Chapter 1: Introduction: Databases and Database Users

- Types of Databases & Database Applications:
  - Banking, University, Airlines, E-commerce, Social media
  - Centralized, Distributed, Cloud, NoSQL

- Basic Definitions:
  - Data: Raw facts
  - Information: Meaningful data
  - Database: Organized data collection
  - DBMS: Software managing databases
  - Database System: DB + DBMS + users + applications

- Typical DBMS Functionality:
  - Define, store, retrieve, update data
  - Access control, backup, concurrency

- Example: University DB with STUDENT, COURSE, ENROLL

- Main Characteristics of DB Approach:
  - Self-describing, insulation, data abstraction
  - Multiple views, data sharing, control of redundancy

- Types of Database Users:
  - DBA, Designers, Developers, End users

- Advantages of DB Approach:
  - Reduced redundancy, improved sharing, security
  - Backup, recovery, data independence

- When Not to Use Databases:
  - Small/simple data, single user, cost or speed constraints

Exam Tips:

- Understand differences between database applications and general applications.
- Know main DBMS functions and user roles.
- Be able to explain advantages and limitations of database approach.

Chapter 2: Database System Concepts and Architecture

- Data Models and Categories:
  - High-level (Conceptual): ER model
  - Logical (Representational): Relational model
  - Physical (Low-level): Storage details

- History of Data Models:
  - Hierarchical, Network, Relational, Object-Oriented/NoSQL

- Schemas, Instances, and States:
  - Schema: Structure
  - Instance/State: Data at a moment

- Three-Schema Architecture:
  - External (user view), Conceptual (logical), Internal (physical)

- Data Independence:
  - Logical: schema changes don't affect apps
  - Physical: storage changes don't affect logical design

- DBMS Languages and Interfaces:
  - DDL, DML, DCL, SQL
  - Menu-based, query-based, app interfaces

- Database Utilities:
  - Backup, recovery, indexing, monitoring

- Architecture:

  - Centralized vs Client-Server

- Classification of DBMS:
  - Based on data model, users, sites, cost, usage

Exam Tips:
- Know different data models and their levels.
- Understand schemas and instances.
- Be able to explain three-schema architecture and data independence.
- Know DBMS languages and common utilities.

Chapter 3: Data Modeling Using the Entity-Relationship (ER) Model

- Database Design Process:
  - Requirements analysis, conceptual (ER), logical, physical, implementation

- Example COMPANY Database:
  - Entities: Employee, Department, Project
  - Relationships: Works_On, Manages, Controls

- ER Model Concepts:
  - Entity: real-world object
  - Attribute: property of entity
  - Relationship: link between entities

- Entities and Attributes:
  - Simple, Composite, Derived, Multivalued attributes

- Entity Types, Key Attributes, Value Sets

- Relationships and Cardinality (1:1, 1:N, M:N)

- Weak Entities: dependent on strong entities

- Roles and Attributes in Relationships

- ER Diagram Notation: rectangles, ellipses, diamonds

- Alternative Notations: UML class diagrams

Exam Tips:
- Be able to identify entities, attributes, and relationships.
- Understand weak entities and relationship roles.
- Practice drawing ER diagrams using correct notation.

Chapter 4: Enhanced Entity-Relationship (EER) Modeling

- EER extends ER with advanced modeling concepts.

- Subclasses and Superclasses:
  - Specialization: create subclasses from a superclass
  - Generalization: combine entities into a superclass

- Categories (Union Types): subclasses with multiple superclasses.

- Attribute and Relationship Inheritance:
  - Subclasses inherit attributes and relationships from superclasses.

- EER adds object-oriented concepts like inheritance.

- Useful for modeling complex real-world applications more accurately.

Exam Tips:
- Know the difference between specialization and generalization.
- Understand how inheritance works in EER.
- Be able to draw EER diagrams with subclasses and superclasses.

Chapter 5: The Relational Data Model and Relational Database Constraints

- Relational Model Concepts:
  - Data stored in tables (relations) with rows (tuples) and columns (attributes).

- Relational Constraints:
  - Domain constraints: attribute values from correct domain
  - Key constraints: primary key uniqueness
  - Entity integrity: primary key not NULL
  - Referential integrity: foreign key matches primary key or NULL

- Relational Schemas:
  - Define table structure, attributes, keys.

- Update Operations:
  - Insert, delete, update

- Dealing with Constraint Violations:
  - Reject or fix updates that break constraints

Exam Tips:
- Know how tables represent entities and relationships.
- Understand primary and foreign keys and integrity constraints.
- Be able to explain update operations and constraint enforcement.

Chapter 8: Functional Dependencies and Normalization for Relational Databases

- Informal Design Guidelines:
  - Clear attribute meanings, avoid redundancy and anomalies
  - Handle null values and spurious tuples

- Functional Dependencies (FDs):
  - X -> Y means knowing X determines Y
  - Armstrong's axioms: reflexivity, augmentation, transitivity

- Normal Forms:

  - 1NF: atomic values, no repeating groups
  - 2NF: no partial dependencies on part of composite key
  - 3NF: no transitive dependencies
  - BCNF: every determinant is a superkey

- Keys:
  - Candidate keys, primary keys, superkeys

- Normalization:
  - Process to reduce redundancy and update anomalies

Exam Tips:

- Practice identifying functional dependencies.
- Understand each normal form and how to apply normalization.
- Know definitions of keys and their roles in normalization.