**UNIT II**

**DATABASE DESIGN**

A well-designed database shall:

- Eliminate Data Redundancy: the same piece of data shall not be stored in more than one place. This is because duplicate data not only waste storage spaces but also easily lead to inconsistencies.
- Ensure Data Integrity and Accuracy

**Entity-Relationship Data Model**

- Classical, popular conceptual data model
- First introduced (mid 70's) as a (relatively minor) improvement to the relational model: pictorial diagrams are easier to read than relational database schemas
- Then evolved as a popular model for the first conceptual representation of data structures in the process of database design

**ER Model: Entity and Entity Set**

- Considering the above example, Student is an entity, Teacher is an entity, similarly, Class, Subject etc are also entities.
- An Entity is generally a real-world object which has characteristics and holds **relationships** in a DBMS.
- If a Student is an Entity, then the complete dataset of all the students will be the **Entity Set**

**ER Model: Attributes**

If a Student is an Entity, then student's roll no., student's name, student's age, student's gender etc will be its attributes.

An attribute can be of many types, here are different types of attributes defined in ER database model:

**1. Simple attribute**: The attributes with values that are atomic and cannot be broken down further are simple attributes. For example, student's age.

**2. Composite attribute**: A composite attribute is made up of more than one simple attribute. For example, student's address will contain, house no., street name, pincode etc.

**3. Derived attribute**: These are the attributes which are not present in the whole database management system, but are derived using other attributes. For example, average age of students in a class.

**4. Single-valued attribute**: As the name suggests, they have a single value.

**5. Multi-valued attribute**: And, they can have multiple values.

**ER Model: Relationships**

- When an Entity is related to another Entity, they are said to have a relationship. For example, A Class Entity is related to Student entity, because students study in classes, hence this is a relationship.
- Depending upon the number of entities involved, a degree is assigned to relationships.
- For example, if 2 entities are involved, it is said to be Binary relationship, if 3 entities are involved, it is said to be Ternary relationship, and so on.

**Working with ER Diagrams**

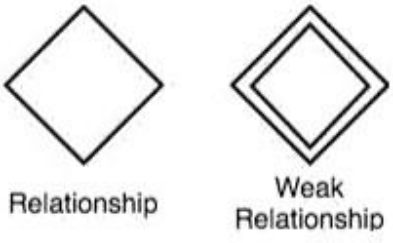
ER Diagram is a visual representation of data that describes how data is related to each other. In ER Model, we disintegrate data into entities, attributes and setup relationships between entities, all this can be represented visually using the ER diagram.

**Components of ER Diagram**

Entitiy, Attributes, Relationships etc form the components of ER Diagram and there are defined symbols and shapes to represent each one of them. Let's see how we can represent these in our ER Diagram. Entity Simple rectangular box represents an Entity.

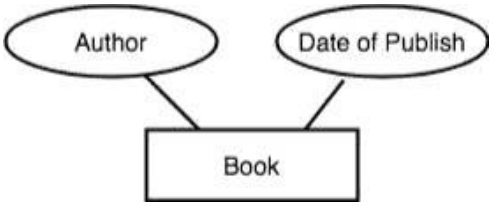

**Relationships between Entities - Weak and Strong**

Rhombus is used to setup relationships between two or more entities.




**Attributes for any Entity**

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




**Weak Entity** A weak Entity is represented using double rectangular boxes. It is generally connected to another entity.


**Key Attribute** for any Entity To represent a Key attribute, the attribute name inside the Ellipse is underlined.


**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**

Double Ellipse, one inside another, represents the attribute which can have multiple values.


**Composite Attribute for any Entity** A composite attribute is the attribute, which also has Attributes

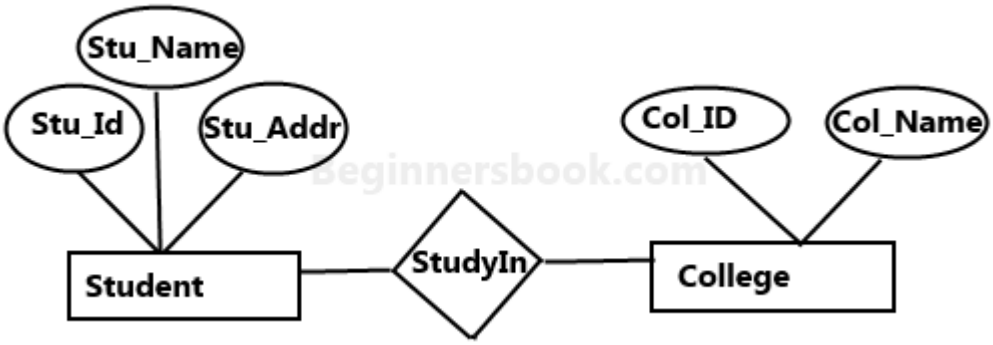

**ER Diagram**:

**Entity** : An Entity can be any object, place, person or class. In ER Diagram, an entity is represented using rectangles. Consider an example of an Organisation- Employee, Manager, Department, Product and many more can be taken as entities in an Organisation.


The yellow rhombus in between represents a relationship.

**ER Diagram**: **Key Attribute**

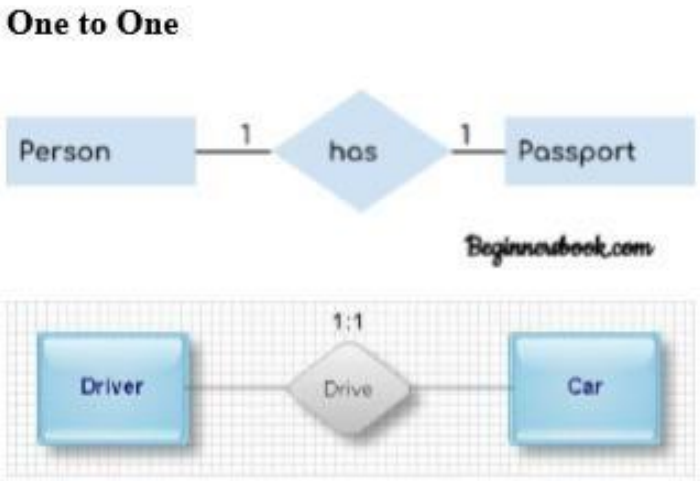
Key attribute represents the main characteristic of an Entity. It is used to represent a Primary key. Ellipse with the text underlined, represents Key Attribute.




**ER Diagram: Binary Relationship**

Binary Relationship means relation between two Entities. This is further divided into three types.

**One to One Relationship** : This type of relationship is rarely seen in real world.

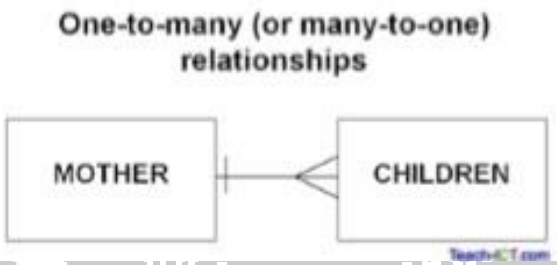



In the above examples, one person can have only one passport,also one passport can belongs to only one person.

One driver can be the driver of one car at a time. A car can have only one driver at a time.

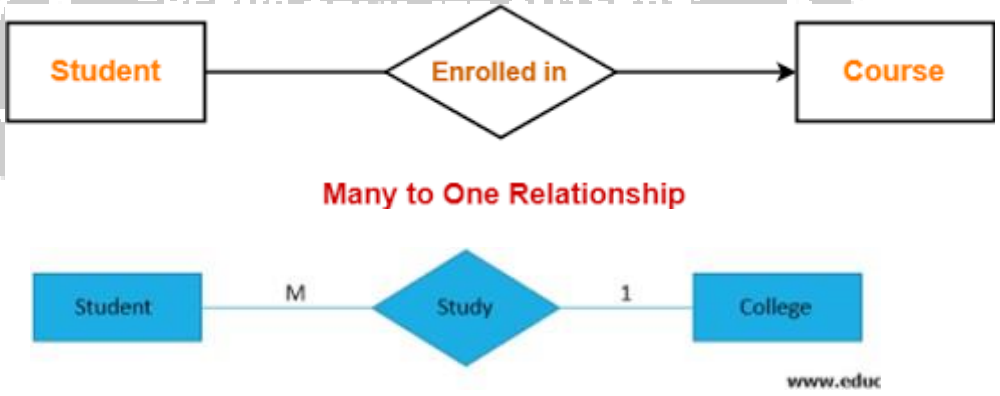
**One to Many Relationship**

The below example showcases this relationship, which means that 1 student can opt for many courses, but a course can only have 1 student. Sounds weird! This is how it is.




**Many to One Relationship**

It reflects business rule that many entities can be associated with just one entity. For example, Student enrolls for only one Course but a Course can have many Students.

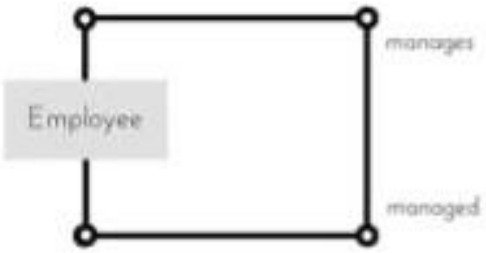



**Many to Many Relationship**

**Many to many**


The above diagram represents that one student can enroll for more than one courses. And a course can have more than 1 student enrolled in it.

**ER Diagram: Recursive Relationship** When an Entity is related with itself it is known as Recursive Relationship.

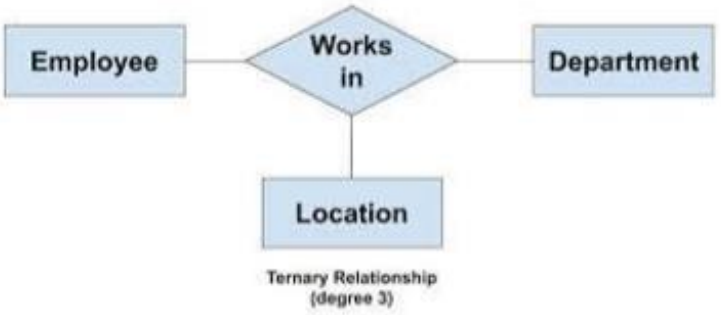


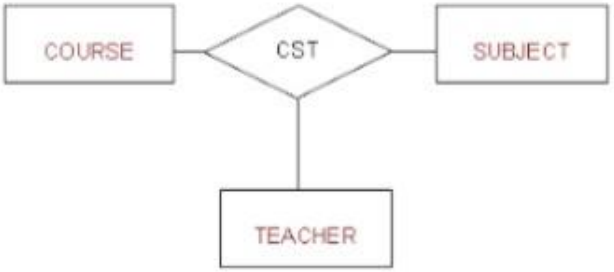

**ER Diagram: Ternary Relationship**

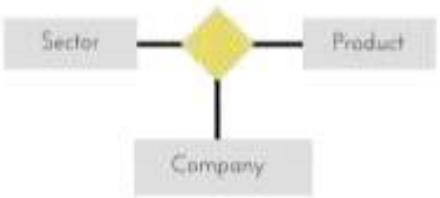
Relationship of degree three is called Ternary relationship.

A Ternary relationship involves three entities. In such relationships we always consider two entites together and then look upon the third.

**Ternary relationship**






- The above relationship involves 3 entities.
- Company operates in Sector, producing some Products.

For example, in the diagram above, we have three related entities, Company, Product and Sector. To understand the relationship better or to define rules around the model, we should relate two entities and then derive the third one. A Company produces many Products/ each product is produced by exactly one company. A Company operates in only one Sector / each sector has many companies operating in it.

Considering the above two rules or relationships, we see that although the complete relationship involves three entities, but we are looking at two entities at a time.