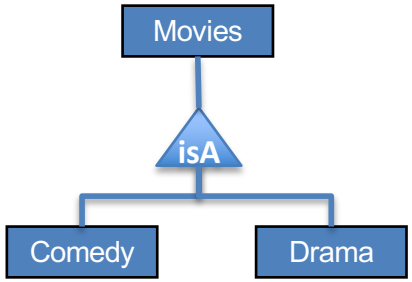
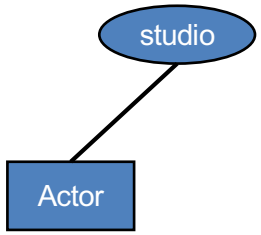

***Entity:*** *Models an object From the real world and Can be by itself*


***Relationship:*** *Connects entities*



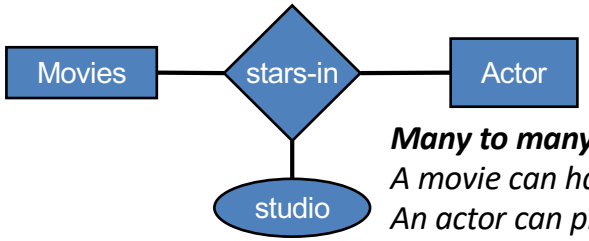

***isA:*** *like subclasses*



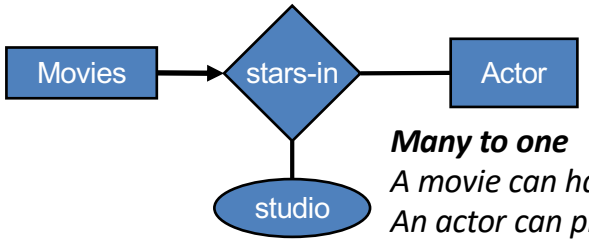

***Attributes:*** *Properties of an entity or relationship. The keys are indicated by underlining the attribute name*

# ER & Relational Model Cheat-Sheet

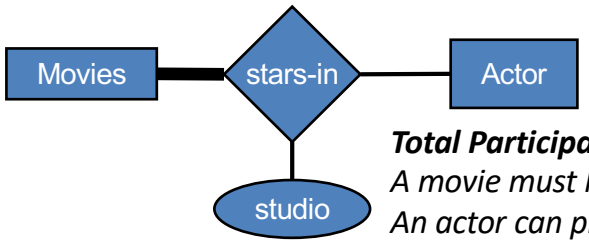
**By Yannis Velegrakis**



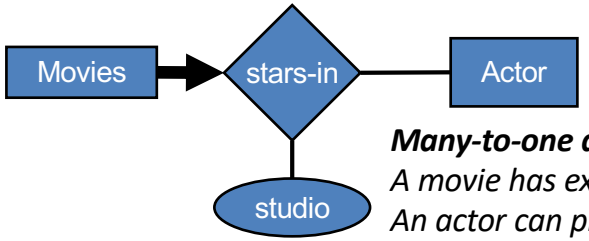

***Many to many***
*A movie can have many(or 0) actors*
*An actor can play in many (or 0) movies*



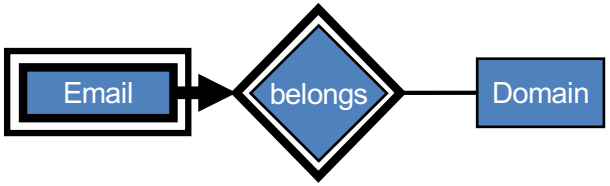

***Many to one***
*A movie can have 0 or 1 actors*
*An actor can play in many (or 0) movies*



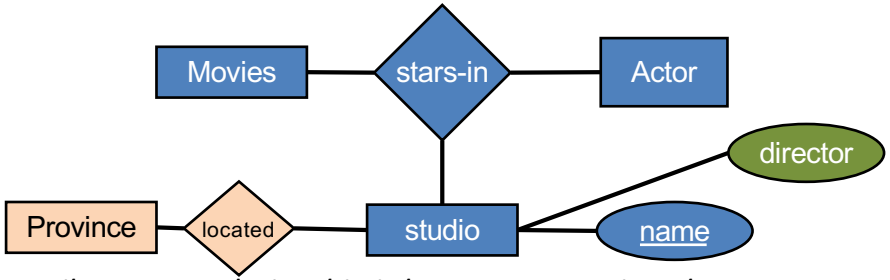

***Total Participation***
*A movie must have at least 1 actor*
*An actor can play in many (or 0) movies*




***Many-to-one and total participation***
*A movie has exactly 1 actor*
*An actor can play in many (or 0) movies*

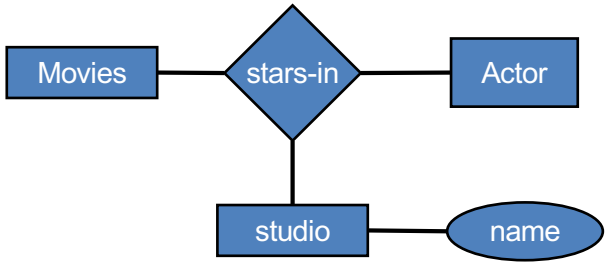



***Weak Entity:*** *Cannot exist without the main entity. But is an entity because we need to connect to it other relationships and also because it does not connect two or more other entities to be a relationship*

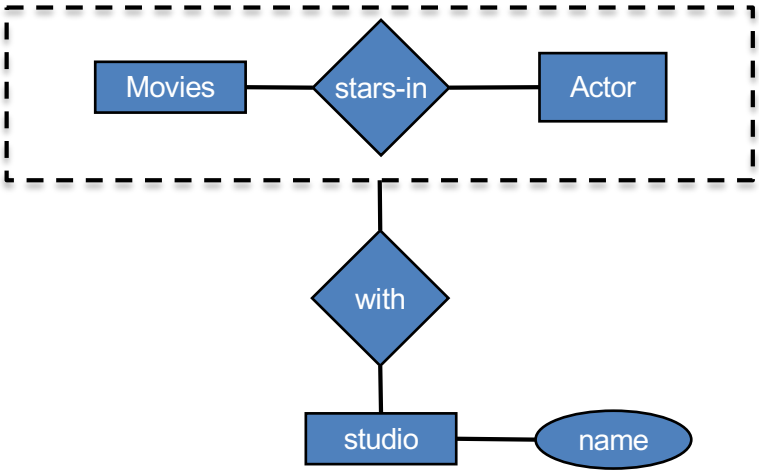



*An attribute on a relationship is better as an entity when*

1. *It has a complex structure (needs to have many attributes, as above)*
2. *Needs to have a relationship with other entities….*
3. *… other cases*




***Ternary Relationship** (more than one entities)*




***Aggregation**: when entities connected are seen as 1 entity*
*Used when the starts-in can exist without a studio, or when the stars-in needs to be one to many to with, or when with has its own attributes.*

***Key:** A set of attributes that uniquely identify a tuple, meaning*
*If I give you the values for these attributes, you can identify*
*At most one tuple in the instance of the database.*

*If X is a key, any **superset** is also a key.*
*All these supersets are called **superkeys**.*

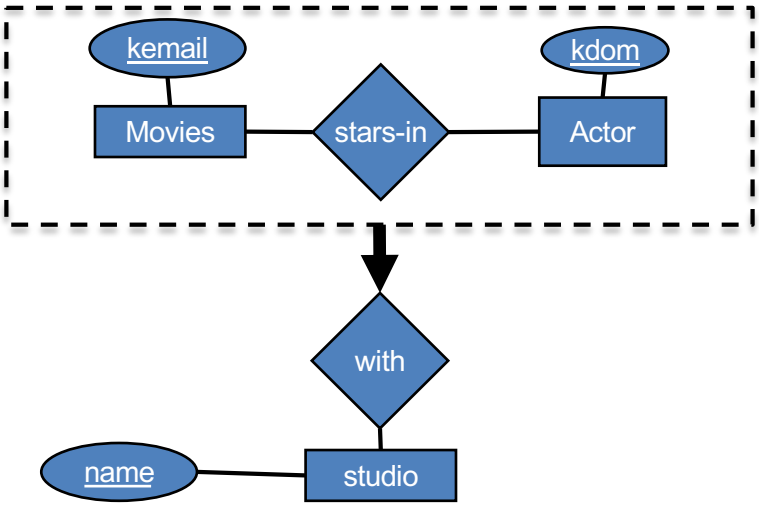
*A **candidate** key is a minimal key.*
*(minimal means that if you remove even one attribute of the set, then the set is not a key anymore.)*

*A **primary** key is the key that the database administrator chose to declare as a key in a table (because in a table you can have only One PRIMARY KEY definition. The rest of the keys, can be defined using the UNIQUE statement.*

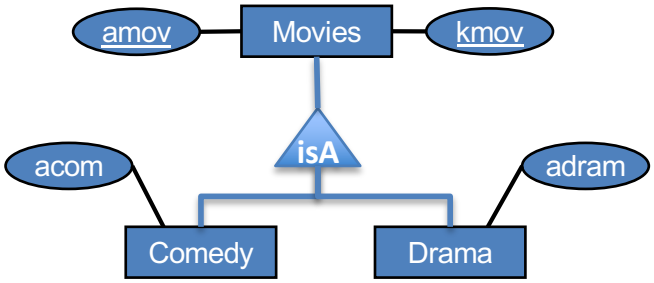
*The keys are indicated in the ER and the relational schemas by underlining the names of the attributes that form the key*

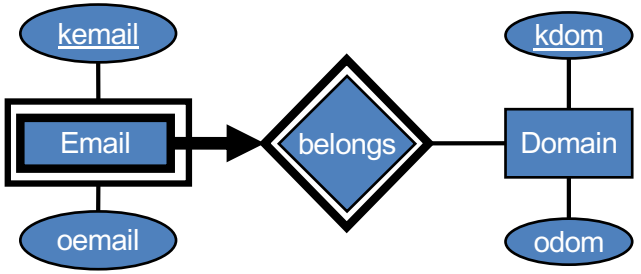
***An ER is translated to a relational as follows:***

1. *Each Entity becomes a relation and its attributes columns*
2. *Each Many to many Relationship becomes a relation with attributes*
   - *A. Attributes of the relationship*
   - *B. The keys of every entity it connects (these attributes become foreign keys to the respective relations derived from the entities.*
   - *C. The key is all the columns (attributes) created in B above*
3. *Each Many to one Relationship becomes a relation as in 2. above but the key is ONLY the key of the Many part.*
4. *Each Many to one and total participation relationship does not become a relation by itself, but its attributes and the keys of the "to one" entities are added in the attributes of the "Many" entity. The keys of the "to one" entity has to be foreign key for sure and has to be requested to be NOT NULL.*
5. *Special cases (see figures):*

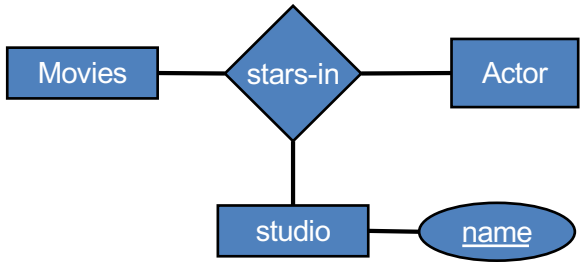



***Starsin****(kemail,kdom,name)*
*Name cannot be null*




***Domain(****kdom,odom)*
***Email(****kdom,kemail,oemail)*




***Ternary Relationship*** *(it falls under case 2 in the rules above)*

***Option 1:***
*Movies(kmov,amov)*
*Comedy(kmov,acom) kmov FK to Movies*
*Drama(kmov,adram) kmov FK to Movies*

***Option 2:***
*Movies(kmov,amov)*
*Comedy(kmov,amov,acom) kmov FK to Movies*
*Drama(kmov,amov,adram) kmov FK to Movies*

***Option 3:***
*Movies(kmov,amov,acom,adram)*