# 1. Database-System Applications – Overview

**Definition:**
A **database system** is a collection of interrelated data and a set of programs to access and manage that data efficiently.

**Main Purpose:**
To store, retrieve, and manage data in a **structured and consistent** way.

**Applications of Database Systems:**

| Area | Example |
|---|---|
| Banking | Transactions, customer records, account management |
| Education | Student info, results, attendance |
| Airlines | Reservations, schedules |
| E-commerce | Product catalog, user profiles, orders |
| Hospitals | Patient info, doctor schedule |
| Social Media | User data, posts, messages |

**Key Points to Remember:**

- Databases provide **data integrity**, **security**, and **concurrency control**.
- Databases reduce **data redundancy** and improve **data consistency**.

---

# 2. Purpose of Database Systems

**Main Goals:**

1. **Data Independence:** Changes in structure shouldn't affect application programs.
2. **Data Sharing:** Multiple users can access the same data simultaneously.
3. **Data Security:** Prevent unauthorized access.
4. **Data Integrity:** Ensure data accuracy and consistency.
5. **Data Recovery:** Recover data after system failure.

---

# 3. View of Data

Databases use **multiple levels of abstraction** to simplify data handling:

| Level | Description | Example |
|---|---|---|
| **Physical level** | How data is stored in memory/disks | Data blocks, indexes |
| **Logical level** | What data is stored and relationships | Tables, attributes |

| Level | Description | Example |
|---|---|---|
| **View level** | How users see data | Student view, teacher view |

 Example:
A college database might have a table `Student(RollNo, Name, Dept, Marks)`.

- Physical: Stored in disk blocks
- Logical: Defined as a table
- View: Students only see their marks, admins see all data.

---

# 💬 4. Database Languages

### a. Data Definition Language (DDL)

Used to define the structure of a database.
**Example:**

```
CREATE TABLE Student (
   RollNo INT PRIMARY KEY,
   Name VARCHAR(50),
   Dept VARCHAR(30)
);
```

### b. Data Manipulation Language (DML)

Used to insert, delete, update, or query data.
**Example:**

```
INSERT INTO Student VALUES (101, 'Sharif', 'CS');
```

### c. Data Control Language (DCL)

Used for permissions and access control.
**Example:**

```
GRANT SELECT ON Student TO User1;
```

### d. Transaction Control Language (TCL)

Used to manage transactions.
**Example:**

```
COMMIT;   -- Save changes
ROLLBACK; -- Undo changes
```

---

# 📓 5. Relational Databases: Database Design

A **relational database** stores data in **tables (relations)**.

| Concept | Description |
|---|---|
| **Table (Relation)** | A set of tuples (rows) and attributes (columns). |
| **Tuple** | A single record (row). |
| **Attribute** | A data field (column). |
| **Primary Key** | Unique identifier for a record. |

 Example:

```
Student(RollNo, Name, Dept, Marks)
```

---

# 🏛 6. Database Architecture

Database systems follow a **3-tier architecture**:

1. **Internal Level** – Physical storage of data.
2. **Conceptual Level** – Logical structure of the whole database.
3. **External Level** – Individual user views.

**Advantages:**

- Data abstraction
- Security
- Flexibility

 Some systems also follow a **3-tier application architecture**:

- **Presentation Tier** (UI)
- **Application Tier** (Server logic)
- **Database Tier** (Data storage)

---

# 💎 7. Data Mining and Information Retrieval

## Data Mining

Process of extracting *useful patterns* or *knowledge* from large datasets.

**Examples:**

- Predicting customer behavior
- Fraud detection
- Recommendation systems (like Netflix)

**Techniques:** Classification, Clustering, Association rules (like “People who buy bread often buy butter”).

---

### Information Retrieval (IR)

Finding *unstructured data* (like text, documents, web pages).

**Example:**
Google search engine retrieves web pages based on keywords.

---

# □ 8. Overview of the Design Process

Database design goes through **several stages:**

1. **Requirement Analysis** → Gather user needs
2. **Conceptual Design** → Create ER diagram
3. **Logical Design** → Convert ER model to relational schema
4. **Physical Design** → Define indexes, storage
5. **Implementation** → Create tables, relations in SQL

---

# □ 9. The Entity-Relationship (ER) Model

Used for **conceptual design** of databases.

**Basic Concepts:**

| Concept | Symbol | Example |
|---|---|---|
| **Entity** | Rectangle | Student |
| **Attribute** | Oval | Name, Age |
| **Relationship** | Diamond | Enrolls |
| **Primary Key** | Underlined | RollNo |

□ Example:
Student —(Enrolls)—> Course

---

# ⚖ 10. Constraints

**Constraints** are rules applied to ensure **data integrity**.

| Type | Meaning | Example |
|---|---|---|
| **Primary Key** | Unique ID | RollNo |
| **Foreign Key** | Links tables | DeptID in Student refers to Department table |
| **NOT NULL** | Field cannot be empty | Name NOT NULL |
| **CHECK** | Restrict values | CHECK (Marks >= 0) |

---

# □ 11. Removing Redundant Attributes in Entity Sets

**Redundant attributes** are **duplicate or unnecessary** attributes in an entity.
They waste storage and can cause inconsistency.

□ Example:
If both `TotalMarks` and `Marks1 + Marks2 + Marks3` are stored, `TotalMarks` is redundant.

**Solution:**
Remove computed or duplicated attributes.
Use **normalization** to eliminate redundancy.

---

# 🖉 12. Entity-Relationship Diagrams (ER Diagrams)

A **graphical representation** of entities, attributes, and relationships.

**Symbols:**

- Rectangle → Entity
- Oval → Attribute
- Diamond → Relationship
- Line → Connection

□ Example:

```
Student ──(Enrolls)── Course
```

Where:

- Student has attributes: RollNo, Name, Dept
- Course has attributes: CourseID, Title

**Use:**
Helps in **visualizing and designing** the database before actual implementation.

---

# 🏁 Summary for Exam

| Topic | Key Point |
|---|---|
| Database Purpose | Manage data efficiently and securely |
| Data View | Physical, Logical, View levels |
| DB Languages | DDL, DML, DCL, TCL |
| Relational Design | Tables, Keys, Relationships |
| Architecture | 3-tier data abstraction |
| Data Mining | Discovering patterns in data |
| Information Retrieval | Searching unstructured data |
| ER Model | Conceptual design tool |
| Constraints | Maintain data integrity |
| Redundancy | Remove duplicates via normalization |
| ER Diagram | Visual representation of schema |