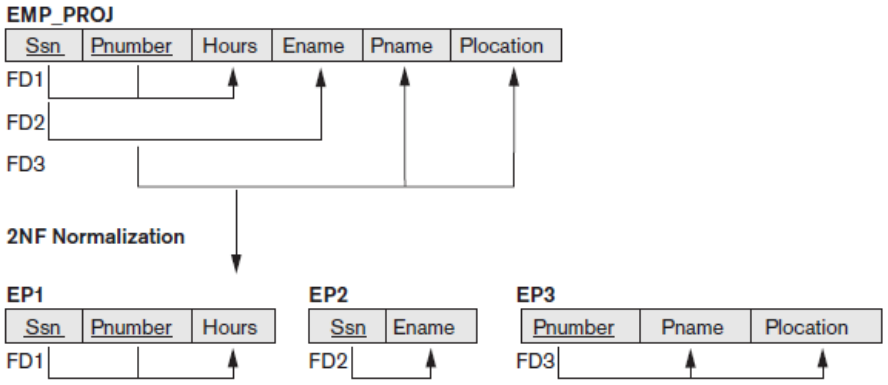
The National Institute of Engineering, Mysuru
Department of CS&E/ IS&E/ AI&ML

Course: Database Management Systems		Course Code: BCS403
Max Marks: 25	TEST - 2	Time:1hr (3:45pm - 4:45pm)
Date: 13/05/2025		Semester: 4

Scheme and Solution

Q.No.	Questions	Marks
1	Apply your understanding of Second Normal Form (2NF) by constructing and normalizing a database schema of your own. Clearly identify partial dependencies and explain how they are eliminated to achieve 2NF. A relation schema R is in second normal form (2NF) if every non-prime attribute A in R is fully functionally dependent on the primary key. R can be decomposed into 2NF relations via the process of 2NF normalization.[2M] ***Sample Schema [Any valid schema can be considered] [3M]***  	5
2	Given the relational schema below, write SQL queries to answer the following question: STUDENT(StudentID, StudentName, Department) COURSE(CourseID, CourseName, Department) ENROLLMENT(StudentID, CourseID, Grade) FACULTY(FacultyID, FacultyName, Department) TEACHES(FacultyID, CourseID) ***[Any valid query can be considered] each correct query 1Mark [5x1M]*** i) List all students along with the names of the courses they are enrolled in. SELECT S.StudentName, C.CourseName FROM STUDENT S JOIN ENROLLMENT E ON S.StudentID = E.StudentID JOIN COURSE C ON E.CourseID = C.CourseID; (or) SELECT S.StudentName, C.CourseName FROM STUDENT S, ENROLLMENT E, COURSE C WHERE S.StudentID = E.StudentID AND E.CourseID = C.CourseID;	5

	ii) Retrieve names of students who are not enrolled in any course. SELECT StudentName FROM STUDENT WHERE StudentID NOT IN (SELECT StudentID FROM ENROLLMENT); (or) SELECT S.StudentName FROM STUDENT S LEFT JOIN ENROLLMENT E ON S.StudentID = E.StudentID WHERE E.CourseID IS NULL; (or) SELECT StudentName FROM STUDENT MINUS SELECT S.StudentName FROM STUDENT S JOIN ENROLLMENT E ON S.StudentID = E.StudentID; iii) List the departments along with the number of courses they offer. SELECT Department, COUNT(*) AS NumberOfCourses FROM COURSE GROUP BY Department; iv) Find the names of students who have enrolled in more than 3 courses. SELECT S.StudentName FROM STUDENT S JOIN ENROLLMENT E ON S.StudentID = E.StudentID GROUP BY S.StudentID, S.StudentName HAVING COUNT(E.CourseID) > 3; v) Find the course with the highest number of enrollments. SELECT C.CourseName FROM COURSE C JOIN ENROLLMENT E ON C.CourseID = E.CourseID GROUP BY C.CourseID, C.CourseName ORDER BY COUNT(E.StudentID) DESC LIMIT 1;	
3	Consider the relational schema given in Question 2 and write SQL queries for the following: (i) Create a view that shows each student's name, the course they are enrolled in, and the corresponding grade. [2M] CREATE VIEW StudentCourseGrades AS SELECT S.StudentName, C.CourseName, E.Grade FROM STUDENT S, ENROLLMENT E, COURSE C WHERE S.StudentID = E.StudentID AND E.CourseID = C.CourseID; (ii) Assume that the table Enrollment_Log (LogID INT PRIMARY KEY, StudentID INT, CourseID INT, Date DATE) already exists. Create a trigger that inserts a new record into the Enrollment_Log table each time a new row is inserted into the ENROLLMENT table. [3M] Assumtion: The LogID column in the Enrollment_Log table is defined as an AUTO_INCREMENT primary key	5

	DELIMITER $$ CREATE TRIGGER t_LogEnrollment AFTER INSERT ON ENROLLMENT FOR EACH ROW BEGIN INSERT INTO Enrollment_Log (StudentID, CourseID, Date) VALUES (NEW.StudentID, NEW.CourseID, CURDATE()); END$$ DELIMITER ;	
4	Discuss the various reasons for a transaction to fail in the middle of execution. [any 5 reasons 1 Mark each] 1.A computer failure (system crash): A hardware, software, or network error occurs in the computer system during transaction execution. Hardware crashes are usually media failures—for example, main memory failure. 2. A transaction or system error: Some operation in the transaction may cause it to fail, such as integer overflow or division by zero. Transaction failure may also occur because of erroneous parameter values or because of a logical programming error. Additionally, the user may interrupt the transaction during its execution. 3. Local errors or exception conditions detected by the transaction:During transaction execution, certain conditions may occur that necessitate cancellation of the transaction. For example, data for the transaction may not be found. 4. Concurrency control enforcement:The concurrency control method may abort a transaction because it violates serializability. It may abort one or more transactions to resolve a state of deadlock among several transactions. 5. Disk failure: Some disk blocks may lose their data because of a read or write malfunction or because of a disk read/write head crash. This may happen during a read or a write operation of the transaction. 6. Physical problems and catastrophes:This refers to an endless list of problems that includes power or air-conditioning failure, fire, theft, sabotage, overwriting disks or tapes by mistake, and mounting of a wrong tape by the operator.	5
5	In a banking application, the ACCOUNT (AccountNumber, AccountHolderName, Balance) table is used to manage account information. During a fund transfer of ₹10,000 from Account A to Account B, the following steps are performed: 1. ₹10,000 is deducted from Account A 2. ₹10,000 is added to Account B However, a power failure occurs after the first step is completed, but before the second step. a) Describe the inconsistency that may occur due to this failure. b) Analyze the situation to identify which ACID properties are preserved or violated, and justify your conclusions with appropriate reasoning. a) Inconsistency Due to the Failure [2M] If a power failure occurs after the debit from Account A but before the credit to Account B, here's what happens: Inconsistency:	5

	₹10,000 is removed from Account A. ₹10,000 is not added to Account B. The total money in the system is reduced by ₹10,000, violating the basic principle of conservation of funds in banking. This leads to a data inconsistency, where the system reflects an inaccurate total balance. b) ACID Property Analysis[3M] Atomicity – Violated Atomicity requires that a transaction is all-or-nothing: either both the debit and credit happen, or neither does.Here, only the debit happened — the credit failed. Therefore, atomicity is violated, because the transaction was partially executed. Consistency – Violated Consistency ensures that a transaction brings the database from one valid state to another, preserving all rules. The failure leads to an inconsistent state, funds deducted but not credited. Hence, consistency is violated.	