DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE

Department of AI & DS

U23AIT31 – DATABASE MANAGEMENT SYSTEMS

QUESTION BANK (2 Mark Q&A + 16 Mark Questions)

Faculty: Ms. P. Varshini | Year/Sem: II/III | Batch: AI&DS

PART – A: TWO MARK QUESTIONS WITH ANSWERS

QUESTION	ANSWER
UNIT I – CONCEPTUAL DATA MODELING	
Q1. Define a Database.	**Ans:** A database is an organized collection of structured data stored electronically, managed by a DBMS to enable efficient access, retrieval, and manipulation.
Q2. What is a DBMS?	**Ans:** A Database Management System (DBMS) is software that provides an interface between the database and its users/programs to define, create, maintain, and control access to data.
Q3. What is a Database System Development Lifecycle?	**Ans:** It is the set of phases involved in developing a database system: planning, system definition, requirements collection, design, implementation, testing, and maintenance.
Q4. Define Entity and Entity Set.	**Ans:** An entity is a real-world object with an independent existence. An entity set is a collection of entities of the same type sharing the same attributes.
Q5. What is an Attribute? Name its types.	**Ans:** An attribute is a property or characteristic of an entity. Types: Simple, Composite, Multi-valued, Derived, and Null attributes.
Q6. Define a Relationship and Relationship Set.	**Ans:** A relationship is an association among two or more entities. A relationship set is a collection of similar relationships between entity sets.
Q7. What is an ER Diagram?	**Ans:** An Entity-Relationship Diagram is a graphical representation showing entities, their attributes, and the relationships between them in a database.
Q8. What is the Enhanced ER (EER) Model?	**Ans:** EER extends the basic ER model with concepts like specialization, generalization, aggregation, and inheritance to model complex real-world scenarios.
Q9. Define Specialization and Generalization.	**Ans:** Specialization: Dividing an entity into sub-entities. Generalization: Combining several entities into a higher-level entity based on common properties.
Q10. What is a UML Class Diagram?	**Ans:** A UML Class Diagram represents the structure of a system by showing classes, their attributes, operations, and relationships between objects.

UNIT II – RELATIONAL MODEL AND SQL	
Q11. Define a Relation and Tuple.	**Ans:** A relation is a table with rows and columns. A tuple is a single row/record in a relation representing one instance of an entity.
Q12. What is a Primary Key?	**Ans:** A primary key is a minimal set of attributes that uniquely identifies each tuple in a relation and cannot contain NULL values.
Q13. Define Referential Integrity Constraint.	**Ans:** It states that a foreign key value must either match a primary key value in the referenced relation or be NULL, ensuring consistency between related tables.
Q14. What is SQL? Expand it.	**Ans:** SQL stands for Structured Query Language. It is a standard language for creating, manipulating, and querying relational databases.
Q15. Differentiate DDL and DML.	**Ans:** DDL (Data Definition Language) defines database structure (CREATE, ALTER, DROP). DML (Data Manipulation Language) manipulates data (INSERT, UPDATE, DELETE, SELECT).
Q16. What is a View in SQL?	**Ans:** A view is a virtual table derived from one or more base tables using a SELECT statement. It does not store data physically.
Q17. What are Integrity Constraints?	**Ans:** Rules enforced on database data to maintain accuracy and consistency. Types: Domain, Entity integrity, Referential integrity, and Key constraints.
Q18. Define Foreign Key.	**Ans:** A foreign key is an attribute in one relation that references the primary key of another relation, establishing a link between two tables.
Q19. What is SQL programming / PL/SQL?	**Ans:** PL/SQL is a procedural extension of SQL that allows writing programs with variables, loops, conditions, and exception handling within the database.
Q20. What is a NULL value in SQL?	**Ans:** NULL represents a missing, unknown, or inapplicable value in a database. It is different from zero or an empty string.
UNIT III – RELATIONAL DATABASE DESIGN AND NORMALIZATION	
Q21. What is Normalization?	**Ans:** Normalization is the process of organizing a database to reduce redundancy and improve data integrity by decomposing relations into smaller, well-structured ones.
Q22. Define Functional Dependency.	**Ans:** A functional dependency X → Y means the value of attribute set X uniquely determines the value of attribute set Y in a relation.
Q23. What is 1NF?	**Ans:** A relation is in 1NF if all attribute values are atomic (indivisible) and each column contains values of a single type with no repeating groups.
Q24. Define 2NF.	**Ans:** A relation is in 2NF if it is in 1NF and every non-prime attribute is fully functionally dependent on the entire primary key (no partial dependency).

Q25. What is 3NF?	**Ans:** A relation is in 3NF if it is in 2NF and no non-prime attribute is transitively dependent on the primary key.
Q26. Define BCNF.	**Ans:** Boyce-Codd Normal Form: A relation is in BCNF if for every functional dependency X → Y, X is a super key of the relation.
Q27. What are Update Anomalies?	**Ans:** Problems that occur in unnormalized databases: Insertion anomaly (can't add data without other data), Deletion anomaly (deleting data loses other data), Update anomaly (updating one record requires updating many).
Q28. Define Minimal Cover (Canonical Cover).	**Ans:** A minimal cover is a simplified set of functional dependencies equivalent to the original set, with no redundant dependencies or extraneous attributes.
Q29. What is Lossless Decomposition?	**Ans:** A decomposition is lossless if the original relation can be reconstructed exactly by joining the decomposed relations without generating spurious tuples.
Q30. What are Armstrong's Axioms?	**Ans:** Rules to infer all functional dependencies: Reflexivity, Augmentation, and Transitivity. Used to find the closure of a set of functional dependencies.
UNIT IV – TRANSACTION MANAGEMENT	
Q31. Define a Transaction.	**Ans:** A transaction is a logical unit of database operations that must be executed atomically; it either completes fully or not at all.
Q32. What are ACID Properties?	**Ans:** Atomicity (all or nothing), Consistency (valid state), Isolation (transactions don't interfere), Durability (committed changes persist permanently).
Q33. What is a Schedule?	**Ans:** A schedule is a sequence of instructions from multiple transactions that specifies the chronological order in which the instructions are executed.
Q34. Define Serializability.	**Ans:** A schedule is serializable if its outcome is equivalent to some serial schedule of the same transactions, ensuring correctness in concurrent execution.
Q35. What is Concurrency Control?	**Ans:** Concurrency control manages simultaneous transaction execution to ensure consistency and isolation without interference between transactions.
Q36. What is a Deadlock?	**Ans:** A deadlock occurs when two or more transactions are waiting indefinitely for each other to release locks, resulting in a circular wait situation.
Q37. Define Two-Phase Locking (2PL).	**Ans:** A concurrency control protocol where each transaction has two phases: Growing phase (only acquires locks) and Shrinking phase (only releases locks).
Q38. What is a Lock?	**Ans:** A lock is a mechanism to control concurrent access to data items. Types: Shared lock (read) and Exclusive lock (write).

Q39. Define Conflict Serializability.	**Ans:** A schedule is conflict serializable if it can be transformed into a serial schedule by swapping non-conflicting operations.
Q40. What is Recovery in DBMS?	**Ans:** Recovery is the process of restoring a database to a correct state after a failure using techniques like log-based recovery, checkpointing, and shadow paging.
UNIT V – OBJECT RELATIONAL AND NO-SQL DATABASES	
Q41. What is an Object-Relational Database?	**Ans:** An ORDB extends the relational model with OOP features like user-defined types, inheritance, and object identifiers to handle complex data.
Q42. Define Object Identifier (OID).	**Ans:** An OID is a unique, system-generated identifier for each object in an object-relational database, used to reference objects regardless of their content.
Q43. What is a UDT (User-Defined Type)?	**Ans:** A UDT is a custom data type created by users in SQL/object-relational systems to represent complex structures beyond built-in data types.
Q44. What is the CAP Theorem?	**Ans:** CAP states that a distributed database can guarantee only two of three properties simultaneously: Consistency, Availability, and Partition Tolerance.
Q45. What is a No-SQL Database?	**Ans:** NoSQL databases are non-relational databases designed for large-scale data storage and horizontal scalability. Types: Document, Key-Value, Column-family, and Graph stores.
Q46. What is MongoDB?	**Ans:** MongoDB is a document-oriented NoSQL database that stores data as BSON (Binary JSON) documents, allowing flexible schema design.
Q47. Define CRUD Operations.	**Ans:** CRUD stands for Create, Read, Update, and Delete – the four basic operations performed on database records.
Q48. What are Collection Types in ORDB?	**Ans:** Collection types include sets, multisets (bags), lists, and arrays that allow an attribute to hold multiple values as a structured collection.
Q49. What is a Column-based Database?	**Ans:** A column-family store (e.g., HBase) stores data in columns rather than rows, optimized for read-heavy analytical workloads on large datasets.
Q50. Differentiate SQL and NoSQL databases.	**Ans:** SQL: Fixed schema, ACID compliant, uses tables and relations. NoSQL: Flexible schema, BASE model, supports documents/key-value/graph structures, horizontally scalable.

PART – B: SIXTEEN MARK QUESTIONS

UNIT I – CONCEPTUAL DATA MODELING

1. Explain the Database System Development Lifecycle with each phase in detail.
2. Describe the Entity-Relationship Model. Explain various types of attributes and relationships with examples and ER notation.
3. Explain the Enhanced Entity-Relationship (EER) Model. Discuss Specialization, Generalization, Aggregation, and Inheritance with examples.
4. Draw and explain the ER diagram for a Hospital Management System with all entities, attributes, and relationships.
5. Explain UML Class Diagrams and how they are used to model database systems. Compare ER model with UML.

UNIT II – RELATIONAL MODEL AND SQL

1. Explain the Relational Data Model. Discuss Relational Model concepts, properties of a relation, and various integrity constraints.
2. Explain SQL Data Definition Language (DDL) commands with syntax and examples: CREATE, ALTER, DROP, TRUNCATE.
3. Explain SQL Data Manipulation Language (DML) with examples. Write queries for SELECT with WHERE, GROUP BY, HAVING, ORDER BY, and JOIN operations.
4. Explain Views in SQL with creation, advantages, and limitations. Write SQL procedures and functions with examples (SQL Programming/PL/SQL).
5. What are integrity constraints? Explain each type with SQL syntax: Primary Key, Foreign Key, Unique, Not Null, Check, and Default constraints.

UNIT III – RELATIONAL DATABASE DESIGN AND NORMALIZATION

1. Explain Functional Dependencies and Armstrong's Axioms. How is the closure of a set of functional dependencies computed? Illustrate with an example.
2. Explain Normalization in detail. Describe 1NF, 2NF, 3NF, and BCNF with examples and step-by-step normalization process.
3. What are update anomalies? Explain Insertion, Deletion, and Modification anomalies with examples and how normalization resolves them.
4. Explain Minimal Cover (Canonical Cover) and how to compute it. Also explain Lossless and Dependency-preserving decomposition.
5. Explain the mapping of ER and EER diagrams to Relational schemas with rules and examples.

UNIT IV – TRANSACTION MANAGEMENT

1. Define a Transaction. Explain ACID properties with examples and the importance of each property in database systems.
2. What is a Schedule? Explain Conflict Serializability and View Serializability with examples. How is a Precedence Graph used to test conflict serializability?
3. Explain Two-Phase Locking (2PL) protocol in detail. Discuss its variants: Basic 2PL, Strict 2PL, Rigorous 2PL, and Conservative 2PL.
4. What are Concurrency Control problems? Explain Lost Update, Dirty Read, Unrepeatable Read, and Phantom Read with examples and solutions.

5. Explain Transaction Recovery techniques: Log-based recovery, Deferred Update, Immediate Update, Checkpointing, and Shadow Paging.

UNIT V – OBJECT RELATIONAL AND NO-SQL DATABASES

1. Explain Object-Relational Database concepts: OID, Reference types, Row types, UDTs, Subtypes, Supertypes, and User-defined routines.

2. What is the CAP Theorem? Explain with examples. How does it apply to the design of distributed No-SQL database systems?

3. Explain MongoDB as a Document-based NoSQL database. Describe its data model, BSON format, collections, and CRUD operations with examples.

4. Explain Column-based databases (HBase). Compare it with Row-based databases and discuss when to use each.

5. Compare Relational, Object-Relational, and No-SQL databases. Explain Object Query Language (OQL) and Mapping EER to ODB schema with examples.