

Entity-Relationship model

Entity-Relationship model

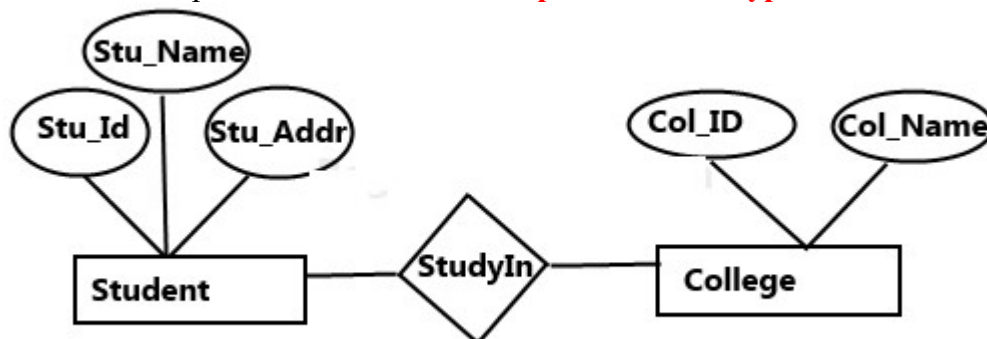
- ER Model represents the **overall logical structure of database**
- An **Entity-relationship model (ER model)** describes the **structure of a database** with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**.
- An ER model is a design or blueprint of a database that can later be implemented as a database.
- The main components of E-R model are:
 - a) entity set
 - b) relationship set
 - c) Attributes

a) Entity Set

- An entity is a **“thing” or “object”** in the real world.
- Entity is represented using “Rectangle” symbol in ER diagrams
- Example
 - Person → customer, employee, teacher
 - Book
 - Loan etc
- Entity set is a **set of entities of same type** that share the same properties / attributes.
- Set of all persons who are customers at a given bank. (**customer & loan**) are entity sets.

b) Relationship set

- Relationship is an **association among several entities**.
- Example: “Tamil” is customer of loan L15. Here customer is relationship.
Tamil ----- **customer** ----- loan
- Relationship set is a **set of relationships of the same type**.

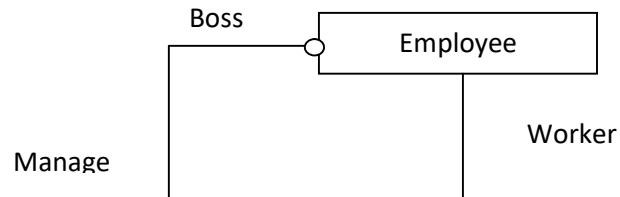


- Here,
 - Entity → Student college
 - Relationship → “StudyIn”
 - Attributes → Stu_name, Stu_Id, Stu_Addr, Col_ID, Col_Name

Types of Relationship

i) Unary relationship

- Association is maintained within **a single Entity**
- Example



- Employee is a single entity includes Boss & Worker.
- Here, Boss Manages Worker

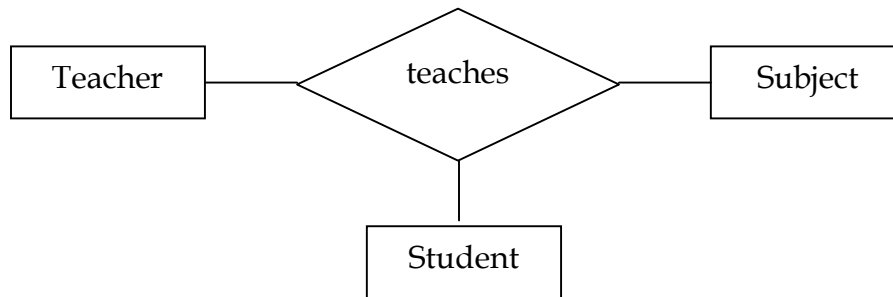
ii) Binary Relationship

- Two entities are associated with a relationship
- Example



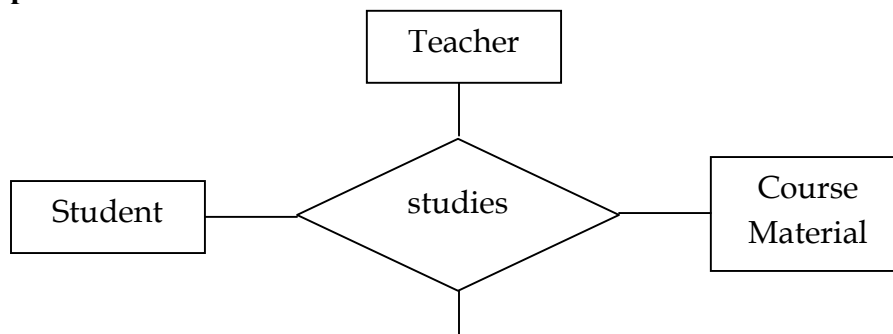
iii) Ternary Relationship

- Three entities are associated with a relationship
- Example



iv) Quaternary Relationship

- Four entities are associated with a relationship
- Example**



c) Attributes

- An entity is represented by set of attributes.
- Attributes are descriptive properties possessed by each entity.
- Attributes are the **properties which define the entity type**.
- For example, Roll_No, Name, DOB, Age, Address, Mobile_No are the attributes which defines entity type Student.
- In ER diagram, attribute is represented by an oval.

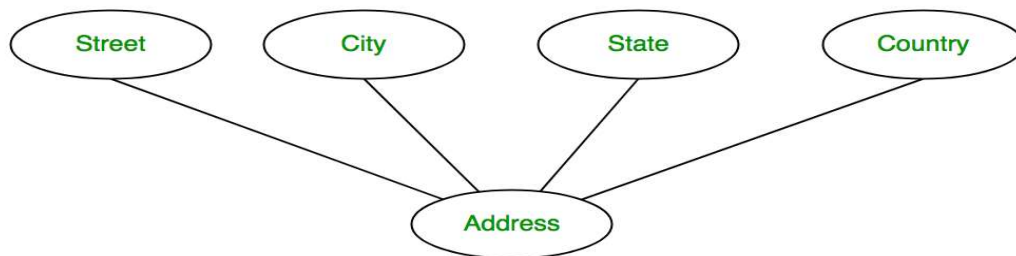


Types of Attributes

- **Key Attribute**
 - The attribute which **uniquely identifies each entity** in the entity set is called key attribute.
 - For example, Roll_No will be unique for each student.
 - In ER diagram, key attribute is represented by an oval with underlying lines.



- **Simple Attributes**
 - It is an attribute composed of single component.
 - Cannot be further sub divided
 - Example: rollno, mark etc.,
- **Composite Attribute**
 - Composite attributes composed of multiple components
 - An attribute **composed of many other attribute** is called as composite attribute.
 - For example, Address attribute of student Entity type consists of
 - Street, City, State, and Country.
 - Name contain → First name, middle name, last name.
 - In ER diagram, composite attribute is represented by an oval comprising of ovals.



- **Single-valued attributes**
 - Holds only one single value.
 - It is also called as Atomic attributes.
 - Example
 - Cust_id → 101
 - Roll_no → 5
 - Door_no → 1

- **Multivalued Attribute**

- An attribute consisting **more than one value** for a given entity.
- For example, Phone_No (can be more than one for a given student).
Hobby attributes (may hold multiple value like read, write, painting)
- In ER diagram, multivalued attribute is represented by double oval.

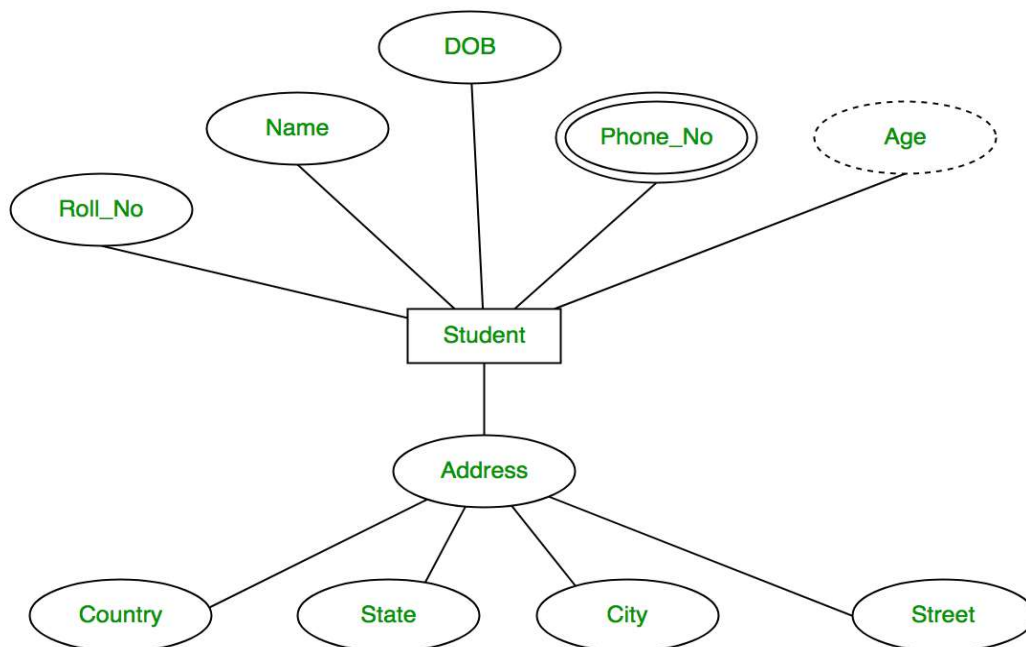


- **Derived Attribute**

- An attribute which can be **derived from other attributes** of the entity type is known as derived attribute.
- e.g.; Age (can be derived from DOB).
- In ER diagram, derived attribute is represented by dashed oval.



The complete entity type **Student** with its attributes can be represented as:



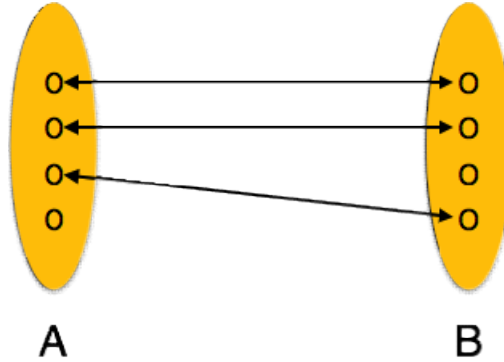
Constraints

- An ER enterprise may define certain constraints to which the content of database system must conform.

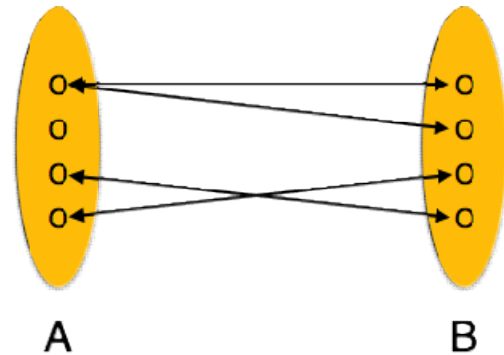
i) Mapping Cardinalities

- **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.

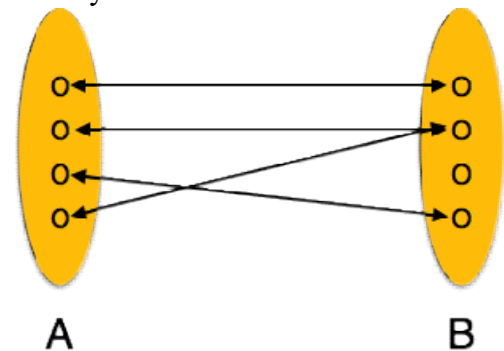
- **One-to-one** – One entity from entity set A can be associated with at most one entity of entity set B and vice versa.



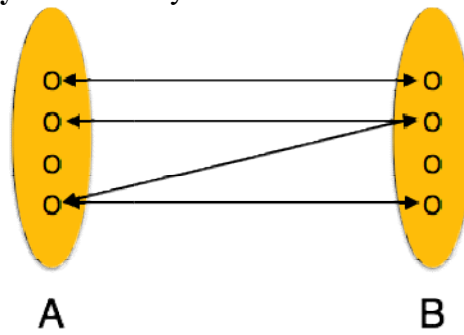
- **One-to-many** – One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.



- **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.

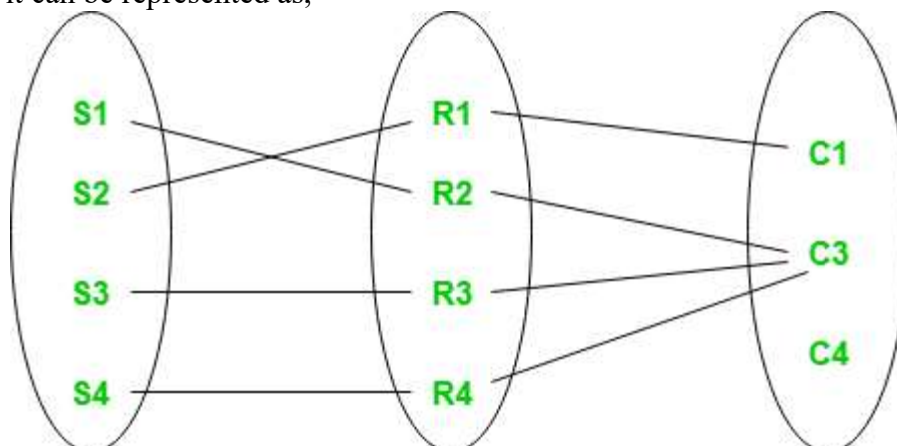
ii) Participation constraints

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

The diagram depicts the 'Enrolled in' relationship set with Student Entity set having total participation and Course Entity set having partial participation.



Using set, it can be represented as,



- Every student in Student Entity set is participating in relationship (total participation)
- but there exists a course C4 which is not taking part in the relationship (partial participation)