Module\_5 LAB EXERCISES

**Introduction to SQL:**

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

Ans:

2: Insert five records into the students table and retrieve all records using the SELECT statement.

Ans:

**SQL Syntax:**

1: Write SQL queries to retrieve specific columns (student\_name and age) from the students table.

Ans:

2: Write SQL queries to retrieve all students whose age is greater than 10.

Ans:

**SQL Constraints:**

1: Create a table teachers with the following columns: teacher\_id (Primary Key), teacher\_name (NOT NULL), subject (NOT NULL), and email (UNIQUE).

Ans:

2: Implement a FOREIGN KEY constraint to relate the teacher\_id from the teachers table with the students table.

Ans:

**Main SQL Commands and Sub-commands (DDL):**

1: Create a table courses with columns: course\_id, course\_name, and course\_credits. Set the course\_id as the primary key.

Ans:

2: Use the CREATE command to create a database university\_db.

Ans:

ALTER Command:

1: Modify the courses table by adding a column course\_duration using the ALTER command.

Ans:

2: Drop the course\_credits column from the courses table.

Ans:

DROP Command:

1: Drop the teachers table from the school\_db database.

Ans:

2: Drop the students table from the school\_db database and verify that the table has been removed.

Ans:

Data Manipulation Language (DML):

1: Insert three records into the courses table using the INSERT command.

Ans:

2: Update the course duration of a specific course using the UPDATE command.

Ans:

3: Delete a course with a specific course\_id from the courses table using the DELETE command.

Ans:

Data Query Language (DQL):

1: Retrieve all courses from the courses table using the SELECT statement.

Ans:

2: Sort the courses based on course\_duration in descending order using ORDER BY.

Ans:

3: Limit the results of the SELECT query to show only the top two courses using LIMIT.

Ans:

Data Control Language (DCL):

1: Create two new users user1 and user2 and grant user1 permission to SELECT from the courses table.

Ans:

2: Revoke the INSERT permission from user1 and give it to user2.

Ans:

Transaction Control Language (TCL):

1: Insert a few rows into the courses table and use COMMIT to save the changes.

Ans:

2: Insert additional rows, then use ROLLBACK to undo the last insert operation.

Ans:

3: Create a SAVEPOINT before updating the courses table, and use it to roll back specific changes.

Ans:

SQL Joins:

1: Create two tables: departments and employees. Perform an INNER JOIN to display employees along with their respective departments.

Ans:

2: Use a LEFT JOIN to show all departments, even those without employees.

Ans:

SQL Group By:

1: Group employees by department and count the number of employees in each department using GROUP BY.

Ans:

2: Use the AVG aggregate function to find the average salary of employees in each department.

Ans:

SQL Stored Procedure:

1: Write a stored procedure to retrieve all employees from the employees table based on department.

Ans:

2: Write a stored procedure that accepts course\_id as input and returns the course details.

Ans:

SQL View:

1: Create a view to show all employees along with their department names.

Ans:

2: Modify the view to exclude employees whose salaries are below $50,000.

Ans:

SQL Triggers:

1: Create a trigger to automatically log changes to the employees table when a new employee is added.

Ans:

2: Create a trigger to update the last\_modified timestamp whenever an employee record is updated.

Ans:

Introduction to PL/SQL:

1: Write a PL/SQL block to print the total number of employees from the employees table.

Ans:

2: Create a PL/SQL block that calculates the total sales from an orders table.

Ans:

PL/SQL Control Structures:

1: Write a PL/SQL block using an IF-THEN condition to check the department of an employee.

Ans:

2: Use a FOR LOOP to iterate through employee records and display their names.

Ans:

SQL Cursors:

1: Write a PL/SQL block using an explicit cursor to retrieve and display employee details.

Ans:

2: Create a cursor to retrieve all courses and display them one by one.

Ans:

Rollback and Commit Savepoint:

1: Perform a transaction where you create a savepoint, insert records, then rollback to the savepoint.

Ans:

2: Commit part of a transaction after using a savepoint and then rollback the remaining changes.

Ans: