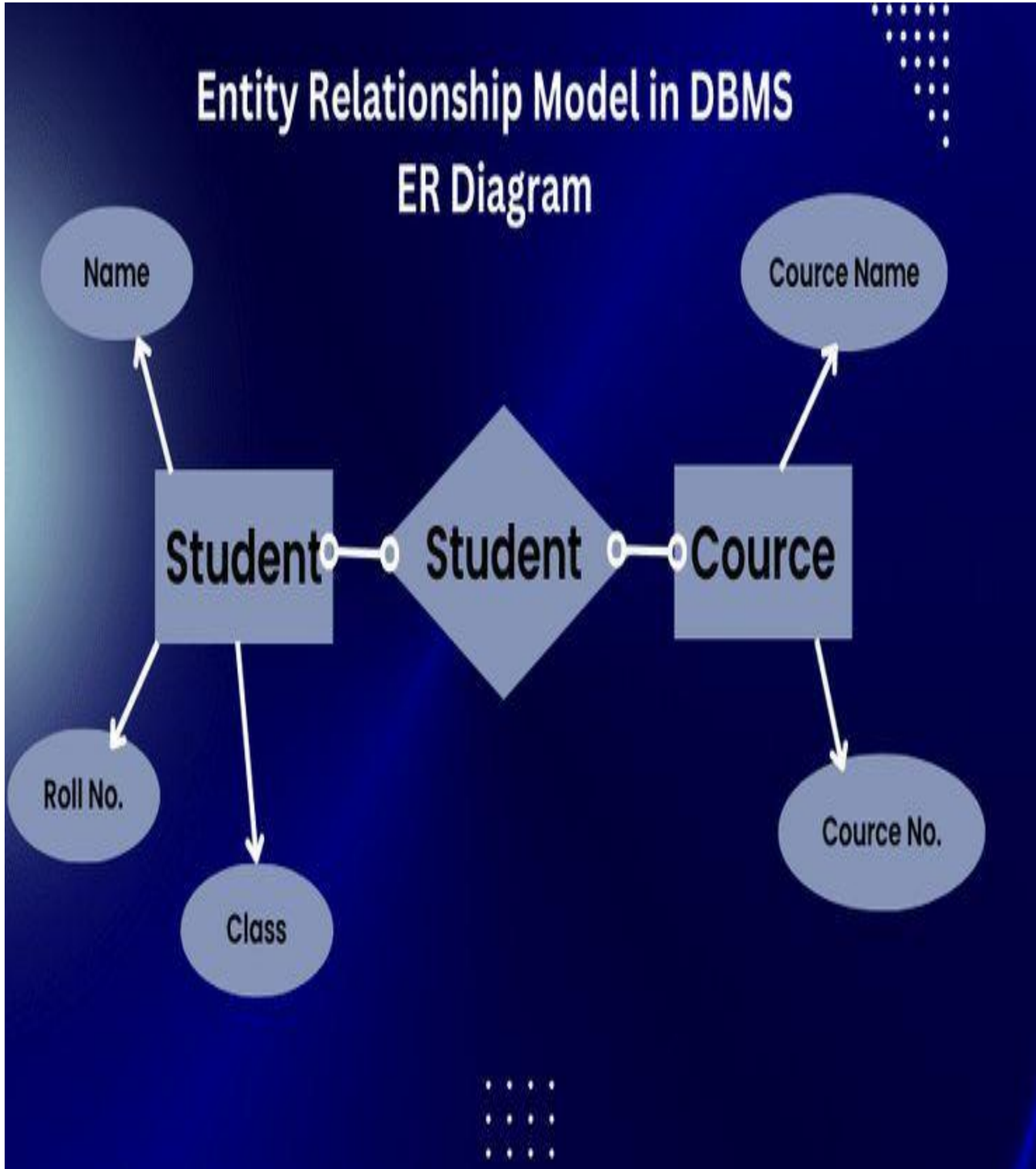
ER Diagram

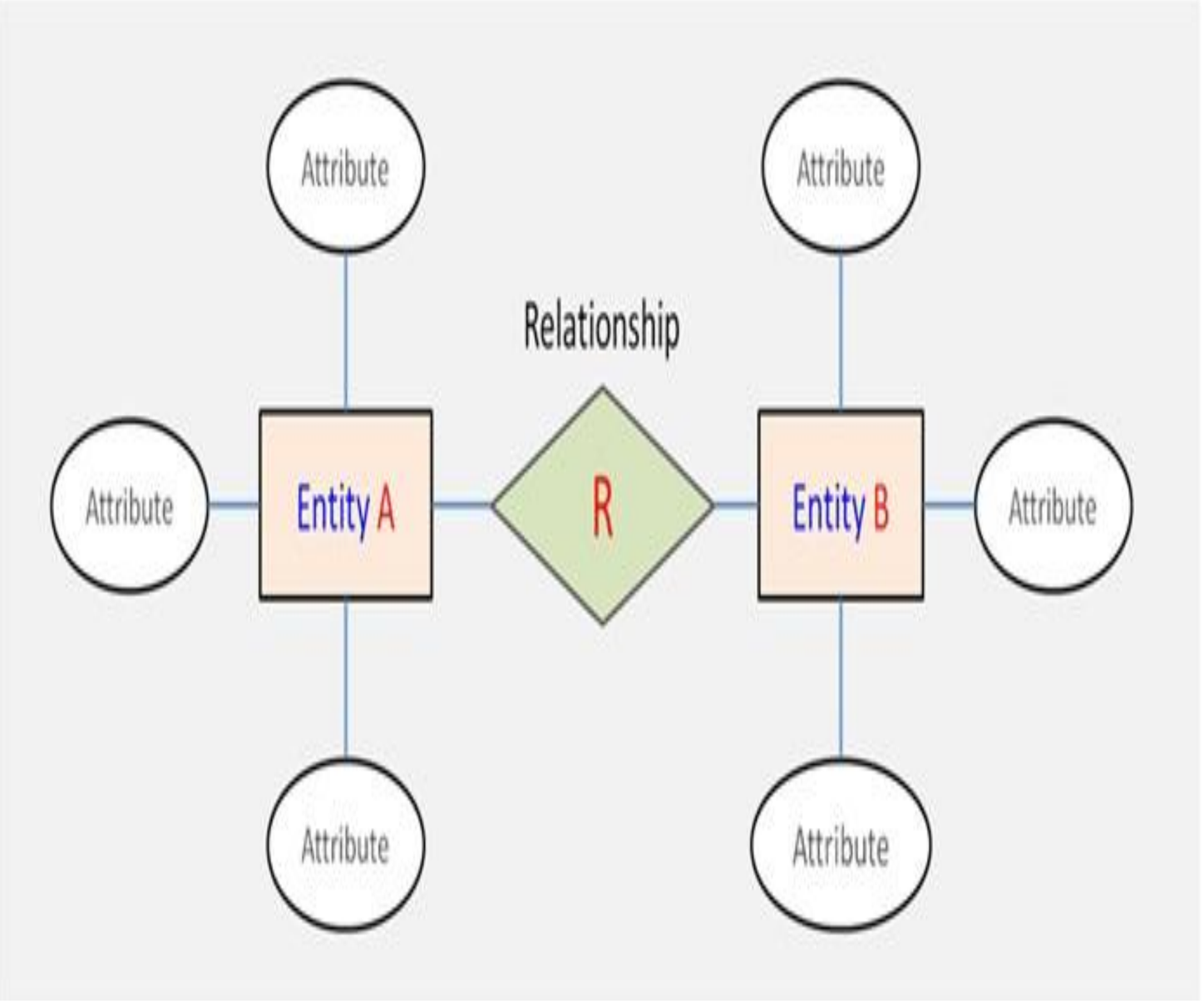
Dr Kiran Waghmare

CDAC Mumbai



ER Diagram (Entity-Relationship Diagram)

- **Definition:**
  - A graphical representation that shows how entities (objects, people, concepts) relate to each other in a database system.
- **Purpose:**
  - Used in database design to structure data logically.
  - Helps in understanding system architecture before implementation.
- **Created by:**
  - Peter Chen in 1976.
- **Real-time Example:**
  - In a university system:
  - Entities: Student, Course, Professor
  - Relationship: Enrolls, Teaches



ER Diagram : History of ER Models

Peter Chen

Charles Bachman

James Martin

Introduced by Peter Chen in 1976.
Later refined by Charles Bachman, James Martin, and others.
Influenced the creation of Unified Modeling Language (UML).

Importance in Database Design

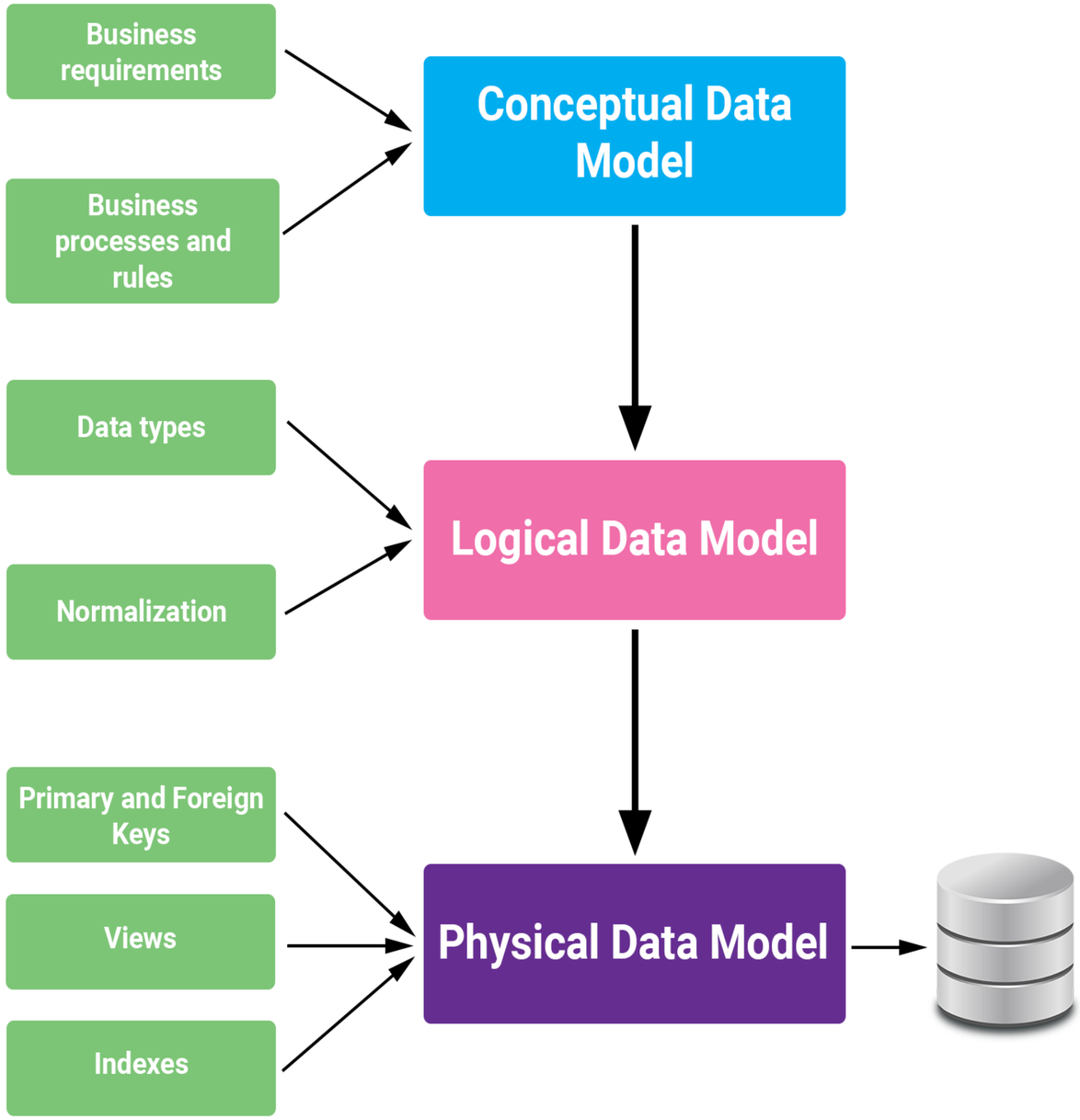
- **Visual clarity:**
  - Makes it easier to understand data structure before implementation
- **Communication tool:**
  - Helps developers and stakeholders communicate requirements
- **Foundation for database schema:**
  - Used to convert into relational schema (tables)
- **Ensures normalization:**
  - Identifies redundant data and promotes better structure

Types of Data Models in DBMS

- Associative Data Model
- Network Model
- Object-Oriented Model
- Relational Model
- Entity-Relationship Model
- Hierarchical Model
- Semi-structured Model
- Object-Relational Model

Types of Data Models

- **1. Conceptual Data Model**
  - Highest-level model with least detail.
  - Focuses on overall scope and architecture.
  - Often omitted for small systems.
- **2. Logical Data Model**
  - More detailed than conceptual.
  - Technology-independent, defines entities, attributes, and relationships clearly.
  - Represents business rules and operations.
- **3. Physical Data Model**
  - Contains technology-specific details.
  - Maps logical design to actual database.
  - Includes indexes, constraints, keys, and data types.
  - May include extended ERD using superclasses and subclasses.

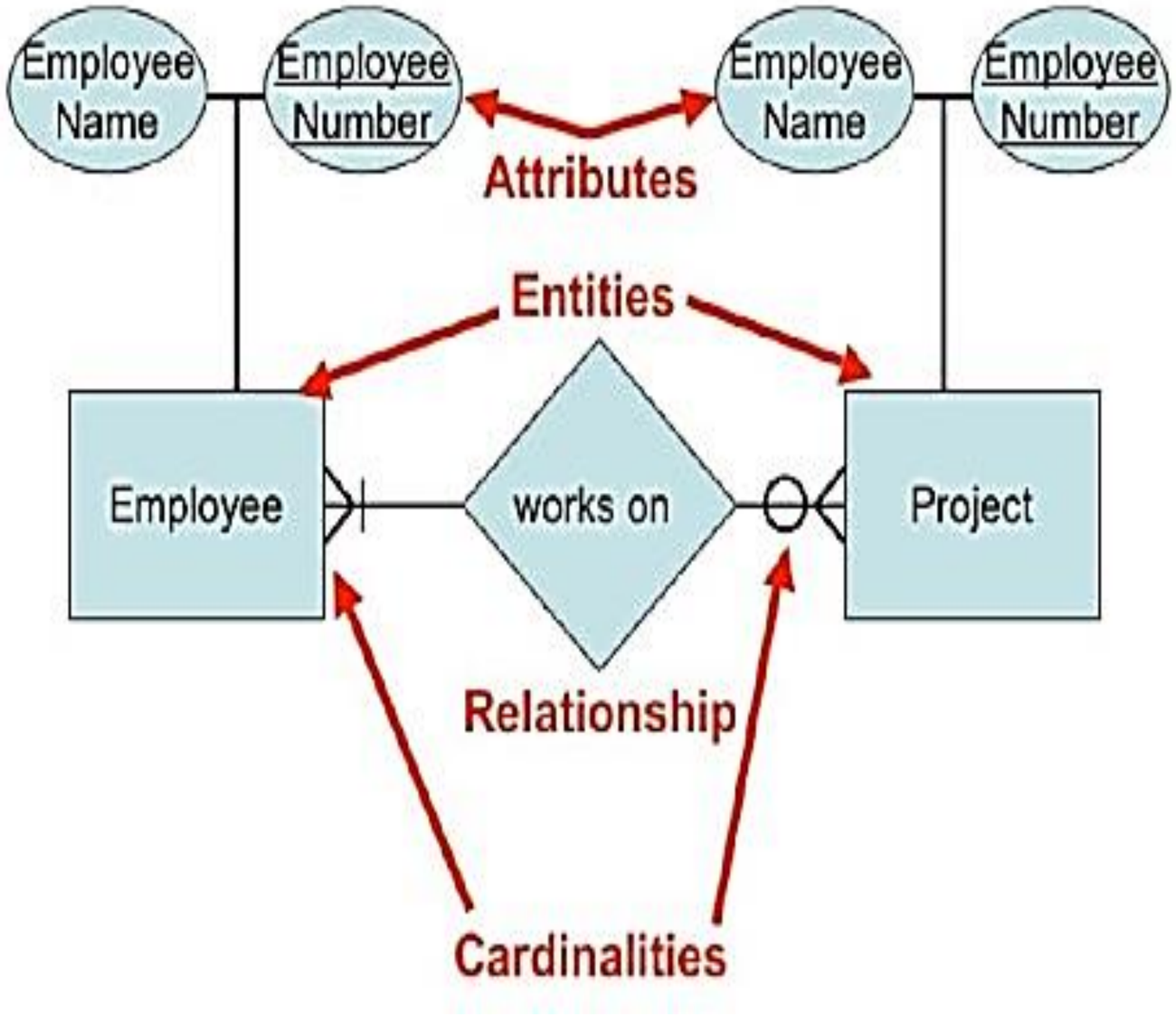


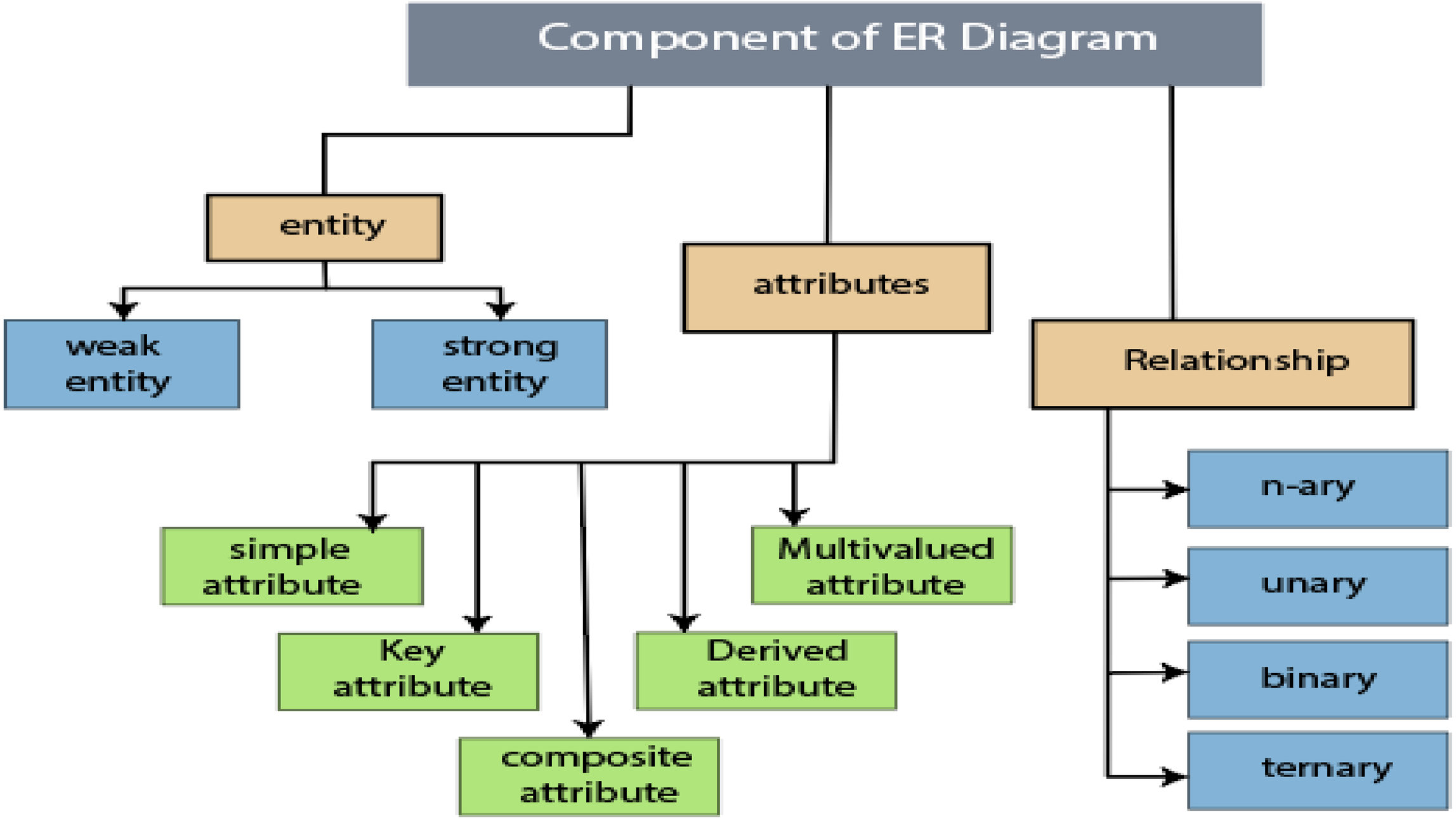
Difference Between ER Model and Relational Model

Feature	ER Model	Relational Model
Purpose	Conceptual design of the database	Logical implementation in DBMS
Representation	Entities, relationships, attributes (graphical)	Tables (rows and columns)
Focus	What data to store and how it relates	How data is stored and queried
Elements	Entity, Attribute, Relationship	Table, Primary Key, Foreign Key
Diagram Tool	ER Diagram	Schema or table definitions

Uses of ER Diagrams

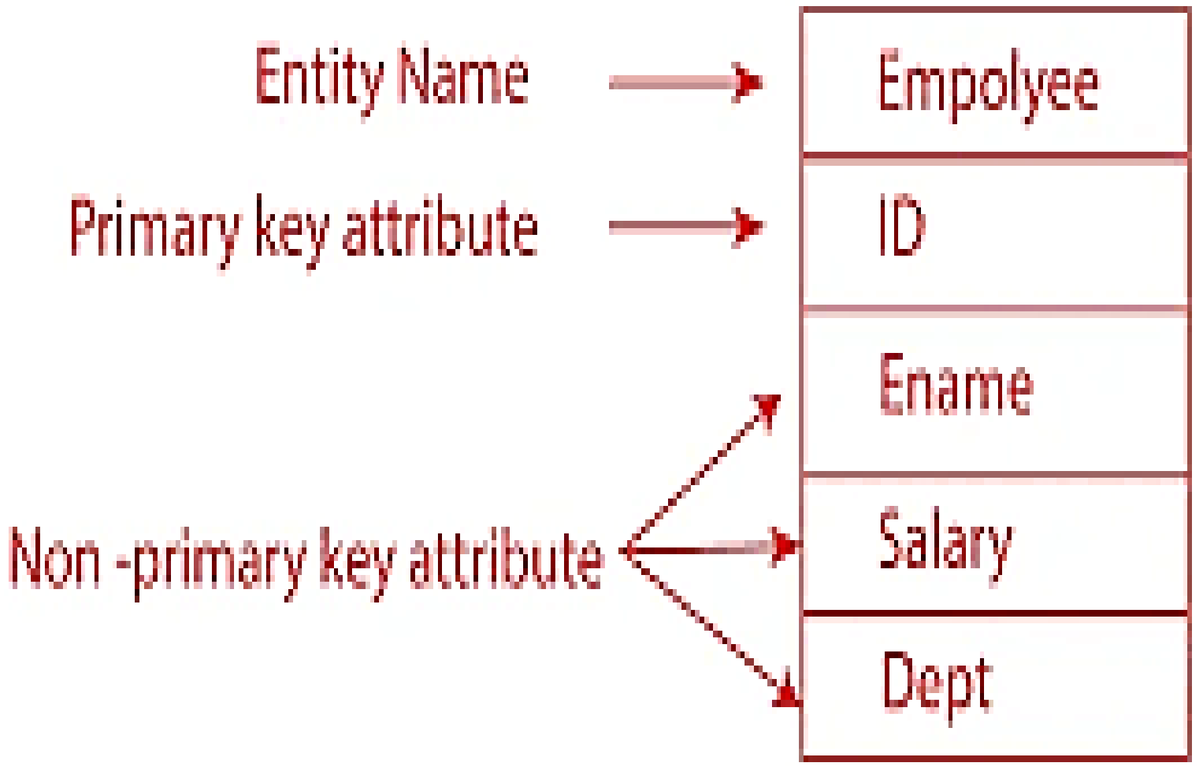
- **Database Design:**
  - Logical and physical layouts
- **Troubleshooting:**
  - Detect logic errors in schemas
- **Business Information Systems:**
  - Streamline data processes
- **BPR**
  - (Business Process Re-engineering)
- **Education & Research:**
  - Structuring and storing relational data

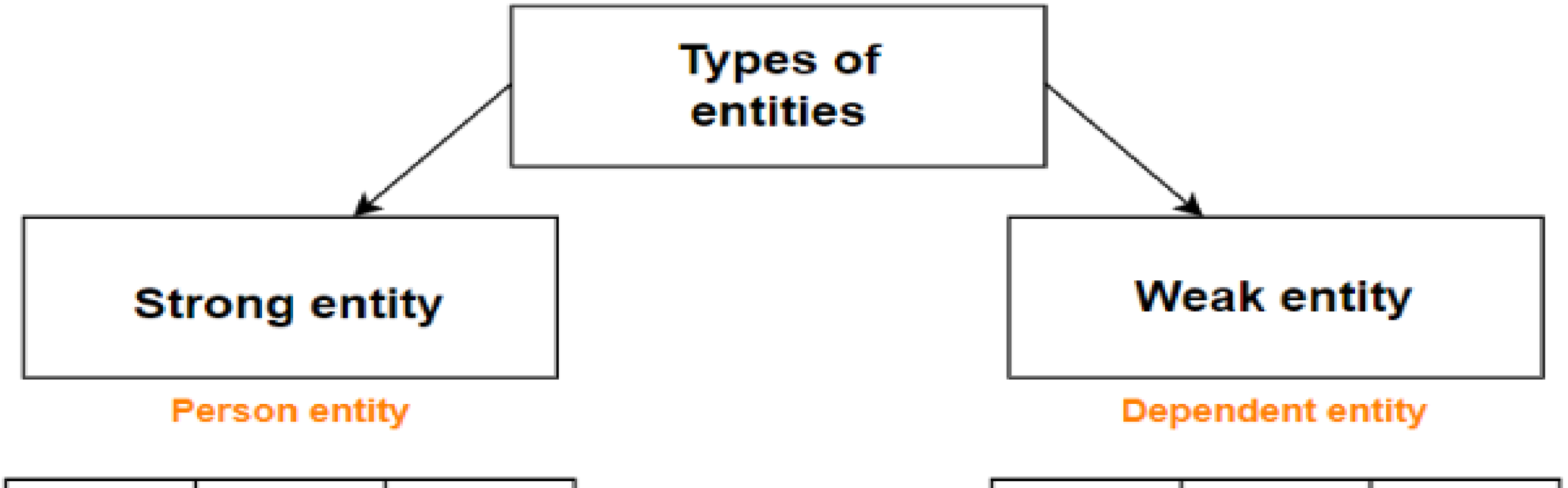



Component of ER Diagram
entity
weak entity
strong entity
attributes
simple attribute
Key attribute
composite attribute
Derived attribute
Multivalued attribute
Relationship
n-ary
unary
binary
ternary

Components of ER Diagrams

- **a. Entity**
  - Real-world object, concept, or event
  - Examples:
    - Employee, Invoice, Classroom
- **b. Entity Categories**
  - **Strong:**
    - Independent (e.g., Student)
  - **Weak:**
    - Depends on another entity (e.g., Dependent)
  - **Associative:**
    - Connects other entities with attributes (e.g., Enrollment with Date)
- **Mapping ERD to Natural Language**
  - Entity = Noun → e.g., Student
  - Attribute = Adjective → e.g., Sophomore
  - Relationship = Verb → e.g., Enrolls





Person entity

S.ID	S.Name	S.Age
1	Riti	20
2	Raj	21

Student table

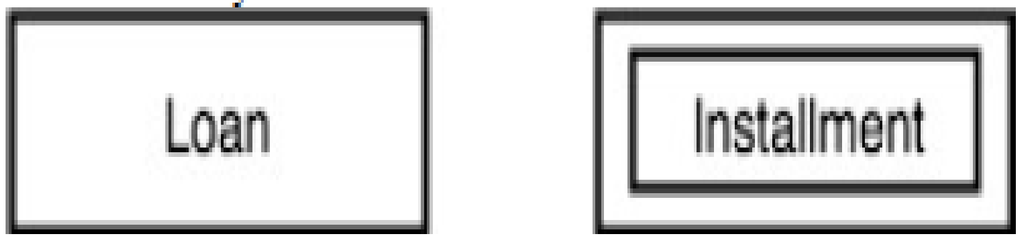
Dependent entity

C.ID	C.Name	S.ID
01	Physics	1
02	Maths	2

Course table

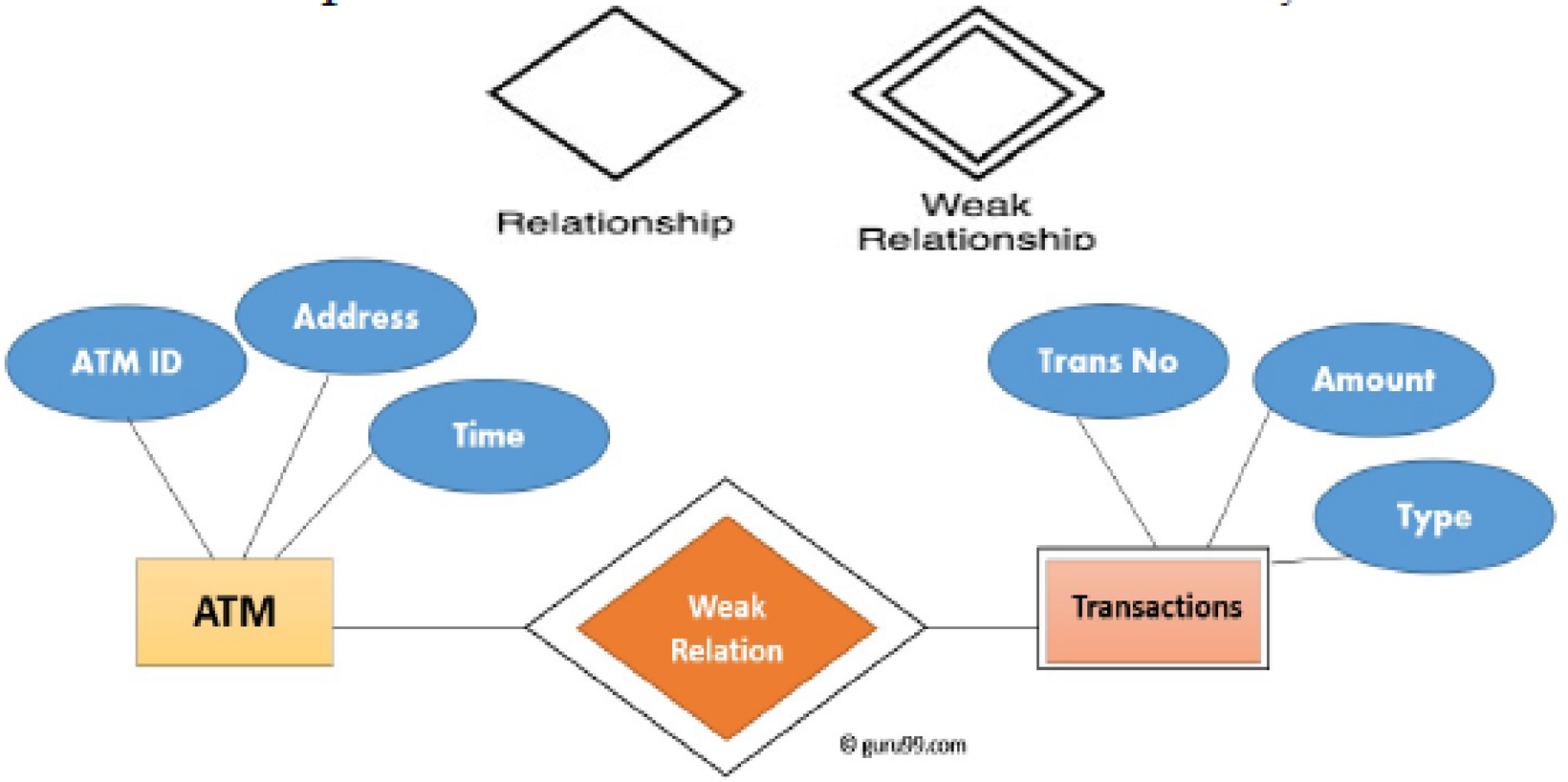
Weak Entity

- A weak entity is a type of entity which doesn't have its key attribute.
- It can be identified uniquely by considering the primary key of another entity.
- They don't have primary keys, and have no meaning in the diagram without their parent entity.
- A weak Entity is represented using double rectangular boxes. It is generally connected to another entity.



Relationships between Entities - Weak and Strong

- Rhombus is used to setup relationships between two or more entities.
- Weak Relationships are connections between a weak entity and its owner.



- In above example, "Trans No" is a discriminator within a group of transactions in an ATM.

Attribute

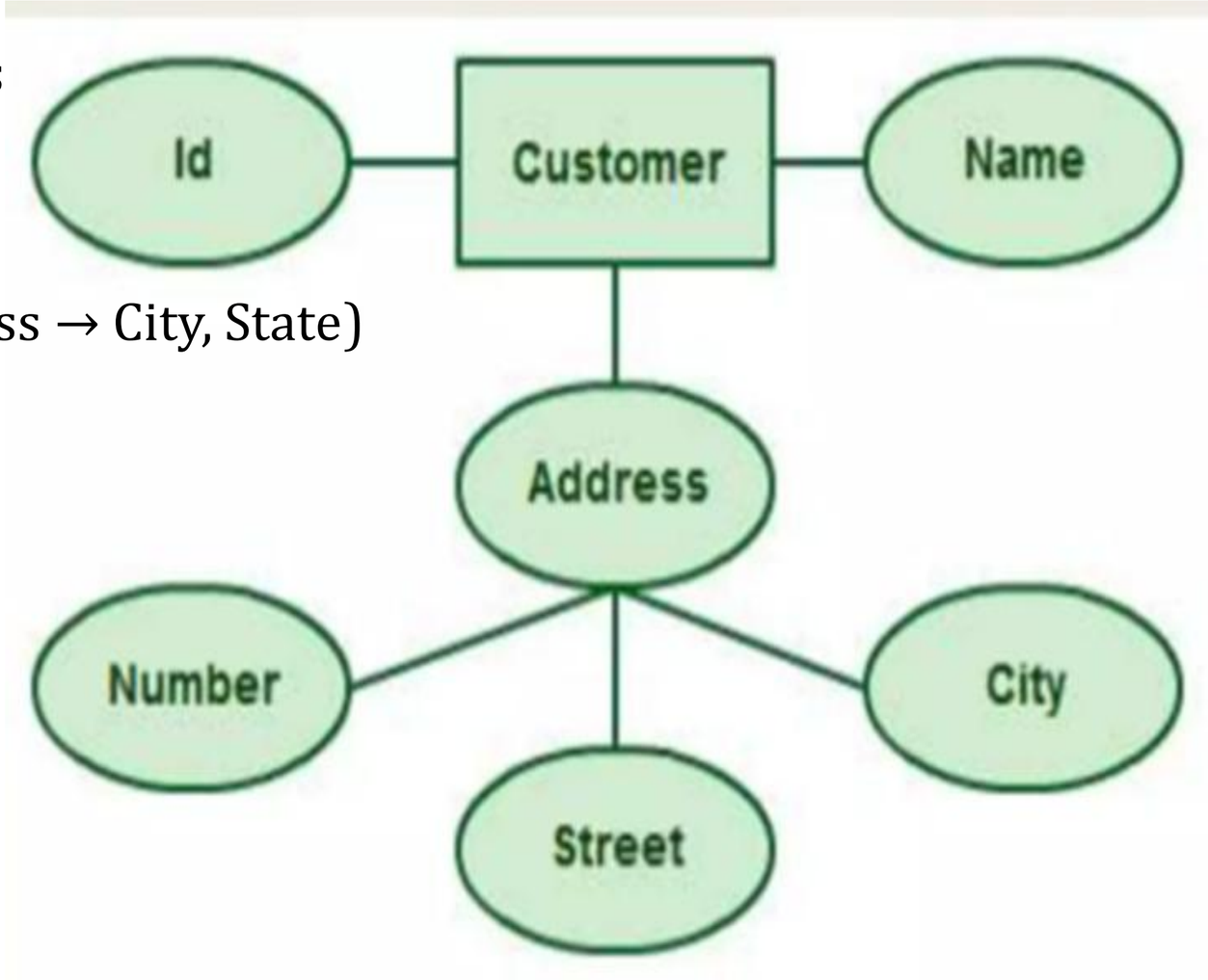
Attribute: A property or characteristic of an entity.

Descriptive attribute: A property or characteristic of a relationship (versus of an entity.)

Often shown as an oval or circle.

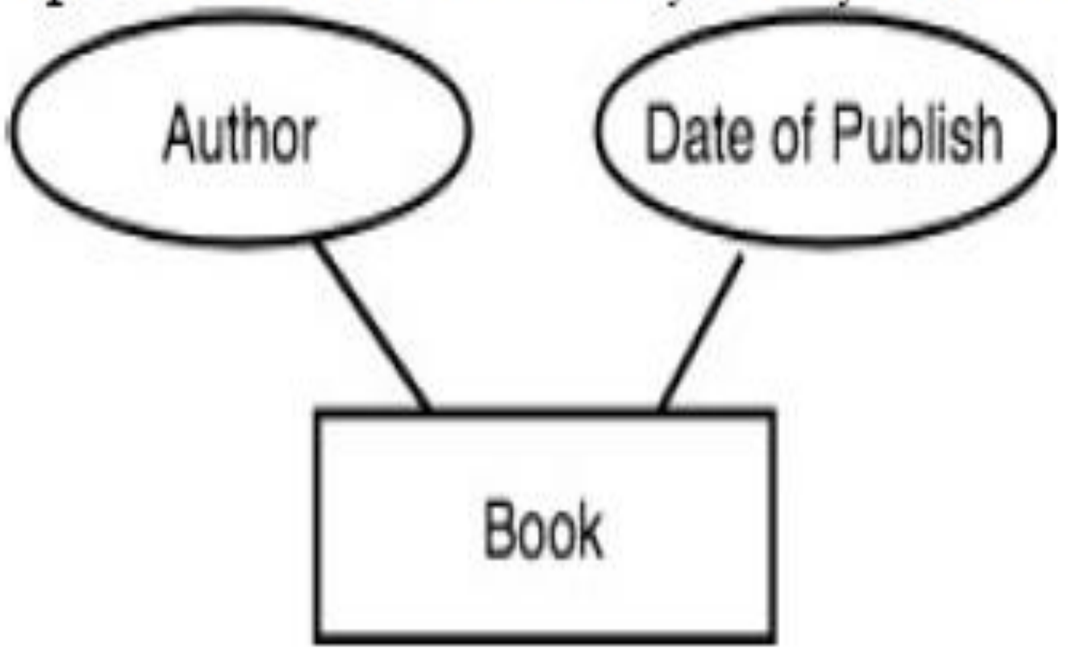
Attributes

- **Attributes**
  - Describe properties of entities/relationships
- **Types:**
  - Simple: Cannot be split (e.g., Age)
  - Composite: Can be broken down (e.g., Address → City, State)
  - Derived: Computed (e.g., Age from DOB)
  - Single-ValuedMulti-Valued
- **Example:**
  - Composite: Address (Street, City, State)
  - Derived: Age = CurrentYear - YearOfBirth



Attributes for any Entity

- Ellipse is used to represent attributes of any entity. It is connected to the entity.

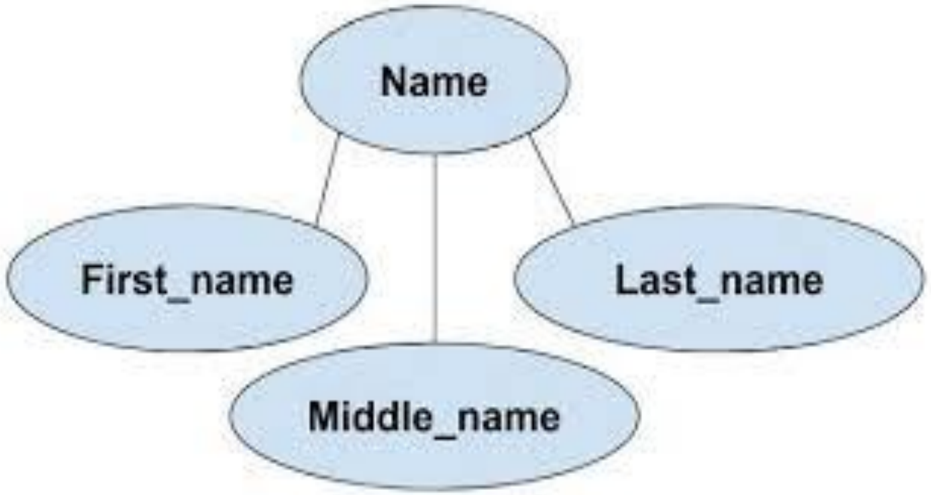


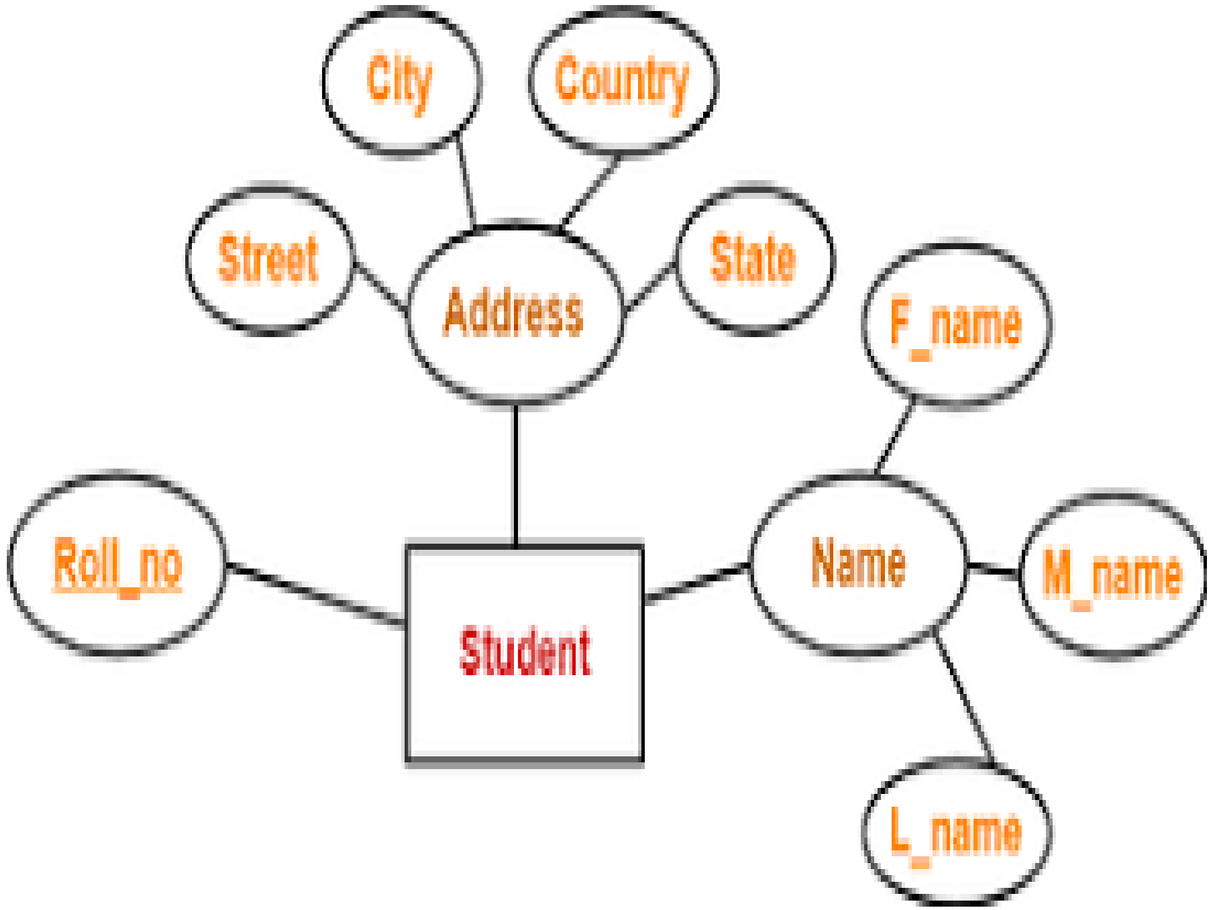
Key Attribute for any Entity

- To represent a Key attribute, the attribute name inside the Ellipse is underlined. (Primary key)

Composite Attribute

Attribute that composed of many other attribute





Derived Attribute for any Entity

- Derived attributes are those which are derived based on other attributes, for example, age can be derived from date of birth.
- To represent a derived attribute, another dotted ellipse is created inside the main ellipse.

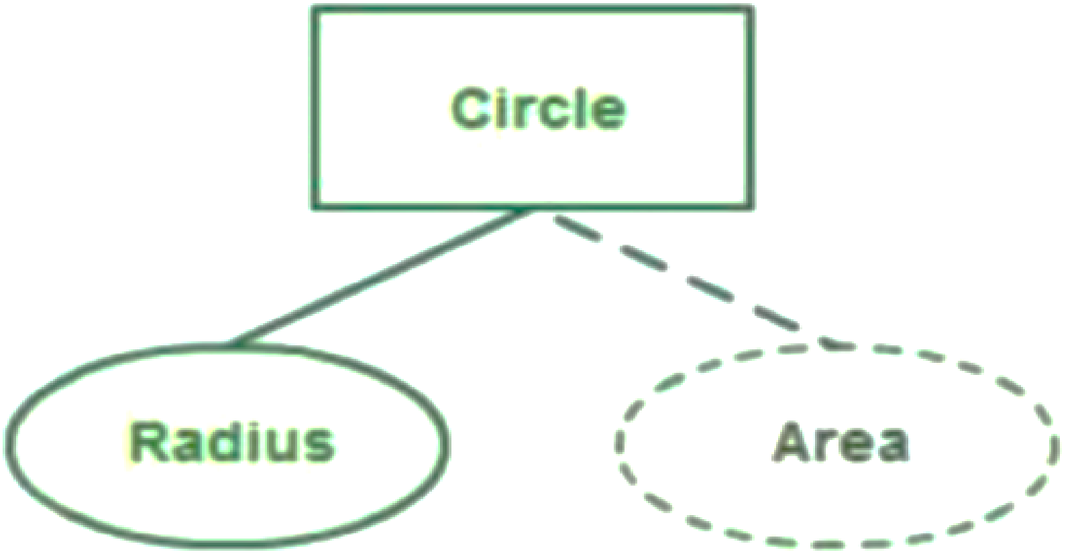
Multivalued Attribute for any Entity

- Multivalued attributes are those that are can take on more than one value.
- Double Ellipse, one inside another, represents the attribute which can have multiple values.

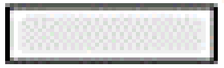
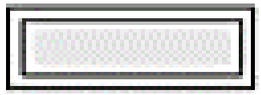
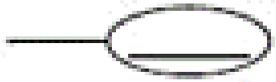
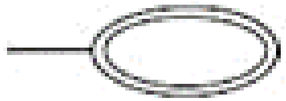
Diagram of Multivalued and Derived Attribute

- Multivalued Attribute

- Derived Attribute



E-R Diagram Symbols and their meaning

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Indentifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute

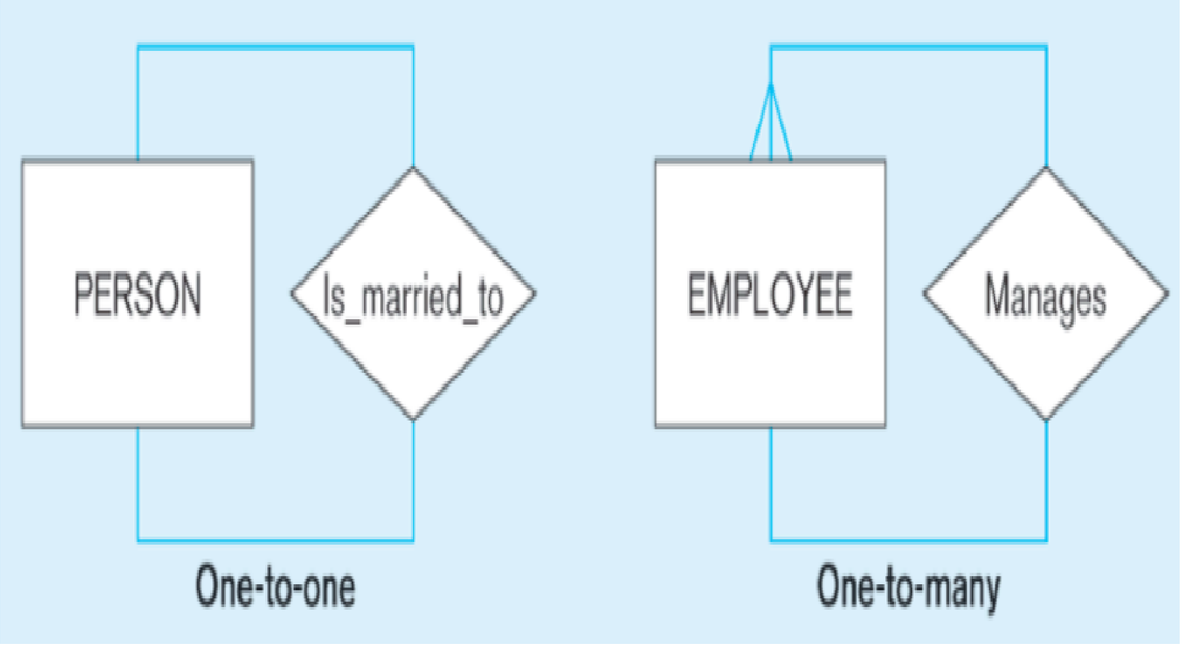
Relationships

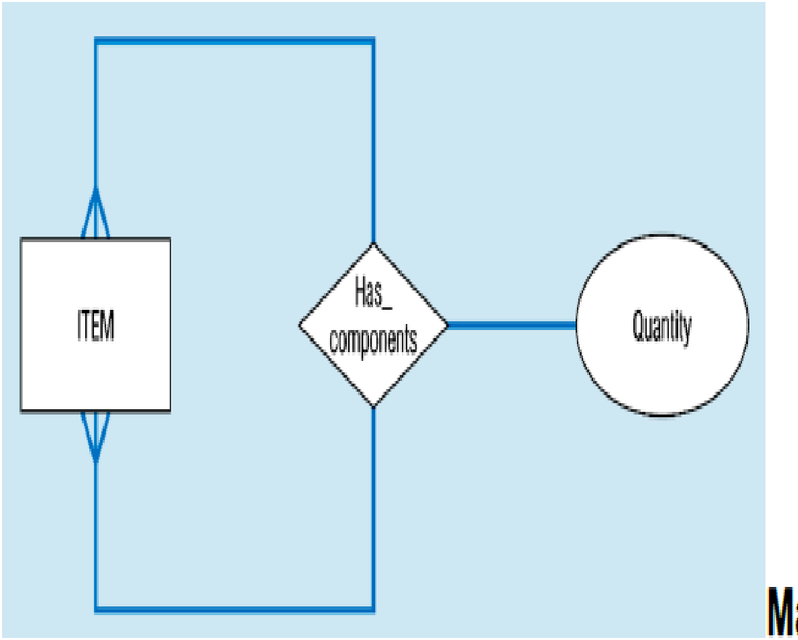
- **Represent associations between entities**
- **Types:**
  - **Unary**: Self-linking (e.g., Person married to Person)
  - **Binary**: Two entities (e.g., Student–Course)
  - **Ternary**: Three entities (e.g., Supplier–Product–Warehouse)
  - **N-ary**: More than 3 entities

- **Example:**
  - Student enrolls in Course
  - Doctor performs Surgery in OperationRoom

1. Unary Relationship

When there is **only ONE entity set participating in a relation**, the relationship is called as unary relationship. For example, one person is married to only one person.

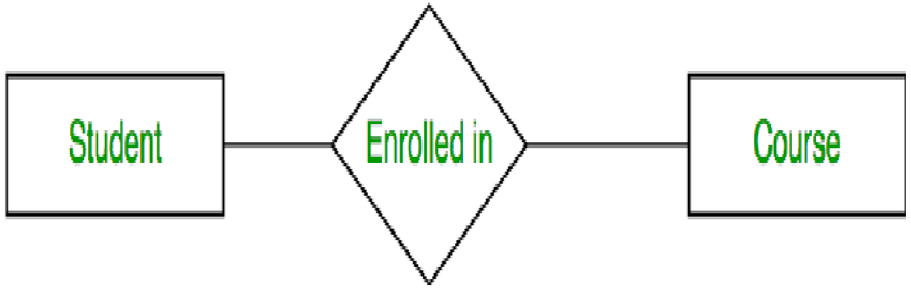




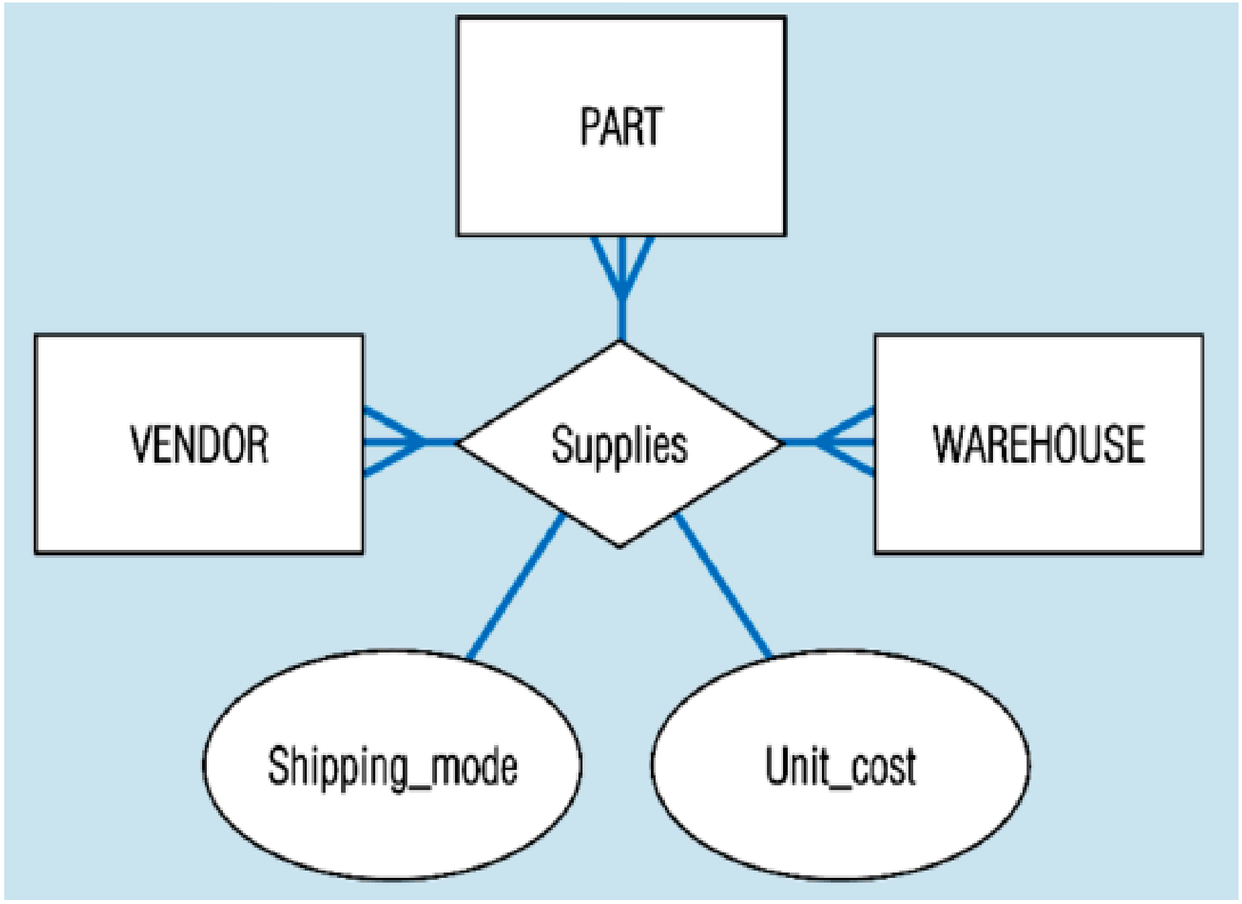
Many-to-many

2. Binary Relationship

When there are **TWO entities set participating in a relation**, the relationship is called as binary relationship.For example, Student is enrolled in Course.



3. Ternary Relationship

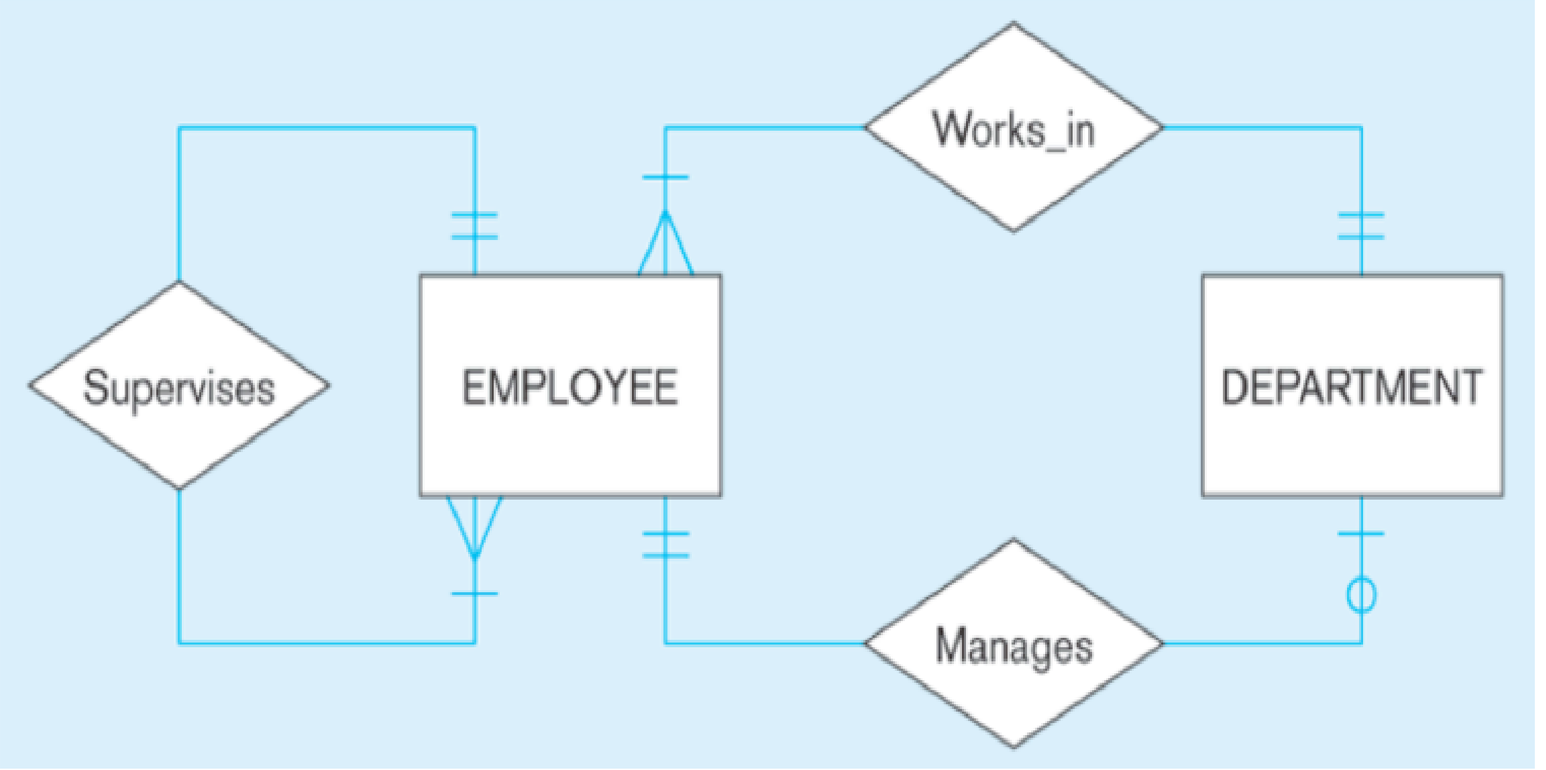


4. n-ary Relationship

When there are n entities set participating in a relation, the relationship is called as n-ary relationship.

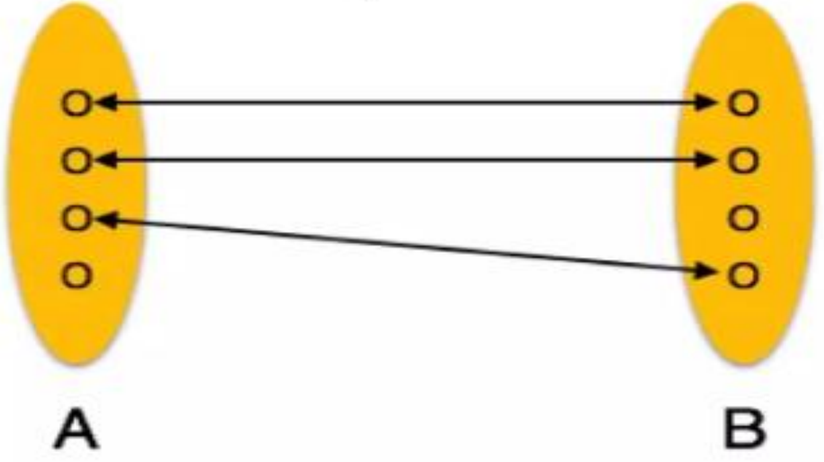
Multiple relationship

Entities can be related to one another in more than one way

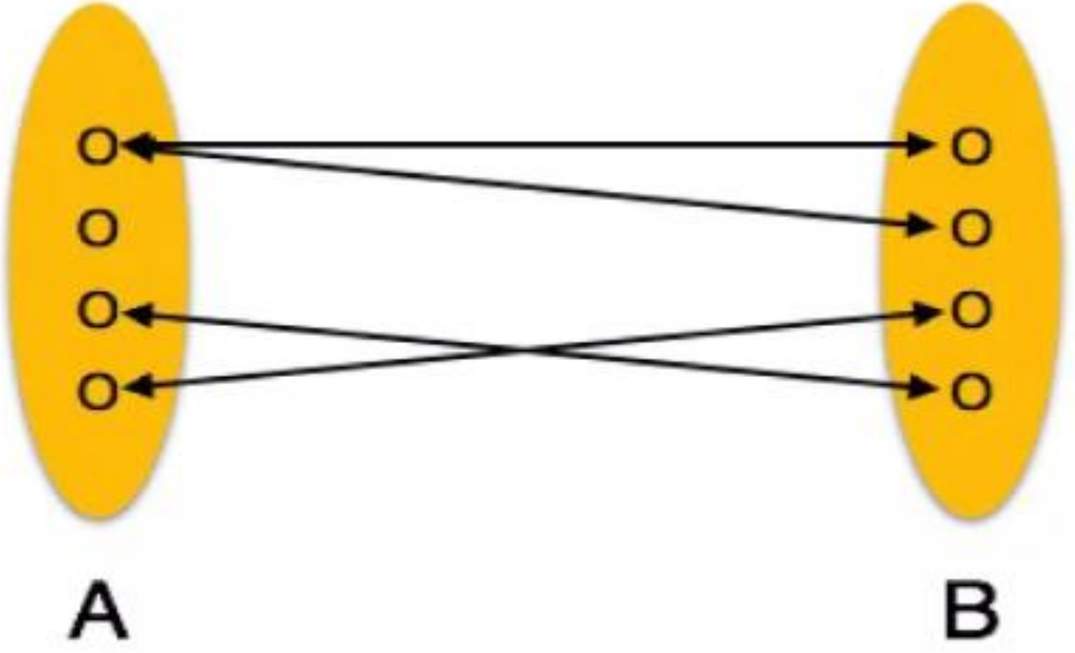


Mapping Cardinalities

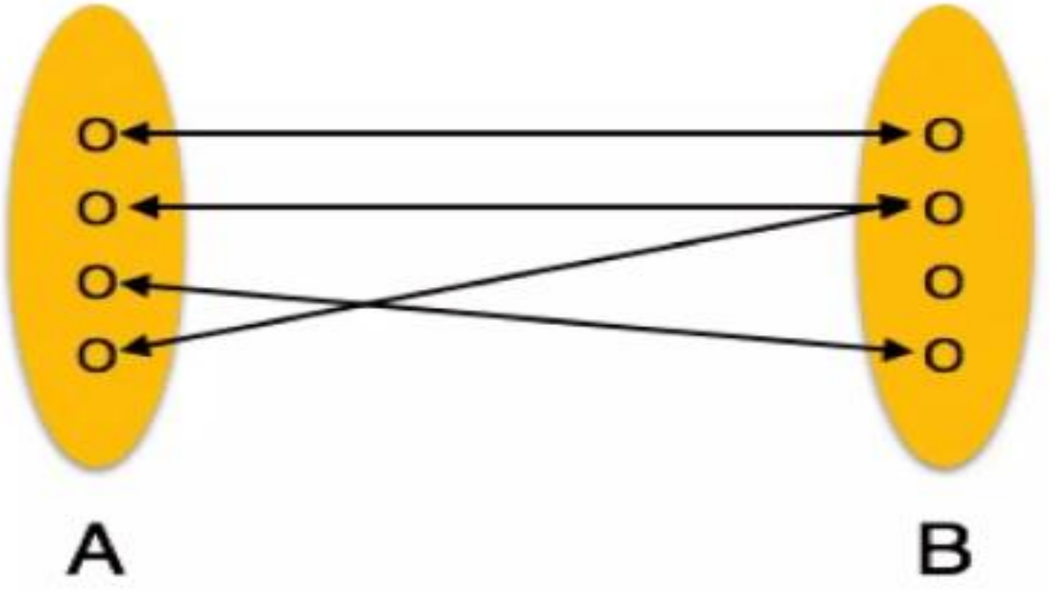
- **Cardinality** defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.
- **One-to-one** – One entity from entity set A can be associated with at most one entity of entity set B and vice versa.



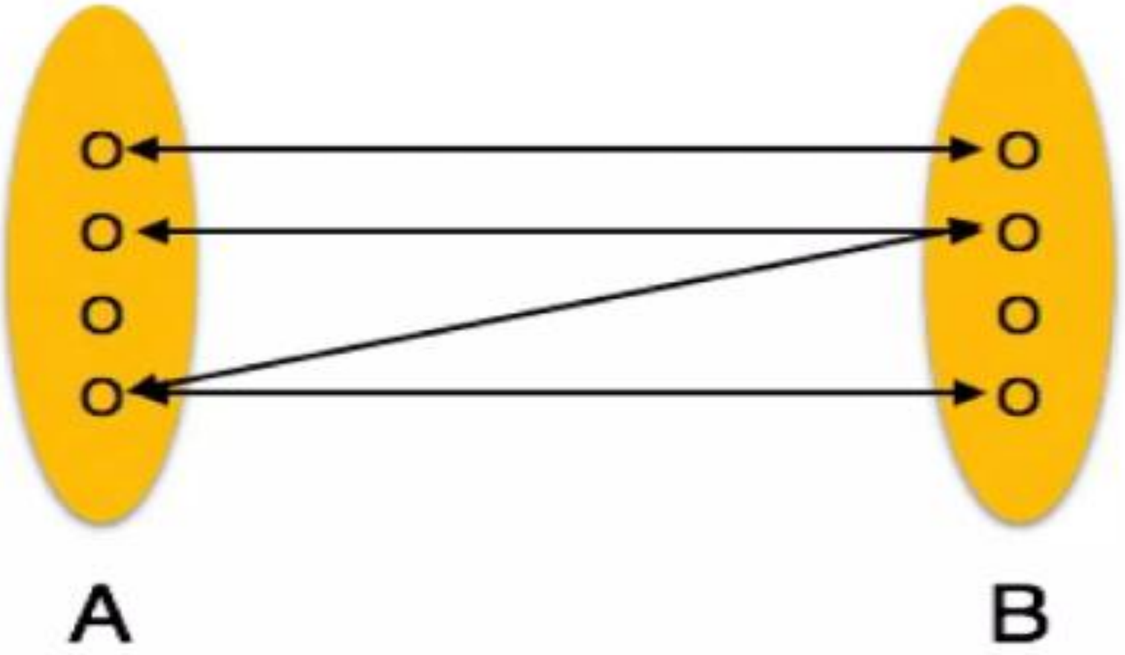
- **One-to-many** – One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.

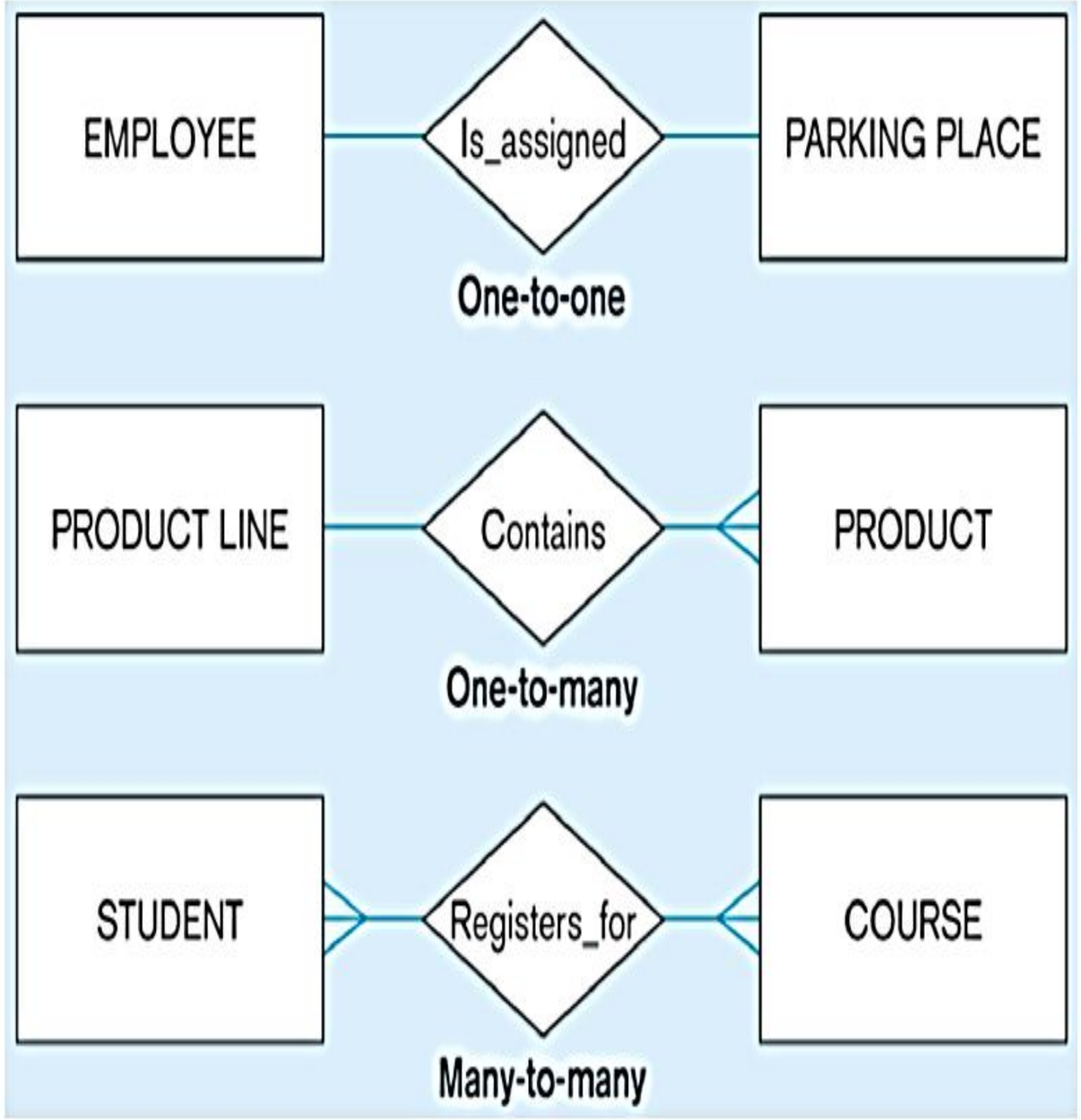


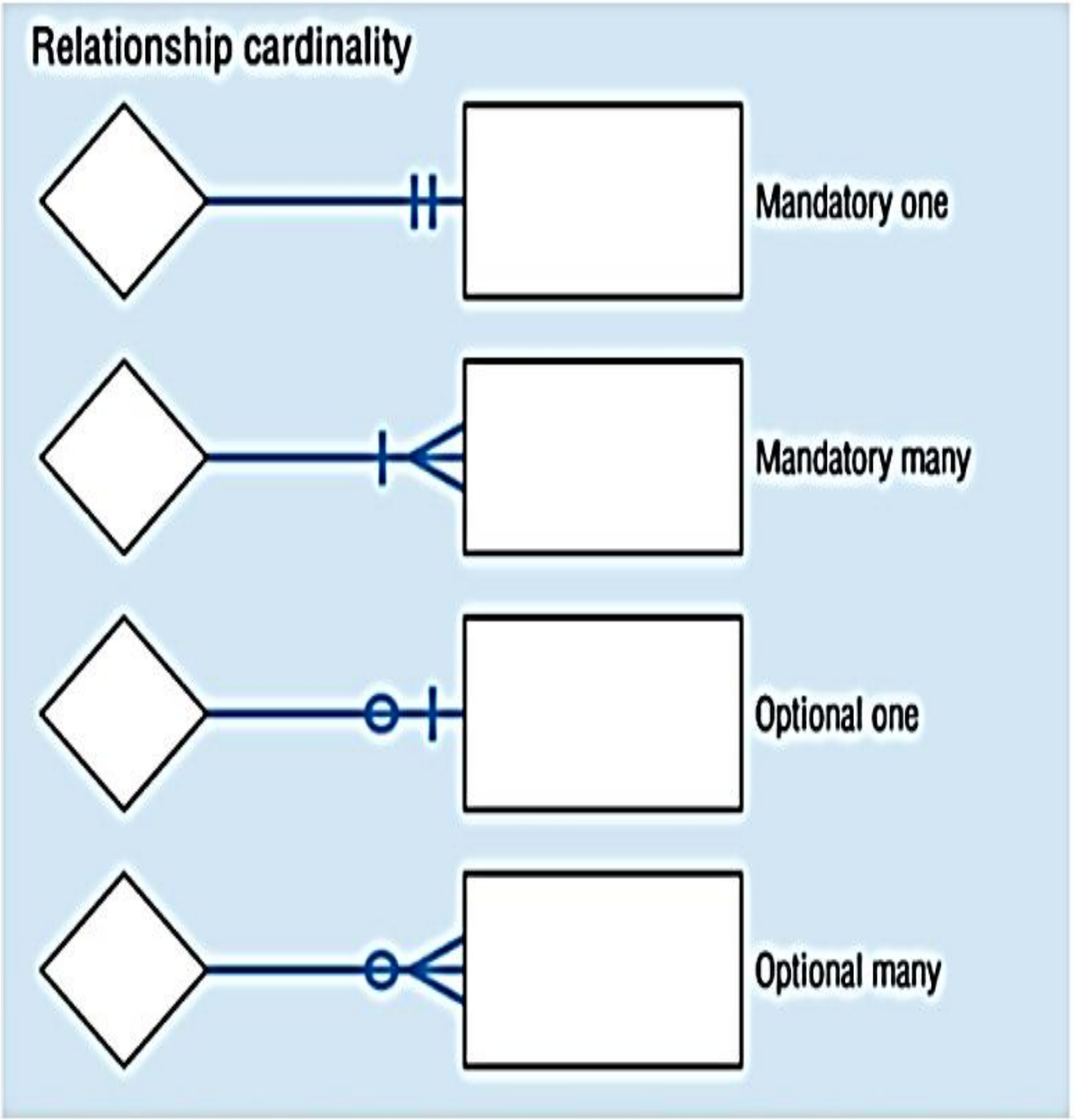
- **Many-to-one** – More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.



- **Many-to-many** – One entity from A can be associated with more than one entity from B and vice versa.

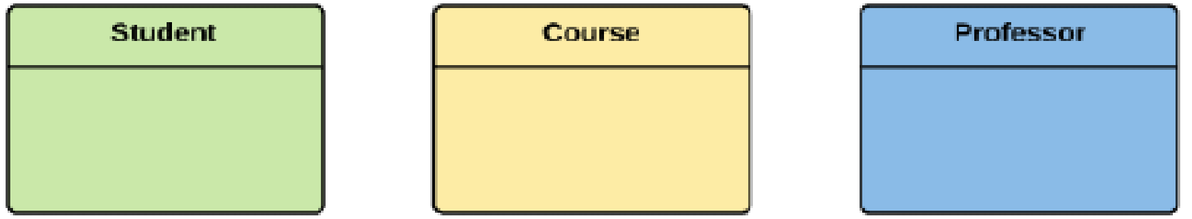



EMPLOYEE
Is_assigned
PARKING PLACE
One-to-one
PRODUCT LINE
Contains
PRODUCT
One-to-many
STUDENT
Registers_for
COURSE
Many-to-many

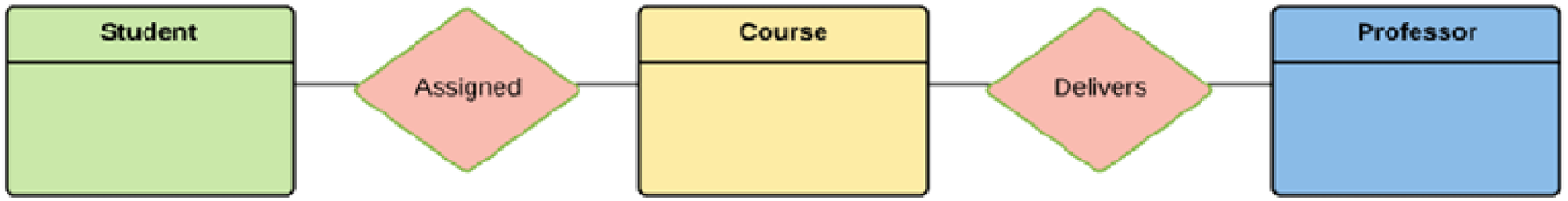

Relationship cardinality
Mandatory one
Mandatory many
Optional one
Optional many

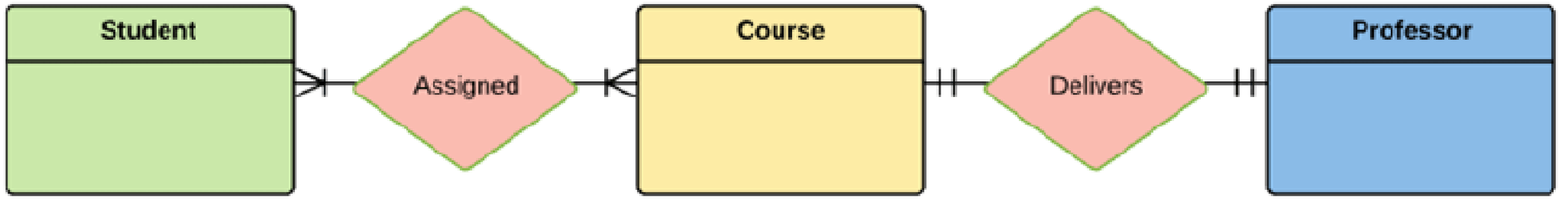
Steps to Create an ERD

2.

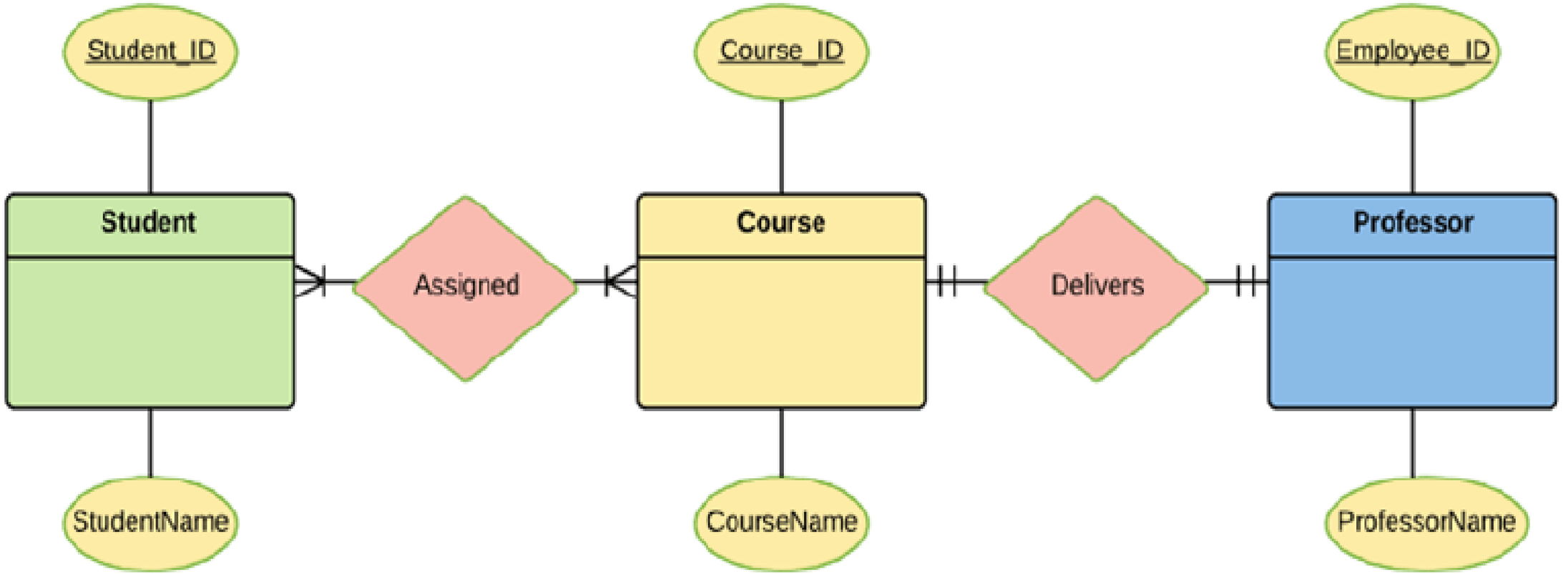


3.





4,5.

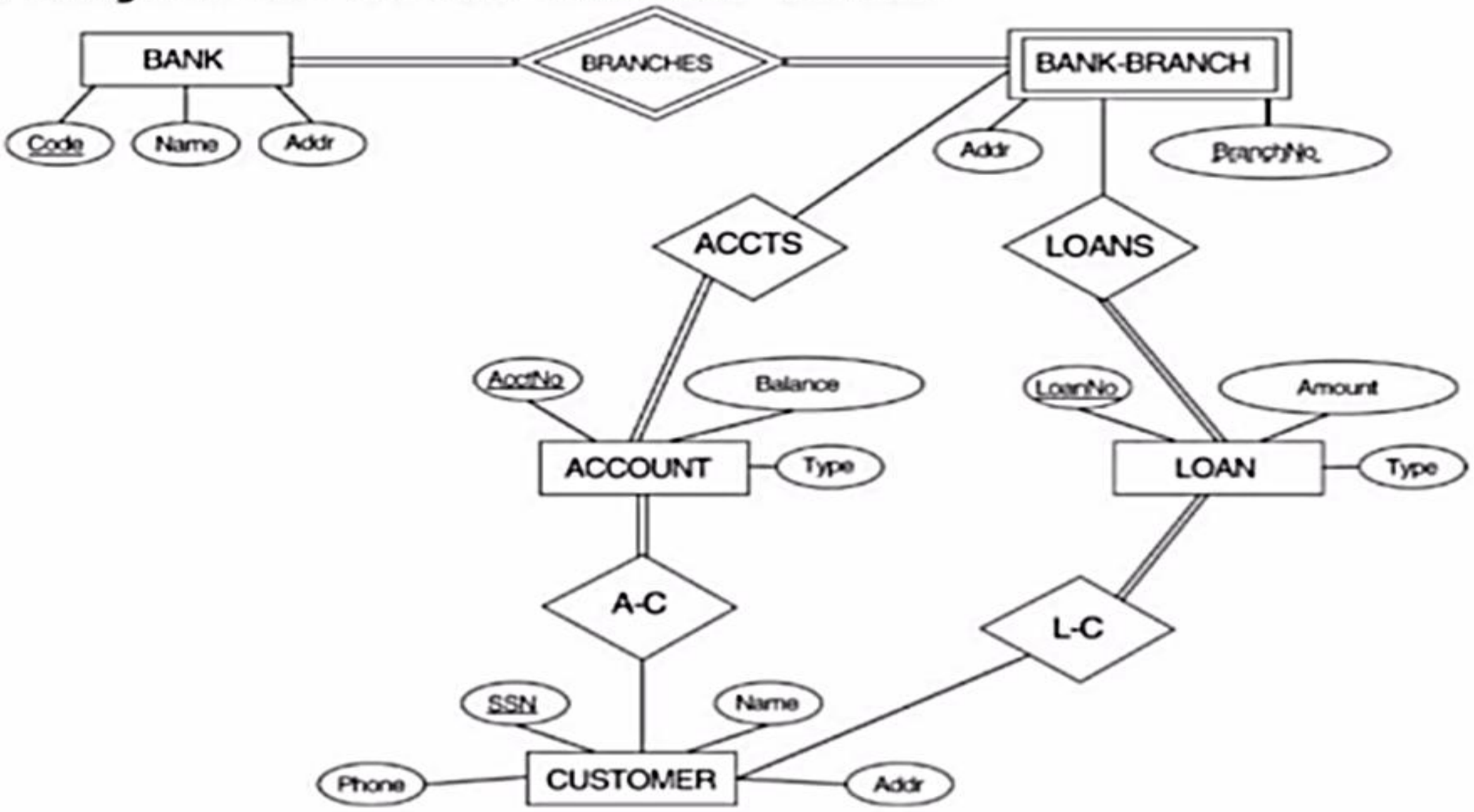

Student_ID
Course_ID
Employee_ID
Student
Assigned
Course
Delivers
Professor
StudentName
CourseName
ProfessorName

6.


Student
Student_ID
StudentName
Assigned
Course
Course_ID
CourseName
Delivers
Professor
Employee_ID
ProfessorName

7.

An ER Diagram for a Bank database Schema



Enhanced / Extended ER Model (EER Model)

- **Key Concepts in EER Model**
- **Specialization**
  - Breaking a high-level entity into more specific sub-entities.
  - Example: Employee → Manager, Developer
- **Generalization**
  - Combining multiple lower-level entities into a higher-level entity.
  - Example: Car and Bike → Vehicle
- **Aggregation**
  - Treating a relationship as an abstract entity.
  - Useful when relationships participate in other relati
- **Subclass and Superclass**
  - Subclass: More specific entity (inherits from superclass).
  - Superclass: More general entity.
  - Example: Person (superclass) → Student, Teacher (subclasses)
- **Union / Category**
  - A subclass that is derived from multiple superclasses.
  - Superclasses can belong to different entity sets.

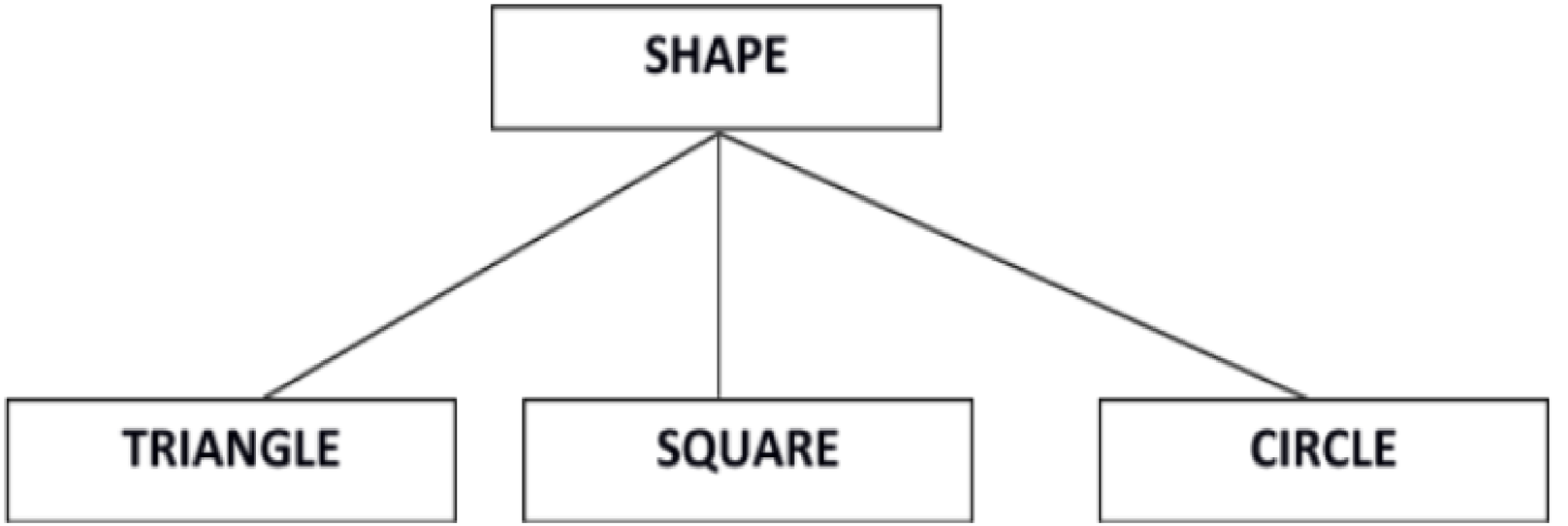
EER Diagram Features

- Uses double rectangles for subclasses/superclasses.
- ISA triangles to represent inheritance ("is a" relationships).
- Dashed boxes/lines for aggregations.

Subclasses and Super class

- Super class is an entity that can be divided into further subtype.

- For **example** – consider Shape super class.

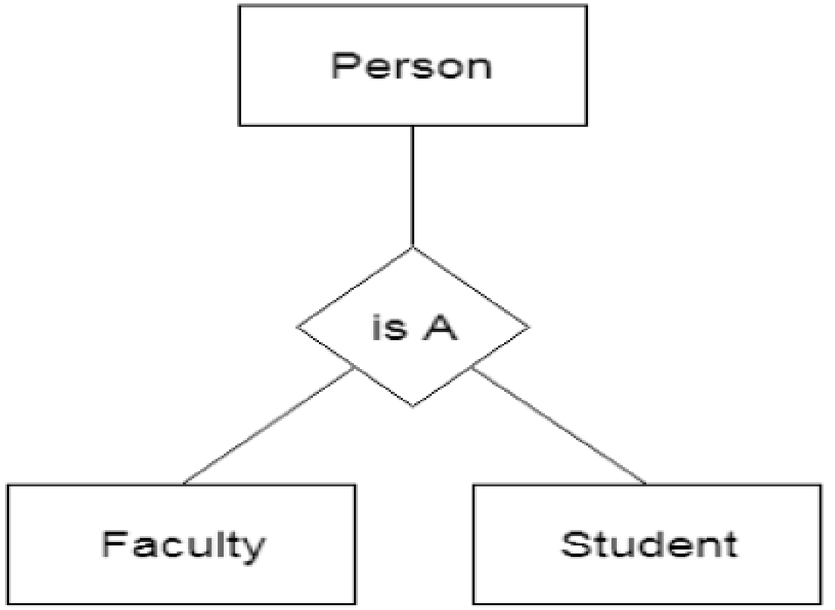


- Super class shape has sub groups: Triangle, Square and Circle.
- Sub classes are the group of entities with some unique attributes. Subclass inherits the properties and attributes from super class.

Specialization and Generalization

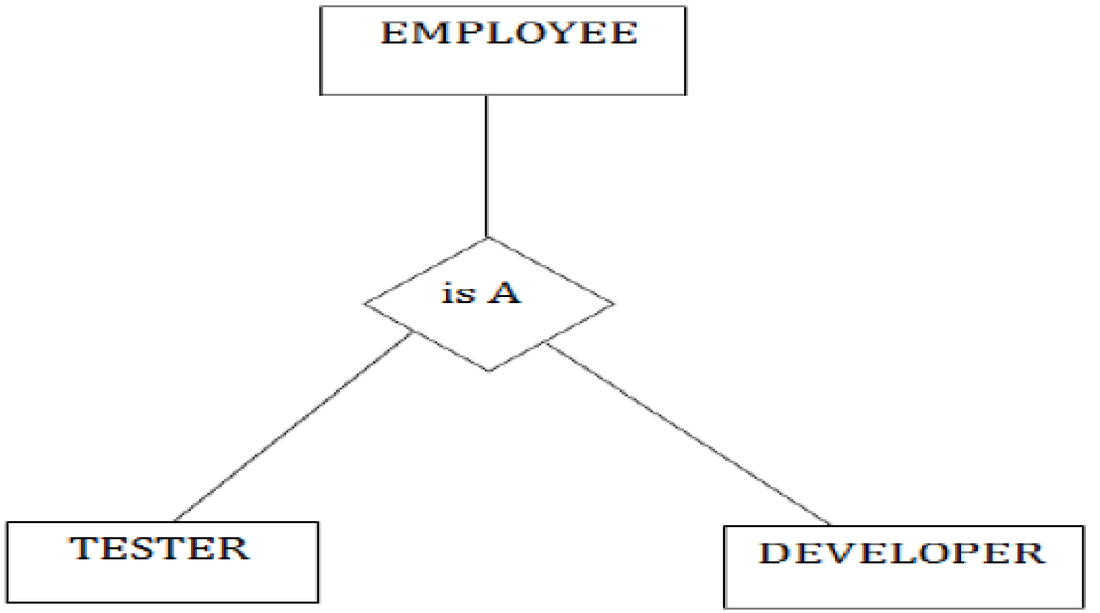
- **Generalization** is like a bottom-up approach in which two or more entities of lower level combine to form a higher level entity if they have some attributes in common.
- In generalization, entities are combined to form a more generalized entity, i.e., subclasses are combined to make a superclass.

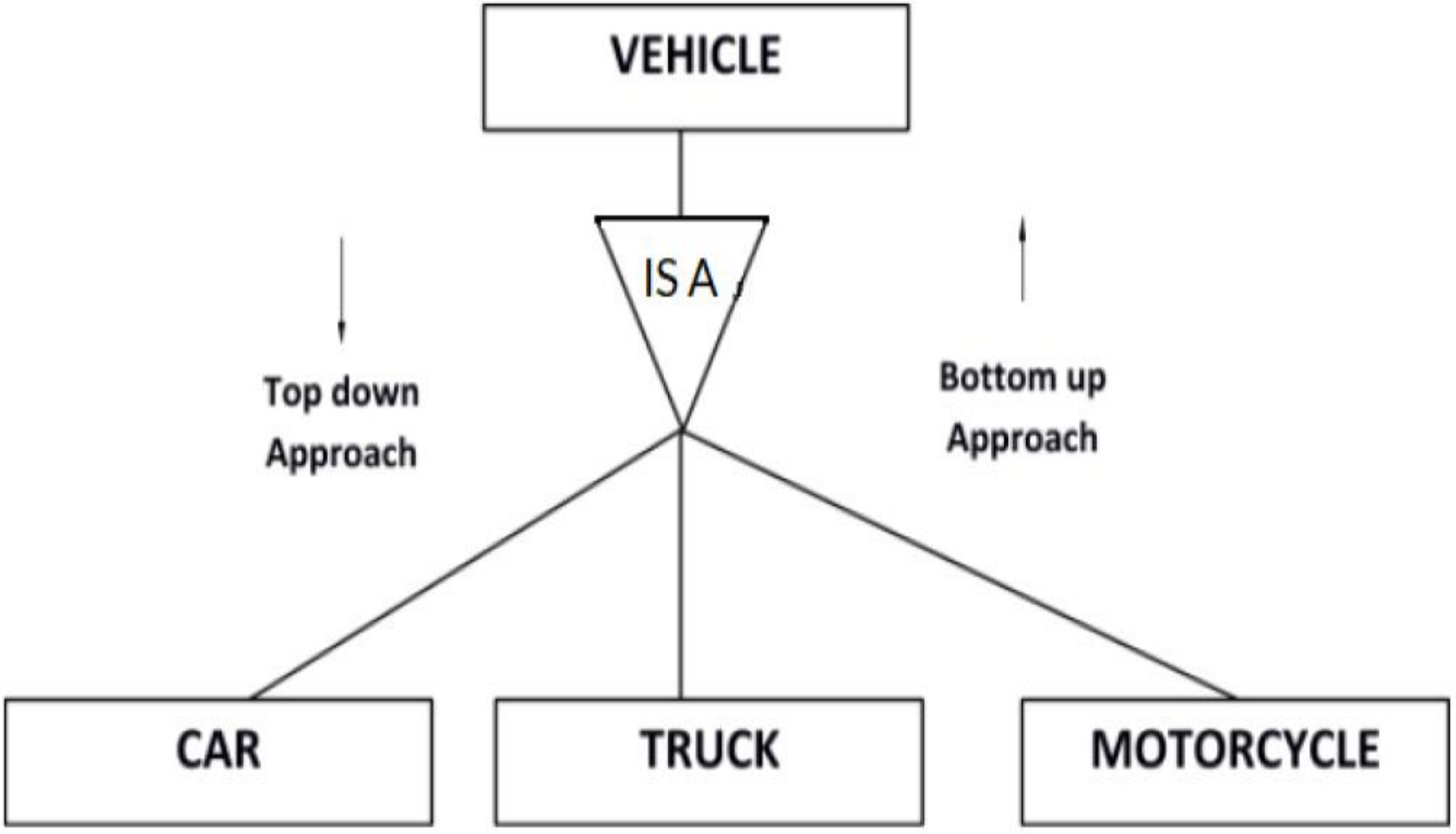
For example, Faculty and Student entities can be generalized and create a higher level entity Person.



- **Specialization** is a top-down approach, and it is opposite to Generalization. In specialization, one higher level entity can be broken down into two lower level entities.
- Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics.
- Normally, the superclass is defined first, the subclass and its related attributes are defined next, and relationship set are then added.

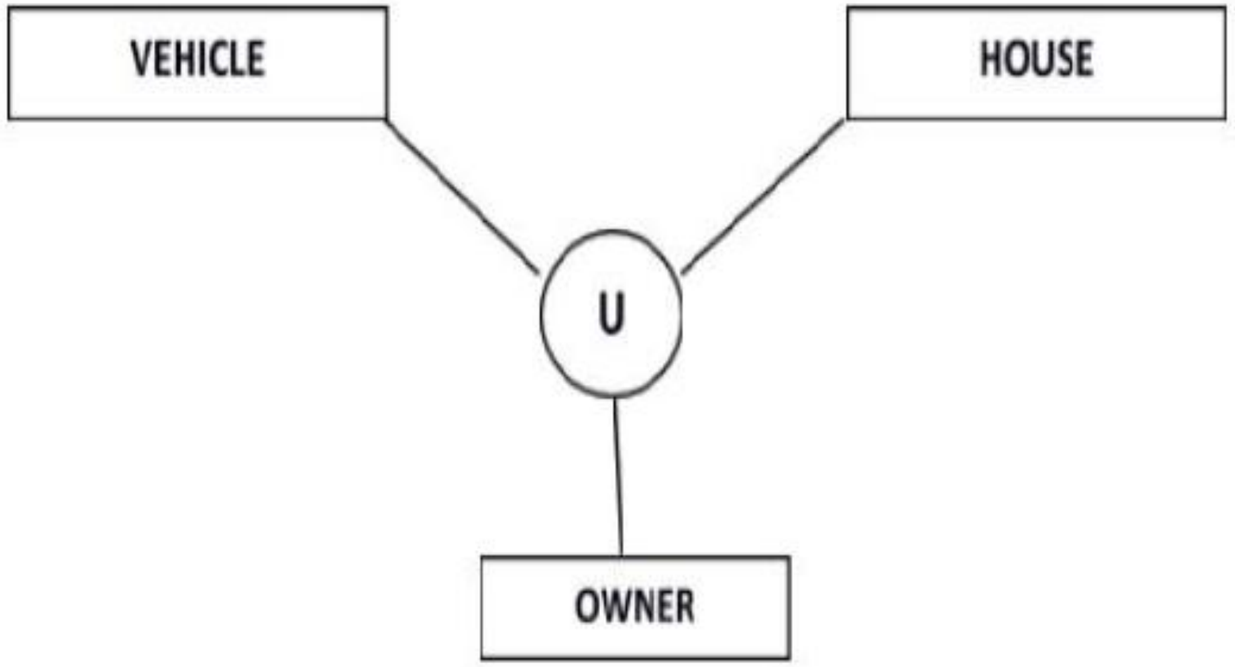
For example: In an Employee management system, EMPLOYEE entity can be specialized as TESTER or DEVELOPER based on what role they play in the company.




VEHICLE
IS A
Top down
Approach
Bottom up
Approach
CAR
TRUCK
MOTORCYCLE

Category or Union

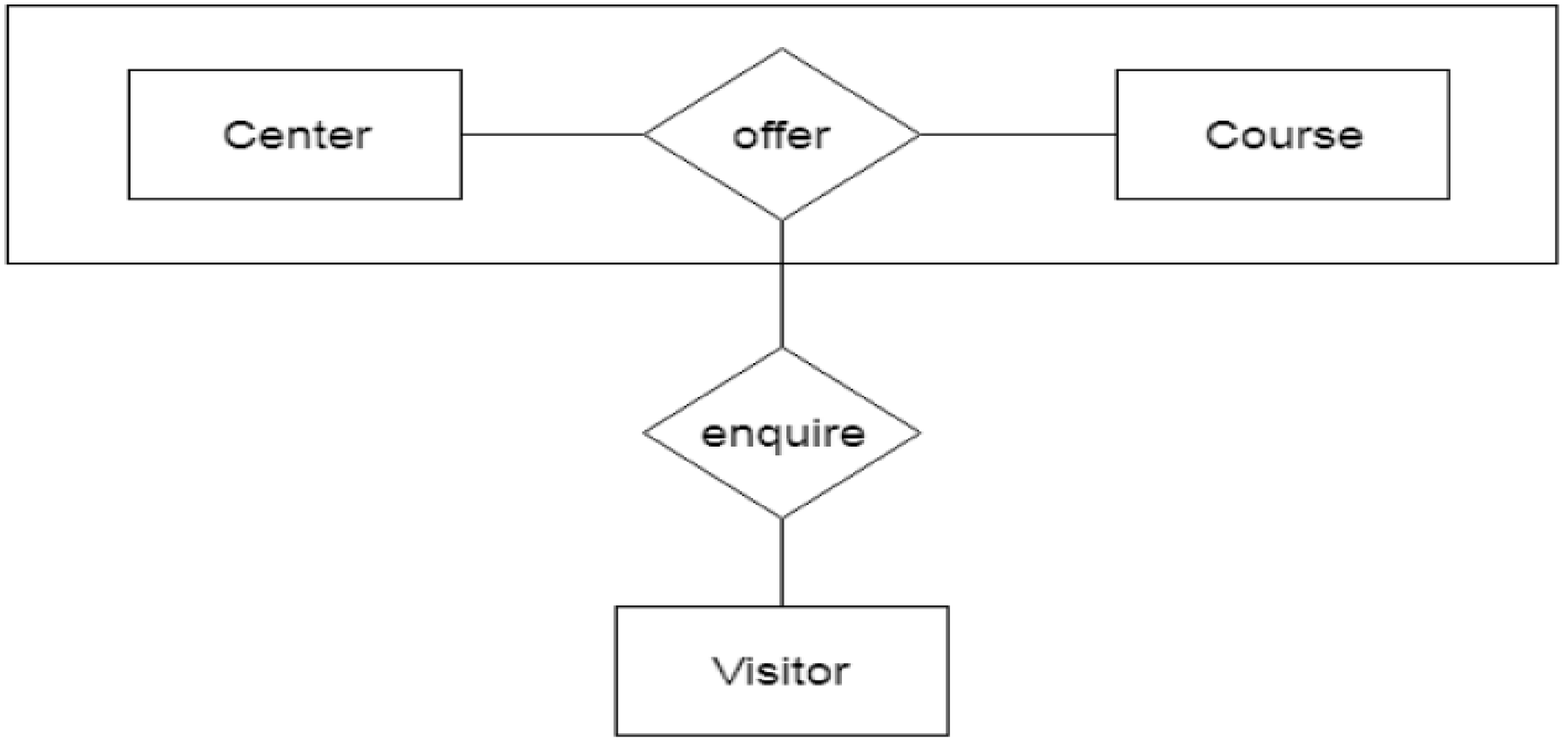
- Relationship of one super or sub class with more than one super class.

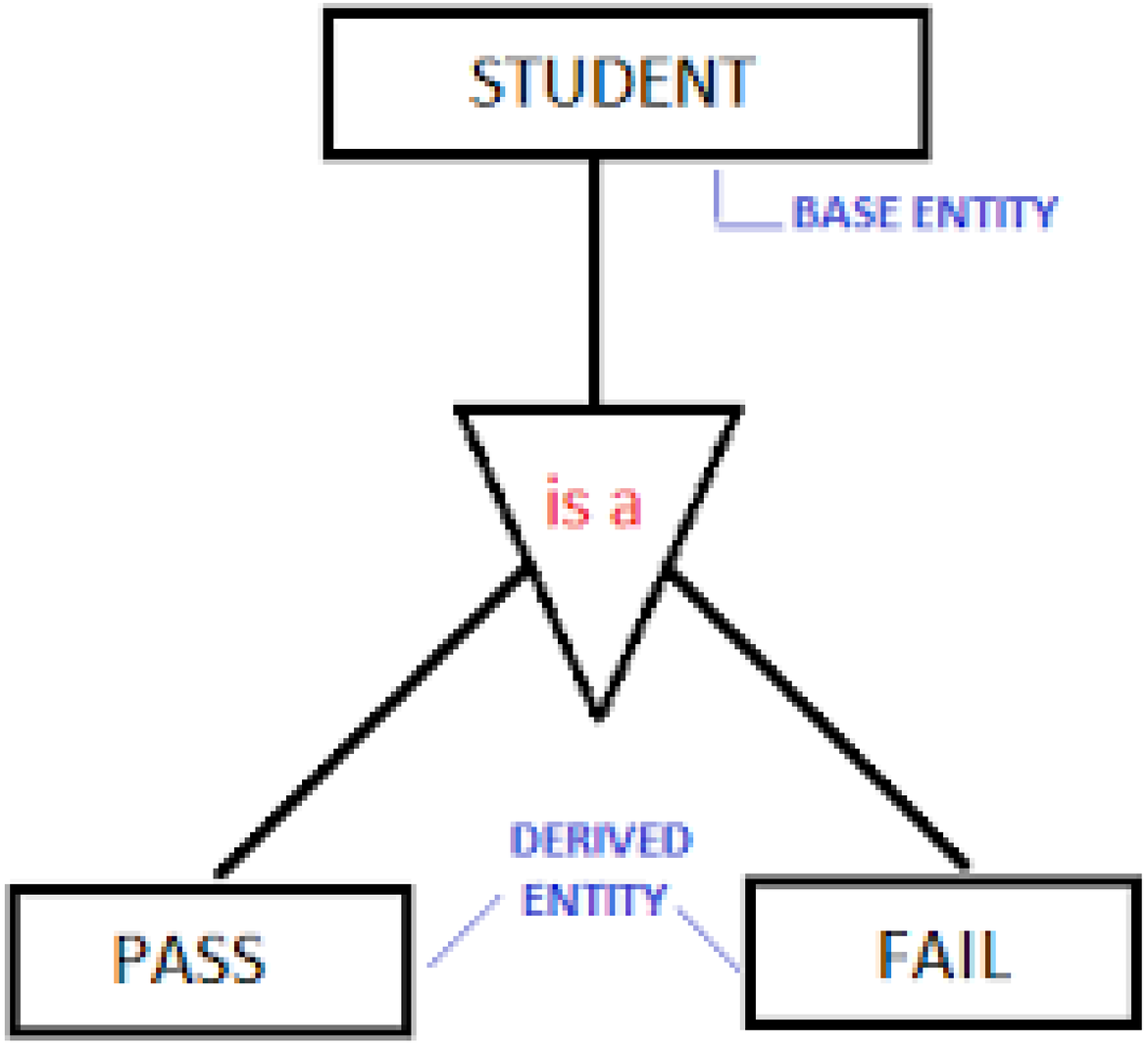


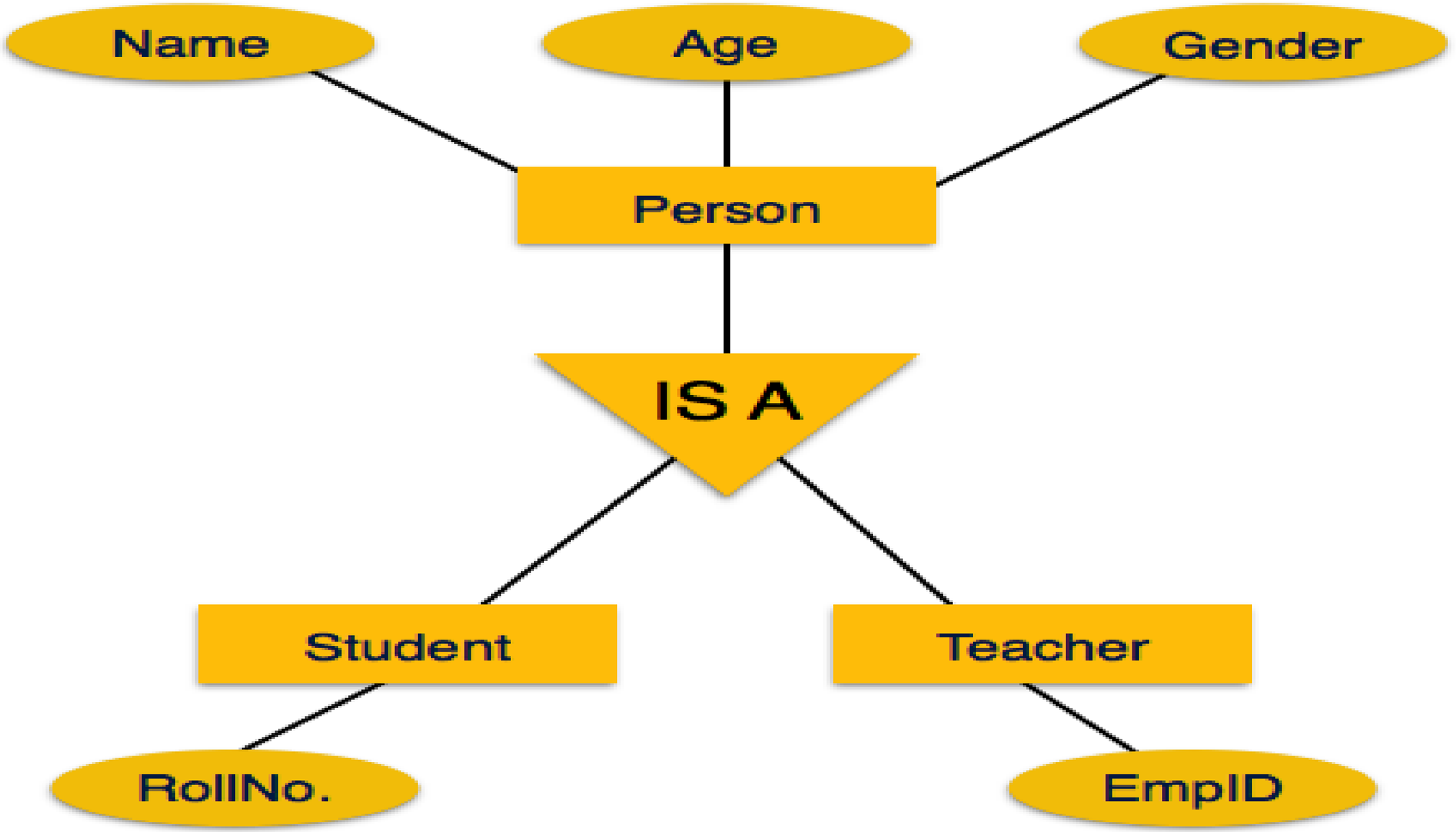
- Owner is the subset of two super class: Vehicle and House.

Aggregation

In aggregation, the relation between two entities is treated as a single entity. In aggregation, relationship with its corresponding entities is aggregated into a higher level entity.

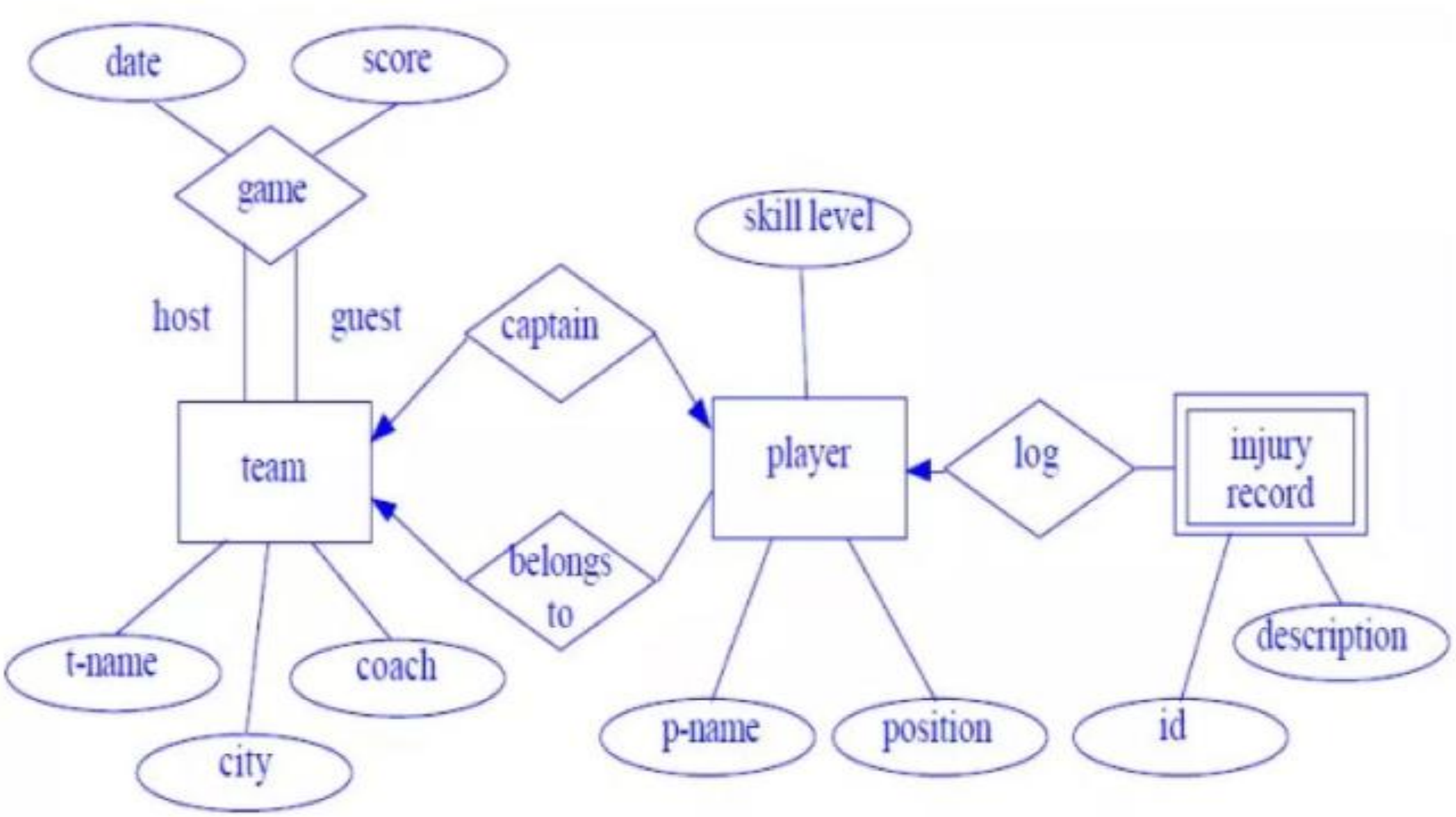



STUDENT
BASE ENTITY
is a
DERIVED
ENTITY
PASS
FAIL


Name
Age
Gender
Person
IS A
Student
Teacher
RollNo.
EmpID

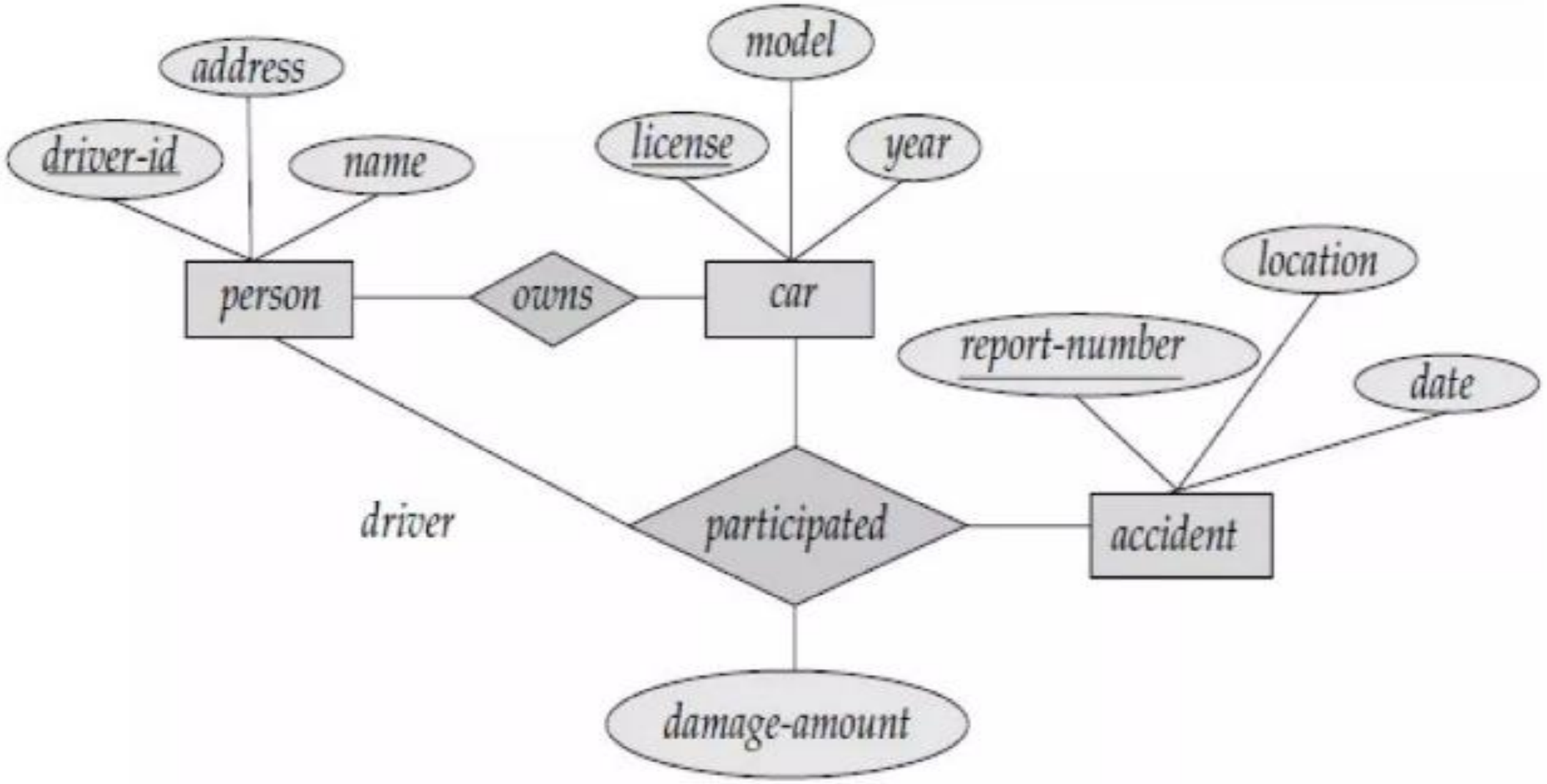
ER Diagram Example

- **Suppose you are given the following requirements for a simple database for the National Hockey League (NHL):**
- **the NHL has many teams,**
- **each team has a name, a city, a coach, a captain, and a set of players,**
- **each player belongs to only one team,**
- **each player has a name, a position (such as left wing or goalie), a skill level, and a set of injury records,**
- **a team captain is also a player,**
- **a game is played between two teams (referred to as host_team and guest_team) and has a date (such as May 11th, 1999) and a score (such as 4 to 2).**
- **Construct a clean and concise ER diagram for the NHL database.**


date
score
game
host
guest
captain
skill level
team
player
log
injury record
belongs to
t-name
coach
city
p-name
position
id
description

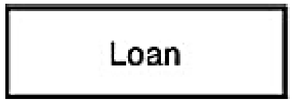
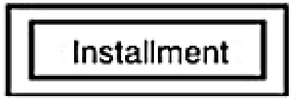
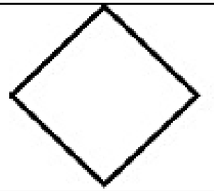
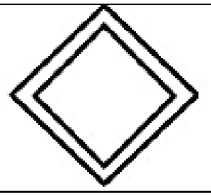
Car insurance tables:

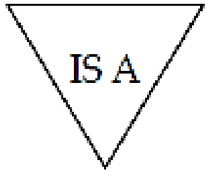
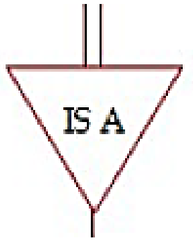
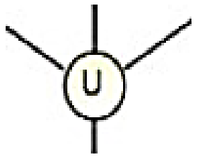
- person (driver-id, name, address)
- car (license, year,model)
- accident (report-number, date, location)
- participated(driver-id, license, report-number, damage-amount)



E-R diagram for a Car-insurance company.

ER diagram symbols

Symbol	Name
	Entity
	Weak Entity
	Relationship
	Weak Relationship
	Attribute / Simple Composite / Single valued Attributes
	Key Attribute
	Multi valued Attributes
	Derived Attributes

	Partial Key Attributes
	Generalization / Specialization
	Total Generalization
	Super class / Sub class
	Union / Category