Database Management System 5

ER Modeling

Hrudaya Kumar Tripathy,

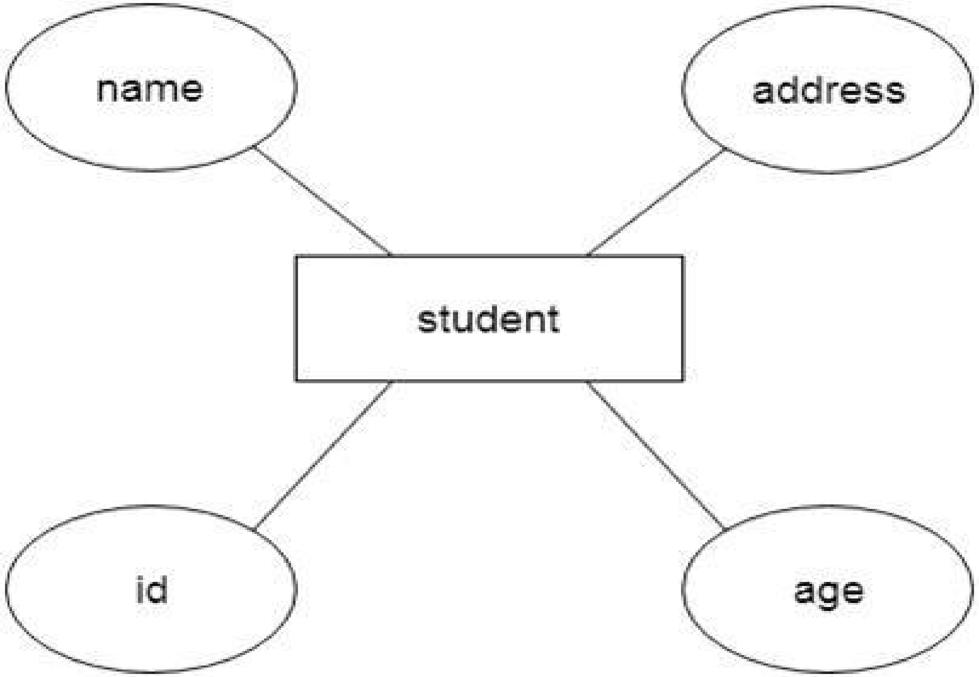
School of Computer Engineering, KIIT

- ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.

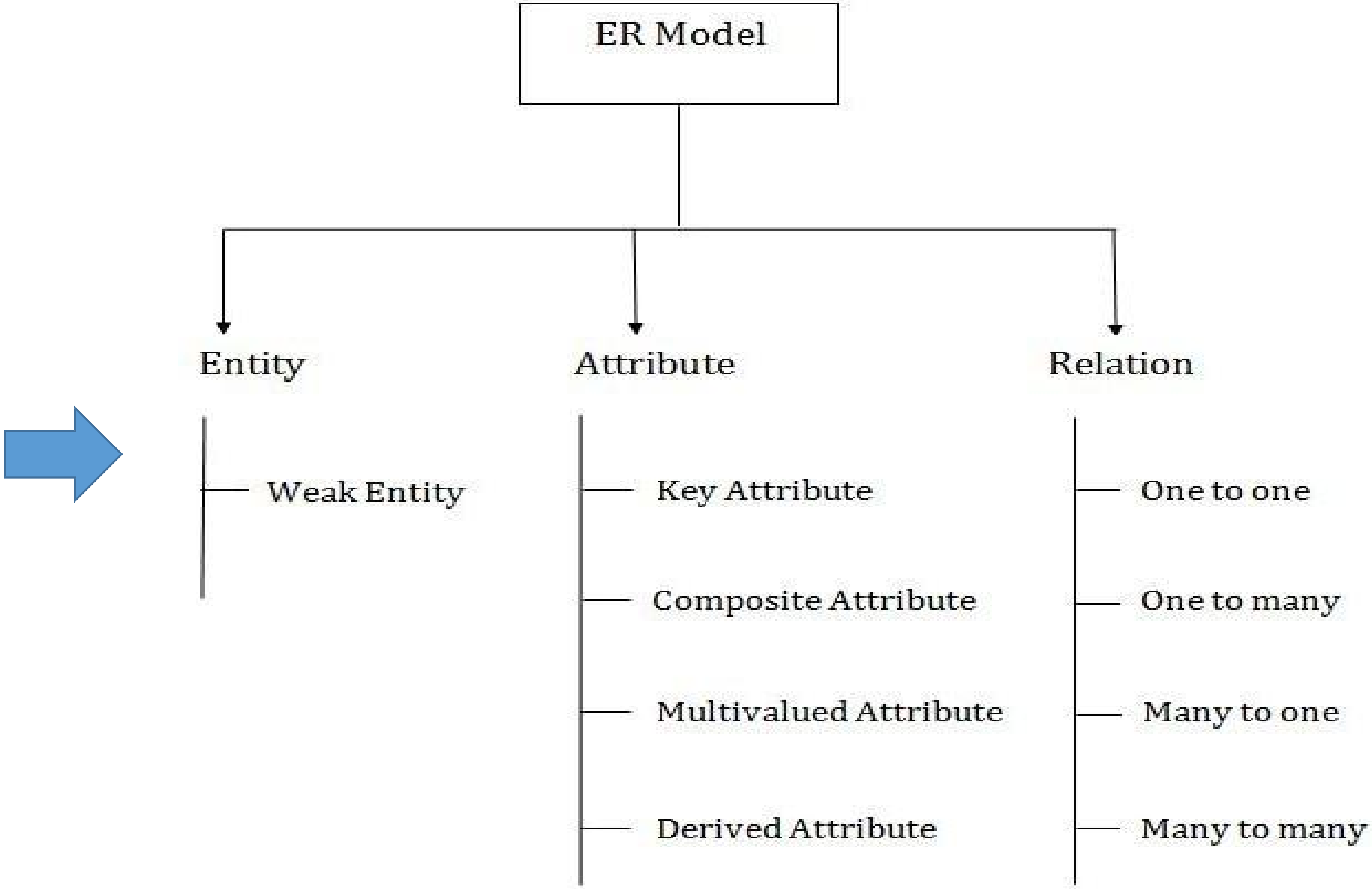
- It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.

- In ER modeling, the database structure is portrayed as a diagram called an entity- relationship diagram.

- For example, Suppose we design a school database. In this database, the student will be an entity with attributes like address, name, id, age, etc. The address can be another entity with attributes like city, street name, pin code, etc and there will be a relationship between them.



Component of ER Diagram



ER Model

Entity

- Weak Entity

Attribute

- Key Attribute
- Composite Attribute
- Multivalued Attribute
- Derived Attribute

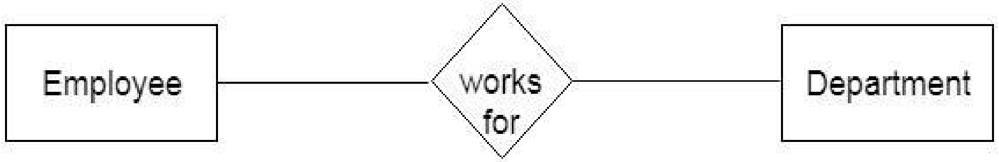
Relation

- One to one
- One to many
- Many to one
- Many to many

Entity:

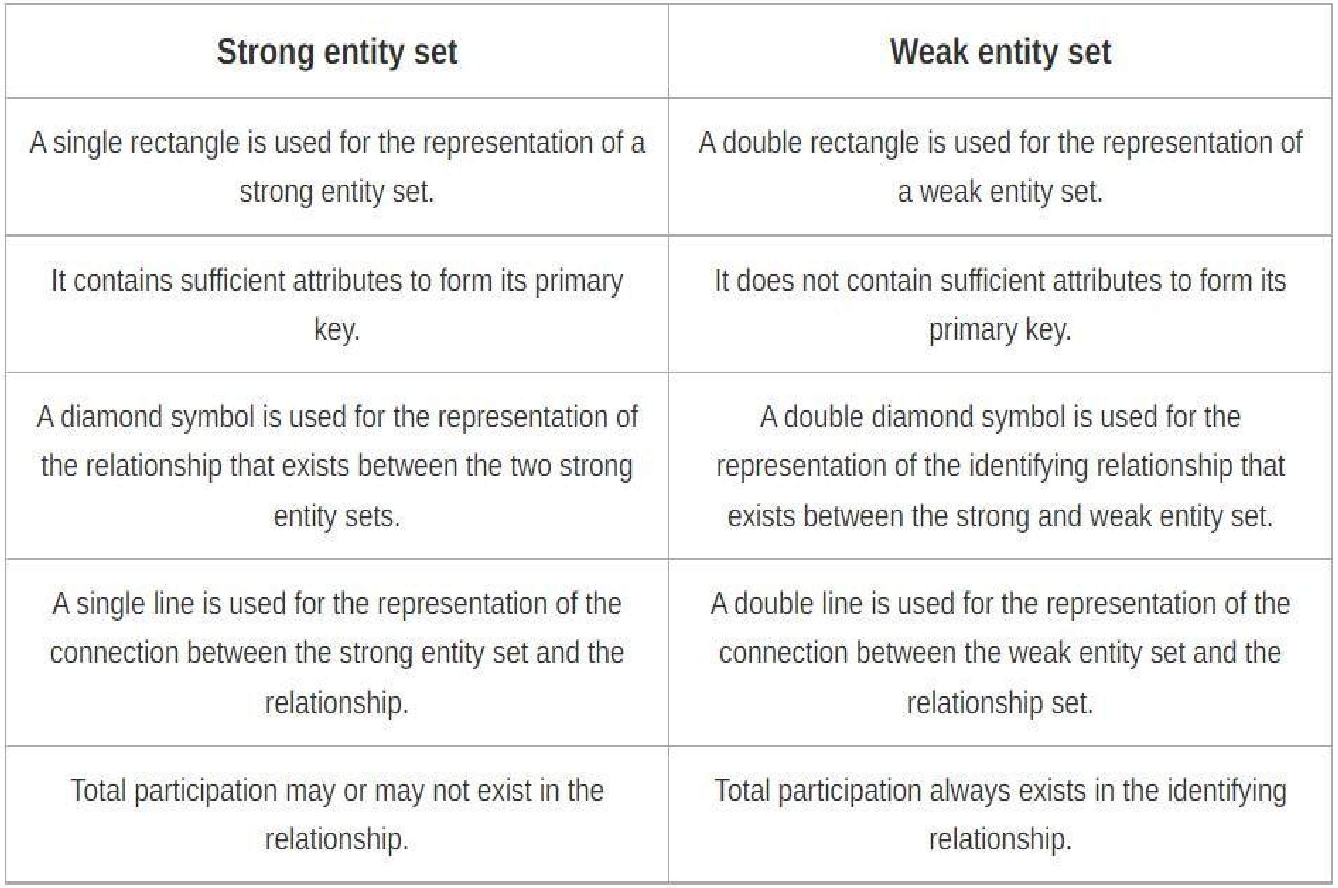
- An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

- Consider an organization as an example- manager, product, employee, department etc. can be taken as an entity.



- An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.

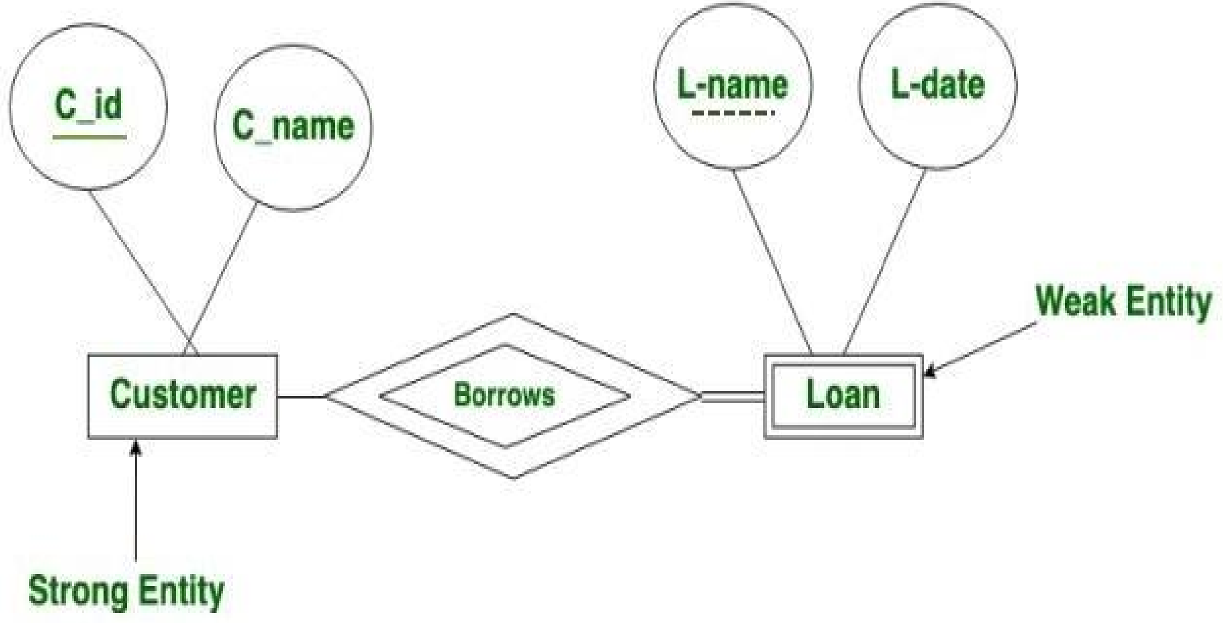
- Weak entities are represented with double rectangular box in the ER Diagram and the identifying relationships are represented with double diamond.



Strong entity set	Weak entity set
A single rectangle is used for the representation of a strong entity set.	A double rectangle is used for the representation of a weak entity set.
It contains sufficient attributes to form its primary key.	It does not contain sufficient attributes to form its primary key.
A diamond symbol is used for the representation of the relationship that exists between the two strong entity sets.	A double diamond symbol is used for the representation of the identifying relationship that exists between the strong and weak entity set.
A single line is used for the representation of the connection between the strong entity set and the relationship.	A double line is used for the representation of the connection between the weak entity set and the relationship set.
Total participation may or may not exist in the relationship.	Total participation always exists in the identifying relationship.

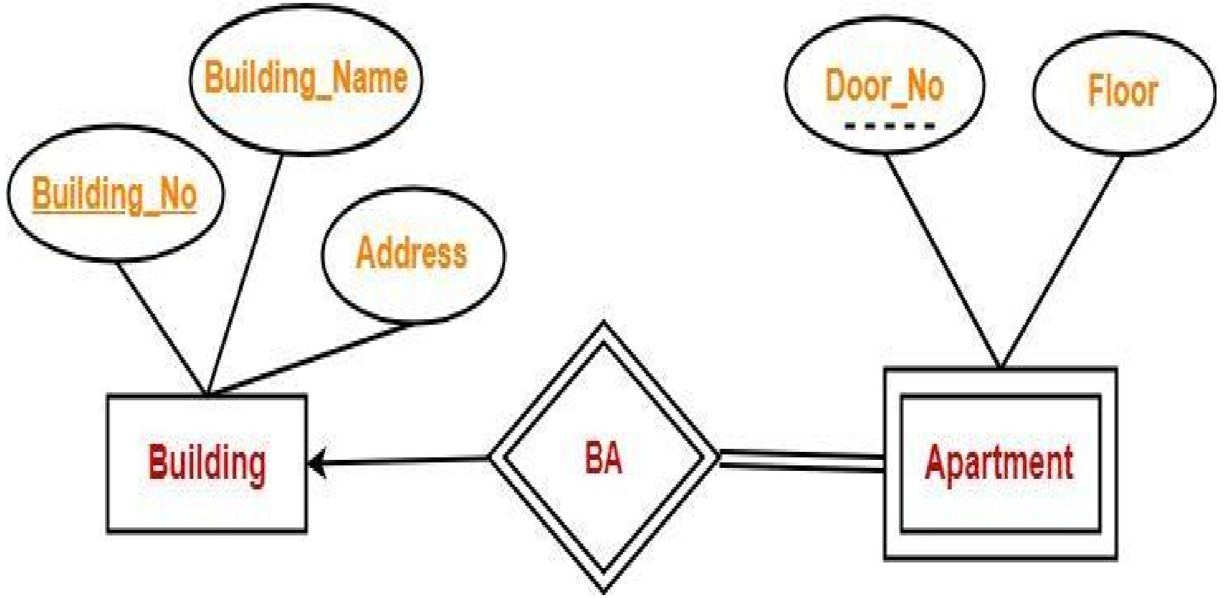
- ✓ The discriminator (partial key) of a weak entity set is the set of attributes that distinguishes between all the entities of a weak entity set.

- ✓ The primary key of a weak entity set is created by combining the primary key of the strong entity set on which the weak entity set is existence dependent and the weak entity set’s discriminator.



There are two entities Customer and Loan in which Customer is strong entity set and Loan is weak entity set.

Loan entity has L-name which is the discriminator. Customer entity has C_id as a primary key.
So the primary key for Loan is (C_id, L-name).



One strong entity set "Building" and one weak entity set "Apartment" are related to each other.

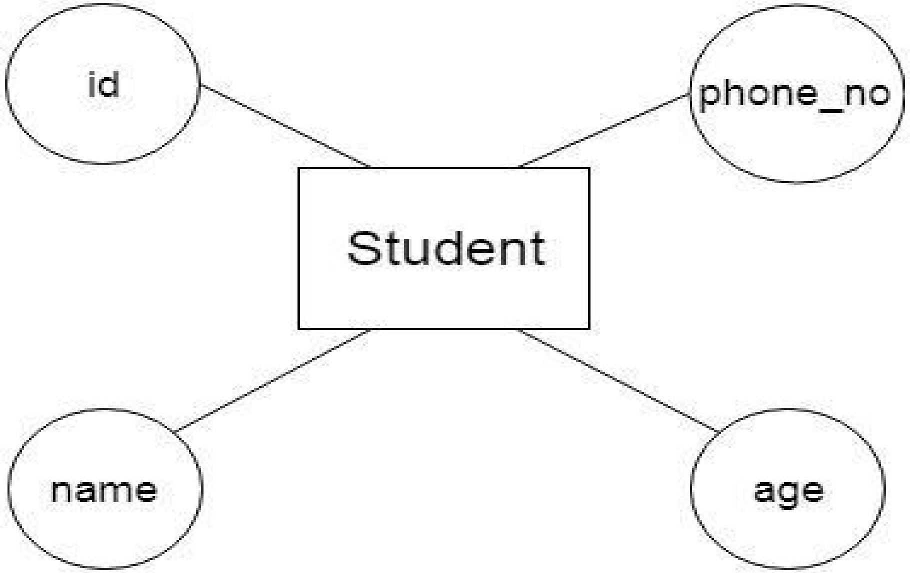
Strong entity set "Building" has building number as its primary key. Door number is the discriminator of the weak entity set "Apartment". This is because door number alone can not identify an apartment uniquely as there may be several other buildings having the same door number.

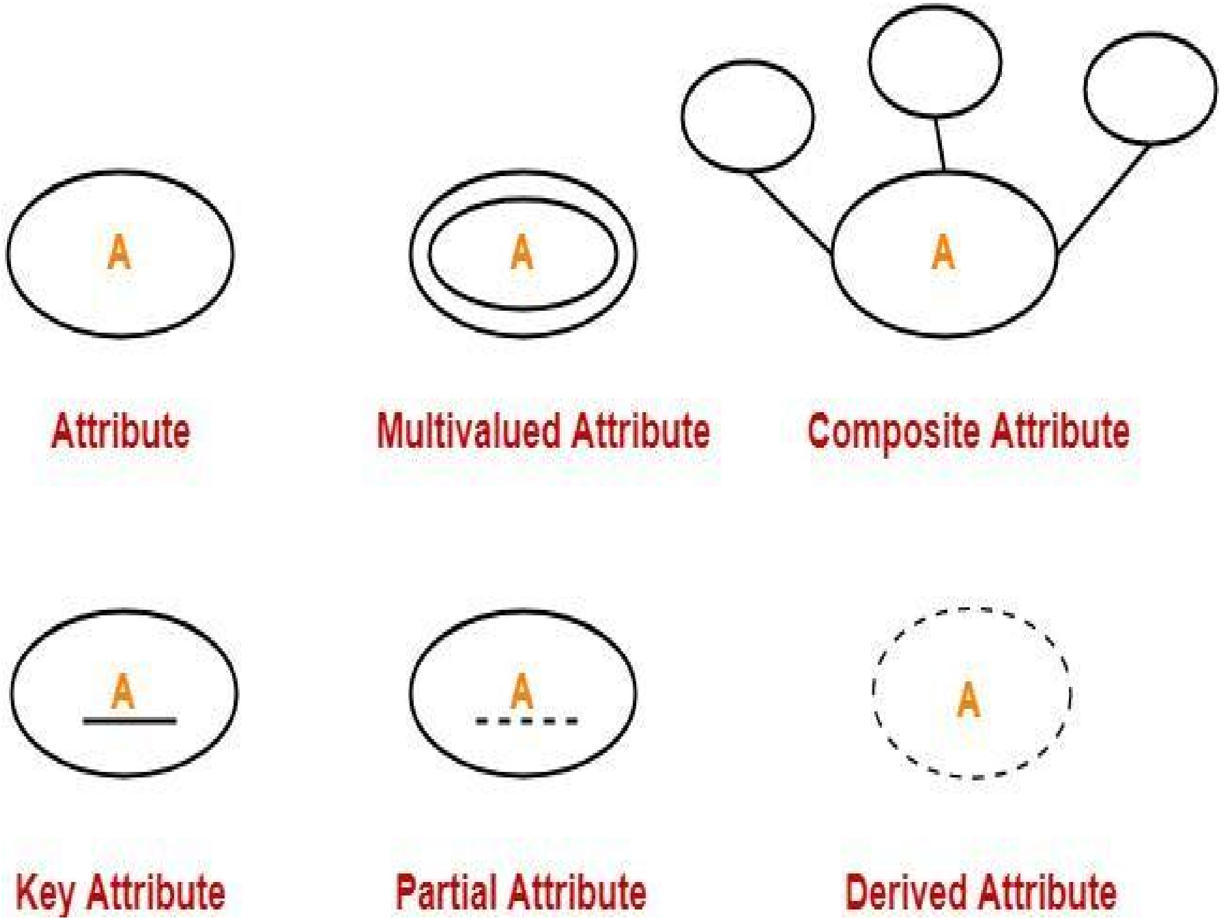
To uniquely identify any apartment, First, building number is required to identify the particular building. Secondly, door number of the apartment is required to uniquely identify the apartment.

Attribute:

- The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.

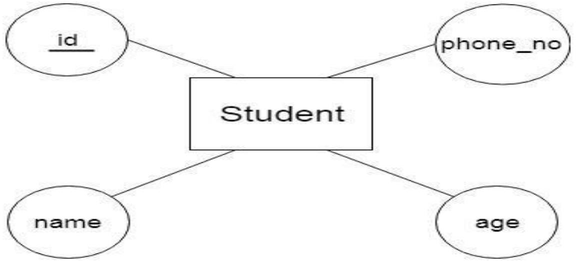
- For example, id, age, contact number, name, etc. can be attributes of a student.





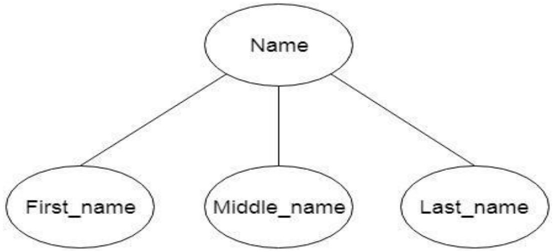
Key vs Non-key Attribute:

The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined. All other attributes other are non-key attributes.



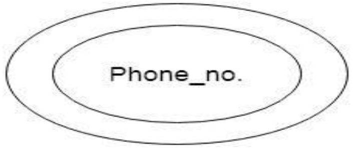
Simple vs Composite Attribute:

An attribute that has a discrete value & can't be subdivided is called a simple attribute. On the other hand, a composite attribute can be meaningfully subdivided into smaller subparts.



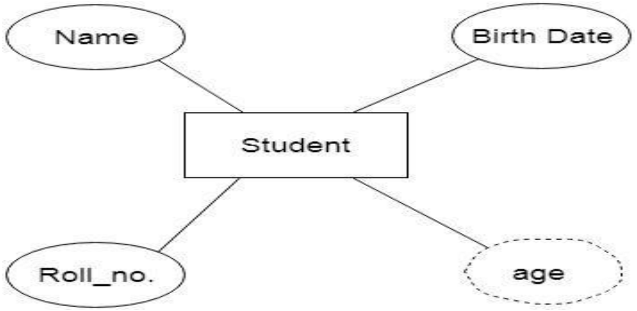
Single-valued vs Multi-valued Attribute:

Most attributes have a single value for a particular entity & are referred to as single-valued attribute. However, attributes that can have more than one value are known as multi-valued attributes. It is represented by double oval.



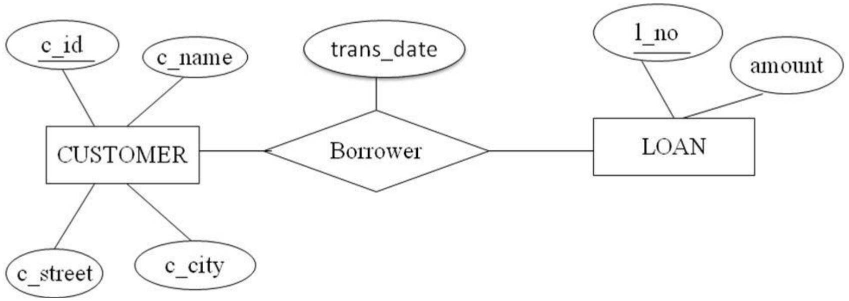
Stored vs Derived Attribute:

The attribute with independent existence is called as stored attribute where as the attribute, whose value is depending on other stored attribute, is called as derived attribute. The derived attribute is represented by the dotted oval.



Descriptive attributes:

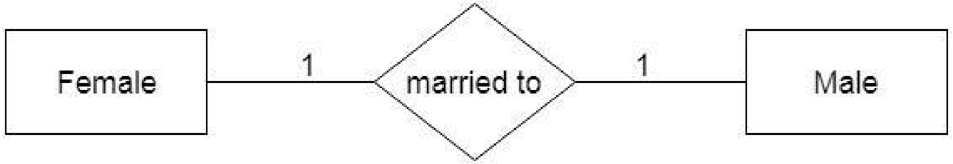
A relationship may also have attributes called descriptive attributes for representing the description about the association



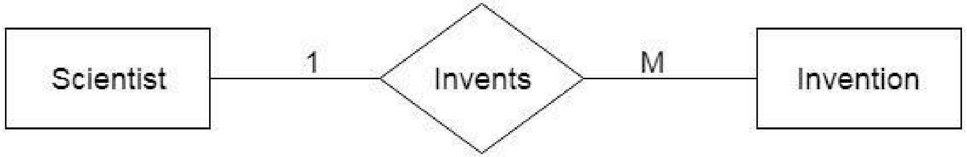
Relationship

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

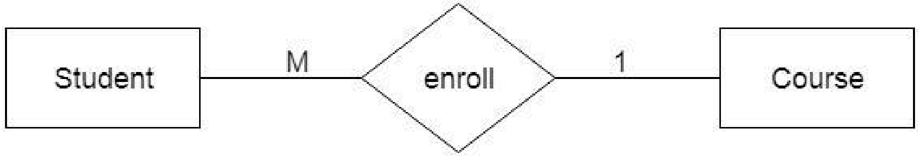
One-to-One Relationship: When only one instance of an entity is associated with the relationship, then it is known as one to one relationship. For example, A female can marry to one male, and a male can marry to one female.



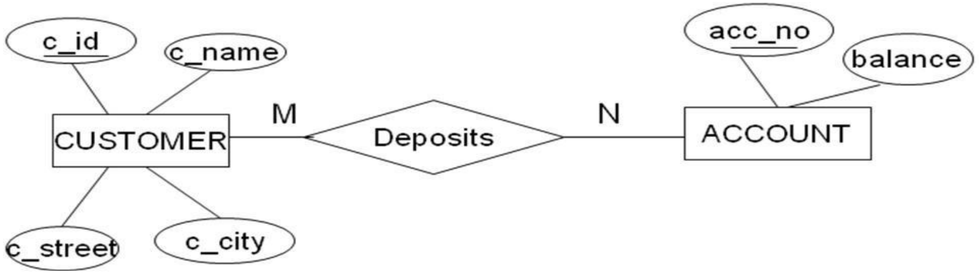
One-to-many relationship: When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship. For example, Scientist can invent many inventions, but the invention is done by the only specific scientist.



Many-to-one relationship: When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship. For example, Student enrolls for only one course, but a course can have many students.



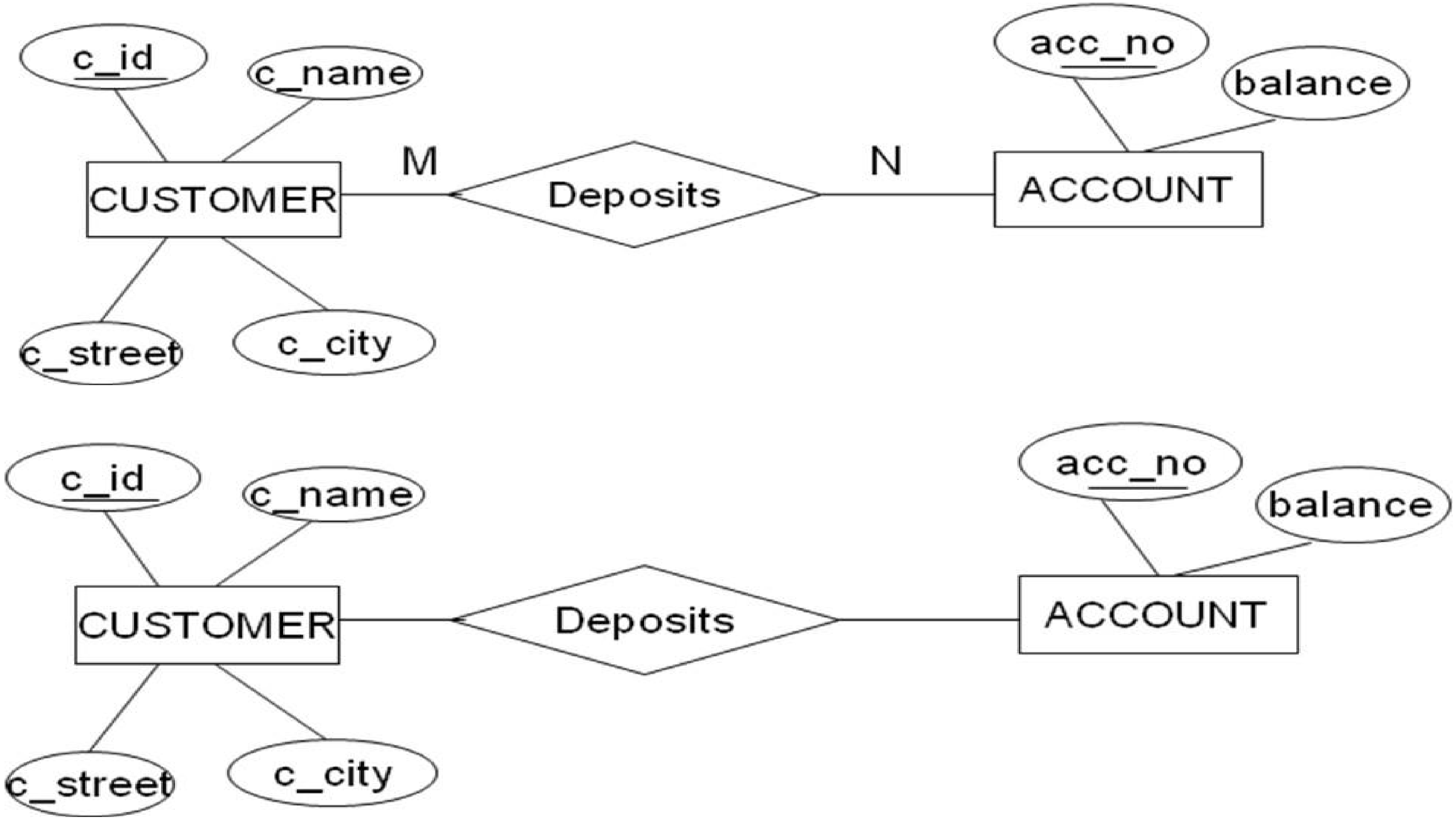
Many-to-many relationship: When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship. For example, Many customers can have many accounts in a bank.



Mapping Cardinality Representation

M:N relationship

An entity in A is associated with any number (zero or more) of entities in B and vice versa

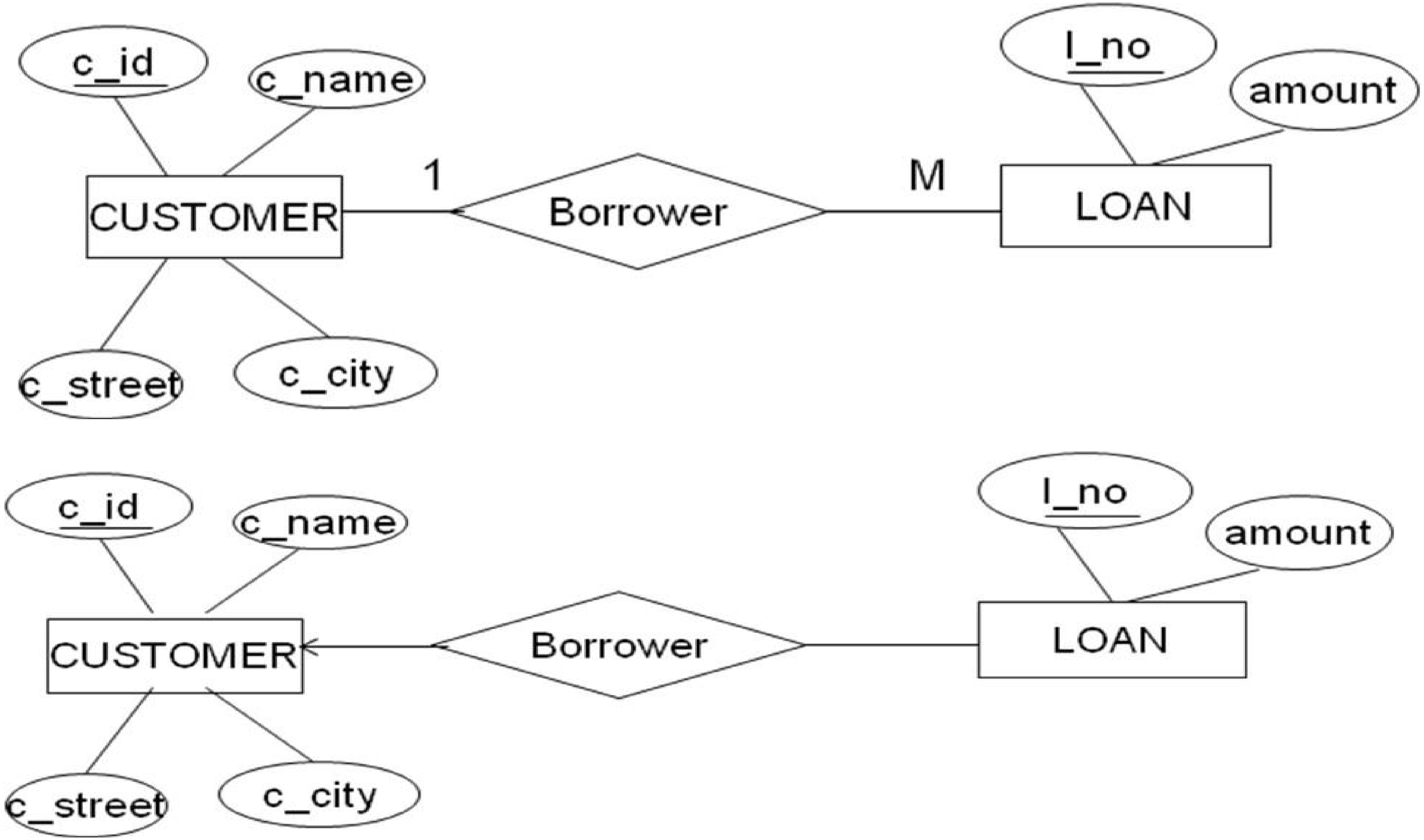


For example, Many customers can have many accounts in a bank.

Mapping Cardinality Representation

1:M relationship

An entity in A is associated with any number (zero or more) of entities in B; an entity in B, however, is associated with no more than 1 entity set of A

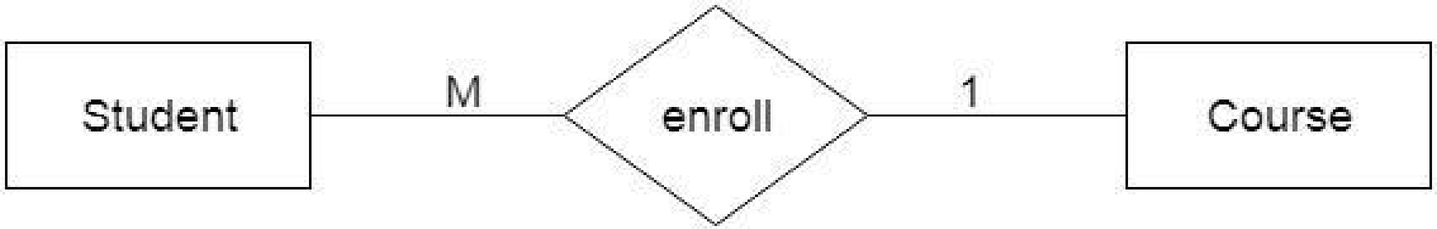


For example, Scientist can invent many inventions, but the invention is done by the only specific scientist.

Mapping Cardinality Representation

M:1 relationship

When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.

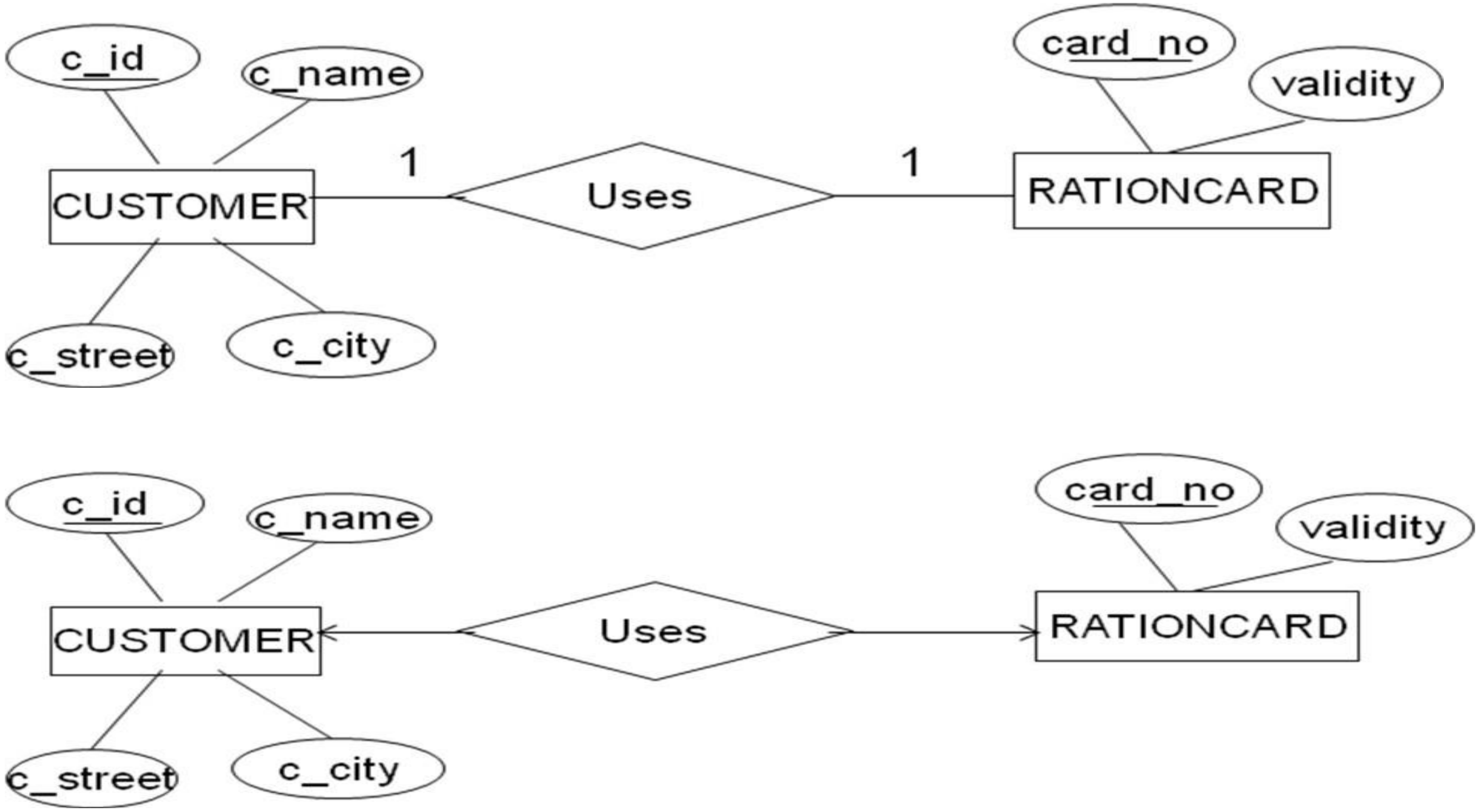


For example, Student enrolls for only one course, but a course can have many students.

Mapping Cardinality Representation

1:1 relationship

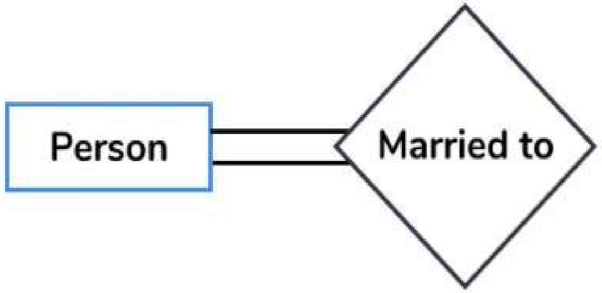
An entity in A is associated with no more than 1 entity of B; and an entity in B is associated with no more than 1 entity of A



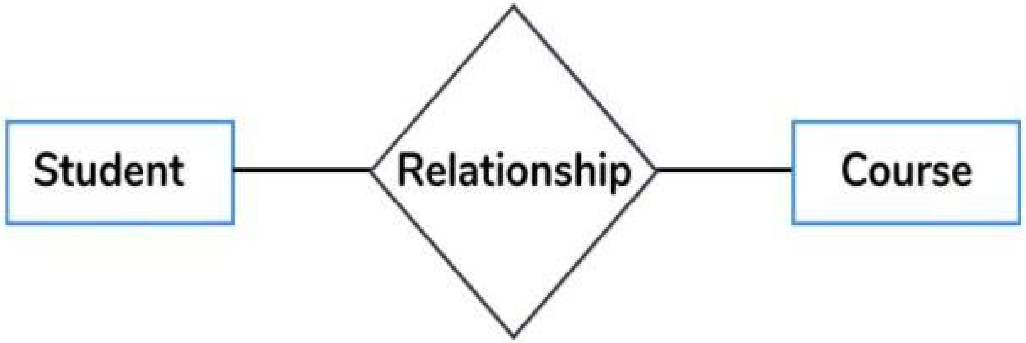
For example, A female can marry to one male, and a male can marry to one female.

There are four types of relationship that exist between Entities: **Unary, Binary, Ternary & Recursive**

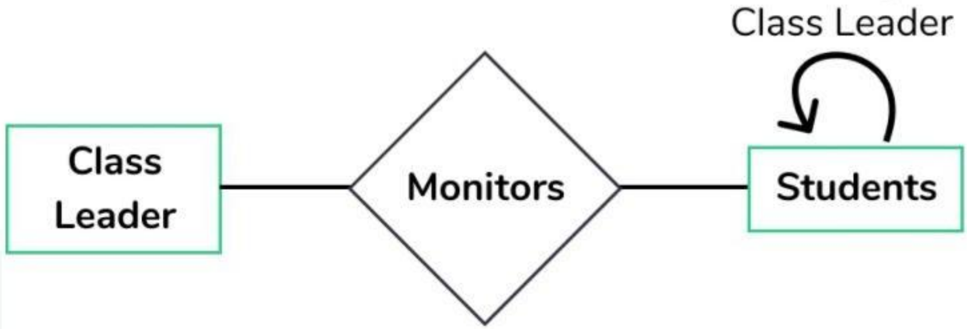
Unary Relationship



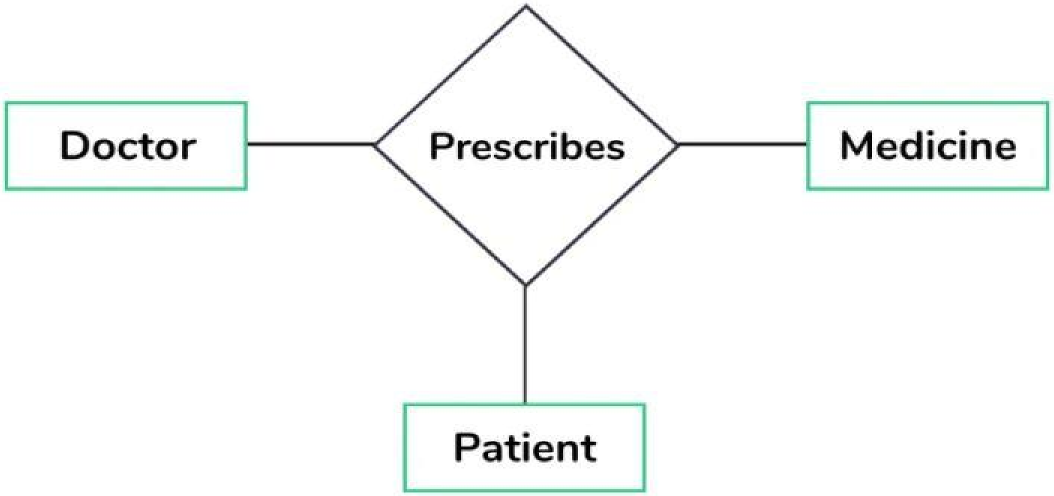
Binary Relationship



Recursive Relationship



Ternary Relationship



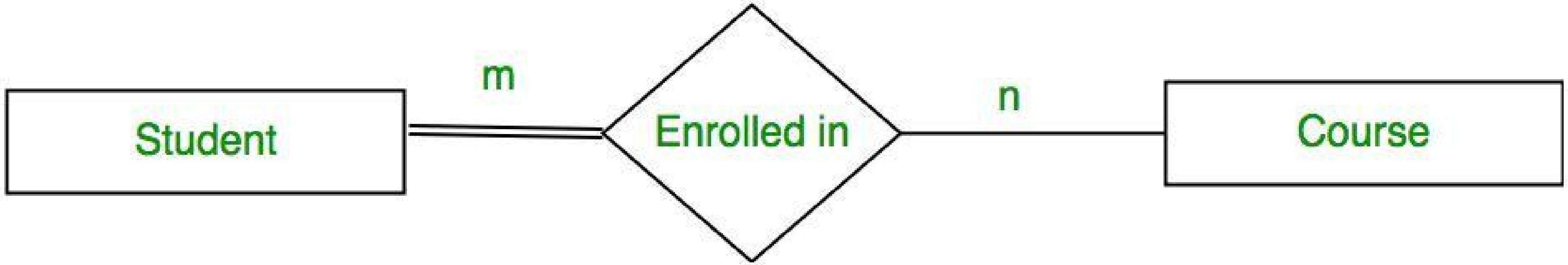
Participation Constraints

- 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. It is shown by “single line”.

- The diagram depicts the ‘Enrolled in’ relationship set with STUDENT Entity set having total participation and COURSE Entity set having partial participation.



The Enhanced ER Model

- As the complexity of data increased in the late 1980s, it became more and more difficult to use the traditional ER Model for database modelling.

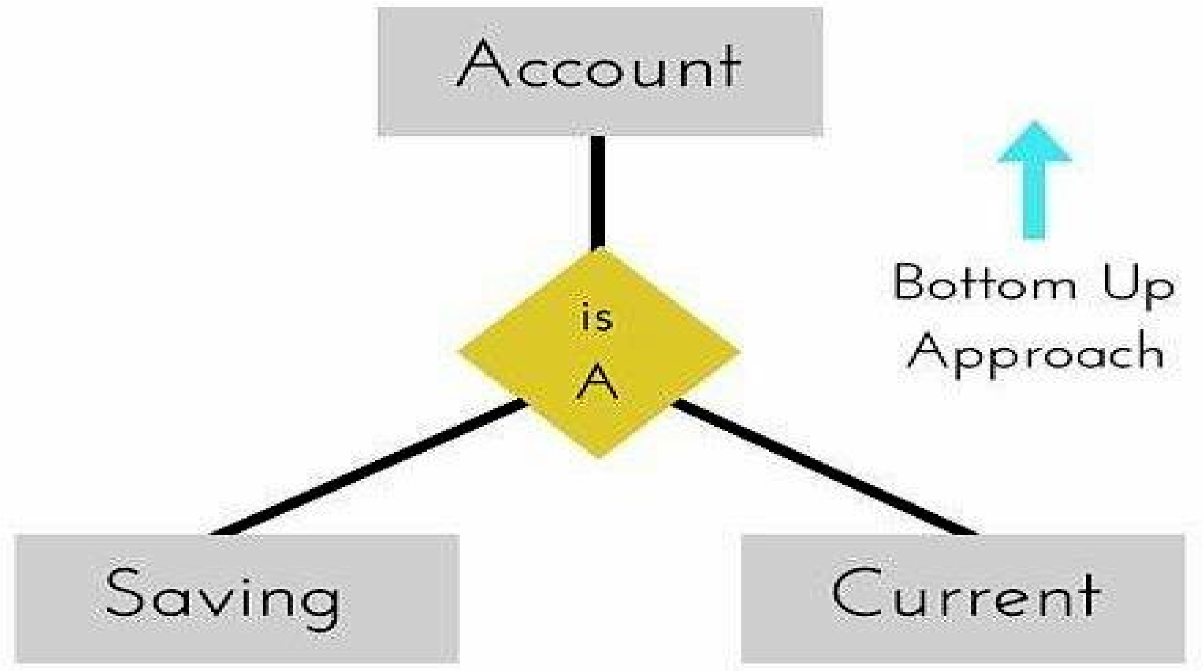
- Hence some improvements or enhancements were made to the existing ER Model to make it able to handle the complex applications better.

- Hence, as part of the Enhanced ER Model, along with other improvements, three new concepts were added to the existing ER Model, they were:

  - *Generalization*
  - *Specialization*
  - *Aggregration*

Generalization

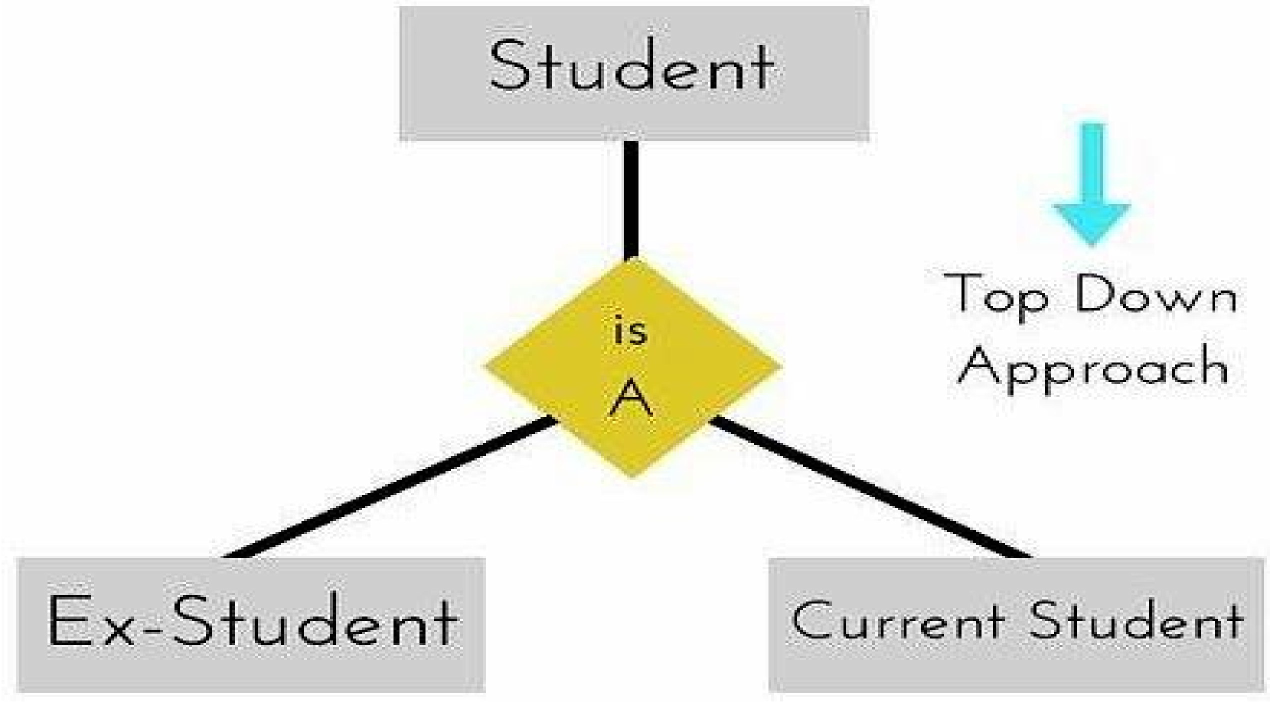
- Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalization, the higher level entity can also combine with other lower level entities to make further higher level entity.
- It's more like Superclass and Subclass system, but the only difference is the approach, which is bottom-up.
- Hence, entities are combined to form a more generalised entity, in other words, sub-classes are combined to form a super-class.



For example, Saving and Current account types entities can be generalised and an entity with name Account can be created, which covers both.

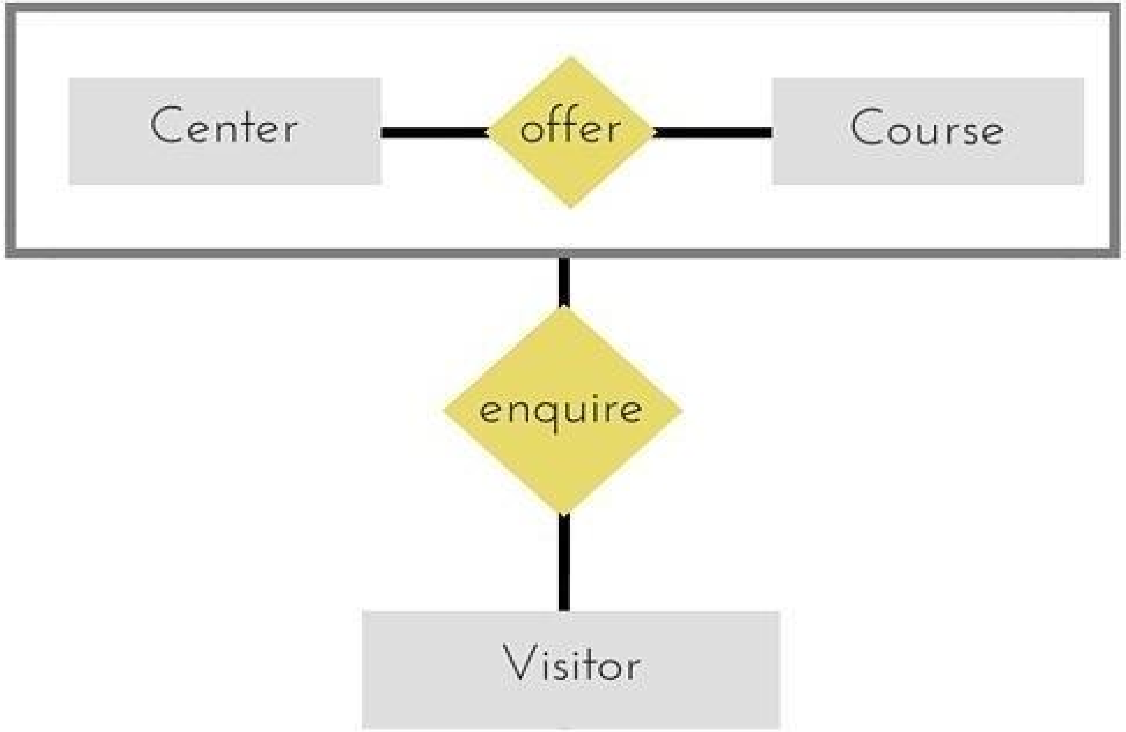
Specialization

- Specialization is opposite to Generalization.
- It is a top-down approach in which one higher level entity can be broken down into two lower level entity. In specialization, a higher level entity may not have any lower-level entity sets, it's possible.



Aggregration

- Aggregration is a process when relation between two entities is treated as a single entity.



In the diagram above, the relationship between Center and Course together, is acting as an Entity, which is in relationship with another entity Visitor. Now in real world, if a Visitor or a Student visits a Coaching Center, he/she will never enquire about the center only or just about the course, rather he/she will ask enquire about both.

Keys in DBMS

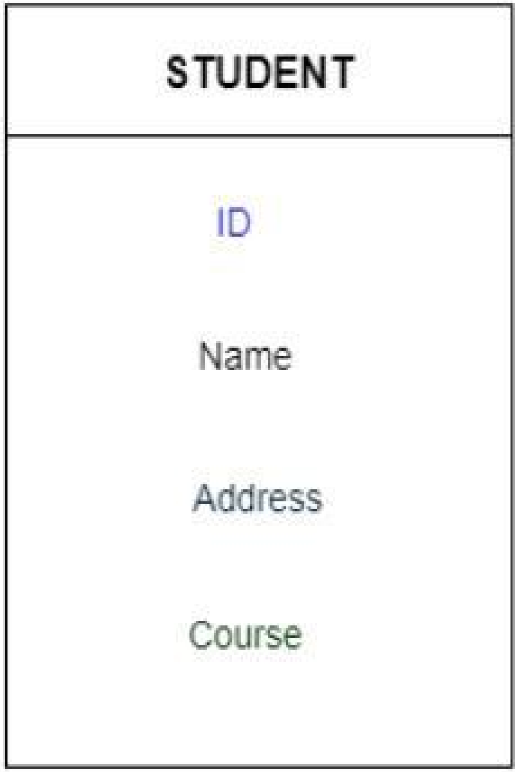
- Keys play an important role in the relational database.

- It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationships between tables.

- For example:

In Student table, ID is used as a key because it is unique for each student.

In PERSON table, passport_number, license_number, SSN are keys since they are unique for each person.



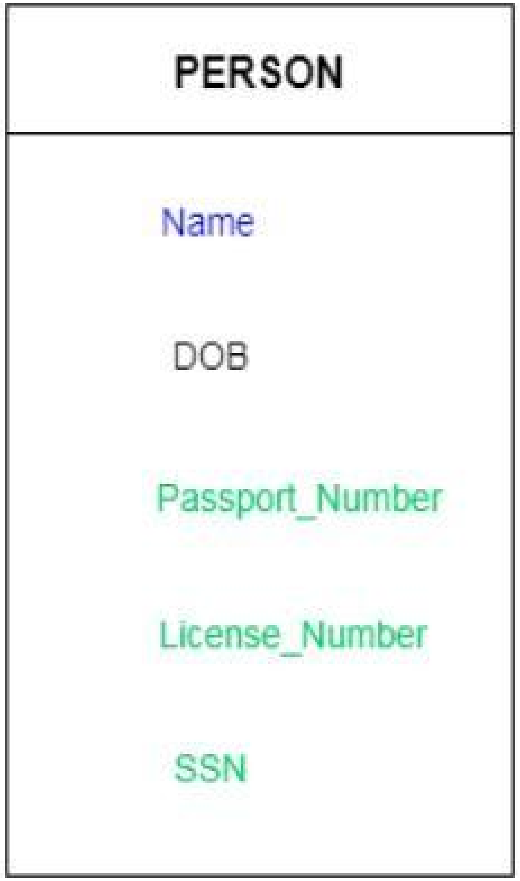
STUDENT

ID

Name

Address

Course



PERSON

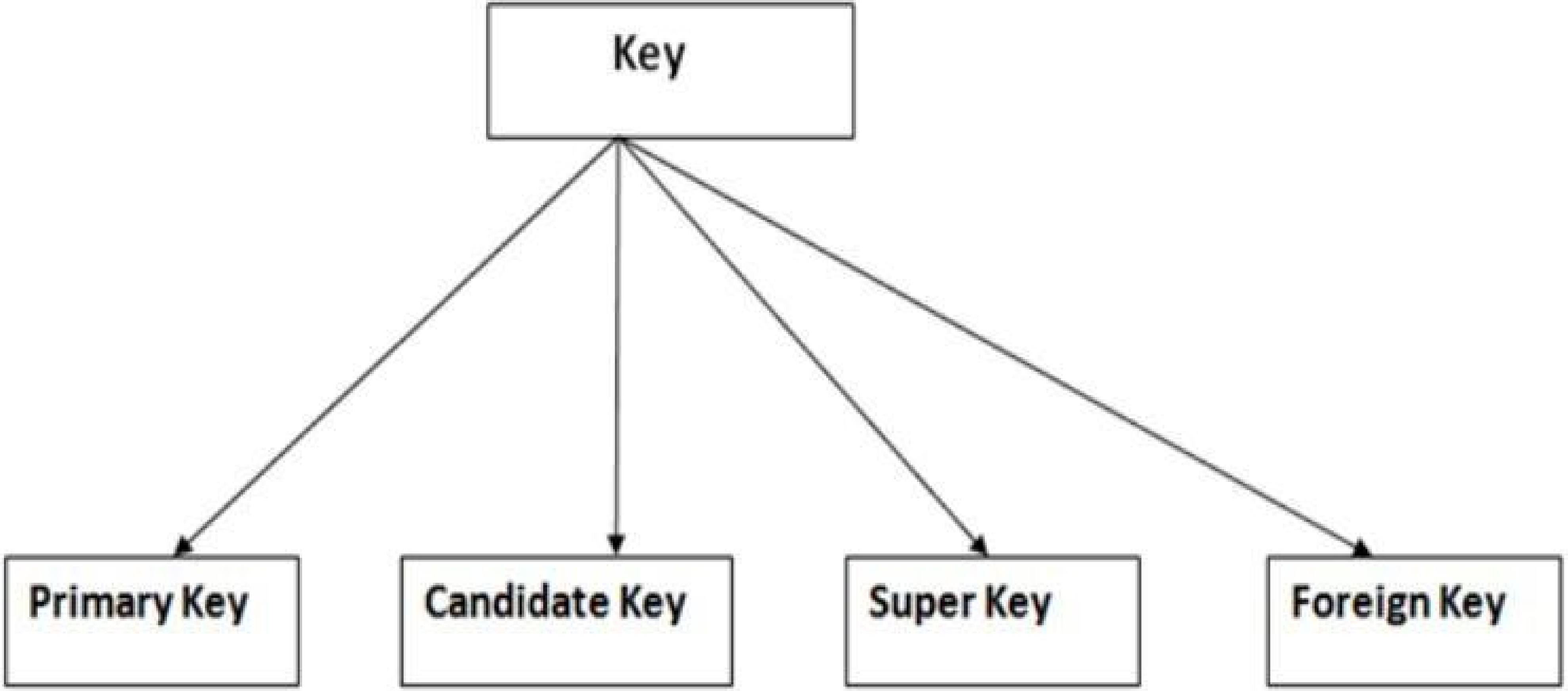
Name

DOB

Passport_Number

License_Number

SSN



Key

Primary Key

Candidate Key

Super Key

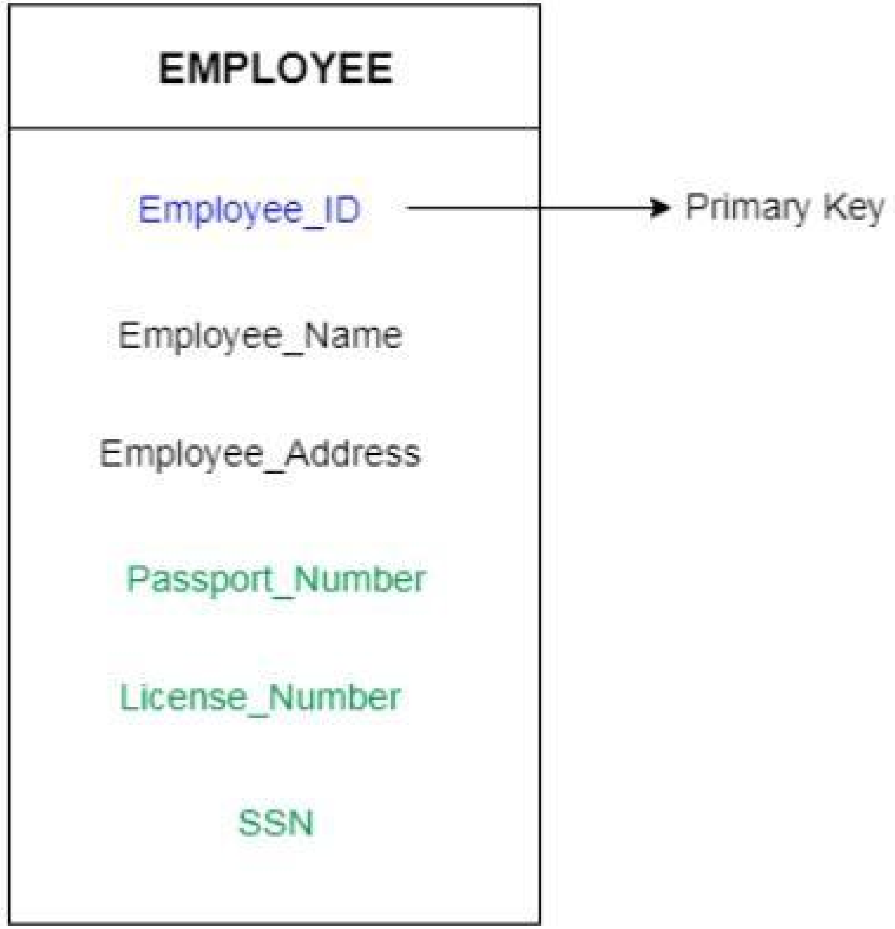
Foreign Key

Primary key

- ❑It is the key which is used to identify one and only one instance of an entity uniquely. An entity can contain multiple keys as we saw in PERSON table. The key which is most suitable from those lists become a primary key.

- ❑In the EMPLOYEE table, ID can be primary key since it is unique for each employee. In the EMPLOYEE table, we can even select License_Number and Passport_Number as primary key since they are also unique.

- ❑For each entity, selection of the primary key is based on requirement and developers.



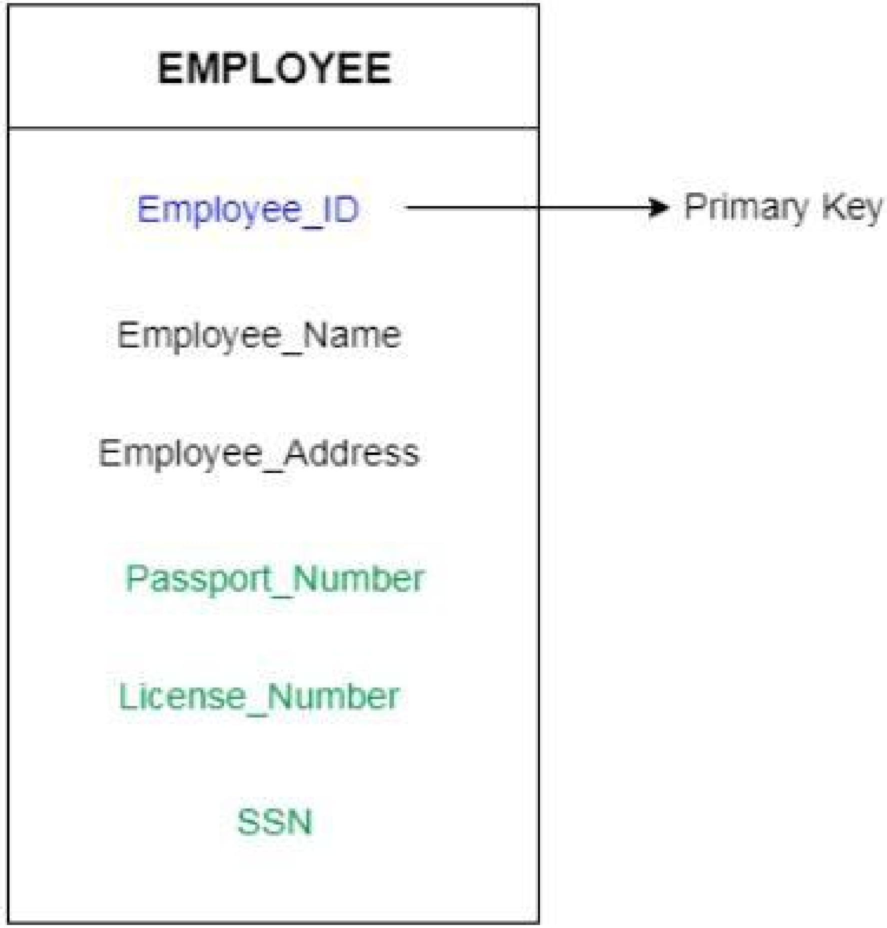
Candidate key

- A candidate key is an attribute or set of an attribute which can uniquely identify a tuple.

- The remaining attributes except for primary key are considered as a candidate key. The candidate keys are as strong as the primary key.

- For example:

In the EMPLOYEE table, id is best suited for the primary key. Rest of the attributes like SSN, Passport_Number, and License_Number, etc. are considered as a candidate key.



EMPLOYEE

Employee_ID → Primary Key

Employee_Name

Employee_Address

Passport_Number

License_Number

SSN

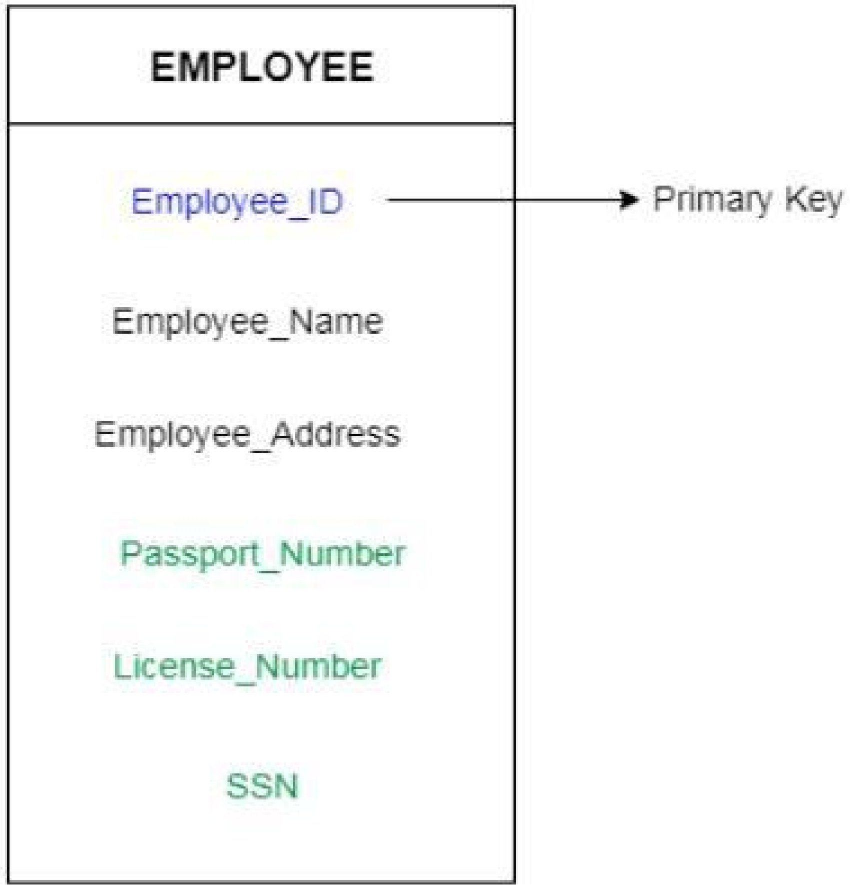
Super key

- Super key is a set of an attribute which can uniquely identify a tuple. Super key is a superset of a candidate key.

- For example:

In the above EMPLOYEE table, for(EMPLOEE_ID, EMPLOYEE_NAME) the name of two employees can be the same, but their EMPLYEE_ID can't be the same. Hence, this combination can also be a key.

- The super key would be EMPLOYEE-ID, (EMPLOYEE_ID, EMPLOYEE- NAME), etc.



EMPLOYEE

Employee_ID

Primary Key

Employee_Name

Employee_Address

Passport_Number

License_Number

SSN

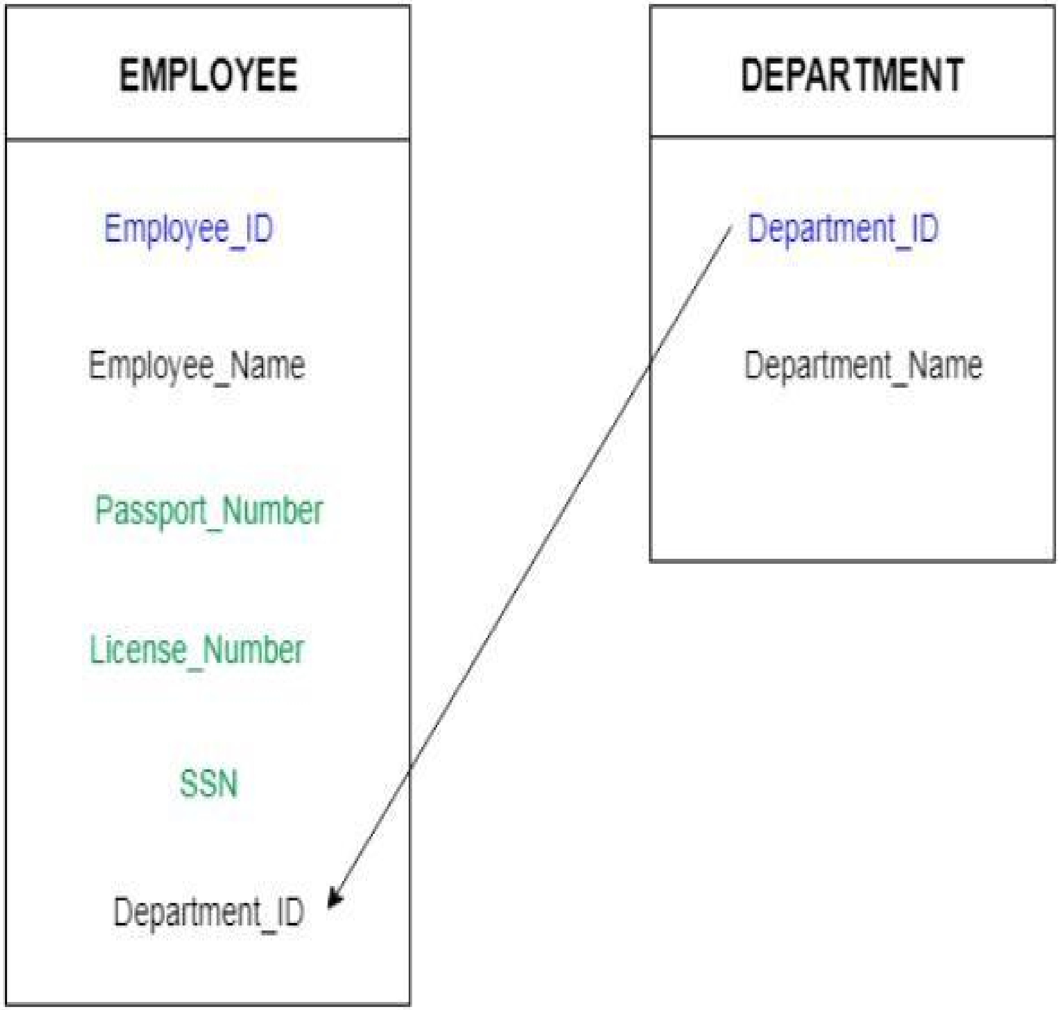
Foreign key

- ❑ Foreign keys are the column of the table which is used to point to the primary key of another table.

- ❑ In a company, every employee works in a specific department, and employee and department are two different entities. So we can't store the information of the department in the employee table. That's why we link these two tables through the primary key of one table.

- ❑ We add the primary key of the DEPARTMENT table, Department_Id as a new attribute in the EMPLOYEE table.

- ❑ Now in the EMPLOYEE table, Department_Id is the foreign key, and both the tables are related.


EMPLOYEE
Employee_ID
Employee_Name
Passport_Number
License_Number
SSN
Department_ID
DEPARTMENT
Department_ID
Department_Name