Roll No.												

PRESIDENCY UNIVERSITY

BENGALURU

Make- Up Examinations – December 2025

Date: 30 – 12- 2025 **Time:** 09:30am – 12:30pm

School: SOCSE	**Program:** B. Tech.	
Course Code: CSE2074	**Course Name:** Database Management Systems	
Semester: MK	**Max Marks**: 100	**Weightage**: 50%

CO - Levels	CO1	CO2	CO3	CO4	CO5
Marks	26	26	24	24	-

Instructions:

(i) Read all questions carefully and answer accordingly.

(ii) Do not write anything on the question paper other than roll number.

Part A

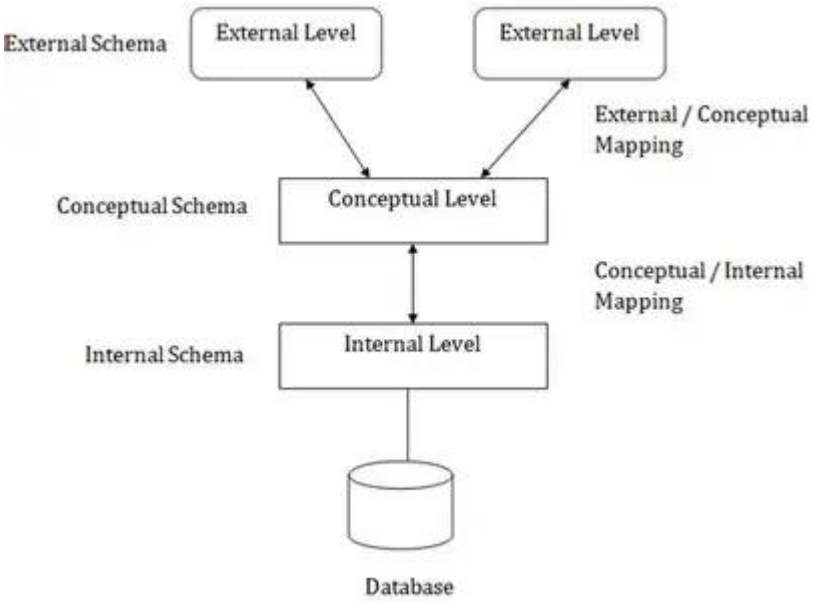
Answer ALL the Questions. Each question carries 2marks.				10Q x 2M=20M
1.	List the benefits of database approach.	**2 Marks**	**L1**	**CO1**
2.	List and explain the different types of database end users.	**2 Marks**	**L1**	**CO1**
3.	Analyze the output of the following statement? $\sigma_{subject = "database"}(Books)$ Retrieving all the database subject books	**2 Marks**	**L3**	**CO1**
4.	Explain various data types used in SQL.	**2 Marks**	**L1**	**CO2**
5.	What is Natural Join in SQL? Explain with a simple SQL query. A **Natural Join** in SQL is a type of join that automatically joins tables based on **columns with the same name and compatible data types** in both tables. It eliminates the need to specify the join condition explicitly. SELECT * FROM EMPLOYEE NATURAL JOIN DEPARTMENT;	**2 Marks**	**L2**	**CO2**
6.	What is Embedded SQL? Explain with a simple code snippet.	**2 Marks**	**L2**	**CO2**

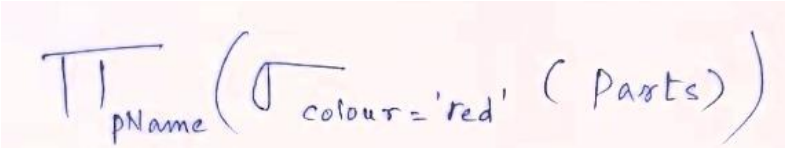
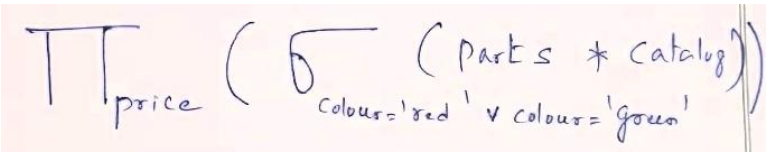
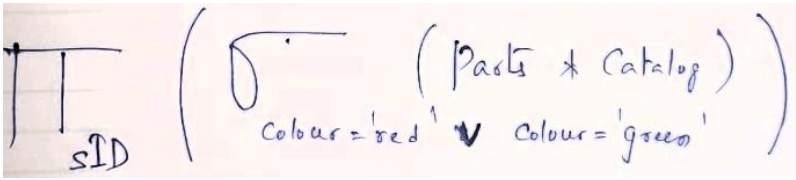
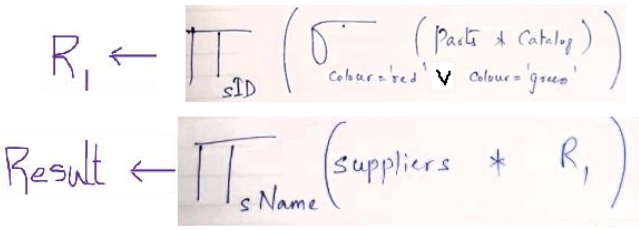
	Embedded SQL is SQL code that is written **within a host programming language** such as C, C++, Java, or Python. It allows the host program to **interact with a database** by embedding SQL statements directly in the program code. **Embedded SQL** bridges SQL and general-purpose programming languages, enabling database operations inside full applications. **Code snippet:** String url = "jdbc:mysql://localhost:3306/mydatabase"; Connection con = DriverManager.getConnection(url, "user", "password"); Statement st = con.createStatement(); String sql = "SELECT ENO, ENAME FROM EMPLOYEE"; ResultSet rs = st.executeQuery(sql);			
7.	What is Primary Key? Explain the desirable characteristics of it. A **Primary Key** is a field (or a combination of fields) in a database table that uniquely identifies each record (row) in that table. **Desirable Characteristics of a Primary Key** 1. **Uniqueness** ○ The primary key must contain **unique values** for each record to ensure each entry is distinct. 2. **Not Null** ○ A primary key **cannot contain null values** because it must always identify a record.	**2 Marks**	**L1**	**CO3**
8.	Define trivial and non-trivial functional dependencies. A functional dependency is **trivial** if the dependent attribute is a **subset** of the determinant. **Definition:** $A \rightarrow B$ is **trivial** if $B \subseteq A$ **Non-Trivial Functional Dependency** A functional dependency is **non-trivial** if the dependent attribute is **not** a subset of the determinant. **Definition:** $A \rightarrow B$ is **non-trivial** if $B \not\subset A$	**2 Marks**	**L1**	**CO3**
9.	What are concurrent transactions? Explain the need of concurrency control in brief. **Concurrent transactions** are multiple transactions that execute simultaneously (or overlap in time) in a multi-user database	**2 Marks**	**L1**	**CO4**

	environment. These transactions may access and manipulate the same data at the same time. Concurrency control is required to ensure the **integrity**, **consistency**, and **isolation** of transactions when they run concurrently. Without it, several problems can arise: **Problems Without Concurrency Control:** 1. **Lost Update** o When two transactions update the same data and one update overwrites the other. 2. **Dirty Read** o A transaction reads data that has been modified by another transaction but not yet committed. 3. **Non-repeatable Read** o A transaction reads the same data twice and gets different results because another transaction modified the data in between. 4. **Phantom Read** o A transaction reads a set of rows that satisfy a condition, but another transaction inserts new rows that also satisfy the condition before the first transaction finishes.			
10.	Write about SQL support for transactions. **BEGIN / START TRANSACTION** Starts a new transaction. **COMMIT** Saves all changes made during the transaction. COMMIT; **ROLLBACK** Undoes all changes made during the transaction. **SAVEPOINT** Sets a point within a transaction to which you can later roll back. **ROLLBACK TO SAVEPOINT** Rolls back to a previously defined savepoint. **SET AUTOCOMMIT** Enables or disables automatic commit after each SQL statement.	**2 Marks**	**L1**	**CO4**

Part B

Answer the Questions. **Total Marks 80M**

11.	**a.**	Explain the different types of DBMS users.	**5 Marks**	**L2**	**CO1**
	b.	Explain the three-schema Architecture. What is the difference between logical data independence and physical data independence?  Three-schema architecture provides clear data abstraction layers. Logical data independence protects applications from changes in the data model. Physical data independence protects the data model from changes in how data is physically stored. *(Block diagram with explanation: 3+2 = 5)*	**5 Marks**	**L2**	**CO1**
	c.	Explain in detail the components of DBMS environment with a block diagram.	**10 Marks**	**L2**	**CO1**
		Or			
12.	**a.**	Explain various unary operators used in Relational Algebra with proper examples. SELECT, PROJECT, RENAME operations in relation algebra. *(1.5 + 1.5 + 2 = 5 Marks)*	**5 Marks**	**L2**	**CO1**
	b.	Construct E-R diagram of the bank. It provides different kinds of bank accounts and loans. It operates number of branches.	**5 Marks**	**L2**	**CO1**

	c.	Consider the following relations Suppliers(sID, sName, address) Parts(pID, pName, colour) Catalog(sID, pID, price) Write the relational algebra for the following requirements: i. Find the names of all red parts.  ii. Find all prices for parts that are red or green.  iii. Find the sIDs of all suppliers who supply a part that is red or green  iv. Find the names of all suppliers who supply a part that is red or green. 	**10 Marks**	**L3**	**CO1**

13.	**a.**	What is a View? How can it be created and dropped? Demonstrate with simple SQL queries.	**5 Marks**	**L3**	**CO2**
	b.	Explain the following SQL commands/Keywords: **(i)** REVOKE: The REVOKE command removes user access rights or privileges to the database objects REVOKE privilege_name ON object_name FROM {user_name |PUBLIC |role_name} **(ii)** GRANT: SQL GRANT is a command used to provide access or privileges on the database objects to the users.	**5 Marks**	**L2**	**CO2**

		GRANT privilege_name ON object_name TO {user_name |PUBLIC |role_name}; **(iii)** COMMIT: The **COMMIT** statement in SQL is used to permanently save all changes made during the current transaction. It ensures that all modifications to the database are saved and cannot be undone unless explicitly reversed by another transaction. **(iv)** ROLLBACK: The **ROLLBACK** command is used to undo the changes made during the current transaction. It reverts the database to its previous state before the transaction began. This command is useful when an error occurs, or you decide not to save the changes made during the transaction. **(v)** DISTINCT: The DISTINCT keyword in SQL is used to return only unique values from a column or a combination of columns in a table. This is particularly useful when you want to eliminate duplicate entries from your result set. SELECT DISTINCT Country FROM Customers;			
	c.	Consider the following relations for a company database Application. EMPLOYEE (ENO, ENAME, SEX, DOB, DOJ, BASIC_PAY, DEPTNO) DEPARTMENT (DEPTNO, DNAME) PROJECT (PROJNO, PNAME, DEPTNO) WORKS_ON (ENO, PROJNO, HOURS) **i.** Calculate the numbers of male and female employees. **ii.** List the names of employees who are working for CSE department. **iii.** Count the number of employees who are working on JAVA project. **iv.** List the employees who are not working on any projects. **v.** Create a view to show the names of all the employees along with their department names.	**10 Marks**	**L3**	**CO2**
		Or			
14.	**a.**	What is a Database Trigger? Create a simple UPDATE trigger using SQL on a table EMPLOYEE.	**5 Marks**	**L3**	**CO2**
	b.	What are aggregate functions? List the aggregate functions supported by SQL.	**5 Marks**	**L3**	**CO2**
	c.	**Consider the relation schema given below:** EMPLOYEE (EMPNO, NAME, OFFICE, AGE) BOOK (ISBN, TITLE, AUTHOR, PUBLISHER)	**10 Marks**	**L3**	**CO2**

		LOAN (EMPNO, ISBN, DATE) **Implement the SQL queries for the following requirements.** **i.** Find the names of employees who have borrowed a book published by McGraw-Hill. **SELECT NAME** **FROM EMPLOYEE E, BOOK B, LOAN L** **WHERE PUBLISHER = 'McGraw-Hill'** **AND E.EMPNO = L.EMPNO;** **ii.** List the names of the employees who have not borrowed any books. **SELECT * FROM EMPLOYEE** **WHERE EMPNO NOT IN (SELECT EMPNO FROM LOAN);** **iii.** List the names of the employees who have borrowed more than two books. **SELECT NAME** **FROM EMPLOYEE** **WHERE EMPNO IN (SELECT EMPNO FROM LOAN** **HAVING COUNT(*) >= 2** **GROUP BY EMPNO);** **iv.** Retrieve the total number of books present in the library. SELECT COUNT(*) FROM BOOK; **v.** Generate a report containing EMPNO, NAME, ISBN, AUTHOR, PUBLISHER, and BORROWED DATE. **SELECT E.EMPNO, NAME, ISBN, AUTHOR, PUBLISHER, DATE** **FROM EMPLOYEE E, LOAN L, BOOK B** **WHERE B.ISBN = L.ISBN** **AND E.EMPNO = L.EMPNO;**			

15.	**a.**	Explain the informal design guidelines used in relational schema design.	**5 Marks**	**L1**	**CO3**
	b.	Illustrate with an example what is meant by partial functional dependency and describe how this type of dependency relates to 2NF.	**5 Marks**	**L2**	**CO3**
	c.	Construct a Database to illustrate BCNF (Boyce-Codd Normal Form). How does it different from 3NF? Explain with an example.	**10 Marks**	**L3**	**CO3**
		Or			
16.	**a.**	Explain the Armstrong's Rules of Inference**.**	**5 Marks**	**L2**	**CO3**
	b.	Given a relation R(A, B, C, D) and Functional Dependency set FD = { AB → CD, B → C }, determine whether the given R is in	**5 Marks**	**L3**	**CO3**

		2NF? If not convert it into 2 NF. Definition of 2NF: No non-prime attribute should be partially dependent on Candidate Key Since R has 4 attributes: - A, B, C, D, and Candidate Key is AB, Therefore, prime attributes (part of candidate key) are A and B while a non-prime attribute are C and D a) FD: AB → CD satisfies the definition of 2NF, that non-prime attribute (C and D) are fully dependent on candidate key AB b) FD: B → C does not satisfy the definition of 2NF, as a non-prime attribute(C) is partially dependent on candidate key AB(i.e. key should not be broken at any cost) As FD B → C, the above table R(A, B, C, D) is not in 2NF Convert the table R(A, B, C, D) in 2NF: Since FD: B → C, our table was not in 2NF, let's decompose the table R1(B, C) Since the key is AB, and from FD AB → CD, we can create R2(A, B, C, D) but this will again have a problem of partial dependency B → C, hence R2(A, B, D). Finally, the decomposed table which is in 2NF a) R1(B, C) b) R2(A, B, D)			
	c.	What is normalization? Design your own databases to illustrate 1NF, 2NF, and 3NF.	**10 Marks**	**L3**	**CO3**

17.	**a.**	With a neat Sketch explain the different states of a database transaction.	**5 Marks**	**L1**	**CO4**
	b.	Differentiate strict two-phase locking protocol and rigorous two-phase locking protocol.	**5 Marks**	**L2**	**CO4**
	c.	What are concurrent transactions? Why concurrency control is required? How is it implemented in DBMS? Explain in detail.	**10 Marks**	**L3**	**CO4**
Or					
18.	**a.**	List and explain the ACID properties of a database transaction.	**5 Marks**	**L1**	**CO4**
	b.	Demonstrate recoverable schedule with suitable example. In a Database Management System (DBMS), a recoverable	**5 Marks**	**L2**	**CO4**

		schedule ensures that if a transaction Tj reads a value written by transaction Ti, then Ti should commit before Tj commits. This property ensures the database can be recovered correctly in case of a crash. **Example of Recoverable Schedule:** Let's consider two transactions: • T1: Writes value of X • T2: Reads value of X written by T1 **Step-by-step:** 1. T1 writes to X → W1(X) 2. T2 reads X → R2(X) 3. T1 commits → C1 4. T2 commits → C2 In this schedule, since T2 reads X written by T1, it waits for T1 to commit before committing itself. Why is this recoverable? If the system crashes before T1 commits, T2 would not commit either. So we can recover safely — there are no inconsistent reads or uncommitted dependencies.			
	c.	Explain Time stamp ordering algorithm. Explain in detail the ARIES recovery method.	**10 Marks**	**L2**	**CO4**