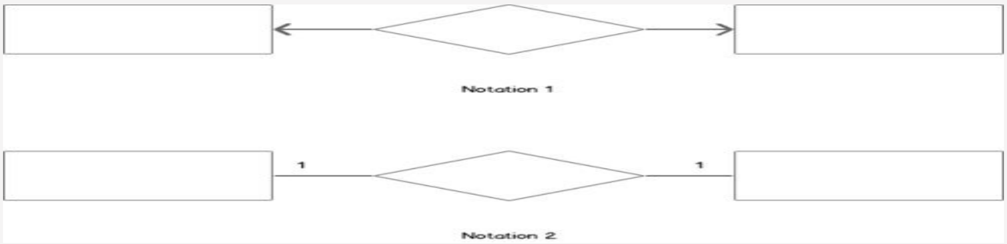
Mapping Cardinalities

Mapping cardinality or cardinality ratios express the number of entities to which another entity can be associated via a relationship set. They are of the following types:

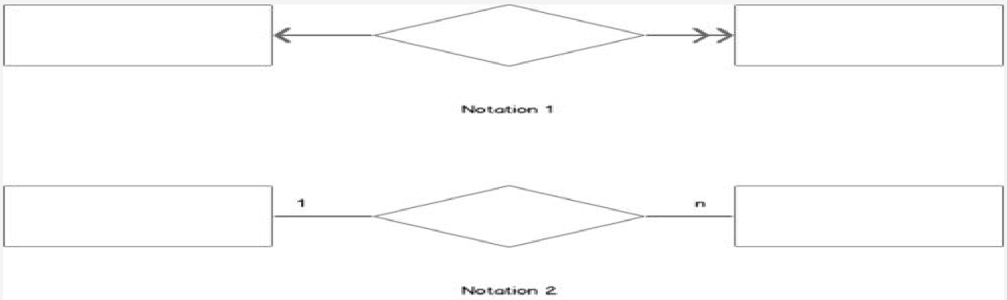
i. One-to-One:

An entity in A is associated with at-most one entity in B and vice-versa.



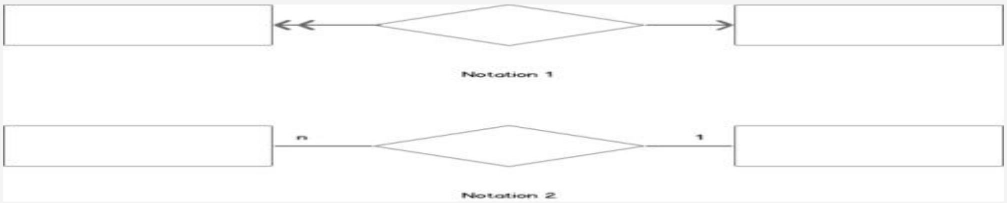
ii. One-to-Many:

An entity in A is associated with any number of entities in B but an entity in B can only be associated to at-most one entity in A.



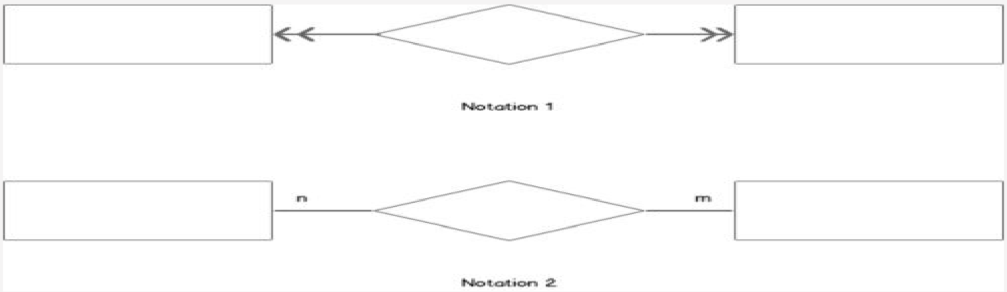
iii. Many-to-One:

An entity in A is at-most associated to one entity in B but an entity in B is associated to any number of entities in A.



iv. Many-to-Many:

An entity in A is associated to any number of entities in B and vice-versa.



b. Participation Constraint:

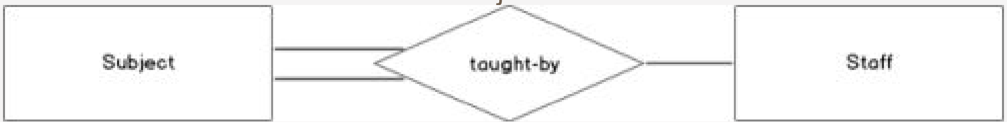
Participation constraint decides whether all the entities in the entity set take part in the relationship or not. They are of 2 types:

i. Total Participation:

If all the entities in the entity set take part i the relationship i.e. the relationship holds true for all the entities then its called Total Participation. its denoted by a double line from Entity to the relationship.

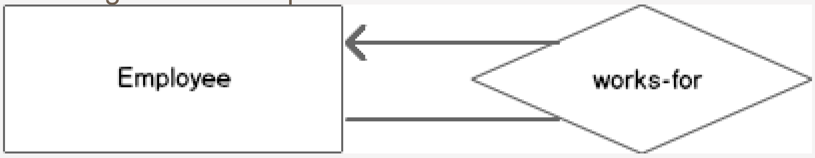
ii. Partial Participation:
If all the entities do not take part in the participation i.e. the relationship does not hold true for all the entities then its called Partial Participation.Its denoted by a single line from Entity to the relationship.

Example : We can give a common example wherein a **Subject** is **taught-by** a **Staff** wherein all the subjects have to be taught by some staff but not all staff like the peon or the administrative staff do not teach a subject.

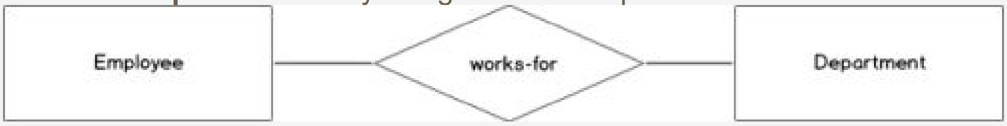


There are 5 types of relationships :

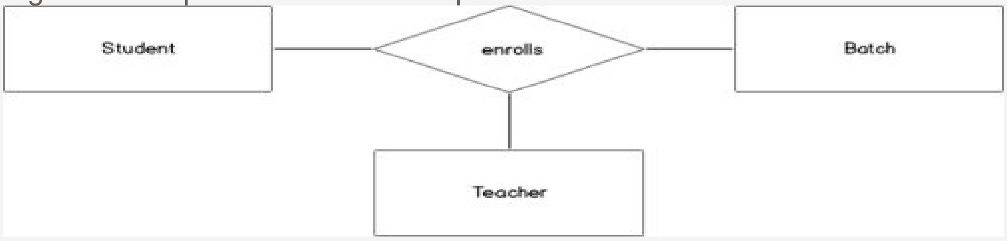
i. Unary Relationship:
Relationship which associates to an entity of the same type.Example **:** A person belonging to the **Employee** entity set is the manager for some other person of the same **Employee** entity set using a relationship **works-for.**



ii. Binary Relationship:
Relationship which associates one entity to another entity. Example : **Employee** entity is related to **Department** entity using a relationship **works-for.**



iii. Ternary Relationship:
Relationship in which 3 entities are associated to each other using a single relationship and all the 3 entities are a must for the relationship to exist. Example: **Student enrolls** for a **Batch** which has a batch T**eacher,** where Student, Batch and Teacher are 3 entities which together take part in the relationship enrolls.



iv. Quaternary Relationship:
Here the 4 entities take part in the relationship and together have to co-exist to make the relationship possible. Example : **Student studies** for a **Subject** using a particular **Course-Material** under a particular T**eacher.**

v. N-ary Relationship:
Here n-entities take part in a relationship simultaneously to make the relationship to exist.

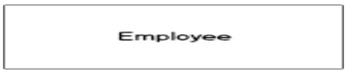
Description: In designing a database schema for a particular application we must avoid two pitfalls :

- **Redundancy**: A bad design may repeat information which leads many problems in updating records , alteration of records, deletion of records, etc.
- **Incompleteness**: A bad design may make certain aspects of the enterprise difficult or impossible to model like in a bank database application, giving loan to a person whose account does nor exist in the same bank.

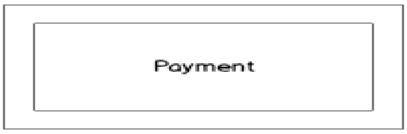
Therefore the most commonly used modelling technique for building a good database design is **Entity Relational Model (ER Model)**.

ER Model:

1. Entity Sets: An **entity** is a thing or object in the real world that is distinguishable from all the other objects. Example : Student, Staff, Subject , etc. It may have a physical or logical existence. Entities of the same type that share the same properties are called **Entity Sets.** Example is teachers, peons, administrative staff, etc would be clubbed under the same entity called Employee. Its denoted by a rectangle in an ER diagram.

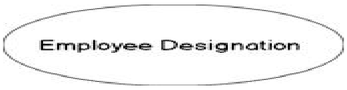


There can be a scenario wherein an entity is distinguishable but is dependent on another entity for it to exist, such an entity is called **Weak Entity**. Example : Database of a bank having **Loan** as an entity having attributes like loan number, loan amount, loan duration, etc and **Payment** as another entity having attributes like payment amount, receipt number , etc . Now **Payment** is a weak entity because payment has no meaning unless it being done for a **Loan**. Its denoted by coinciding rectangles in an ER diagram.

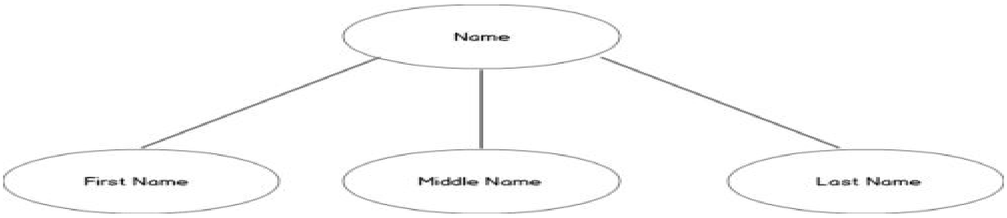


2. Attributes: An entity has some properties that are descriptive in nature are called **Attributes.** Example of attributes of Employee entity set are : employee id, full name of the employee, date of birth, telephone number, address , etc.

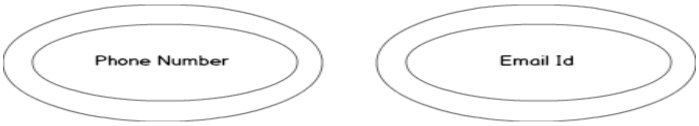
Types of attributes are: a. **Simple and Single-Valued**: It is an attribute consisting of only one field. Example : employee designation. Its denoted by an ellipse.



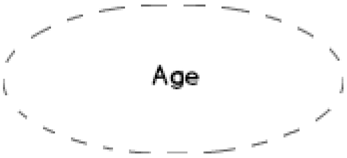
b. **Composite**: It is an attribute consisting of multiple fields. Example: name consisting of first name, middle name and last name as well as address consisting of building name, flat number, landmark, city, state , etc. Its denoted by many ellipses containing individual field names joined to a single ellipse containing the composite attribute.



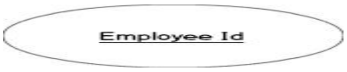
c. **Multivalued**:It is an attribute that has multiple values. Example : phone number and email ids can have multiple values for a person.Its denoted by coinciding ellipses.



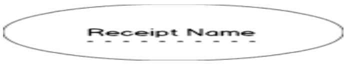
d. **Derived**: It is an attribute that is not physically stored in the database can be extracted from existing attributes. Example : age can be extracted from the difference between date of birth and current date. Its denoted by dotted ellipses.



e. **Primary Attribute**: For an entity to exist in a database it should have an attribute that unique to it, such an attribute id called primary attribute. Example : employee id, account number, etc. Its denoted by an ellipse with an underline underneath the attribute name.

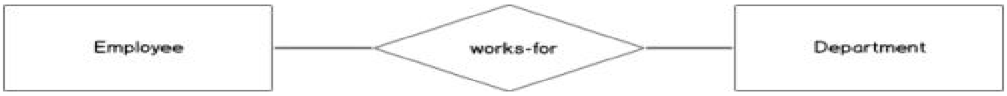


f. **Discriminator**: For weak entity, the attribute that uniquely identifies it is called as a discriminator. Example : Receipt number when payment is a weak entity depending on the loan entity. Its denoted by an ellipse with dotted underline underneath the attribute name.

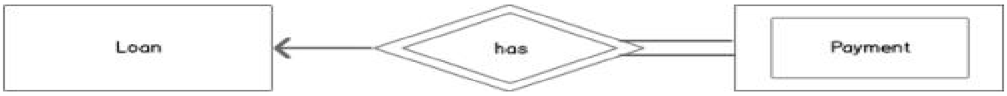


3. Relationships: A **relationship** is defined as the association among several entities and **relationship sets** is a set of relationships of the same type. Relationships are of 2 types:

a. Strong Relationship: Relationship which associates several strong entities. Example : **Employee** entity and **Department** entity are related to each other using the relationship **works-for**.



b. Weak Relationship Relationship which associates several weak entities. Example : **Payment** weak entity is related to a strong entity using a weak relationship **has** is related to entity **Loan.**


Loan
has
Payment