2^{nd} week scribe Notes

Entity relationship diagram

Documenting a software system using diagrams and symbols is called data modeling.
Highest level of abstraction for data modeling is called Entity relationship diagram(ERD).
Entities,attributes and relationships are three components of ERD.

- Entities: number of tables in your database.
- Attributes:Information such as property.all attributes have values.
- Relationships:How tables are linked to each other.

Entity:

Each table in your database is considered as a entity.
Ex:An entity can be an object with physical existence(Ex:student,car) or an object with conceptual existence(Ex:course,job).

Attributes:

Entities are represented by their properties called attributes.All attributes have values.

Ex:

I. Item entity may have attributes item type,item price,item source,brand.
II. Doctor entity may have Name,gender,specialization as attributes.

Values of attributes are of three types

i. Simple
ii. Composite
iii. Multi valued

Simple: these attributes are single atomic values.
Ex:roll no,gender.

Composite:It has multiple components which again has simple attributes or composite attributes.
Ex:Name(First name,Middle name,Last name)

Multi valued:It has multiple values for that attribute
Ex:skills of a surgeon.

Composite and multi valued attributes may be nested arbitarily to any number of levels.

Ex:Previous degrees of a student is a composite multi valued attribute denoted by {previous degrees (college,year,Degree,field)}.

An attribute of a an entity type for which each entity type must have a unique value is called a key attribute of the entity type
Ex:roll no of a student.

A key attribute may be composite and an entity type may have more than one key.

Value sets are similar to data types.value of attribute A for entity e is denoted as A(e).

Relationships:

A relationship relates two or more distinct entities with a specific meaning.
Relationship of same type are grouped into a relation type.
Relationship type:description of a relationship which identifies the relationship name and the participating entity types.
Relationship set is the current set of relationship instances represented in the database.

Relationships:

a. One to one(1:1)
b. One to many(1:N) or many to one(N:1)
c. Many to many(M:N)

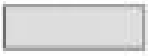
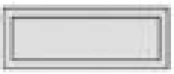
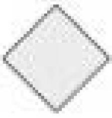
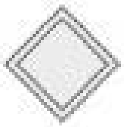
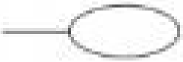
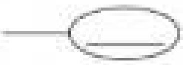
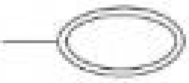
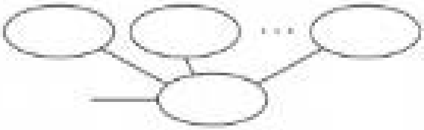
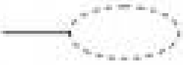
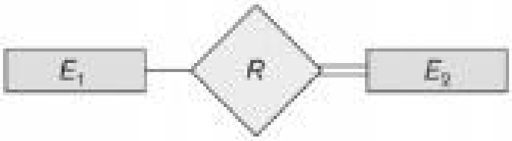
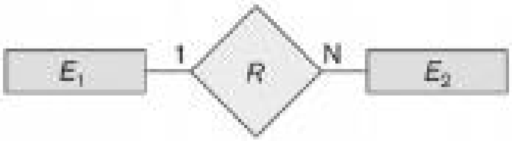
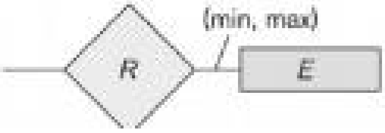
Cardinality ratio specifies the maximum participation ratio in a relationship.
Exsistence dependency constraint(participation constraint)specifies minimum participation

i. Zero
ii. One or more(mandatory participation)

A weak entity is a entity which does not have key attribute.so it must have an identifying relation ship with another entity.

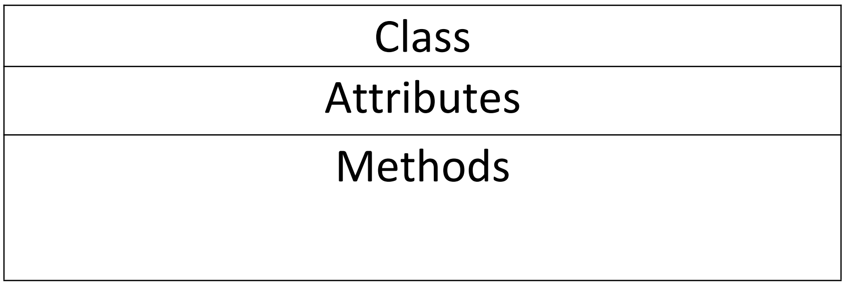
Number of entitys in a relationship is the degree of the relationship

Ex:binary => degree 2
N -ary => degree n

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Indentifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1: N for E_1:E_2 in R
	Structural Constraint (min, max) on Participation of E in R

UML class diagram:

UML(unified modeling language) is a software engineering language that was developed to create a standard way of visualising the design of a system.



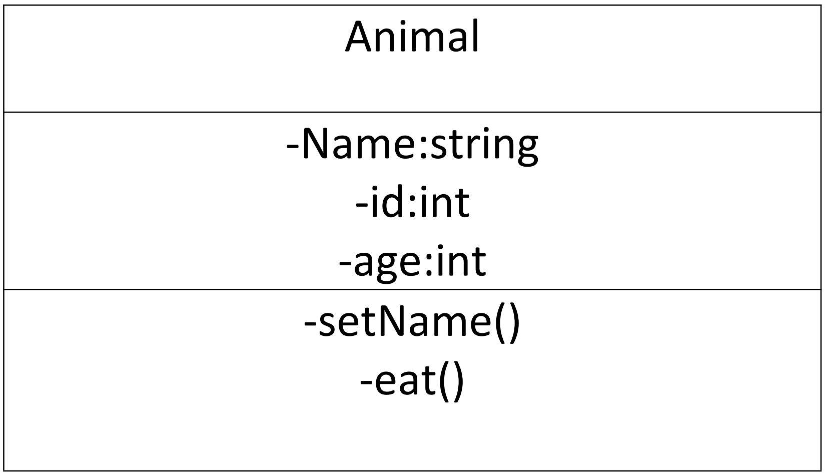
Class

Attributes

Methods

Upper section: It contains the name of the class.
Middle section: It contains the attribute of the class.
Bottom section:includes class operations I.e methods.

Ex:
zoo system:



Animal

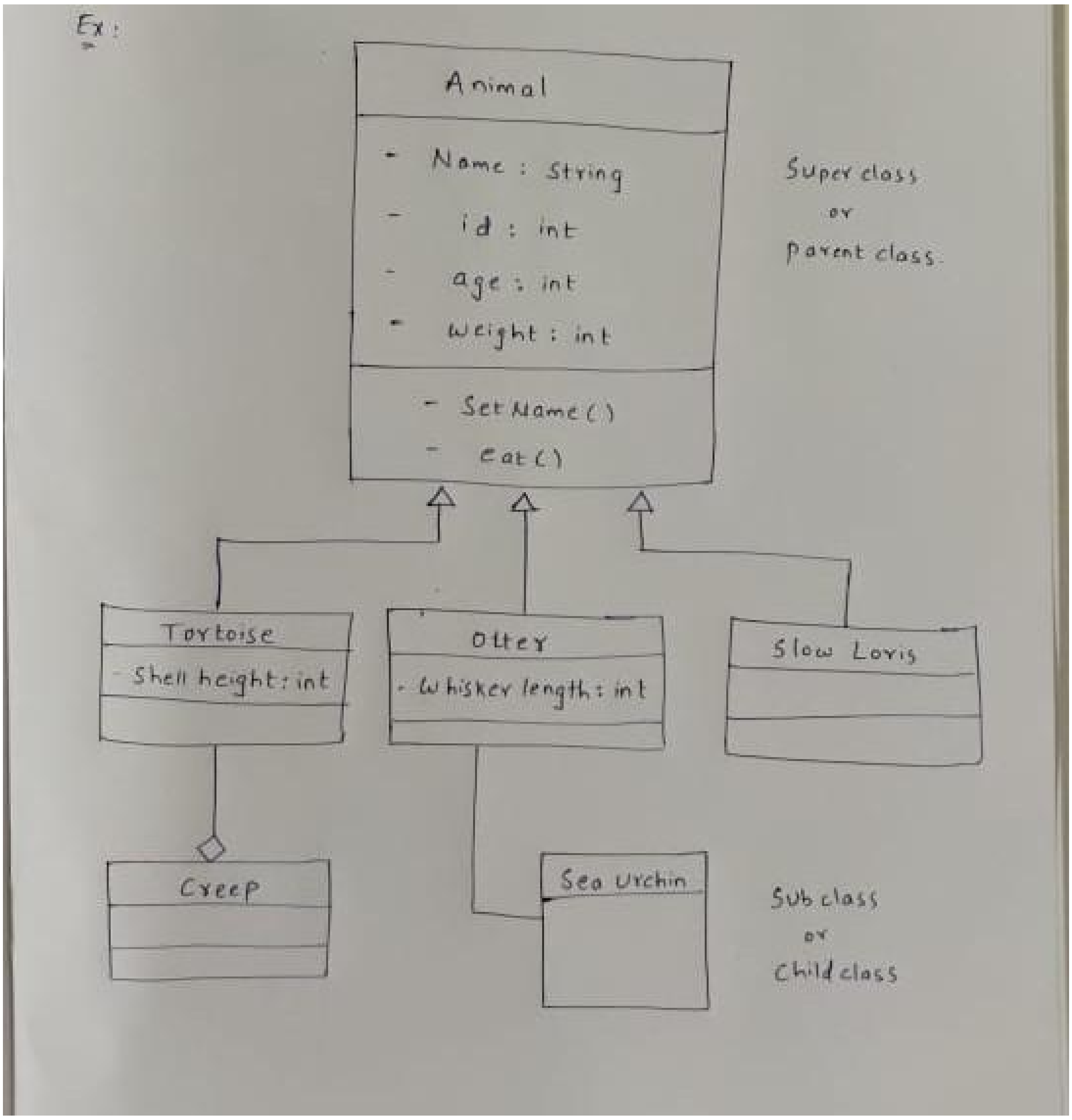
-Name:string
-id:int
-age:int

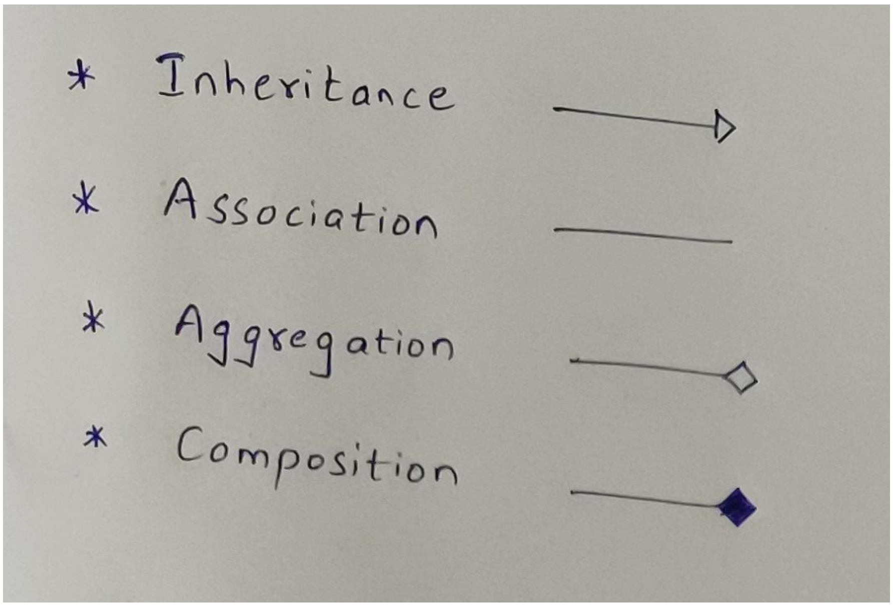
-setName()
-eat()

All classes have different access levels

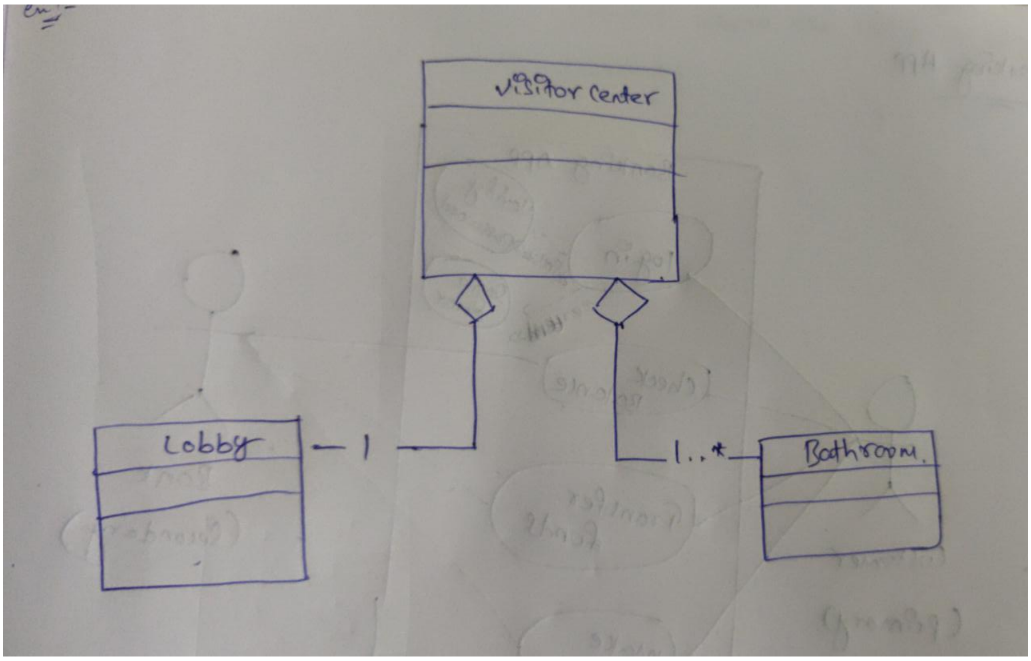
- Public(+)
- Private(-)
- Protected(#)
- Package(~)
- Derived(|)
- Static(underlined)

Ex:



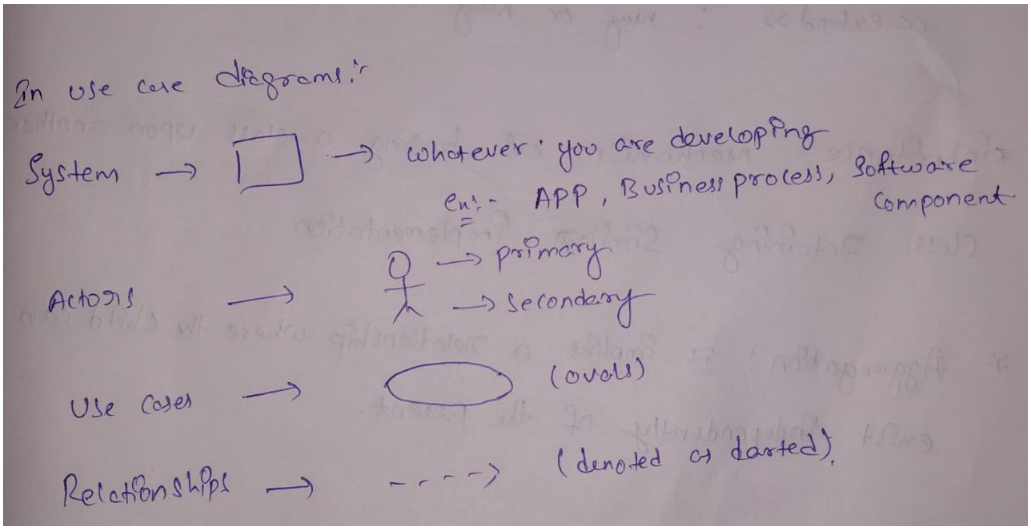


- * Inheritance ———▷
- * Association ———
- * Aggregation ———◇
- * Composition ———◆



Note:

0. .1	zero to one
N	specific number
0. .*	one to many
1. .*	one to many
m..n	specific number range



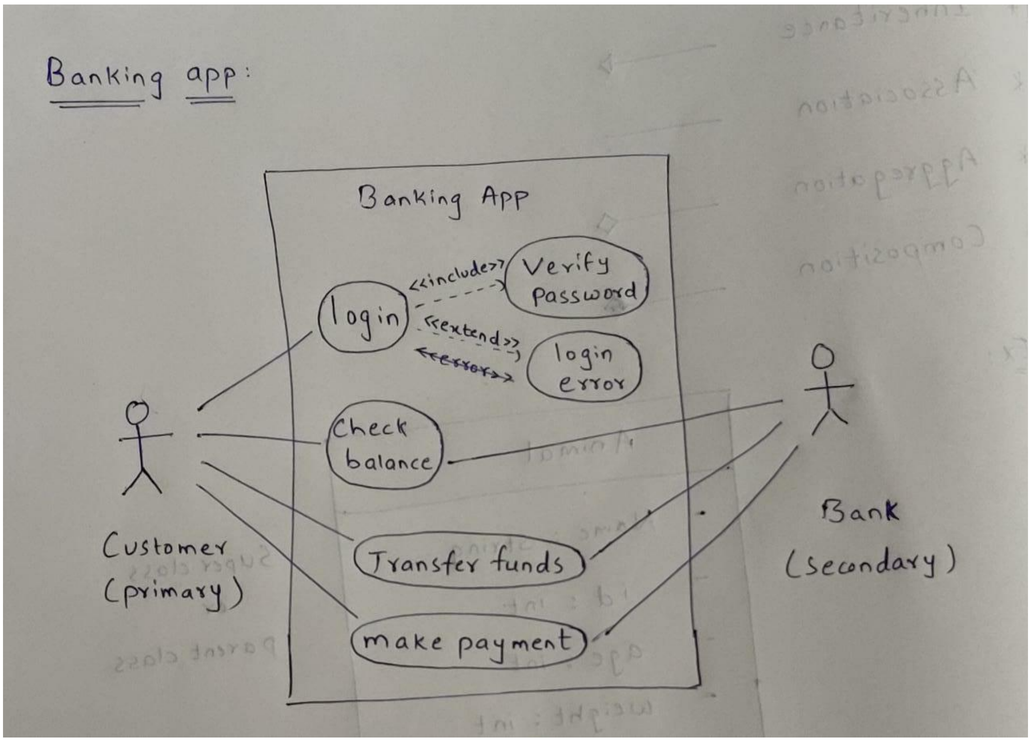
In use case diagrams:-

System → → whatever you are developing
ex:- APP, Business process, Software Component

Actors → → primary
→ secondary

Use Cases → (ovals)

Relationships → - - - -> (denoted as darted).



Association: Basic communication or interaction
<<include>>:Happens every time
<<extend>>:may or may not happen
In above example verifying password happens every time you try to login so it is considered as include association.but displaying login error happens when the password is wrong so it is a extend association.

Inheritance:Mechanism of basing a class upon another class retaining similar implementation.

Aggregation:It implies a relationship where the child can exist independently of the parent.

Composition:It implies a relationship where the child cannot exist independent of the parent.

Enhanced Entity Relationship Model (EER Model)

Features of EER Model

- EER creates a design which is more accurate to the database schemas.
- It reflects the data properties and constraints more precisely.
- It includes all modelling concepts of the ER model and object-oriented concepts.
- Diagrammatic technique helps us to portray the EER schema.
- It includes the concept of specialization and generalization.
- It is used to represent a collection of objects i.e. union of objects of different entity types.

A. Subclass and Superclass

- Subclass and Superclass relationship leads the concept of Inheritance.
- The relationship between subclass and superclass is denoted with ⓓ symbol.
- Superclass/Subclass relationships are called IS-A relationships.

1. Superclass

- Superclass is an entity type that has a relationship with one or more subtypes.

- An entity cannot exist in database merely by being member of any superclass.
  For example: Shape superclass is having sub groups as Square, Circle, Triangle.

2. Subclass

- Subclass is a group of entities with unique attributes.
- Subclass inherits properties and attributes from its superclass.
  For example: Square, Circle, Triangle are the sub class of Shape superclass.

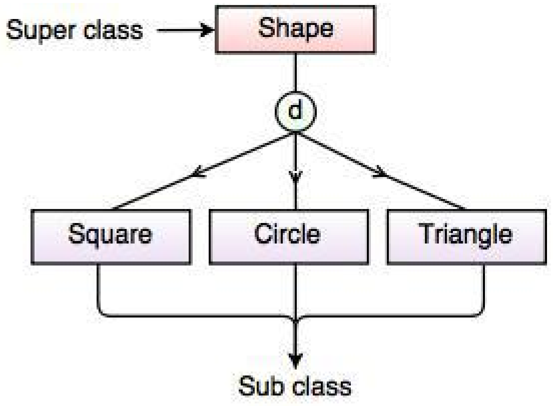


Fig. Super class/Sub class Relationship

3. A class with no subclass of its own is called **leaf node**.

4. A subclass with more than one superclass is known as **shared subclass**.

5. So apparently, shared subclass inherits attributes and relationships from multiple classes, which is called **multiple inheritance**.

6. If there are no shared subclasses **single inheritance** would exist.

B. Specialization and Generalization

1. Generalization

- Generalization is the process of generalizing the entities which contain the properties of all the generalized entities.
- It is a bottom approach, in which two or more lower level entities combine to form a higher level entity.
- Generalization is the reverse process of Specialization.
- It defines a general entity type from a set of specialized entity type.
- It supresses the difference among several entity types by identifying the common features.

For example:

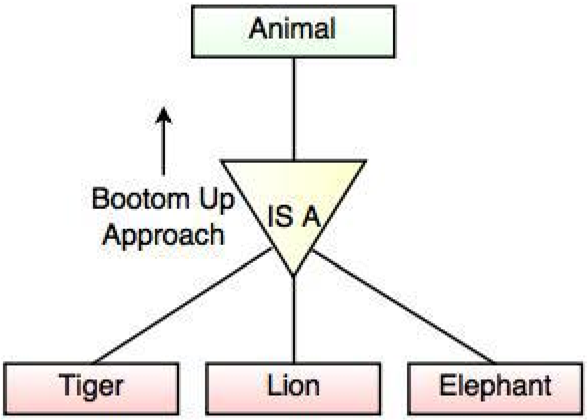


Fig. Generalization

In the above example, Tiger, Lion, Elephant can all be generalized as Animals.

2. Specialization

- Specialization is a process that defines a group entities which is divided into sub groups based on their characteristic.
- It is a top down approach, in which one higher entity can be broken down into two or more lower level entities.
- It maximizes the difference between the members of an entity by identifying the unique characteristic or attributes of each member.
- It defines one or more subclass for the superclass and also forms the superclass/subclass relationship. **For example**

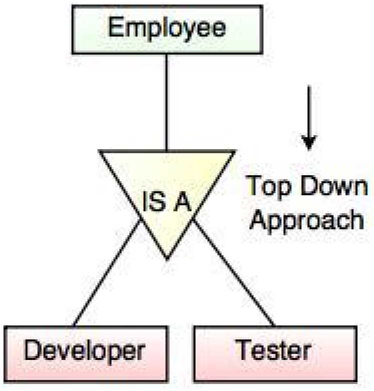
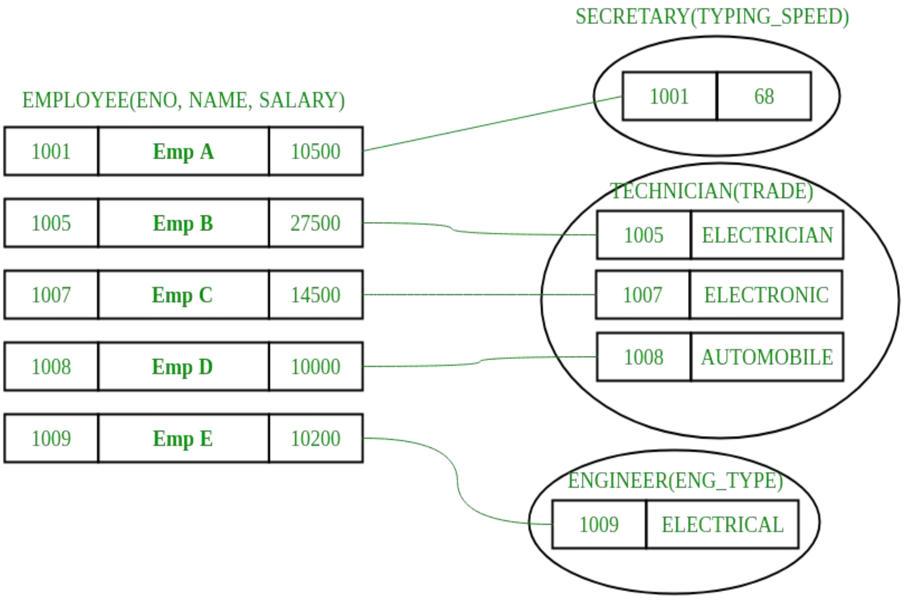


Fig. Specialization

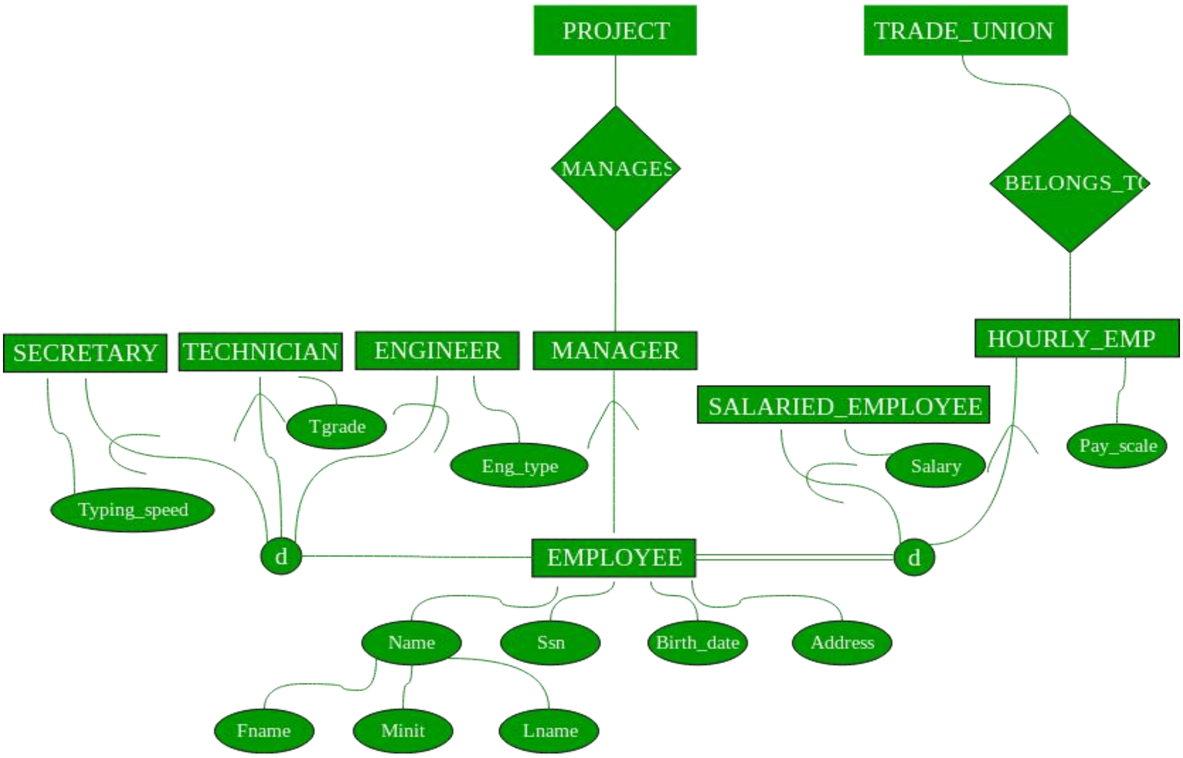
In the above example, Employee can be specialized as Developer or Tester, based on what role they play in an Organization.

Example – This example is an instance of **“sub-class”** relationships. Here we have four sets Employee, Secretary, Technician and Engineer. Employee is superclass of rest three and set of individual subclass is subset of Employee set.



- An entity belonging to a subclass is related with some superclass entity. For instance emp no. 1001 is a secretary, and his typing speed is 68. Emp no 1009 is engineer (sub-class) and her trade is “Electrical”, so forth.
- Sub-class entity “inherits” all attributes of super-class; for example employee 1001 will have attributes emp no. , name, salary, and typing speed.

Enhanced ER model of above example –



3. Specialization and Generalization Hierarchies and Lattices

- A subclass may have further subclasses(nested),forms a hierarchy or a lattice of specializations.
- Specialization hierarchy is constrained that each subclass participates only in one subclass relationship i.e. having only one parent resulting in tree structure or strict hierarchy.
- On the other hand is Specialization lattice, in which a subclass can have more than one subclass relationships.

- The same is reflected with the Generalization hierarchy and Generalization lattice.

C. Category or Union

- Category represents a single superclass or subclass relationship with more than one super class.
- It can be a total or partial participation.
  For example Car booking : Car owner can be a person, a bank (holds a possession on a Car) or a company. Category (sub class) → Owner is a subset of the union of the three superclasses → Company, Bank, and Person. A Category member must exist in at least one of its superclasses.

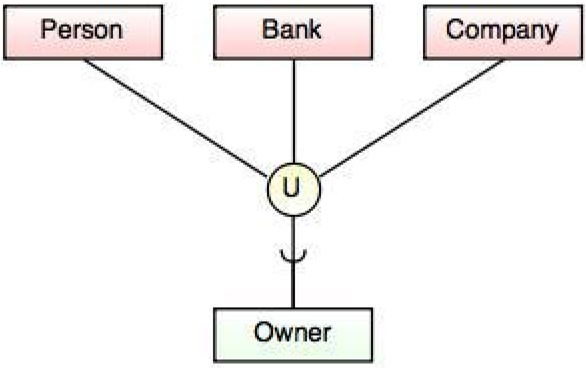


Fig. Categories (Union Type)

D. Aggregation

- Aggregation is a process that represent a relationship between a whole object and its component parts.

- It abstracts a relationship between objects and viewing the relationship as an object.
- It is a process when two entity is treated as a single entity.

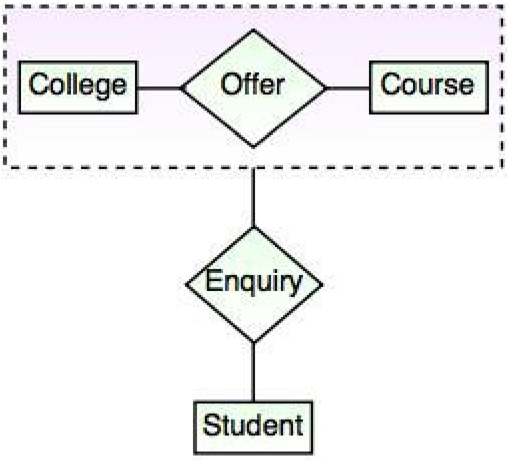


Fig. Aggregation

In the above example, the relation between College and Course is acting as an Entity in Relation with Student.

REFERENCE:

Tutorial point.
(geeksforgeeks)
comet.lehman.cuny.edu
cs.toronto.edu

http://iips.icci.edu.iq/images/exam/databases-ramaz.pdf

https://www.tutorialspoint.com/dbms/er_model_basic_concepts.html

ftp://www.cs.uregina.ca/pub/class/215/erd/index.html#att

https://www.lucidchart.com/pages/uml-class-diagram?a=0#top-info