UNIT -1

INTRODUCTION TO DATABASES AND DATA MODELING

Data Models

A **data model** is a conceptual framework that describes how data is **structured**, **stored**, and **manipulated** in a database. It provides rules and standards for representing real-world entities and relationships inside a DBMS.

Data models help in:

- Organizing data logically
- Ensuring consistency and integrity
- Enabling communication between designers, developers, and users
- Guiding the database design process

Types of Data Models

1. Hierarchical Data Model

- Organizes data in a **tree-like structure** (parent–child).
- Each child has **only one parent** (1:N relationship).
- Data is accessed through **navigational pointers**.

Features

- Fast access for hierarchical relationships
- Simple structure

Limitations

- No support for many-to-many relationships
- Changes in structure require major redesign
- Not flexible

Example: File systems, IBM IMS.

---

2. Network Data Model

- Data is organized as a **graph** structure (records and sets).
- A child can have **multiple parents** (M:N relationship possible).

Features

- More flexible than hierarchical model
- Supports complex relationships

Limitations

- Complex to design
- Navigational data access → difficult for users

Example: CODASYL DBTG model.

---

3. Relational Data Model (RDM)

(Most widely used model)

- Represents data in **tables (relations)**.
- Each table has **tuples (rows)** and **attributes (columns)**.
- Uses **keys**, **constraints**, and **relationships**.

Features

- High flexibility
- Easy to understand (table form)
- Uses **SQL** for operations
- Ensures **data integrity**

Constraints used

- Primary key
- Foreign key
- Unique
- Not null
- Check

Example: MySQL, Oracle, SQL Server, PostgreSQL.

---

4. Entity–Relationship (ER) Model

- A high-level, **conceptual model** used in database design.
- Represents the real world as:
  - **Entities** (objects)
  - **Attributes** (properties)
  - **Relationships** (associations between entities)

Features

- ER diagrams help visualize the design
- Easy communication between designers and users
- Later converted to a relational model

---

5. Extended ER (EER) Model

Adds advanced features to ER:

- **Specialization**
- **Generalization**
- **Aggregation**
- **Category (union type)**

Useful when modelling complex real-world applications.

---

6. Object-Oriented Data Model

- Combines database concepts with object-oriented principles.
- Stores data as **objects** with:
  - Attributes
  - Methods
  - Encapsulation

Features

- Supports complex data types
- Good for multimedia, CAD, engineering applications

---

7. Object-Relational Data Model

- Hybrid model combining **relational** and **object-oriented** features.
- Supports:
  - User-defined types
  - Inheritance
  - Complex structures
  - Large objects (LOB)

Example: Oracle ORDBMS, PostgreSQL.

---

8. Physical Data Model

- Describes **how data is stored** in hardware.
- Includes:
  - File organization
  - Indexing

- Record placement
- Storage structures

Used by database administrators (DBAs) for performance tuning.