



KAMALDNP

kamaldnp.com • IT Training, Varanasi

PDF created by Raj Kamal using AI

Model Practice Paper — Database Management System

BTEUP Diploma in Computer Science & Engineering • 3rd Semester • Session 2026-27
Session 2026-27

Practice Questions / Model Paper — NOT a BTEUP board paper

Time: 3 Hours

Max. Marks: 60

General Instructions:

- All questions are compulsory. Each question has three parts (a), (b) and (c) — **answer any two** of the three parts in every question.
- Each part carries 6 marks; each question carries 12 marks.
- Draw neat, labelled diagrams wherever they are asked or wherever they support your answer.
- Write your answers in your own words, in the same order as the questions.

About this paper: This is an original practice (model) paper prepared by KAMALDNP Academy from the official BTEUP 2026-27 syllabus topics for this subject. It is NOT a BTEUP board paper, and no question in it is promised to appear in your examination. In the verified 2026-27 scheme this theory paper carries 60 marks (external) in 3 hours, and this model paper is scaled to that maximum; its structure follows the board's last published paper pattern (5 questions, each with three parts of which any two are to be answered). The 2026-27 scheme's own paper format has not been published yet — when your board paper is in your hand, its printed instructions govern.

Q1. (From: Unit 1 — Introduction) — Answer any two of the following:

- (a) Describe database system concepts and the general architecture of a database system. State two advantages of using a DBMS over keeping separate data files. (6)
- (b) Explain the components of a DBMS in brief. (6)
- (c) Discuss the three views of data — external view, conceptual view and internal view — with the help of a diagram. What is meant by a schema definition? (6)

Q2. (From: Unit 2 — Data Model and Keys) — Answer any two of the following:

- (a) Draw and describe an E-R diagram for a small library with two entities: Member (MemberID, Name, Address) and Book (ISBN, Title, Author). A member can borrow many books, and a book can be borrowed by many members over time. Show the key attributes and state where you would place the attribute IssueDate. (6)
- (b) Differentiate between the hierarchical model, the network model and the E-R model. Give one advantage and one disadvantage of each model. (6)
- (c) Explain the concept of keys in a database: primary key, candidate key, super key and foreign key. Also differentiate between a strong entity set and a weak entity set, and state what constraints do. (6)

Q3. (From: Unit 3 — Relational Model) — Answer any two of the following:

- (a) Explain the relational data model. Describe the main relational database constraints in brief. (6)
- (b) State the rules used in mapping an E-R model to a relational model. Then map this case: Department (DeptID, DeptName) has many Employees; Employee (EmpID, EmpName, Salary) belongs to exactly one department. Write the resulting tables with their primary keys and foreign keys. (6)
- (c) Given Student (RollNo, Name, Branch, Marks), Enrolled (RollNo, CourseCode) and Course (CourseCode, CourseName), write relational algebra expressions to find: (i) the names of all CSE branch students; (ii) RollNo and Name of CSE students with Marks greater than 70; (iii) the course names taken by the student with RollNo 101. (6)

Q4. (From: Unit 4 — Relational Database Design) — Answer any two of the following:

- (a) Explain functional dependencies and normalisation for relational databases. Name the types of normalisation in order. (6)
- (b) A table is OrderData (OrderID, ProductID, ProductName, Quantity, ProductPrice), where one order contains many products. It is given that ProductID decides ProductName and ProductPrice, and the pair (OrderID, ProductID) decides Quantity. State why this table is not in 2NF and normalise it to 2NF, writing the resulting tables with their keys. (6)
- (c) A table in 2NF is Staff (EmpID, EmpName, DeptID, DeptName, DeptLocation), where EmpID decides DeptID, and DeptID decides DeptName and DeptLocation. Normalise it to 3NF, writing the resulting tables with their keys. Also state the condition a table must satisfy to be in BCNF. (6)

Q5. (From: Unit 5 — SQL / MySQL) — Answer any two of the following:

- (a) Name the commonly used MySQL data types for whole numbers, text, dates and decimal numbers. Then write the SQL data definition command to create a table Student with columns RollNo (integer, primary key), Name (text up to 50 characters, must not be empty), Branch (text up to 20 characters), Marks (integer) and AdmissionDate (date). **(6)**
- (b) For a table Employee (EmpID, EmpName, Salary, Dept), write SQL commands to: (i) insert a new employee 105, Asha Verma, salary 32000, department Sales; (ii) set the salary of employee 103 to 30000; (iii) delete the employee with EmpID 104; (iv) list the name and salary of employees whose salary is greater than 25000. **(6)**
- (c) Explain the arithmetic, comparison and logical operators used in SQL, with one symbol of each type. Then write an SQL query on the Employee table of part (b) to show, for each department, the number of employees and the average salary, keeping only departments whose average salary is greater than 28000 and showing the highest average first. **(6)**



Model Answer Outline — Database Management System

The key points a full-marks answer contains, question by question.

Practice Questions / Model Paper — NOT a BTEUP board paper

Q1(a)

- A database is an organised collection of related data; a DBMS is the software that creates, stores, retrieves and protects it, so users work through programs instead of handling files directly.
- Architecture idea: users and application programs at the top, query processor and storage manager in the DBMS in the middle, and the database files at the bottom.
- Advantages over separate files: controlled redundancy, data consistency, sharing of data, security and backup/recovery support.
- In the client-server form, many users send queries to one server that holds the database, so everyone sees the same current data.
- The schema (overall design) is separated from the data itself, so data can be reorganised without changing every program.

Q1(b)

- Hardware: the computer, disks and network on which the database runs.
- Software: the DBMS itself, the operating system, and the application programs that use the database.
- Data: the actual stored data plus the data dictionary (metadata) that describes tables, columns and constraints.
- Users: database administrator (DBA), designers, application programmers and end users; the DBA controls schema, access and recovery.
- Procedures: the rules and instructions for using, backing up and maintaining the database.

Q1(c)

- External (view) level: each user or group sees only the part of the data it needs, in its own view; this gives security and simplicity.
- Conceptual level: the whole logical design — all tables, their columns and relationships — written in the schema definition.
- Internal level: how the data is actually stored on disk — files, indexes and storage details.
- Schema definition states the structure (tables, data types, keys and constraints) that the DBMS then enforces on all data.
- The three levels give data independence: storage details can change without changing the conceptual design, and views can change without changing stored data.

Q2(a)

- Entities: Member (key MemberID, attributes Name and Address), Book (key ISBN, attributes Title and Author).
- Relationship: Member BORROWS Book. It is many-to-many (M:N) because one member can borrow many books and one book can be borrowed by many members over time.
- The relationship carries its own attribute IssueDate, because the date belongs to the borrowing event, not to the member alone or the book alone.
- Diagram description: rectangle for each entity, ellipse for each attribute with the key attribute underlined, and a diamond labelled BORROWS joining the two rectangles, marked M on the Member side and N on the Book side.
- Also state any assumption used, for example that only one copy of each ISBN is kept, so the design stays small and clear.

Q2(b)

- Hierarchical model: records form a tree with one root; each child has exactly one parent. Advantage: simple and fast for one-to-many data. Disadvantage: a record with two parents cannot be shown naturally, so data is duplicated.
- Network model: records form a graph and a child may have more than one parent through set types. Advantage: many-to-many links are possible. Disadvantage: structure and queries become complex.
- E-R model: a design model using entities, attributes and relationships, drawn as a diagram before any tables are made. Advantage: easy to discuss with users and to convert to tables. Disadvantage: it is a design tool, not a storage system by itself.
- Comparison line: hierarchical is the most rigid, network is more flexible but complex, and E-R is used at the design stage for any of the others.

Q2(c)

- Super key: any set of columns that can uniquely identify a row. Candidate key: a minimal super key (no column can be removed). Primary key: the candidate key chosen by the designer; it cannot be NULL and cannot repeat.
- Foreign key: a column in one table that refers to the primary key of another table; it enforces the link between the tables.
- Constraints are rules the DBMS enforces, such as NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY and CHECK, so wrong data is rejected at entry.
- Strong entity set: has its own primary key, for example Student with RollNo. Weak entity set: has no full key of its own and depends on a strong (owner) entity, for example a dependent of an employee.
- A weak entity is identified through the owner's key plus its own partial key (discriminator), and is drawn with a double rectangle in an E-R diagram.

Q3(a)

- In the relational model all data is stored in tables (relations) made of rows (tuples) and columns (attributes); each column has a fixed domain of allowed values.

- Each row is unique and is identified by a primary key; row and column order do not matter.
- Domain constraint: a value must belong to the column's domain, for example Marks between 0 and 100 and of integer type.
- Entity integrity: no part of a primary key may be NULL, because a row must always be identifiable.
- Referential integrity: a foreign key value must either match an existing primary key value in the referred table or be NULL where allowed; this stops orphan rows.

Q3(b)

- Rule for a 1:N relationship: put the primary key of the 'one' side as a foreign key in the table of the 'many' side.
- Department is the 'one' side and Employee the 'many' side, giving: Department (DeptID, DeptName) with DeptID as primary key.
- Employee (EmpID, EmpName, Salary, DeptID), with EmpID as primary key and DeptID as foreign key referring to Department.
- The department name is NOT repeated inside Employee; keeping it once in Department avoids update anomalies. A separate table for WORKS_IN is not needed for a 1:N relationship.
- If the relationship were M:N, a third table with both keys would be created — state this to show the contrast.

Q3(c)

- (i) Selection then projection: first select rows with Branch = 'CSE' (sigma), then project the Name column (pi). Expression: $\pi \text{ Name } (\sigma \text{ Branch} = \text{'CSE'} (\text{Student}))$.
- (ii) Selection with two conditions joined by AND, then projection of two columns: $\pi \text{ RollNo, Name } (\sigma \text{ Branch} = \text{'CSE'} \text{ AND Marks} > 70 (\text{Student}))$.
- (iii) Join Enrolled with Course on CourseCode (natural join or equi-join), then select RollNo = 101, then project CourseName: $\pi \text{ CourseName } (\sigma \text{ RollNo} = 101 (\text{Enrolled join Course}))$.
- Selection (sigma) chooses rows, projection (pi) chooses columns, and join combines two tables on a matching column.
- Symbols and the table/column names given in the question must be used exactly, so the expression can be checked step by step.

Q4(a)

- A functional dependency $X \rightarrow Y$ means the value of X uniquely decides the value of Y; for example $\text{EmpID} \rightarrow \text{EmpName}$.
- Normalisation is the step-by-step process of organising tables using functional dependencies, so that redundancy and insert, update and delete anomalies are removed.
- Anomalies aimed at: the same fact stored many times can be updated in one place and missed in another, and a fact cannot be inserted or is lost on deletion because it is tied to another fact.

- Types named in the syllabus, in order: 1NF, 2NF, 3NF, then BCNF, fourth normal form (4NF, which deals with multivalued dependencies) and fifth normal form (5NF).
- Each higher form keeps the properties of the lower forms and removes one more kind of dependency problem.

Q4(b)

- Given dependencies: the full key is (OrderID, ProductID); ProductID → ProductName, ProductPrice is a partial dependency, because ProductName and ProductPrice depend on only part of the key. Quantity is the only column depending on the full key.
- The table is in 1NF (values are atomic) but not in 2NF, because of this partial dependency.
- Decompose into Product (ProductID, ProductName, ProductPrice) with ProductID as primary key.
- And OrderItem (OrderID, ProductID, Quantity) with (OrderID, ProductID) as primary key and ProductID as foreign key referring to Product.
- Result: product name and price are stored once per product, so a price change is made in exactly one place; both new tables are in 2NF.

Q4(c)

- Given dependencies: EmpID → DeptID and DeptID → DeptName, DeptLocation, so DeptName and DeptLocation reach EmpID only through DeptID — a transitive dependency, which 3NF does not allow for non-key columns.
- The table should be stated to be already in 2NF (single-column key EmpID, so no partial dependency is possible).
- Decompose into Employee (EmpID, EmpName, DeptID), with EmpID primary key and DeptID foreign key.
- And Department (DeptID, DeptName, DeptLocation), with DeptID as primary key.
- BCNF condition to state: for every functional dependency $X \rightarrow Y$ in the table, X must be a super key; 3NF is slightly weaker, allowing Y to be part of a candidate key instead.

Q5(a)

- Common MySQL types to name: INT for whole numbers, VARCHAR(n) for variable-length text, CHAR(n) for fixed-length text, DATE for dates, and DECIMAL(p,s) or FLOAT for decimal numbers.
- Data definition commands are CREATE, ALTER and DROP; the answer should create the table, not insert rows.
- Expected statement: CREATE TABLE Student (RollNo INT PRIMARY KEY, Name VARCHAR(50) NOT NULL, Branch VARCHAR(20), Marks INT, AdmissionDate DATE);
- PRIMARY KEY on RollNo enforces uniqueness and NOT NULL; NOT NULL on Name stops a student being stored without a name.
- Column order, types and constraints must match the task exactly — marks are given for the correct command, table name and constraints.

Q5(b)

- INSERT: INSERT INTO Employee (EmpID, EmpName, Salary, Dept) VALUES (105, 'Asha Verma', 32000, 'Sales'); giving the column list makes the statement safe against column-order changes.
- UPDATE with WHERE: UPDATE Employee SET Salary = 30000 WHERE EmpID = 103; without the WHERE clause every row would be changed, so the condition is essential.
- DELETE with WHERE: DELETE FROM Employee WHERE EmpID = 104; again the WHERE clause limits the change to one row.
- SELECT with condition: SELECT EmpName, Salary FROM Employee WHERE Salary > 25000; only the wanted columns are listed, not *.
- Each statement ends with a semicolon, and string values are written in single quotes while numbers are not.

Q5(c)

- Operator types: arithmetic (+, -, *, /, %) do calculations; comparison (=, >, <, >=, <=, <> or !=) compare values and are used in WHERE/HAVING; logical (AND, OR, NOT) combine conditions.
- Department-wise grouping: SELECT Dept, COUNT(*) AS Staff, AVG(Salary) AS AvgSalary FROM Employee GROUP BY Dept;
- The HAVING clause filters groups after grouping, unlike WHERE which filters rows before grouping: add HAVING AVG(Salary) > 28000.
- Ordering: add ORDER BY AvgSalary DESC so the department with the highest average salary comes first.
- Complete expected query: SELECT Dept, COUNT(*) AS Staff, AVG(Salary) AS AvgSalary FROM Employee GROUP BY Dept HAVING AVG(Salary) > 28000 ORDER BY AvgSalary DESC;

KAMALDNP • kamaldnp.com • support@kamaldnp.com • +91 8318013442 • Varanasi, UP
Prepared by KAMALDNP Academy from the official BTEUP 2026-27 syllabus. This is a practice/model paper only.