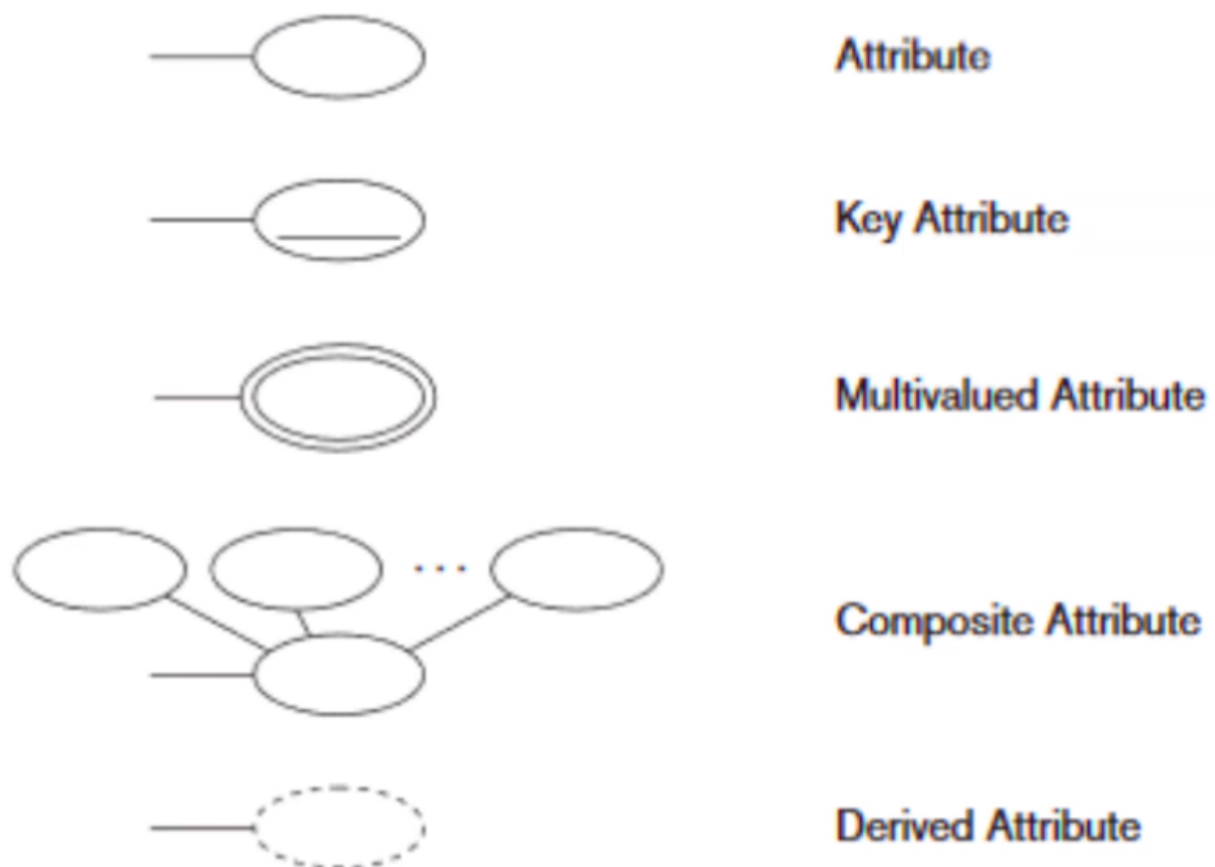


20210902 - DNA TUTORIAL ON THE ER MODEL

ENTITIES

Entities represent objects and have multiple attributes. That describe the data relating to a certain entity.

There are many kinds of attributes such as:



Entity types are entities that possess the same attributes and it describes the schema for an entity

Entity Sets are all the entities of one type that are in a database.

Weak Entities

Weak entities have a **partial key** that identifies it among the other weak entities that are related to a strong entity

RELATIONSHIPS

Binary relationships

These relationships are ones that relate entity types.

Identifying relationships

These relationships are the ones that identify weak entities since they are not identified otherwise.

The weak entities have an existential dependency on strong entities.

Structural Constraints

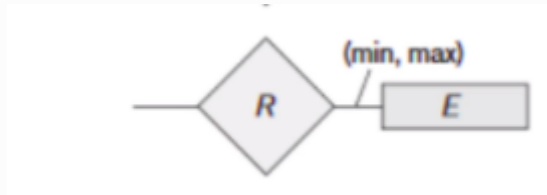
Participation Constraints are either *total participation*, which means that all entities of that set are related to at least one other, or *partial participation* which means that not all entities of this type participate in this relationship.



Cardinality ratio gives an idea of how many entities are related for each entity of the other type

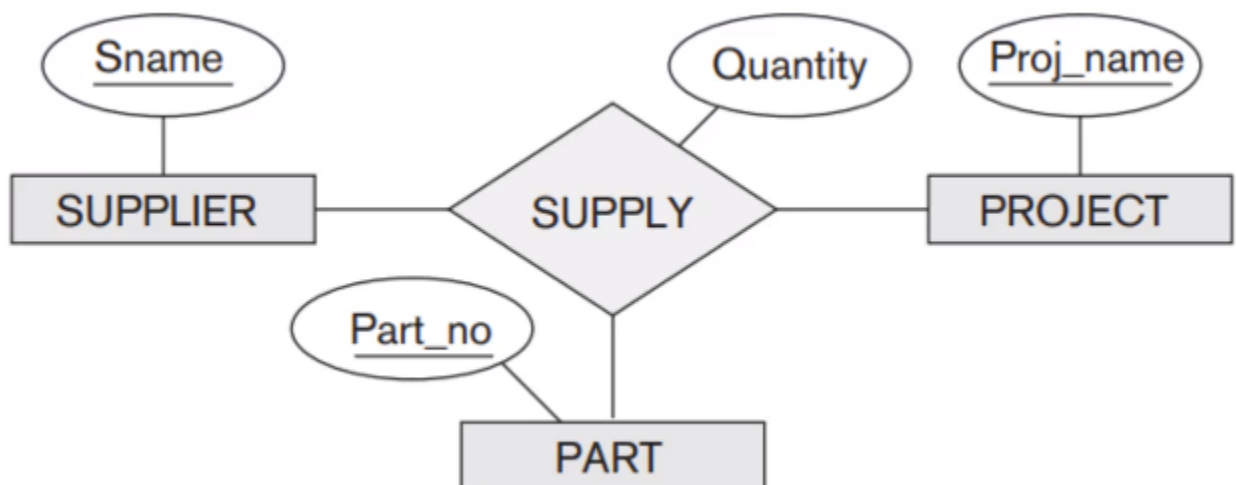


Cardinality Constraints show the max/min number of entities that can participate in a relationship.

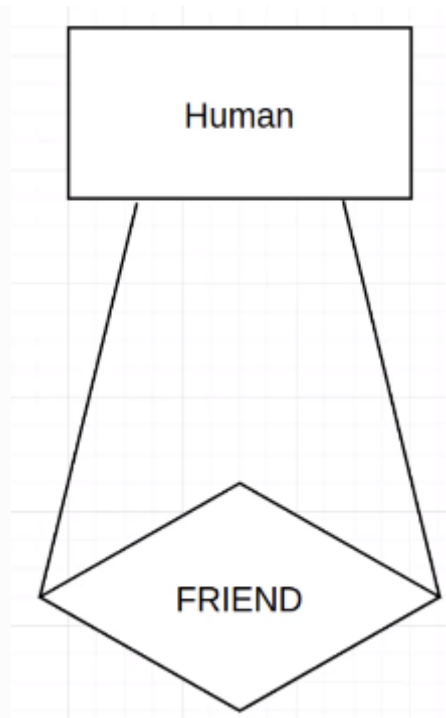


Higher Degree relationships

Relationships can have more than two types of entities participating in it. And **relationships can also have attributes**.



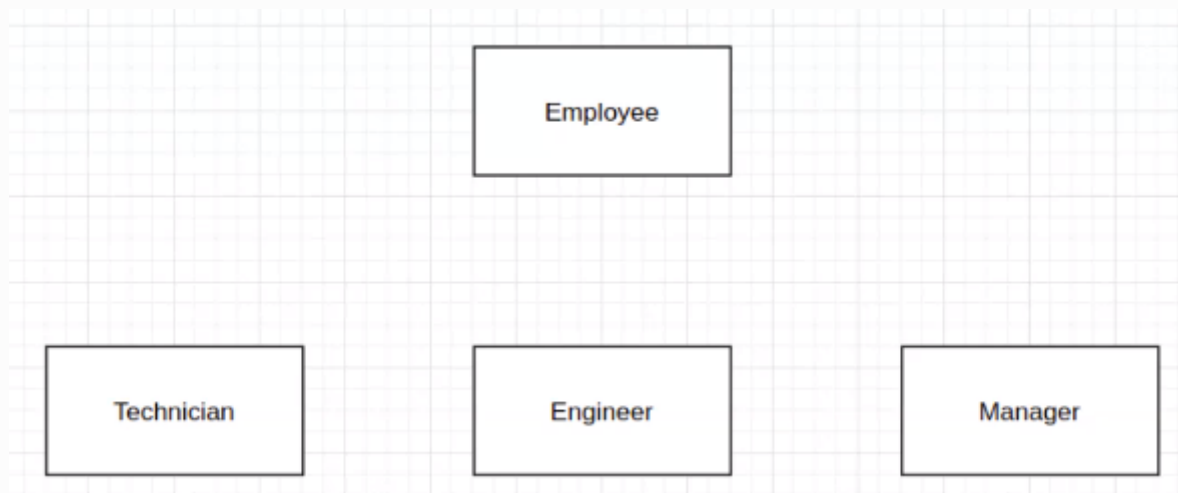
Self Relationships



These are relationships that exist between entities of the same **type**.

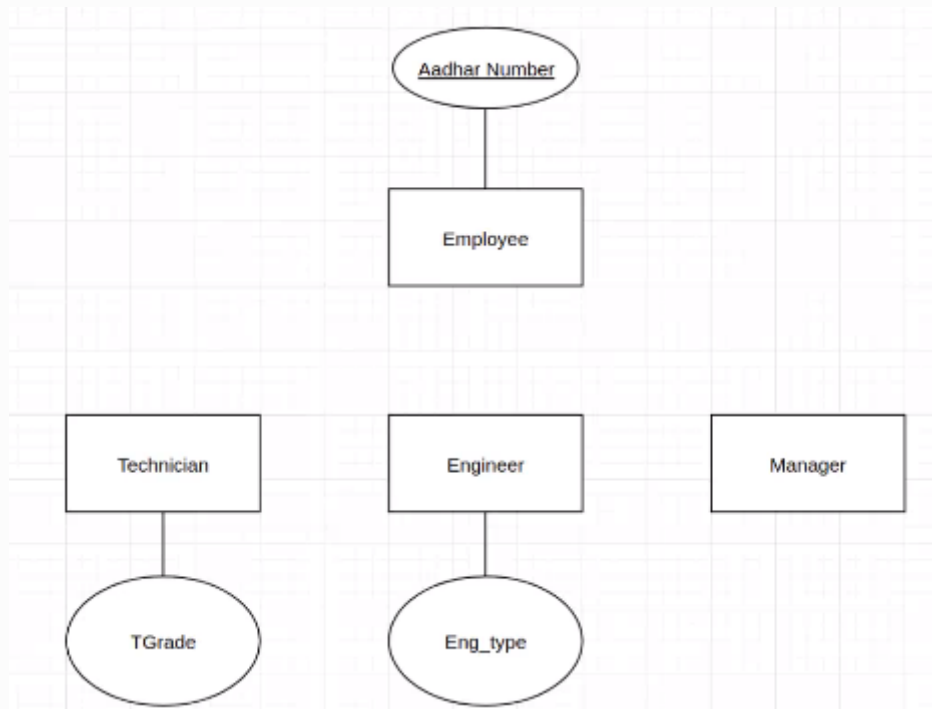
ENHANCED ER MODEL

Subclass / Superclass



An entity can be of a type that is a **subclass** of a **superclass** but need not be since they are already of the superclass type.

Type Inheritance / Local Attributes

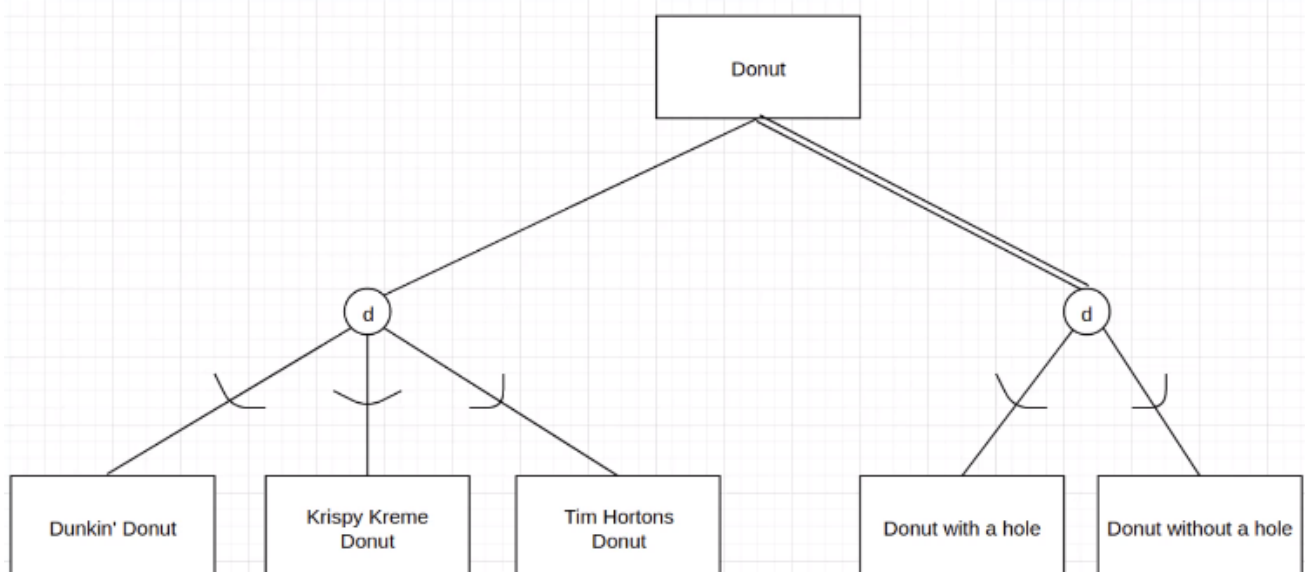


Subclasses will have all the attributes that are a part of the superclass on top of other attributes that are a part of the subclass.

Specialization / Generalization

In the **Generalization** process properties are drawn from particular entities and thus we can create generalized entity. And vice versa with **Specialization**.

Completeness Constraint



Entities of a certain superclass need not be classified into a subclass type

A double line indicates total participation in a Specialization / Generalization.

d stands for **disjoint** i.e. donuts can only be from exclusively one of the subclasses.

o stands for **overlapping** i.e. an entity can be of any of the subclass types.

QUESTIONS FROM THE TEXTBOOK

(watch recording)

PROJECT

There are four phases in the project:

Requirements Document

- Define a mini world
- Define entities
- Understand how they interact
- Translate interactions to relationships
- Define Boundaries
- Define basic system behaviour

Sections in the requirements document:

Refer **template document**.

Functional requirements

- Functional requirements should include
 - Descriptions of data to be entered into the system
 - Descriptions of system reports or other outputs
 - Who can enter the data into the system
 - How the system meets applicable regulatory requirements
- In the scope of this project, Functional requirements should relate to the tasks that the database system will perform, usually in the form of access, searching, reports and sorting (queries). Functional requirements may also provide detail around the data that must be held in the database.