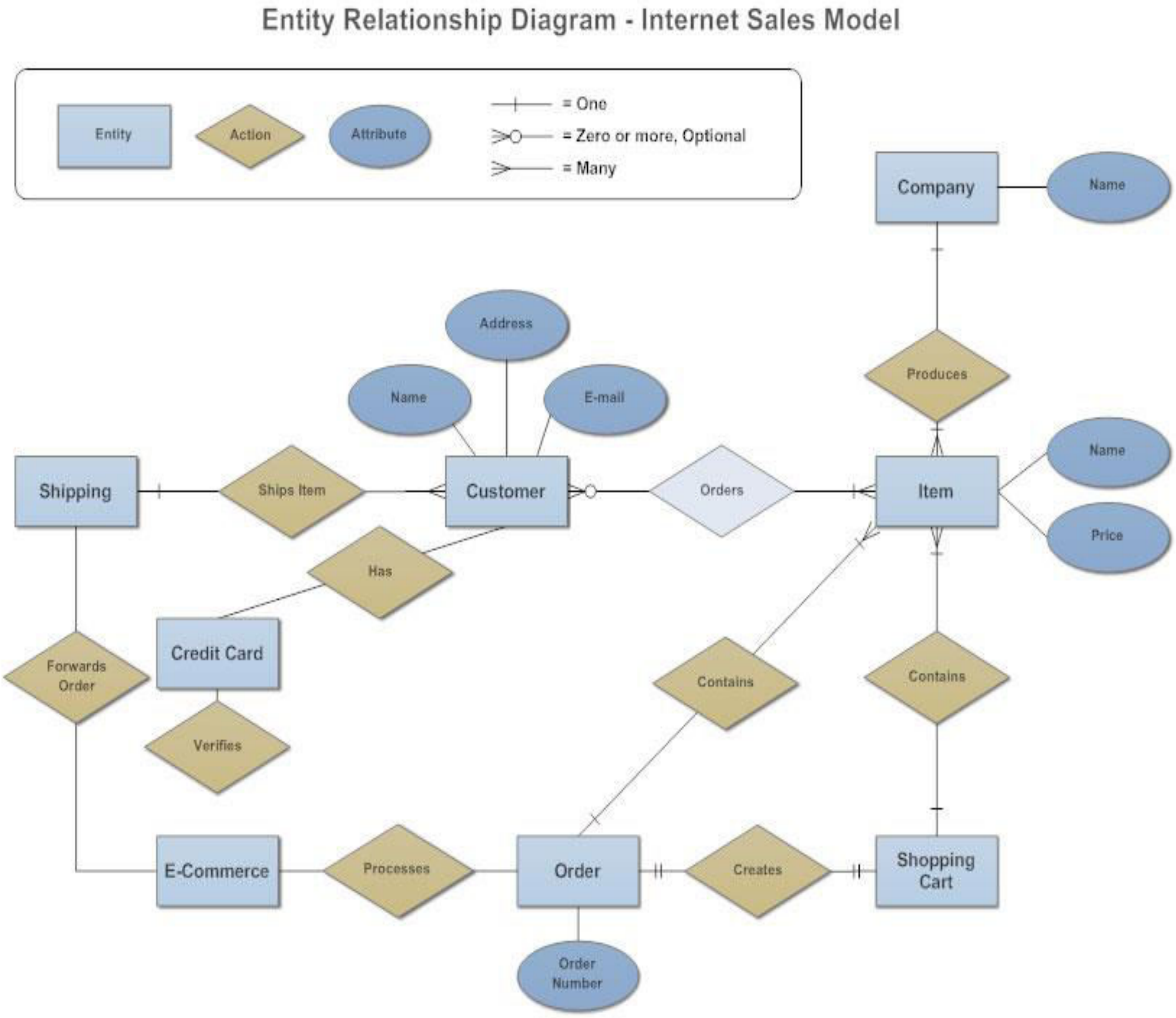
Entity Relationship Diagram



What is an Entity Relationship Diagram (ERD)?

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique.

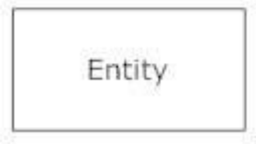
The History of Entity Relationship Diagrams

Peter Chen developed ERDs in 1976. Since then Charles Bachman and James Martin have added some slight refinements to the basic ERD principles.

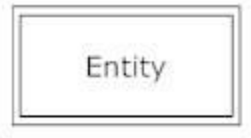
Common Entity Relationship Diagram Symbols

An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

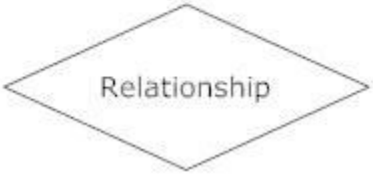
- **Entities**, which are represented by rectangles. An entity is an object or concept about which you want to store information.



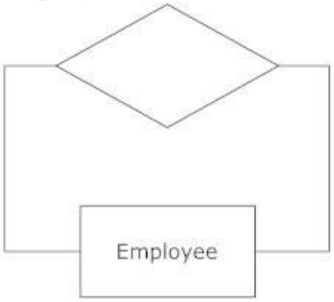
- A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.



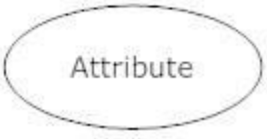
- **Actions**, which are represented by diamond shapes, show how two entities share information in the database.



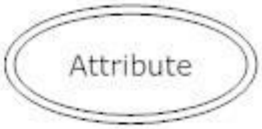
- In some cases, entities can be self-linked. For example, employees can supervise other employees.



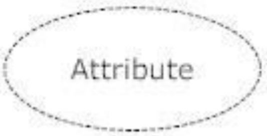
- **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.



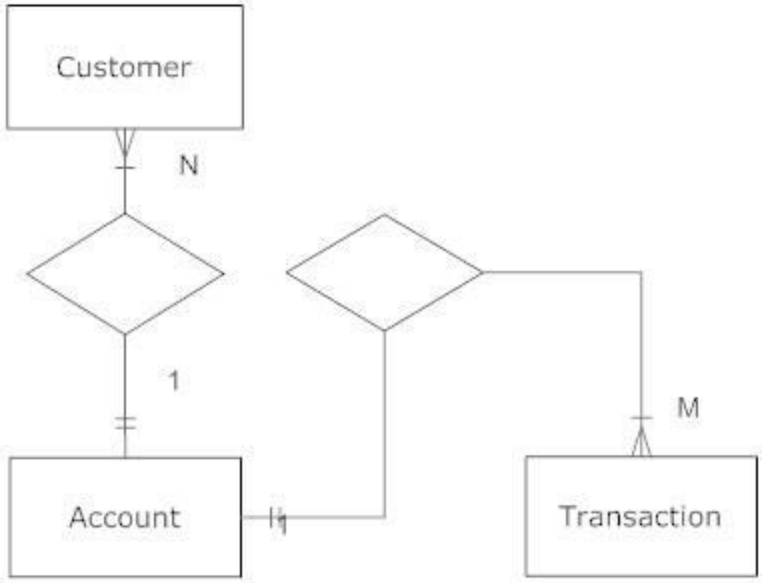
- **A multivalued** attribute can have more than one value. For example, an employee entity can have multiple skill values.



- **A derived attribute** is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

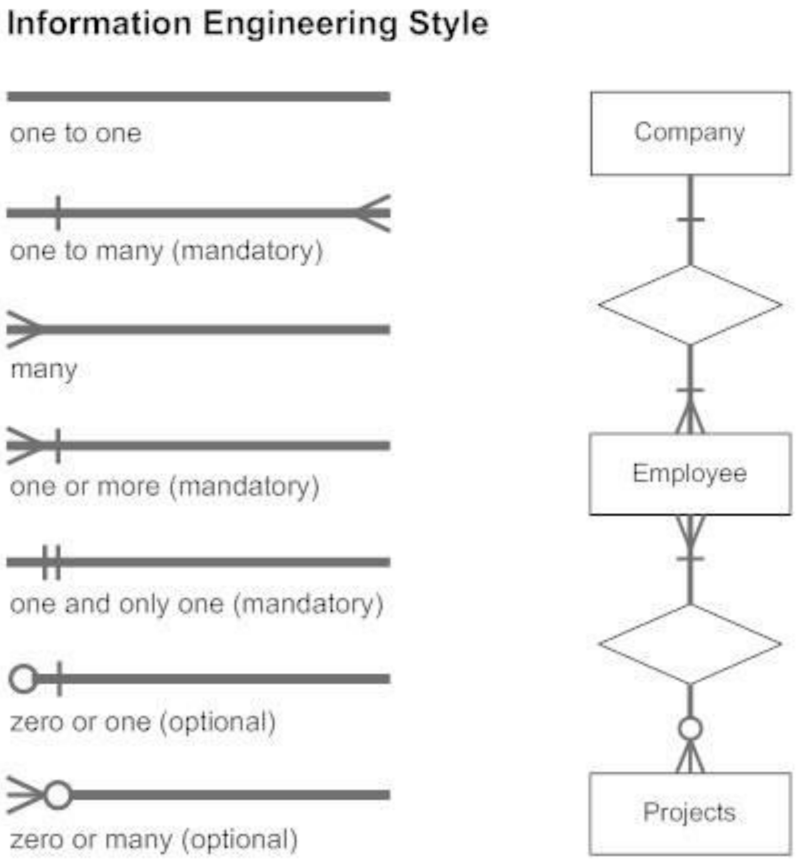


- **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.
- **Cardinality** specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinality describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinality specifies the absolute minimum number of relationships.



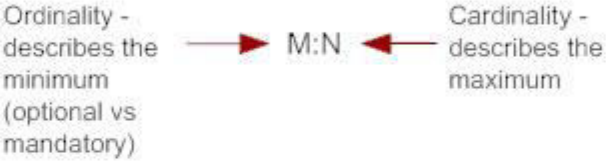
- There are many notation styles that express cardinality.

  Information Engineering Style


Information Engineering Style
one to one
one to many (mandatory)
many
one or more (mandatory)
one and only one (mandatory)
zero or one (optional)
zero or many (optional)
Company
Employee
Projects

Chen Style

Chen Style



Ordinality - describes the minimum (optional vs mandatory)

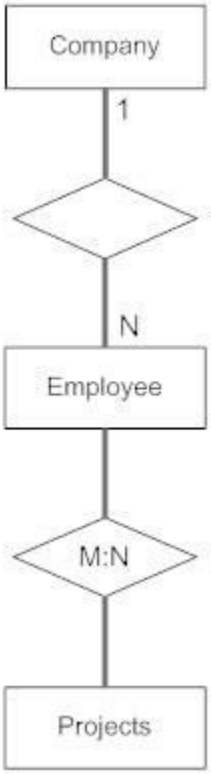
M:N

Cardinality - describes the maximum

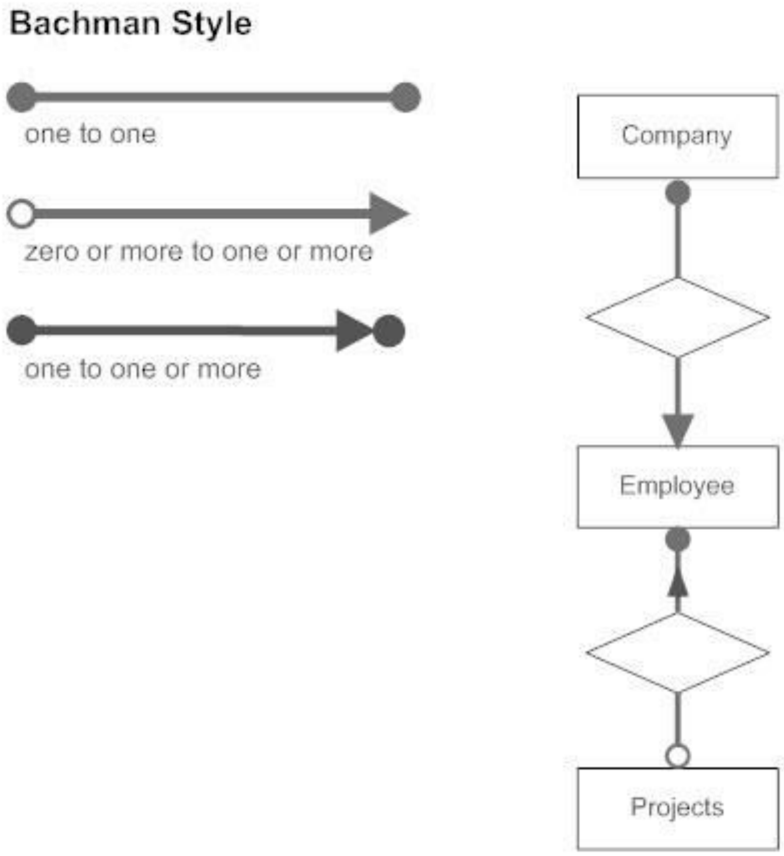
1:N (n=0,1,2,3...)
one to zero or more

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

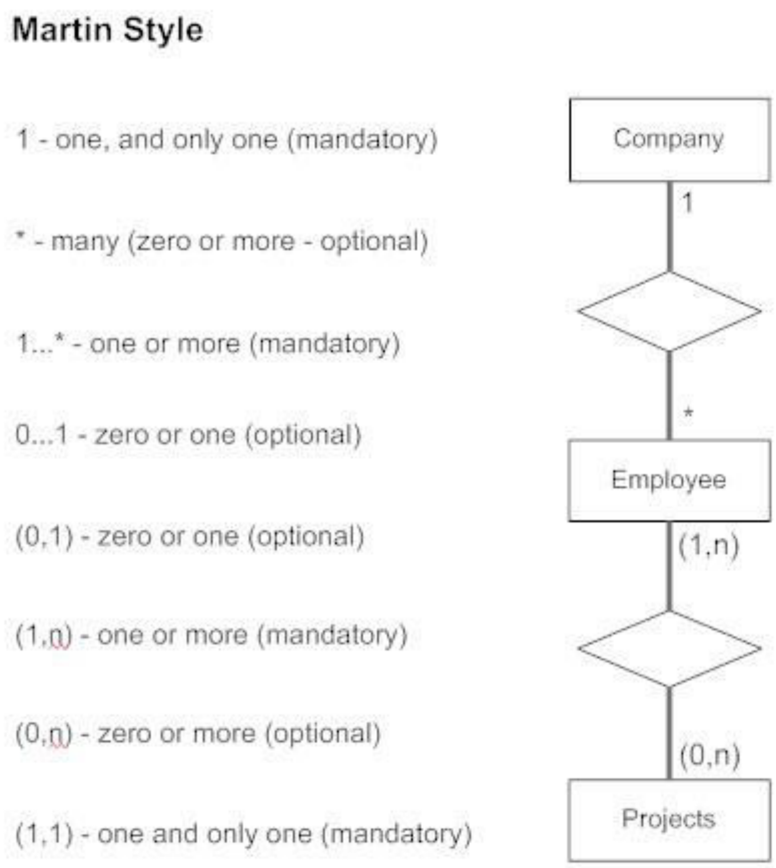
1:1
one to one



Bachman Style



Martin Style


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)
Company
1
*
Employee
(1,n)
(0,n)
Projects

ER Diagram Uses

When documenting a system or process, looking at the system in multiple ways increases the understanding of that system. ERD diagrams are commonly used in conjunction with a data flow diagram to display the contents of a data store. They help us to visualize how data is connected in a general way, and are particularly useful for constructing a relational database.

Entity Relationship Diagram Tutorial

Here are some best practice tips for constructing an ERD:

- **Identify the entities.** The first step in making an ERD is to identify all of the entities you will use. An entity is nothing more than a rectangle with a description of something that your system stores information about. This could be a customer, a manager, an invoice, a schedule, etc. Draw a rectangle for each entity you can think of on your page. Keep them spaced out a bit.

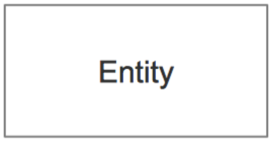
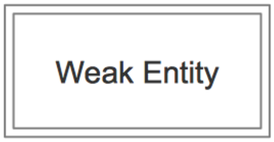
- **Identify relationships.** Look at two entities, are they related? If so draw a solid line connecting the two entities.
- **Describe the relationship.** How are the entities related? Draw an action diamond between the two entities on the line you just added. In the diamond write a brief description of how they are related.
- **Add attributes.** Any key attributes of entities should be added using oval-shaped symbols.
- **Complete the diagram.** Continue to connect the entities with lines, and adding diamonds to describe each relationship until all relationships have been described. Each of your entities may not have any relationships, some may have multiple relationships. That is okay.

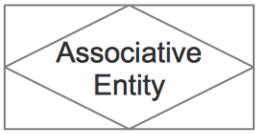
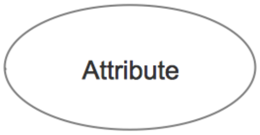
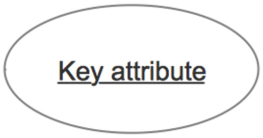
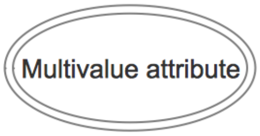
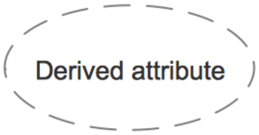
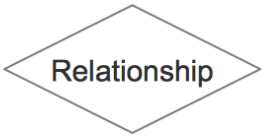
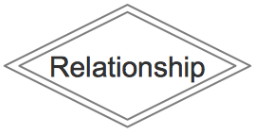
Make Your Own Entity Relationship Diagram Right Now

Tips for Effective ER Diagrams

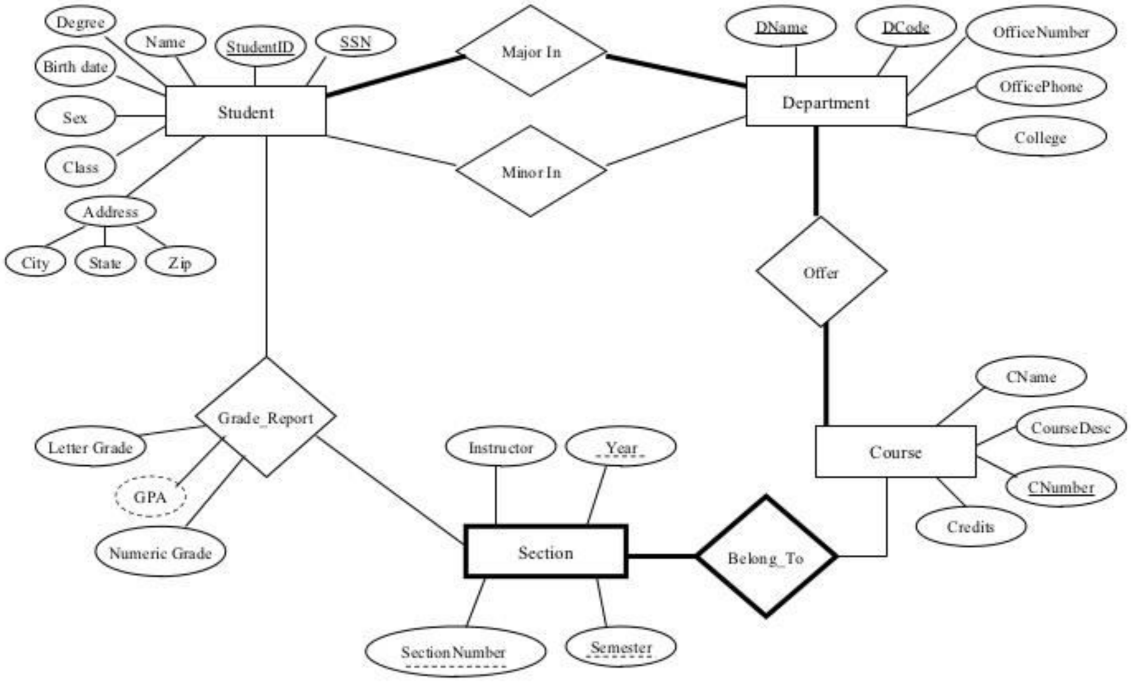
1. Make sure that each entity only appears once per diagram.
2. Name every entity, relationship, and attribute on your diagram.
3. Examine relationships between entities closely. Are they necessary? Are there any relationships missing? Eliminate any redundant relationships. Don't connect relationships to each other.
4. Use colors to highlight important portions of your diagram.

Entity Relationship Diagram Symbols — Chen notation

Symbol	**Shape Name**	**Symbol Description**
Entities		
	Entity	An entity is represented by a rectangle which contains the entity's name.
	Weak Entity	An entity that cannot be uniquely identified by its attributes alone. The existence of a weak entity is dependent upon another entity called the owner entity. The weak entity's identifier is a combination of the identifier of the owner entity and the partial key of the weak entity.

	Associative Entity	An entity used in a many-to-many relationship (represents an extra table). All relationships for the associative entity should be many
Attributes		
	Attribute	In the Chen notation, each attribute is represented by an oval containing atributte’s name
	Key attribute	An attribute that uniquely identifies a particular entity. The name of a key attribute is underscored.
	Multivalued attribute	An attribute that can have many values (there are many distinct values entered for it in the same column of the table). Multivalued attribute is depicted by a dual oval.
	Derived attribute	An attribute whose value is calculated (derived) from other attributes. The derived attribute may or may not be physically stored in the database. In the Chen notation, this attribute is represented by dashed oval.
Relationships		
	Strong relationship	A relationship where entity is existence-independent of other entities, and PK of Child doesn’t contain PK component of Parent Entity. A strong relationship is represented by a single rhombus
	Weak (identifying) relationship	A relationship where Child entity is existence-dependent on parent, and PK of Child Entity contains PK component of Parent Entity. This relationship is represented by a double rhombus.

University ER Diagram



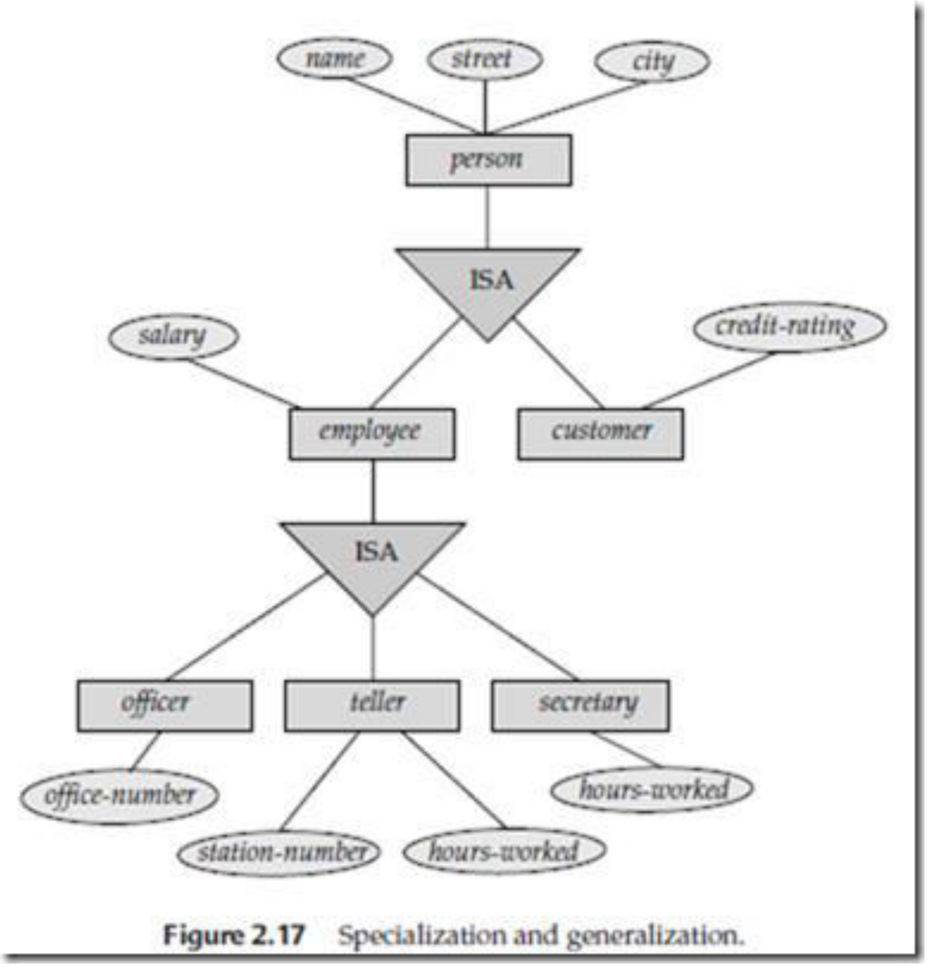
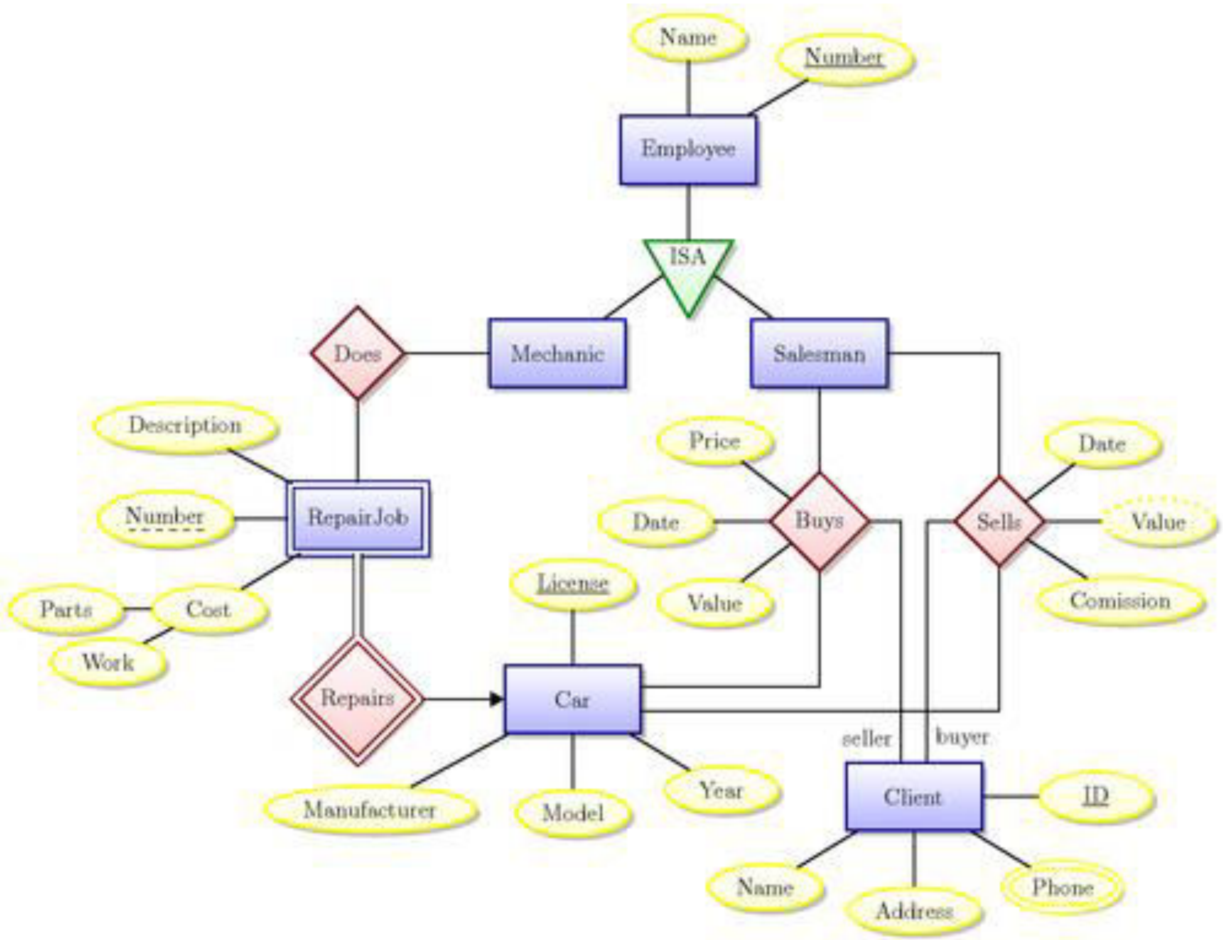


Figure 2.17 Specialization and generalization.


Name
Number
Employee
ISA
Does
Mechanic
Salesman
Description
Price
Date
Number
RepairJob
Date
Buys
Sells
Value
Parts
Cost
License
Value
Comission
Work
Repairs
Car
seller
buyer
Manufacturer
Model
Year
Client
ID
Name
Address
Phone