# Database Design (Rough ER Schema, Final ER Schema, Mapping).

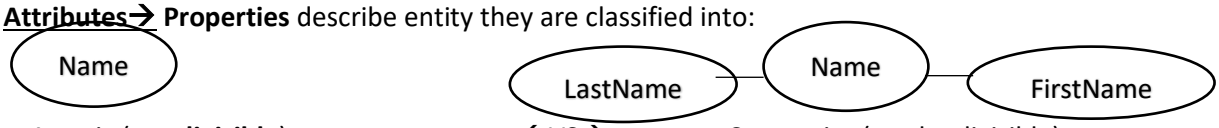
DB=Automation to manual process
DBMS= Software used to build DB systems like: Oracle, My SQL, MS SQL Server,…
Database Design Steps: Rough ER Schema→ Final ER Schema→ Mapping.

I. ***Entity***→ Something that has a meaning **inside system**, may be **physical** or **logical**. It is either a ***regular entity*** (independent existence) or ***weak entity*** (dependent existence).


(Regular Entity) (Weak Entity)

Each weak entity should have an owner entity. There is a relation between week entity and its owner called ***identification relation***

II. **Attributes→ Properties** describe entity they are classified into:




1. Atomic (not **divisible**) ←VS→ Composite (can be divisible)

---


2. Single (**one value** per each entity instance) ←VS→ Multi-values (multi-values per each entity instance)

---


3. Stored (**saved** in DB) ←VS→ Derived (**calculated from stored attributes**)

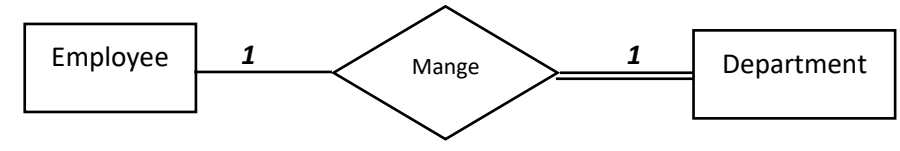
---

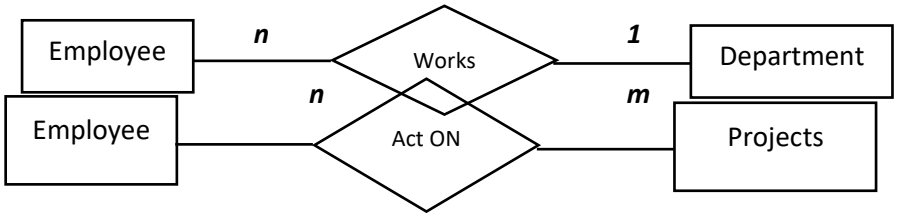

4. Key attribute satisfies (***uniqueness*** – not null – minimality ) ←VS→ Non Key

---

***III.*** ***Relation→*** is an attribute that describes an entity and refers to another entity, so it will be transferred (in the final ER schema) to a relationship between the entity that describes and entity that it refers to. For each entity we should determine 1) **participation** (partial → one line ----total →double line), 2) **cardinality ratio** (1:1←→ 1:n ←→m:n)

***Examples on Relationships:***



***Total Participation*** − Each entity is involved in the relationship. Total participation is represented by double lines.
***Partial participation*** − Not all entities are involved in the relationship. Partial participation is represented by single lines.

## Steps of Final Mapping

**Objectives**: After designing the ER diagram of system, we need to convert it to Relational models which can directly be implemented by any RDBMS like Oracle, MySQL etc. In this article we will discuss how to convert ER diagram to Relational Model for different scenarios.

1. For each **regular entity** make a new table contains all attributes **except** *composite, multi-values, and, relationship.* Start with key attribute as a primary key (PK).
2. For each **weak entity** make a new table contains the PK of its owner entity and put all of the weak entity's attributes like the previous step. The ***primary key*** of the new table: ***PK*** of its owner ***+ a partial key*** taken from weak entity (composite key).
3. For each **multi-values** attribute make a new table that contains the PK of its entity (done in either step 1 or 2) + multi-values attribute. ***The primary key*** of the new table: **PK of entity + multi-value attribute** (composite key).
4. For each relation **m:n** make a new table by putting the primary keys of the entities that participate in this relation and any attributes emit from this relationship. ***The primary key*** of the new table: **PKs of all entities participate in this relation** (composite key).
5. For each relation **1:n** take primary key of the entity in the side of 1 and put it in the table of the entity in the side of n.
6. For each relation **1:1** take primary key of the entity in the side of partial participation (single line side) and put it in the table of the entity in the total participation (double side). In case the participation is the same in both sites (partial ←→ partial or total ←→ total) take any decision.

! Note for any primary key transfers to a new location, it is considered as a foreign key in the new position. This means it references its original place. You are not allowed to enter a value in the foreign key not inserted before in its primary key.