Database Management Systems
SYLLABUS

- **UNIT-I:**

  Introduction To Database And System Architecture: Database Systems and their Applications, Database Vs File System, View of Data, Data Models, Database Languages- DDL and DML, Transaction Management, Database users and Administrators, Database System Structure.

  Introduction to Database Design: ER Diagrams, Entities, Attributes and Entity sets, Relationships and Relationship set, Extended ER Features, Conceptual Design with the ER Model, Logical database Design.

- **UNIT-II**

  SQL: Queries and Constraints: Form of Basic SQL Query, SQL Operators, Set Operators, Nested Queries, Aggregate Operators, NULL values, Integrity Constraints Over Relations, Joins, Introduction to View, Destroying / Altering Tables and Views, Cursors, Triggers and Active Databases.

- **UNIT-III**

  RELATIONAL MODEL: Introduction to Relational Model, Basic Structure, Database Schema, Keys, Relational Algebra and Relational Calculus.

  Storage and Indexing: File Organizations and Indexing-Overview of Indexes, Types of Indexes, Index Data Structures, Tree structured Indexing, Hash based Indexing.

- **UNIT-IV**

SCHEMA REFINEMENT AND NORMAL FORMS : Introduction to Schema Refinement, Functional Dependencies, Reasoning about FD, Normal Forms, Properties of Decomposition.

- **UNIT-V**

TRANSACTION MANAGEMENT TRANSACTIONS: Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability.

Concurrency Control: Lock based Protocols, Timestamp based protocols

Recovery System: Recovery and Atomicity, Log based recovery, Shadow Paging, Recovery with concurrent Transactions, Buffer Management.

- **TEXT/REFERENCE BOOKS**

1. "Data base Management Systems", Raghurama Krishnan, Johannes Gehrke,TATA McGraw Hill 3rd Edition
2. "Data base System Concepts", Silberschatz, Korth, McGraw hill, Vedition.
3. "Introduction to Database Systems", C.J.Date PearsonEducation.
4. "Database Systems design, Implementation, and Management", Rob & Coronel5th Edition.
5. "Database Management Systems", P. Radha Krishna HI-TECH Publications2005.
6. "Database Management System", ElmasriNavate PearsonEducation.
7. "Database Management System", Mathew Leon,Leo.

Course Objectives

1. To understand the different issues involved in the design and implementation of a database system.
2. To understand Structured Query Language for manipulating the Data.
3. To study the physical, conceptual and logical database designs
4. To provide concepts of Transaction, Concurrency and Recovery Management Strategies of a DBMS
5. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Course Outcomes:

1. Identify the role of Database System Applications and the design issues related.
2. Design the logical model for the applications and apply indexing techniques.
3. Construct a Database Schema, manipulate data using a SQL.
4. Can apply the Schema Refinement techniques for a database design for optimized access.
5. For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.

UNIT-I

DBMS

INTRODUCTION TO DATABASE DESIGN

A.PAVITHRA
ASSISTANT PROFESSOR
DEPT OF IT
GRIET, HYDERABAD

Introduction to Database Design

- E-R Model
- Entities
- Attributes
- E-R Diagrams
- Entity Sets
- Relationships and Relationship set
- Extended ER Features
  - Specialization
  - Generalization
  - Aggregation
- Conceptual Design with the ER Model
- Logical database Design

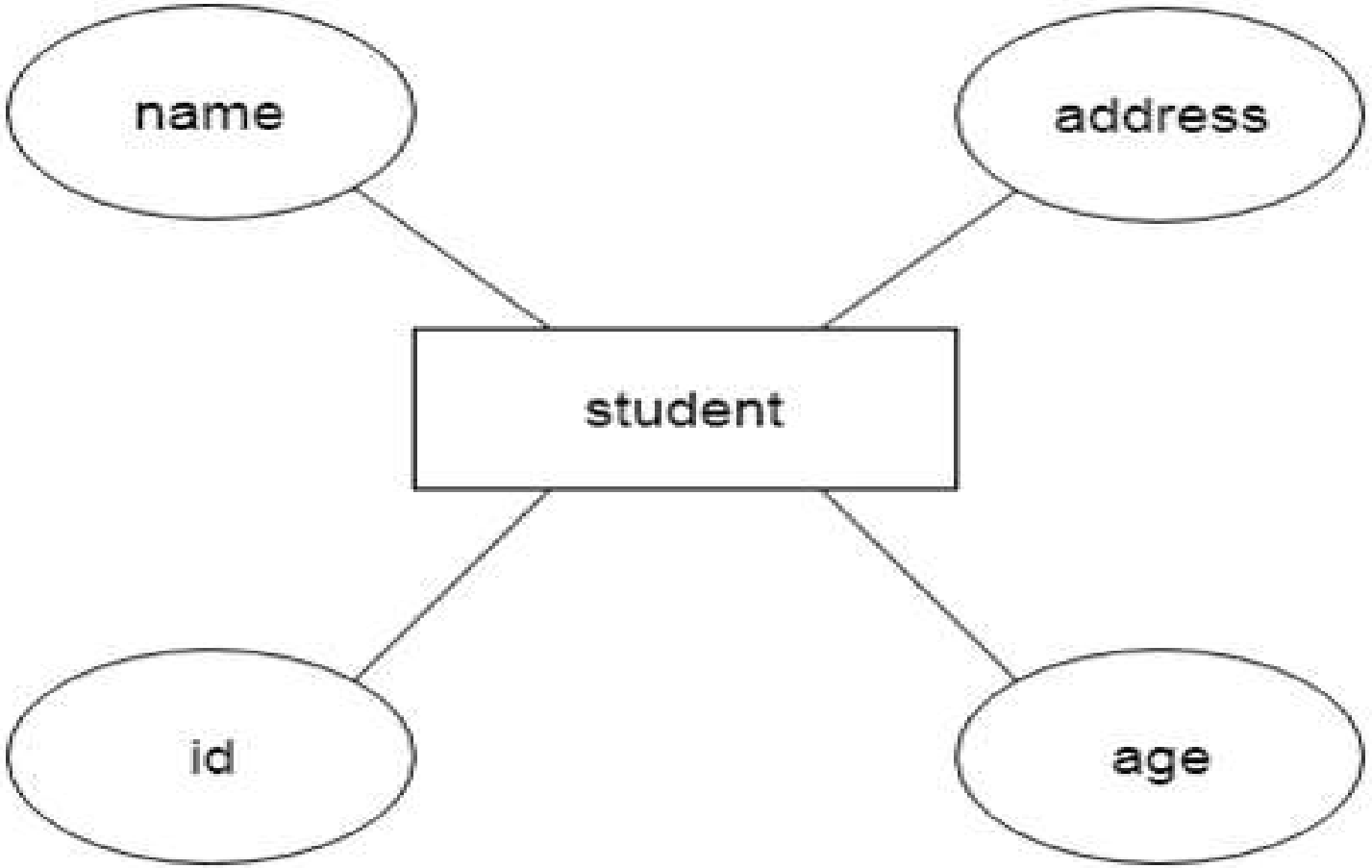
E-R Model

- ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.

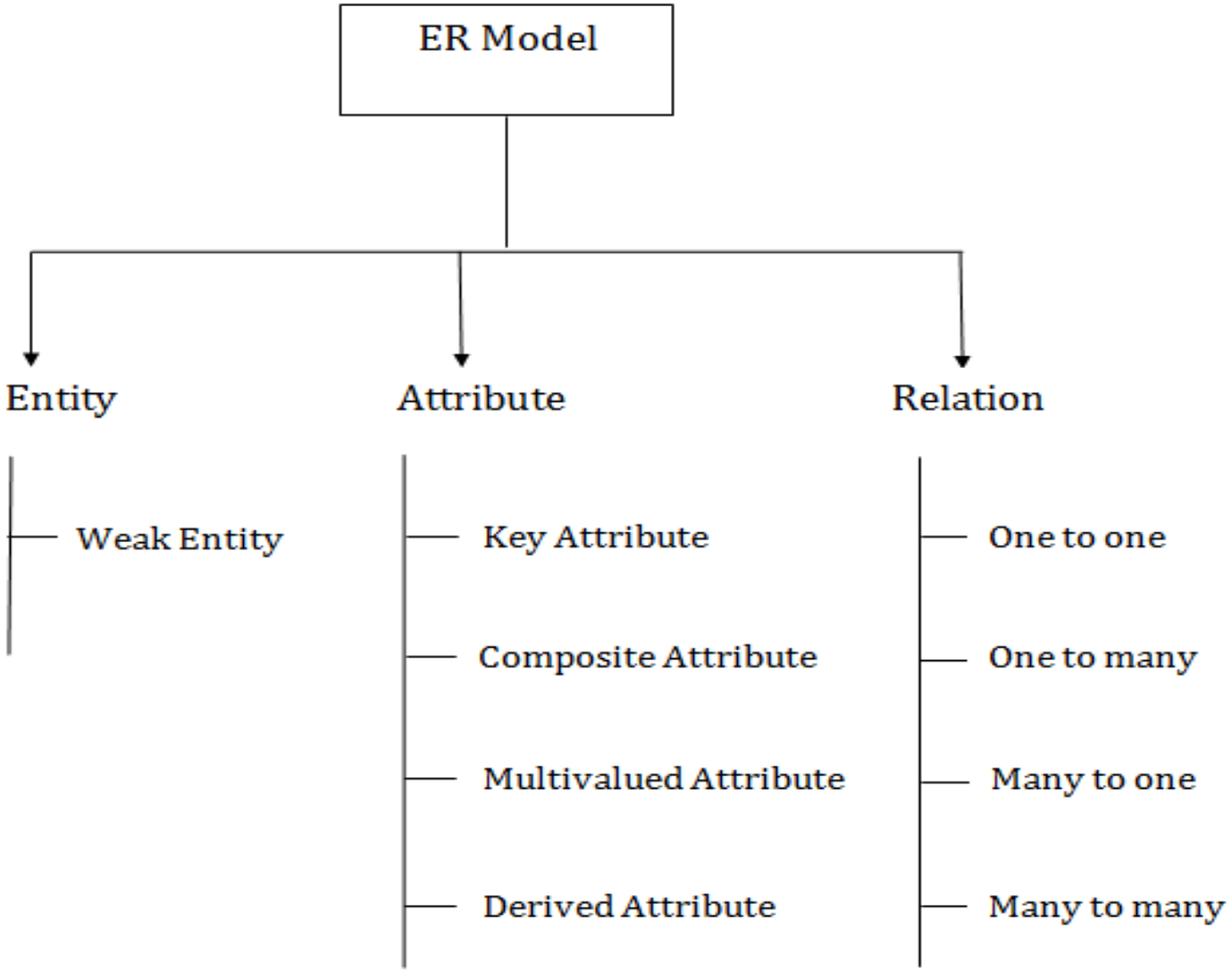
- It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.

- In ER modelling, the database structure is portrayed as a diagram called an entity-relationship diagram.
- Ex: School Student database

Example:



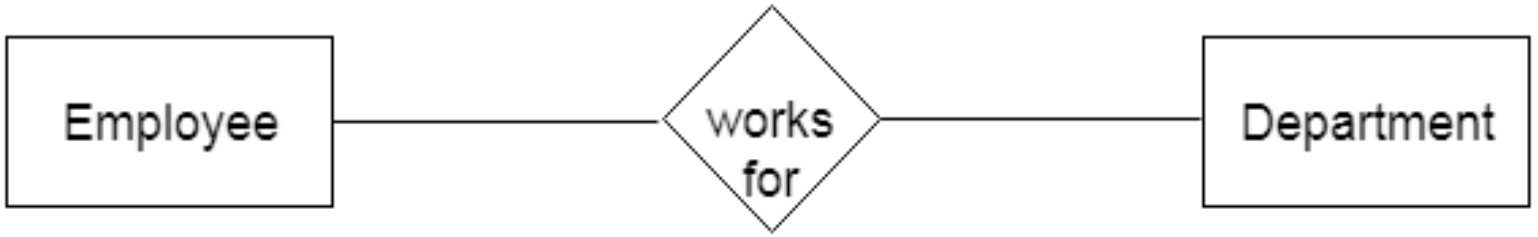
Component of ER Diagram



Entity

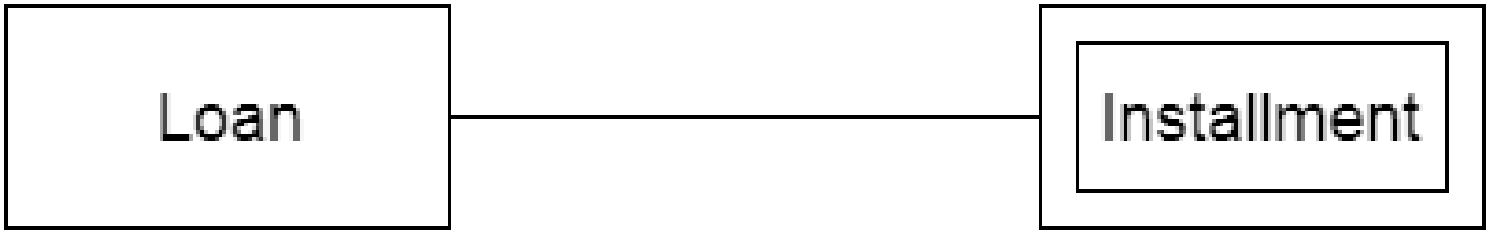
- An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

- Consider an organization as an example- manager, product, employee, department etc. can be taken as an entity.



a. Weak Entity

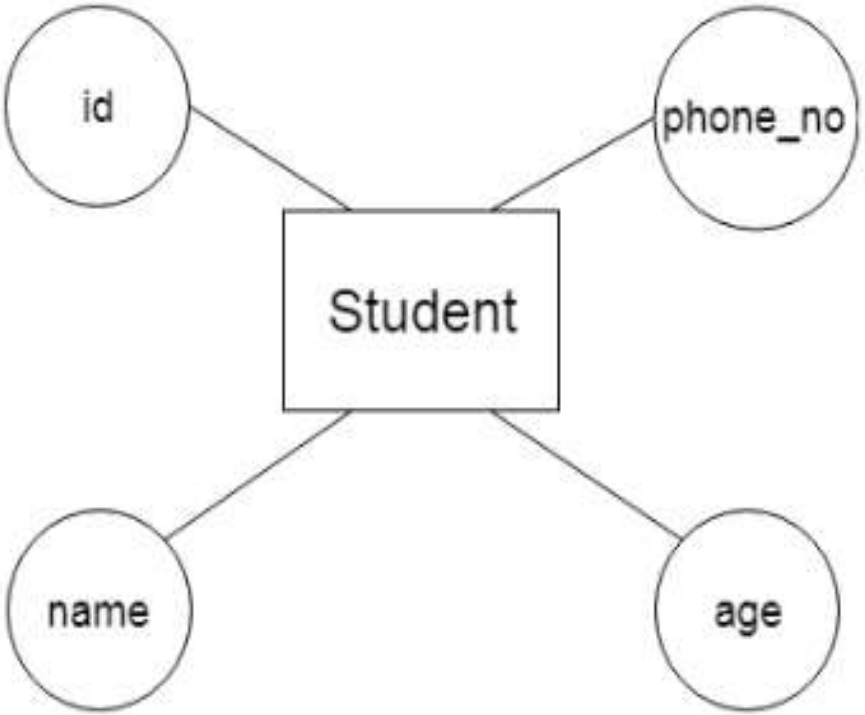
- An entity that depends on another entity called a weak entity.
- The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.



ATTRIBUTE

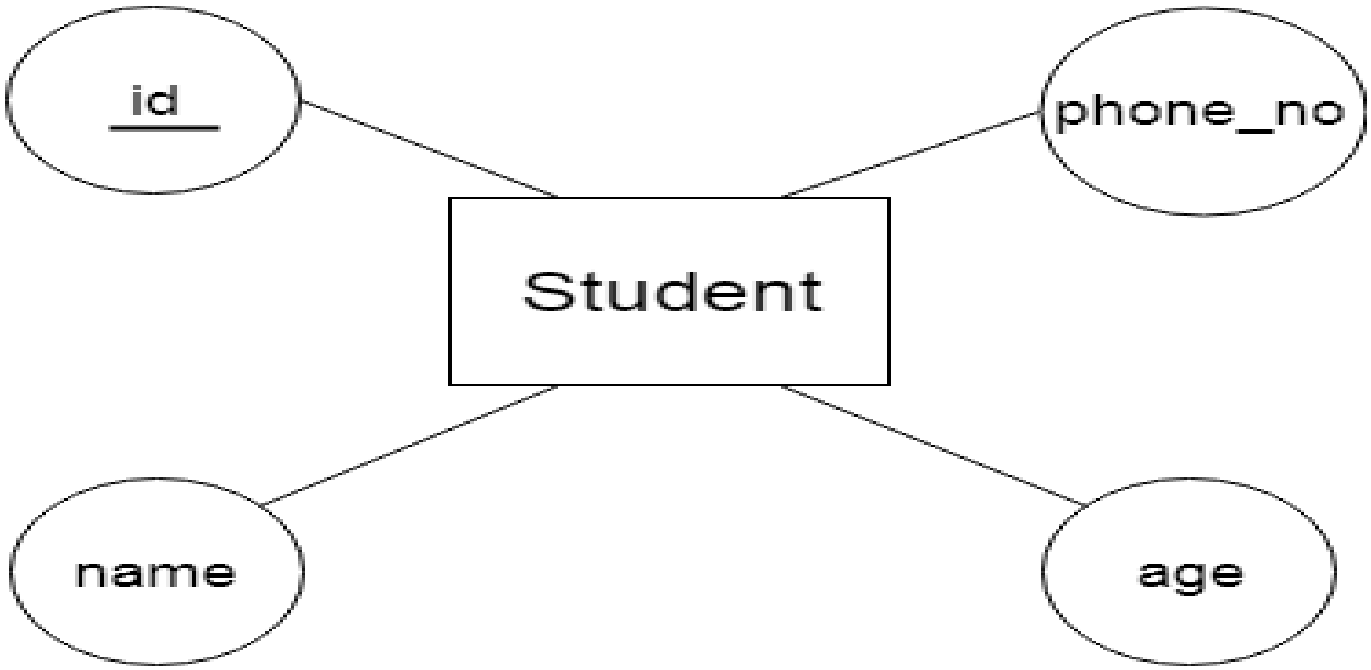
The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.

For example, id, age, contact number, name, etc. can be attributes of a student.



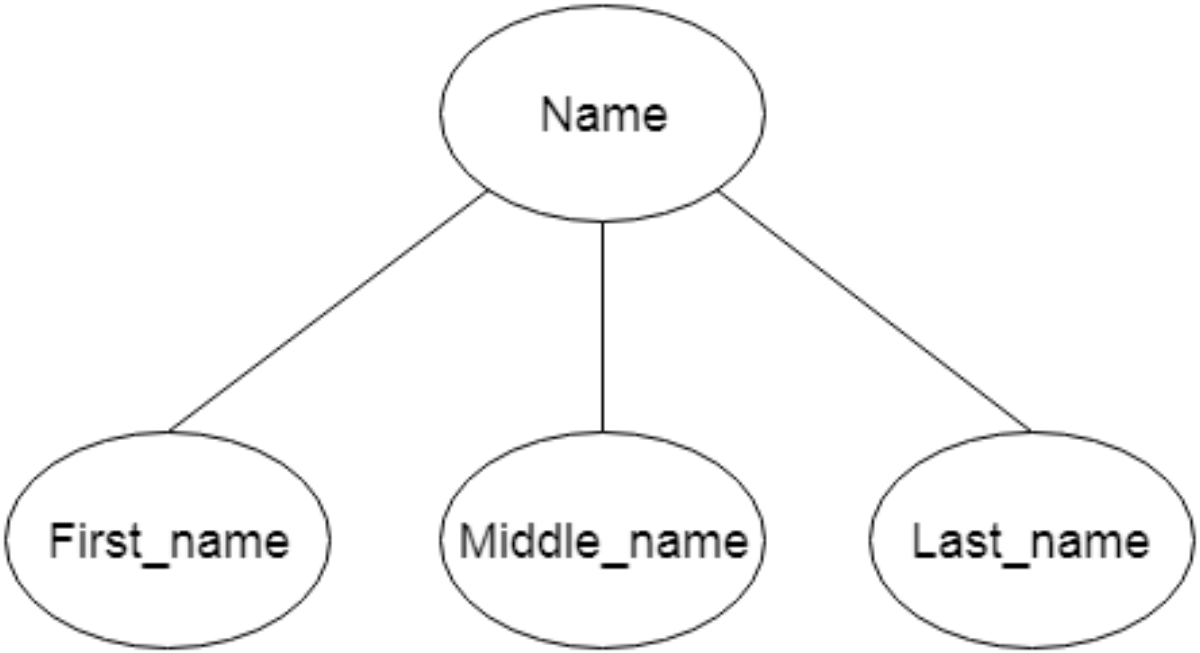
a. Key Attribute

- The key attribute is used to represent the main characteristics of an entity. It represents a primary key.
- The key attribute is represented by an ellipse with the text underlined.



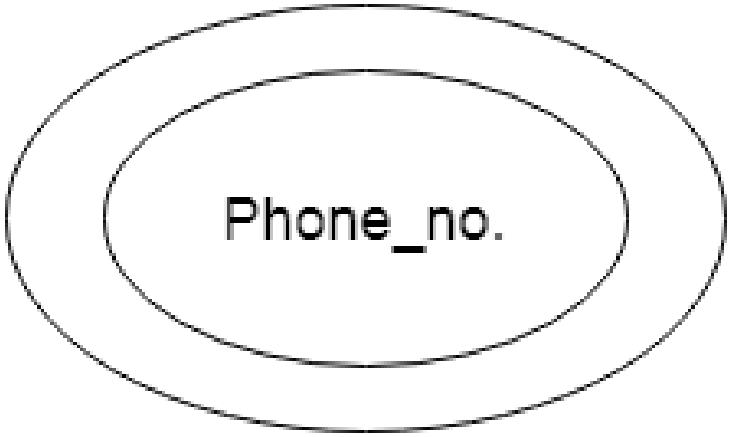
b. Composite Attribute

- An attribute that composed of many other attributes is known as a composite attribute.
- The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.



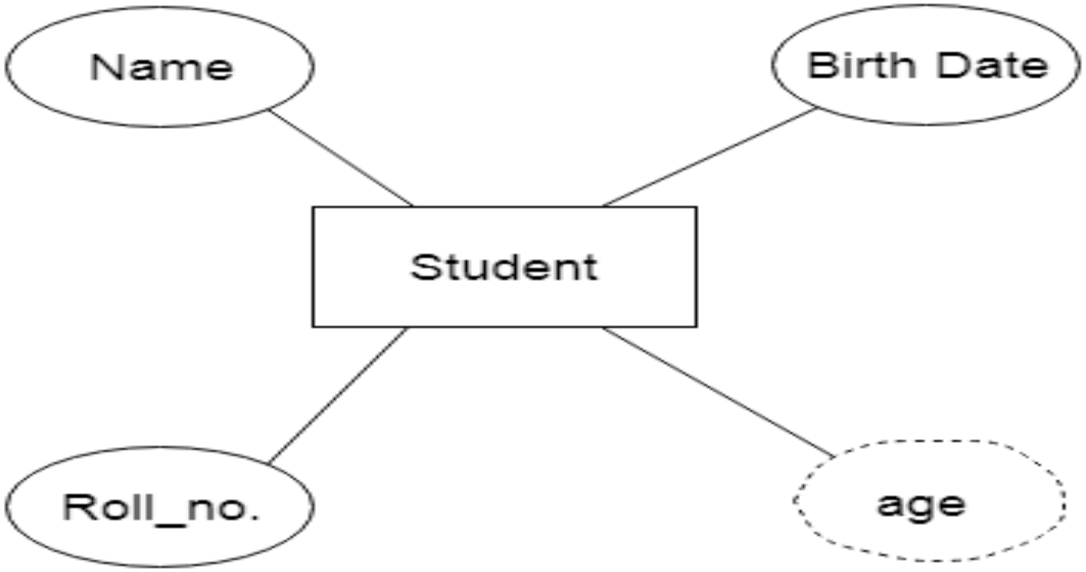
c. Multi valued Attribute

- An attribute can have more than one value. These attributes are known as a multi valued attribute. The double oval is used to represent multi valued attribute.
- **For example,** a student can have more than one phone number.



d. Derived Attribute

- An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.
- **For example,** A person's age changes over time and can be derived from another attribute like Date of birth.



3. Relationship

- A relationship is used to describe the relation between entities.
- Diamond or rhombus is used to represent the relationship.

1) ONE-TO-ONE RELATIONSHIP:

When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.

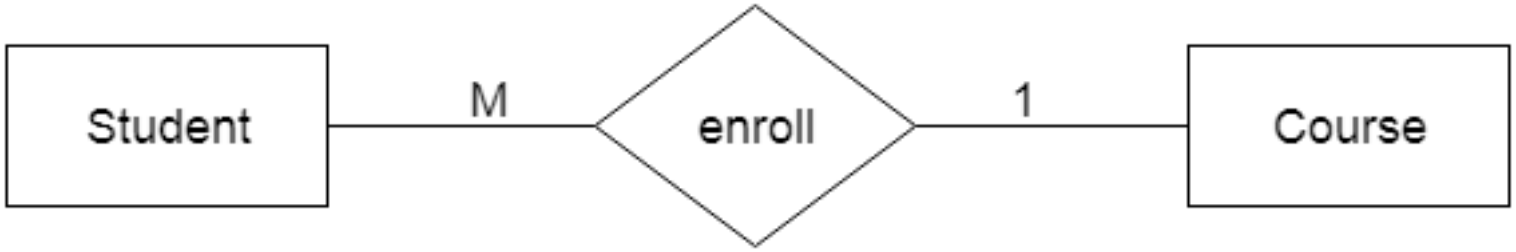
Ex: A female can marry to one male, and a male can marry to one female.

2) One-to-many relationship

- When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a **one-to-many relationship**.
- **For example,** Scientist can invent many inventions, but the invention is done by the only specific scientist.

3) Many-to-one relationship

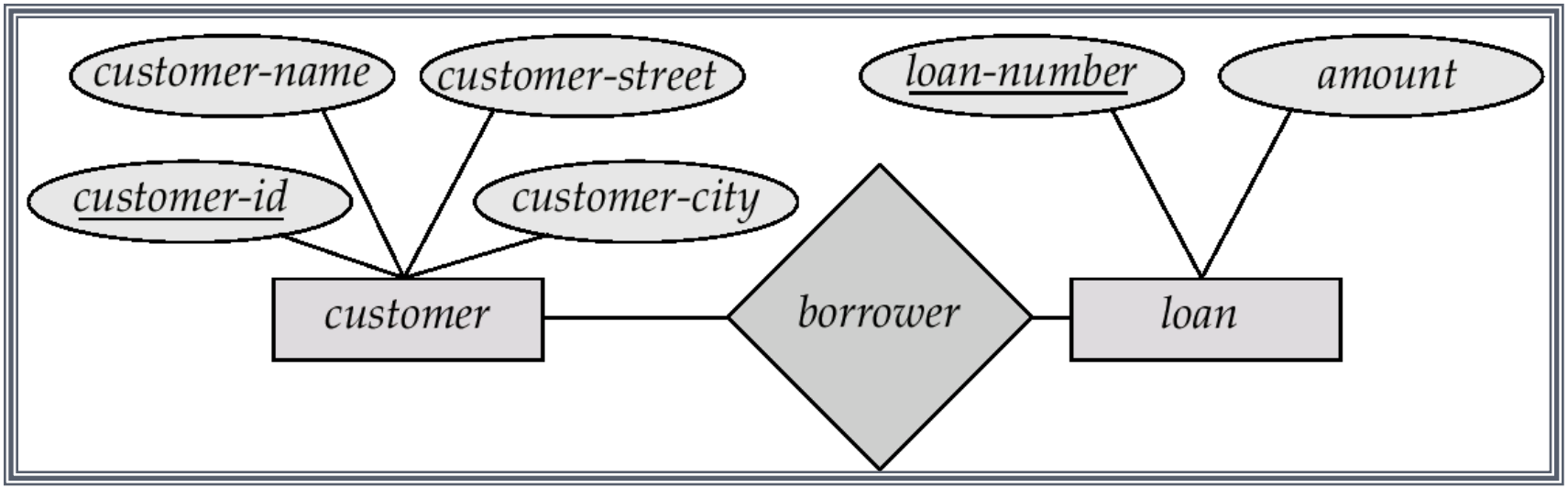
- When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a **many-to-one relationship**.
- **For example,** Student enrolls for only one course, but a course can have many students.



4) Many-to-many relationship

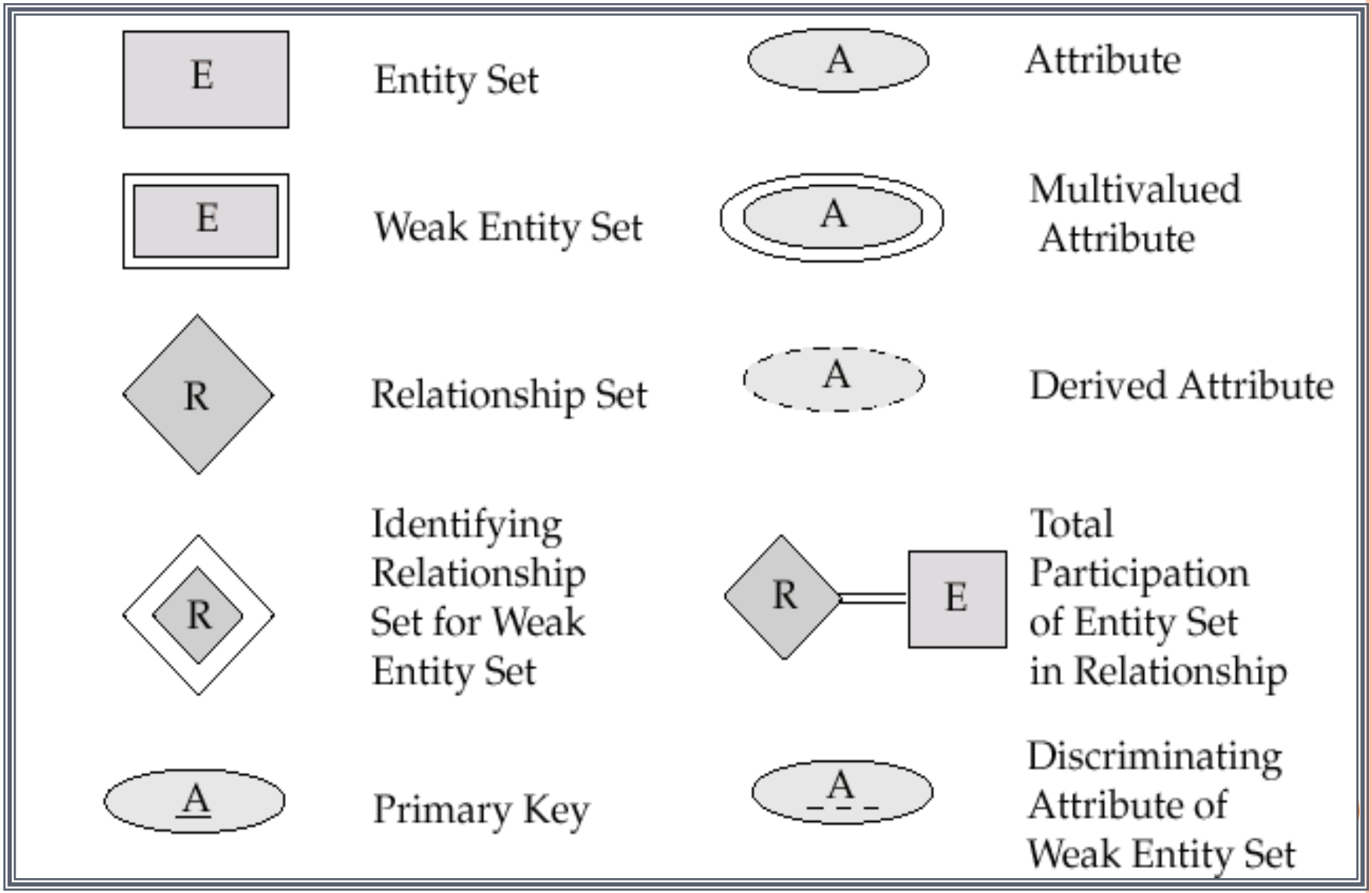
- When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship.
- **For example,** Employee can assign by many projects and project can have many employees.

E-R Diagrams

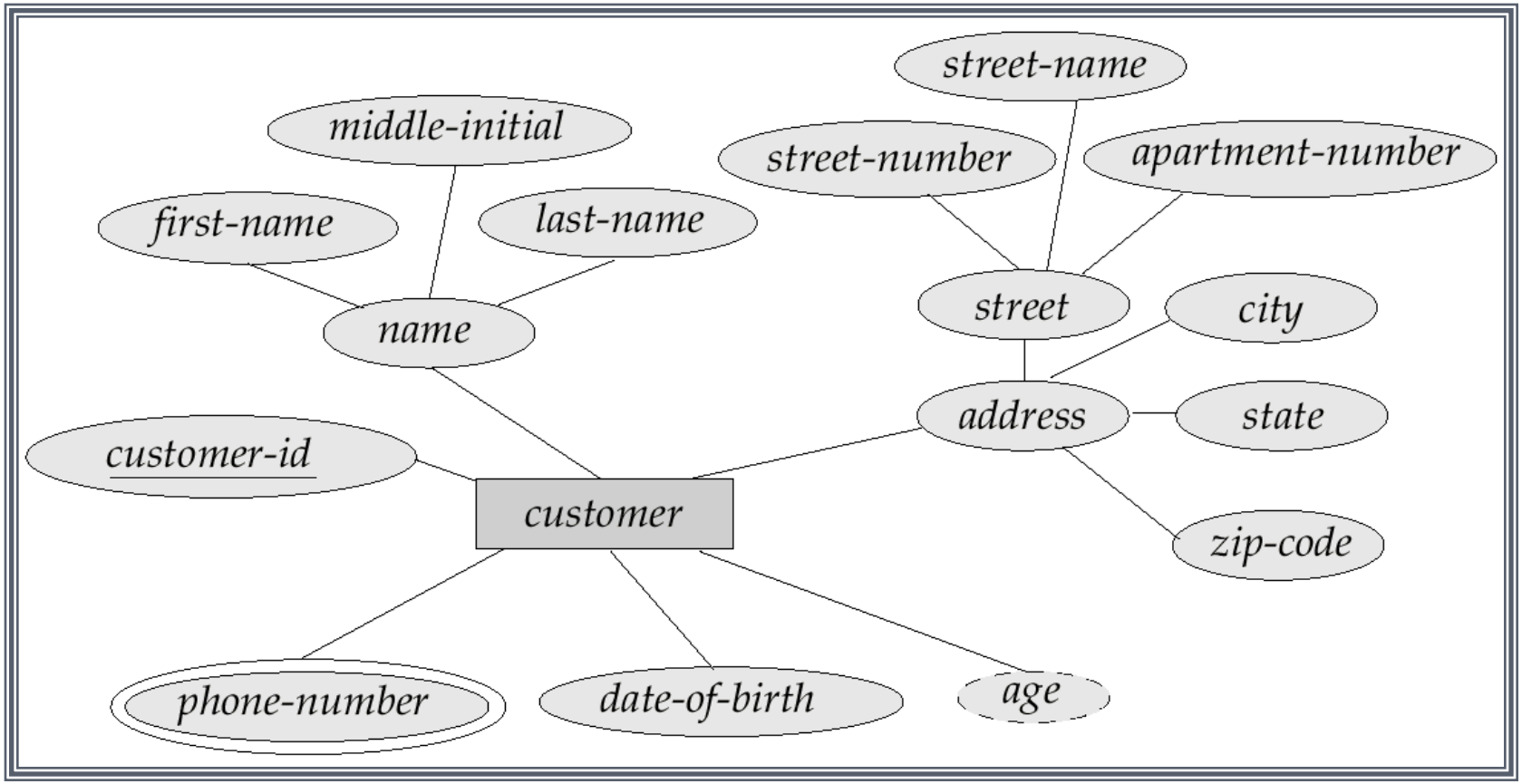


- **Rectangles** represent entity sets.
- **Diamonds** represent relationship sets.
- **Lines** link attributes to entity sets and entity sets to relationship sets.
- **Ellipses** represent attributes
  - **Double ellipses** represent multivalued attributes.
  - **Dashed ellipses** denote derived attributes.
- **Underline** indicates primary key attributes

Symbols Used in E-R Notations



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



Entity Sets

- An *entity set* is a collection of entities of the same type that share the same properties.

Example: set of all persons, companies, trees, holidays

Entity Sets *customer* and *loan*

customer-id	customer-name	customer-street	customer-city
321-12-3123	Jones	Main	Harrison
019-28-3746	Smith	North	Rye
677-89-9011	Hayes	Main	Harrison
555-55-5555	Jackson	Dupont	Woodside
244-66-8800	Curry	North	Rye
963-96-3963	Williams	Nassau	Princeton
335-57-7991	Adams	Spring	Pittsfield

customer

loan-number	amount
L-17	1000
L-23	2000
L-15	1500
L-14	1500
L-19	500
L-11	900
L-16	1300

loan

RELATIONSHIP:

- The association among entities is called a relationship.

- **For example,** an employee **works_at** a department, a student **enrolls** in a course. Here, **Works_at** and **Enrolls** are called relationships.

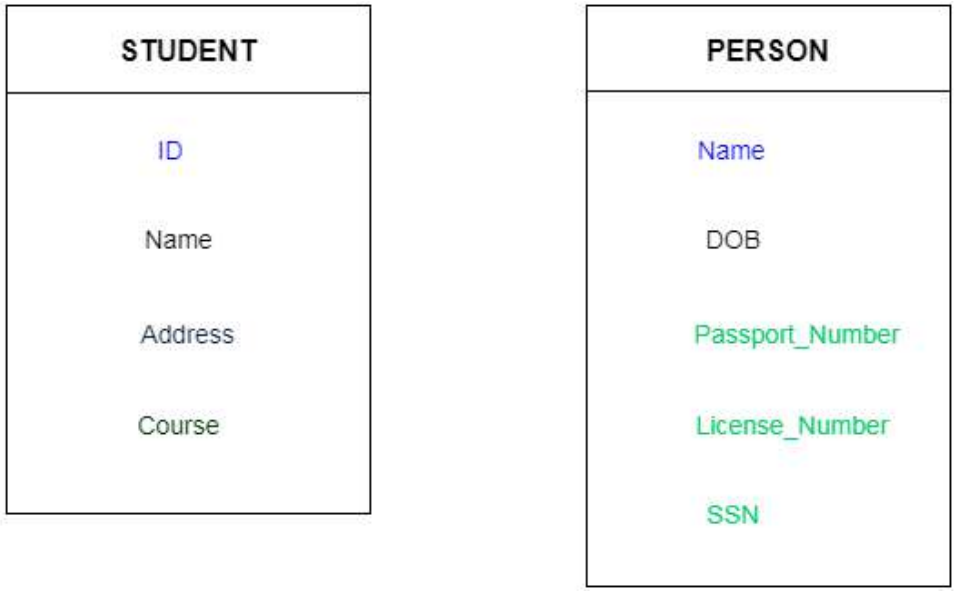
RELATIONSHIP SET

- A set of relationships of similar type is called a relationship set.

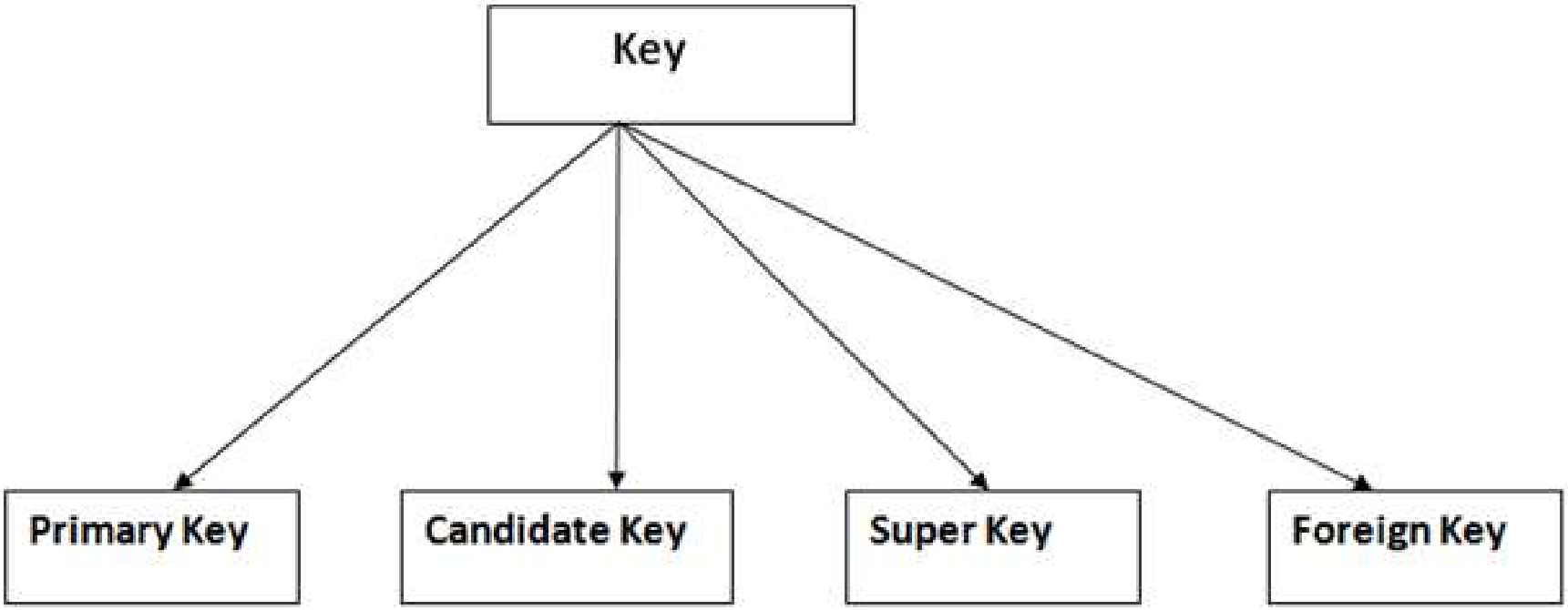
- Like entities, a relationship too can have attributes. These attributes are called **descriptive attributes**.

Keys

- Keys play an important role in the relational database.
- It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationships between tables.
  - **For example:** In Student table, ID is used as a key because it is unique for each student. In PERSON table, passport_number, license_number, SSN are keys since they are unique for each person.

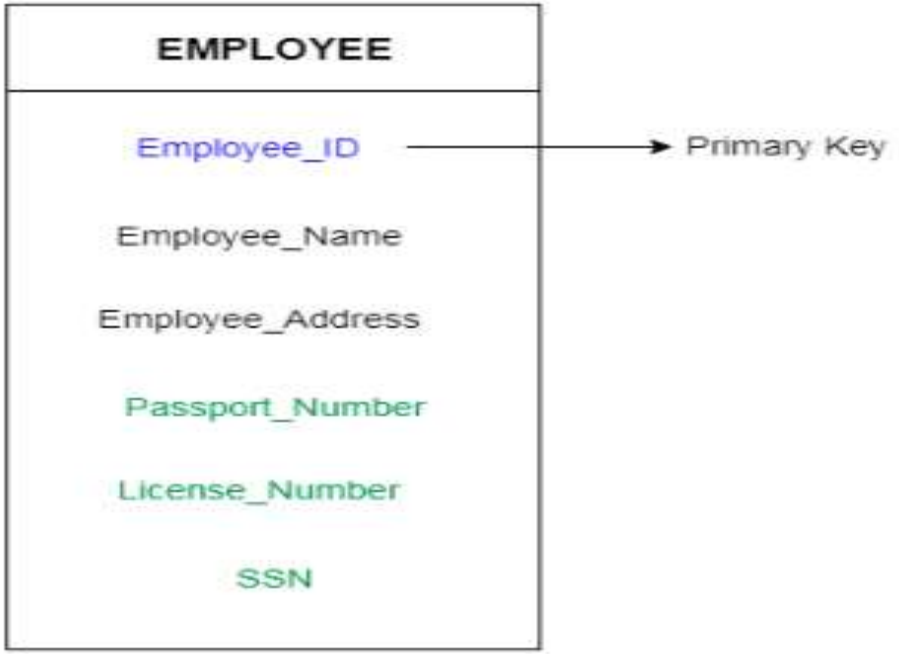


Type of keys



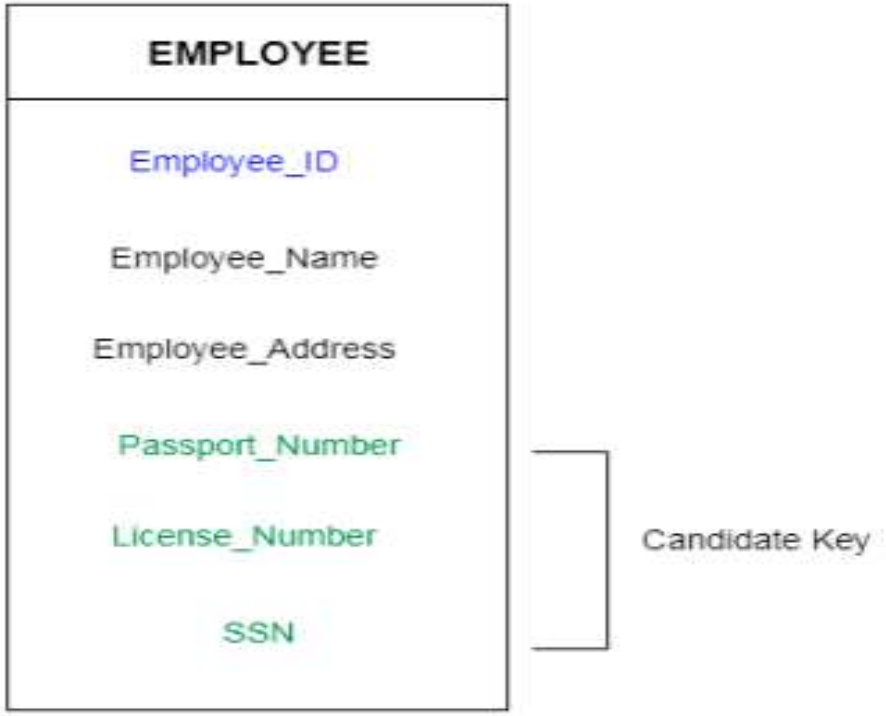
1.Primary Key

- It is the first key which is used to identify one and only one instance of an entity uniquely.
- An entity can contain multiple keys as we saw in PERSON table. The key which is most suitable from those lists become a primary key.



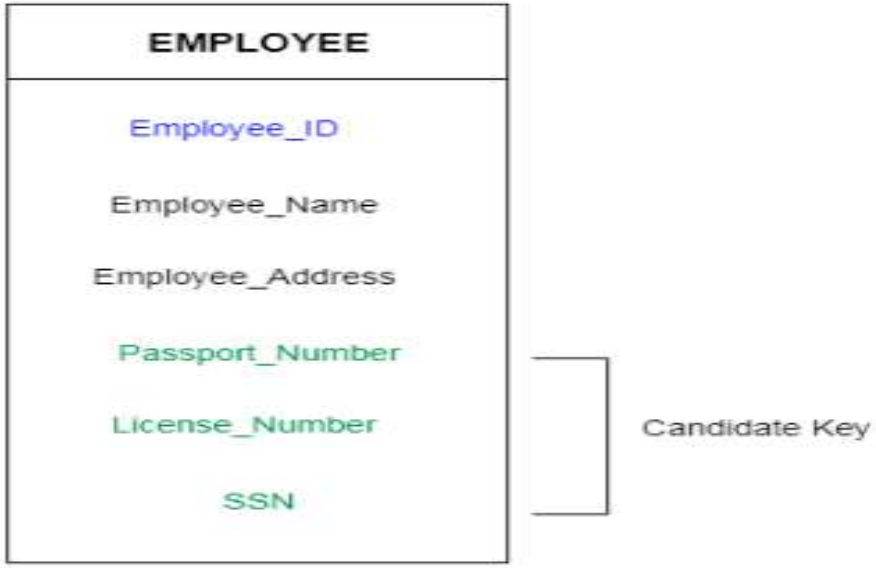
2.Candidate key

- A candidate key is an attribute or set of an attribute which can uniquely identify a tuple.
- The remaining attributes except for primary key are considered as a candidate key. The candidate keys are as strong as the primary key.



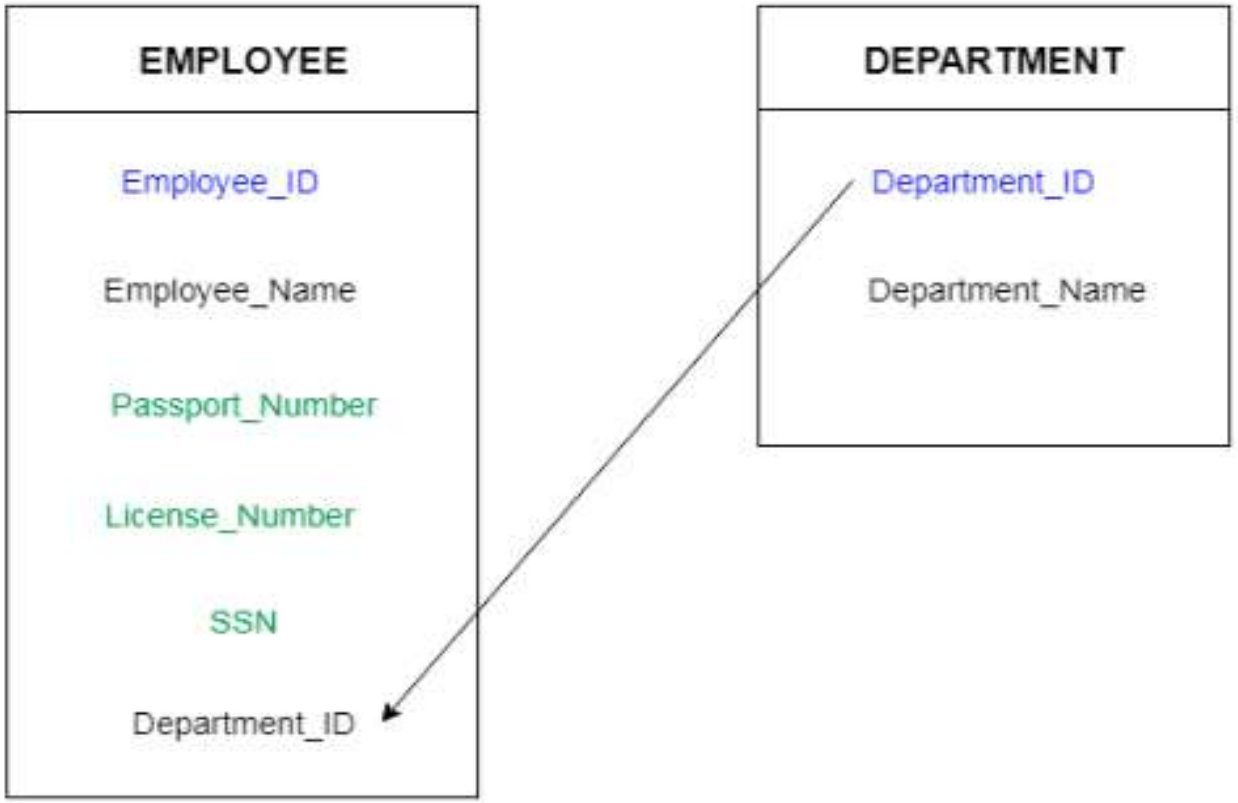
3.Super key

- Super key is a set of an attribute which can uniquely identify a tuple. Super key is a superset of a candidate key.
- **For example:** In the above EMPLOYEE table, for(EMPLOEE_ID, EMPLOYEE_NAME) the name of two employees can be the same, but their EMPLYEE_ID can't be the same. Hence, this combination can also be a key.
- The super key would be EMPLOYEE-ID, (EMPLOYEE_ID, EMPLOYEE-NAME), etc.



4.Forein key

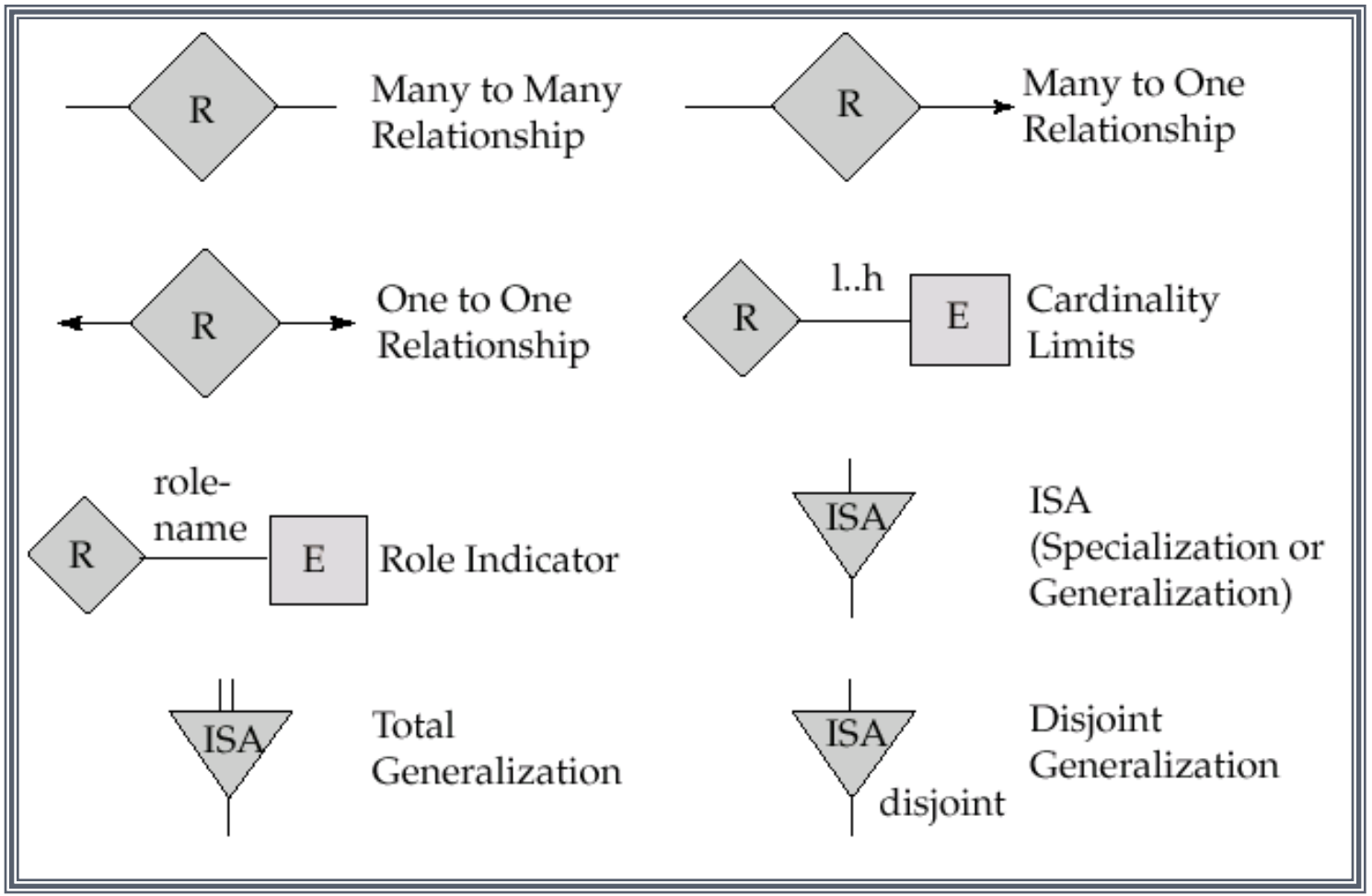
- Foreign keys are the column of the table which is used to point to the primary key of another table.



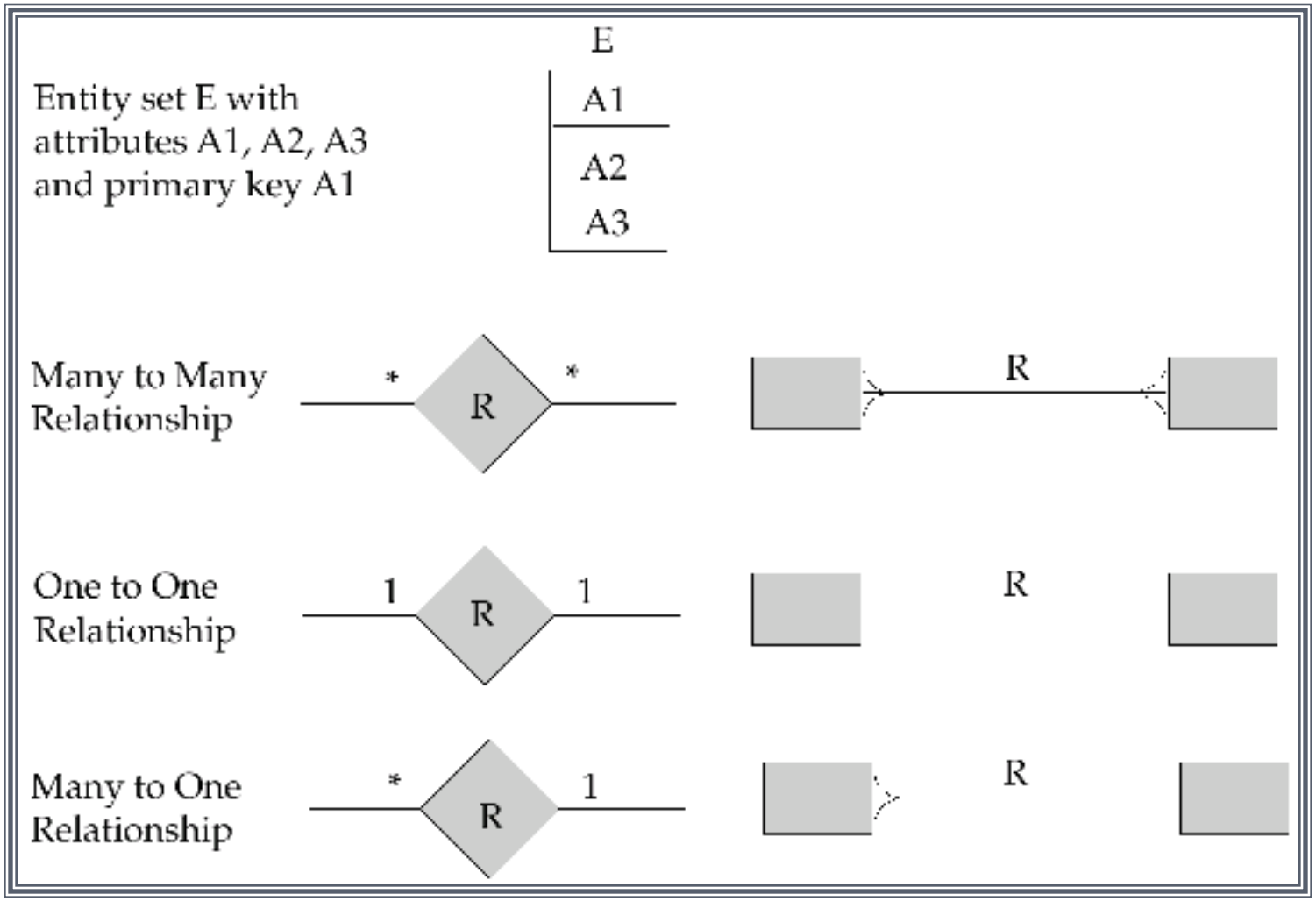
Degrees of Relationship (Cardinality)

- The **degree of relationship** (also known as cardinality) is the number of occurrences in one entity which are associated (or linked) to the number of occurrences in another.
- There are four degrees of relationship, known as:
  1. one-to-one (1:1)
  2. one-to-many (1:M)
  3. Many-to-one(M:1)
  4. many-to-many (M:N)

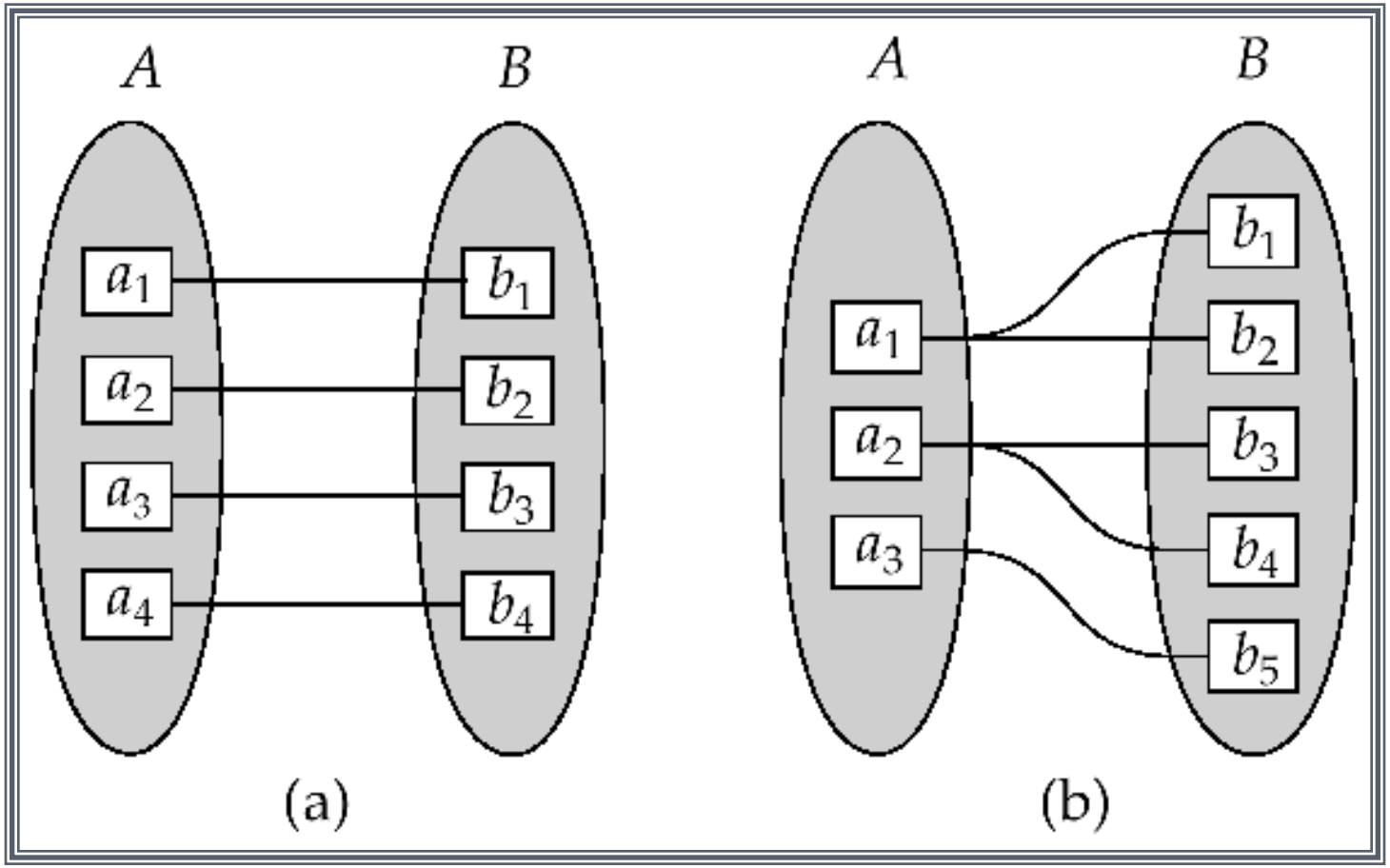
Summary of Symbols (Cont.)



Alternative E-R Notations



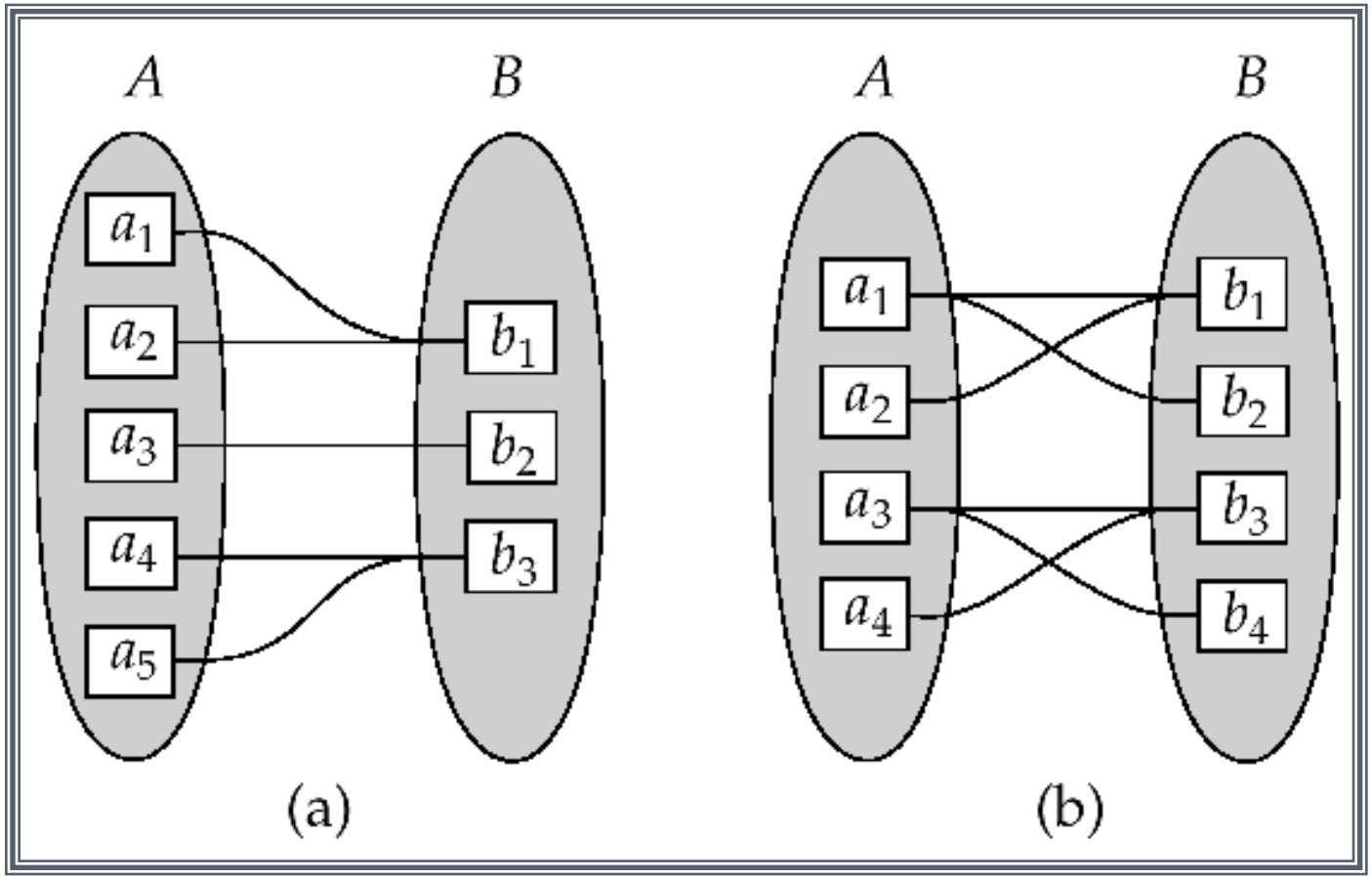
Mapping Cardinalities



One to one

One to many

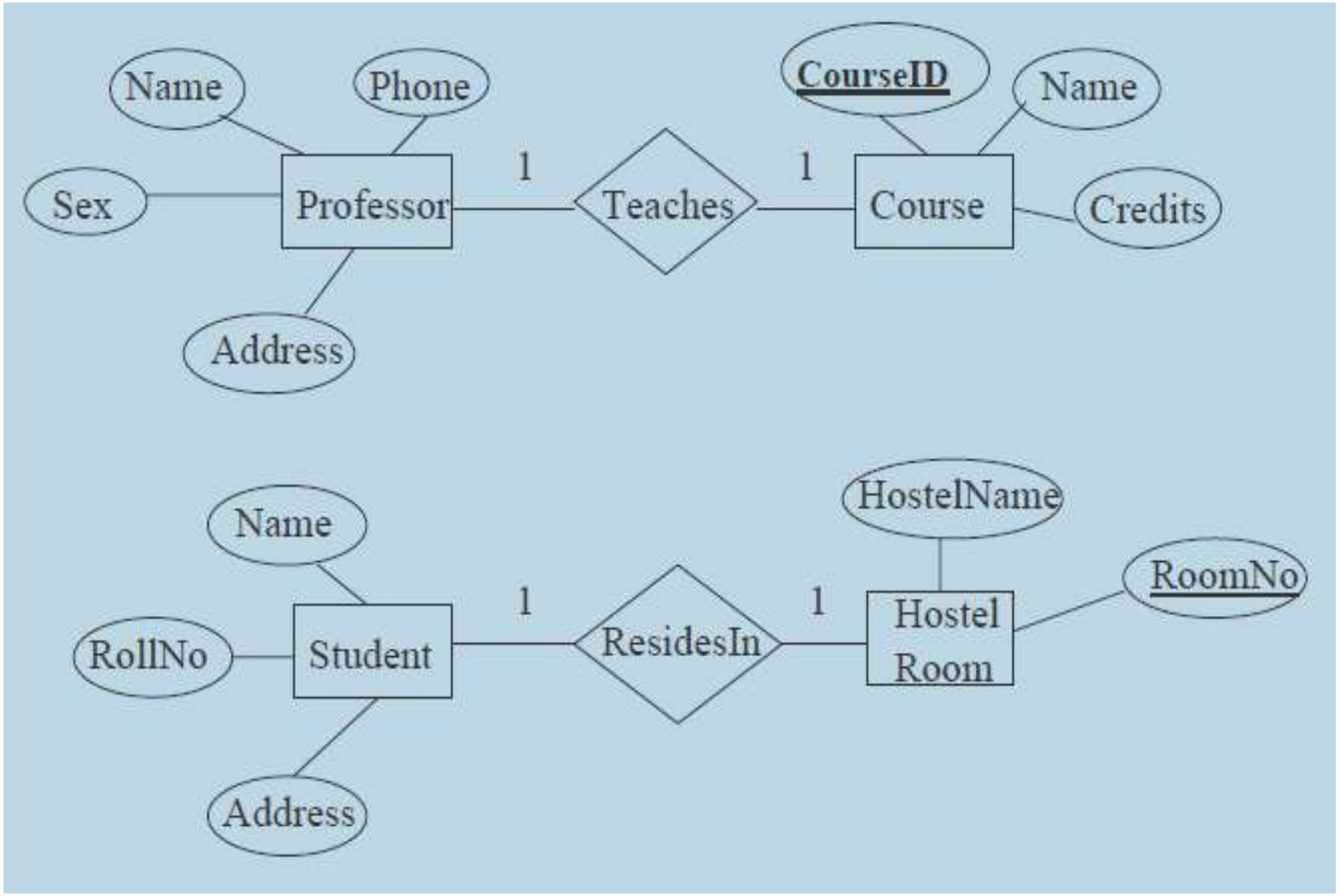
Mapping Cardinalities



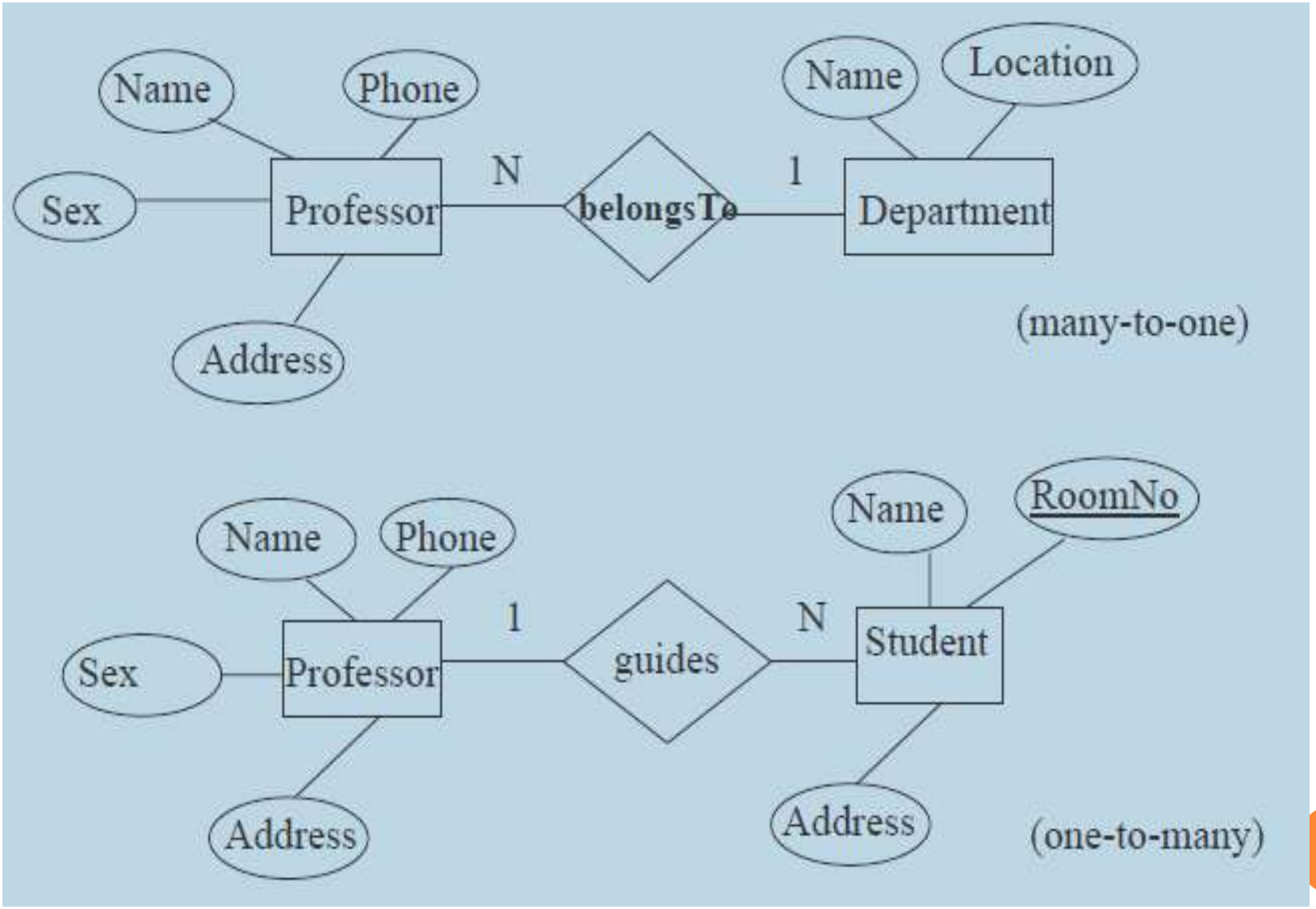
Many to one

Many to many

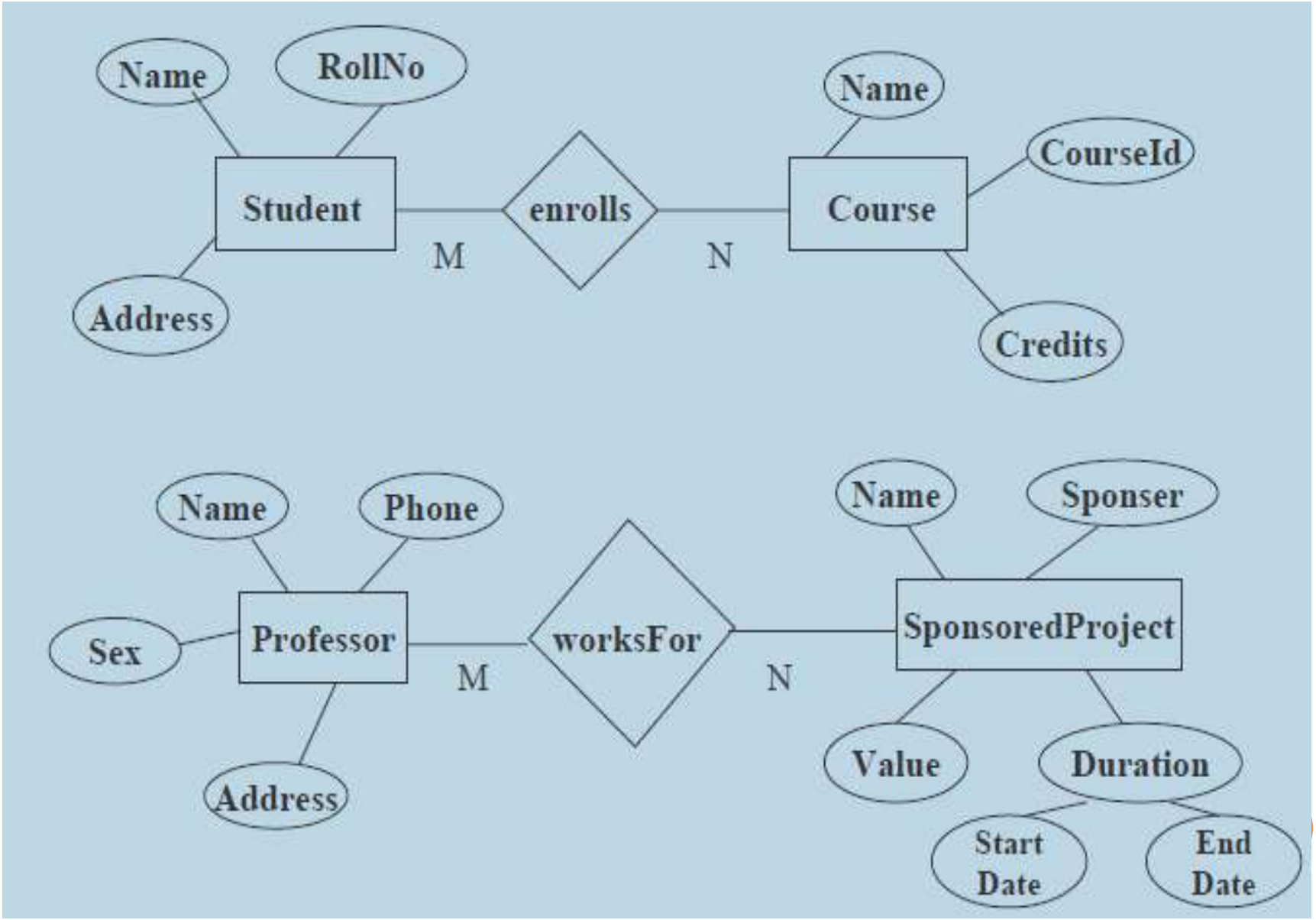
One- one mapping



One- many/ many-one mapping



Many - many mapping

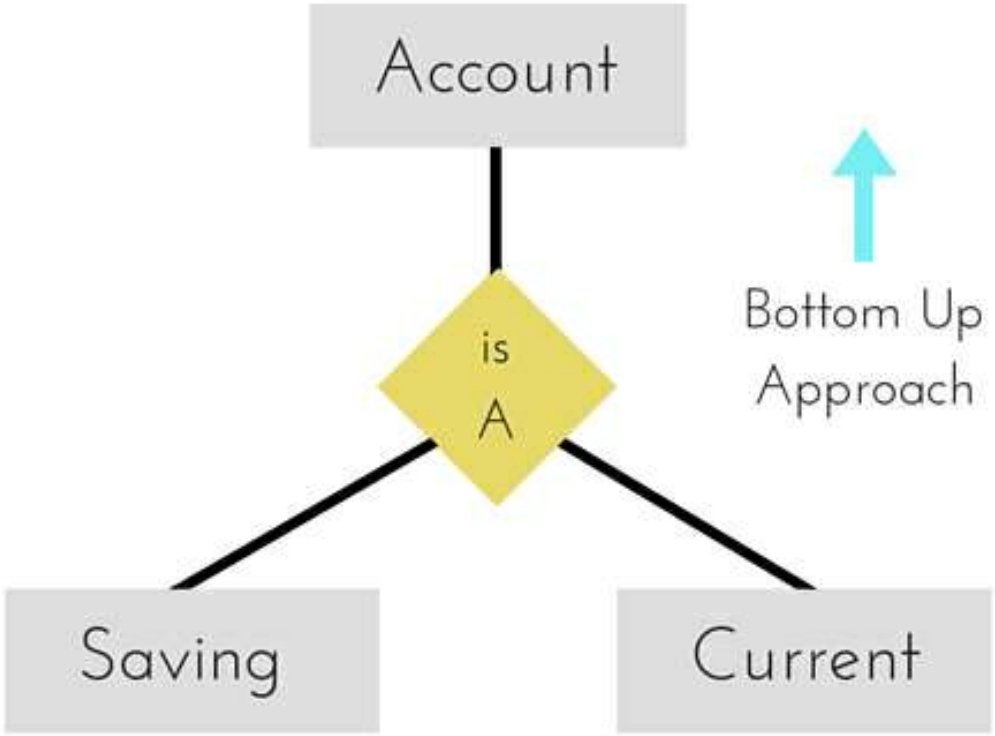


Extended E-R Features

- As part of the **Extended ER Model**, along with other improvements, three new concepts were added to the existing ER Model, they were:
  1. Generalization
  2. Specialization
  3. Aggregration

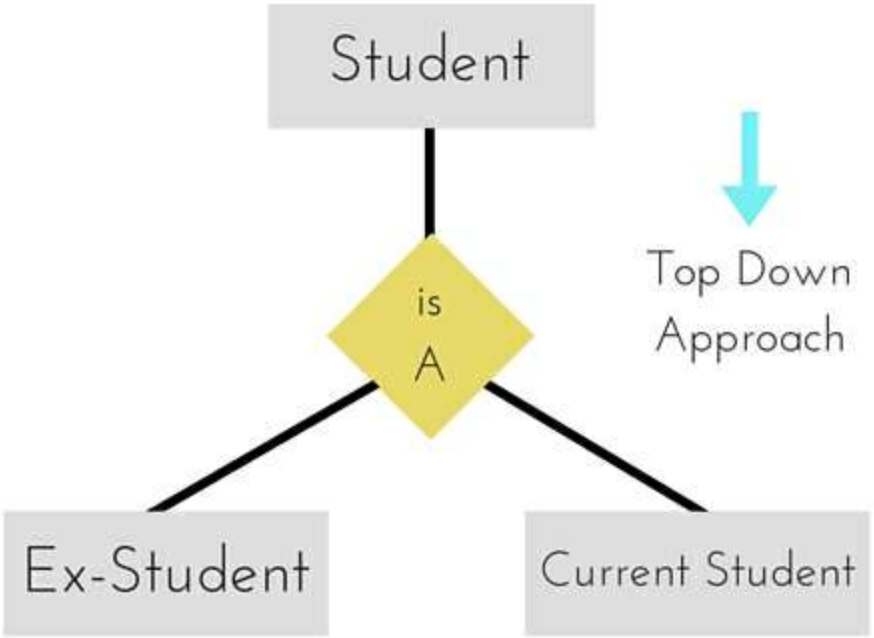
Generalization

Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity.



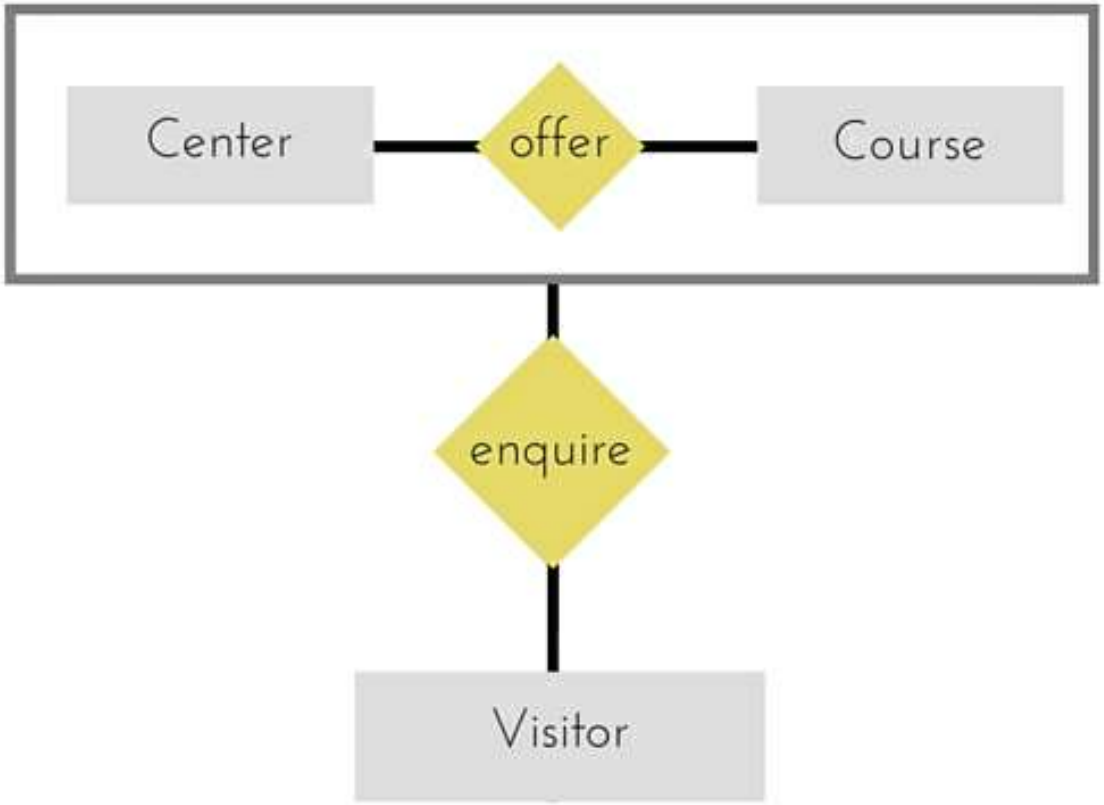
Specialization

Specialization is opposite to Generalization. It is a top-down approach in which one higher level entity can be broken down into two lower ...



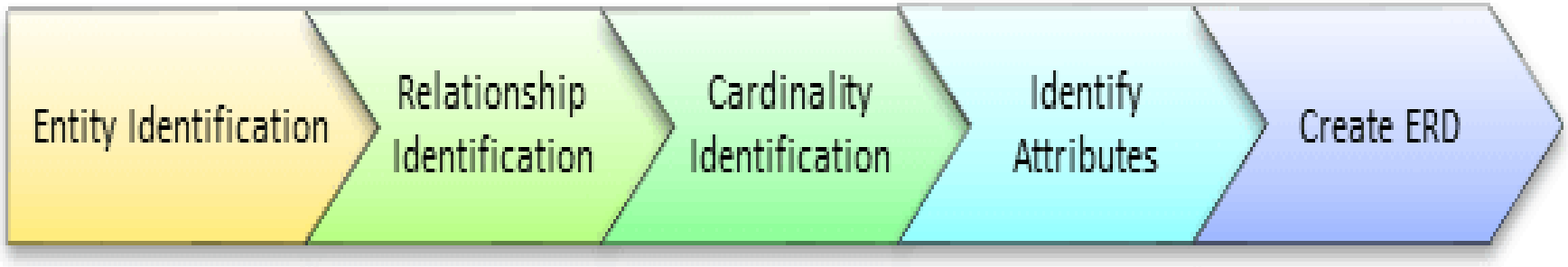
Aggregation

Aggregration is a process when relation between two entities is treated as a **single entity**.



Conceptual Design with the ER Model

- The **ER** or (Entity Relational **Model**) is a high-level **conceptual** data **model diagram**.
- Entity-Relation **model** is based on the real-world entities and the relationship between them.
- **ER modeling** helps you to analyze data requirements systematically to produce a well-designed **database**



Logical database Design

- A **logical design** is a conceptual, abstract **design.**

- The process of **logical design** involves arranging **data** into a series of **logical** relationships called entities and attributes.

- An entity represents a chunk of information. In relational **databases**, an entity often maps to a table.

- Creating a logical data design is an information-gathering, iterative process.
- **It includes the following steps:**
  - *Define the tables you need based on the information your business requires.*
  - *Determine the relationships between the tables.*
  - *Determine the contents (or columns) of each table.*
  - *Normalize the tables to at least the third normal form.*
  - *Determine the primary keys and the column domain.*
- A *domain* is the set of valid values for each column.
- For example, the domain for the customer number can include all positive numbers.

End of Unit 1