DATABASE MANAGEMENT SYSTEM

Paper IV
Comprehensive Question Bank Across All Units
Strictly organized by Marks Distribution (1, 2, 5, and 8 Marks)

UNIT I: INTRODUCTION TO DATABASE, ARCHITECTURE, AND ER/EER MODELS

1 Mark Questions

Q1. Define a database management system (DBMS).

Q2. What is a database schema?

Q3. Define a database instance.

Q4. What is metadata or data dictionary?

Q5. State the meaning of data independence.

Q6. Name the three levels of the ANSI-SPARC database architecture.

Q7. What is a data model?

Q8. Define an Entity Type.

Q9. What is an Entity Set?

Q10. What is an attribute in an ER model?

Q11. Define a primary key.

Q12. What is a composite attribute? Give an example.

Q13. What is a multivalued attribute?

Q14. Define a weak entity type.

Q15. What is an identifying relationship?

Q16. What is partial key (discriminator) of a weak entity set?

Q17. Define cardinality ratio in a binary relationship.

Q18. What is participation constraint (total vs partial)?

Q19. What does specialization mean in an Enhanced ER (EER) model?

Q20. Define generalization in an EER model.

2 Mark Questions

Q1. Differentiate between a database schema and a database instance.

Q2. Explain logical data independence with a brief example.

Q3. Explain physical data independence with a brief example.

Q4. What are the main duties or roles of a Database Administrator (DBA)?

Q5. Differentiate between simple and composite attributes.

Q6. Differentiate between stored and derived attributes with examples.

Q7. Explain the concept of key attributes in an Entity Relationship model.

Q8. Describe structural constraints of a relationship type.

Q9. What is a recursive relationship type? Give a brief illustration.

Q10. Explain weak entity types and how they are represented visually in an ER diagram.

Q11. Describe the purpose of naming conventions in ER modeling.

Q12. What is an attribute defined as a domain? Explain briefly.

Q13. Explain subclass and superclass concepts in the EER model.

Q14. What is a category or union type in an EER model?

Q15. State the difference between specialization and generalization.

5 Mark Questions

Q1. Explain the traditional file system approach and discuss why a DBMS is preferred over it.

Q2. Describe the three-schema architecture of a database system with its advantages.

Q3. Discuss different types of database users and their interfaces to access the database system.

Q4. Explain different categories of data models (high-level, low-level, representational).

Q5. Elaborate on the concepts of Entity, Entity Types, Entity Sets, Attributes, and Keys in ER design.

Q6. Explain structural constraints on relationship types, including cardinality ratios and participation constraints.

Q7. Describe weak entity types completely, highlighting their identifying relationships and partial keys with an example.

Q8. Discuss Enhanced Entity-Relationship (EER) model concepts including specialization, generalization, and attribute inheritance.

Q9. Explain constraints on specialization and generalization (disjointness and completeness constraints).

Q10. Design a brief ER diagram for a University database system, mentioning key entities, attributes, and relationships.

8 Mark Questions

Q1. Provide an in-depth analysis of the database system architecture. Sketch a well-labeled block diagram showing its components and detail how query processing and storage management are handled internally.

Q2. Discuss the Entity-Relationship (ER) model extensively. Explain all types of attributes, entities, keys, and structural constraints, and provide standard graphical notations used to represent them in ER diagrams.

Q3. Elaborate on the Enhanced Entity-Relationship (EER) model. Contrast specialization and generalization with detailed examples, and discuss all four combinations of specialization constraints with real-world scenarios.

Q4. Formulate a complete ER design strategy for a Company Management System that monitors employees, departments, and multiple projects. Identify all entities, attributes, primary keys, structural constraints, and draw the complete diagram.

Q5. Explain the complete step-by-step mapping algorithm used to convert a conceptual Entity-Relationship (ER) diagram into a logical Relational Database Schema, covering weak entities and binary relationships.

UNIT II: RELATIONAL DATA MODEL, SQL, AND RELATIONAL ALGEBRA/CALCULUS

1 Mark Questions

Q1. What is a tuple in the relational data model?

Q2. Define a relation schema.

Q3. What does the domain of an attribute signify?

Q4. What is a candidate key?

Q5. Define a foreign key.

Q6. State the entity integrity constraint.

Q7. State the referential integrity constraint.

Q8. What SQL keyword is used to remove duplicates from a SELECT query?

Q9. Which command is used to add a new column to an existing table in SQL?

Q10. What is the purpose of the SQL CHECK constraint?

Q11. Name the basic DML statements in SQL.

Q12. Which relational algebra operation acts as a horizontal filter?

Q13. Which relational algebra operation acts as a vertical filter?

Q14. Is relational algebra a procedural or non-procedural language?

Q15. What is the result of a Cartesian Product between relation R (m tuples) and S (n tuples)?

Q16. Write the SQL clause used to filter groups formed by the GROUP BY clause.

Q17. What does the wildcard character '%' represent in an SQL LIKE clause?

Q18. Define the JOIN operation in relational algebra.

Q19. What is the symbol used for the PROJECT operation in relational algebra?

Q20. What is a safe expression in relational calculus?

2 Mark Questions

Q1. Explain the difference between a candidate key and a primary key.

Q2. Describe the purpose of the foreign key constraint with a brief scenario.

Q3. Explain the Entity Integrity constraint with an example.

Q4. Explain the Referential Integrity constraint with an example.

Q5. Differentiate between the DROP and TRUNCATE statements in SQL.

Q6. How do the INSERT and UPDATE statements differ in their operations?

Q7. Explain the usage of the SQL DEFAULT constraint.

Q8. What is the difference between the WHERE clause and the HAVING clause?

Q9. Describe the SELECT (sigma) operation in relational algebra with its syntax.

Q10. Describe the PROJECT (pi) operation in relational algebra with its syntax.

Q11. Explain the concept of Natural Join in relational algebra.

Q12. What is the Division operation in relational algebra used for? Give a brief scenario.

Q13. Differentiate between Tuple Relational Calculus and Domain Relational Calculus.

Q14. Explain basic SQL data types for handling character data (CHAR vs VARCHAR).

Q15. What are aggregate functions in SQL? List any four.

5 Mark Questions

Q1. Explain the relational model concepts, including domains, attributes, tuples, relations, characteristics of relations, and relational integrity constraints.

Q2. Discuss SQL data definition language commands (CREATE, ALTER, DROP) along with data types and integrity constraints creation.

Q3. Explain retrieval queries in SQL covering nested subqueries, grouping, aggregation, and ordering operations.

Q4. Describe INSERT, DELETE, and UPDATE statements in SQL with clear syntax and examples for each.

Q5. Explain the unary operations SELECT and PROJECT in relational algebra with sample tables and expressions.

Q6. Discuss binary operations in relational algebra: Cartesian Product, Set operations (Union, Intersection, Set Difference).

Q7. Explain the different types of JOIN operations in relational algebra (Theta join, Equijoin, Natural join, Outer joins).

Q8. Elaborate on the DIVISION operation in relational algebra, providing a complete example with relations.

Q9. Explain the core features of Tuple Relational Calculus (TRC) and discuss how it differs from relational algebra.

Q10. Consider two relations: Employee(EmpID, Name, DeptID, Salary) and Department(DeptID, DeptName). Write SQL queries to find names of employees earning above 50,000 and to count employees in each department.

8 Mark Questions

Q1. Explain Structured Query Language (SQL) comprehensively. Discuss its classification into DDL, DML, and DCL commands, providing syntax and fully realized execution examples for table manipulation, modification, and data retrieval.

Q2. Discuss Relational Algebra in detail. Explain all fundamental operations (Select, Project, Union, Set Difference, Cartesian Product, Rename) and illustrate how complex database queries are constructed using combinations of these operators.

Q3. Elaborate on Join operations in databases. Formulate definitions, notation, and structural outcomes for Inner Joins, Natural Joins, Left Outer Joins, Right Outer Joins, and Full Outer Joins, supplemented with clear tabular data tracing.

Q4. Compare Relational Algebra, Tuple Relational Calculus, and Domain Relational Calculus. Discuss procedurality, safety of expressions, declarative nature, and how relational completeness is achieved among these paradigms.

Q5. Given a relational schema: Student(RollNo, Name, Age, Branch), Course(CourseID, CourseName, Credits), and Enrollment(RollNo, CourseID, Grade). Formulate both Relational Algebra expressions and SQL statements to retrieve complex data cross-referencing all three tables.

UNIT III: DATABASE DESIGN THEORY AND NORMALIZATION

1 Mark Questions

Q1. What is data redundancy?

Q2. Define functional dependency.

Q3. What is a trivial functional dependency?

Q4. What does closure of an attribute set signify?

Q5. State Armstrong's axiom of Reflexivity.

Q6. State Armstrong's axiom of Augmentation.

Q7. State Armstrong's axiom of Transitivity.

Q8. What is a prime attribute?

Q9. Define non-prime attribute.

Q10. What is the condition for a relation to be in First Normal Form (1NF)?

Q11. Define partial functional dependency.

Q12. What condition satisfies Second Normal Form (2NF)?

Q13. What is a transitive functional dependency?

Q14. Define Third Normal Form (3NF) based on primary keys.

Q15. In which normal form is every determinant required to be a super key?

Q16. What is a multivalued dependency (MVD)?

Q17. Which normal form deals with multivalued dependencies?

Q18. What is a join dependency?

Q19. Which normal form is achieved through join dependencies?

Q20. Can a table be in BCNF but not in 3NF?

2 Mark Questions

Q1. Explain functional dependency X -> Y with a formal definition.

Q2. Differentiate between trivial and non-trivial functional dependencies.

Q3. What are Armstrong's axioms? List the three primary rules.

Q4. Explain what a partial dependency means with a simple relation example.

Q5. Explain transitive dependency with a simple relation example.

Q6. Why is Boyce-Codd Normal Form (BCNF) considered stricter than Third Normal Form (3NF)?

Q7. Define First Normal Form (1NF) and explain the concept of atomic values.

Q8. What is the purpose of normalization in database design?

Q9. Define closure of an attribute set and its notation.

Q10. What is an insertion anomaly? Give a brief description.

Q11. What is a deletion anomaly? Give a brief description.

Q12. What is a modification or update anomaly?

Q13. Define Multivalued Dependency (MVD) and state its standard notation.

Q14. What is a join dependency, and which normal form relies on it?

Q15. Explain lossless-join decomposition briefly.

5 Mark Questions

Q1. Discuss the problems or anomalies that arise when databases are designed poorly without proper theoretical guidelines.

Q2. Explain functional dependencies completely, covering definition, properties, and inference rules (Armstrong's axioms).

Q3. Describe the step-by-step algorithm to find the closure of an attribute set under a given set of functional dependencies.

Q4. Explain First Normal Form (1NF) with a non-atomic relation example and demonstrate how it is converted to 1NF.

Q5. Define Second Normal Form (2NF). Provide a relation with partial dependency and show its decomposition into 2NF tables.

Q6. Define Third Normal Form (3NF). Provide an example demonstrating transitive dependency and show how it is resolved.

Q7. Explain Boyce-Codd Normal Form (BCNF). Contrast it with 3NF and show a scenario where 3NF is satisfied but BCNF is violated.

Q8. Discuss Multivalued Dependency (MVD) and Fourth Normal Form (4NF) with a suitable normalization example.

Q9. Explain Join Dependencies and Fifth Normal Form (5NF), highlighting their practical significance in advanced design theory.

Q10. Given a relation R(A, B, C, D) and functional dependencies F = {A -> B, B -> C, C -> D}. Determine the candidate key(s) of the relation and identify its highest normal form status.

8 Mark Questions

Q1. Write a definitive essay on database normalization. Trace the design progression from 1NF through 5NF, detailing the anomalies each normal form eradicates, the structural criteria required, and concrete structural decompositions.

Q2. Discuss functional dependencies and the process of finding candidate keys. Given a relation R(A, B, C, D, E) with F = {A -> BC, CD -> E, B -> D, E -> A}, find all candidate keys and check if the decomposition into R1(A,B,C) and R2(C,D,E) is lossless.

Q3. Analyze Boyce-Codd Normal Form (BCNF) exhaustively. Illustrate why BCNF is necessary even when a relation is in 3NF, present a detailed mathematical proof or structural counter-example, and explain dependency preservation constraints.

Q4. Explain multi-valued dependencies and 4NF, along with join dependencies and 5NF. Provide comprehensive scenarios for each, demonstrating why standard 3NF or BCNF is insufficient to handle independent multi-valued facts.

Q5. Detail properties of relational decompositions. Prove the mathematical criteria needed to evaluate whether a specific table decomposition guarantees both a Lossless-Join condition and Dependency Preservation simultaneously.

UNIT IV: TRANSACTION PROCESSING CONCEPTS AND CONCURRENCY CONTROL

1 Mark Questions

Q1. What is a transaction in a database system?

Q2. List the ACID properties of transactions.

Q3. Which ACID property guarantees that all operations of a transaction are completed or none are?

Q4. Which component of a DBMS ensures the Durability property?

Q5. What is a schedule in transaction processing?

Q6. Define a serial schedule.

Q7. What is conflict serializability?

Q8. What is view serializability?

Q9. Define a recoverable schedule.

Q10. What is a cascadeless schedule?

Q11. What is concurrency control?

Q12. Name the two types of locks used in locking techniques.

Q13. What is a shared lock (S-lock)?

Q14. What is an exclusive lock (X-lock)?

Q15. State the basic rule of Two-Phase Locking (2PL).

Q16. What is a deadlock?

Q17. What does a timestamp mean in a transaction context?

Q18. What is the system log used for?

Q19. Define starvation in concurrency control.

Q20. What is the commit point of a transaction?

2 Mark Questions

Q1. Explain the Atomicity property of a database transaction.

Q2. Explain the Consistency preservation property of a transaction.

Q3. Explain the Isolation property of a database transaction.

Q4. Explain the Durability property of a database transaction.

Q5. Differentiate between active, committed, and aborted states of a transaction.

Q6. What is a schedule? Distinguish between serial and non-serial schedules.

Q7. Explain conflict operations in a schedule with their two conditions.

Q8. What is a cascade-less schedule? Why is it preferred?

Q9. Describe shared and exclusive locks with their compatibility matrix.

Q10. Explain the growing phase and shrinking phase in a Two-Phase Locking (2PL) protocol.

Q11. What is Strict 2PL? State its operational rule.

Q12. Explain how a timestamp ordering protocol enforces transaction sequence.

Q13. What is the Thomas Write Rule in concurrency control?

Q14. Define a wait-for graph (WFG) used in deadlock detection.

Q15. How does the roll-back mechanism work during transaction aborts?

5 Mark Questions

Q1. Explain the lifecycle states of a database transaction with a neatly structured state transition diagram.

Q2. Discuss the ACID properties of transactions in detail, highlighting the subsystem of DBMS responsible for enforcing each.

Q3. Explain the concept of serializability and discuss why non-serial but serializable schedules are essential for concurrency.

Q4. Elaborate on conflict serializability and describe how a precedence graph is constructed to test for conflict serializability.

Q5. Discuss recoverability of schedules, differentiating between recoverable, cascade-free, and strict schedules with concrete examples.

Q6. Explain the basic Locking techniques for concurrency control, including lock upgrading, downgrading, and compatibility parameters.

Q7. Describe the Two-Phase Locking (2PL) protocol for concurrency control. Explain its limitations regarding deadlocks and cascading rollbacks.

Q8. Discuss Concurrency Control based on Time-Stamp Ordering, explaining read_ts, write_ts, and protocol operation rules.

Q9. Explain the deadlock problem in transaction environments. Discuss deadlock prevention schemes (wait-die and wound-wait) based on timestamps.

Q10. Describe deadlock detection using a wait-for graph and explain the process of recovery from a deadlock situation.

8 Mark Questions

Q1. Provide a comprehensive analysis of Transaction Processing Systems. Discuss transaction states, full details of ACID properties, and evaluate the precise real-world issues (lost updates, dirty reads, unrepeatable reads) that occur due to concurrent execution without control.

Q2. Analyze Serializability thoroughly. Formulate definitions and structural proofs for both Conflict Serializability and View Serializability, explaining testing strategies via precedence graphs and dependency structures.

Q3. Elaborate on the Two-Phase Locking (2PL) Protocol. Compare Basic 2PL, Conservative 2PL, Strict 2PL, and Rigorous 2PL with respect to lock points, concurrency level, cascade-free attributes, and prevention of deadlocks.

Q4. Discuss Concurrency Control through Timestamp Ordering. Detail how the basic protocol handles read and write requests from transactions, explain how the Thomas Write Rule modifies this behavior, and evaluate its overall performance profile.

Q5. Investigate the Deadlock problem in Distributed or Centralized DBMS environments. Formulate an exhaustive breakdown of prevention mechanisms (including lock timeouts, wait-die, wound-wait) versus detection/recovery techniques, evaluating rollback choice costs.