

# Assignment 01

## Mini Project: Employee Management System

### Scenario:

Assume you are a database developer for **ABC IT Solutions**. Your task is to design a small employee management system. This will later support API endpoints like:

- GET /employees → List all employees
- GET /employee/:id → Get employee by ID
- POST /employee → Add a new employee
- PUT /employee/:id → Update employee data
- DELETE /employee/:id → Delete an employee

You only need to create database, **tables, constraints, triggers, and stored procedures** to support these operations.

### 1. Database and Tables

- Create a database called office\_db.
- Create the following tables:
  - employees (emp\_id, emp\_name, dept\_id, salary)
  - departments (dept\_id, dept\_name)
  - employee\_audit (audit\_id, emp\_id, action\_type, action\_time, details)

### 2. Constraints

- emp\_name must NOT be NULL.
- emp\_id should be the PRIMARY KEY.
- department in employees should reference dept\_id in departments.
- Add a UNIQUE constraint on a sample column (e.g., email).
- Add a DEFAULT value for a column (e.g., city = 'Colombo').

### 3. Triggers

- Create a trigger to prevent inserting negative salaries.
- Create triggers to log every INSERT, UPDATE, and DELETE on employees into employee\_audit.

#### 4. Stored Procedures

- Procedure to retrieve all employees.
- Procedure to get an employee by ID (IN parameter).
- Procedure to count employees by department (OUT parameter).
- Optional: procedure to add a new employee using a transaction that also logs the action.

#### 5. Testing

- Insert sample employees and departments.
- Test each trigger and procedure.
- Include a brief note explaining how your setup could support the API endpoints above.

#### 1. Submission Requirements

1. **SQL Scripts:** Submit all SQL scripts you wrote for:
  - Creating the database and tables
  - Adding constraints
  - Creating triggers
  - Creating stored procedures
2. **Testing Evidence:** Provide **screenshots** showing:
  - Tables created successfully
  - Sample records inserted
  - Triggers firing correctly (audit logs)
  - Stored procedures returning expected results
3. **Optional Database Dump:** You may also submit a **database dump (.sql file)** of office\_db as additional evidence.
4. **Short Explanation:** Include a few lines explaining how your SQL setup could support the endpoints described (GET, POST, PUT, DELETE).