Study Material

Computer Science

As per UG syllabus w. e. f session 2020-2021

University of Kalyani

Semester III

Course Name: Database Management Systems

Course Code: COM.SC-G-CC-L-301C

Prepared by

Priya Das (SACT)

Department of Computer Science

Chakdaha College

Data Model

A data model in a Database Management System (DBMS) is a conceptual framework for organizing and defining how data is stored, managed, and accessed within a database system. It provides a structure for data and a set of operations for managing it.

Types of Data Models:

1. Hierarchical Data Model:

- Organizes data in a tree-like structure.
- Each record (node) has a parent/child relationship.
- Example: File systems, where directories contain subdirectories and files.

  Advantages: Fast traversal for hierarchical data.

  Disadvantages: Limited flexibility; difficult to represent complex relationships.

2. Network Data Model:

- Uses a graph structure, with records represented as nodes and relationships as edges.
- Allows many-to-many relationships.

  Advantages: More flexible than the hierarchical model.

  Disadvantages: Complex to implement and query.

3. Relational Data Model:

The Relational Data Model is one of the most widely used data models in database management systems. It organizes data into structured tables (relations) consisting of rows and columns, enabling efficient storage, retrieval, and manipulation of data.

This model ensures data integrity and consistency through constraints like **entity integrity** (primary keys cannot be null) and **referential integrity** Data manipulation is performed using relational algebra operations like selection, projection, union, and join, often implemented through SQL.

Advantages:

- **Simplicity**: Easy to understand and implement using tabular structures.
- **Flexibility**: Data can be easily queried and manipulated using SQL.

- **Data Integrity**: Constraints ensure data accuracy and consistency.

Disadvantages: May become inefficient for very complex relationships.

4. Entity-Relationship (ER) Model:

The **Entity-Relationship (ER) Model** is a high-level conceptual data model used to design and represent the structure of a database. It provides a visual way to define the data, their attributes, and the relationships among them. The model is typically represented using an **Entity-Relationship Diagram (ERD)**.

Key Components of ER Model

- **Entities**: An entity is a real-world object with a distinct identity represented as rectangles in an ER diagram.
  Example: student

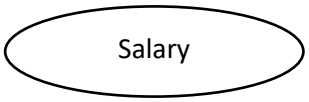
- **Attributes**: An attribute is a property or characteristic of an entity that is used to describe an entity. Attribute represented as ellipses in an ER diagram.

Types of Attributes

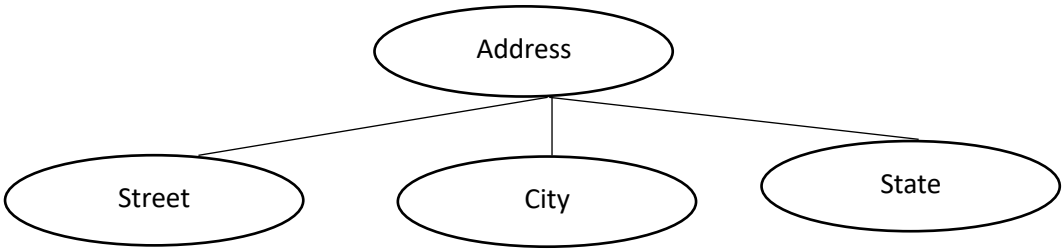
- **Simple attributes**

  These are independent qualities that cannot be further deconstructed; they are also referred to as atomic attributes.

  Example: employee ID and salary.



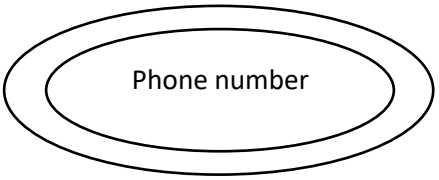
- **Composite attributes**
  These are attributes that are made up of multiple simple attributes.
  Example: An employee's address, which can be split down by street, city, state.



- **Multi-valued attributes**

For every instance of an entity, these attributes may have several values.

Example: Email addresses, phone numbers.



- **Derived attributes**

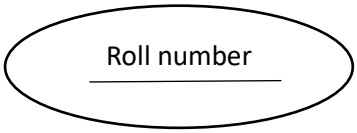
These attributes are calculated or derived using other attributes in the database.

Example: Age is derived from date of birth.

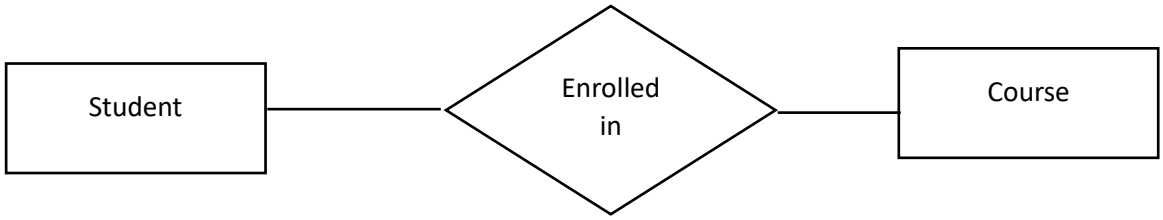
- **Key attributes**

These attributes can identify an entity uniquely in an entity set.

Examples: roll numbers.



- **Relationships:** A relationship is an association between two or more entities that defines how they are connected. Relationships help to structure the database by linking related data stored in different tables.



Types of Relationships in DBMS

1. One-to-One (1:1) Relationship
2. One-to-Many (1:M) Relationship
3. Many-to-Many (M:N) Relationship

Advantages:

- **Clear Representation**: Provides a visual and structured way to model the database.
- **Ease of Conversion**: Can be easily translated into relational schemas

Disadvantages: Does not directly address data manipulation or implementation details.