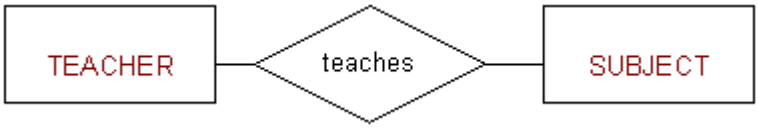
Relationships

Degree

The ***degree*** of a relationship is the number of entity types that participate in the relationship. The three most common relationships in ER models are ***Binary***, ***Unary*** and ***Ternary***

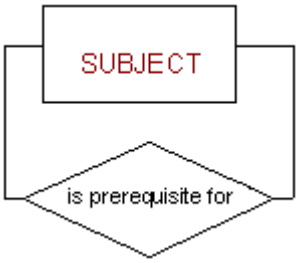
A **binary relationship** is when two entities participate and is the most common relationship degree.

For Example:



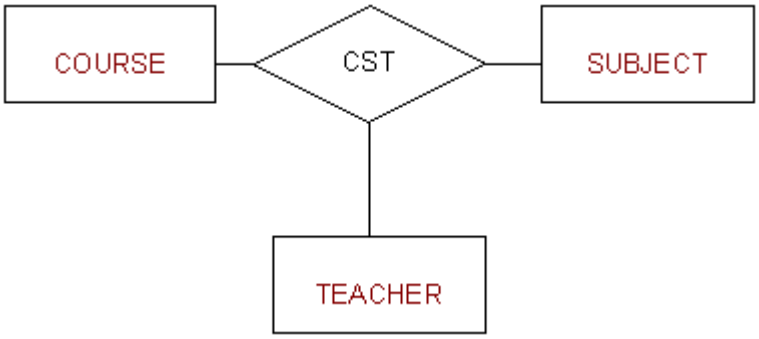
A **unary relationship** is when both participants in the relationship are the same entity.

For Example:
Subjects may be prerequisites for other subjects.



A **ternary relationship** is when three entities participate in the relationship.

For Example:
The University might need to record which teachers taught which subjects in which courses.



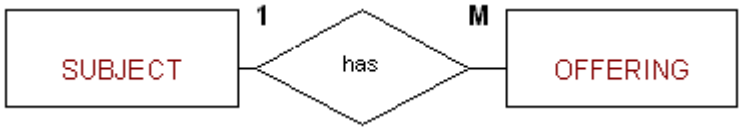
^ top

Connectivity

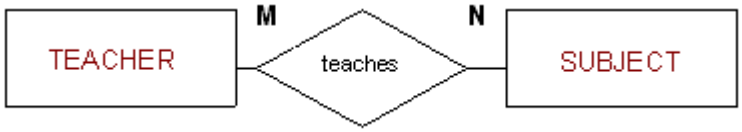
The ***connectivity*** of a relationship is its classification. It may be a **one to one (1:1)**, **one to many (1:M)** or **many to many (M:N)** relationship. A relationships connectivity is represented by a **1**, **M** or **N** next to the related entity.

one to one (1:1)
A Principal Teacher manages one Department
Each Department is managed by one Principal Teacher

one to many (1:M)
A Subject can be offered many times
Each Offering belongs to one Subject



many to many (M:N)
A Teacher can teach many different Subjects
Each Subject can be taught by many Teachers



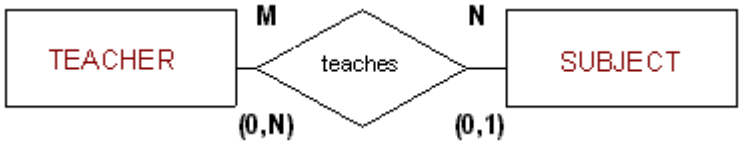
^ top

Cardinality

The ***cardinality*** of a relationship is the number of instances of entity B that can be associated with entity A. There is a minimum cardinality and a maximum cardinality for each relationship, with an **unspecified** maximum cardinality being shown as **N**. Cardinality limits are usually derived from the organisations policies or external constraints.

For Example:
At the University, each Teacher can teach an unspecified maximum number of subjects as long as his/her weekly hours do not exceed 24 (this is an external constraint set by an industrial award). Teachers may teach 0 subjects if they are involved in non teaching projects. Therefore, the cardinality limits for TEACHER are **(O,N)**.

The University's policies state that each Subject is taught by only one teacher, but it is possible to have Subjects that have not yet been assigned a teacher. Therefore, the cardinality limits for SUBJECT are **(0,1)**.



^ top

Existence Dependency

Existence dependency means that an instance of one entity cannot exist without the existence of some other related entity.

For example:
A Subject being offered at the University, must have the relevant Offering details associated with it, so OFFERING is existence-dependent on SUBJECT.

^ top

Weak Entity

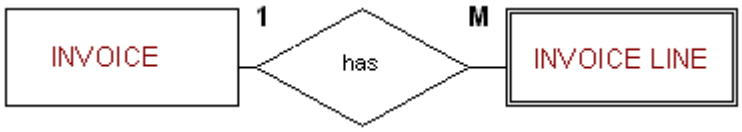
A ***weak entity*** is an entity type that, in addition to being existence dependent, has a primary key that has been totally or partially constructed from the entity it depends on.

For example:
We have established that OFFERING is existence-dependent on SUBJECT, but OFFERING has its own primary key (Offering#), which has not been derived from SUBJECT. Therefore, OFFERING is not a weak entity.

However, in the case of an Invoice that has Invoice Lines, an INVOICE LINE must be associated with an INVOICE. Therefore INVOICE LINE would be existence-dependent on INVOICE. Furthermore, part of the primary key for INVOICE LINE would contain the primary key of INVOICE, in order that it be associated with an Invoice. Therefore, INVOICE LINE is a weak entity. The primary key of a weak entity is sometimes called a *weak key*.

INVOICE (Invoice#, Date, Customer#)
INVOICE LINE (Invoice#, Part#, Quantity)

A weak entity is represented by a double lined rectangle.



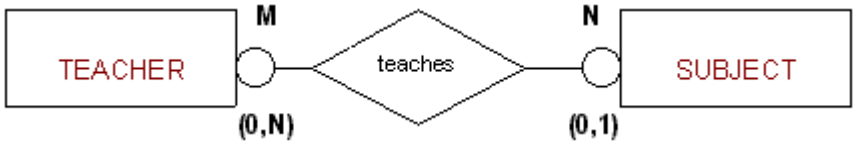
^ top

Mandatory/Optional Relationships

Participation by an entity in a relationship may be **optional** or **mandatory**.

For example:
At the University, Teachers may or may not teach a Subject (if they are involved in non teaching projects). Conversely, Subjects may or may not have a teacher currently assigned to teach it. Therefore, TEACHER is optional to SUBJECT and SUBJECT is optional to TEACHER

An optional entity is represented by a circle on the side of the optional entity.



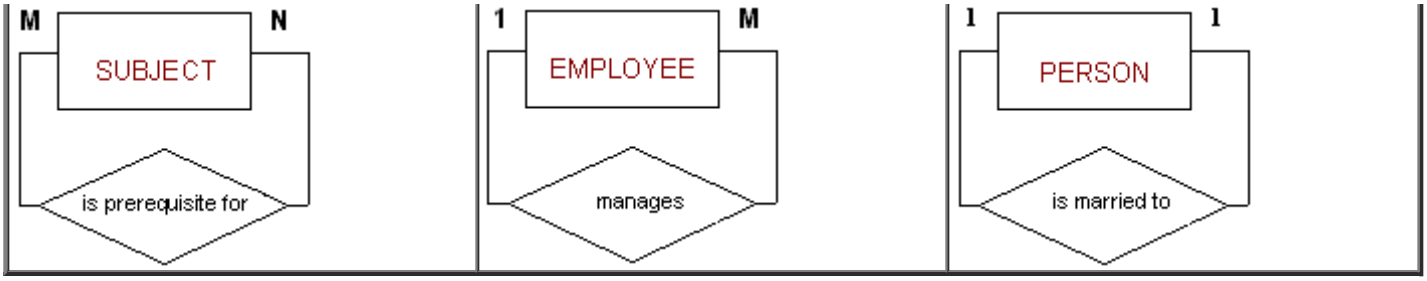
^ top

Recursive Relationships

Recursive relationships occur within unary relationships. The relationship may be one to one, one to many or many to many. That is the cardinality of the relationship is unary. The connectivity may be **1:1**, **1:M**, or **M:N**.

For example:

M:N unary relationship: A Subject may have many other Subjects as prerequisites and each Subject may be a prerequisite to many other Subjects	**1:M unary relationship:** An Employee may manage many Employees, but an Employee is managed by only one Employee.	**1:1 unary relationship:** A Person may be married to only one Person.

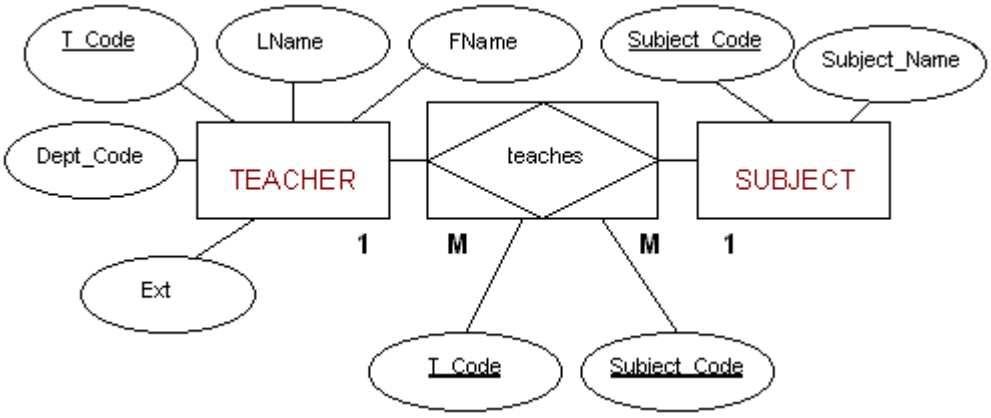


^ top

Composite Entities (Gerunds)

Many to many relationships are special cases in ER diagramming. Quite often it is difficult to decide whether you need to model a relationship or an entity. For example, in the University database, a TEACHER can teach many different SUBJECTs and a SUBJECT can be taught by many different TEACHERs. Such situations are modelled using a ***composite entity*** (or gerund), which is usually decomposed to several one to many relationships later in the modelling process. The composite entity (sometimes known as a linking table when implemented), must contain the primary keys of the associated entities, as its foreign keys. Composite entities are common in ternary relationships.

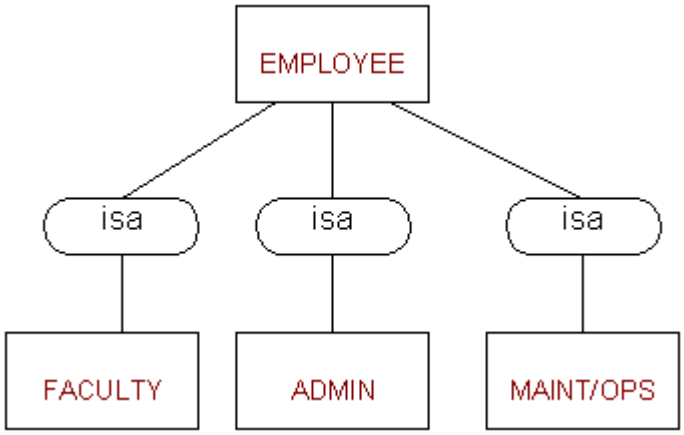
A composite entity is represented by enclosing the relationship (diamond) in a rectangle.



^ top

Generalisation

Generalisation is the concept that some entities are the ***subtypes*** of other more general entities. They are represented by an **"*is a*"** relationship. Faculty ISA subtype of employee. One method of representing subtype relationships is shown below



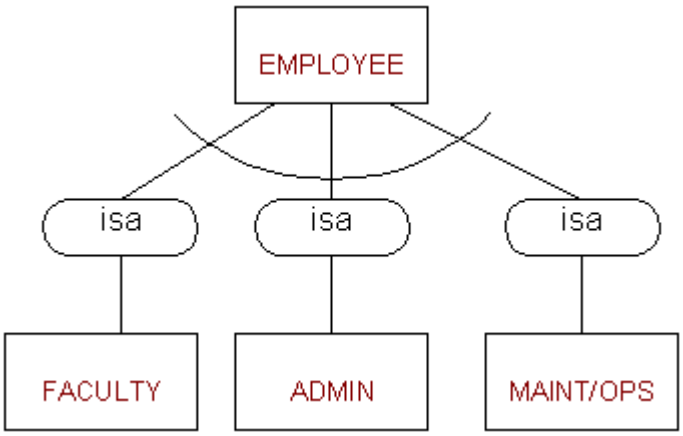
The important point to note with generalisation hierarchies is that the ***supertype*** contains all the shared values and the ***subtype*** all the specific values. So that the employee supertype contains all attributes common to employees and the subtype faculty contains only those attributes specific to faculty members.

The primary key for the supertype and the primary key for the subtype are the same. Many DBMS do not directly support generalisation relationships.

^ top

Mutually exclusive subtypes

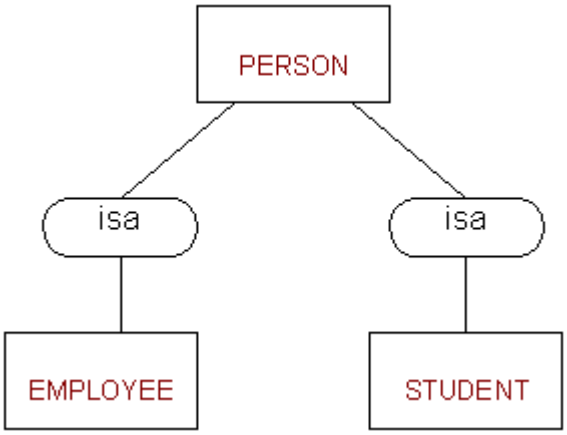
The subtypes of a supertype can be ***mutually exclusive***. Therefore each instance of the supertype is categorised as exactly one subtype. That is an employee can be categorised as a faculty member or as admin or as maintenence & operations. The employee cannot be associated with more than one subtype in this example.



^ top

Non exclusive subtypes

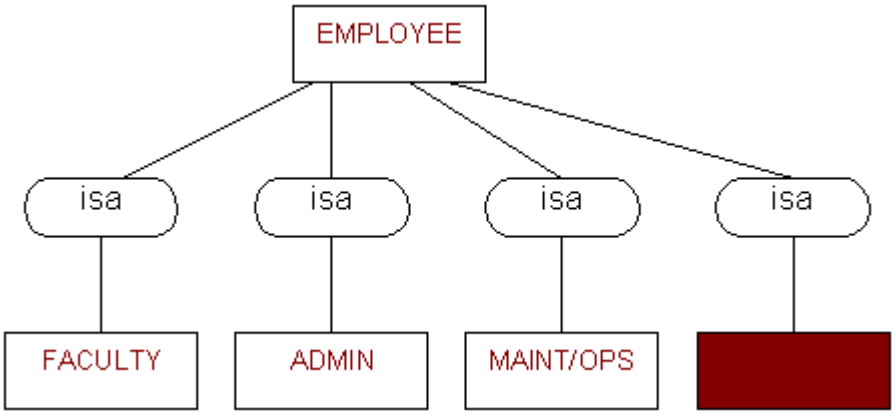
In some examples membership of subtypes may overlap. A person may be both an employee and a student. A student may be studying both graduate and undergraduate courses at the same time. These are called ***non exclusive subtypes*** and can be diagrammed as below.



^ top

Non exhaustive subtypes

Subtypes modelled may not represent all the possible subtypes. These are called ***non exhaustive subtypes*** and are represented as below.



Now look at the University's completed ERD with the additional components discussed in this section.

^ top