



Department of Computer Science and Engineering,
Anna University, Chennai- 600025
CS23401 – Database Management System (R 2023)
End Sem Practical
Jan- April 2026
Year/Sem/Batch : II/IV/ P

26- Mar- 2026

Time: 2 Hours

Maximum Marks: 20

Instructions: Answer ANY 2 questions

Question 1: Multi-Vendor E-Commerce Platform (10 Marks)

An e-commerce company manages vendors selling products to customers.

Schema:

- Vendor(VendorID, VendorName, Rating)
- Customer(CustID, CustName, City)
- Product(ProdID, ProdName, Price, VendorID)
- Orders(OrderID, CustID, OrderDate)
- OrderDetails(OrderID, ProdID, Quantity)

Tasks:

(a) Create all tables with appropriate constraints, including ON DELETE CASCADE where necessary (2 Marks)

(b) Write optimized JOIN queries: (3 Marks)

- Display CustName, VendorName, total order value per order
- List vendors who have never sold any product (use OUTER JOIN + NULL filtering)

(c) Write a correlated nested query: (2 Marks)

- Find customers whose total purchase value is greater than the average purchase value of all customers

(d) Transaction Control: (3 Marks)

- Insert a new order with multiple items
- Demonstrate partial failure and use SAVEPOINT with ROLLBACK
- Create a trigger on OrderDetails such that If total order value for an order exceeds ₹1,00,000,

Question 2: Banking & Fraud Detection System (10 Marks)

The bank wants to track suspicious transactions.

Schema:

- Customer(CustID, Name, City)
- Account(AccNo, CustID, Balance, Type(savings/current))
- Transaction(TxnID, AccNo, Amount, TxnType(debit/credit), TxnDate)

Tasks:

(a) Create tables with constraints including CHECK (Balance ≥ 0) (2 Marks)

(b) Advanced FUNCTION queries: (3 Marks)

- Find top 3 accounts with highest transaction volume
- Calculate running balance using aggregation function. The running balance should be computed such that: 1. Deposits are treated as positive values, 2. Withdrawals are treated as negative values, 3. Balance is calculated incrementally for each transaction.
- Find customers with more than 3 transactions per day

(c) SET OPERATIONS: (2 Marks)

- Customers with savings accounts but no current accounts
- Customers appearing in both high-value transactions (>50,000) and frequent transactions (>5 per month)

(d) TRIGGER (complex logic): (3 Marks)

- Create a trigger to:
 - Block transactions > ₹1,00,000 if balance < ₹2,00,000
 - Log such attempts into a separate table Fraud_Log

Question 3: Smart Hospital Resource Optimization (10 Marks)

A hospital optimizes doctor workload and patient allocation.

Schema:

- Doctor(DocID, Name, Dept, Salary)
- Patient(PatID, Name, Disease)
- Appointment(AppID, DocID, PatID, Date, Fee)

Tasks:

(a) Create tables with constraints and default values (2 Marks)

(b) Complex JOIN queries: (3 Marks)

- Display doctor-wise revenue (sum of fees)
- Find doctors handling patients from multiple departments (self-join or grouping logic)

(c) Nested queries (multi-level): (2 Marks)

- Find doctors earning more than the average revenue of their department

(d) USER-DEFINED FUNCTION + TRIGGER: (3 Marks)

- Function to compute total revenue of a doctor
- Trigger to prevent booking if a doctor already has more than 10 appointments per day

Question 4: University Course Registration & Audit System (10 Marks)

University tracks student enrollments and academic performance.

Schema:

- Student(StudID, Name, Dept)
- Course(CourseID, CourseName, Credits)
- Enrollment(StudID, CourseID, Marks, Semester)

Tasks:

(a) Create tables with composite primary keys and constraints (2 Marks)

(b) Advanced JOIN + FUNCTION queries: (3 Marks)

- Calculate GPA of each student
- List toppers in each department

(c) SET OPERATIONS + Nested Queries: (3 Marks)

- Students enrolled in all courses offered by their department (DIVISION-like query)
- Students who failed in at least one subject but passed in another

(d) Transaction Control + Audit TRIGGER: (2 Marks)

- Update marks with COMMIT/ROLLBACK demonstration
- Trigger to log any update in marks into Audit_Log table