



Exercise: 05

SQL Queries

22 – Jan – 2026

Observation (5 Marks)

1. Write the syntax with example for the following **SET OPERATIONS**:
 - a. UNION
 - b. UNION ALL
 - c. INTERSECT
 - d. EXCEPT/MINUS
2. Write the syntax with example for the following **JOIN operations**:
 - a. INNER JOIN
 - b. LEFT OUTER JOIN
3. What is a **Stored Function**? Write the general syntax for creating a function.
4. What is a **Stored Procedure**? Write the syntax for creating a procedure with input parameters.
5. What is a **Trigger**? Write the syntax for creating a trigger and mention its types.

Execution (15 Marks)

(Create the required tables such as STUDENT, DEPARTMENT, COURSE, ENROLLMENT, MARKS with appropriate attributes and constraints.)

1. Display the student IDs of students enrolled in **UG courses** and **PG courses** using **UNION**.
2. Display the student IDs of students enrolled in **both UG and PG courses** using **INTERSECT**.
3. Display the student IDs of students enrolled in **UG courses but not in PG courses** using **EXCEPT / MINUS**.
4. Display **student name and department name** using **INNER JOIN**.
5. Display **all students and their enrolled courses**, including students who are not enrolled in any course, using **LEFT JOIN**.
6. Display **course name and number of students enrolled** in each course using **JOIN** and **GROUP BY**.
7. Create a **FUNCTION** to calculate **Grade** based on marks:
 - Marks $\geq 90 \rightarrow A$
 - Marks $\geq 75 \rightarrow B$
 - Marks $\geq 60 \rightarrow C$
 - Else $\rightarrow D$
8. Use the function to display **student name, marks, and grade**.
9. Create a **PROCEDURE** that accepts **Department ID** as input and displays all students belonging to that department.
10. Execute the procedure for a given department ID.
11. Create a **BEFORE INSERT trigger** on the MARKS table to **prevent insertion of negative marks**.

12. Create an **AFTER INSERT trigger** on the MARKS table to automatically insert records into a **MARKS_LOG** table.
13. Create a table STUDENT_AUDIT to store (*Student_ID, Student_Name, Department_Name, Operation_Type, Operation_Date*).
14. Create an **AFTER INSERT trigger** on the STUDENT table that inserts records into STUDENT_AUDIT by **joining STUDENT and DEPARTMENT tables**.
15. Create an **AFTER DELETE trigger** on the STUDENT table that logs deleted student details along with department name into STUDENT_AUDIT using a **JOIN**.