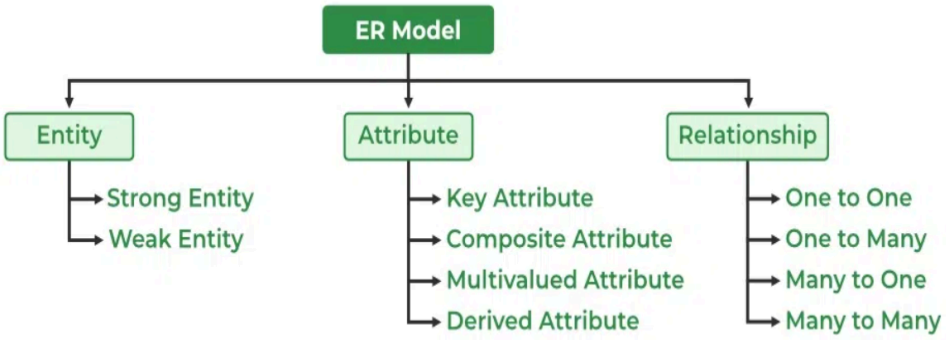
Entity-Relationship (ER) Model in RDBMS – Complete Guide

---

1. Introduction

The Entity-Relationship Model (ER Model) is a conceptual model for designing a database. This model represents the logical structure of a database, including entities, their attributes, and relationships between them.

- **Entity:** An object that is stored as data. E.g: *Student*, *Course,* or *Company*.
- **Attribute:** Properties that describe an entity. E.g: *StudentID*, *CourseName*, or *EmployeeEmail*.
- **Relationship:** A connection between entities. E.g: *Student* enrolls in a *Course*



An Entity Relationship Diagram (ERD) is a graphical representation that shows how entities, such as people, objects, or concepts, relate to each other within a system. ERDs are commonly used in database design to visualize the relationships between entities and their attributes.

- **Represent the logical structure of databases.**
- **Use symbols (rectangles, diamonds, ovals, lines) to show entities and relationships.**
- **Used in designing and debugging relational databases.**

Need to Draw ER Diagram

ER diagram or ER model, is drawn to visually represent the relationships between entities in a database system. Some benefits of drawing ER diagrams for database design are:

1. **It helps in understanding the data relationships.**
2. **ER diagrams are like a blueprint for designing a database.**

3. **Helps in communicating about databases with database designers, developers, users, etc.**
4. **They help in describing different relationships and operations within an organization.**

How to Draw an Entity Relation Diagram (ERD)

A step-by-step process to draw an entity relation diagram (ERD) is:

Step 1: Identifying Entities

Determine the main objects you want to represent in the database. E.g., "students", "courses", or "products".

Step 2: Defining Attributes

Identify the properties(attributes) of properties of each entity. These attributes provide more details about an entity.

Step 3: Specifying Relationships

Create relationships between entities to specify how entities interact with each other. Relationships are verbs like "teaches", "studies", or "sells".

Step 4: Drawing Entities

Draw entities as rectangle and write the name.

Step 5: Adding Attributes

To add attributes of a entity write attributes inside the rectangle or connect them with lines.

Step 6: Connecting Entities

Draw lines between the related entities to represent their connection.

Step 7: Specifying Cardinality

Indicate the minimum and maximum number of relationship instances associated with an entity using notations like crow's foot.

Step 8: Organizing ER Diagram

Organize all entities and relationships in a clean way for better readability and understanding.

Draw Entity Relationship Diagram Example

After learning the steps of how to draw an entity relationship diagram, we will create a demo ER diagram.

Let’s take an example of ER diagram for a bank through which we can learn how to design an ER and understand all the required methods.

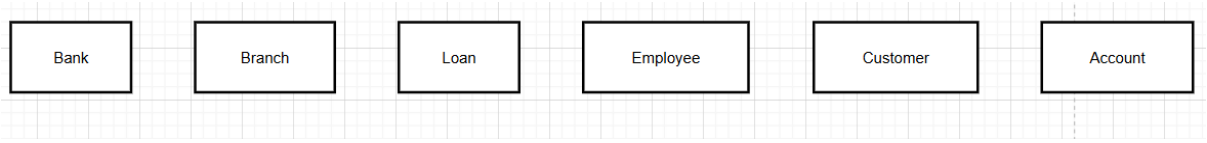
Entity Relationship Diagram for BANK

We will follow the steps mentioned above, to draw entity relationship diagram for bank.

Defining Entities

A thing in the real world with an independent existence. It is may be an object with physical existence (ex: house, person) or with a conceptual existence (ex: course, job). The are represented by rectangle.

Entities for Bank are: Bank, Branch, Employee, customer, loan, account.



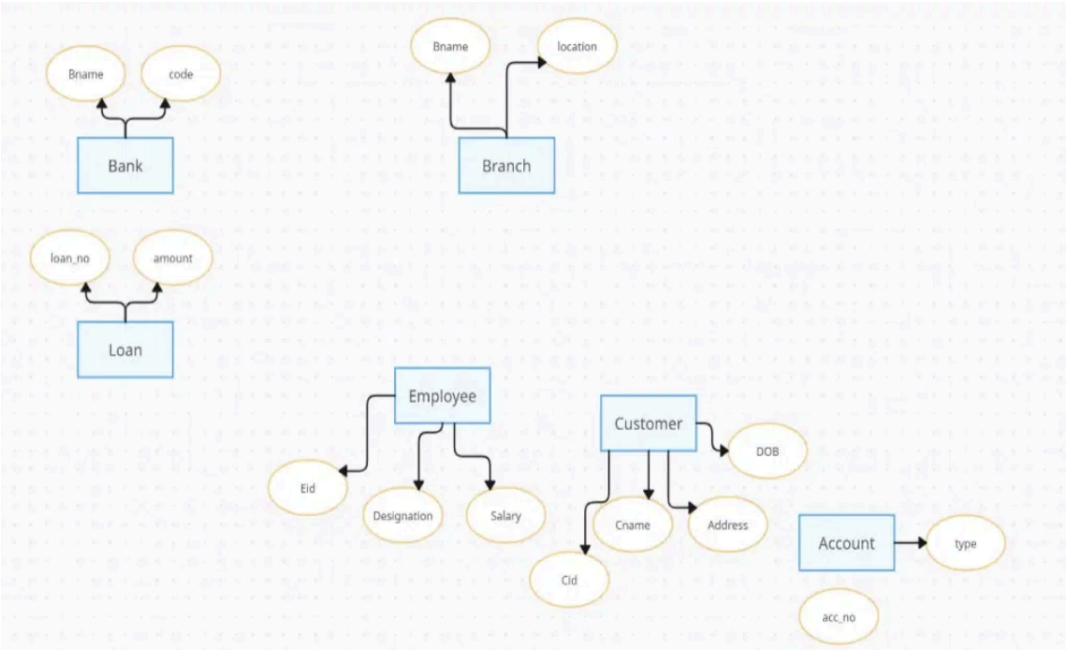
Entities

Adding Attributes

Attributes are the kind of properties that describe the entities. They are represented by ovals.

Attributes for Bank are:

- **For Bank Entity the Attributes are Bname, code.**
- **For Branch Entity the Attributes are Blocation, Bname.**
- **For Employee Entity the Attributes are Eid, Designation, salary.**
- **For Customer Entity the Attributes are Cid, Cname, Address, DOB.**
- **For Loan Entity the Attributes are Loan_no, amount, rate.**
- **For Account Entity the Attributes are acc_no, type.**



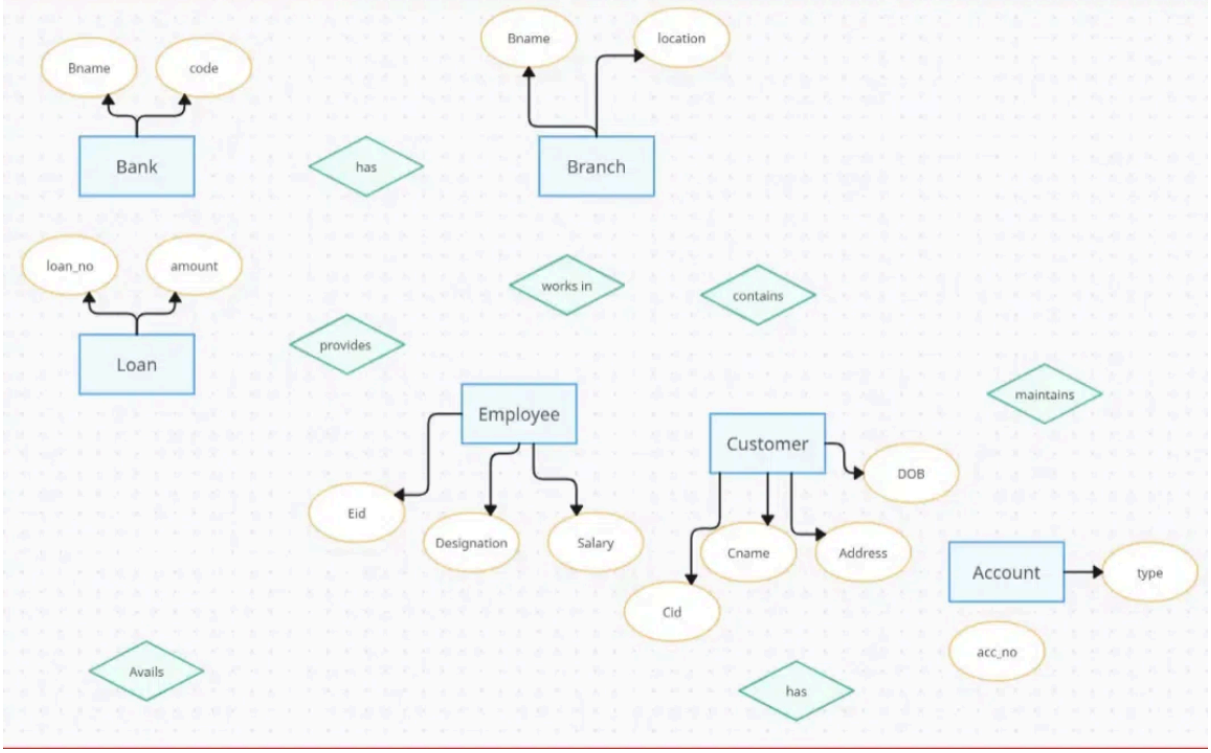
Attributes

Establishing Relationships

Entities have some relationships with each other. Relationships define how entities are associated with each other.

Let's Establishing Relationships between them are:

- **The Bank has branches.**
- **The Branch provides loan.**
- **The Employee works in branch.**
- **The Branch contains customers.**
- **The Customers has account.**
- **The Branch maintains account.**
- **The Customer avails loan.**



Relationships

Specifying Cardinality

Cardinality defines the numerical constraints on the relationships between entities. It is a notation that tells the ERD reader whether there are one, many or some combination of those factors between each entity.

1. One to One relationship(1:1)

In the entity relationship diagram, the one to one relationship means that single entity in one table is associated with single entity in another table. For example, one driver have only one license.

2. One to Many relationship(1:N)

One to many relationships means that single entity in one table is related to more than one entities in another table. For example, one bank has many branches.

3. Many to One(N:1)

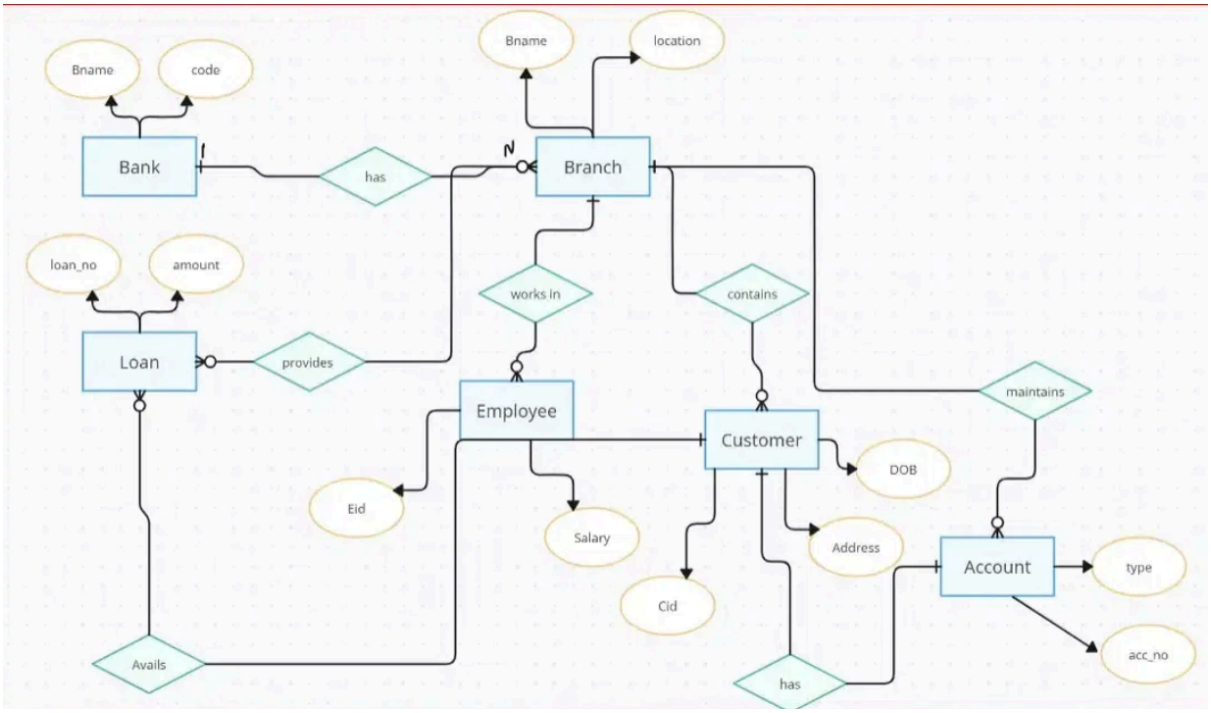
Many-to-one relationship in entity relationship diagram means that multiple entities are related to only single entity in another table. For example, many developers works on single project.

4. Many to Many relationship(M:N)

Many to many relationship means that multiple entities in one table is associated with multiple entities in another table. For example, multiple customers can have multiple accounts.

Specify cardinality for Bank:

- **Bank and branch has One to Many relationship (a bank has multiple branches).**
- **Branch and loan has also One to Many relationship(one branch can provide multiple loans).**
- **Branch and employee has One to Many relationship(one branch employs multiple employees).**
- **Branch and account has One to Many relationship(one branch maintains multiple accounts).**
- **Branch and customer has One to Many relationship(one branch has multiple customers; each customer belongs to one primary branch).**
- **Customer and account has One to Many relationship(one customer can have multiple accounts, each account belongs to exactly one customer).**
- **Customer and loan has One to Many relationships(each customer can have multiple loans; each loan belongs to exactly one customer)**



ER Diagram

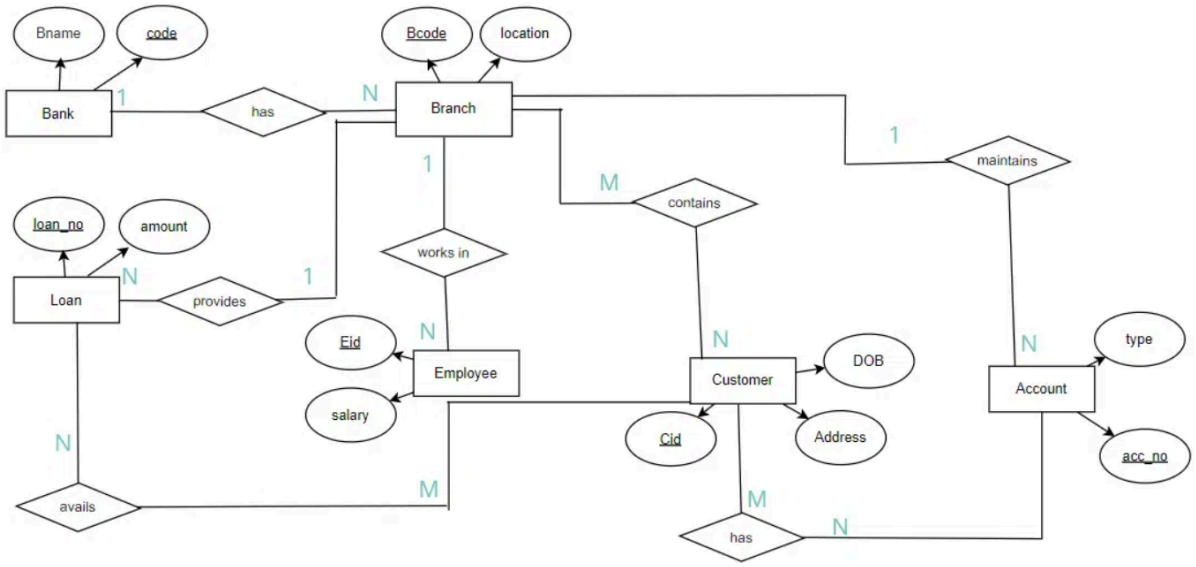
Identify Primary Keys

Primary keys are the unique identifier for each record in database table. It is denoted by an underline under the attribute name.

- **The Primary key of Bank is code.**
- **The Primary key of Branch is branch_code.**
- **The Primary key of Employee is Eid.**
- **The Primary key of Customer is Cid.**
- **The Primary key of Loan is loan_no.**
- **The Primary key of Account is acc_no.**

Final ER Diagram

The below diagram is our final entity relationship diagram for bank with all entities, their attributes and the relationship between them with the PRIMARY KEY and Cardinality ratio.



Final ER Diagram

ER Model in Database Design Process

We typically follow the below steps for designing a database for an application.

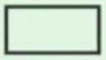
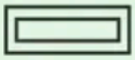
- **Gather the requirements (functional and data) by asking questions to the database users.**
- **Create a logical or conceptual design of the database. This is where ER model plays a role. It is the most used graphical representation of the conceptual design of a database.**
- **After this, focus on Physical Database Design (like indexing) and external design (like views)**

Use ER Diagrams In DBMS

- **ER diagrams represent the E-R model in a database, making them easy to convert into relations (tables).**
- **These diagrams serve the purpose of real-world modeling of objects which makes them intently useful.**
- **Unlike technical schemas, ER diagrams require no technical knowledge of the underlying DBMS used.**
- **They visually model data and its relationships, making complex systems easier to understand.**

Symbols Used in ER Model

ER Model is used to model the logical view of the system from a data perspective which consists of these symbols:

Figures	Symbols	Represents
Rectangle		Entities in ER Model
Ellipse		Attributes in ER Model
Diamond		Relationships among Entities
Line		Attributes to Entities and Entity Sets with Other Relationship Types
Double Ellipse		Multi-Valued Attributes
Double Rectangle		Weak Entity

Entity

An Entity represents a real-world object, concept or thing about which data is stored in a database. It act as a building block of a database. Tables in relational database represent these entities.

Example of entities:

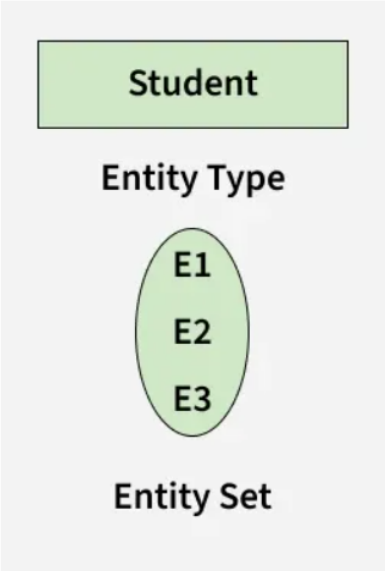
- **Real-World Objects: Person, Car, Employee etc.**
- **Concepts: Course, Event, Reservation etc.**
- **Things: Product, Document, Device etc.**

The entity type defines the structure of an entity, while individual instances of that type represent specific entities.

Entity Set

An entity refers to an individual object of an entity type, and the collection of all entities of a particular type is called an entity set. For example, E1 is an entity that belongs to the entity type "Student," and the group of all students forms the entity set.

In the ER diagram below, the entity type is represented as



Entity Set

We can represent the entity sets in an ER Diagram but we can't represent individual entities because an entity is like a row in a table, and an ER diagram shows the structure and relationships of data, not specific data entries (like rows and columns). An ER diagram is a visual representation of the data model, not the actual data itself.

Types of Entity

There are two main types of entities:

1. Strong Entity

A Strong Entity is a type of entity that has a key Attribute that can uniquely identify each instance of the entity. A Strong Entity does not depend on any other Entity in the Schema for its identification. It has a primary key that ensures its uniqueness and is represented by a rectangle in an ER diagram.

2. Weak Entity

A Weak Entity cannot be uniquely identified by its own attributes alone. It depends on a strong entity to be identified. A weak entity is associated with an identifying entity (strong entity), which helps in its identification. A weak entity are represented by a double rectangle. The participation of weak entity types is always total. The relationship between the weak entity type and its identifying strong entity type is called identifying relationship and it is represented by a double diamond.

An entity is a “thing” or “object” in the real world. An entity contains attributes, which describe that entity. So anything about which we store information is called an entity. Entities are recorded in the database and must be distinguishable, i.e., easily recognized from the group. In this article, we will see the difference between strong and weak entities.

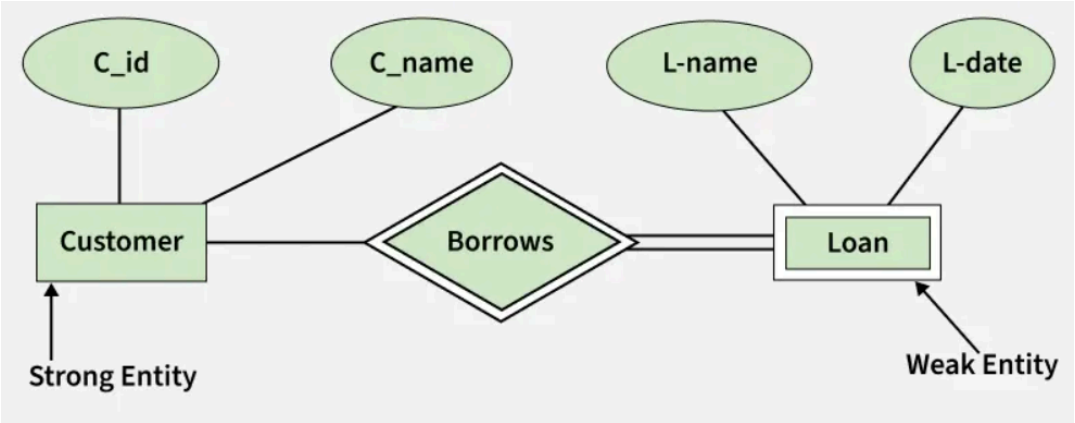
What is a Strong Entity?

A strong entity is not dependent on any other entity in the schema. A strong entity will always have a primary key. Strong entities are represented by a single rectangle. The

relationship of two strong entities is represented by a single diamond. Various strong entities, when combined together, create a strong entity set.

What is a Weak Entity?

A weak entity is dependent on a strong entity to ensure its existence. Unlike a strong entity, a weak entity does not have any primary key. It instead has a partial discriminator key. A weak entity is represented by a double rectangle. The relation between one strong and one weak entity is represented by a double diamond. This relationship is also known as an identifying relationship.



Weak Entity

In ER models, strong entities can exist independently, whereas weak entities depend on strong entities. If you want to solidify your understanding of ER diagrams and entity relationships, the GATE CS Self-Paced Course covers database concepts in-depth, including practical exercises on ER modeling

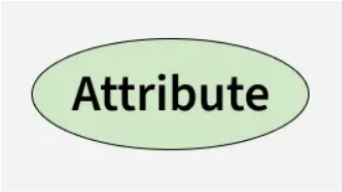
Difference Between Strong and Weak Entity

Strong Entity	Weak Entity
Strong entity always has a primary key.	While a weak entity has a partial discriminator key.
Strong entity is not dependent on any other entity.	Weak entity depends on strong entity.
Strong entity is represented by a single rectangle.	Weak entity is represented by a double rectangle.
Two strong entity's relationship is represented by a single diamond.	While the relation between one strong and one weak entity is represented by a double diamond.

Strong Entity	Weak Entity
Strong entities have either total participation or partial participation.	A weak entity has a total participation constraint.

Attributes in ER Model

Attributes are the properties that define the entity type. For example, for a Student entity Roll_No, Name, DOB, Age, Address, and Mobile_No are the attributes that define entity type Student. In ER diagram, the attribute is represented by an oval.



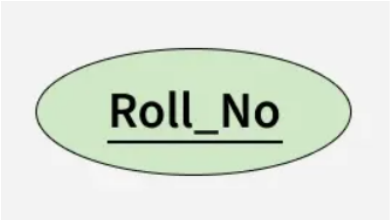
Attribute

Attribute

Types of Attributes

1. Key Attribute

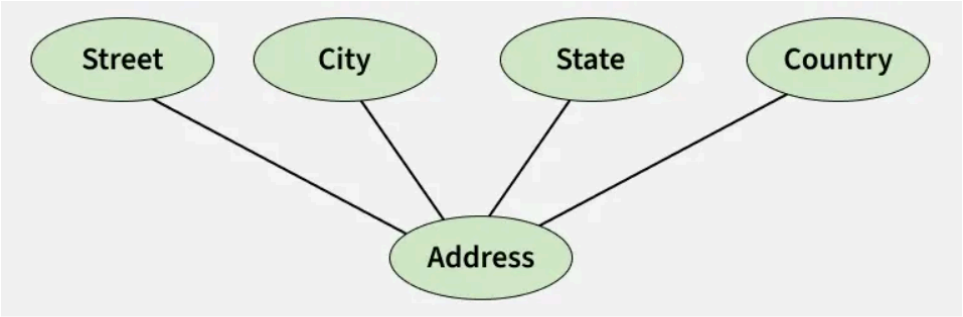
The attribute which uniquely identifies each entity in the entity set is called the key attribute. For example, Roll_No will be unique for each student. In ER diagram, the key attribute is represented by an oval with an underline.



Key Attribute

2. Composite Attribute

An attribute composed of many other attributes is called a composite attribute. For example, the Address attribute of the student Entity type consists of Street, City, State, and Country. In ER diagram, the composite attribute is represented by an oval comprising of ovals.



Composite Attribute

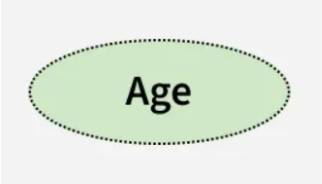
3. Multivalued Attribute

An attribute consisting of more than one value for a given entity. For example, Phone_No (can be more than one for a given student). In ER diagram, a multivalued attribute is represented by a double oval.

Multivalued Attribute

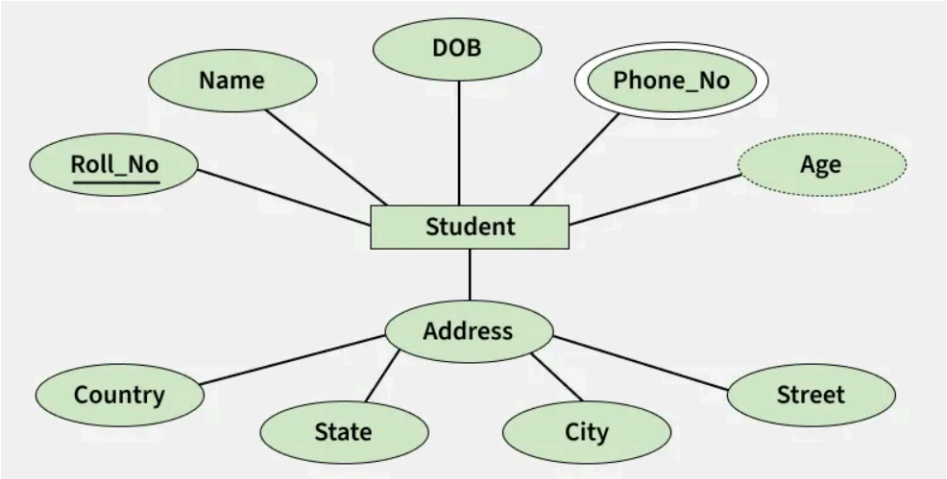
4. Derived Attribute

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



Derived Attribute

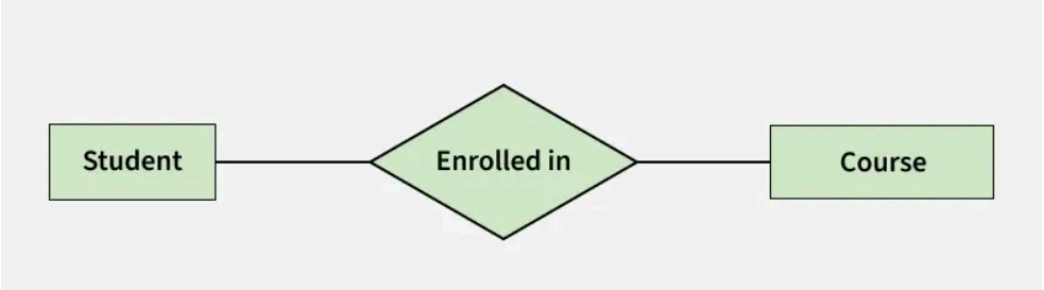
The Complete Entity Type Student with its Attributes can be represented as:



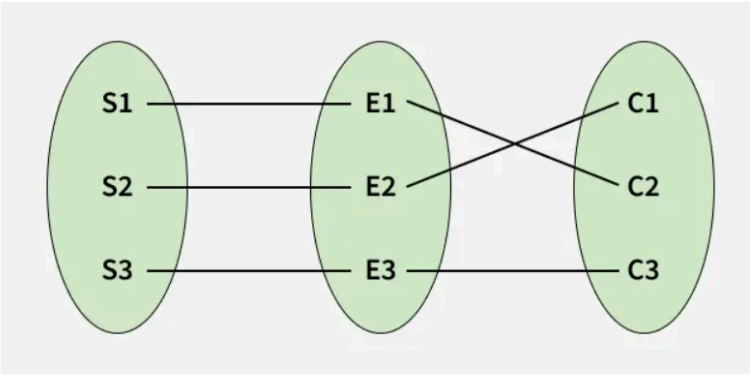
Entity and Attributes

Relationship Type and Relationship Set

A Relationship Type represents the association between entity types. For example, 'Enrolled in' is a relationship type that exists between entity type Student and Course. In ER diagram, the relationship type is represented by a diamond and connecting the entities with lines.



A set of relationships of the same type is known as a relationship set. The following relationship set depicts S1 as enrolled in C2, S2 as enrolled in C1, and S3 as registered in C3.

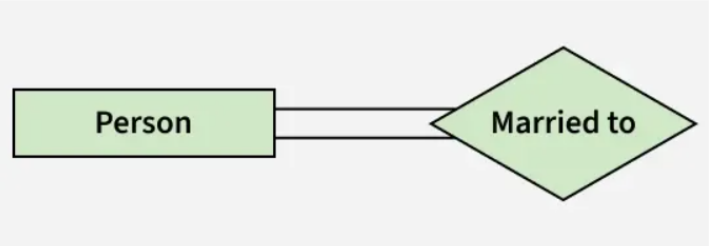


Relationship Set

Degree of a Relationship Set

The number of different entity sets participating in a relationship set is called the degree of a relationship set.

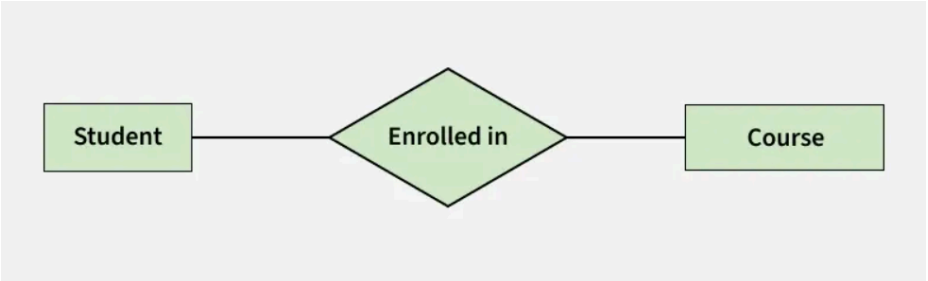
1. Unary/Recursive Relationship: When there is only ONE entity set participating in a relation, the relationship is called a unary relationship. For example, one person is married to only one person.



Unary Relationship

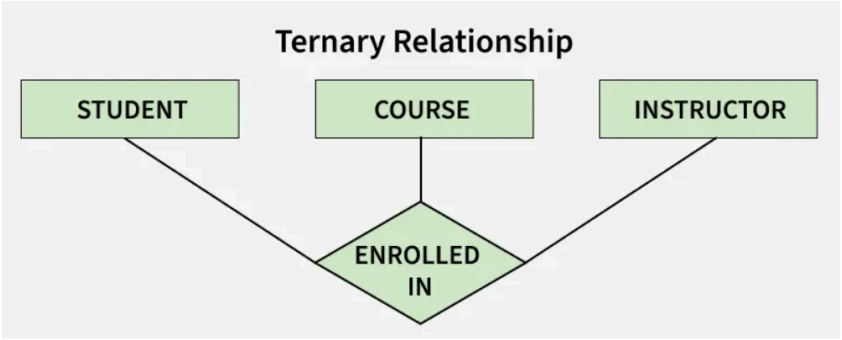
Read more about - Recursive Relationship

2. Binary Relationship: When there are TWO entities set participating in a relationship, the relationship is called a binary relationship. For example, a Student is enrolled in a Course.



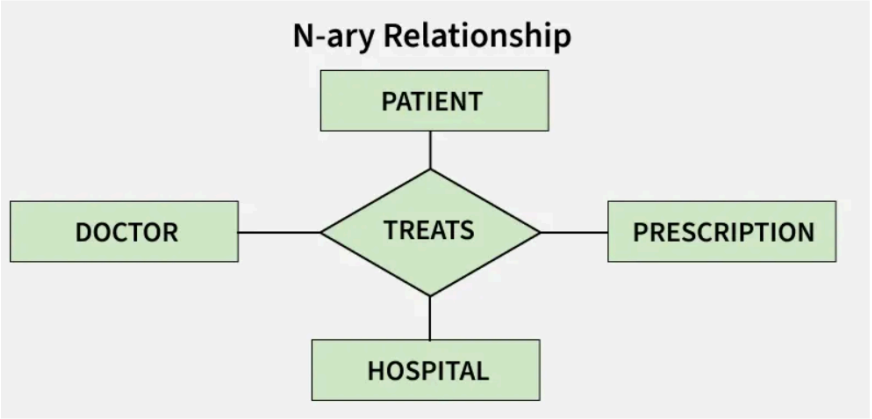
Binary Relationship

3. Ternary Relationship: When there are three entity sets participating in a relationship, the relationship is called a ternary relationship.



Ternary Relationship

4. N-ary Relationship: When there are n entities set participating in a relationship, the relationship is called an n-ary relationship.



N-ary Relationship

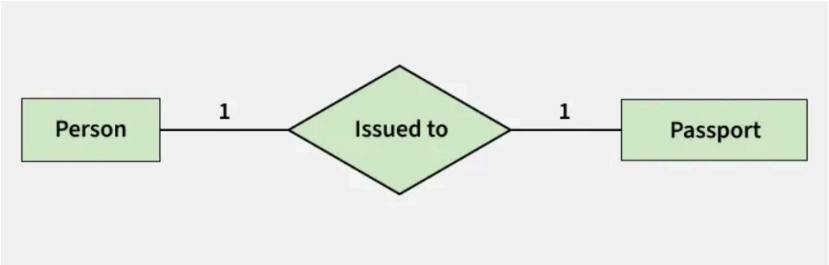
Cardinality in ER Model

The maximum number of times an entity of an entity set participates in a relationship set is known as cardinality.

Cardinality can be of different types:

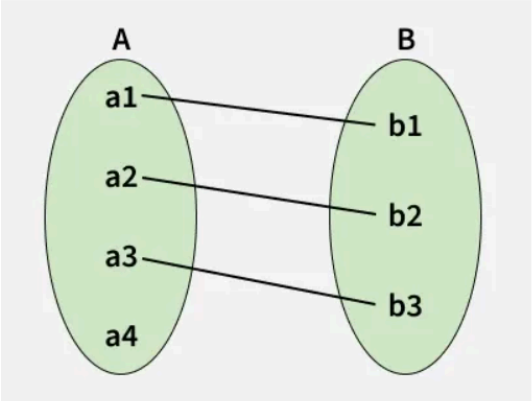
1. One-to-One

When each entity in each entity set can take part only once in the relationship, the cardinality is one-to-one. Let us assume that one person can be issued only one passport, and one passport is issued to only one person. So, the relationship will be One-to-One (1 : 1), meaning that each person has a single passport, and each passport belongs to a single person.



one to one cardinality

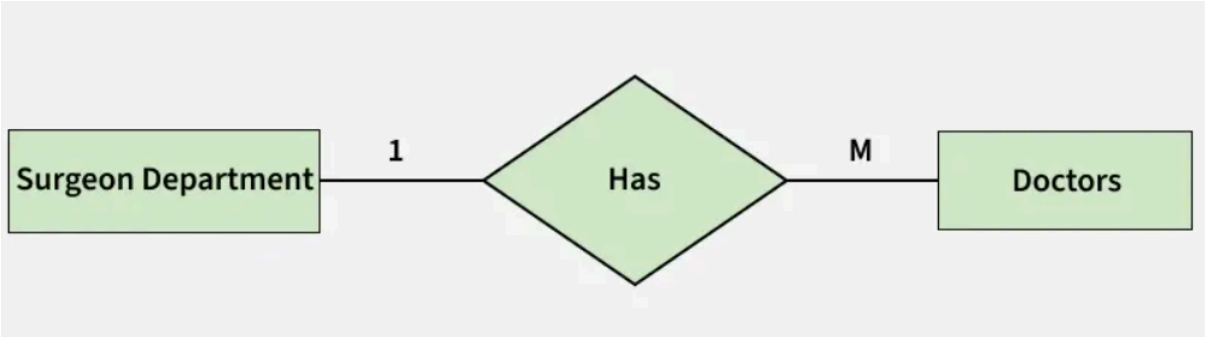
Using Sets, it can be represented as:



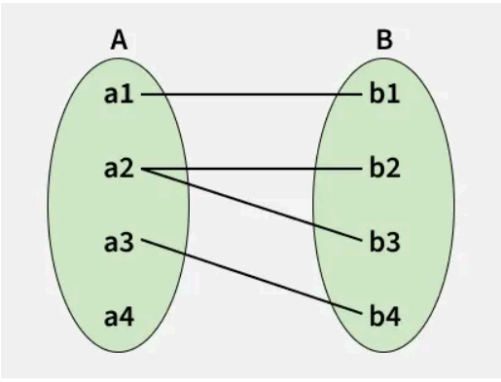
Set Representation of One-to-One

2. One-to-Many

In a one-to-many relationship, one entity can be associated with multiple entities. For example, a single Surgeon Department can have many Doctors. Therefore, the cardinality of this relationship is 1 to M, meaning one department can have many doctors.



Using sets, one-to-many cardinality can be represented as:

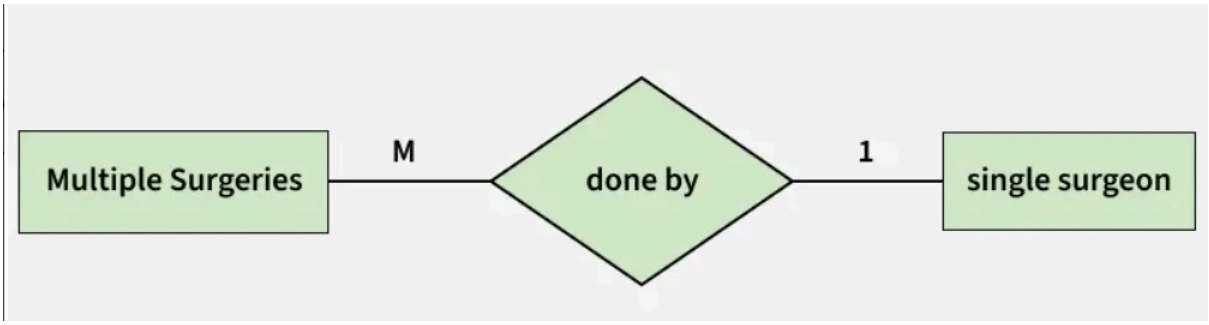


Set Representation of One-to-Many

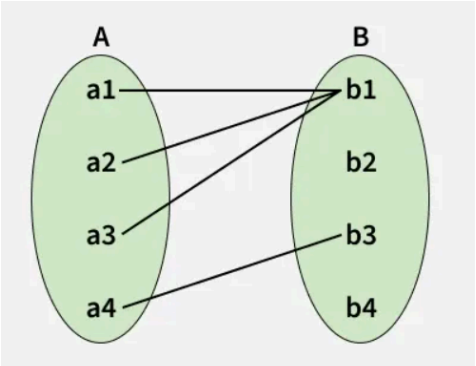
3. Many-to-One

When entities in one entity set can take part only once in the relationship set and entities in other entity sets can take part more than once in the relationship set, cardinality is many to one.

Let us assume that multiple surgeries can be performed by one surgeon, but one surgery is performed by only one surgeon. So, the cardinality will be M to 1, meaning that many surgeries can be done by a single surgeon, but each surgery is done by only one surgeon.



Using Sets, it can be represented as:

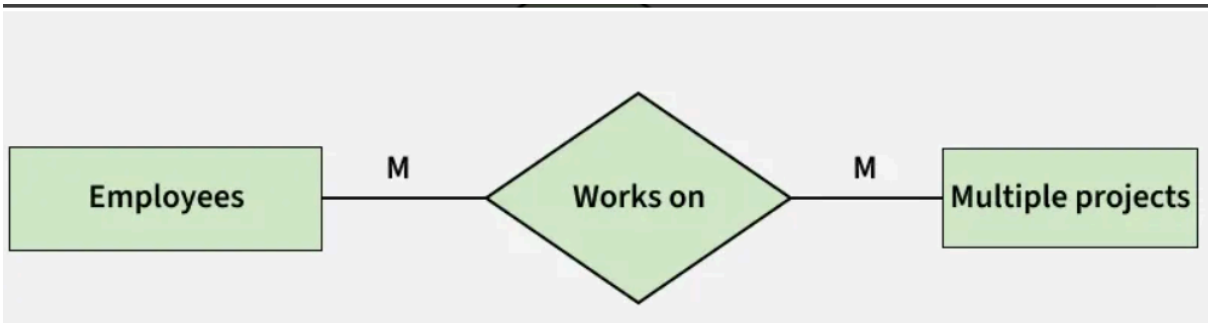


Set Representation of Many-to-One

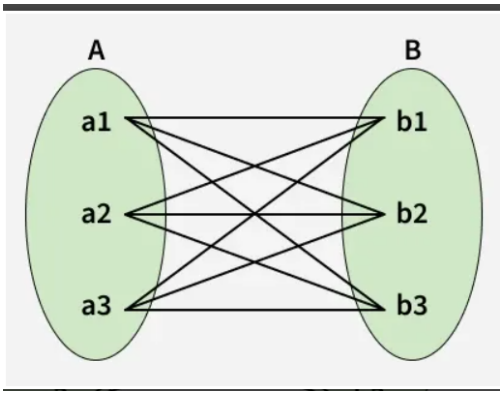
In this case, each student is taking only 1 course but 1 course has been taken by many students.

4. Many-to-Many

When entities in all entity sets can take part more than once in the relationship cardinality is many to many. Let us assume that an employee can work on multiple projects and each project can have multiple employees working on it. So, the relationship will be many-to-many (M:N), meaning that one employee may be associated with several projects, and one project may involve several employees.



Using Sets, it can be represented as:



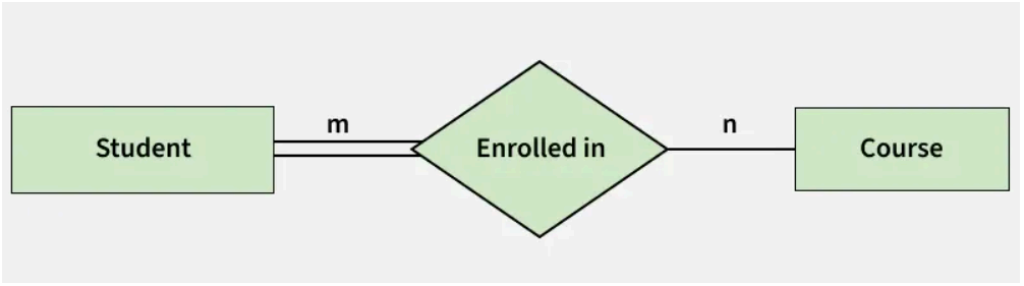
In this example, student A1 is enrolled in B1,B2 and B3, and course B3 is taken by A1, A2, and A3. Therefore, this represents a many-to-many relationship.

Participation Constraint

Participation Constraint is applied to the entity participating in the relationship set.

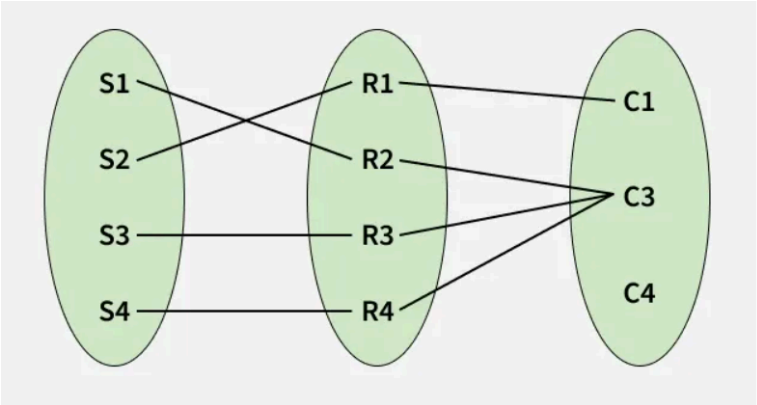
1. **Total Participation: Each entity in the entity set must participate in the relationship. If each student must enroll in a course, the participation of students will be total. Total participation is shown by a double line in the ER diagram.**
2. **Partial Participation: The entity in the entity set may or may NOT participate in the relationship. If some courses are not enrolled by any of the students, the participation in the 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.



Total Participation and Partial Participation

Using Set, it can be represented as,



Set representation of Total Participation and Partial Participation

Every student in the Student Entity set participates in a relationship but there exists a course C4 that is not taking part in the relationship.

How to Draw an ER Diagram

1. **Identify Entities: The very first step is to identify all the Entities. Represent these entities in a Rectangle and label them accordingly.**
2. **Identify Relationships: The next step is to identify the relationship between them and represent them accordingly using the Diamond shape. Ensure that relationships are not directly connected to each other.**
3. **Add Attributes: Attach attributes to the entities by using ovals. Each entity can have multiple attributes (such as name, age, etc.), which are connected to the respective entity.**
4. **Define Primary Keys: Assign primary keys to each entity. These are unique identifiers that help distinguish each instance of the entity. Represent them with underlined attributes.**
5. **Remove Redundancies: Review the diagram and eliminate unnecessary or repetitive entities and relationships.**
6. **Review for Clarity: Review the diagram make sure it is clear and effectively conveys the relationships between the entities.**