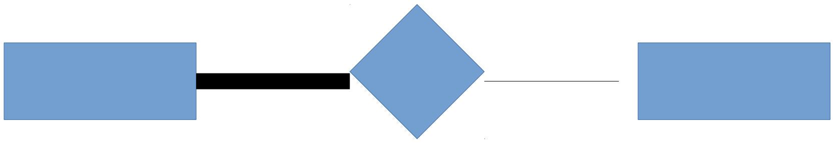
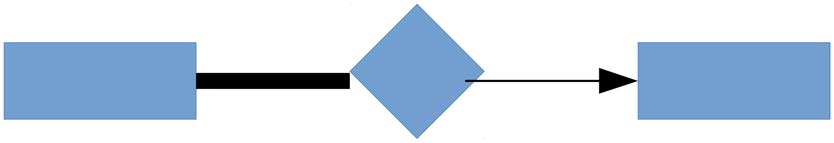
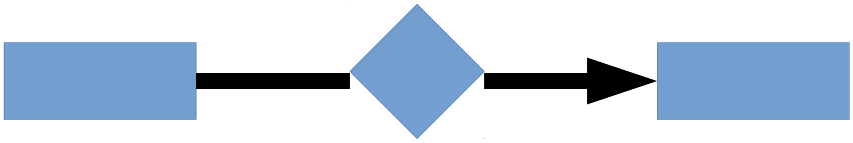


The dark or double arrow from Professor to Teaches defines a participation constraint, meaning every professor must teach ***at least one*** class.



The arrow from into Course defines an *at most one* limitation on the Teaches relationship. When coupled with the same *at least one* participation constraint from above (the dark line between professor and teaches) the combination define an ***exactly one*** relationship.



The same *exactly one* relationship from above is defined by the dark line from Teaches to Professor combined with the arrow from Teaches into Course. The addition of the dark or double line into Course adds a participation constraint requiring that every Course is taught by *at least one* professor.

Problem 2: Database Design

1) belongs to is a *many-to-one* relationship from Artist to Label

2)
Each album must be produced by at most one label.
Each album must have at least one song.
Each artist must sing at least one song.
Each artist can belong to at most one label.
Each label must have at least one artist.

3)
Produces(_LabelName_, _AlbumName_)
Sings(_ArtistId_, _SongId_)
AppearsOn(_SongId_, _AlbumId_)

Song(_id_, name, duration)
Album(_id_, name)
Label(_name_, address)
Artist(_id_, name, dob, label_name)

- I combined the Artist and Belongs to entities because each artist can only belong to one label so an additional table is not needed here to represent that relationship. Instead the label an artist belongs to is represented by the label_name introduced to the Artist entity.
- Each value of the label_id attribute in Artist must match a value of the name attribute from the Label relation.

Problem 3: Combining Relations

1)

Ra	Rb	Rc	Sa	Sb
1	2	3	2	3
1	2	3	3	4
1	2	3	7	6
3	4	3	3	3
3	4	3	3	4
3	4	3	7	6
7	6	5	2	3
7	6	5	3	4
7	6	5	7	6

2)

A	B	C
3	4	3
7	6	5

3)

A	B	C
1	2	3
3	4	3
7	6	5

4)

A	B	C
2	3	_
3	4	3
7	6	5

5)

A	B	C
1	2	3
3	4	3
7	6	5
2	3	_

```
#1
π Oscar.year, Oscar.type, Movie.name (
    σ Person.name='Meryl Streep' (
        (
            ((Oscar) ⋈ Oscar.person_id = Person.id (Person))
                 ⋈ Oscar.movie_id = Movie.id (Movie)
         )
    )
)

#2
(π Actor.actor_id (Actor)) -
(π Actor.actor_id (
    σ Movie.year ≥ 2010 (
        (Actor) ⋈ Actor.movie_id = Movie.id (Movie)
    )
))


#3
π Person.name, Oscar.type, Oscar.year (
  (σ Person.pob LIKE '%Mexico' (Person)) ⟕ Person.id = Oscar.person_id (Oscar)
)


# For my own amusement

#5
π Movie.name, Movie.year (σ Movie.name LIKE '%Lady%' (Movie))

#6
σ Person.pob LIKE '%UK' (
  (Person) ⋈ Person.id = Oscar.person_id (Oscar)
)

#7 – how to get MIN in RA?

#8
π Movie.name, Movie.year
(σ Movie.name = 'Good Will Hunting' ∨ Movie.name = 'Mystic River' (Movie))


#....it's late :)
```