

What are the basic components of SQL syntax?

Keywords, identifiers, operators, clauses, expressions, and statements.

Write the general structure of an SQL SELECT statement.

```
SELECT column1, column2 FROM table_name WHERE condition ORDER BY column;
```

Explain the role of clauses in SQL statements.

Clauses refine SQL statements, e.g., WHERE filters, ORDER BY sorts, GROUP BY groups, HAVING filters groups.

Lab Exercises

```
SELECT student_name, age FROM students;
```

```
SELECT * FROM students WHERE age > 10;
```

3. SQL Constraints

What are constraints in SQL? List and explain the different types of constraints.

Constraints enforce rules on data. Types: PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, CHECK, DEFAULT.

How do PRIMARY KEY and FOREIGN KEY constraints differ?

PRIMARY KEY uniquely identifies each row. FOREIGN KEY references another table’s key to enforce relationships.

What is the role of NOT NULL and UNIQUE constraints?

NOT NULL ensures a column has no NULL values. UNIQUE ensures all values in a column are distinct.

Lab Exercises

```
CREATE TABLE teachers(teacher_id INT PRIMARY KEY, teacher_name
VARCHAR(50) NOT NULL, subject VARCHAR(50) NOT NULL, email
VARCHAR(100) UNIQUE);
```

```
CREATE TABLE students (student_id INT PRIMARY KEY,student_name
VARCHAR(100) NOT NULL,class VARCHAR(50),teacher_id INT,FOREIGN KEY
(teacher_id) REFERENCES teachers(teacher_id));
```

4. Main SQL Commands and Sub-commands (DDL)

Define the SQL Data Definition Language (DDL).

DDL defines and manages database objects like tables and schemas.

Explain the CREATE command and its syntax.

CREATE defines new databases or tables. Syntax: CREATE TABLE table_name (col datatype,...);

What is the purpose of specifying data types and constraints during table creation?

It ensures correct storage, integrity, and efficiency.

Lab Exercises

```
CREATE TABLE courses(course_id INT PRIMARY KEY, course_name
VARCHAR(50), course_credits INT);
```

```
CREATE DATABASE university_db;
```

5. ALTER Command

What is the use of the ALTER command in SQL?

ALTER modifies an existing table structure.

How can you add, modify, and drop columns from a table using ALTER?

ADD introduces new columns, MODIFY changes type, DROP removes columns.

Lab Exercises

```
ALTER TABLE courses ADD course_duration VARCHAR(20);

ALTER TABLE courses DROP COLUMN course_credits;
```

6. DROP Command

What is the function of the DROP command in SQL?

DROP deletes database objects permanently.

What are the implications of dropping a table from a database?

The table and its data are removed permanently.

Lab Exercises

```
USE school_db;

DROP TABLE teachers;

DROP TABLE students;

SHOW TABLES;
```

7. Data Manipulation Language (DML)

Define the INSERT, UPDATE, and DELETE commands in SQL.

INSERT adds rows, UPDATE modifies rows, DELETE removes rows.

What is the importance of the WHERE clause in UPDATE and DELETE operations?

WHERE limits changes to specific rows; without it, all rows are affected.

Lab Exercises

```
INSERT INTO courses VALUES(1,'Math',3);
INSERT INTO courses VALUES(2,'Science',4);

INSERT INTO courses VALUES(3,'English',2);

UPDATE courses SET course_duration='6 months' WHERE course_id=1;

DELETE FROM courses WHERE course_id=3;
```

8. Data Query Language (DQL)

What is the SELECT statement, and how is it used to query data?

SELECT retrieves rows from tables.

Explain the use of the ORDER BY and WHERE clauses in SQL queries.

WHERE filters rows; ORDER BY sorts results.

Lab Exercises

```
SELECT * FROM courses;

SELECT * FROM courses ORDER BY course_duration DESC;

SELECT * FROM courses LIMIT 2;
```

9. Data Control Language (DCL)

What is the purpose of GRANT and REVOKE in SQL?

GRANT assigns privileges, REVOKE removes them.

How do you manage privileges using these commands?

Administrators control access rights through GRANT and REVOKE.

Lab Exercises

```
CREATE USER 'user1' IDENTIFIED BY 'pass1';

CREATE USER 'user2' IDENTIFIED BY 'pass2';
```

```
GRANT SELECT ON courses TO user1;

REVOKE INSERT ON courses FROM user1;

GRANT INSERT ON courses TO user2;
```

10. Transaction Control Language (TCL)

What is the purpose of the COMMIT and ROLLBACK commands in SQL?

COMMIT saves changes permanently. ROLLBACK undoes uncommitted changes.

Explain how transactions are managed in SQL databases.

Transactions ensure ACID properties: Atomicity, Consistency, Isolation, Durability.

Lab Exercises

```
START TRANSACTION;

INSERT INTO courses VALUES(4,'History',3); COMMIT;

START TRANSACTION;

INSERT INTO courses VALUES(5,'Biology',4); ROLLBACK;

START TRANSACTION;

SAVEPOINT sp1;

UPDATE courses SET course_duration='12 months' WHERE course_id=2;

ROLLBACK TO sp1;

COMMIT;
```

11. SQL Joins

Explain the concept of JOIN in SQL. What is the difference between INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN?

JOIN merges rows from tables using common keys. INNER returns matches, LEFT keeps all left rows, RIGHT keeps all right rows, FULL keeps all rows.

How are joins used to combine data from multiple tables?

By linking common columns across tables.

Lab Exercises

```
CREATE TABLE departments (dept_id INT PRIMARY KEY,dept_name VARCHAR(50) NOT NULL);

CREATE TABLE employees (emp_id INT PRIMARY KEY,emp_name VARCHAR(50) NOT NULL,dept_id INT,FOREIGN KEY (dept_id) REFERENCES departments(dept_id));

SELECT e.emp_name, d.dept_name FROM employees e INNER JOIN departments d

ON e.dept_id = d.dept_id;

SELECT d.dept_name, e.emp_name FROM departments d LEFT JOIN employees e

ON e.dept_id = d.dept_id;
```

12. SQL Group By

What is the GROUP BY clause in SQL? How is it used with aggregate functions?

GROUP BY groups rows and applies aggregates like COUNT, SUM, AVG.

Explain the difference between GROUP BY and ORDER BY.

GROUP BY groups data, ORDER BY sorts data.

Lab Exercises

```
SELECT dept_id, COUNT(*) FROM employees GROUP BY dept_id;

SELECT dept_id, AVG(salary) FROM employees GROUP BY dept_id;
```

13. SQL Stored Procedure

What is a stored procedure in SQL, and how does it differ from a standard SQL query?

A stored procedure is a precompiled SQL program. It differs from queries as it can accept parameters and be reused.

Explain the advantages of using stored procedures.

They improve performance, security, and reduce repetition.

Lab Exercises

```
DELIMITER $$

CREATE PROCEDURE GetEmployeesByDept(IN dept INT)

BEGIN

   SELECT * FROM employees  WHERE dept_id = dept;

END $$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE GetCourseDetails(IN cid INT)

BEGIN

   SELECT * FROM courses  WHERE course_id = cid;

END $$

DELIMITER ;
```

14. SQL View

What is a view in SQL, and how is it different from a table?

A view is a virtual table created from a query result. Unlike tables, it doesn’t store data.

Explain the advantages of using views in SQL databases.

Views simplify queries, provide abstraction, and enhance security.

Lab Exercises

```
CREATE VIEW emp_dept AS SELECT e.emp_name, d.dept_name FROM employees e JOIN departments d ON e.dept_id = d.dept_id;

CREATE OR REPLACE VIEW emp_dept AS SELECT e.emp_name, d.dept_name FROM employees e JOIN departments d ON e.dept_id = d.dept_id WHERE e.salary >= 50000;
```

15. SQL Triggers

What is a trigger in SQL? Describe its types and when they are used.

A trigger is an automatic action fired on INSERT, UPDATE, DELETE. Types: BEFORE, AFTER, INSTEAD OF.

Explain the difference between INSERT, UPDATE, and DELETE triggers.

INSERT triggers fire when rows are added, UPDATE when modified, DELETE when removed.

Lab Exercises

```
DELIMITER //

CREATE TRIGGER after_insert_emp AFTER INSERT ON employees FOR EACH ROW

BEGIN

  INSERT INTO log_table (action, log_time) VALUES ('New employee added', NOW());

END;//

DELIMITER ;

DELIMITER //

CREATE TRIGGER update_timestamp BEFORE UPDATE ON employees FOR EACH ROW

BEGIN

    SET NEW.last_modified = NOW();

END;

//DELIMITER ;
```