DBMS Module 1-4 Summary

# Module 1: Introduction to DBMS

1. What is a Database?  
- A collection of related data with real-world meaning.  
- Must be reliable, accurate, and meaningful.

2. What is DBMS?  
- Software to define, construct, and manage databases.  
- Handles access, security, concurrency, backup, and recovery.

3. Characteristics of DBMS:  
- Self-describing  
- Program-data independence  
- Data abstraction  
- Support for multiple views  
- Multiuser access

4. Advantages of DBMS:  
- Reduces redundancy  
- Enhances security and integrity  
- Data persistence  
- Efficient access and performance

5. Evolution of Databases:  
- Hierarchical, Network → Relational → Object-Oriented → XML → Big Data

6. Data Models:  
- High-level (ER), Low-level (physical), Representational (relational)

7. Schema vs Instance:  
- Schema: Structure of database (stable)  
- Instance: Current data (dynamic)

8. Three-Schema Architecture:  
- External, Conceptual, Internal

9. Data Independence:  
- Logical and Physical

10. DB Languages:  
- DDL, DML, DCL

11. ER Model Basics:  
- Entity, Attribute, Relationship, Keys, Cardinality

12. Specialization/Generalization:  
- Inheritance relationships in data models

# Module 2: Relational Model & Algebra

1. Relational Model Terms:  
- Relation = Table, Tuple = Row, Attribute = Column

2. Functional Dependencies (FD):  
- X → Y: Y is functionally dependent on X  
- Types: Full, Partial, Transitive

3. Normalization:  
- 1NF: Atomic values  
- 2NF: No partial dependencies  
- 3NF: No transitive dependencies  
- BCNF: Superkey-based FDs only

4. Relational Algebra Operators:  
- σ (Select), π (Project), ρ (Rename)  
- ∪, ∩, − (Set operations)  
- ⨝ (Joins): Natural, Equi, Theta  
- ÷ (Division): For “all” queries

5. Extended Operators:  
- Aggregate functions (SUM, AVG, etc.)  
- Grouping, Outer joins

6. Mapping ER to Relations:  
- Regular entity → table with PK  
- Weak entity → add owner’s PK  
- 1:1 → FK in one table  
- 1:N → FK in N-side  
- M:N → new table

# Module 3: Normalization & SQL

1. Functional Dependencies and Design:  
- Aim to eliminate redundancy and anomalies.

2. Normal Forms:  
- 1NF → Atomic values  
- 2NF → No partial dependency  
- 3NF → No transitive dependency  
- BCNF → Every determinant is a superkey  
- 4NF → No multivalued dependencies  
- 5NF → No join dependencies

3. SQL Basics:  
- CREATE, ALTER, DROP  
- Constraints: PRIMARY KEY, UNIQUE, CHECK, REFERENCES

4. SQL Queries:  
- SELECT, WHERE, ORDER BY, DISTINCT  
- LIKE, BETWEEN, IN

5. JOINs:  
- INNER, OUTER, NATURAL

6. Aggregates and Grouping:  
- GROUP BY, HAVING

7. INSERT, DELETE, UPDATE examples

8. Views, Triggers, Assertions (basics)

# Module 4: Advanced SQL & Transactions

1. NULL and 3-valued Logic:  
- Use IS NULL/IS NOT NULL

2. Nested Queries:  
- Simple and Correlated

3. Set Membership:  
- IN, EXISTS, ANY, ALL

4. JOINs and GROUPING:  
- INNER, OUTER joins; GROUP BY and HAVING

5. CASE and WITH:  
- Use for conditional logic and temporary subqueries

6. Views and Assertions:  
- Virtual tables, complex conditions

7. Triggers:  
- Automatic actions after events

8. Transactions and ACID:  
- Atomicity, Consistency, Isolation, Durability

9. Concurrency Issues:  
- Lost update, dirty read, unrepeatable read

10. Serializability:  
- Conflict and View Serializability

11. Recovery:  
- Logs, Commit, Abort, Undo, Redo

12. Isolation Levels:  
- READ UNCOMMITTED → SERIALIZABLE