**Lab 5: SQL Queries, View, Index, Functions, Procedures, and Triggers**

**Objective**

The objective of this lab is to enable students to practice SQL programming skills on the database designed in Lab 4. Students will develop SQL queries ranging from basic retrieval to advanced operations, and implement database functions, stored procedures, and triggers to enforce business rules and automate tasks within their schema.Additionally, students will practice SQL to define views that simplify data access.

**Tasks**

Each group is required to:

1. **Use the Database Designed in Lab 4**
   * All queries, functions, procedures, and triggers must be based on the group’s own database developed in Lab 4.
2. **Write SQL Queries**
   * **Basic queries:**
     + Select all records from a table.

example: Retrieve all books in the library

* + - Filter rows using WHERE conditions (at least 5 queries per table).

example: List members who joined after 2022-01-01

Show all books in the "Computer Science" category

* + - Sort results using ORDER BY (at least 5 queries per table).

Example: List the orders placed in the first quarter of 2025, with the results displayed in descending order by order date.

* + - Perform basic aggregations (COUNT, SUM, AVG, MAX, MIN) (at least 5 queries per table).
  + **Intermediate queries:**
    - Join multiple tables using INNER JOIN, LEFT JOIN, etc.(at least 10queries )
    - Use GROUP BY and HAVING. (at least 10 queries).
    - Apply subqueries in WHERE or FROM clauses. (at least 10 queries).
  + **Advanced queries:**
    - Nested subqueries(at least 5 queries per table)..
    - Use of EXISTS, IN, and ANY/ALL(at least 5 queries per table).
    - Queries involving set operations (UNION, INTERSECT, EXCEPT) (at least 5 queries per table).

1. **Create Functions**
   * Implement at least four **user-defined functions** (scalar or table-valued) relevant to the group’s database domain.
   * Example: a function to calculate discounts, student GPA, or employee age.
2. **Create Stored Procedures**
   * Write at least four **stored procedures** that performs a multi-step operation.
   * Example: inserting new records across multiple tables, updating related data, or generating a summary report.
3. **Create Triggers**
   * Define at least four **triggers** to enforce business rules or maintain consistency.
   * Example: prevent deletion of a parent record if child records exist, automatically log changes in an audit table, or update inventory counts after a sale.
4. Create View and Index:
   * Define at least two views that simplify complex queries.
   * Example: View for borrowed books with member details
   * Create at least two indexes to improve query performance.
     + One index should be on a single column (e.g., Member.Email).
     + One index should be a composite index (e.g., BORROW(MemberID, BookID)
5. **Document Results**
   * Include SQL code with explanations for each query, function, procedure, and trigger.
   * Provide screenshots of successful execution results.

**Submission Format**

* **File type:** Microsoft Word (.doc/.docx) or PDF (.pdf).
* **Report structure:**
  1. Cover page (Lab number, course name, group members, date).
  2. Objective.
  3. SQL queries (basic → intermediate → advanced).
  4. Functions with explanation and execution results.
  5. Procedures with explanation and execution results.
  6. Triggers with explanation and execution results.
  7. View, index with explanation and execution results.
  8. Conclusion and reflection.