

## 2. E/R Design Considerations

# What you will learn in this section



Relationships cont'd: multiplicity, multi-way

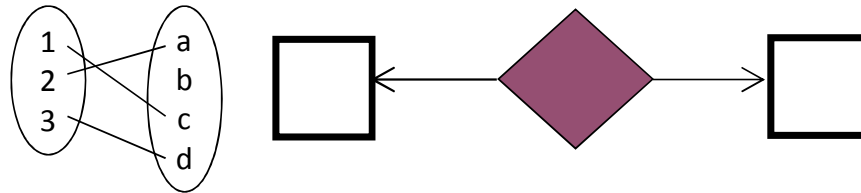
Design considerations

Conversion to SQL

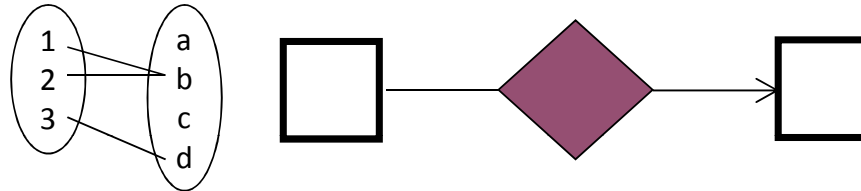
# Multiplicity of E/R Relationships

# Multiplicity of E/R Relationships

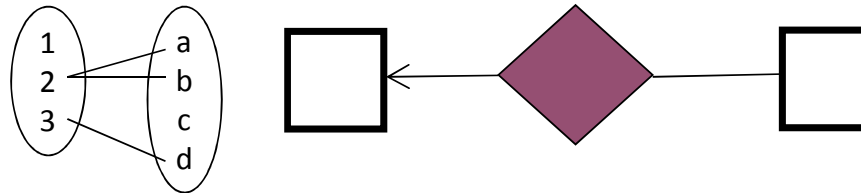
One-to-one:



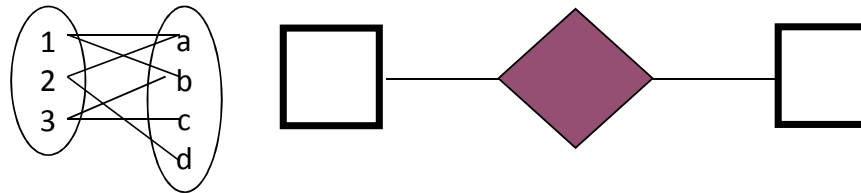
Many-to-one:



One-to-many:



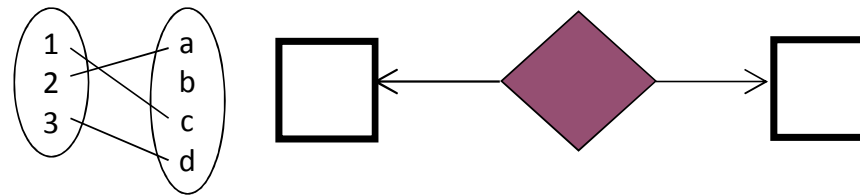
Many-to-many:



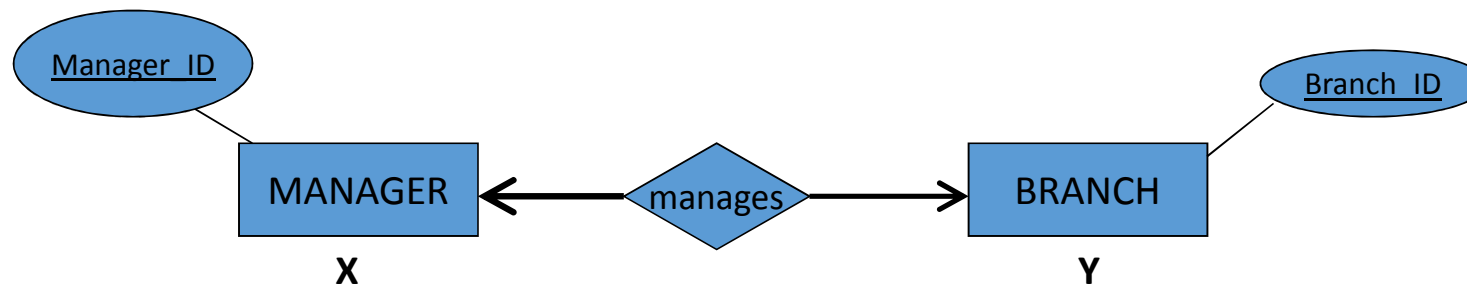
Indicated using  
arrows

# Example: One to One Relationships (1:1)

One-to-one:



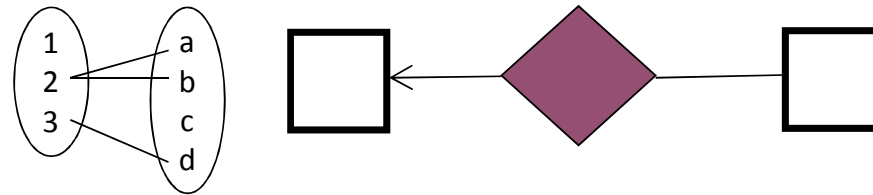
Def : Maximum one X for each Y and one Y for each X



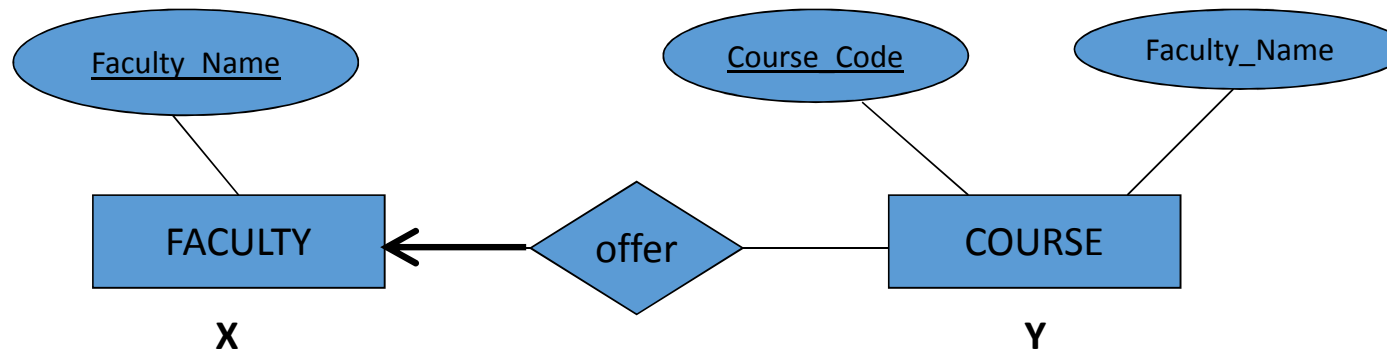
**An employee (manager) can manage only one branch at one time and each branch only has one manager.**

# Example: One to Many Relationships (1:M)

One-to-many:

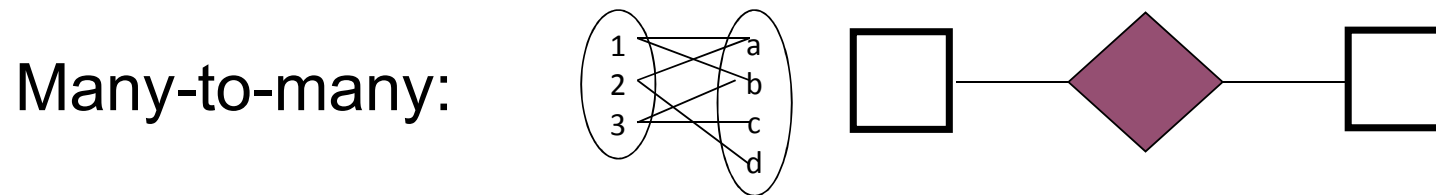


Def : Maximum one X for each Y but possibly many Y for each X

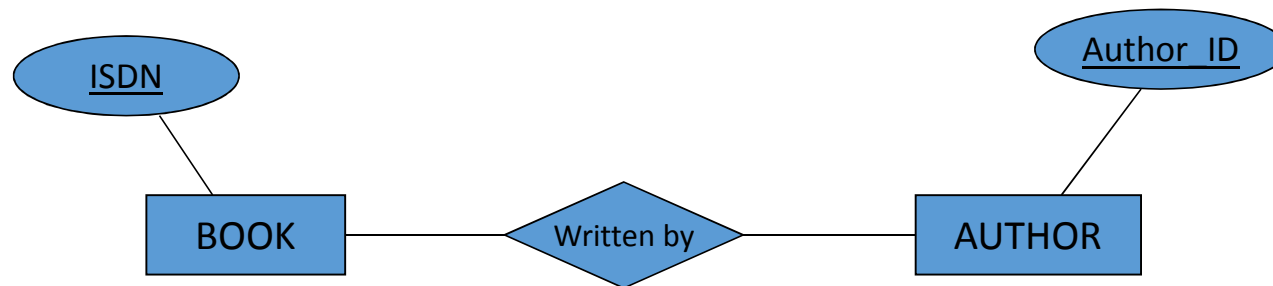


**At university, one faculty offers many courses for students but one course is offered by one faculty only.**

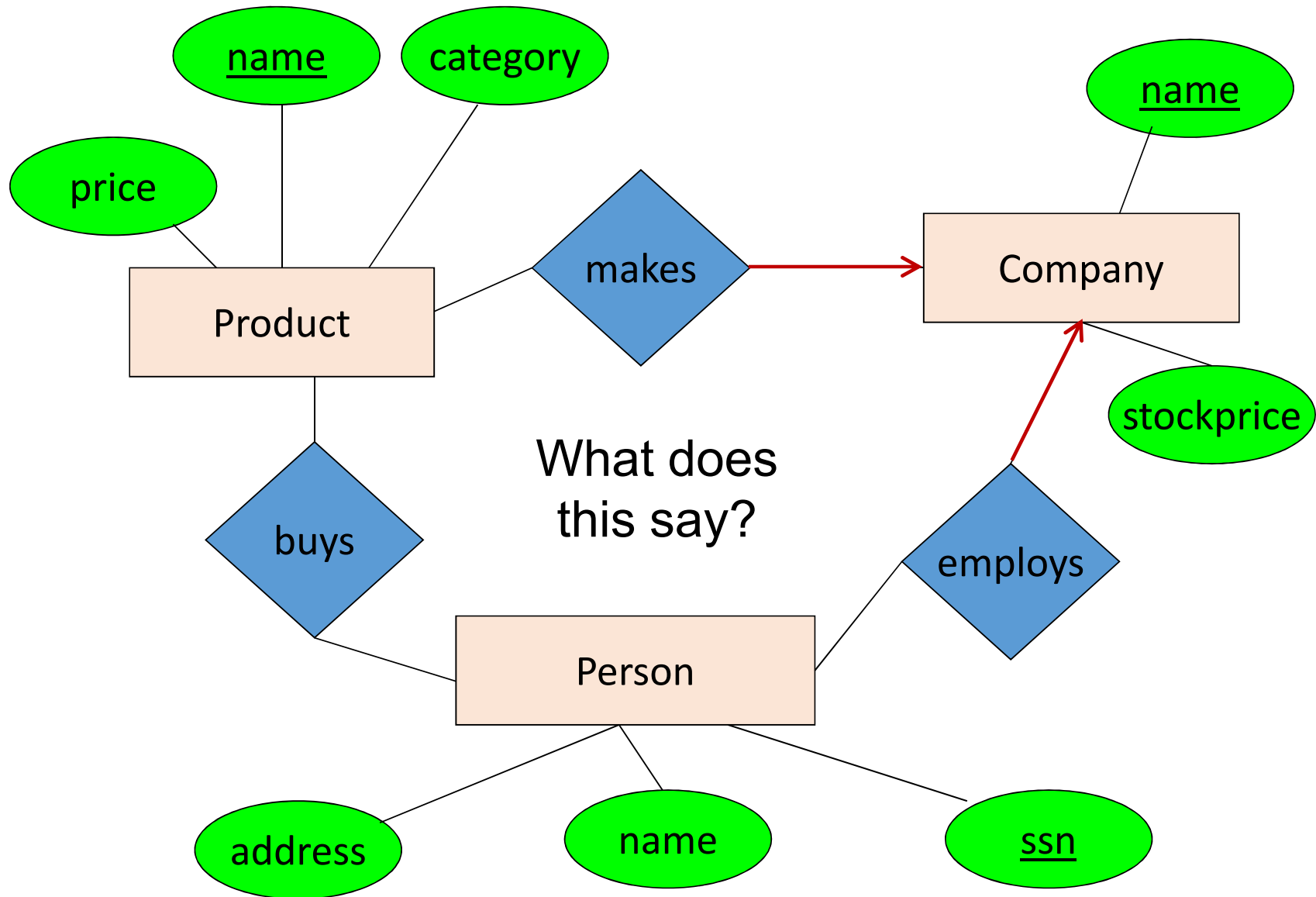
# Example: Many to Many Relationships (M:N)



Def : Possibly many X's for each Y and many Y's for each X



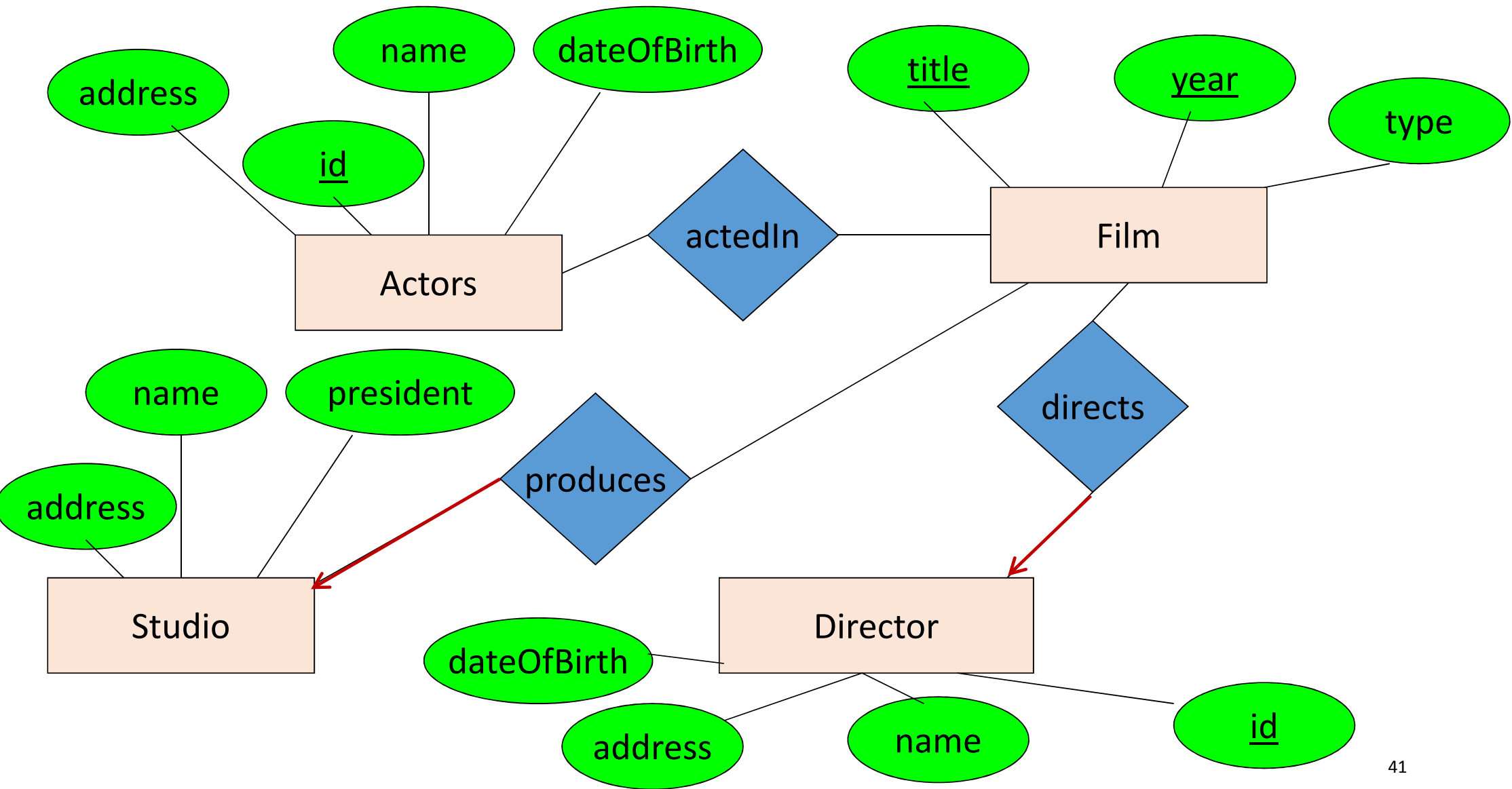
**A book may have more than one author and an author may write more than one book.**



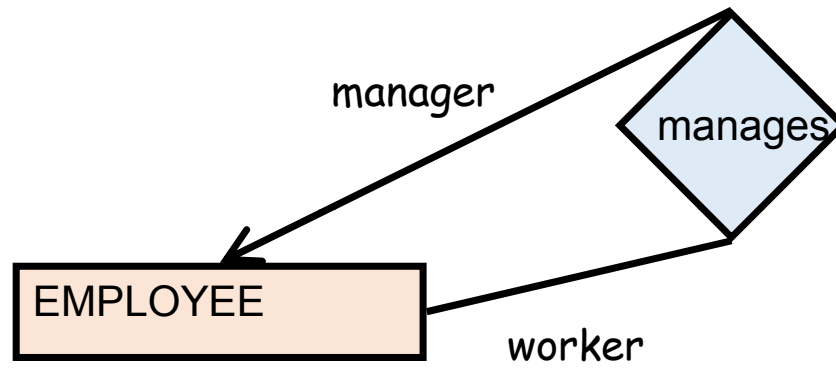


# Activity in the class

- For actors and directors, we want to store their name, a unique identification number, address and birthday (why not age?)
- For films, we want to store the title, year of production and type (thriller, comedy, etc.)
- We want to know who directed and who acted in each film. Every film has one director. We store the salary of each actor for each film
- For Studio we want to store name, address and name of its president. A movie can be produced by only one studio.



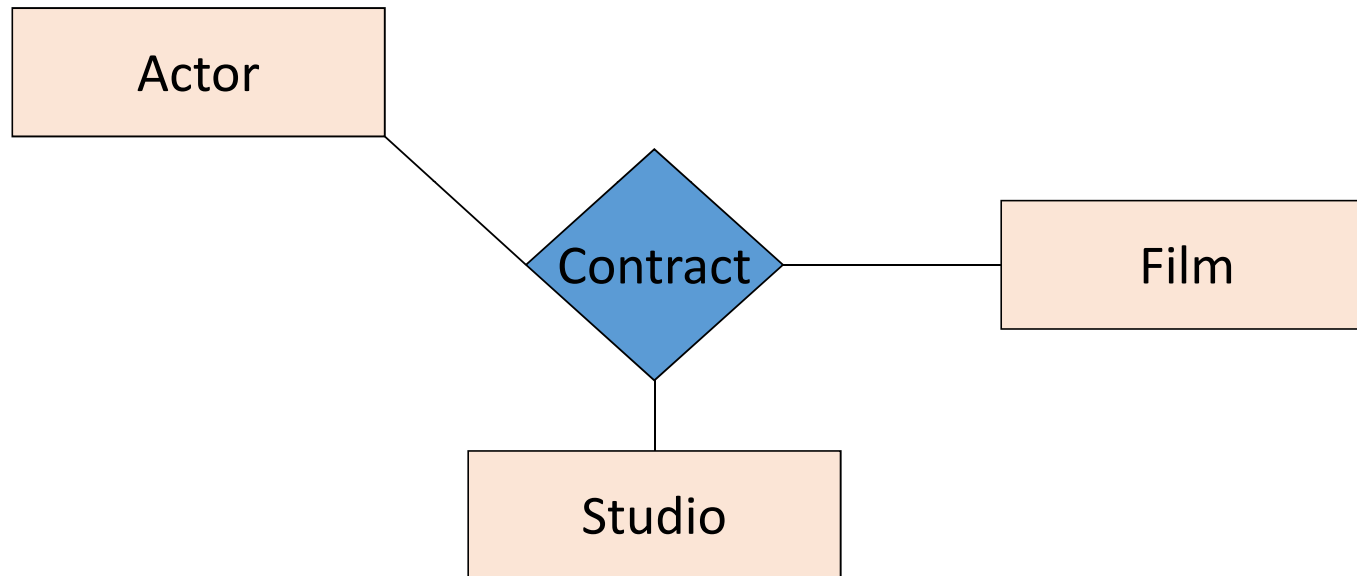
# Self Relationships or Recursive Relationships



# Multi-way Relationships

# Multi-way Relationships

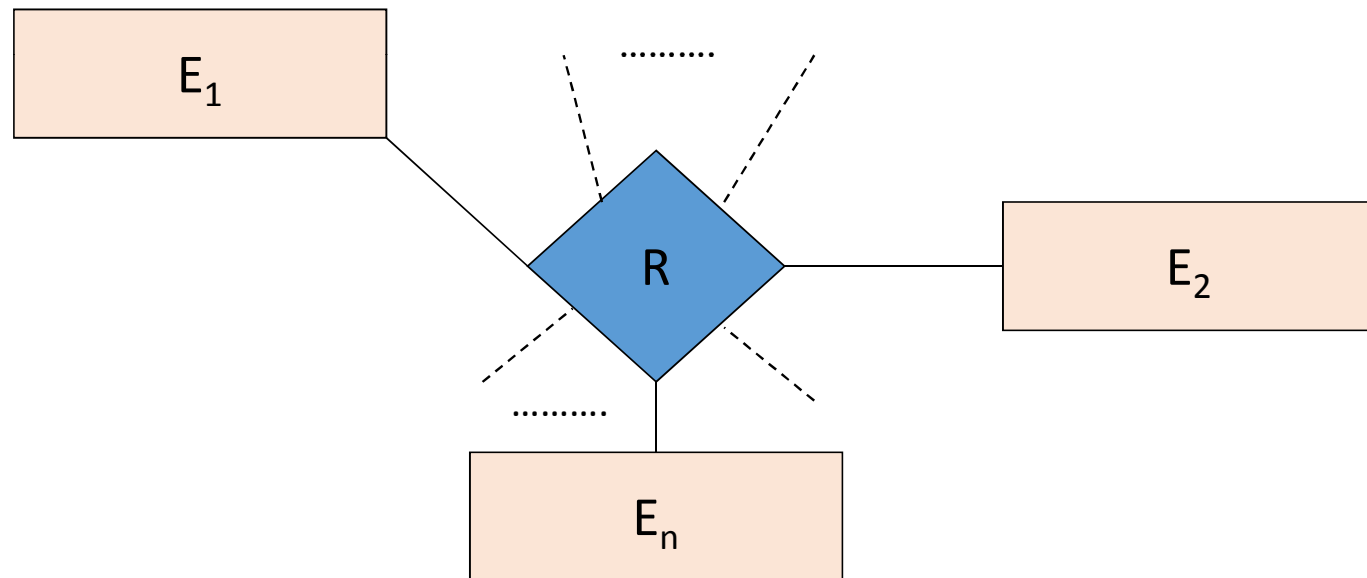
How do we model that a studio has made a contract with a particular actor to act in a particular movie?



NB: Can still model as a mathematical set (how?)

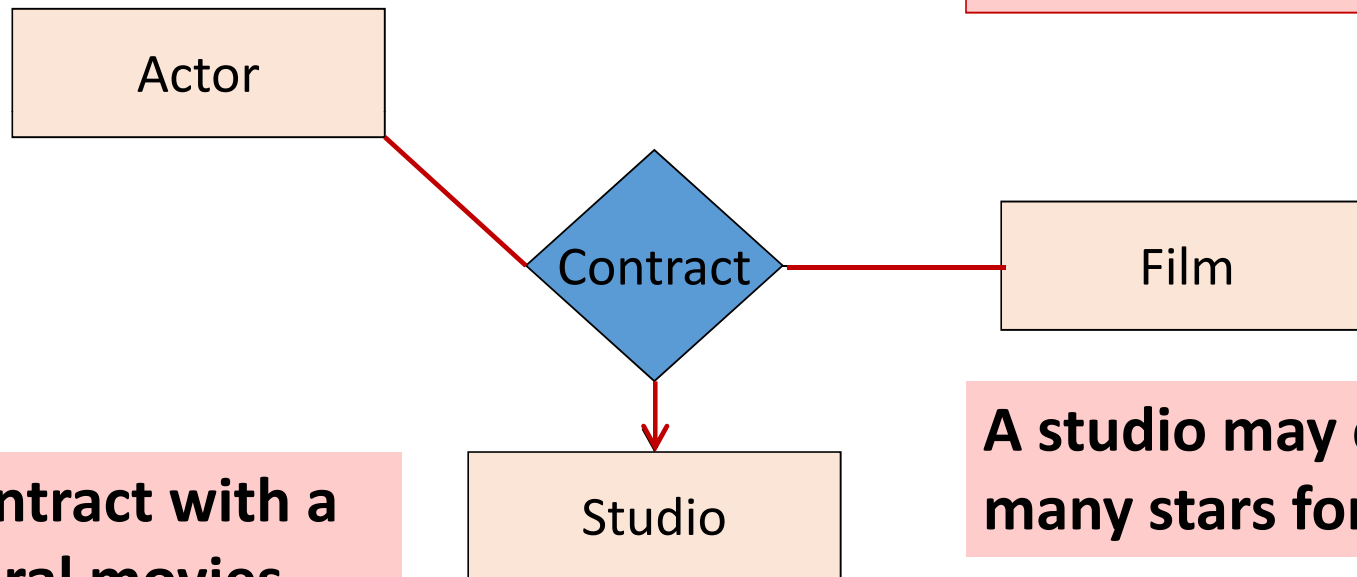
## Formal mathematical definition

- An  $n$ -ary relationship set  $R$  involves exactly  $n$  entity sets:  $E_1, \dots, E_n$ .
- Each relationship in  $R$  involves exactly  $n$  entities:  $e_1$  in  $E_1, \dots, e_n$  in  $E_n$
- Formally,  $R \subseteq E_1 \times \dots \times E_n$



# Arrows in Multiway Relationships

Q: What does the arrow mean ?



**For a particular star and movie there is only one studio under the contract**

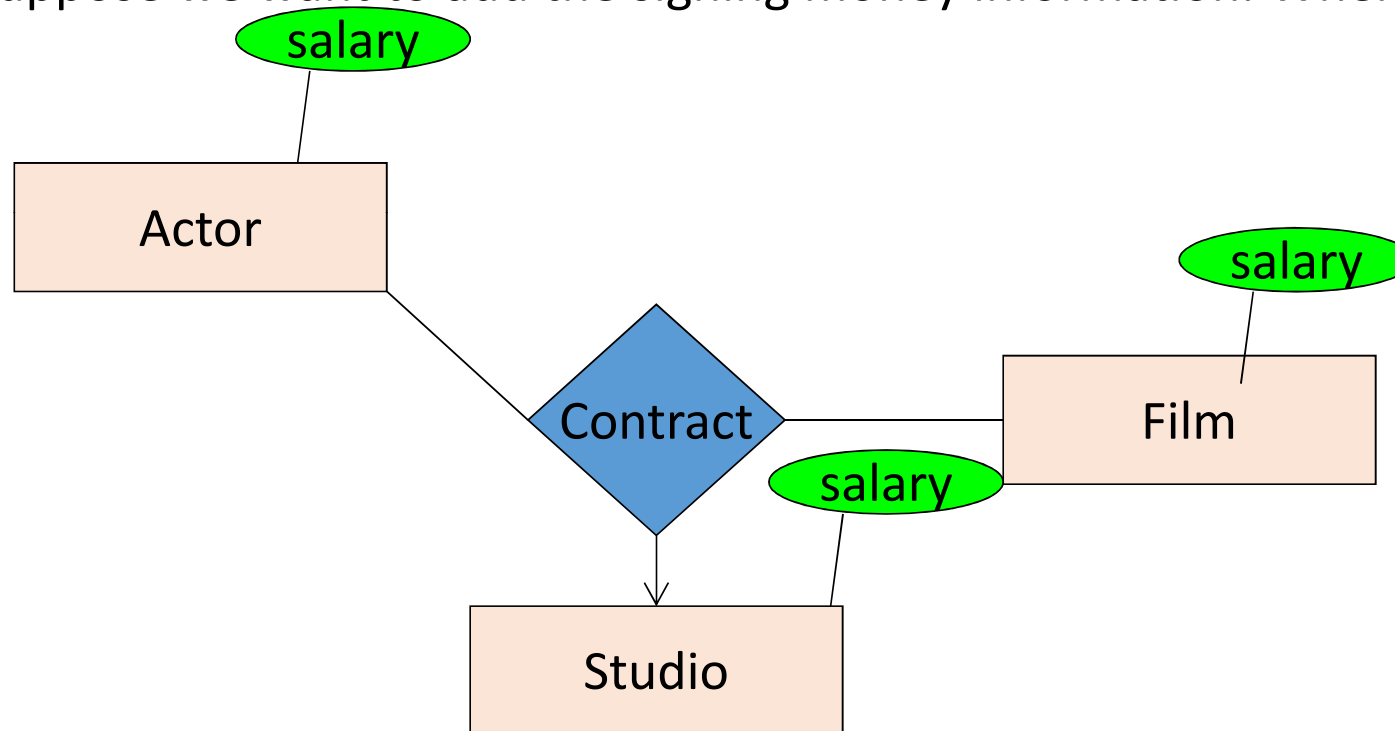
**A star may contract with a studio for several movies**

**A studio may contract with many stars for a movie**



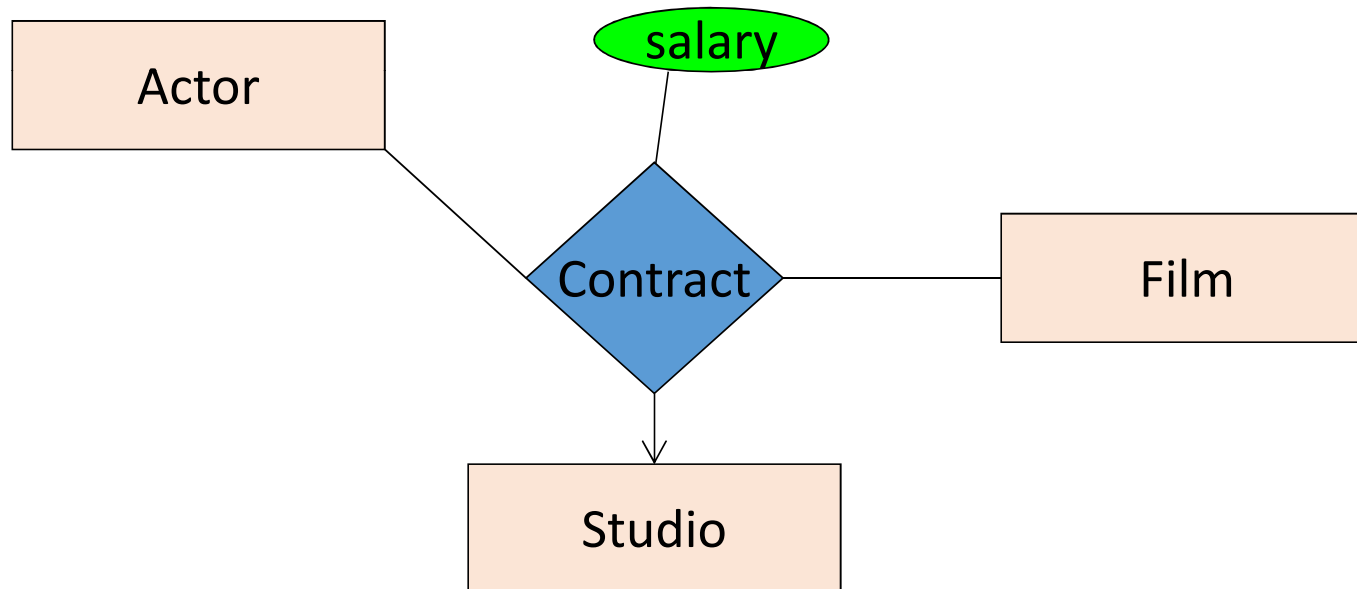
# Attributes in Relationships

Q: Suppose we want to add the signing money information. Where do we put?

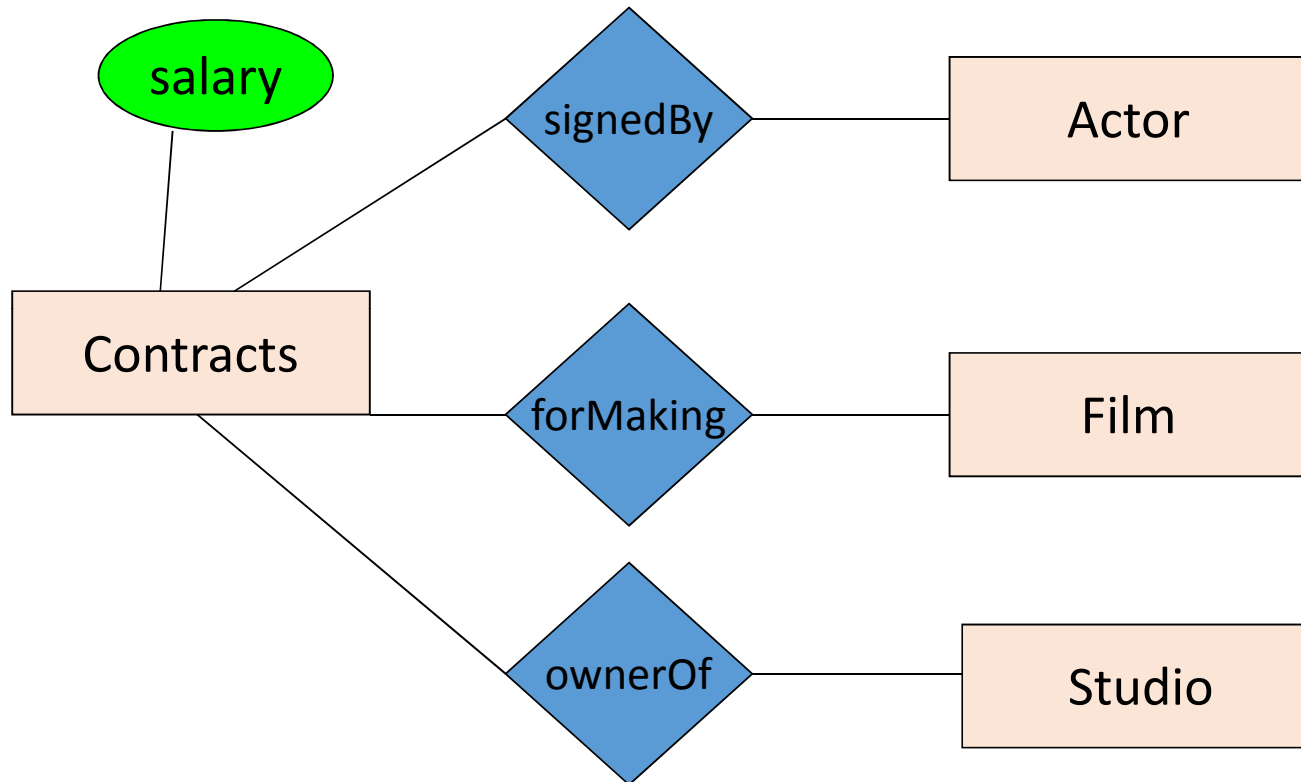


# Attributes in Relationships: The correct way

Q: Suppose we want to add the signing money information. Where do we put?

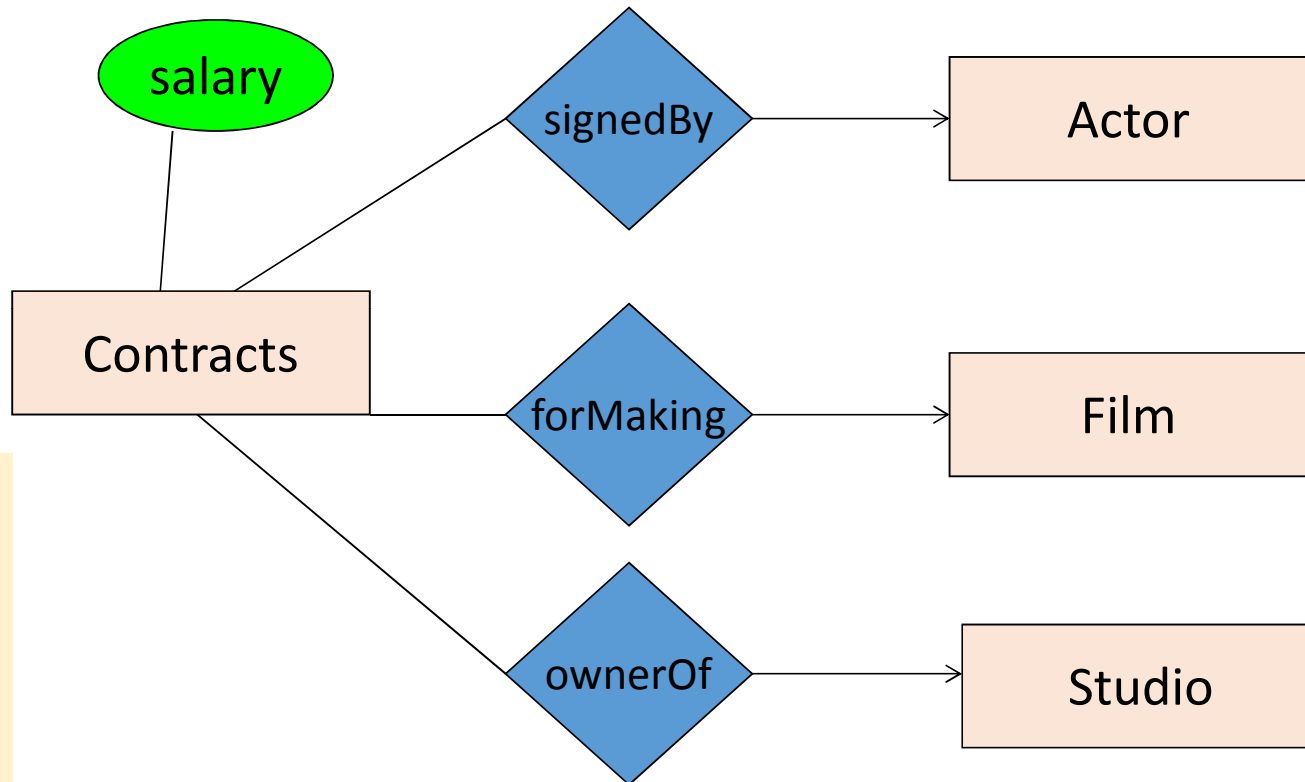


# Converting Multi-way Relationships to Binary Relations



From what we had on previous slide to this - what did we do?

# Converting Multi-way Relationships to New Entity + Binary Relationships



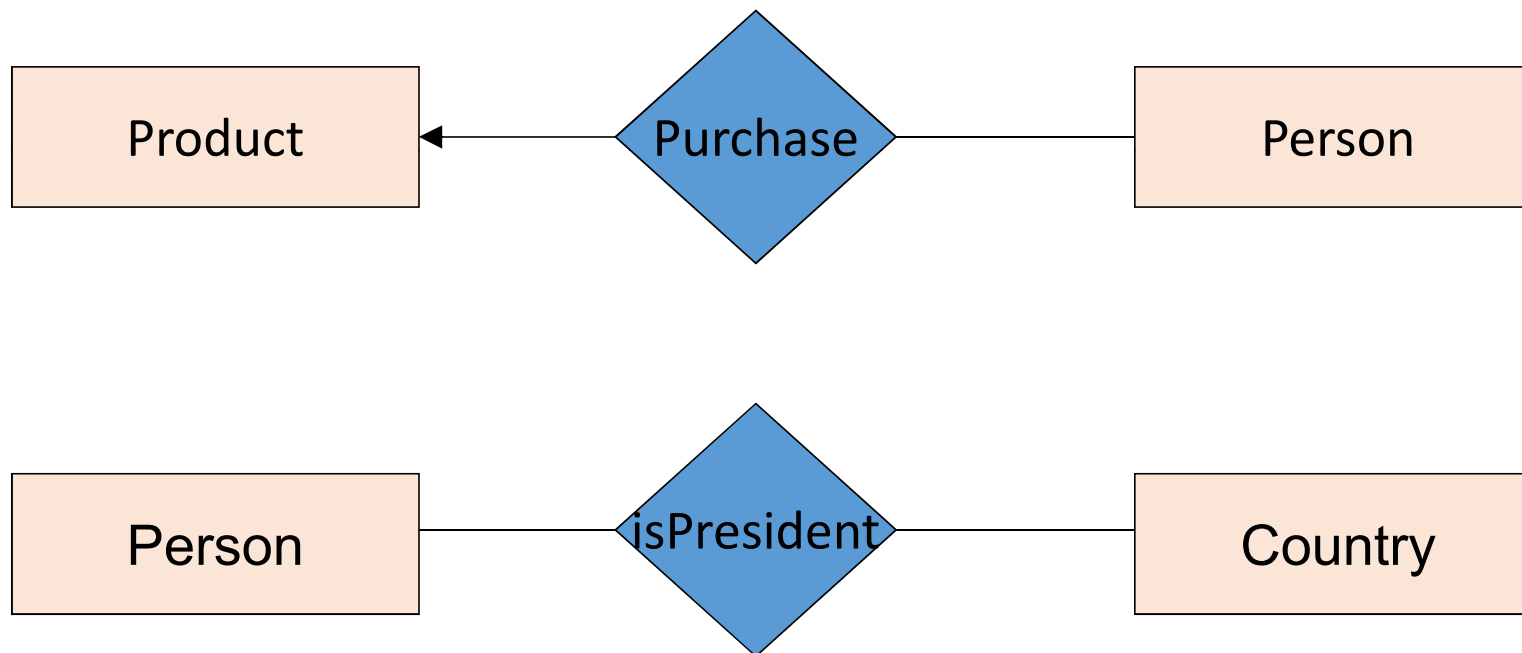
Side note:  
What arrows  
should be  
added here?  
Are these  
correct?

For a particular contract there can be only one star, one movie and one studio. But an actor or a movie or a studio may have several contracts

# 3. Design Principles

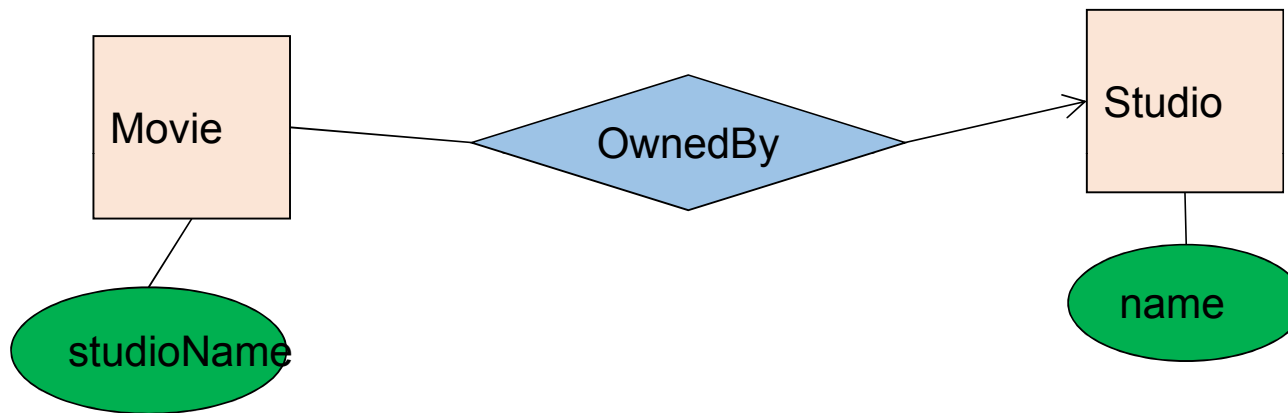
# Design Principles

What's wrong with these examples?



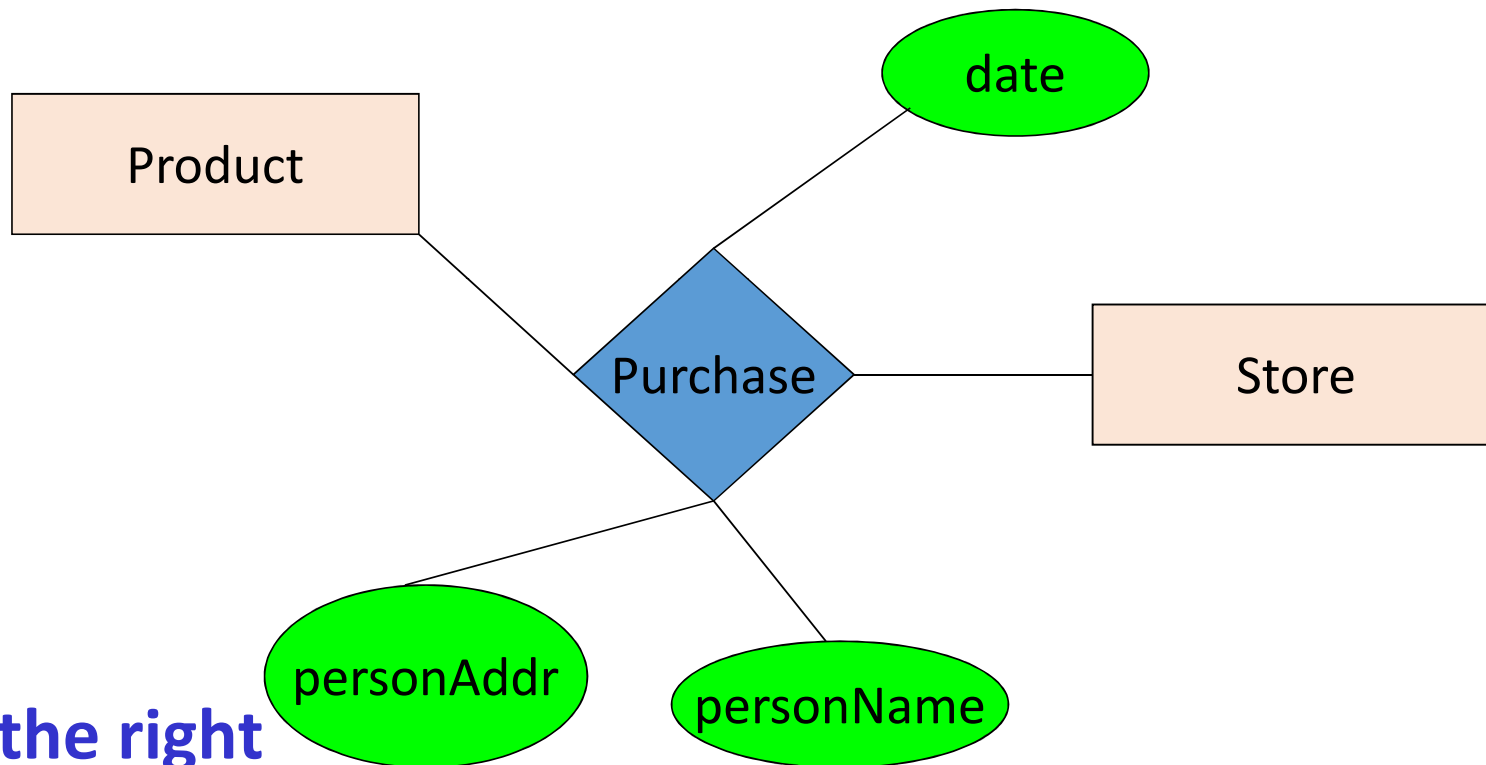
**Moral: be faithful!**

# Design Principles: What's Wrong?



**Moral: should not model the same information in multiple ways**

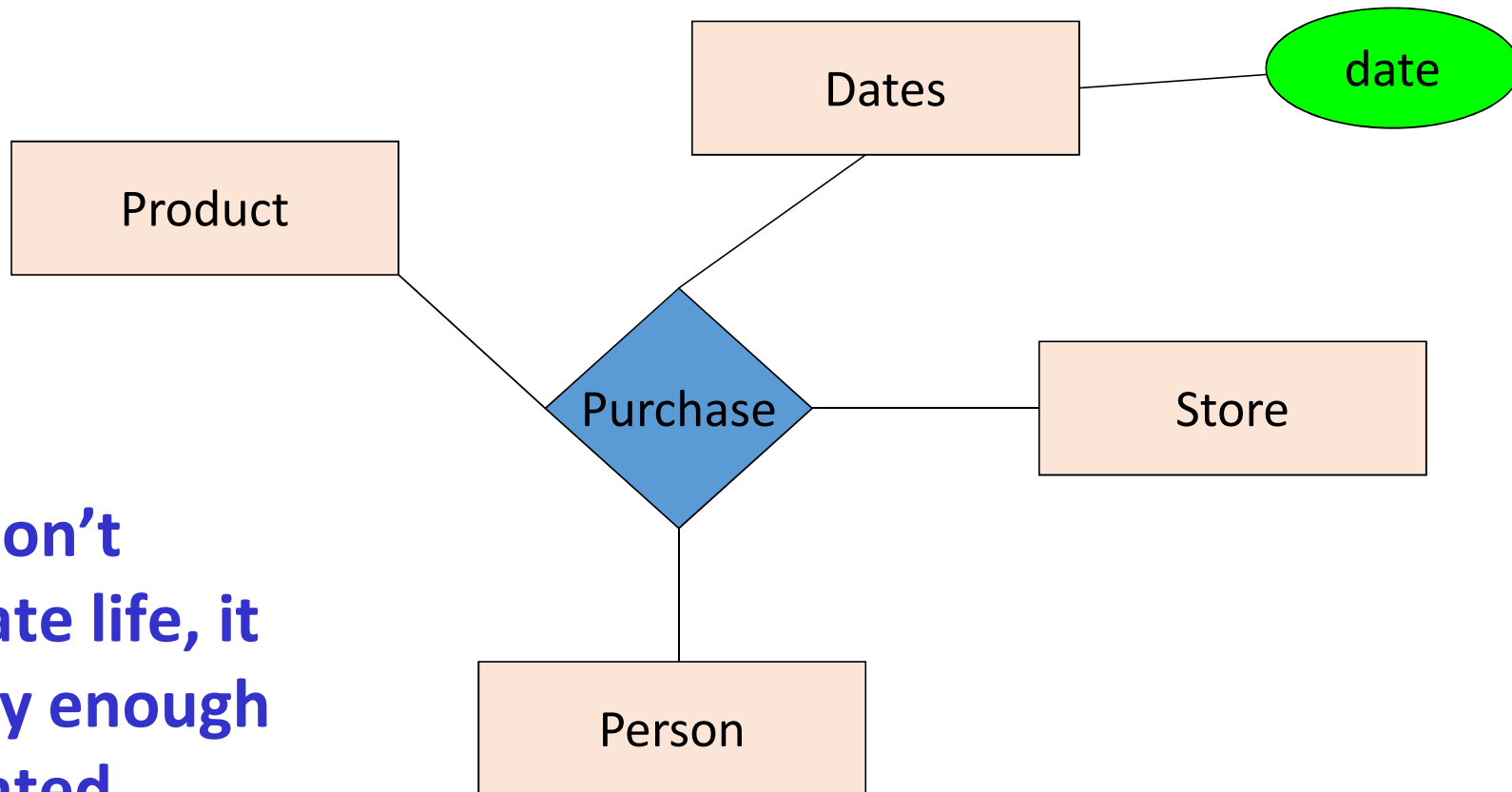
# Design Principles: What's Wrong?



**Moral: pick the right kind of elements (entities/relationships).**



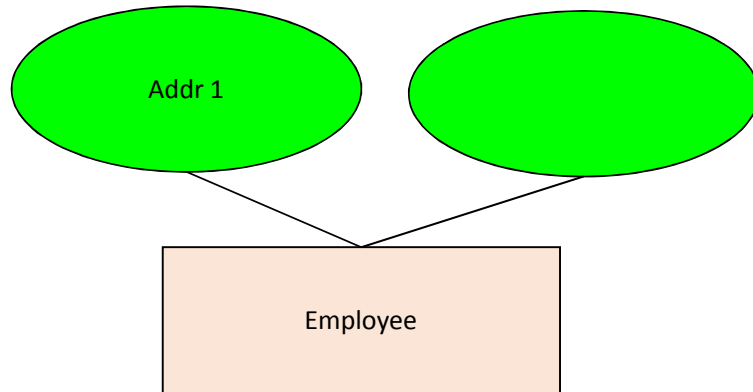
# Design Principles: What's Wrong?



**Moral: don't  
complicate life, it  
is already enough  
complicated.**

# Design Principles: Entity vs. Attribute

Should address  
(A) be an  
attribute?

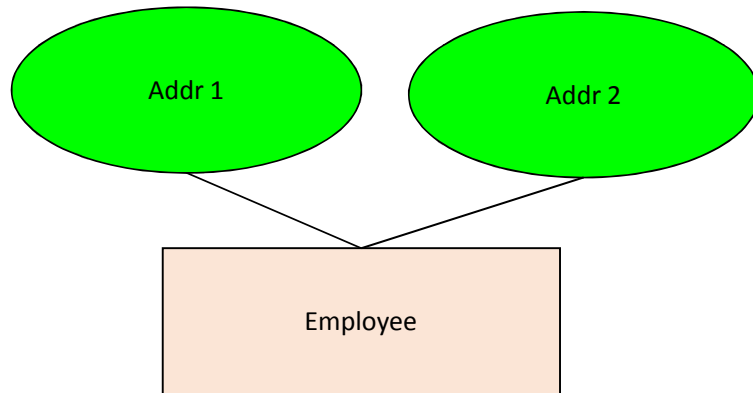


How do we handle  
employees with multiple  
addresses here?

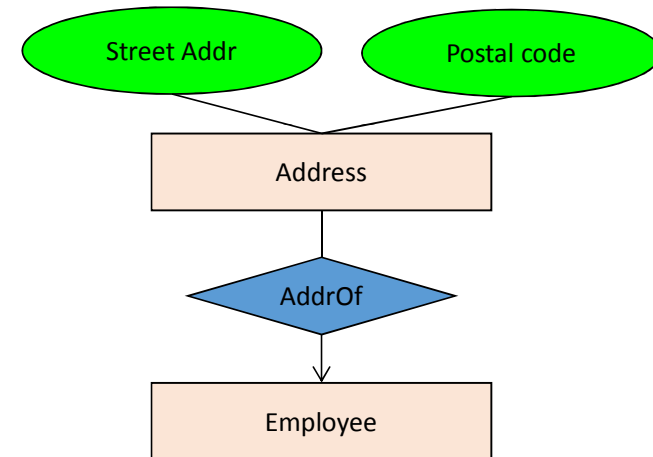
How do we handle  
addresses where internal  
structure of the address  
(e.g. postal code, city,  
division) is useful?

# Design Principles: Entity vs. Attribute

Should address  
(A) be an  
attribute?



Or (B) be an  
entity?



In general, when we want to record several values, we choose new entity

# From E/R Diagrams to Relational Schema

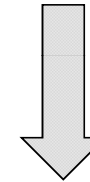
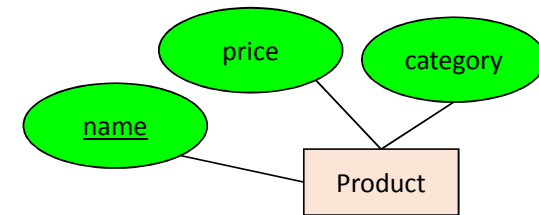
# From E/R Diagrams to Relational Schema

- Key concept:

Both ***Entity sets*** and ***Relationships*** become relations (tables in RDBMS)

# From E/R Diagrams to Relational Schema

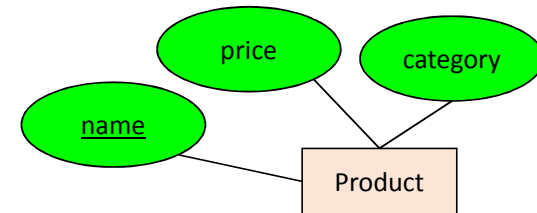
- An entity set becomes a relation (collection of tuples / table)
  - Each tuple is one entity
  - Each tuple is composed of the entity's attributes, and has the same primary key



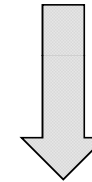
Product

<u>name</u>	price	category
Gizmo1	99.99	Camera
Gizmo2	19.99	Edible

# From E/R Diagrams to Relational Schema



```
CREATE TABLE Product(  
  name CHAR(50) PRIMARY KEY,  
  price FLOAT,  
  category VARCHAR(30)  
)
```



Product

<u>name</u>	price	category
Gizmo1	99.99	Camera
Gizmo2	19.99	Edible

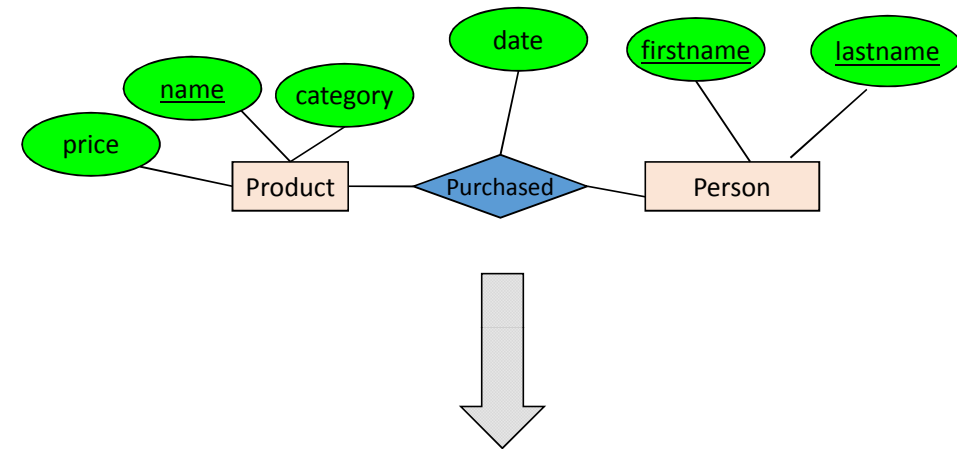
# From E/R Diagrams to Relational Schema

## Converting Many-to-Many Relationships to Relational Schema



# From E/R Diagrams to Relational Schema

- A relation between entity sets  $A_1, \dots, A_N$  *also* becomes a set of tuples / a table
  - Each row/tuple is one relation, i.e. one unique combination of entities  $(a_1, \dots, a_N)$
  - Each row/tuple is
    - composed of the **union of the entity sets' attributes**
    - has the entities' primary keys as foreign keys
    - has the union of the entity sets' keys as primary key



Purchased

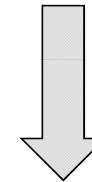
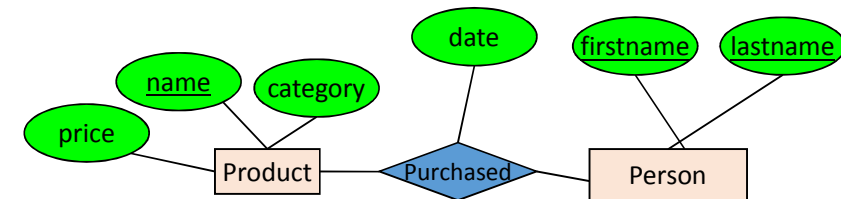
<u>name</u>	<u>firstname</u>	<u>lastname</u>	<u>date</u>
Gizmo1	Bob	Joe	01/01/15
Gizmo2	Joe	Bob	01/03/15
Gizmo1	JoeBob	Smith	01/05/15

# From E/R Diagrams to Relational Schema

```

CREATE TABLE Purchased(
  name CHAR(50),
  firstname CHAR(50),
  lastname CHAR(50),
  date DATE,
  PRIMARY KEY (name, firstname, lastname),
  FOREIGN KEY (name)
    REFERENCES Product(name),
  FOREIGN KEY (firstname, lastname)
    REFERENCES Person(firstname, lastname)
)

```



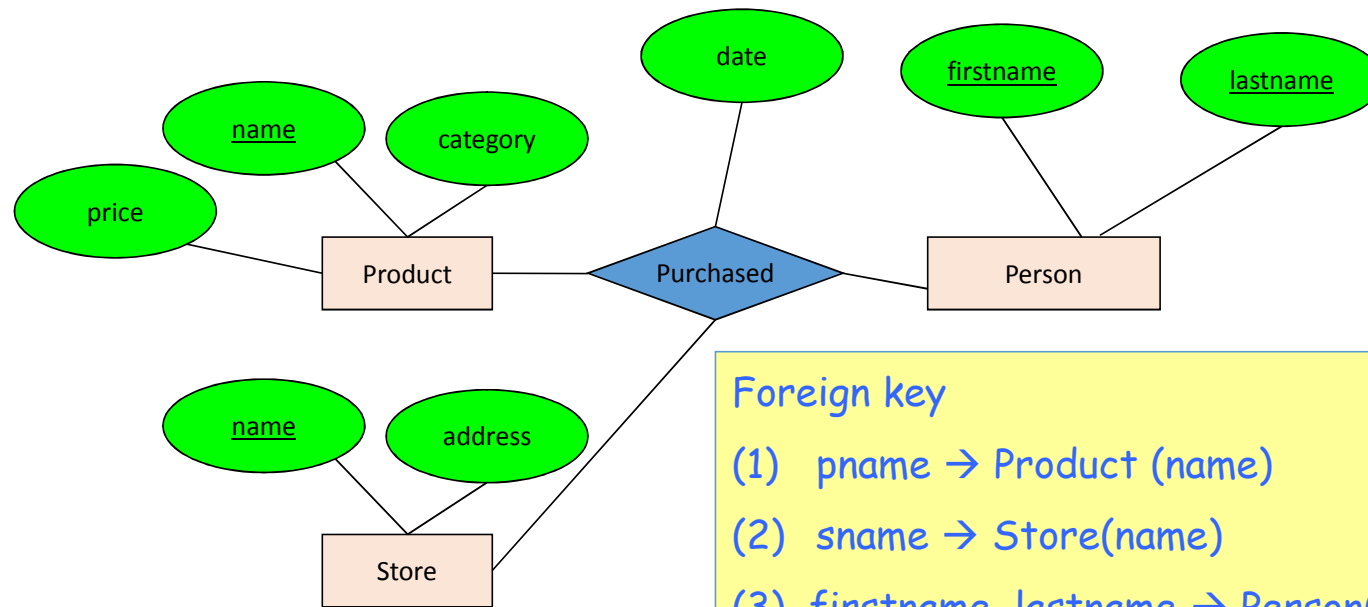
Purchased

<u>name</u>	<u>firstname</u>	<u>lastname</u>	date
Gizmo1	Bob	Joe	01/01/15
Gizmo2	Joe	Bob	01/03/15
Gizmo1	JoeBob	Smith	01/05/15

# From E/R Diagram to Relational Schema

How do we represent this as a relational schema?

Purchased (pname, sname, firstname, lastname, date)



## Foreign key

- (1) pname → Product (name)
- (2) sname → Store(name)
- (3) firstname, lastname → Person(firstname, lastname)

# From E/R Diagrams to Relational Schema

## Converting One-to-Many Relationships to Relational Schema

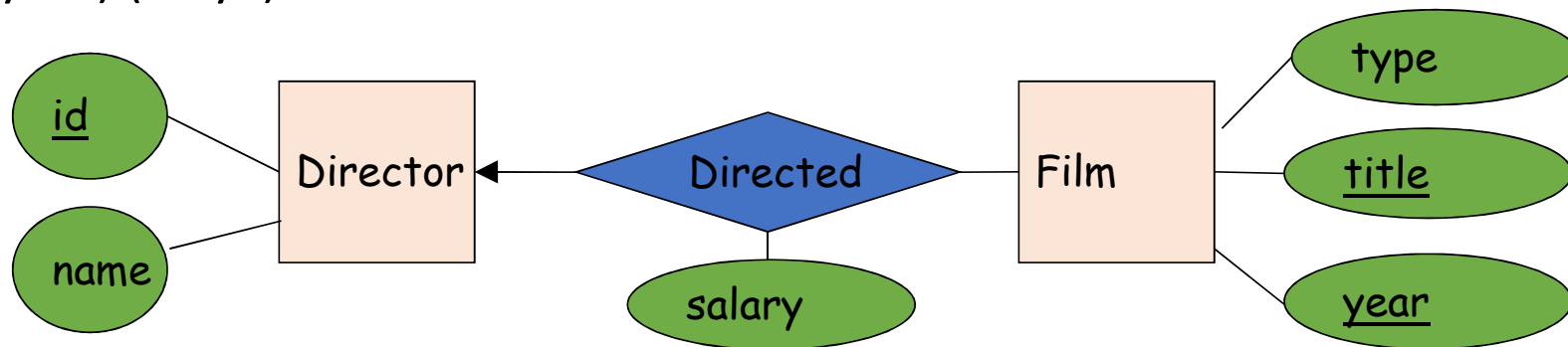
# From E/R Diagram to Relational Schema

- For many to one relations we have two options:

## Option 1:

Same as without key constraints (3 tables), except that the primary key of Directed is now (title, year) (why?)

```
Film (title, year, type)
Director (id, name)
Directed (title, year, id, salary)
```



# From E/R Diagram to Relational Schema

```
CREATE TABLE Film(  
  title CHAR(50),  
  year INT,  
  type CHAR(40),  
  PRIMARY KEY (title, year)  
)
```

```
CREATE TABLE Director(  
  id INT,  
  name CHAR(40),  
  PRIMARY KEY (id)  
)
```

Film (title, year, type)

Director (id, name)

Directed (title, year, id, salary)

```
CREATE TABLE Directed(  
  title CHAR(50),  
  year INT,  
  id INT,  
  salary FLOAT,  
  PRIMARY KEY (title, year),  
  FOREIGN KEY (title, year)  
    REFERENCES Film (title, year),  
  FOREIGN KEY (id)  
    REFERENCES Director(id)  
)
```

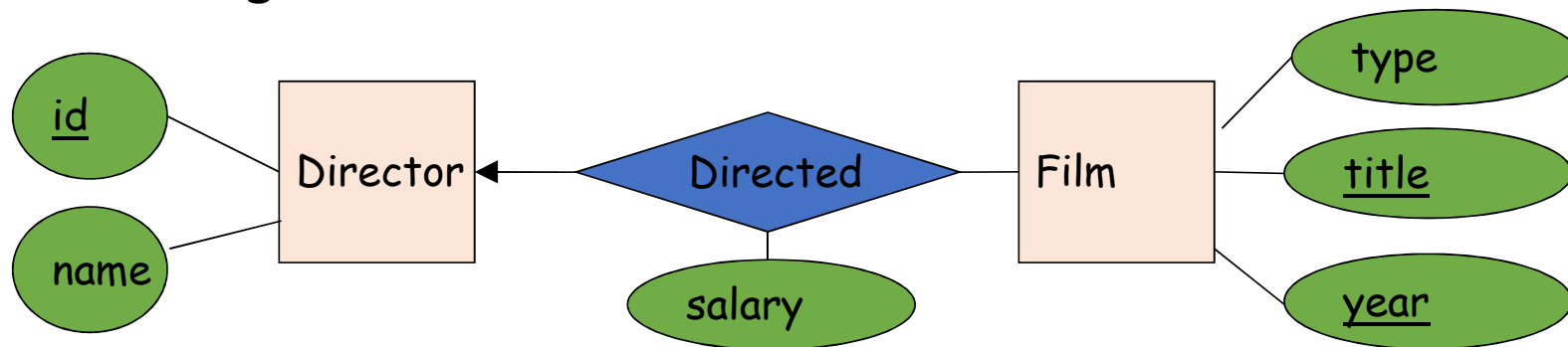
# From E/R Diagram to Relational Schema

## Option 2:

- Do not create a table for the relationship
- Add information columns that would have been in the relationship's relation to the relation of the **entity which does not have the incoming arrow**

Film (title, year, type, id, salary)

Director (id, name)



# From E/R Diagram to Relational Schema

```
CREATE TABLE Film(  
  title CHAR(50),  
  year INT,  
  type CHAR(40),  
  id INT,  
  salary FLOAT,  
  PRIMARY KEY (title, year),  
  FOREIGN KEY (id)  
    REFERENCES Director(id)  
)
```

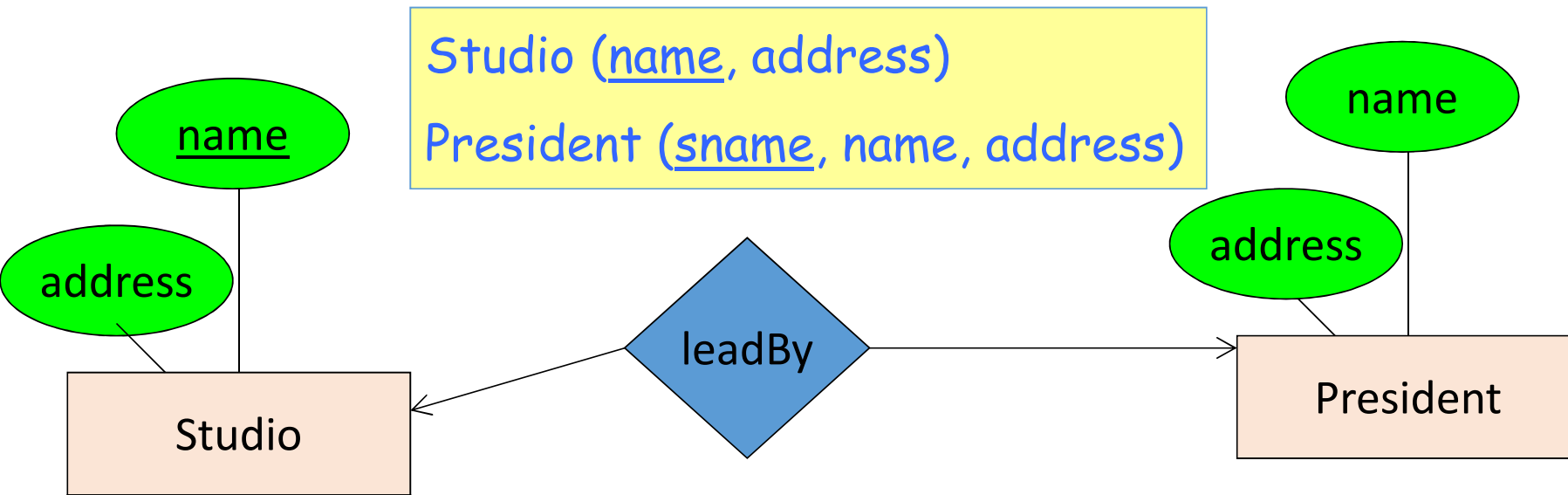
```
Film (title, year, type, id, salary)  
Director (id, name)
```

```
CREATE TABLE Director(  
  id INT,  
  name CHAR(40),  
  PRIMARY KEY (id)  
)
```



# From E/R Diagrams to Relational Schema

## Converting (1:1) Relationships to Relational Schema



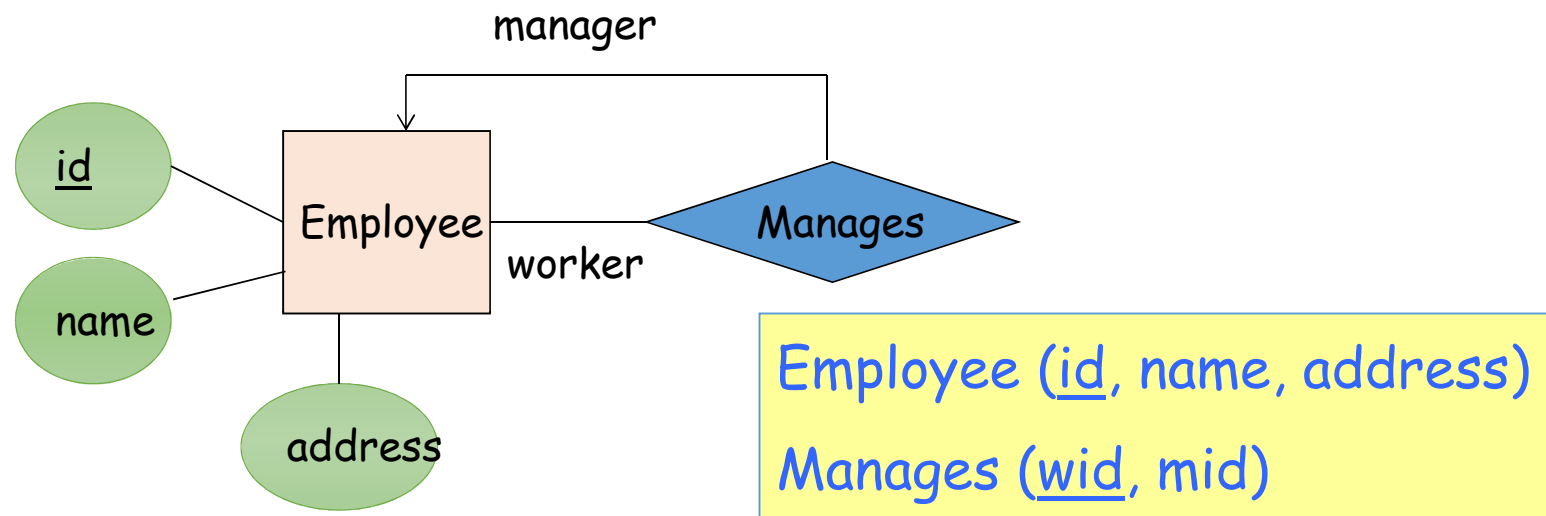
```

CREATE TABLE Studio(
  name CHAR(40),
  address CHAR(40),
  PRIMARY KEY (name)
)
  
```

```

CREATE TABLE President(
  sname CHAR(40),
  name CHAR(40),
  address VARCHAR(50),
  PRIMARY KEY (sname),
  FOREIGN KEY (sname)
    REFERENCES Studio(name)
)
  
```

# Translating Recursive Relationship Sets (without constraints)



What are all the relations created for this diagram?