Entity Relationship Model (ERM):

As earlier we have discussed about the unified modeling language and we have also discussed about the types of the UML diagrams. Entity relationship diagram is one of them, which is proposed by Peter Chen in 1976. In this lecture we will discuss about entity relationship model and diagram in detail. First we will discuss about the entity relationship data model.

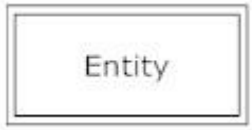
In modeling which is the fundamental concept in software engineering, ER data model is used for describing the data or information related to software product in an abstract way which is ultimately being implemented in the database, such database is called rational database. ER Model is the top down approach to database design. So, ER data model graphically represents the conceptual database or database design of the system. This conceptual design is independent of implementation the user can implement database in any DBMS using ER data model. In other words, ER model don't use any type of DBMS so we have to convert that conceptual database design into actual database. The main components of ER model are entities and relationship among them. Similarly, an entity can have attributes also.

So, here we will discuss every component individually.

1. Entity/ Data object:

Entity is a real world item or a concept which can exist on its own. It can be a physical object or it can be a concept. The entity is represented by a rectangular shape.

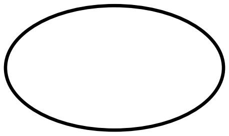
Entity can be a week entity which means that entity must be defined by a foreign key relationship with another entity and it can't be defined independently or by its attributes only. Week entity is represented as follows:



Entity is further divided into entity type and entity set. Entity type in a database is defined by its name and attribute, which defines the collection of entities having similar attributes. However, single occurrence of an entity type is known as entity instance. Moreover, entity set is defined by collection of entities of a particular entity type in a database at any point of time. It is usually mentioned by the same name as an entity type.

2. Attribute:

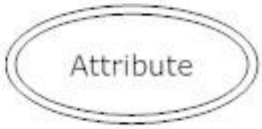
These are the properties that describe the entities. It is represented by oval shape. While the set of attribute values of a given attribute is the value set or domain.



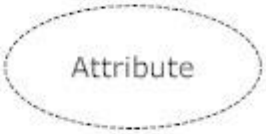
Attributes are further divided into following types:

- Simple attribute
- Composite attribute
- Single-valued attribute
- Multi-valued attribute
- Derived attribute
- Sorted attribute

A multivalued attribute mean an attribute having more than one values. Multivalues attribute is represented as follows:

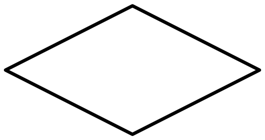


While derived attribute is based on another attribute and can be represented as :



3. Relationship

When attribute of one entity refers to the attribute of another entity then there exists a relationship between these two entities. The relationship is represented by a diamond shape.



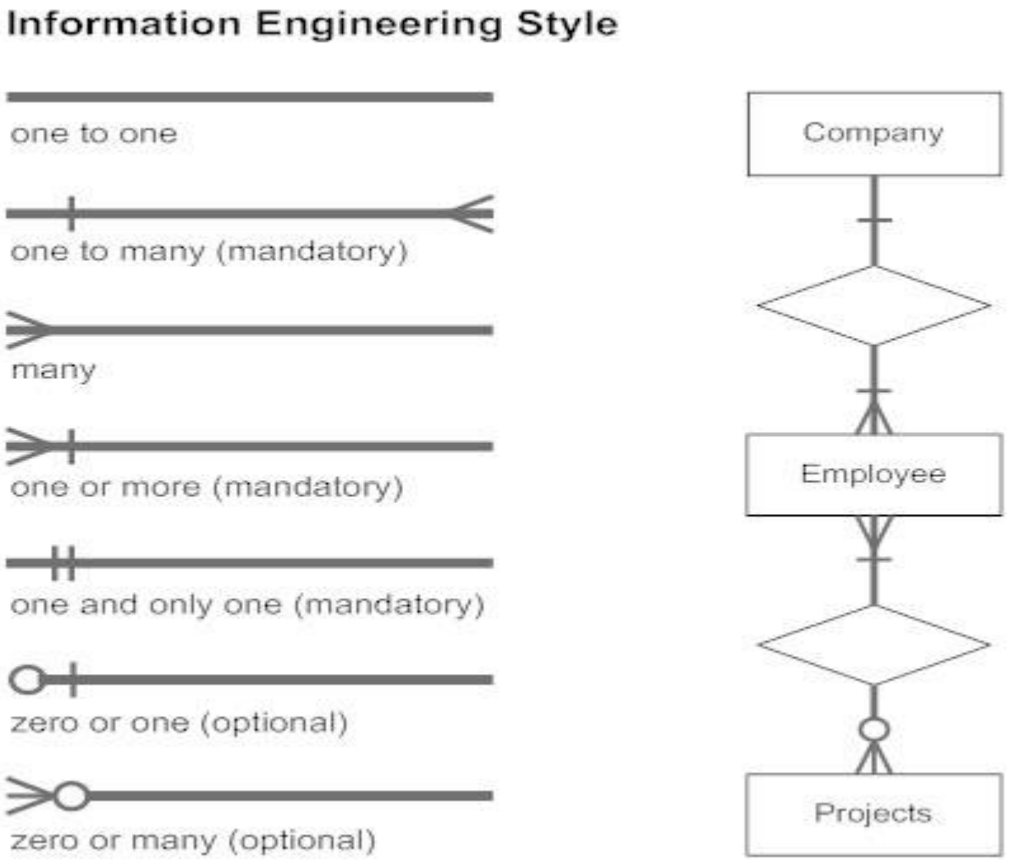
On the basis of participating entities, relationship is divided into following categories.

- Unary /recursive relationship
- Binary relationship
- Ternary relationship
- Quaternary relationship

While on the basis of cardinality relationship can be divided into following categories.

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

Relationship between entities are represented in the following ways as depicted in the diagram with the example.



There are some other ways also used to present the relationship between entities, such as Chen style and Martin style.

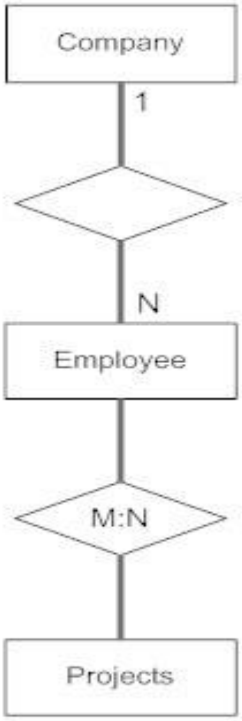
Chen Style

Ordinality - describes the minimum (optional vs mandatory) → M:N ← Cardinality - describes the maximum

1:N (n=0,1,2,3...)
one to zero or more

M:N (m and n=0,1,2,3...)
zero or more to zero or more
(many to many)

1:1
one to one



Martin Style

1 - one, and only one (mandatory)

* - many (zero or more - optional)

1...* - one or more (mandatory)

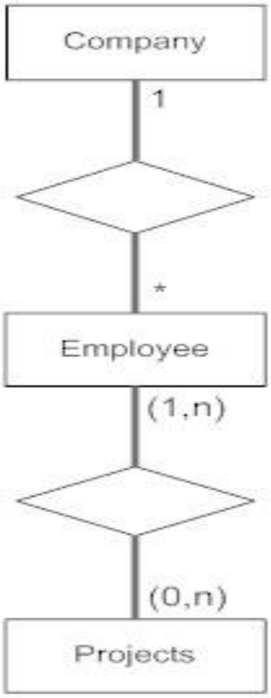
0...1 - zero or one (optional)

(0,1) - zero or one (optional)

(1,n) - one or more (mandatory)

(0,n) - zero or more (optional)

(1,1) - one and only one (mandatory)



Why to use ER Model?

The major question here is that why we need ER model or ER diagram or to say why it is important in software development life cycle. In development process, we have to store data related to the software product, for this purpose we create the database. So, to create the database or to use DBMS we must be able to represent the data in it. Relational model is used for the representation of data in the database. This model represents data in the form of tables which is more expressive way. To use the right set of tables' graphical model can be used, so at this stage ER model is used. ER modeling is the high level data modeling technique to create useful and precise conceptual models. Moreover, entity relationship diagrams are used to represent the database or conceptual model pictorially in ER model, which automatically maps the set of tables to graphical model.

Entity Relationship Diagram (ERD):

In ER model the structure of database is represented diagrammatically by using the entity relationship diagram. Basically ERD is used to conduct date modeling activity. It represents information about a system visually or in the form of graphical model. Developers use the ERD to show their understanding of user requirements; after that users evaluate it to make sure that their needs and requirements are being confronted. After approval form end-user, the ERD provides specifications for the developers that what must be completed. By usage of accurate model or ERD the developers can get confident that they are developing a useful system.

Moreover, it describes relationship between data object. DBA and system analyst used ERD to perform data modeling and explain the diagram to stakeholders. Components of ERD are data object, relationship and attributes. Data object can be a person, place, object, event or concept about which data is to be maintained. It represents a set or collection of objects in the real world that share the same properties. However, attributes are the named property or characteristics of an entity. And relationship is the association between the instances of one or more entity types.

Benefits of ERD:

1. Visual representation

The foremost and most important ERD benefit is that it provides a visual representation of the design. It is normally crucial to have an ERD if you are looking to come up with an effective database design. This is because the patterns assist the designer in focusing on the way the database will primarily work with all the data flows and interactions. It is common to the ERD being used together with data flow diagrams so as to attain a better visual representation.

2. Effective communication

An ERD clearly communicates the key entities in a certain database and their relationship with each other. ERD normally uses symbols for representing three varying kinds of information. Diamonds are used for representing the relationships, ovals are usually used for representing attributes and boxes represent the entities. This allows a designer to effectively communicate what exactly the database will be like.

3. Simple to understand

ERD is easy to understand and simple to create. In effect, this design can be used to be shown to the representatives for both approval and confirmation. The representatives can also make their contributions to the design, allowing the possibilities of rectifying and enhancing the design.

4. High flexibility

The ERD model is quite flexible to use as other relationships can be derived easily from the already existing ones. This can be done using other relational tables and mathematical formulae. The ERD thereby acts like the blueprint for the database and it allows the creation of an accurate design that reflects the needs of the project.

- Visual representation
- Effective communication
- Simple to understand
- High flexibility

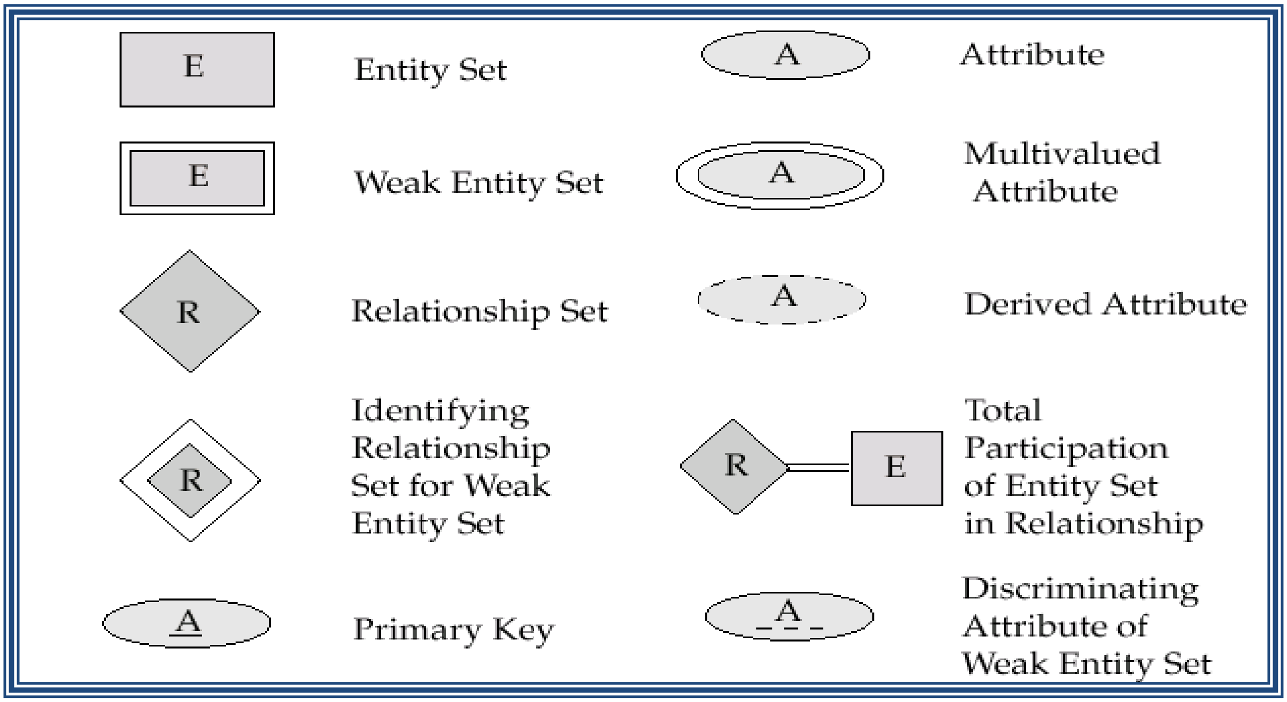
Process of Developing ERD:

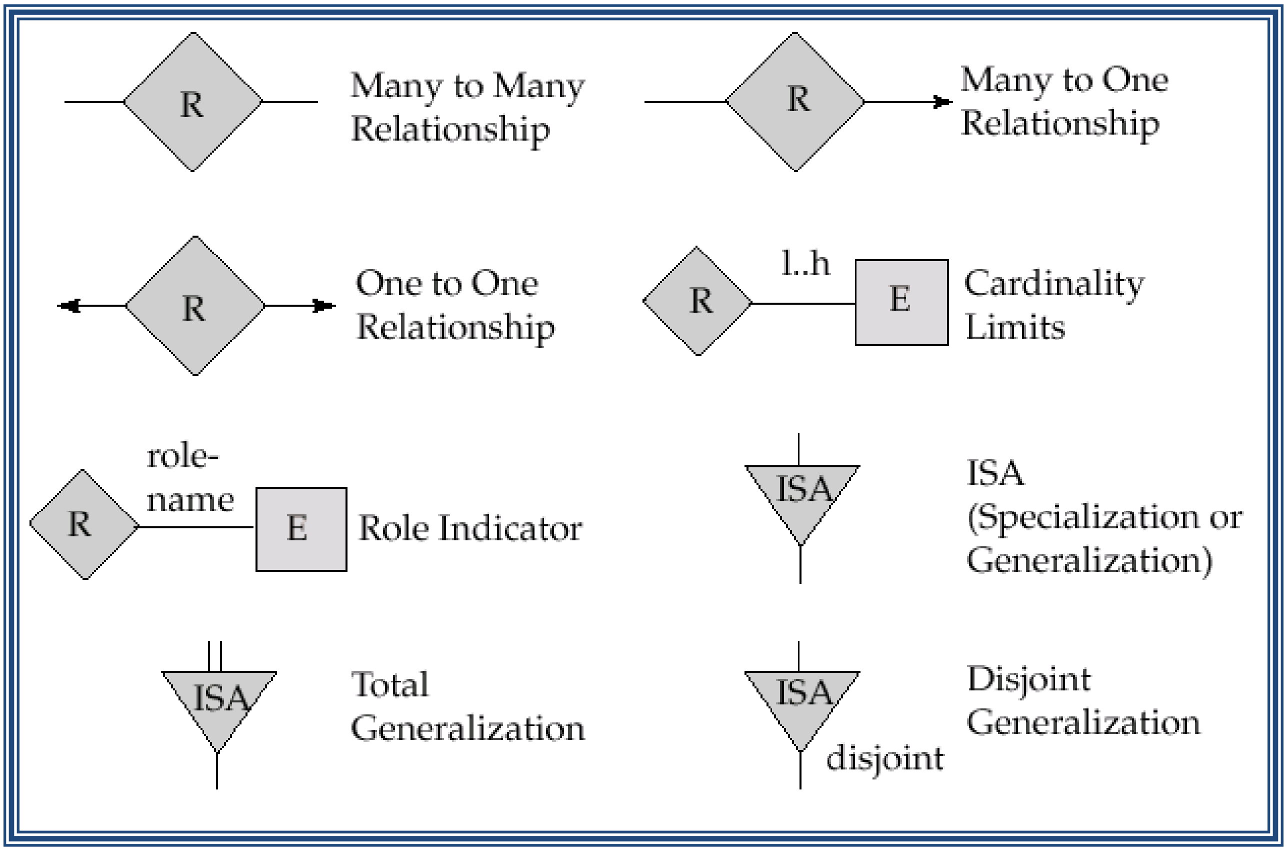
- Identify the entity
- Identify the entity's attributes
- Identify the Primary Keys
- Identify the relationship between entities
- Identify the Cardinality constraint
- Draw the ERD
- Check the ERD

Or you can go another way by following these steps.

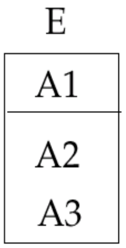
1. Identify Entities
2. Find Relationships
3. Draw Rough ERD
4. Fill in Cardinality
5. Define Primary Keys
6. Draw Key-Based ERD
7. Identify Attributes
8. Map Attributes
9. Draw fully attributed ERD
10. Check Results

Notations/ Symbols used in ERD:

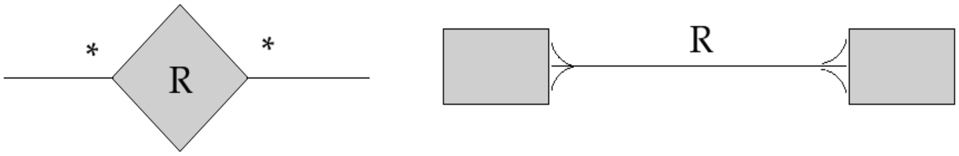

E
Entity Set
A
Attribute
E
Weak Entity Set
A
Multivalued Attribute
R
Relationship Set
A
Derived Attribute
R
Identifying Relationship Set for Weak Entity Set
R
E
Total Participation of Entity Set in Relationship
A
Primary Key
A
Discriminating Attribute of Weak Entity Set



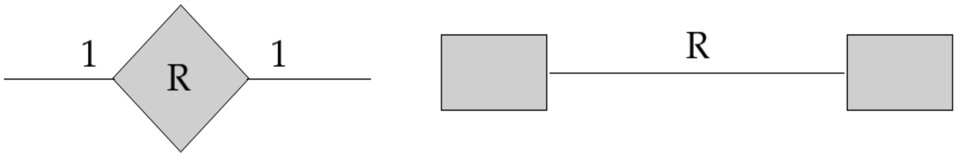
Entity set E with
attributes A1, A2, A3
and primary key A1



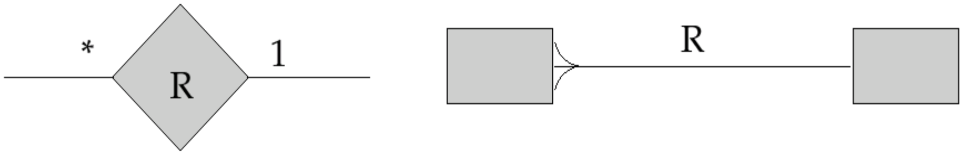
Many to Many
Relationship



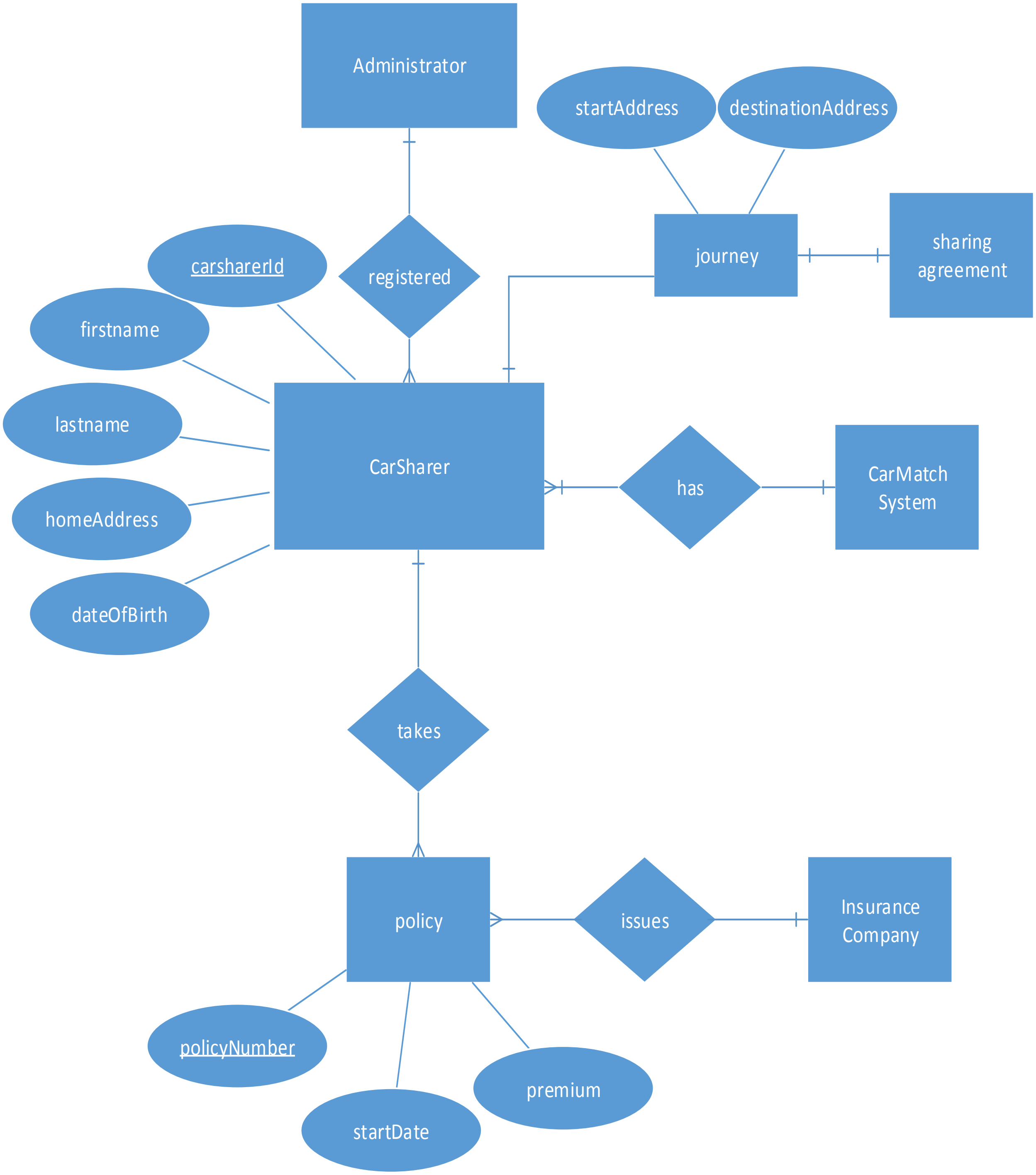
One to One
Relationship



Many to One
Relationship



CarMatch ERD:



Recommended Link:

https://www.lucidchart.com/pages/ER-diagram-symbols-and-meaning

http://creately.com/blog/diagrams/er-diagrams-tutorial/

http://www.tutorialspoint.com/dbms/dbms_generalization_aggregation.htm

Reference Links:

http://www.aw-bc.com/info/riccardi/database/Riccardi_ch4.PDF

https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model

http://www.cs.sfu.ca/CourseCentral/354/jpei/slides/ER-diagram.pdf

http://www.slideshare.net/anniyappa/entity-relationship-modelling

http://ecomputernotes.com/fundamental/what-is-a-database/advantages-and-disadvantages-of-e-r-model