DATABASE MANAGEMENT SYSTEMS – ANSWERS (TEST I & II, SEMESTER IV)

PART A (7 x 2 = 14 Marks)

1. What is Normalization in Database Design?

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. It involves dividing large tables into smaller ones and defining relationships between them. Forms like 1NF, 2NF, and 3NF help progressively refine the design.

2. What is the purpose of a database system in modern organizations?

To store, retrieve, and manage data efficiently. Databases improve decision-making, ensure data consistency, enable secure access, and support business intelligence tools.

3. What are primary keys and foreign keys?

- **Primary Key**: Uniquely identifies each record in a table.
- **Foreign Key**: A field in one table that refers to the primary key in another, establishing a relationship.

4. What are relational databases?

A relational database stores data in tables (relations), which consist of rows and columns. It uses SQL for querying and supports operations like joins and constraints.

5. What is SQL and how is it related to database languages?

SQL (Structured Query Language) is a standard language used to communicate with relational databases. It supports DDL, DML, DCL, and TCL commands.

6. What is a database system?
A software system that uses a database and a DBMS to store, manipulate, and manage data efficiently.

7. What is a view in a database system?
A view is a virtual table derived from one or more base tables. It simplifies complex queries and restricts data access.

PART B (3 x 4 = 12 Marks)

8(a). What are the main components of SQL and their functions?

- **DDL (Data Definition Language)**: Used to define and modify database structures (e.g., CREATE, ALTER, DROP).
- **DML (Data Manipulation Language)**: Used to retrieve and manipulate data (e.g., SELECT, INSERT, UPDATE, DELETE).
- **DCL (Data Control Language)**: Manages access control (e.g., GRANT, REVOKE).
- **TCL (Transaction Control Language)**: Ensures transaction integrity (e.g., COMMIT, ROLLBACK, SAVEPOINT).

Together, these enable full control over a relational database: structure definition, data manipulation, access authorization, and transaction safety.

OR

8(b). What are the types of operations used to modify a database?

- **INSERT**: Adds new rows to a table.
- **UPDATE**: Modifies existing data in a table.
- **DELETE**: Removes rows from a table.
- **MERGE**: Combines insert and update operations based on a condition (also known as UPSERT).
  These operations enable dynamic data manipulation and are fundamental to data integrity.

9(a). What are the fundamental relational concepts in database systems?

- **Entity and Attributes**: Real-world objects and their properties.
- **Domains**: Valid sets of values for attributes.
- **Tuples and Relations**: Rows and tables respectively.
- **Keys**: Primary (uniquely identifies tuples), Candidate (possible primary keys), Foreign (creates relationships).
- **Integrity Constraints**: Rules ensuring data correctness (e.g., entity integrity, referential integrity).
- **Normalization**: Structuring data to minimize redundancy and improve consistency.
  These concepts ensure data is stored logically, efficiently, and without anomalies.

OR

9(b). What are the set operations in SQL?
Set operations allow combining or comparing multiple query results:

- **UNION**: Returns distinct records from two SELECT statements.

- **UNION ALL**: Includes duplicates as well.
- **INTERSECT**: Returns records common to both queries.
- **EXCEPT/MINUS**: Returns records from the first query not found in the second. These help construct complex query logic cleanly and efficiently.

10(a). What are the differences between ODBC and JDBC?

Feature	ODBC	JDBC
Language	C/C++	Java
Platform	Platform-independent	Java-specific
API Type	Native	Object-oriented
Use Case	General DB access	Java-based applications

ODBC is ideal for multi-language environments, while JDBC is tailor-made for Java applications with integrated exception handling.

OR

10(b). What are the main components of a DBMS?

- **Database Engine**: Processes queries and manages access.
- **Query Processor**: Parses, optimizes, and executes SQL.
- **Storage Manager**: Controls how data is stored on disk.
- **Transaction Manager**: Maintains ACID properties.
- **Metadata Manager**: Maintains data about data (schemas, indexes).
- **Concurrency and Recovery Managers**: Handle multiple users and ensure consistency during failures.

These components work together to provide seamless, reliable, and high-performance data management.

PART C (2 x 12 = 24 Marks)

11(a). What are the advantages and disadvantages of DBMS?

Advantages:

- **Data Independence**: Physical storage changes don’t affect app logic.
- **Minimized Redundancy**: Prevents data duplication.
- **Improved Integrity**: Constraints and transactions keep data valid.
- **Security & Access Control**: User-based permission models.

- **Concurrent Access**: Multi-user support with locking mechanisms.
- **Backup & Recovery**: Automatic and manual options available.

Disadvantages:

- **Costly Hardware/Software**: High initial investment.
- **Complexity**: Requires expert administration.
- **Potential Performance Overhead**: Additional layers of abstraction.

OR

11(b). What are the advantages of using UML in software design?
UML (Unified Modeling Language) is used to visually represent systems. In database design, it:

- **Clarifies Requirements**: Through use-case and activity diagrams.
- **Models Structure**: Class diagrams represent tables/entities.
- **Improves Communication**: Stakeholders grasp models faster.
- **Supports Reusability & Consistency**: Modular design with clear documentation.
- **Links Business Logic to DB Schema**: Ensures software and data design remain aligned.

12(a). What are the differences between Specialization and Generalization?

Concept	Specialization	Generalization
Direction	Top-down	Bottom-up
Purpose	Classify an entity into sub-entities	Combine entities into a common superclass
Example	Employee → Developer, Manager	Car, Bike → Vehicle

Use in DBMS: Enhances semantic clarity in ER diagrams, supports inheritance and constraint propagation.

OR

12(b). What are the key steps in the database design process?

1. **Requirements Collection & Analysis**: Understand data and operations.
2. **Conceptual Design**: Build ER model.
3. **Logical Design**: Convert ER model into relational schema.
4. **Schema Refinement**: Apply normalization (1NF to 3NF or BCNF).
5. **Physical Design**: Define indexing, partitions, storage.
6. **Implementation & Testing**: Load data, test queries.
7. **Monitoring & Optimization**: Tune queries, adjust schema as needed.

This approach ensures a scalable, high-performance, and user-oriented database system.

DATABASE MANAGEMENT SYSTEMS – ANSWERS (TEST II)

PART A (7 x 2 = 14 Marks)

1. What is buffer management in DBMS?

It handles data transfer between main memory and disk, using a buffer pool to minimize I/O and improve performance.

2. What is the purpose of a recovery system in DBMS?

To restore the database to a consistent state after failure, ensuring durability and atomicity.

3. What is a B+ Tree?

A self-balanced tree structure used in indexing. Leaf nodes store actual data pointers; internal nodes guide the search.

4. What is a remote backup system?

Stores copies of data at off-site locations to ensure recovery in case of local data loss or disasters.

5. What is RAID?

Redundant Array of Independent Disks – technique to combine multiple drives to improve performance and fault tolerance.

6. Name two types of joins.

- Inner Join
- Outer Join (Left/Right/Full)

7. What is a query tree? A tree representation of query operations. Nodes are relational algebra operations; leaves are base tables.

SECTION B

8(a). What is UML? How is it used in database design?

UML supports:

- **Class Diagrams**: Represent tables and relationships.
- **Sequence Diagrams**: Show interaction between components (e.g., transactions).
- **Use-Case Diagrams**: Define user roles and actions.
- **State Diagrams**: Track record state transitions.
- UML enhances collaborative modeling, useful for large enterprise applications.

8(b). Explain heap and clustered file organization.

- **Heap**:
  - No ordering.
  - Fast for inserts.
  - Slower for reads.
- **Clustered**:
  - Records are stored in sorted order.
  - Good for range-based queries.
  - Faster read times.

9(a). What are primary and secondary indexes?

- **Primary**:
  - On unique fields.
  - Usually sorted.
  - 1 per table.
- **Secondary**:
  - Non-unique fields.
  - Can be multiple.
  - Useful for searching non-key columns.

9(b). Describe the structure and benefits of B+ Trees.

- **Structure**: Multi-level tree with root, intermediate, and leaf nodes. Only leaf nodes hold actual data pointers.
- **Benefits**:
  - Maintains sorted order.
  - Supports range and equality searches.

    - Balanced, so access is logarithmic time.
    - Great for large databases.

10(a). Explain the ACID properties of a transaction.

- **Atomicity**: Ensures all-or-nothing execution.
- **Consistency**: Guarantees valid state transitions.
- **Isolation**: Simultaneous transactions do not conflict.
- **Durability**: Changes persist even after crash.
  These properties ensure reliability and predictability in DB operations.

10(b). How are transactions implemented in a DBMS?

- **Logging**: Write-ahead logs capture changes.
- **Checkpointing**: Save stable states.
- **Concurrency Control**: Locking protocols (2PL), time-stamp ordering.
- **Recovery Manager**: Handles crashes via redo/undo operations.

11(a). Selection operation in query processing

- **Definition**: Filters rows satisfying a condition.
- **Methods**:
    - Sequential scan
    - Index scan
    - Selective filters
    - Parallelization
- **Optimizations**:
    - Use composite indexes
    - Materialized views
    - Predicate pushdown in DB engines

11(b). Join Operations

- **Nested Loop Join**: Easy but slow O(n*m)
- **Sort-Merge Join**: Good for pre-sorted data
- **Hash Join**: Efficient for large datasets; O(n)
- Choose join based on data size, indexing, and memory constraints.

12(a). Serializability & Concurrency Control

- **Serializability**: Equivalent to serial order execution.
- **Lock-based protocols**:
    - Binary locks (read/write)
    - Two-Phase Locking (2PL)
    - Strict 2PL: Wait until commit to release locks
- **Goal**: Prevent dirty reads, lost updates, phantom reads.

12(b). Failure with Loss of Non-Volatile Storage

- **Issue**: Power loss or hardware failure deletes all disk data.
- **Recovery**:
    - Remote/cloud backup (AWS, GCP)
    - RAID redundancy
    - Write-Ahead Logging (WAL)
    - Replication and mirroring

Backup strategies are vital for business continuity and disaster recovery.
