Bachelor of Computer Applications (BCA)

MYSQL (SQL/PL-SQL) LAB WITH PROJECTS

(OBCASE309P24)

Self-Learning Material
(SEM III)

Jaipur National University
Centre for Distance and Online Education

Established by Government of Rajasthan
Approved by UGC under Sec 2(f) of UGC ACT 1956
&
NAAC A+ Accredited

TABLE OF CONTENTS

EXPERT COMMITTEE

Prof. Sunil Gupta
(Computer and Systems Sciences, JNU Jaipur)

Dr. Shalini Rajawat
(Computer and Systems Sciences, JNU Jaipur)

Dr. Satish Pandey
(Computer and Systems Sciences, JNU Jaipur)

COURSE COORDINATOR

Mrs. Rashmi Choudhary
(Computer and Systems Sciences, JNU Jaipur)

UNIT PREPARATION

Unit Writer(s)	Assisting & Proofreading	Unit Editor
Mrs. Rashmi Choudhary (Computer and Systems Sciences, JNU Jaipur)	Ms. Heena Shrimali (Computer and Systems Sciences, JNU Jaipur)	Mr. Ram Lal Yadav (Computer and Systems Sciences, JNU Jaipur)

Secretarial Assistance

Mr. Mukesh Sharma

COURSE INTRODUCTION

Welcome to the "MySQL (SQL/PL-SQL) Lab with Projects" course, where you will gain hands-on experience in managing and manipulating relational databases using MySQL, SQL, and PL/SQL. This course is designed to provide practical, real-world exposure to database management systems through a series of interactive lab sessions and projects.

In this course, you will explore MySQL, a leading open-source database system, and develop proficiency in SQL (Structured Query Language) and PL/SQL (Procedural Language/SQL). The lab-based format allows you to engage directly with the tools and techniques used to create, query, and manage databases. You will learn to construct and optimize SQL queries, design complex data structures, and implement efficient data operations.

The course emphasizes project-based learning, where you will work on a variety of practical projects that simulate real-world scenarios. These projects are designed to challenge you and help you apply the concepts learned in the lab. You will be tasked with designing databases, writing and debugging queries, and developing stored procedures and triggers to handle complex data operations.

Throughout the course, you will also focus on best practices for database design and performance optimization. You will gain experience in creating scalable database solutions, ensuring data integrity, and automating processes with PL/SQL. By applying these skills to real-world projects, you will build a portfolio that demonstrates your ability to manage and manipulate databases effectively.

By the end of the course, you will have a solid foundation in MySQL, SQL, and PL/SQL, complemented by practical experience from completing projects that showcase your skills. Whether you are aiming for a career in database administration, software development, or data analysis, this course will provide you with the hands-on experience and expertise needed to excel in the field of database technology.

Course Outcomes:

At the completion of the course, a student will be able to:

1. Learn to use key concepts related to SQL including DDL, DML, DCL and DTL commands.
2. Developing PL/SQL elements like Cursors, Procedures, functions, triggers.
3. Applying cursors, procedures, functions and triggers on various databases to perform different updating and manipulations in existing tables in the database.
4. Building of databases for different project applications.

Acknowledgements:

The content we have utilized is solely educational in nature. The copyright proprietors of the materials reproduced in this book have been tracked down as much as possible. The editors apologize for any violation that may have happened, and they will be happy to rectify any such material in later versions of this book.

Question 1: Sports Team Management Database

Program Statement:

Design and implement a database to manage information about sports teams, individual players, game schedules, and player statistics. This system should allow for tracking of multiple teams across different sports.

Solution Hints:

- **Tables to Create:**
    - **Teams** with fields for **TeamID**, **TeamName**, **SportType**.
    - **Players** with fields for **PlayerID**, **Name**, **TeamID**, **Position**, **Stats**.
    - **Games** with fields for **GameID**, **Date**, **Location**, **TeamAID**, **TeamBID**.
    - **Statistics** with fields for **StatsID**, **PlayerID**, **GameID**, **Score**, **Assists**, **Rebounds** (or other sport-specific stats).
- **Key Operations:**
    - Design queries to retrieve player statistics for specific games or seasons.
    - Create aggregate functions to calculate total or average statistics across games.
- **Advanced Features:**
    - Develop views that provide summaries of team performance, player rankings, and upcoming game schedules.
    - Use triggers to automatically update player statistics after each game entry.

Question 2: Inventory Control System

Program Statement:

Construct an inventory management system for a hardware store to track products, suppliers, orders, and sales transactions.

Solution Hints:

- **Tables to Create:**

    - **Products** with fields for **ProductID**, **ProductName**, **Price**, **QuantityInStock**.
    - **Suppliers** with fields for **SupplierID**, **SupplierName**, **ContactInfo**.
    - **Orders** with fields for **OrderID**, **ProductID**, **SupplierID**, **Quantity**, **OrderDate**.
    - **Sales** with fields for **SaleID**, **ProductID**, **QuantitySold**, **SaleDate**.
- **Key Operations:**
    - Implement procedures for ordering products when stock levels fall below a specified threshold.
    - Develop reports to show sales trends and inventory needs.
- **Advanced Features:**
    - Set triggers on the **Sales** table to automatically decrement **QuantityInStock** in the **Products** table upon sale.
    - Create dashboard views for quick inventory and sales summaries.

Question 3: Airline Reservation System

Program Statement:

Develop a database schema for an airline reservation system to manage flight bookings, passenger details, and flight statuses, ensuring efficient handling of high volumes of concurrent transactions.

Solution Hints:

- **Tables to Create:**
    - **Flights** with fields for **FlightID**, **Airline**, **Origin**, **Destination**, **FlightTime**.
    - **Customers** with fields for **CustomerID**, **Name**, **Email**, **PhoneNumber**.
    - **Bookings** with fields for **BookingID**, **FlightID**, **CustomerID**, **BookingDate**, **Status**.
    - **FlightStatus** with fields for **StatusID**, **FlightID**, **Status**, **UpdateTime**.

- **Key Operations:**
    - Write queries to book flights, update flight statuses, and handle customer inquiries about flight details.
    - Ensure transactions are handled correctly to avoid booking conflicts and maintain data consistency.
- **Advanced Features:**
    - Implement stored procedures for common booking transactions and status updates.
    - Use triggers to log changes in flight status or send notifications to customers about flight changes.

Question 4: Academic Conference Management

Program Statement:

Design and implement a database system to manage paper submissions, review assignments, and registration for an academic conference, facilitating easy access to information for attendees and organizers.

Solution Hints:

- **Tables to Create:**
    - **Papers** with fields for **PaperID**, **Title**, **AuthorID**, **SubmissionDate**, **Status**.
    - **Reviews** with fields for **ReviewID**, **PaperID**, **ReviewerID**, **Score**, **Comments**.
    - **Participants** with fields for **ParticipantID**, **Name**, **Affiliation**, **Email**.
    - **Registrations** with fields for **RegistrationID**, **ParticipantID**, **ConferenceID**, **RegistrationDate**.
- **Key Operations:**
    - Develop a system for assigning papers to reviewers and collecting feedback.

    - Generate reports on paper statuses, review outcomes, and participant registrations.
- **Advanced Features:**
    - Create views to streamline the access to papers under review and finalized schedules.
    - Use triggers and transactions to ensure that review assignments are balanced and conflict-free.

Question 5: Online Art Gallery

Program Statement:

Implement a database for managing art sales online, including features to handle artist information, artwork details, sales transactions, and customer interactions.

Solution Hints:

- **Tables to Create:**
    - **Artists** with fields for **ArtistID**, **Name**, **Bio**, **Style**.
    - **Artworks** with fields for **ArtworkID**, **ArtistID**, **Title**, **Year**, **Price**, **Status**.
    - **Customers** with fields for **CustomerID**, **Name**, **Email**, **Address**.
    - **Sales** with fields for **SaleID**, **ArtworkID**, **CustomerID**, **SaleDate**, **Amount**.
- **Key Operations:**
    - Query artworks by artist, style, or availability.
    - Manage sales transactions and update artwork statuses upon purchase.
- **Advanced Features:**
    - Set up procedures for processing payments and managing customer inquiries.
    - Utilize triggers to update artist earnings and send notifications upon artwork sales.

Question 6: Movie Rental Service

Program Statement:

Create a database for a movie rental service that tracks movies, customer rentals, and late fees.

Solution Hints:

- **Tables to Create:Movies**, **Customers**, **Rentals**.
- **Key Operations:** Use date functions to calculate due dates and late fees. Implement complex queries to recommend movies based on customer history.
- **Advanced Features:** Develop triggers to apply late fees to customer accounts automatically.

Question 7: Event Ticketing System

Program Statement:

Develop a database to manage event tickets, venues, and attendee information.

Solution Hints:

- **Tables to Create:Events**, **Tickets**, **Venues**, **Attendees**.
- **Key Operations:** Design a seating allocation system that updates in real-time as tickets are sold. Use aggregation to manage venue capacities.
- **Advanced Features:** Create stored procedures for batch processing of ticket sales and cancellations.

Question 8: Restaurant Reservation System

Program Statement:

Design a system to manage table reservations, customer preferences, and dining histories for a restaurant.

Solution Hints:

- **Tables to Create:Tables**, **Reservations**, **Customers**, **DiningHistory**.
- **Key Operations:** Implement a reservation system that considers table availability and customer preferences. Use triggers to update tables' status.
- **Advanced Features:** Develop views for the staff to quickly see daily reservations and customer notes.

Question 9: Vehicle Maintenance Tracker

Program Statement:

Create a database to track vehicle maintenance schedules, service records, and parts inventory for a mechanic shop.

Solution Hints:

- **Tables to Create:Vehicles**, **Services**, **Parts**, **Inventory**.
- **Key Operations:** Use date functions to schedule regular maintenance. Manage inventory levels with automated alerts when stocks are low.
- **Advanced Features:** Set up procedures for ordering parts and scheduling services based on vehicle service history.

Question 10: Human Resources Management System

Program Statement:

Develop a database to manage employee records, salaries, and department assignments for a corporation.

Solution Hints:

- **Tables to Create:Employees**, **Salaries**, **Departments**, **Positions**.
- **Key Operations:** Implement complex SQL queries for salary reports and department budgets. Use foreign keys to maintain data integrity among tables.
- **Advanced Features:** Write triggers for automatic updates to salaries and promotions based on performance reviews.

Question 11: Banking Transactions Database

Program Statement:

Design a database to manage customer accounts, banking transactions, and branch information for a bank.

Solution Hints:

- **Tables to Create:**
    - **Customers** with fields like **CustomerID**, **Name**, **Address**, **PhoneNumber**.
    - **Accounts** with fields like **AccountID**, **CustomerID**, **BranchID**, **Balance**.
    - **Transactions** with fields like **TransactionID**, **AccountID**, **Type**, **Amount**, **TransactionDate**.
    - **Branches** with fields like **BranchID**, **BranchName**, **Location**.
- **Key Operations:**
    - Develop queries to handle deposits and withdrawals, ensuring transaction integrity with appropriate constraints.
    - Use aggregate functions to report on account balances and transaction volumes.
- **Advanced Features:**
    - Implement triggers to update account balances automatically after each transaction.
    - Create views that simplify user interfaces for bank tellers.

Question 12: Hotel Reservation System

Program Statement:

Create a database for managing hotel room bookings, guest information, and payment records.

Solution Hints:

- **Tables to Create:**
    - **Rooms** with fields like **RoomID**, **Type**, **Price**, **Status**.
    - **Guests** with fields like **GuestID**, **Name**, **Email**, **PhoneNumber**.
    - **Bookings** with fields like **BookingID**, **GuestID**, **RoomID**, **CheckInDate**, **CheckOutDate**, **PaymentStatus**.
- **Key Operations:**
    - Design queries to check room availability and book rooms.
    - Handle complex bookings involving multiple rooms or special requirements.
- **Advanced Features:**
    - Use stored procedures for booking transactions and payment processing.
    - Set triggers to automatically update room statuses upon check-in and check-out.

Question 13: School Management System

Program Statement:

Develop a database to manage student enrollments, teacher assignments, and class schedules in a school.

Solution Hints:

- **Tables to Create:**
    - **Students** with fields like **StudentID**, **Name**, **Grade**, **DateOfBirth**.
    - **Teachers** with fields like **TeacherID**, **Name**, **Subject**.
    - **Classes** with fields like **ClassID**, **Subject**, **TeacherID**, **Schedule**.
- **Key Operations:**
    - Create queries for scheduling classes, assigning teachers, and enrolling students.

    - Develop reports on student performance and teacher workloads.
- **Advanced Features:**
    - Implement triggers for maintaining class sizes within set limits.
    - Use views to provide teachers and parents easy access to student progress reports.

Question 14: E-commerce Order Tracking System

Program Statement:

Implement a database to track orders, customer details, and product inventory for an e-commerce platform.

Solution Hints:

- **Tables to Create:**
    - **Products** with fields like **ProductID**, **Name**, **Price**, **StockQuantity**.
    - **Customers** with fields like **CustomerID**, **Name**, **Email**, **Address**.
    - **Orders** with fields like **OrderID**, **CustomerID**, **OrderDate**, **Status**.
    - **OrderDetails** with fields like **OrderID**, **ProductID**, **Quantity**, **UnitPrice**.
- **Key Operations:**
    - Write queries to manage inventory, process orders, and update order statuses.
    - Analyze sales data to identify trends and inform inventory purchasing.
- **Advanced Features:**
    - Use procedures for order processing and stock management.
    - Set up triggers to alert customers and staff about order status changes.

Question 15: Public Library Catalog

Program Statement:

Create a database to manage a public library's book catalog, member registrations, and book loans.

Solution Hints:

- **Tables to Create:**
    - **Books** with fields like **BookID**, **Title**, **Author**, **ISBN**, **AvailableCopies**.
    - **Members** with fields like **MemberID**, **Name**, **MembershipDate**, **Email**.
    - **Loans** with fields like **LoanID**, **BookID**, **MemberID**, **LoanDate**, **DueDate**.
- **Key Operations:**
    - Develop queries for checking out books, returning them, and managing late fees.
    - Use reports to track popular books and active members.
- **Advanced Features:**
    - Implement triggers to update the count of available copies.
    - Create views for members to search for books and check loan statuses.

Question 16: Music Store Inventory and Sales

Program Statement:

Develop a database to manage a music store's inventory, sales transactions, and customer loyalty programs.

Solution Hints:

- **Tables to Create:**
    - **Instruments** with fields like **InstrumentID**, **Type**, **Brand**, **Price**, **InStock**.
    - **Customers** with fields like **CustomerID**, **Name**, **LoyaltyPoints**.

    - **Sales** with fields like **SaleID**, **InstrumentID**, **CustomerID**, **Date**, **Amount**.
- **Key Operations:**
    - Handle sales transactions and automatically update inventory and loyalty points.
    - Generate monthly sales reports and inventory restock orders.
- **Advanced Features:**
    - Use triggers to manage loyalty points and notify customers of special promotions.
    - Set up stored procedures for common sales processes and discount calculations.

Question 17: Car Rental Service

Program Statement:

Create a database to manage vehicle rentals, customer profiles, and rental agreements for a car rental service.

Solution Hints:

- **Tables to Create:**
    - **Vehicles** with fields like **VehicleID**, **Make**, **Model**, **Year**, **RentalRate**.
    - **Customers** with fields like **CustomerID**, **Name**, **DriverLicenseNumber**, **Phone**.
    - **Rentals** with fields like **RentalID**, **VehicleID**, **CustomerID**, **StartDate**, **EndDate**, **TotalCost**.
- **Key Operations:**
    - Design queries for vehicle availability, customer bookings, and pricing calculations.
    - Manage vehicle maintenance records and schedule service appointments.

- **Advanced Features:**
    - Implement triggers for automatic billing at the end of rental periods.
    - Create views for customers to view their rental histories and available vehicles.

Question 18: Fitness Club Membership Management

Program Statement:

Develop a database to manage memberships, class schedules, and instructor assignments for a fitness club.

Solution Hints:

- **Tables to Create:**
    - **Members** with fields like **MemberID**, **Name**, **MembershipType**, **StartDate**.
    - **Classes** with fields like **ClassID**, **ClassName**, **InstructorID**, **Schedule**.
    - **Instructors** with fields like **InstructorID**, **Name**, **Specialization**.
- **Key Operations:**
    - Schedule classes and link members to their chosen sessions.
    - Track attendance and membership renewals.
- **Advanced Features:**
    - Use procedures to automate membership renewals and class bookings.
    - Set up triggers to notify members and instructors of schedule changes.

Question 19: Corporate Event Planning

Program Statement:

Create a database to manage corporate events, including event scheduling, attendee registrations, and resource allocations.

Solution Hints:

- **Tables to Create:**
    - **Events** with fields like **EventID**, **Title**, **Date**, **Location**, **Budget**.
    - **Attendees** with fields like **AttendeeID**, **Name**, **Company**, **Email**.
    - **Resources** with fields like **ResourceID**, **Type**, **EventID**, **Quantity**.
- **Key Operations:**
    - Manage event logistics, track budgets, and handle attendee registrations.
    - Coordinate resource allocations, such as venues, catering, and equipment.
- **Advanced Features:**
    - Implement triggers for budget tracking and automatic updates to resource inventories.
    - Create views for event organizers to monitor event statuses and attendee lists.

Question 20: Wine Production and Sales Tracker

Program Statement:

Develop a database to track the production, aging, and sales of wine for a winery.

Solution Hints:

- **Tables to Create:**
    - **Wines** with fields like **WineID**, **Name**, **Vintage**, **Type**, **StockLevel**.
    - **Grapes** with fields like **GrapeID**, **Type**, **HarvestDate**, **Quantity**.
    - **Sales** with fields like **SaleID**, **WineID**, **QuantitySold**, **SaleDate**.
- **Key Operations:**
    - Manage grape harvests and track wine production cycles.
    - Handle sales transactions and stock level updates.
- **Advanced Features:**

- Use stored procedures for complex operations like production planning and sales forecasting.
- Implement triggers to update wine stock levels and notify sales teams of inventory changes.