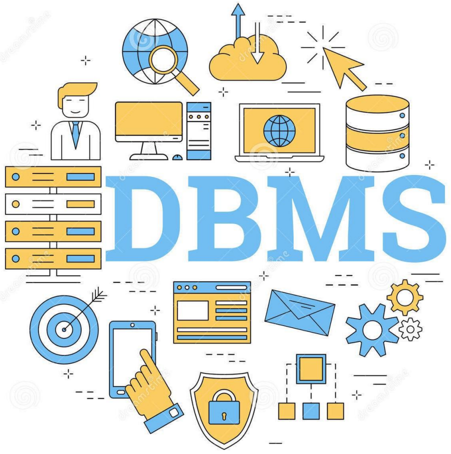
Unit I - Introduction

Purpose of Database System - Views of data – Data models, Database Management system - Three-schema architecture of DBMS, Components of DBMS. **Entity –Relationship Model - Conceptual data modelling - motivation**, entities, entity types, attributes, relationships, relationship types, E/R diagram notations, Examples



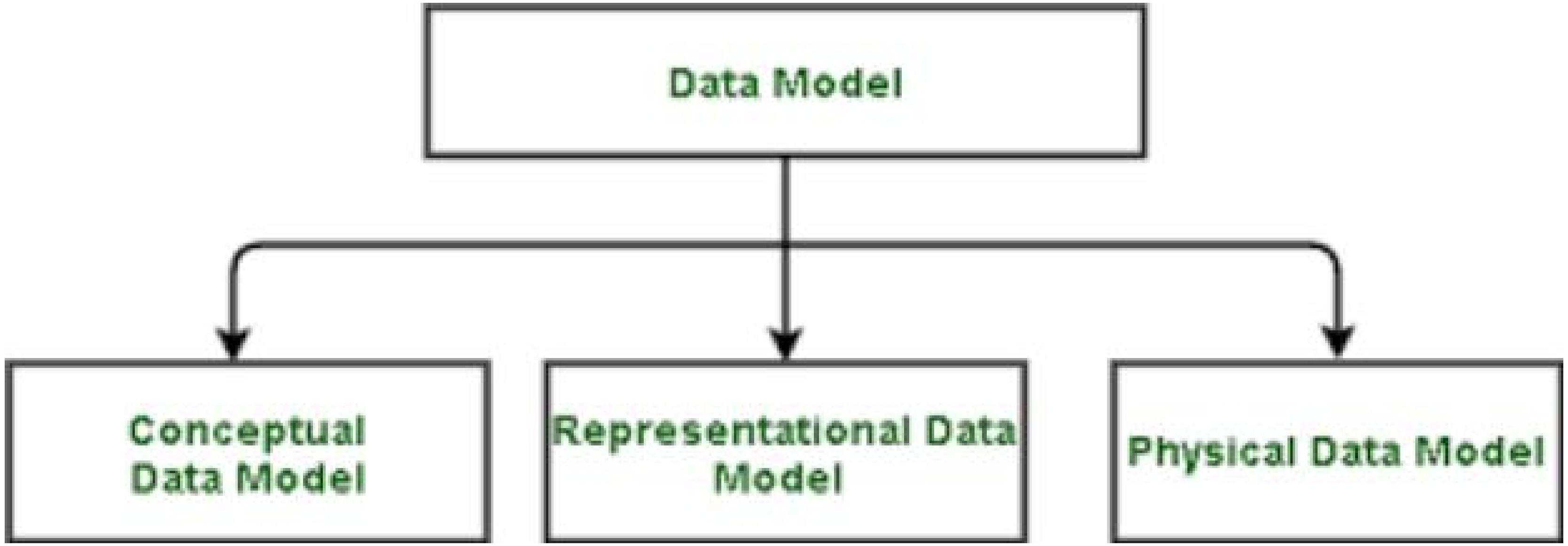
Recap

- **Data Modelling** - process of creating a data model for the data to be stored in a database.
- Two Types
  - ER
  - Relational etc.,

Design Phases

- Initial phase -- characterize fully the data needs of the prospective database users.
- Second phase -- choosing a data model
- Final Phase -- Moving from an abstract data model to the implementation of the database
  - Logical Design - Deciding on the database schema.
    - **Redundancy**
    - **Incompleteness**
  - Physical Design - Deciding on the physical layout of the database

Types of Relational Models



Conceptual data modelling

- **Conceptual Data Model**
  - understand the needs or requirements of the database
  - Requirement-gathering process i.e. before the Database Designers start making a particular database.
  - Popular Model – Entity Relationship Model

Entity-Relationship Model(ER Model): It is a high-level data model which is used to define the data and the relationships between them. It is basically a conceptual design of any database which is easy to design the view of data.

Conceptual data modelling

1. **Entity:** An entity is referred to as a real-world object. It can be a name, place, object, class, etc. These are represented by a **rectangle** in an ER Diagram.

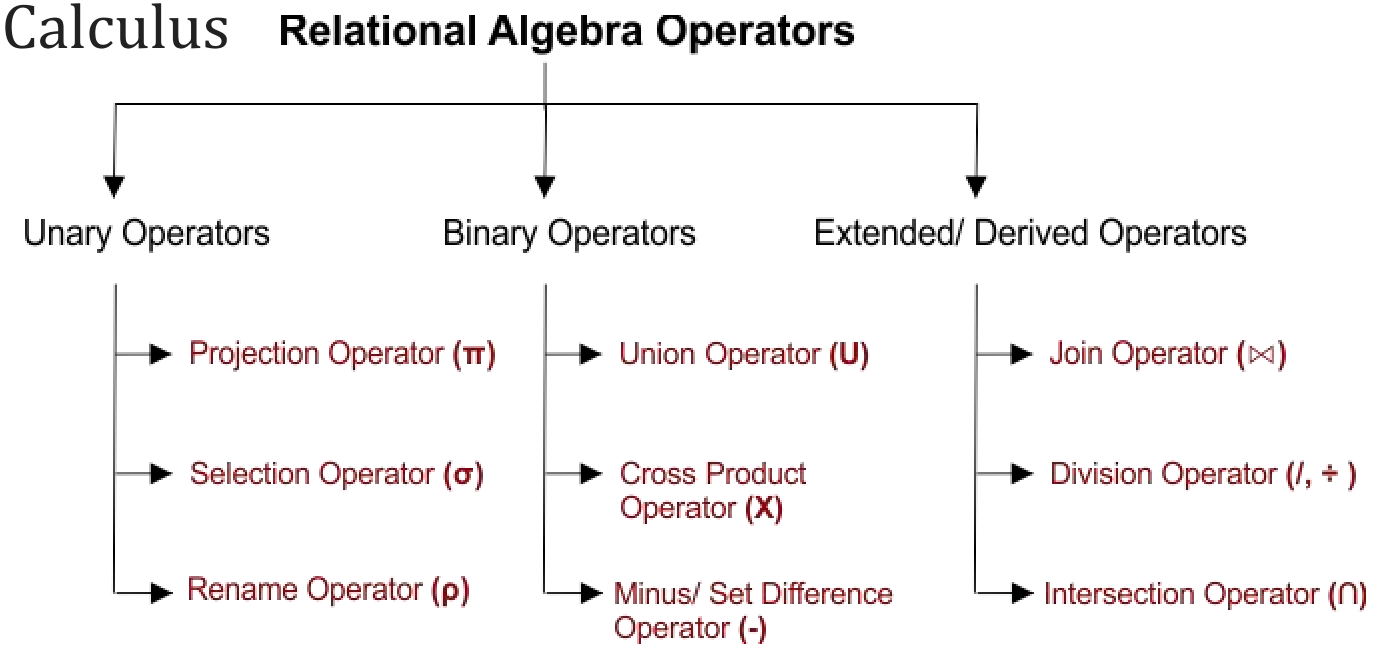
2. **Attributes:** An attribute can be defined as the description of the entity. These are represented by **Eclipse** in an ER Diagram. It can be Age, Roll Number, or Marks for a Student.

3. **Relationship:** Relationships are used to define relations among different entities. Diamonds and Rhombus are used to show Relationships.

Representational Data Model

- Used to represent only the logical part of the database and does not represent the physical structure of the database.

- Relational Algebra or Relational Calculus



Relational Algebra Operators

Unary Operators
- Projection Operator (π)
- Selection Operator (σ)
- Rename Operator (ρ)

Binary Operators
- Union Operator (U)
- Cross Product Operator (X)
- Minus/ Set Difference Operator (-)

Extended/ Derived Operators
- Join Operator (⋈)
- Division Operator (/, ÷)
- Intersection Operator (∩)

Physical Data Model

- Used to practically implement Relational Data Model
- Structured Query Language (SQL) is used to practically implement Relational Algebra.

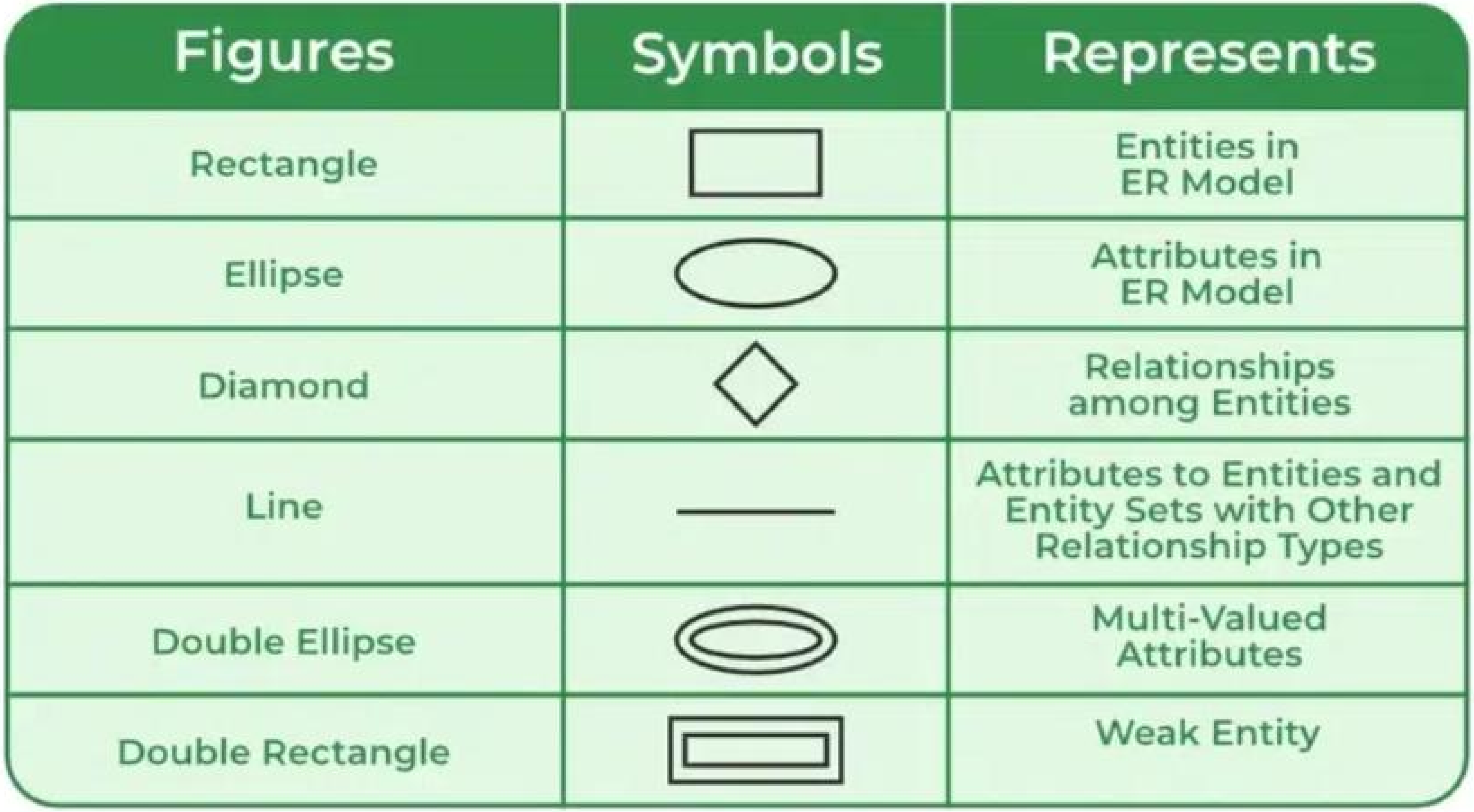
Design Approaches

- Entity Relationship Model
  - Models an enterprise as a collection of *entities* and *relationships*
    - Entity: a “thing” or “object”
      - Described by a set of *attributes*
    - Relationship: an association among several entities
- Represented diagrammatically by an ***entity-relationship diagram***

Why use E-R Diagrams in DBMS?

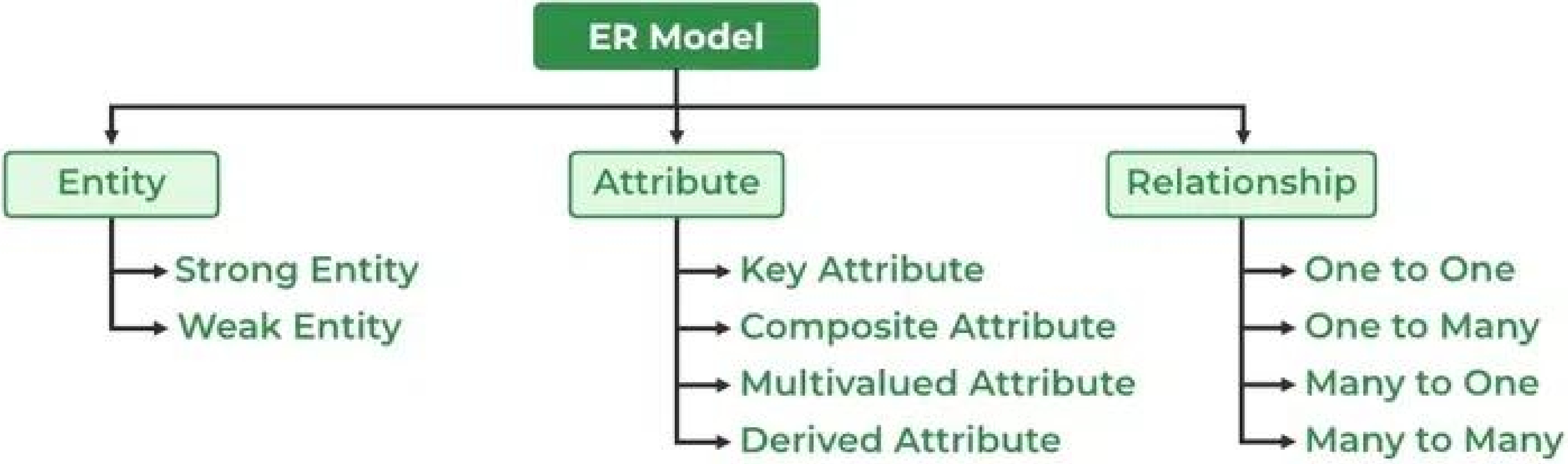
- Makes easy to convert into relations (tables).
- require no technical knowledge and no hardware support.

Symbols used in ER Model



Figures	Symbols	Represents
Rectangle	▭	Entities in ER Model
Ellipse	⬭	Attributes in ER Model
Diamond	◇	Relationships among Entities
Line	—	Attributes to Entities and Entity Sets with Other Relationship Types
Double Ellipse	◎	Multi-Valued Attributes
Double Rectangle	⧈	Weak Entity

Component of ER Model



Motivations

- **Conceptually it is very simple:** ER model is very simple because if we know relationship between entities and attributes, then we can easily draw an ER diagram.
- **Better visual representation:** ER model is a diagrammatic representation of any logical structure of database. By seeing ER diagram, we can easily understand relationship among entities and relationship.
- **Effective communication tool:** It is an effective communication tool for database designer.
- **Highly integrated with relational model:** ER model can be easily converted into relational model by simply converting ER model into tables.
- **Easy conversion to any data model:** ER model can be easily converted into another data model like hierarchical data model, network data model

Thank You!