



[RDBMS- ER model]

[GMRIT _JAVA FSD _2026]

Entity-Relationship (ER) Modeling ->

-> Entity-Relationship (ER) Modeling is a visual representation of data and how it relates to other data in a system.

-> It's like drawing a blueprint for your database before you build it.

-> An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram.

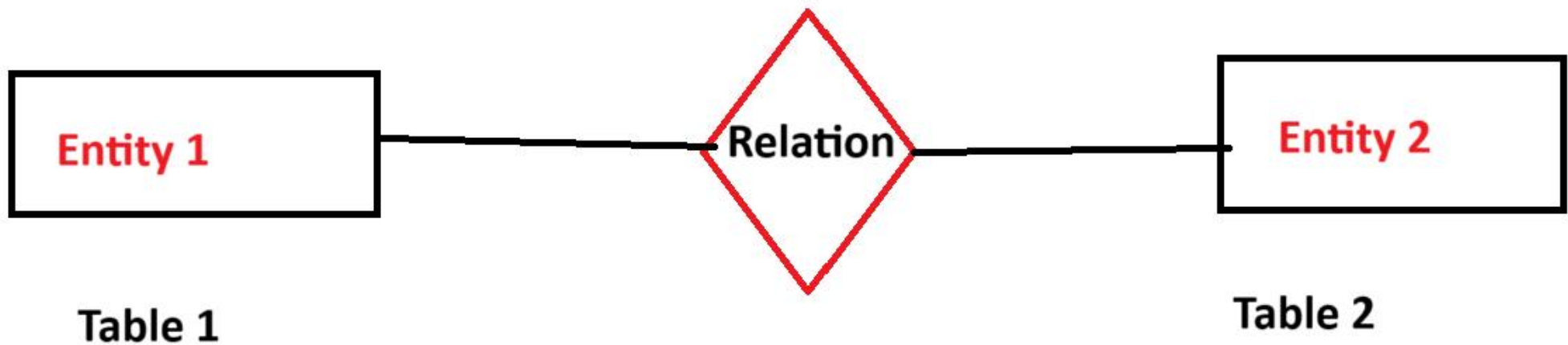
-> which is known as Entity Relationship Diagram (ER Diagram).

What is (ER Diagram)?

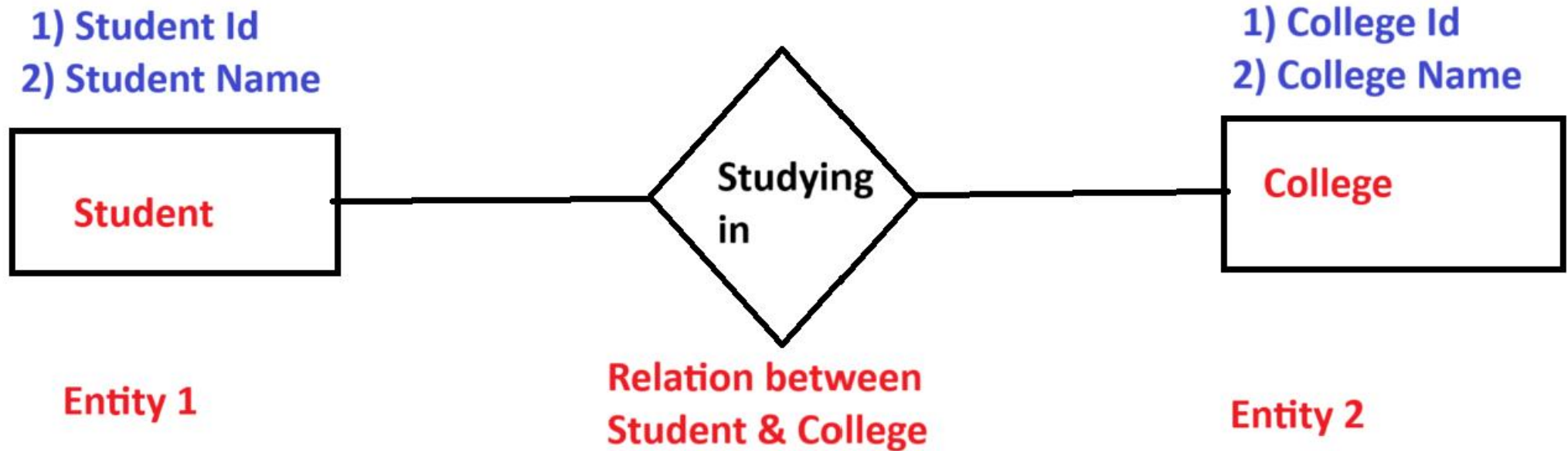
- > It shows the relationship among entity sets .
- > an entity is a table .
- > An entity set is a group of similar entities and these entities can have attributes.
- > ER diagram shows the complete logical structure of a database.

ER Diagram with Example ??

Entity 1 & Entity 2



Student & College



A simple ER Diagram

-> In the above Diagram Student and College both are Entities or tables .

-> What is the Relation b/w them ?

Many to one -> Both Students table and College table are in Many to one relation.

-> Why ? ->

A students can able to study in one college at t time , Not possible to study in Multiple or Many Colleges at a time.

-> Attributes ->

from the above diagram the Student entity or table contains 2 Attribute

- 1) Student Id
- 2) Student name

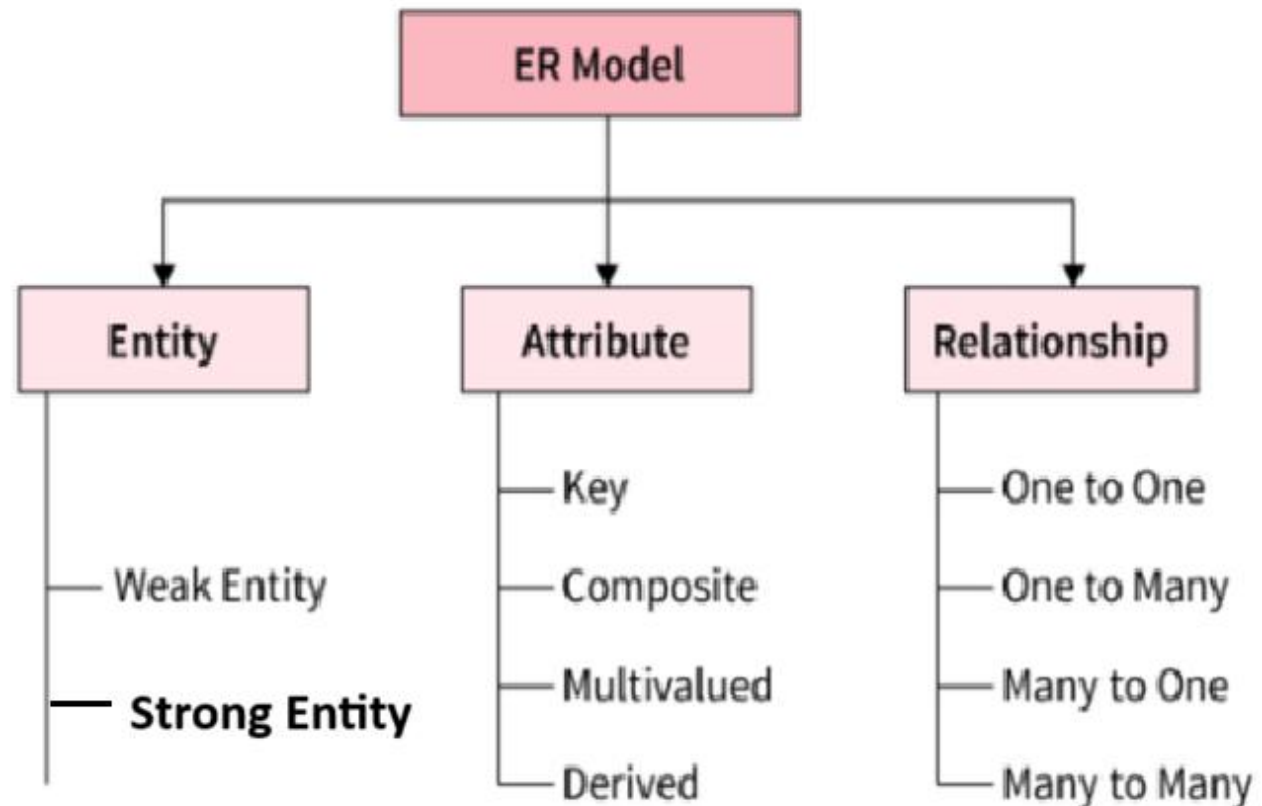
Similarly College entity or table contains 2 attribute

- 1) College Id
- 2) College Name

Component of ER Diagram

The ER diagram consists of three basic concepts:

- Entities
- Attributes
- Relationships



All components Representations of a ER Diagram :



Represents Entity



Represents Attribute



Represents Relationship



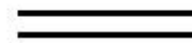
Links Attribute(s) to entity set(s) or
Entity set(s) to Relationship set(s)



Represents Multivalued Attributes



Represents Derived Attributes



Represents Total Participation of Entity



Represents Weak Entity



Represents Weak Relationships



Represents Composite Attributes



Represents Key Attributes / Single Valued
Attributes

- **Rectangle** – It represents entity in the ER Model.
- **Ellipse** – It represents attribute in the ER Model.
- **Diamond** – It represents relationship between entity and attribute.
- **Line** – It links attribute(s) to entity set(s) and entity set(s) to relationship set(s).
- **Doubles Ellipses** – It represents multivalued attributes.
- **Dashed Ellipses** – It denotes derived attributes.
- **Double lines** – It indicates total participation of an entity in a relationship set.
- **Double Rectangle** – It represents weak entity set.
- **Double Diamonds** – It represents weak relationships.
- **Multiple ellipses connected to single ellipse using lines** – It represents composite attribute
- **Ellipse with line inside it** – It represents single values attributes

1. Entity :

- An **entity** is anything in the real world, such as an object, class, person, or place.
- Each **entity** consists of several characteristics or **attributes**
- For example, if a **person** is an **entity**, its **attributes** or characteristics are **age, name, height, weight, occupation, address, hobbies**, and so on.

SIMILARLY the RDBMS ->>>>>>>>>>

- Entity means simply a Table in the data base.

ENTITY

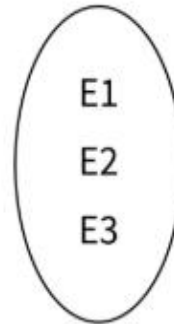
- An entity is represented as *rectangle* in an ER diagram.

Entity set:

- Its is a group of *entities* of similar kinds.
- It can contain *entities* with *attributes* that share similar types of values.

Student

Entity Type



Entity Set

-> *Student1 Entity E1 , Students2 Entity E2, „E3..*

Entity types:

There are 2 types of entities:

1 . Strong Entity

2 . Weak Entity

1 . Strong Entity –

- A strong **entity** is an **entity** type that has a key **attribute**. It doesn't depend on other **entities** in the **schema**.
- A strong **entity** always has a **primary key**,

- and it is represented by a single **rectangle** in the ER diagram.



Strong Entity

Example –

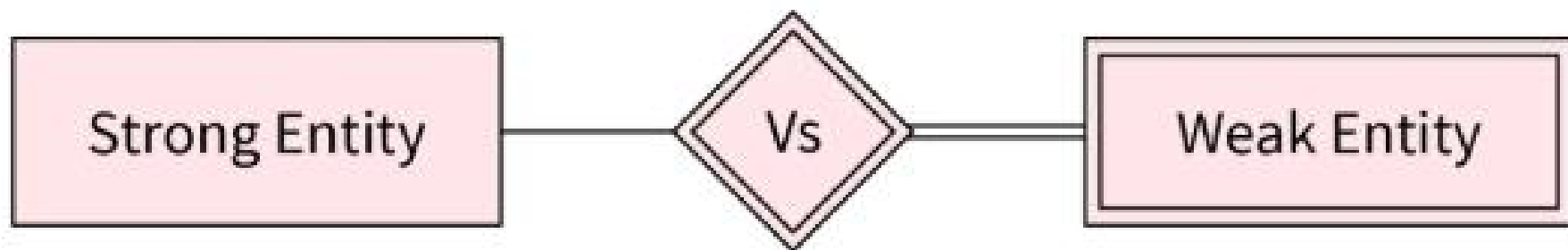
roll_number / Pin_Number identifies each student of the COLLEGE or SCHOOL uniquely and hence, we can say that the student is a strong **entity** type.

2 . Weak Entity –

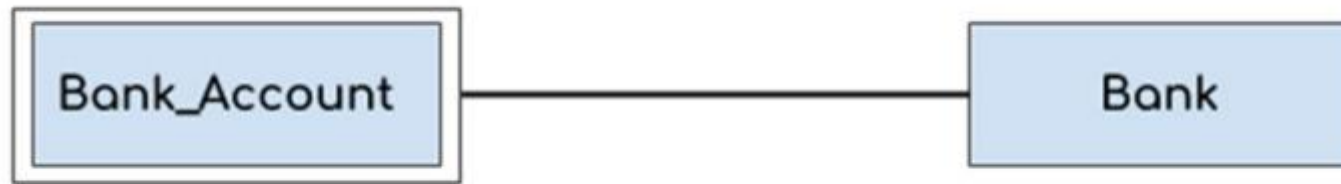
- Weak entity type doesn't have a key attribute and
- Therefore, a foreign key must be used in combination with its attributes .
- They are called Weak entity types because they can't be identified on their own.
- It relies / depends on another powerful / Strong entity for its unique identity.
- A weak entity is represented by a double-outlined rectangle in ER diagrams.



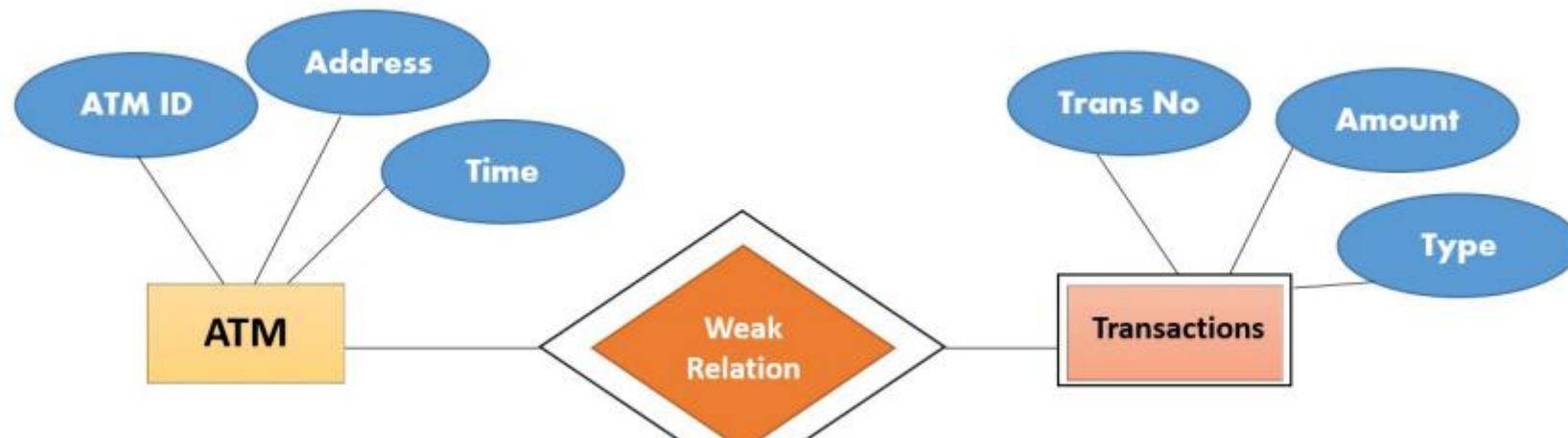
Weak entity Representation



-> Double outer-line Diamond to represent one Strong & one Weak Entity.



Weak entity & Strong Entity Representation



Strong Entity

Weak Entity

II . Attribute :

- An attribute describes the **property /Field** of an entity.
- An attribute is represented as **Oval / Ellipse** in an ER diagram.
- For example, here

id,

Name,

Age, &

Mobile number

are the **attributes** that define the **entity** type Student.

There are 4 types of attributes:

1. Key attribute
2. Composite attribute
3. Multivalued attribute
4. Derived attribute.

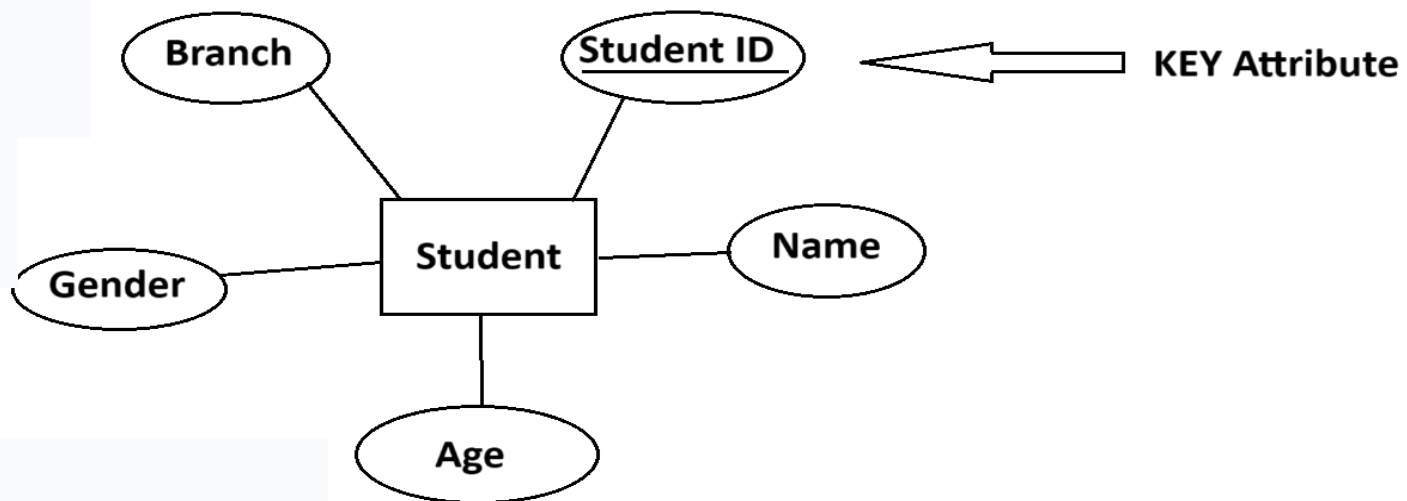
1 . Key attribute :

- A key attribute can **uniquely identify** an entity from an entity set.

For example,

- student **Student ID [Roll Number]** can uniquely identify a student from a set of students.
- Key attribute is represented by oval same as other attributes however the text of key attribute is underlined.

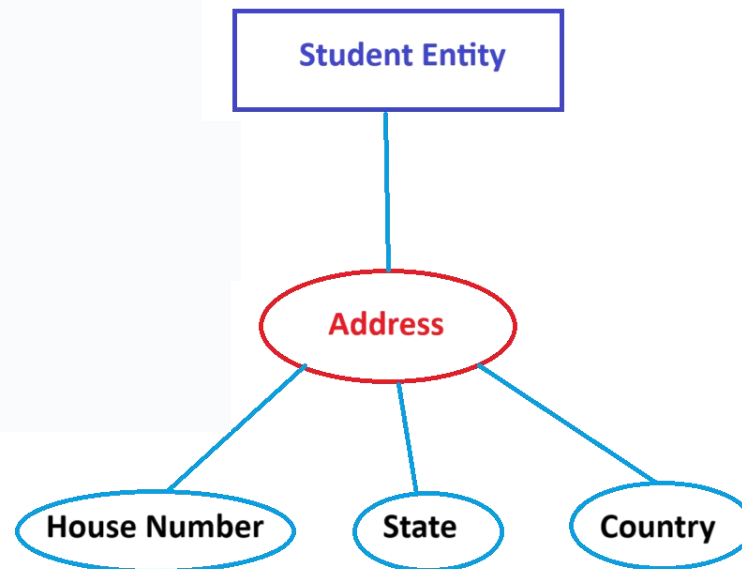
Ex : Student entity with Attributes



2 . Composite attribute:

- An attribute that is a combination of other attributes is known as composite attribute.
- For example,

In student entity, the **student address** is a composite attribute as an address is composed of **other** attributes such as **house Number, state, country**.

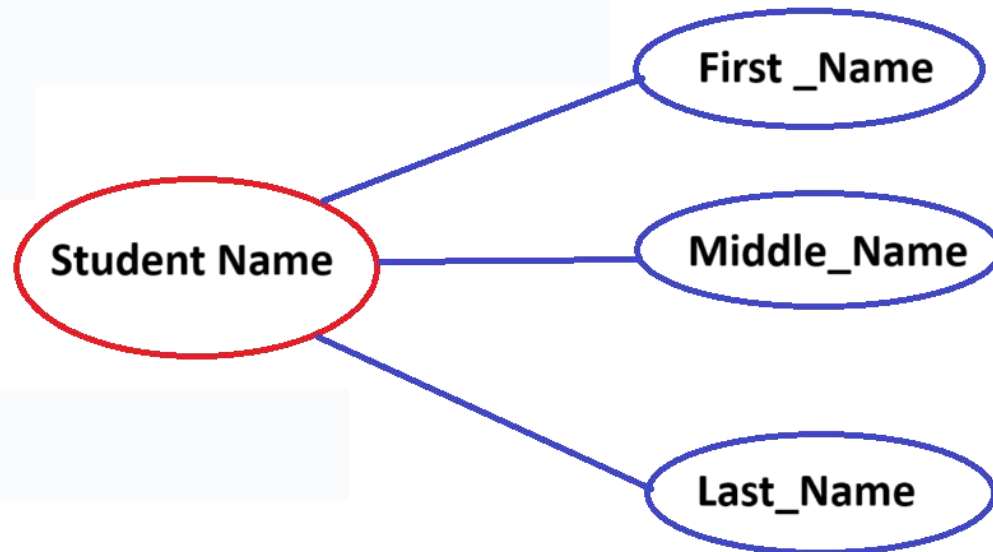


-> Example

- Student name is an Attribute and it contains other attributes as ----> First name ,

Middle Name &

Last Name.



3 . Multivalued attribute:

- An attribute that can **hold multiple values** is known as multivalued attribute.
- It is represented with **double ovals** in an ER Diagram.
- For example –

-> A person can have **more than one phone** numbers so the phone number attribute is multivalued.

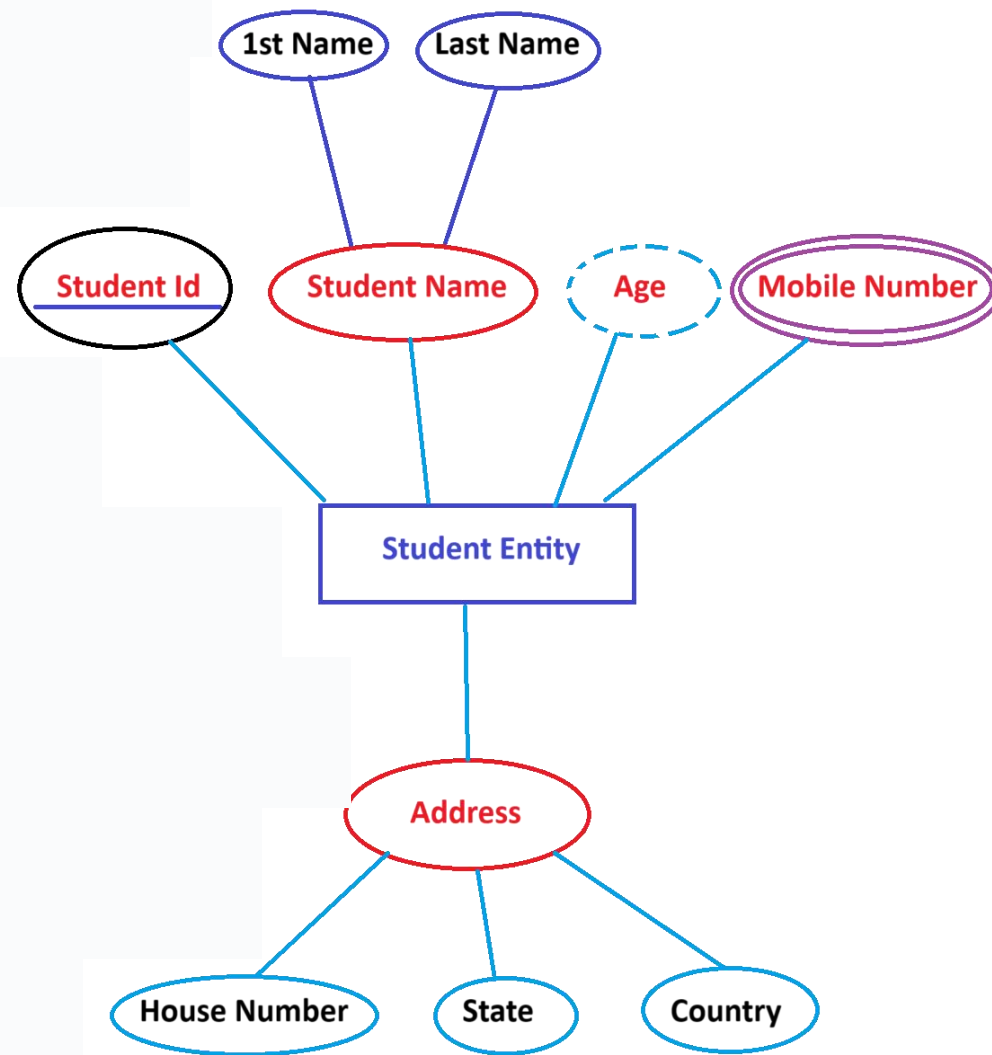


4 . Derived attribute:

- A derived attribute is one whose value is dynamic and *derived from another attribute*.
- It is represented by *dashed oval / ellips* in an ER Diagram.
- For example –

Person age is a derived attribute as it changes over time and can be *derived from another attribute (Date of birth)*





All Entity And Attributes components

III . Relationships

Cardinality:

- Defines the relationship between two entities or entity sets.

SIMPLE WORDS we can say

- Cardinality tells you how many times one thing [table] can be related to another thing [table].

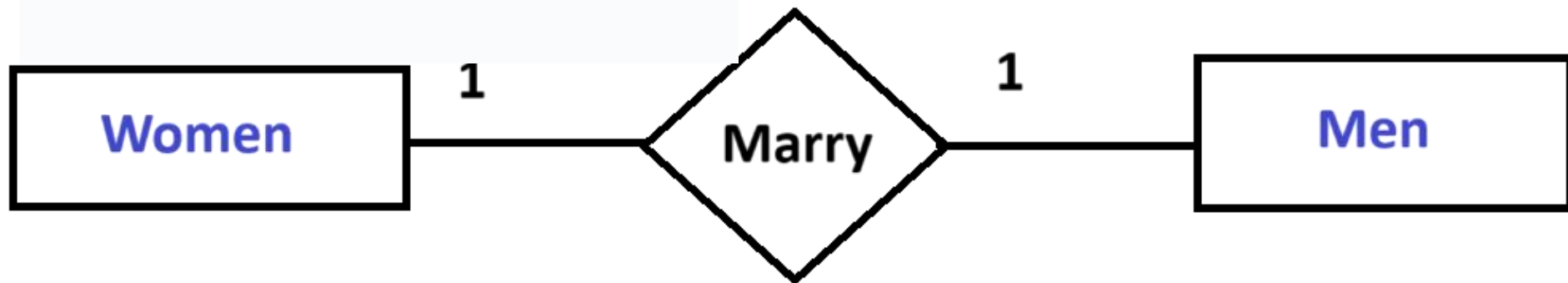
- A relationship is represented by **diamond shape** in ER diagram,
- it shows the relationship among entities.

There are **four types** of cardinal relationships:

1. One to One
2. One to Many
3. Many to One
4. Many to Many

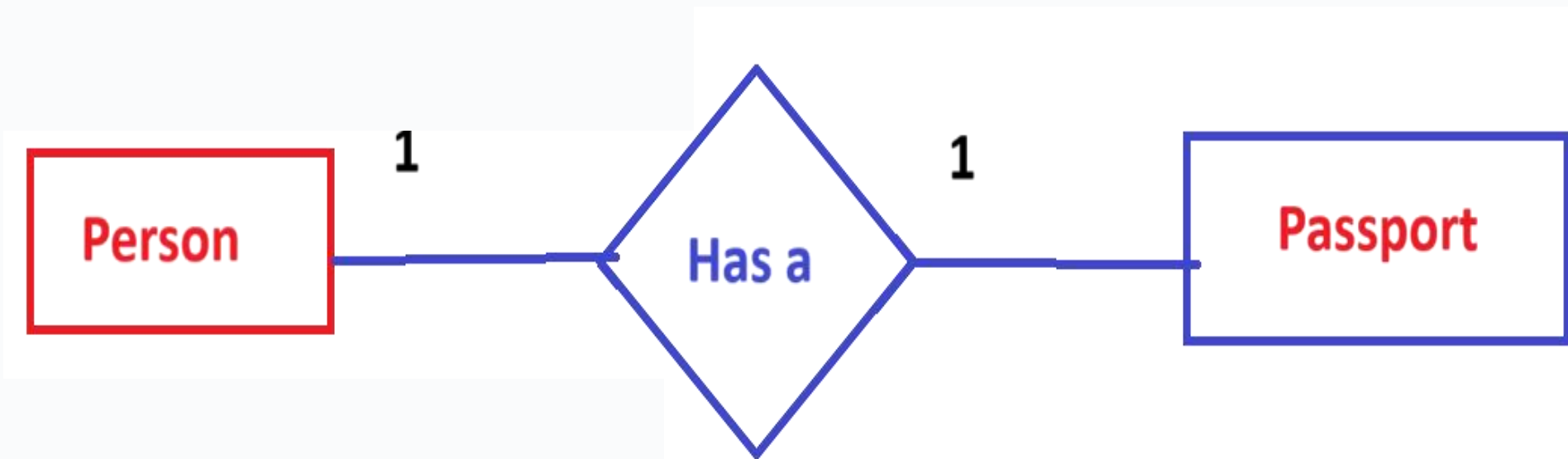
1) One to One Relationship :

- When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship.
- For Example :
One Woman can marry one man



- For example,

a person has only one passport and a passport is given to one person.



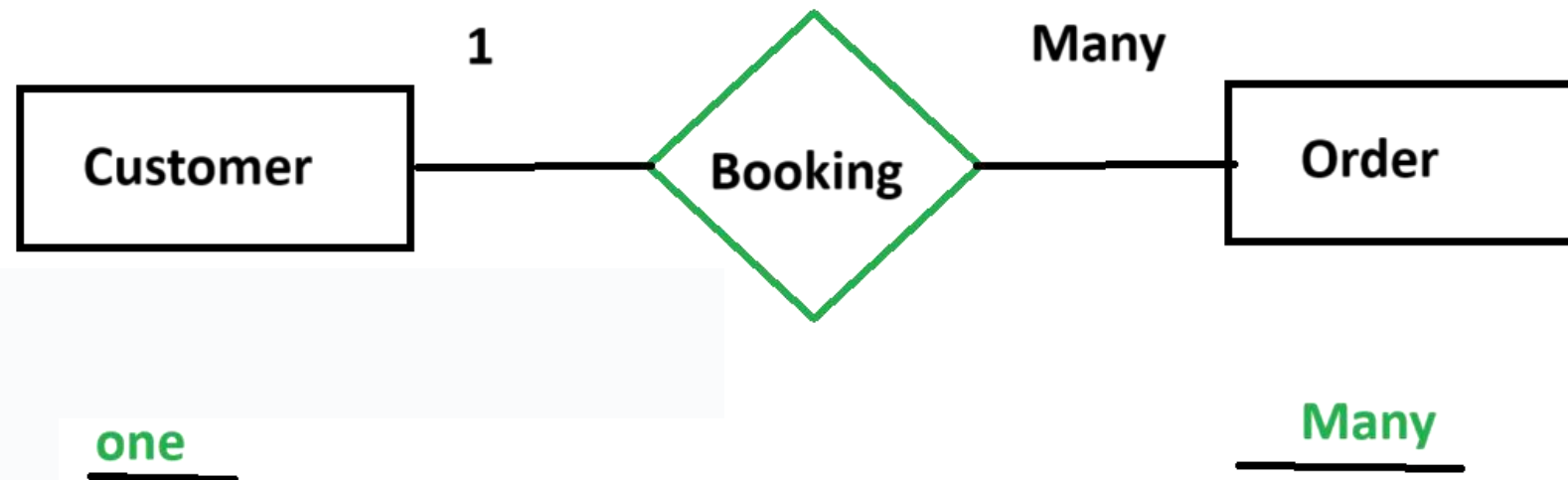
Example :

→ One College & One Principle

→ One Class One CR.

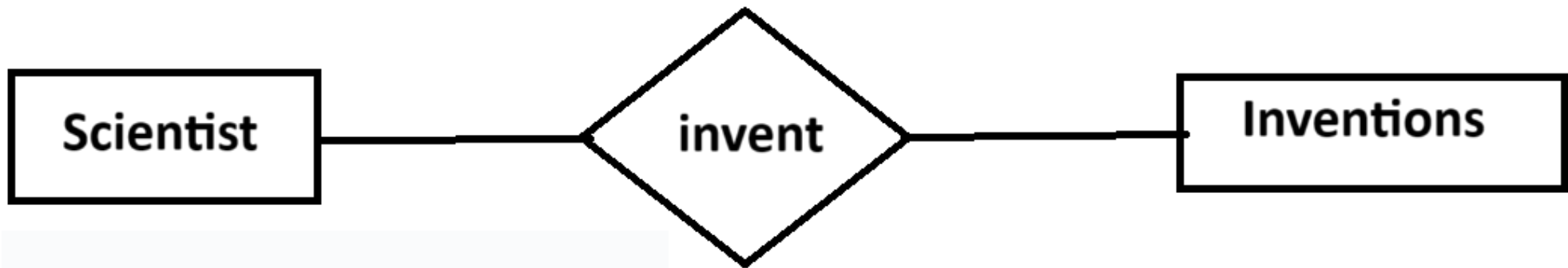
2 . One to Many Relationship

- When a **single instance of an entity** is associated with **more than one instances of another entity** then it is called one to many relationship.
- For example – a customer can place many orders but a order cannot be placed by many customers.



- > A Customer can book or place multiple orders
- > But a Single order cannot be booked by multiple customers.

Ex : Scientist invention



- > A Scientist can invent multiple inventions .
- > But a Single Invention can't be invented by Multiple Scientist .

Examples :

- > one CM for One State People
- > one PM for one Country people
- > one Principle for All college students .

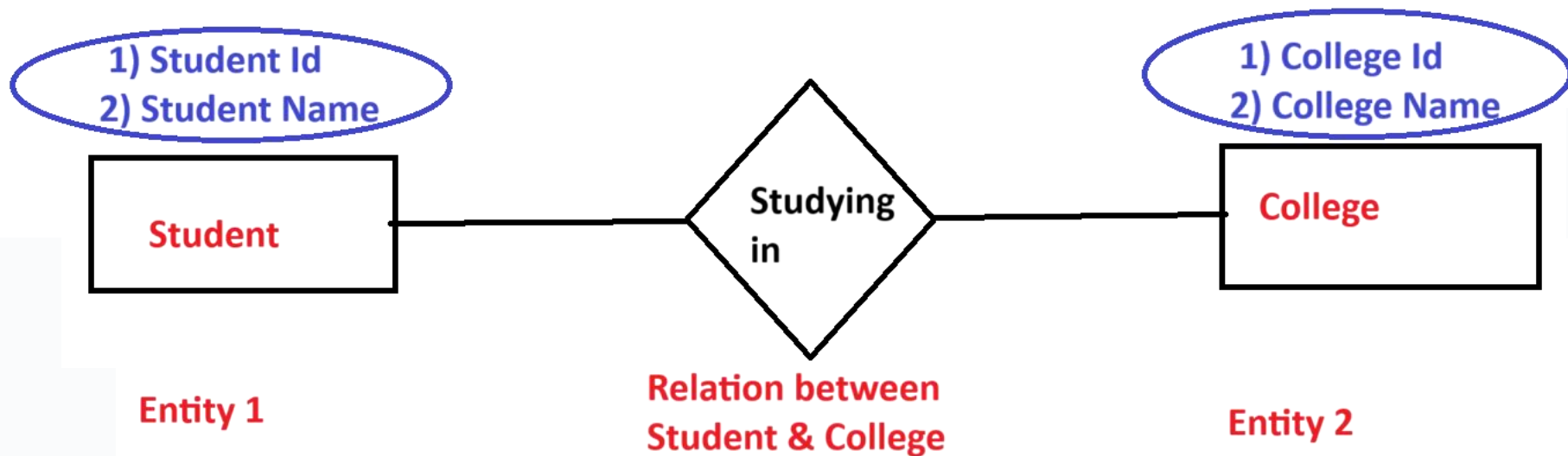
3 . Many to One Relationship :

- When **more than one instances of an entity** is associated with a **single instance of another entity** then it is called many to one relationship.

● For example –

many students can study in a single college but a student cannot study in many colleges at the same time.

Ex:



→ Many Students & Only one college

-> Many Employees for one Organization

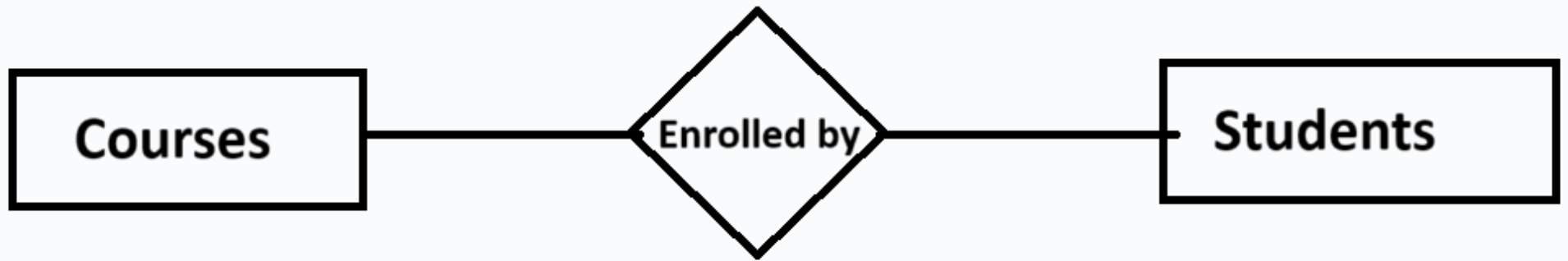
-> Many Voters for to select one CM.

4 . Many to Many Relationship :

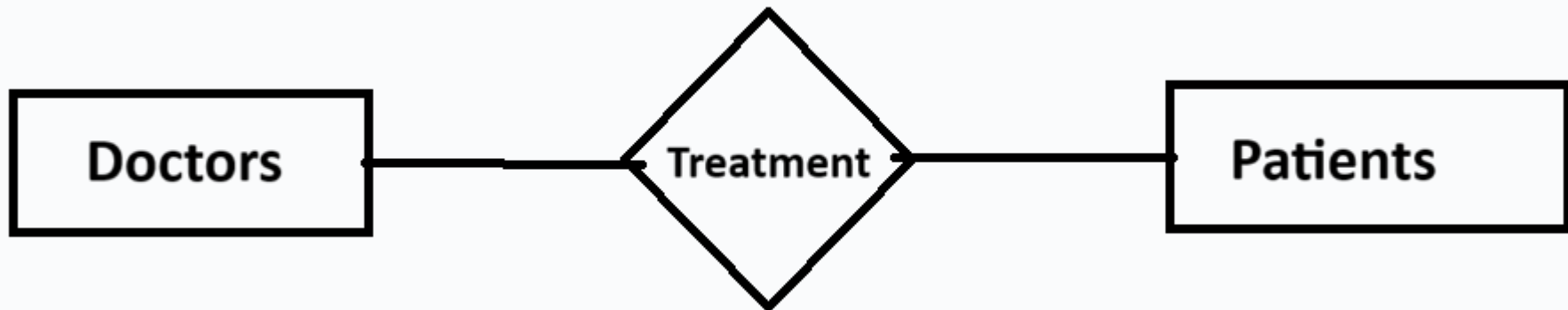
- When **more than one instances of an entity** is associated with **more than one instances of another entity** then it is called many to many relationship.

- For example,

a Students can be assigned to many projects and a project can be assigned to many students.



Ex : Multiple Doctors and Multiple Patients



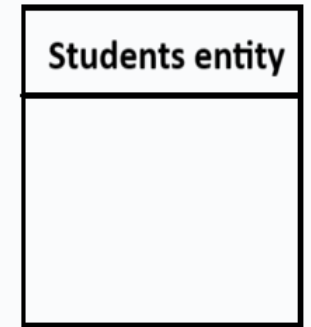
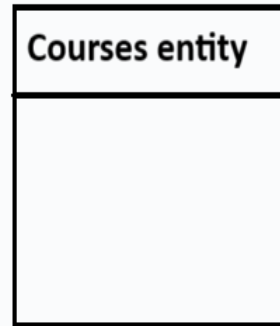
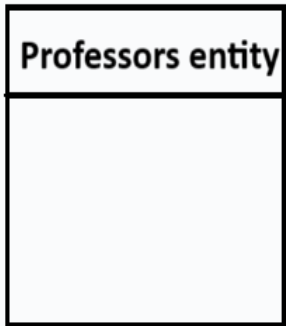
STEPS to Create ER [Entity Relation] - Diagram:

There are 5 steps to create a ER - Diagram in the RDBMS.

- 1) Entity Identification
- 2) Relationship identification
- 3) Cardinality identification
- 4) Attributes identification
- 5) Creating ER - Diagram



a) Entity Identification :

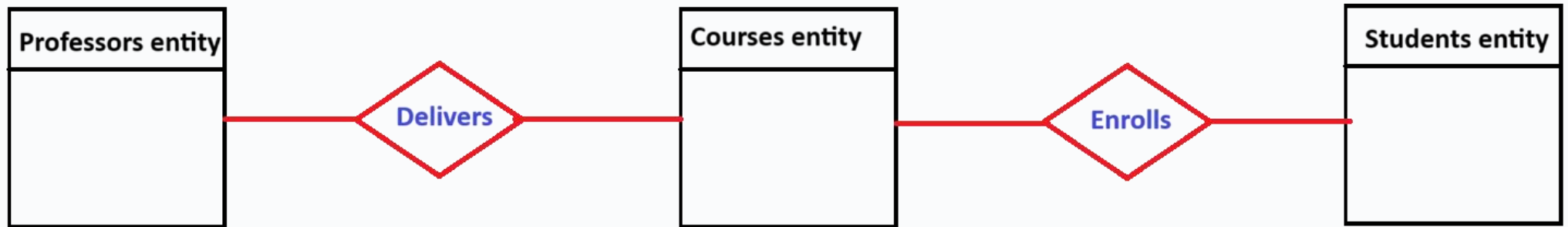


-> Identified 3 tables for the data base design to handle lets for ex : college professors and students data management on some courses .

-> So total 3 entities

-> Professors & Courses & Students entity

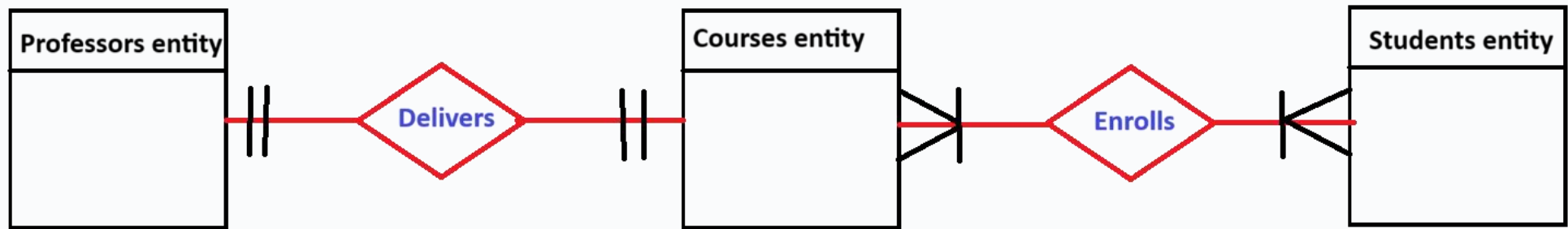
B) Relationship identification :



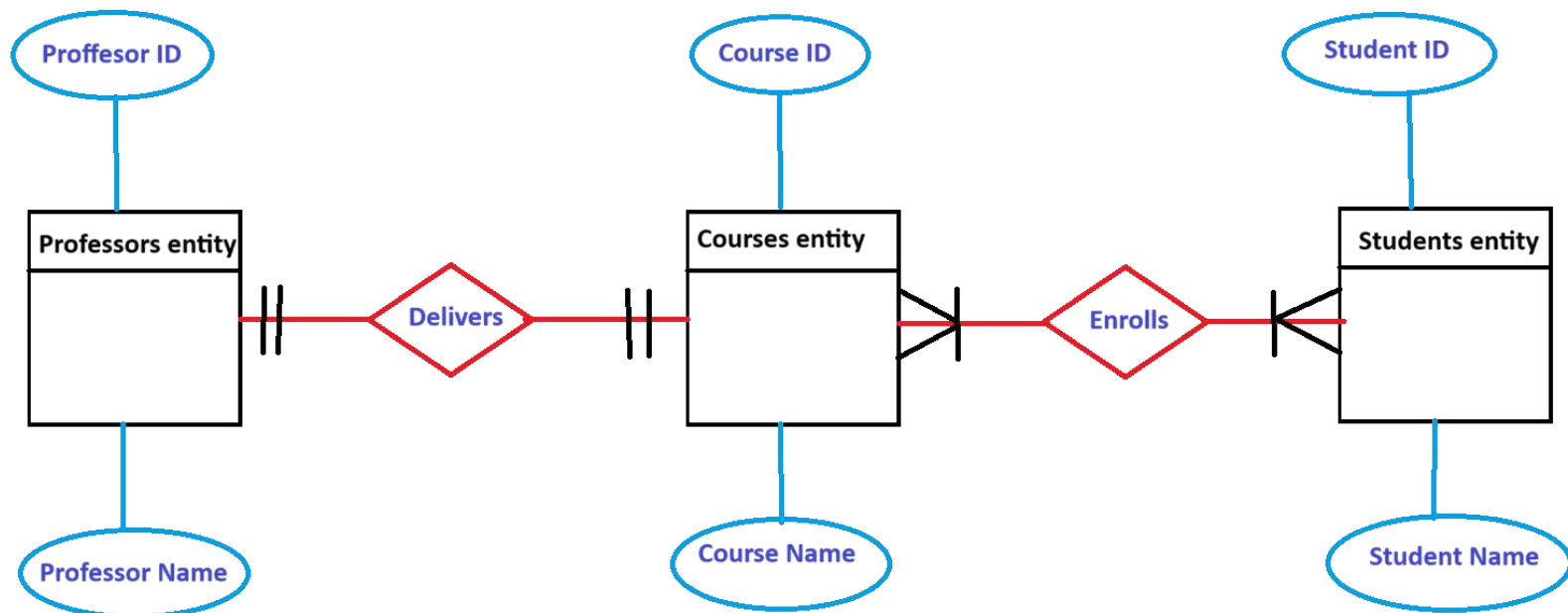
C) Cardinality identification :

-> Professor delivers only one course

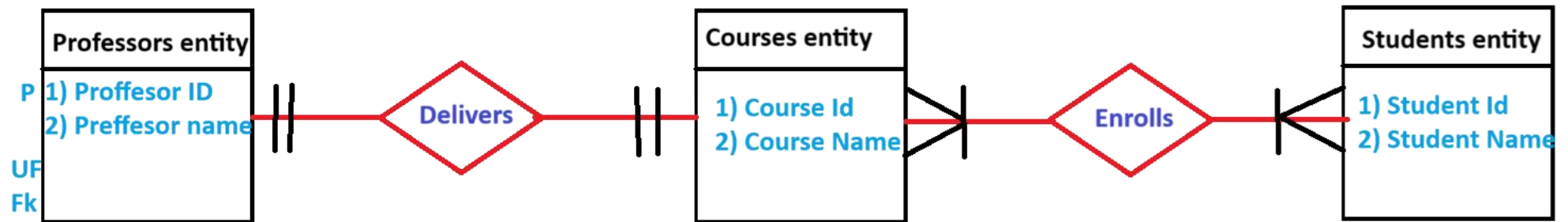
-> But a Students can enroll multiple courses at a time.



D) Attributes identification :



D) Create ER D



Some ER -Diaframs from the bata base

MURARI.DEPARTMENT_DETAILS		
P *	DEPT_ID	VARCHAR2 (5 BYTE)
	DEPT_NAME	VARCHAR2 (20 BYTE)
DEPT_ID_PK (DEPT_ID)		
DEPT_ID_PK (DEPT_ID)		

From the above diagram

Murari . Department_details

-> Murari means -> Data base or Schema NAME

-> Department_details -> Table name in the Murari's schema

-> **P** or **PK** -> Means **Primary key** & each and every table or entity we need to declare the key column with this Primary key based on the requirement.

-> **U** , **UN** -> **Unique**

-> **UF** -> **Unique Field**

-> **F** -> Foreign Key representation of column / Attribute.

-> **F K** -> **Foreign Key.**

Manually created tables and their relations in a ER - MODEL

MURARI.EMPLOYEE_DETAILS		
P *	EMP_ID	VARCHAR2 (5 BYTE)
	EMP_NAME	VARCHAR2 (20 BYTE)
	EMP_SALARY	NUMBER
F	DEPT_ID	VARCHAR2 (5 BYTE)
F	MANAGER_ID	VARCHAR2 (5 BYTE)
EMPLOYEE_DETAILS_PK (EMP_ID)		
EMP_DETAILS_FK (DEPT_ID, MANAGER_ID)		



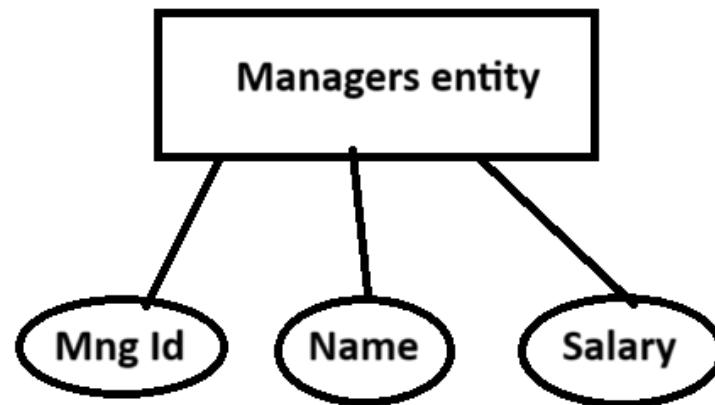
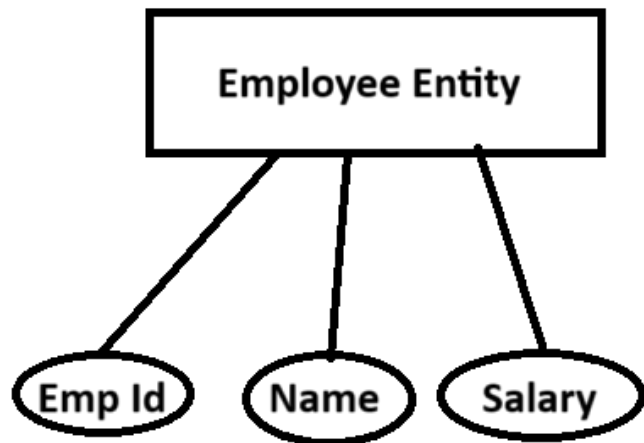
MURARI.MANAGERS_DETAILS		
P *	MANAGER_ID	VARCHAR2 (5 BYTE)
	MANAGER_NAME	VARCHAR2 (20 BYTE)
UF	DEPT_ID	VARCHAR2 (5 BYTE)
MANAGERS_DETAILS_PK (MANAGER_ID)		
MANAGERS_DETAILS_DEPT_ID_MANAGER_ID_UN (DEPT_ID, MANAGER_ID)		
DEPT_ID_FK (DEPT_ID)		

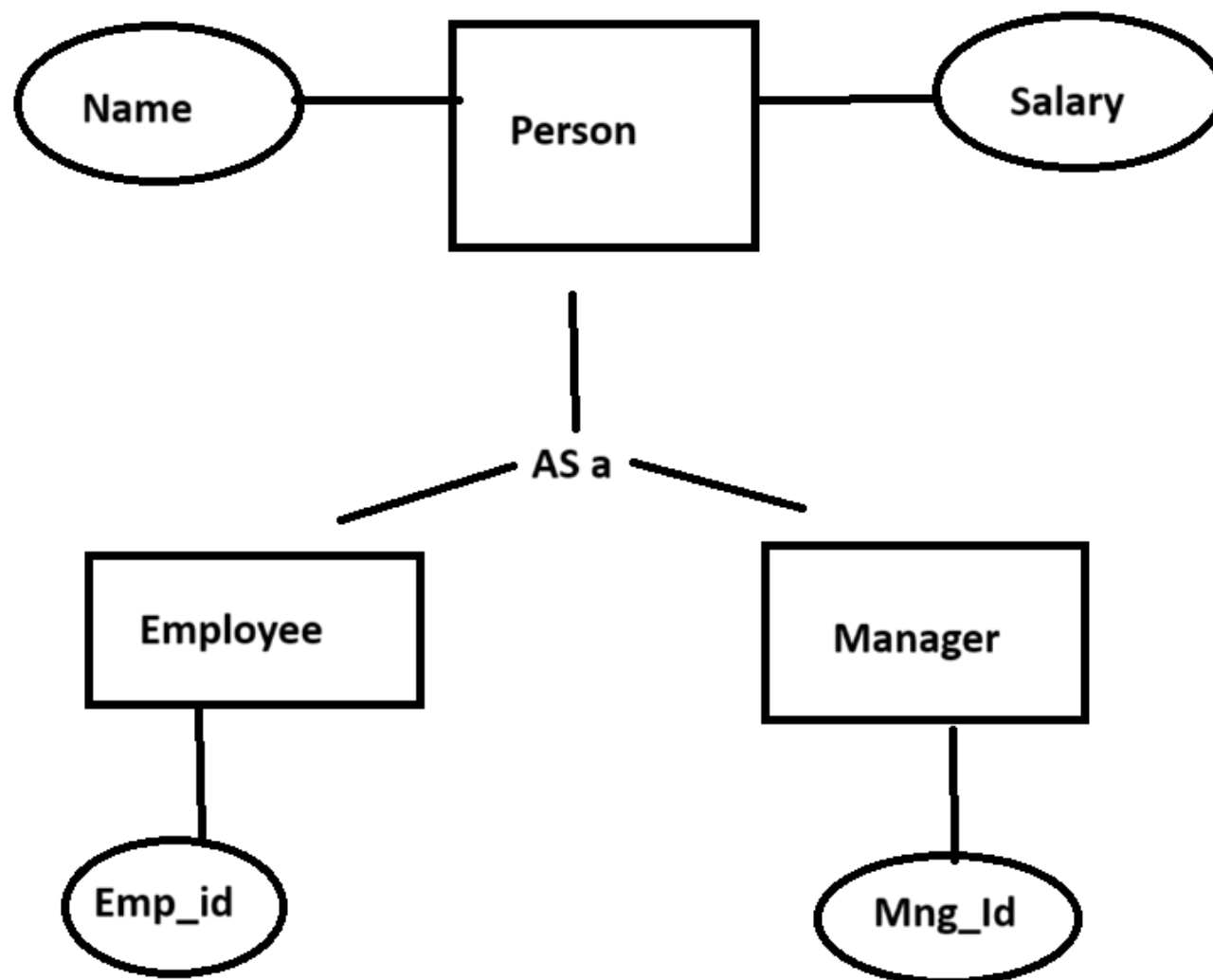


MURARI.DEPARTMENT_DETAILS		
P *	DEPT_ID	VARCHAR2 (5 BYTE)
	DEPT_NAME	VARCHAR2 (20 BYTE)
DEPT_ID_PK (DEPT_ID)		
DEPT_ID_PK (DEPT_ID)		

- > Generalization
- > Specialization
- > Aggregation
- > Normalization

> Generalization :





p

Disaster ins
ins no,
type of, dis
ins amout, 5L

~~Policy Number~~
~~pol_created_date~~
~~Policy_renuual_date~~
~~balance_amt~~

p

Accident ins
ins no,
type of, Acc
ins amout, 3L

~~Policy Number~~
~~pol_created_date~~
~~Policy_renuual_date~~
~~balance_amt~~

p

Theft Ins
ins no,
type of, the
ins amout, 9L

~~Policy Number~~
~~pol_created_date~~
~~Policy_renuual_date~~
~~balance_amt~~

p

Health Insu
ins no,
type of, hel
ins amout, 20L

~~Policy Number~~
~~pol_created_date~~
~~Policy_renuual_dat~~
~~balance_amt~~

Generalization

Common infor_entity

Policy Number
 pol_created_date
 Policy_renuual_date
 balance_amt

The process of common attributes from more than one entity grouping together to form as a new High level entity .

this process called as "Generalization"

-> In other words -> Adding 2 or more entities to form a new Entity .

-> It's a Bottom - Up approach.

-> Must and should it requires more than one entity to form a new entity.

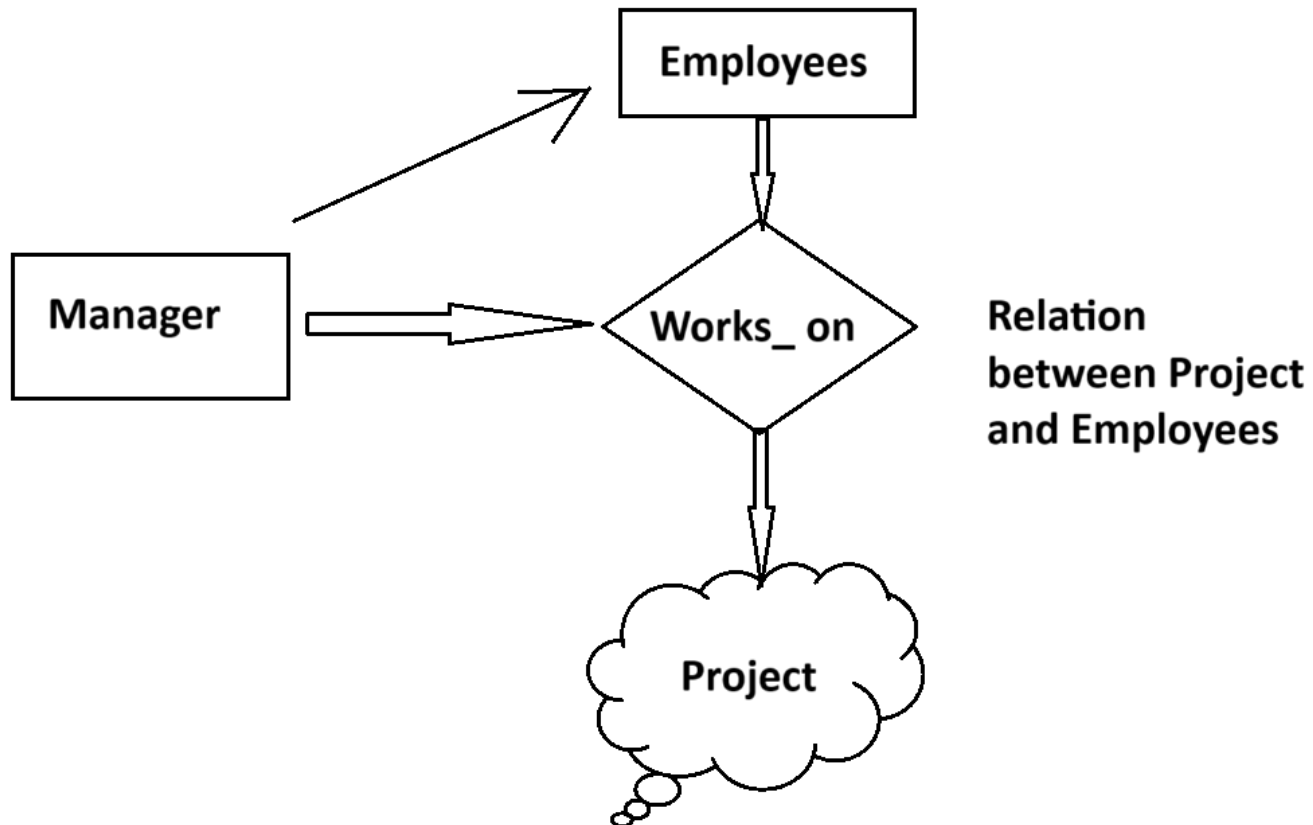
-> Specialization :

- The process of dividing of an entity in to new Sub entity we an call it as “Specialization”.
- Definition: Specialization is the opposite of generalization. It is a top-down approach where a higher-level entity is divided into two or more lower-level entities based on distinct attributes. These lower-level entities are called subclasses.
- Its Top - down approach.

Aggregation :

=====

Aggrigation :



- Definition: Aggregation is a form of abstraction in which a relationship between entities is treated as a higher-level entity itself. This allows you to model relationships between relationships, which the basic ER model does not support.
- Example: If you have a relationship "Works_On" between Employee, Branch, and Manager entities, you can aggregate "Works_On" as a higher-level entity and relate it with other entities as needed.