GLA UNIVERSITY
Accredited with A+ Grade by NAAC
Mathura | Greater Noida

Department of Computer Engineering & Applications

Institute of Engineering & Technology

ER Model and Relational Model

What is a Data Model?

- A framework for organizing data.
- Helps in defining:
  Structure of the database.
  Data types and constraints.
  Relationships among data.

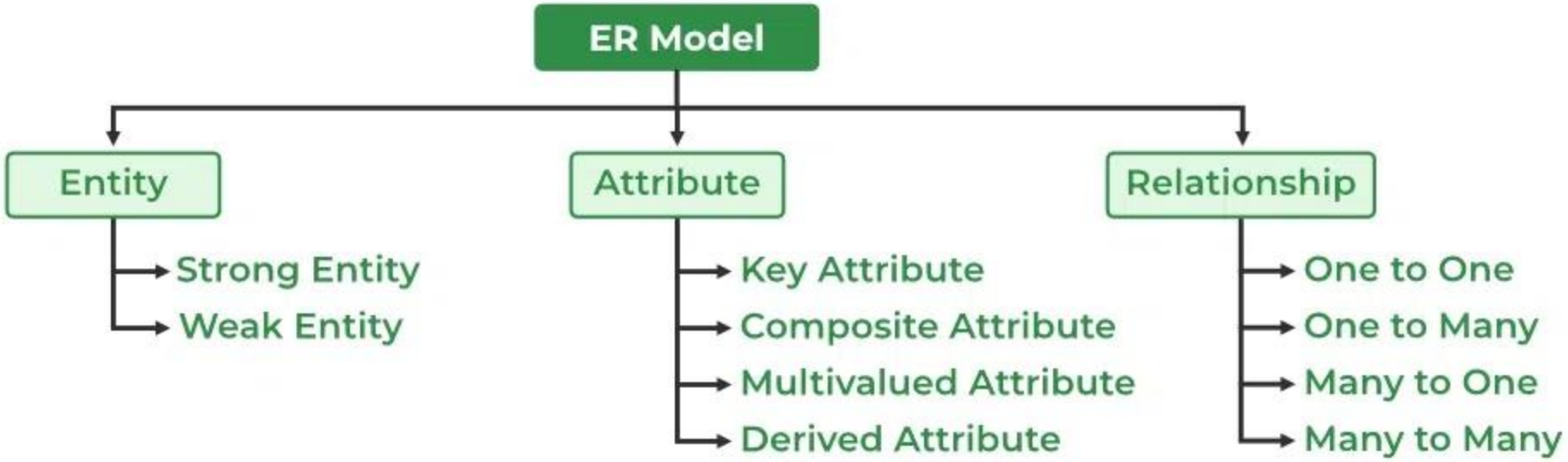
Why Use an ER Model?

- Visual representation of data.
- Easy to understand.
- Helps in designing databases conceptually.
- Bridges communication between technical and non-technical users.

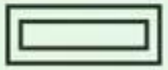
Components of ER Model

- Entities: Objects/things (e.g., Student, Course).
- Attributes: Properties (e.g., StudentName, Age).
- Entity Set: A collection of similar entities.
- Relationships: Associations between entities (e.g., Student enrolls in Course).

Components of ER Model



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

Types of Entities

- Strong Entity – has a primary key.
- Weak Entity – no primary key; depends on strong entity.

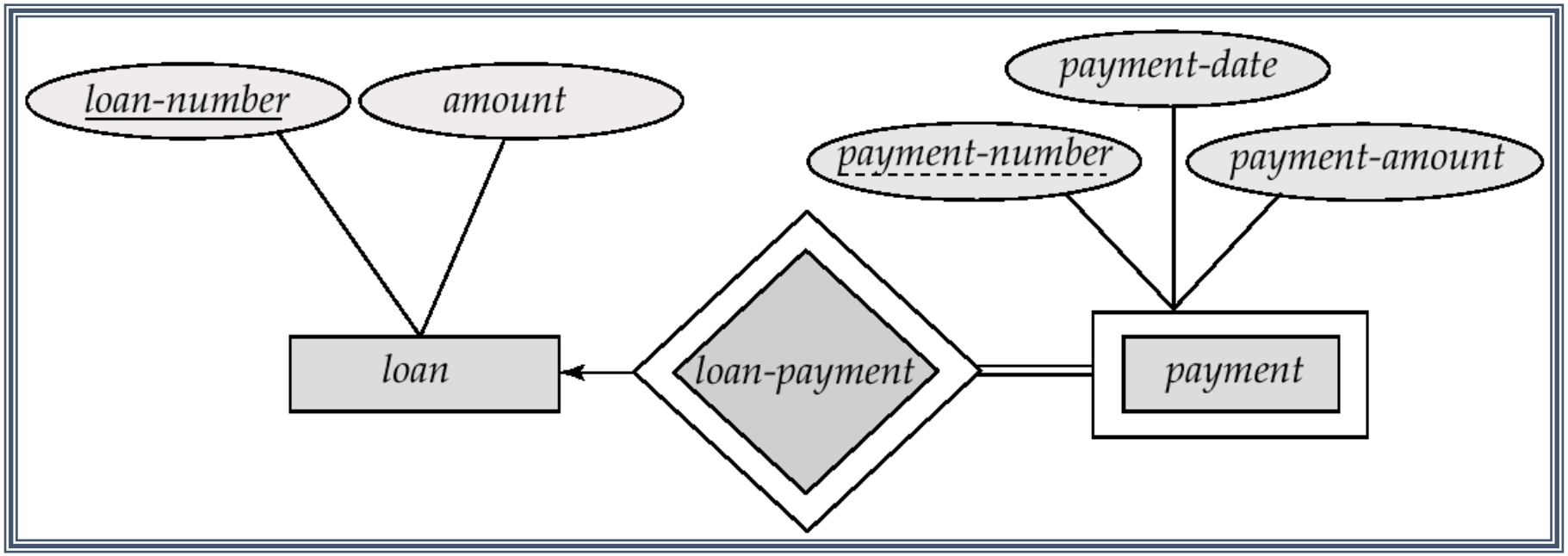
Weak Entity

- *A weak entity* is an entity that is existence-dependent on some other entity. By contrast, *a regular entity* (or “a strong entity”) is an entity which is not weak.
- The existence of a weak entity set depends on the existence of a *identifying entity set*

- A weak entity type can be related to more than one regular entity type.

Weak Entity and Regular/Strong Entity

- We depict a weak entity by double rectangles.
- The identifying relationship is depicted using a double diamond.



Attributes

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

  Example:

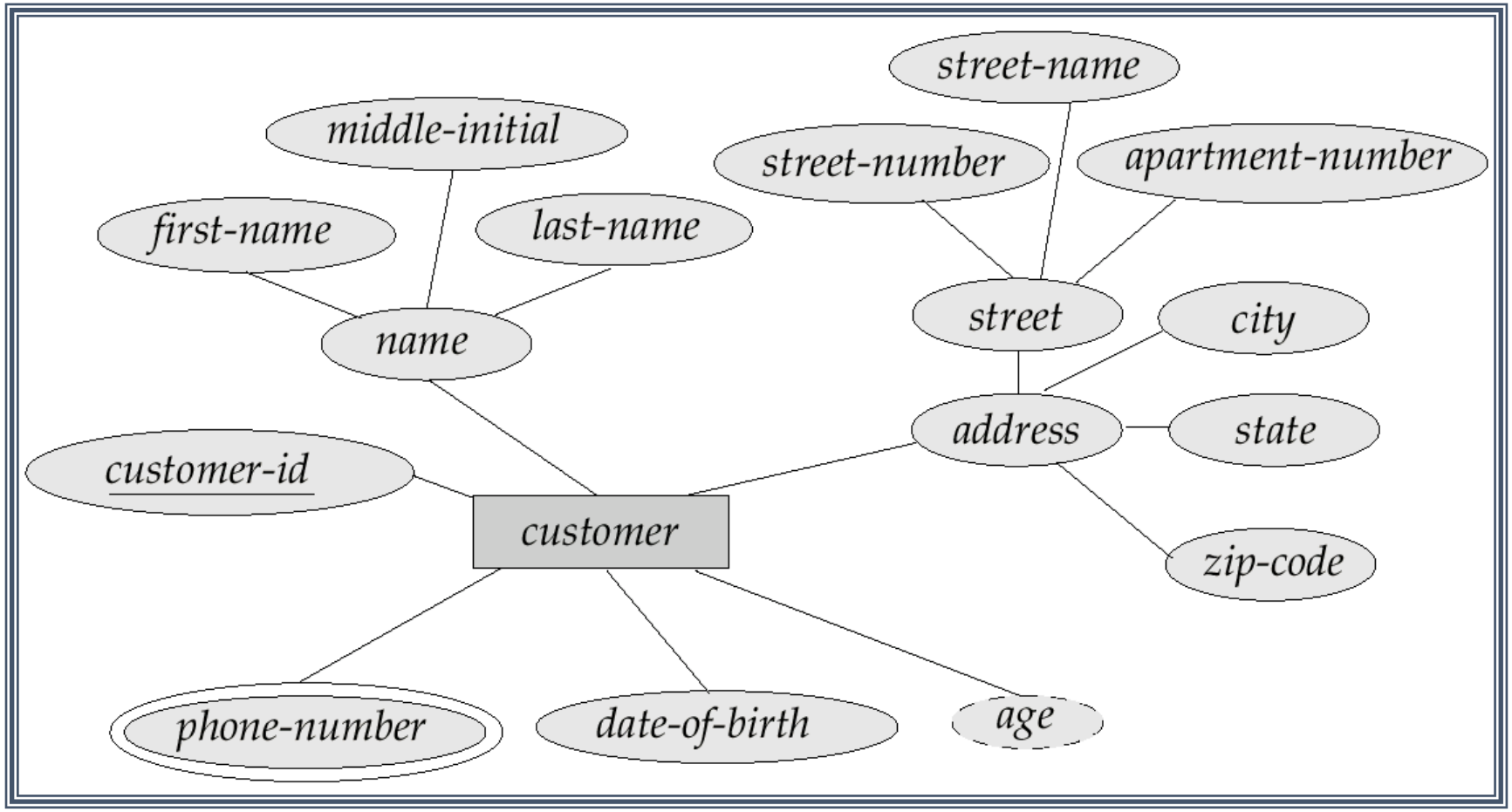
  customer = (customer-id, customer-name, customer-street, customer-city)
  loan = (loan-number, amount)

- *Domain* – the set of permitted values for each attribute

Types of Attributes

- Simple (e.g., Name)
- Composite (e.g., Name → FirstName, LastName)
- Derived (e.g., Age from DOB)
- Multivalued (e.g., PhoneNumbers)

E-R Diagram With Composite, Multivalued, and Derived Attributes

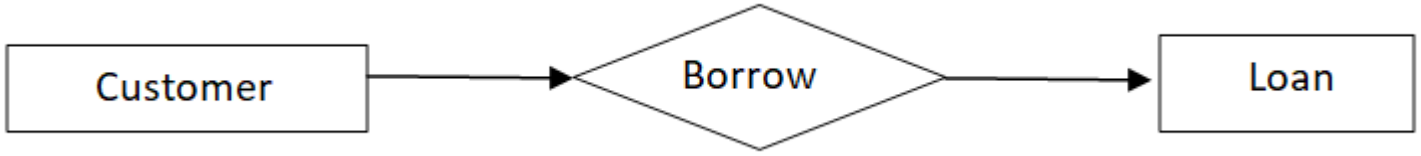


Entity Set

- An *entity* is an object that exists and is distinguishable from other objects.
  - Example: specific person, company, event, plant
- Entities have *attributes*
  - **Example: people have *names* and *addresses***
- An *entity set* is a set of entities of the same type that share the same properties.
  - **Example: set of all persons, companies, trees, holidays**

Relationship Set

- A **Relationship Set** is a set of associations among two or more entity sets in an ER (Entity-Relationship) model.
- **Key Concepts:**
  - **Entity Sets**: Collections of similar types of entities.
  - **Relationship**: Association among entities.
  - **Relationship Set**: Collection of similar relationships.
- **Example:**



Relationship Set

Relationship Set

- **Degree of Relationship:**
  - **Unary**: Relationship among same entity type (e.g., Employee supervises Employee)
  - **Binary**: Between two entities (most common)
  - **Ternary**: Involves three entity sets
- **Participation:**
  - **Total Participation**: Every entity must be involved
  - **Partial Participation**: Some entities may not be involved

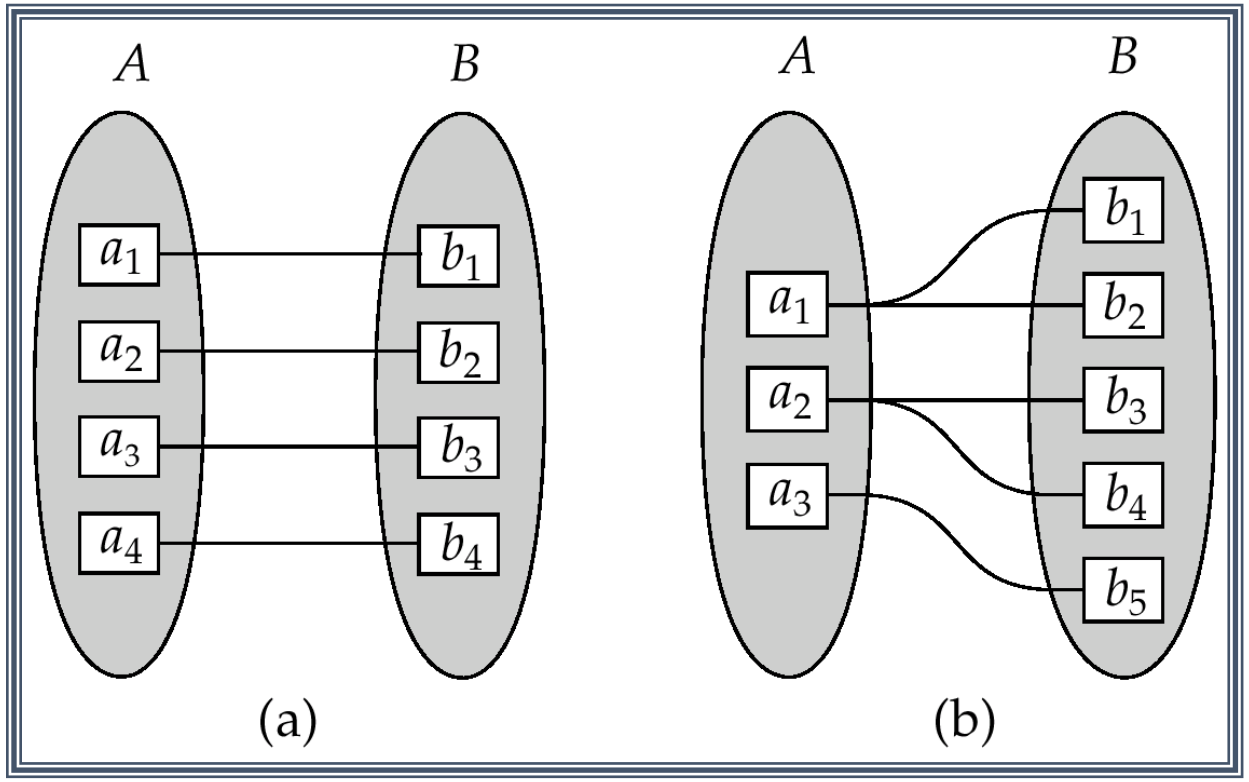
Cardinalities in ER Model

- The maximum number of times an entity of an entity set participates in a relationship set is known as cardinality.

- Express the number of entities to which another entity can be associated via a relationship set.

  - One-to-One (1:1)
  - One-to-Many (1:N)
  - Many-to-One (M:1)
  - Many-to-Many (M:N)

Mapping Cardinalities

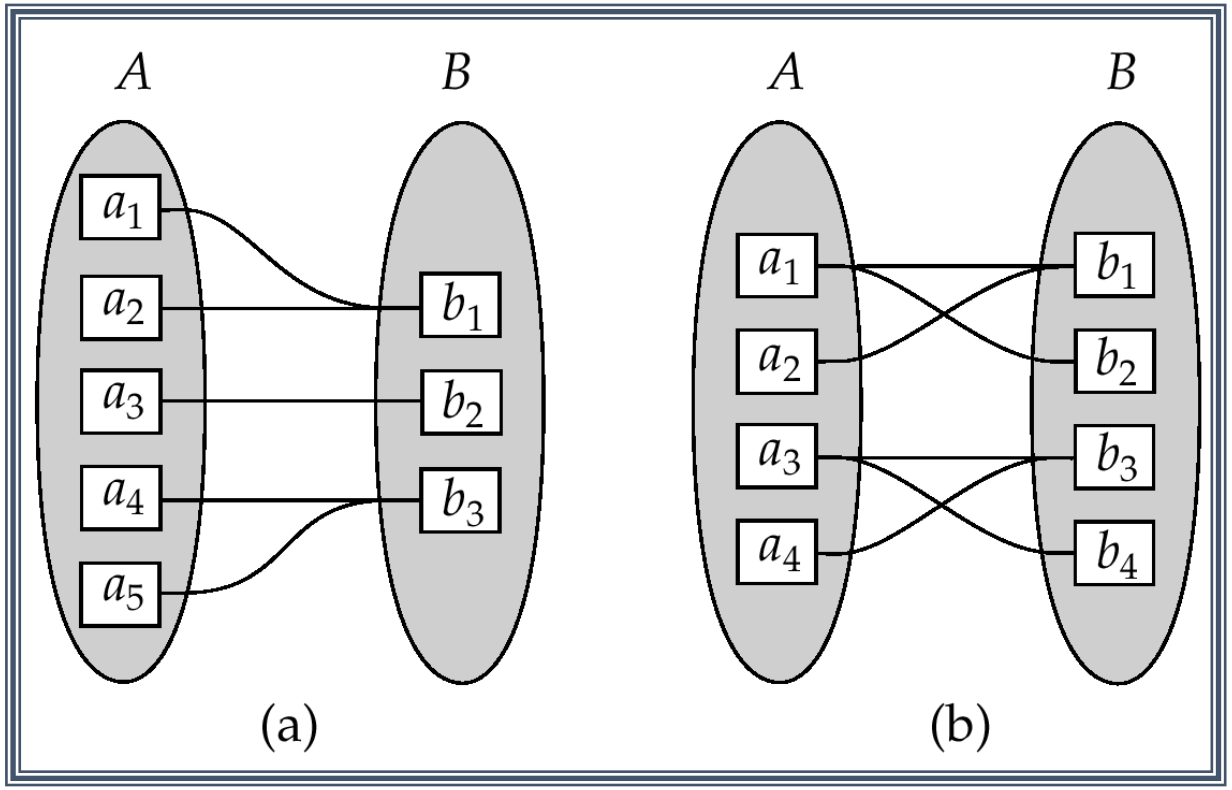


One to one

One to many

Note: Some elements in A and B may not be mapped to any elements in the other set

Mapping Cardinalities



Many to one

Many to many

Note: Some elements in A and B may not be mapped to any elements in the other set

Cardinality: One-to-One (1:1)

- Definition: One entity in set A is related to at most one entity in set B, and vice versa.
- Example:
  - Each person has one passport.
  - Each passport belongs to one person.

Cardinality: One-to-Many (1:N)

- Definition: One entity in set A can be related to many entities in set B, but each entity in B is related to only one in A.
- Example:
  - One department has many employees.
  - Each employee works in one department.

Cardinality: Many-to-Many (M:N)

- Definition: Entities in set A can relate to many entities in set B, and vice versa.
- Example:
  - A student can enroll in many courses.
  - A course can have many students.

Participation Constraint

- It is applied to the entity participating in the relationship set.

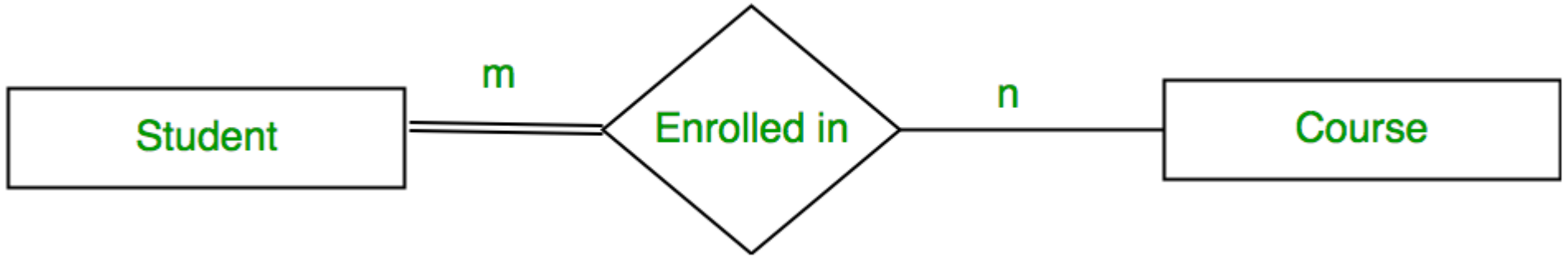
1. **Total Participation**
2. **Partial Participation**

Participation Constraint

1. **Total Participation:**
   - Each entity in the entity set must participate in the relationship.
   - If each student must enroll in a course, the participation of students will be total.
   - Total participation is shown by a double line in the ER diagram.

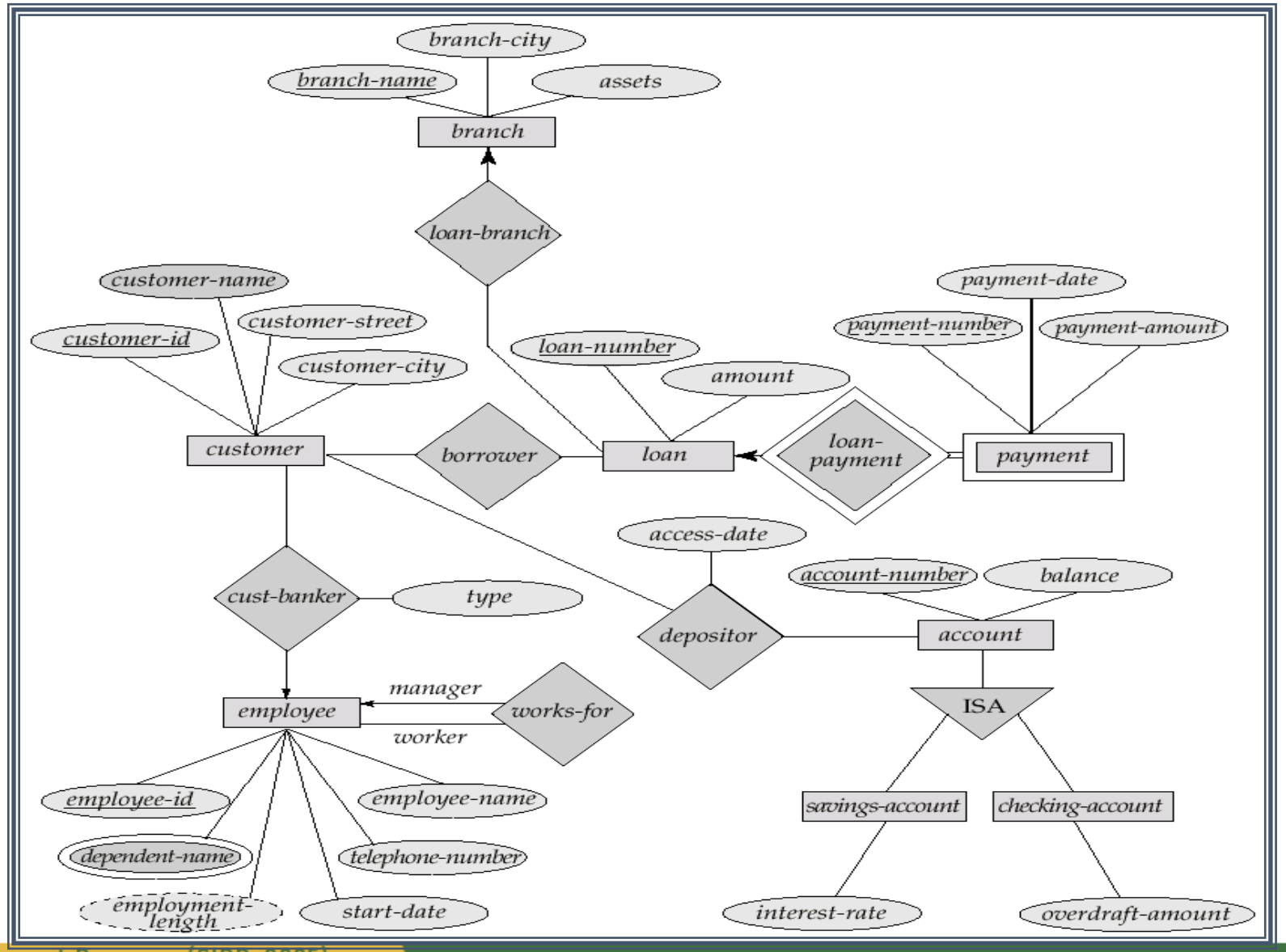
2. **Partial Participation:**
   - The entity in the entity set may or may NOT participate in the relationship.
   - If some courses are not enrolled by any of the students, the participation in the course will be partial.

Participation Constraint

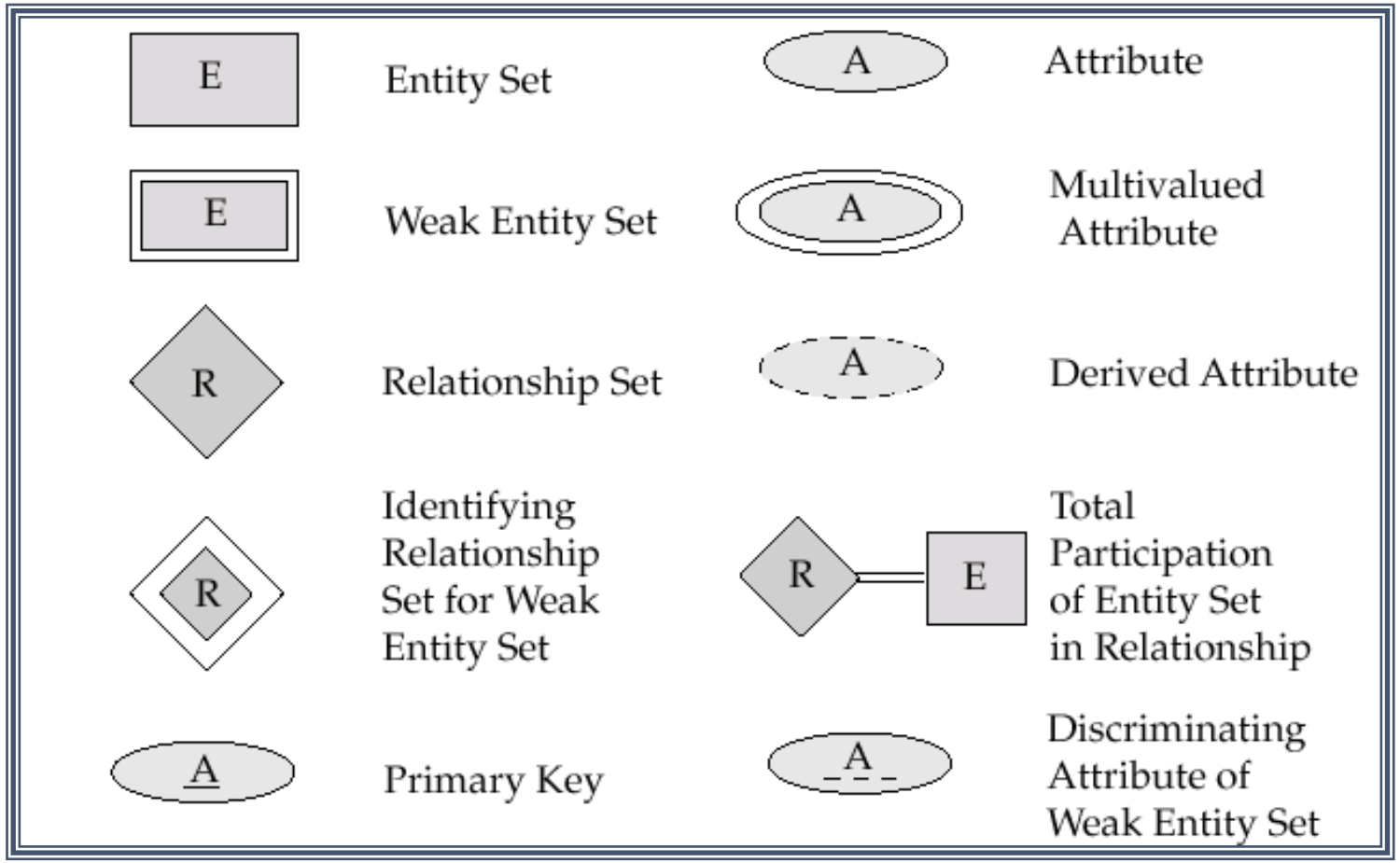


Total Participation and Partial Participation

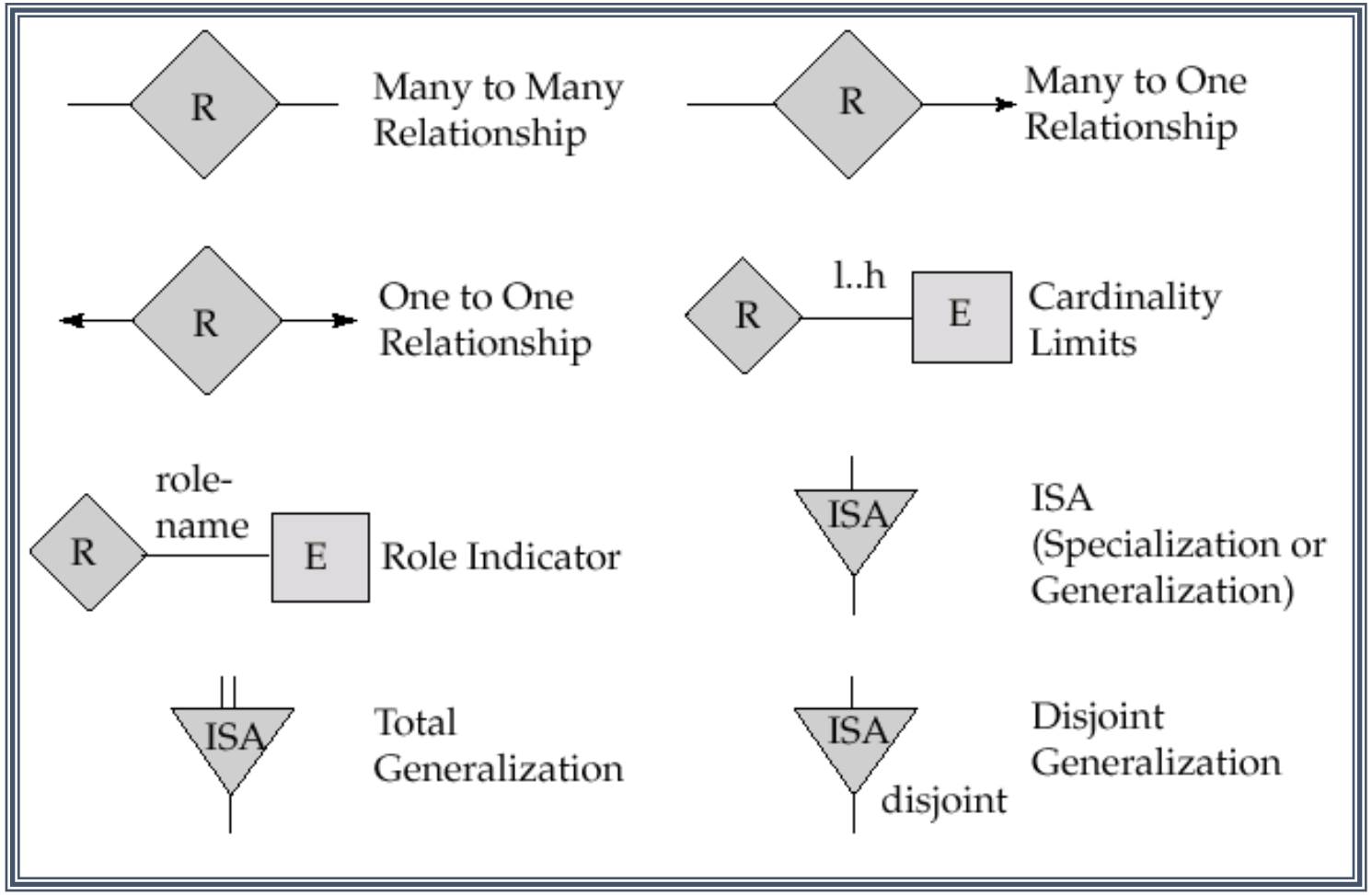
E-R Diagram for a Banking Enterprise



Summary of Symbols Used in E-R Notation



Summary of Symbols Used in E-R Notation



Relational Model

- Logical level model.
- The relational model represents how data is stored and managed in **Relational Databases**.
- Data is organized into **tables**, each known as a **relation**, consisting of **rows** (tuples) and **columns** (attributes).
- Each relation = table; each tuple = row.

Relational Model

- A relational model represents how we can store data in Relational Databases. Here, a relational database stores information in the form of relations or tables.

ER vs Relational Model

Feature	ER Model	Relational Model
Level	Conceptual	Logical
Representation	Diagram	Tables
Purpose	Design	Implementation
Use	Modeling	Querying, Storage

Key Terms in the Relational Model

- **Attribute**: Attributes are the properties that define an entity.
  - Example: ROLL_NO, NAME, ADDRESS etc.

- **Relation Schema**: A relation schema defines the structure of the relation and represents the name of the relation with its attributes.
  - Example: STUDENT (ROLL_NO, NAME, ADDRESS, PHONE, and AGE) is the relation schema for STUDENT.
  - If a schema has more than 1 relation it is called Relational Schema.

- **Tuple**: A Tuple represents a row in a relation. Each tuple contains a set of attribute values that describe a particular entity.
  - Example: (1, RAM, DELHI, 9455123451, 18) is a tuple in the STUDENT table.

Key Terms in the Relational Model

- **Relation Instance**: The set of tuples of a relation at a particular instance of time is called a relation instance.
  - It can change whenever there is an insertion, deletion or update in the database.

- **Degree**: The number of attributes in the relation is known as the degree of the relation.
  - Example: The STUDENT relation has a degree of 5, as it has 5 attributes.

- **Cardinality**: The number of tuples in a relation is known as cardinality.
  - Example: The STUDENT relation defined above has cardinality 4.

Key Terms in the Relational Model

- **Column:** The column represents the set of values for a particular attribute.
  - **Example:** The column ROLL_NO is extracted from the relation STUDENT.

- **NULL Values:** The value which is not known or unavailable is called a NULL value. It is represented by NULL.
  - **Example:** PHONE of STUDENT having ROLL_NO 4 is NULL.

Types of Keys in the Relational Model

1. Primary Key:

- A Primary Key uniquely identifies each tuple in a relation. It must contain unique values and cannot have NULL values. Example: ROLL_NO in the STUDENT table is the primary key.

2. Candidate Key

- A Candidate Key is a set of attributes that can uniquely identify a tuple in a relation. There can be multiple candidate keys, and one of them is chosen as the primary key.

Types of Keys in the Relational Model

3. Super Key

- A Super Key is a set of attributes that can uniquely identify a tuple. It may contain extra attributes that are not necessary for uniqueness.

4. Foreign Key

- A Foreign Key is an attribute in one relation that refers to the primary key of another relation. It establishes relationships between tables. Example: BRANCH_CODE in the STUDENT table is a foreign key that refers to the primary key BRANCH_CODE in the BRANCH table.

Types of Keys in the Relational Model

5. Composite Key

- A Composite Key is formed by combining two or more attributes to uniquely identify a tuple. Example: A combination of FIRST_NAME and LAST_NAME could be a composite key if no one in the database shares the same full name.

Constraints in Relational Model

1. Domain Constraints

Domain Constraints ensure that the value of each attribute A in a tuple must be an atomic value derived from its specified domain, dom(A).

2. **Key Integrity**

Every relation in the database should have at least one set of attributes that defines a tuple uniquely. Those set of attributes is called keys.

e.g.; ROLL_NO in STUDENT is key. No two students can have the same roll number.

Constraints in Relational Model

So a key has two properties:

- It should be unique for all tuples.
- It can’t have NULL values.

3. Referential Integrity Constraints

When one attribute of a relation can only take values from another attribute of the same relation or any other relation, it is called referential integrity.

ER to Relational Mapping

ER Diagram Description

Entities and Relationships:

Student (StudentID, Name, Email)
Course (CourseID, Title, Credits)
Instructor (InstructorID, Name, Department)
Enrolled (Student ↔ Course)
Teaches (Instructor ↔ Course)

ER to Relational Mapping

Mapping Steps

1. Entity Sets → Relations

Student → Student(StudentID PRIMARY KEY, Name, Email)
Course → Course(CourseID PRIMARY KEY, Title, Credits)
Instructor → Instructor(InstructorID PRIMARY KEY, Name, Department)

2. Relationship Sets → Relations

a) Enrolled (Many-to-Many)

Enrolled(StudentID, CourseID, Semester, Grade)

ER to Relational Mapping

Primary Key: (StudentID, CourseID, Semester)

Foreign Keys:
StudentID → Student(StudentID)
CourseID → Course(CourseID)

b) Teaches (Many-to-Many)
Teaches(InstructorID, CourseID, Semester)

Primary Key: (InstructorID, CourseID, Semester)

Foreign Keys:
InstructorID → Instructor(InstructorID)
CourseID → Course(CourseID)

Thank You