Unit 1: Introduction to DBMS and Data Models

1. What is a Database?
A database is an organized collection of related data stored electronically.
Unlike a file system, DBMS reduces redundancy and provides better security and data sharing.

2. Define DBMS and its characteristics.
DBMS is software to store, retrieve, and manage data efficiently.
Features: Data independence, integrity, security, concurrency, and backup.

3. Three levels of data abstraction.
Physical (storage details), Logical (tables/relations), and View (user view).

4. What is Data Independence?
Ability to modify schema at one level without affecting the next.
Types: Logical and Physical.

5. Types of database users.
DBA, Application Programmers, End Users (naive, sophisticated, specialized).

6. Role of DBA.
Controls database design, access, security, backup, and performance.

7. Types of database models.
Hierarchical, Network, Relational, Object-Oriented, ER, and NoSQL models.

8. Logical vs Physical schema.
Logical – structure of data; Physical – storage details.

9. Components of DBMS architecture.
Query Processor, Storage Manager, Transaction Manager, DB Engine, Metadata.

10. Advantages/Disadvantages of DBMS.
✅ Centralized control, security, consistency.
✖ Costly, complex, and slower for small systems.

Unit 2: ER and Extended ER Model

1. Entity, Attribute, Relationship.
Entity – object (Student), Attribute – property (Name), Relationship – link (Enrolls).

2. Types of attributes.
Simple, Composite, Single/Multivalued, Derived, Key attributes.

3. Strong vs Weak entity.
Strong has its own key; Weak depends on another entity.

4. Types of relationships.
One-to-One, One-to-Many, Many-to-Many.

5. Cardinality & Participation.
Cardinality – number of entities in relation;
Participation – total or partial involvement.

6. ER Diagram.
Graphical model showing entities and relationships (e.g., Library system).

7. Specialization & Generalization.
Specialization – dividing entity; Generalization – combining entities.

8. Aggregation & Composition.
Aggregation – "has-a" relationship; Composition – stronger ownership.

9. Mapping cardinality.
Specifies how many entities participate in a relation.

10. ER to Relational model.
Entities → Tables, Attributes → Columns, Relationships → Foreign keys.

Unit 3: Relational Model and Relational Algebra

1. Relational model.
Stores data in tables (relations) with rows (tuples) and columns (attributes).

2. Keys.
Super, Candidate, Primary, Foreign, Alternate — used to identify tuples.

3. Referential integrity.
Ensures foreign key matches a primary key in another table.

4. Relational algebra.
Set of operations to manipulate tables — SELECT, PROJECT, JOIN, etc.

5. Main operations.
SELECT (σ), PROJECT (π), UNION, INTERSECTION, JOIN, CARTESIAN PRODUCT.

6. Theta join & Equi-join.
Theta uses condition (<, >, =); Equi-join uses only equality.

7. Cartesian Product vs Join.
Cartesian combines all tuples; Join combines matching tuples only.

8. View.
A virtual table created from queries; does not store data physically.

9. Tuple & Attribute.
Tuple – row; Attribute – column in a relation.

10. NULL values.
Represent missing/unknown data; handled using IS NULL or default values.

Unit 4: SQL — Structured Query Language

1. What is SQL?
Language to create, query, and manage databases.

2. DDL, DML, DCL, TCL.
DDL – structure; DML – data; DCL – control access; TCL – transactions.

3. Create/Alter Table.

```
CREATE TABLE student(...);
ALTER TABLE student ADD age INT;
```

4. Primary/Foreign Key.
Primary – unique identifier; Foreign – references another table.

5. Salary > 50,000 query.

```
SELECT * FROM employee WHERE salary > 50000;
```

6. Subquery.
Query inside another query.
Example: `SELECT * FROM emp WHERE salary > (SELECT AVG(salary) FROM emp);`

7. Aggregate functions.
SUM, AVG, MAX, MIN, COUNT.

8. GROUP BY & HAVING.
GROUP BY groups rows; HAVING filters groups.

```
SELECT dept, AVG(sal) FROM emp GROUP BY dept HAVING AVG(sal)>50000;
```

9. View.

```
CREATE VIEW high_salary AS SELECT * FROM emp WHERE sal>50000;
```

10. Trigger.
Procedure that runs automatically on events (INSERT, UPDATE, DELETE).

11. 2nd highest salary.

```
SELECT MAX(salary) FROM emp WHERE salary < (SELECT MAX(salary) FROM emp);
```

12. Prime check PL/SQL block.

Loop from 2 to n/2; check divisibility.

13. Views for security.

Views hide sensitive columns from users.

14. Normalization steps.

Split large table into smaller ones to remove redundancy.

15. Example of trigger/procedure.

Trigger to log salary updates; Procedure to calculate bonus.

16. Create student table.

```
CREATE TABLE student(id INT PRIMARY KEY, name VARCHAR(20), age INT
CHECK(age>0));
```

17. JOIN example.

```
SELECT s.name, c.course FROM student s JOIN course c ON s.cid=c.id;
```

Unit 5: Normalization and Functional Dependencies

1. Functional dependency.

If A → B, B depends on A (e.g., RollNo → Name).

2. Need for normalization.

Removes redundancy and anomalies.

3. Types of anomalies.

Insertion, Deletion, Update anomalies.

4. 1NF, 2NF, 3NF.

1NF – atomic values,
2NF – remove partial dependency,
3NF – remove transitive dependency.

5. BCNF.

Stronger than 3NF; each determinant is a candidate key.

6. Decomposition.

Breaking table into smaller relations; must be lossless and dependency-preserving.

7. Multivalued dependency & 4NF.

If A →→ B, then B is multivalued; 4NF removes such dependencies.

8. Denormalization.
Combining tables to improve query performance.

9. Redundancy effect.
Increases storage and inconsistency chances.

10. Convert to 3NF.
Identify FDs → remove partial and transitive dependencies.

Unit 6: Transaction Management and Concurrency Control

1. Transaction.
A sequence of database operations as a single logical unit.

2. ACID properties.
Atomicity, Consistency, Isolation, Durability.

3. States of transaction.
Active → Partially Committed → Committed/Failed → Terminated.

4. Concurrency control.
Manages simultaneous transactions to avoid conflicts.

5. Schedule & Serializability.
Schedule – order of operations; Serializability – result same as serial execution.

6. Deadlock.
Two transactions wait for each other; prevented using timeouts or wait-die schemes.

7. Two-Phase Locking (2PL).
Growing phase (acquire locks), shrinking phase (release locks).

8. Shared vs Exclusive lock.
Shared – read only; Exclusive – read/write.

9. Recovery techniques.
Log-based, checkpoint-based recovery.

10. Log-based recovery.
Maintains log of all operations for undo/redo after failure.

11. Transaction precautions.
Use proper commit/rollback and locking mechanisms.

Unit 7: Advanced Topics (PL/SQL / Indexing / NoSQL)

1. PL/SQL.
Procedural extension of SQL with loops, conditions, and blocks.

2. Structure of PL/SQL block.
`DECLARE` → `BEGIN` → `EXCEPTION` → `END;`

3. Cursors.
Pointers to query result sets. Types: Implicit and Explicit.

4. Indexing.
Improves query speed by reducing data scans.

5. Types of indexes.
Primary, Secondary, Clustered, Non-clustered.

6. B-tree & B+ tree.
Balanced structures used for fast searching and indexing.

7. NoSQL databases.
Non-relational; stores data as documents, key-value pairs (e.g., MongoDB).

8. Relational vs Non-relational.
Relational – structured, fixed schema;
Non-relational – unstructured, flexible schema.

9. Stored Procedures.
Predefined SQL code stored in database for reuse.

10. Exception Handling in PL/SQL.
Handled using `EXCEPTION` block with `WHEN others THEN ...`