MySQL Practice Coding Assignment

Title: Employee Rewards & Performance Management System

Duration: 45-50 Minutes

Difficulty Level: Intermediate

Focus Area: End-to-end SQL operations using MySQL

Scenario Background

You are a Database Developer at TechNova Pvt. Ltd., a mid-sized software company that wants to digitize its Employee Rewards and Performance Management system.
Your task is to design and query the MySQL database to help HR track employee details, their departments, projects, performance ratings, and monthly rewards.

User Stories and Tasks

User Story 1 – Database Setup (DDL)

As a developer, I need to create a normalized database schema to store employee, department, project, and reward data so that it can be queried efficiently later.

Tasks:

1. Create a database named TechNovaDB.
2. Create the following tables with appropriate data types, constraints, and normalization rules:
    - **Department(DeptID, DeptName, Location)**
    - **Employee(EmpID, EmpName, Gender, DOB, HireDate, DeptID)**

- **Project(ProjectID, ProjectName, DeptID, StartDate, EndDate)**
- **Performance(EmpID, ProjectID, Rating, ReviewDate)**
- **Reward(EmpID, RewardMonth, RewardAmount)**

3. Define primary keys, foreign keys, and unique constraints where applicable.
4. Create an index on EmpName and DeptID to optimize frequent lookups.

Concepts Covered: DDL, Normalization, Indexes

User Story 2 – Insert and Manage Data (DML)

As a developer, I should be able to add employee, department, and performance details and modify them when required.

Tasks:

1. Insert at least 5 records each into Department, Employee, Project, Performance, and Reward tables.
2. Update one employee's department.
3. Delete one reward record where the amount is less than 1000.

Concepts Covered: DML (INSERT, UPDATE, DELETE)

User Story 3 – Generate Insights (DQL, Aggregate and Date Functions)

As an HR analyst, I need to generate insights about employees and their performance using SQL queries.

Tasks:

1. Retrieve all employees who joined after 2019-01-01.
2. Find the average performance rating of employees in each department.
3. List employees with their age (use a date function).

4. Find the total rewards given in the current year.
5. Retrieve employees who have received rewards greater than 2000.

Concepts Covered: DQL, Aggregate Functions, Date Functions

User Story 4 – Advanced Queries (Joins and Subqueries)

As an HR manager, I want to view data combining multiple tables for performance review reports.

Tasks:

1. Display Employee Name, Department Name, Project Name, and Rating using appropriate joins.
2. Find the highest-rated employee in each department using a subquery.
3. List all employees who have not received any rewards using a subquery.

Concepts Covered: Joins, Subqueries

User Story 5 – Transaction Control and Optimization

As a developer, I should ensure data consistency while inserting or updating multiple tables.

Tasks:

1. Begin a transaction:
    - Insert a new employee.
    - Assign them to a department.
    - Add their performance record.
2. Rollback the transaction if any insert fails; otherwise, commit it.
3. Analyze a slow query (for example, joining 3-4 tables without an index). Then, re-run it using indexes and observe improvement using the EXPLAIN command.

Concepts Covered: TCL (COMMIT, ROLLBACK), Query Optimization Basics, Index Usage

---

Sample Data References

Department Table:

DeptID	DeptName	Location
101	IT	Bangalore
102	HR	Delhi
103	Finance	Mumbai

Employee Table:

EmpID	EmpName	Gender	DOB	HireDate	DeptID
1	Asha	F	1990-07-12	2018-06-10	101
2	Raj	M	1988-04-09	2020-03-22	102
3	Neha	F	1995-01-15	2021-08-05	101

Expected Deliverables

- SQL script file: TechNova_Assignment.sql
    - Includes DDL, DML, DQL, Joins, Subqueries, and TCL commands.
- Screenshots or logs of results for:
    - EXPLAIN before and after applying indexes.
    - Queries with aggregate functions and joins.

Self-Evaluation Checklist

Criteria	Self-Rating (✔/✘)	Notes
Database schema follows 3NF normalization		
Proper use of primary, foreign keys, and indexes		
Data inserted and manipulated using DML commands		
Used aggregate functions and date operations correctly		
Queries optimized using EXPLAIN and indexes		
Used JOINs and Subqueries effectively		
Demonstrated transaction control (commit/rollback)		
Script executes without syntax errors		

Bonus Challenge (Optional)

1. Create a view named `EmployeePerformanceView` combining Employee, Department, and Performance data.

2. Create a stored procedure named `GetTopPerformers(deptName)` that returns the top 3 employees in that department based on their performance rating.