Lecture 4

Entity-Relationship Model

(part 2)

ER模型

（第2部分）

Outline

- From E/R Diagrams to Relational Designs
- Converting Subclass Structures to Relations

Outline

- From E/R Diagrams to Relational Designs
- Converting Subclass Structures to Relations

From E/R Diagrams to Relational Designs

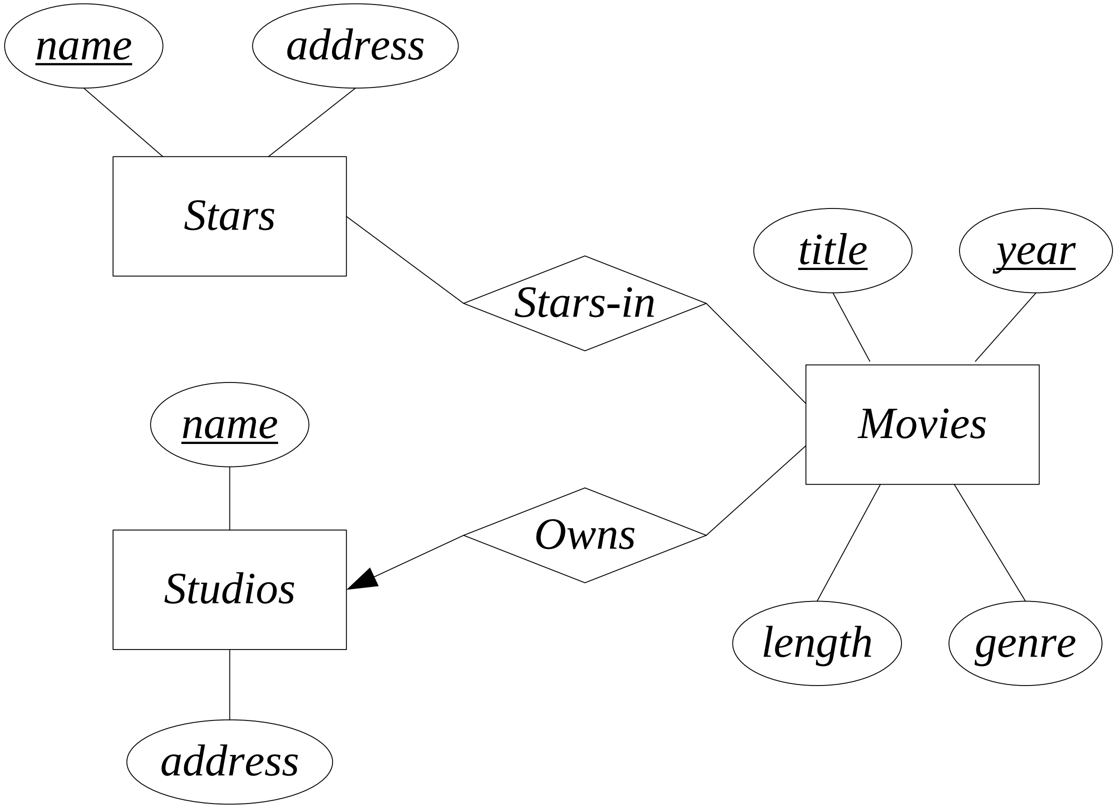
- To a first approximation
  - Turn each entity set into a relation with the same set of attributes, and
  - Replace a relationship by a relation whose attributes are the keys for the connected entity sets.

From E/R Diagrams to Relational Designs

- Special situations
  - Weak entity sets cannot be translated straightforwardly to relations
  - "Isa" relationships and subclasses require careful treatment
  - Sometimes, we do well to combine two relations, especially the relation for an entity set *E* and the relation that comes from a many-one relationship from *E* to some other entity set

From Entity Sets to Relations

- For each non-weak entity set
  - Create a relation of the same name and with the same set of attributes



From Entity Sets to Relations

- For each non-weak entity set
  - Create a relation of the same name and with the same set of attributes

Movies (title, year, length, genre)

title	*year*	*length*	*genre*
Star Wars	1977	124	sciFi
Gone With the Wind	1939	231	drama
Wayne’s World	1992	95	comedy

From Entity Sets to Relations

- For each non-weak entity set
  - Create a relation of the same name and with the same set of attributes

Stars (name, address)

name	*address*
Carrie Fisher	123 Maple St., Hollywood
Mark Hamill	456 Oak Rd., Brentwood
Harrison Ford	789 Palm Dr., Beverly Hills

From E/R Relationships to Relations

- Relationships → Relations

  The relation for a given relationship *R*

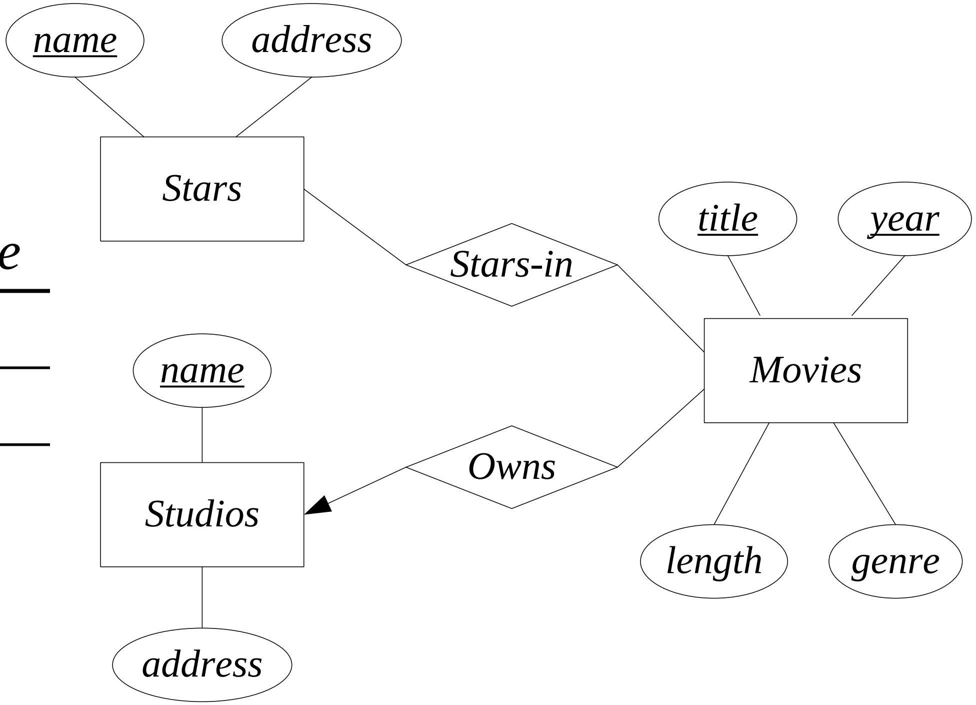
  1. For each entity set involved in relationship *R*, we take its key attribute or attributes as part of the schema of the relation for *R*
  2. If the relationship has attributes, then these are also attributes of relation for *R*

From E/R Relationships to Relations

- Relationships → Relations

Owns (title, year, studioName)

title	*year*	*studioName*
Star Wars	1977	Fox
Gone With the Wind	1939	MGM
Wayne's World	1992	Paramount

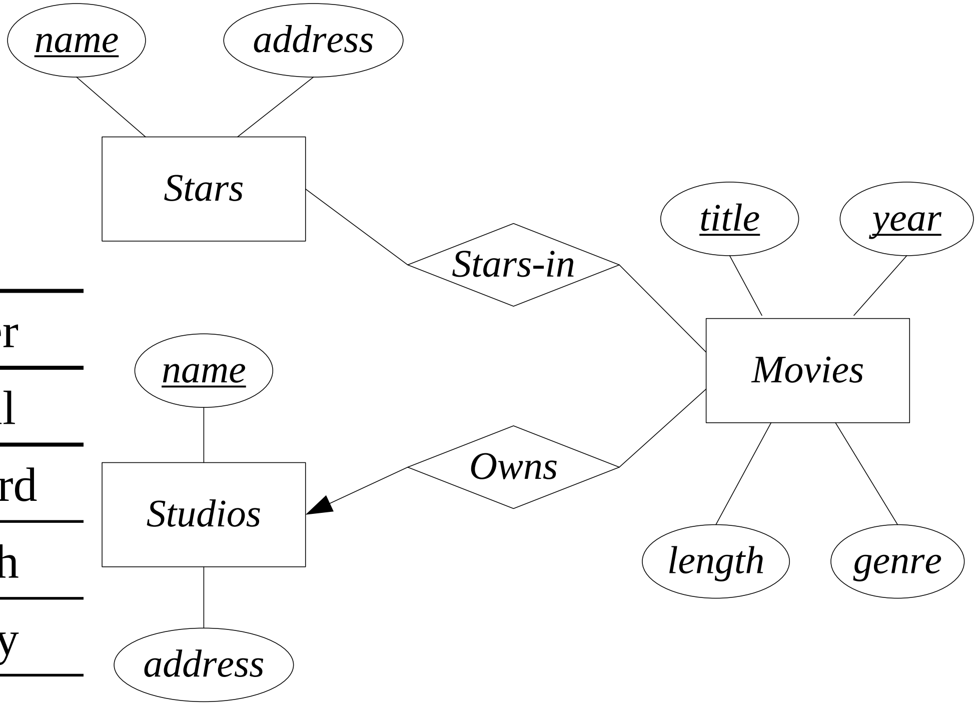


From E/R Relationships to Relations

- Relationships → Relations

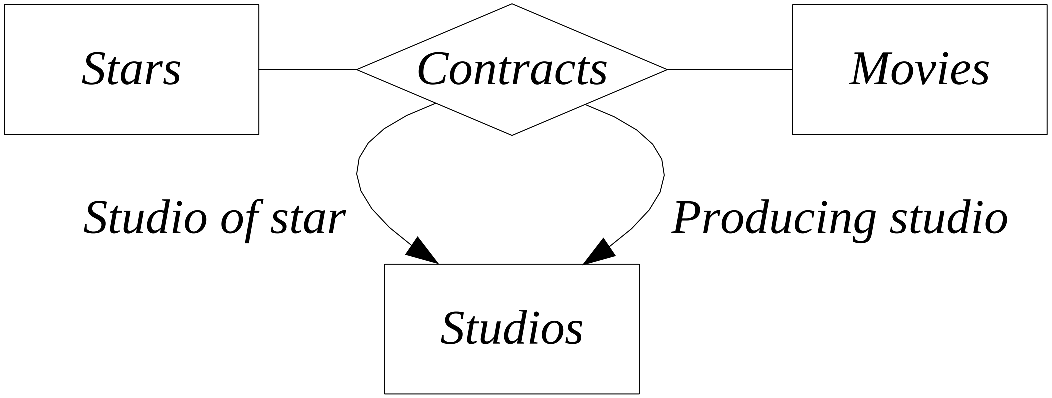
StarsIn (title, year, starName)

title	*year*	*starName*
Star Wars	1977	Carrie Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
Gone With the Wind	1939	Vivien Leigh
Wayne's World	1992	Dana Carvey
Wayne's World	1992	Mike Meyers



From E/R Relationships to Relations

- Relationships → Relations



Contracts (starName, title, year, studioOfStar, producingStudio)

Combining Relations

- One common situation
  - An entity set *E* with a many-one relationship *R* from *E* to *F*
  - Combine *E* and *F* into one relation with a schema consisting of
    1. All attributes of *E*
    2. The key attributes of *F*
    3. Any attributes belonging to relationship *R*
  - For an entity *e* of *E* that is not related to any entity of *F*, the attributes of type (2) and (3) will have null values in the tuple for *e*.

Combining Relations

- Example

Owns (title, year, studioName)

title	*year*	*studioName*
Star Wars	1977	Fox
Gone With the Wind	1939	MGM
Wayne’s World	1992	Paramount

Movies (title, year, length, genre)

title	*year*	*length*	*genre*
Star Wars	1977	124	sciFi
Gone With the Wind	1939	231	drama
Wayne’s World	1992	95	comedy

Movies (title, year, length, genre, studioName)

title	*year*	*length*	*genre*	*studioName*
Star Wars	1977	124	sciFi	Fox
Gone With the Wind	1939	231	drama	MGM
Wayne’s World	1992	95	comedy	Paramount

Combining Relations

StarsIn (title, year, starName)

title	*year*	*starName*
Star Wars	1977	Carrie Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
Gone With the Wind	1939	Vivien Leigh
Wayne’s World	1992	Dana Carvey
Wayne’s World	1992	Mike Meyers

Movies (title, year, length, genre, studioName)

title	*year*	*length*	*genre*	*studioName*
Star Wars	1977	124	sciFi	Fox
Gone With the Wind	1939	231	drama	MGM
Wayne’s World	1992	95	comedy	Paramount

Movies (title, year, length, genre, studioName, starName)

title	*year*	*length*	*genre*	*studioName*	*starName*
Star Wars	1977	124	sciFi	Fox	Carrie Fisher
Star Wars	1977	124	sciFi	Fox	Mark Hamill
Star Wars	1977	124	sciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne’s World	1992	95	comedy	Paramount	Dana Carvey
Wayne’s World	1992	95	comedy	Paramount	Mike Meyers

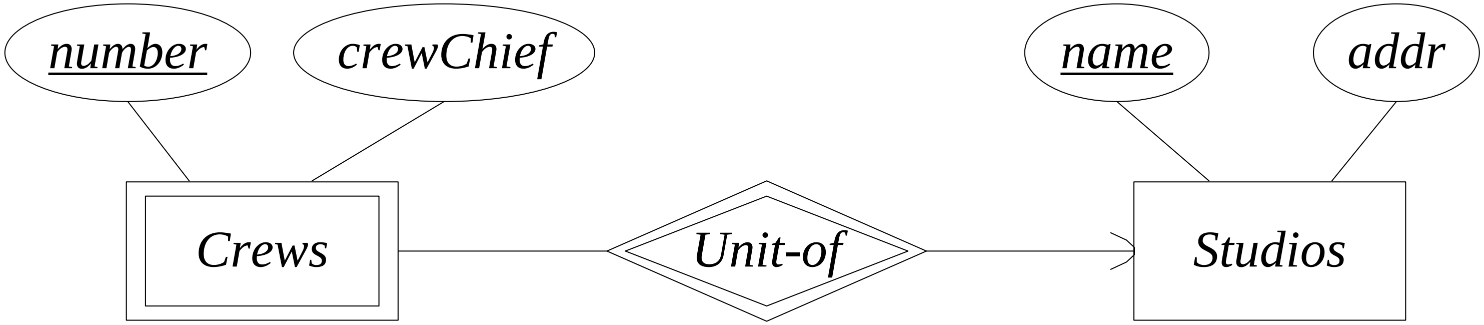
Bad Design

Handling Weak Entity Sets

- To do three things
  1. The relation for the weak entity set *W* itself must include not only the attributes of *W* but also the key attributes of the supporting entity sets.
  2. The relation for any relationship in which the weak entity set *W* appears must use as a key for *W* all of its key attributes, including those of other entity sets that contribute to *W*’s key.
  3. However, a supporting relationship *R*, from the weak entity set *W* to a supporting entity set, need not to be converted to a relation at all.

Handling Weak Entity Sets

- Example



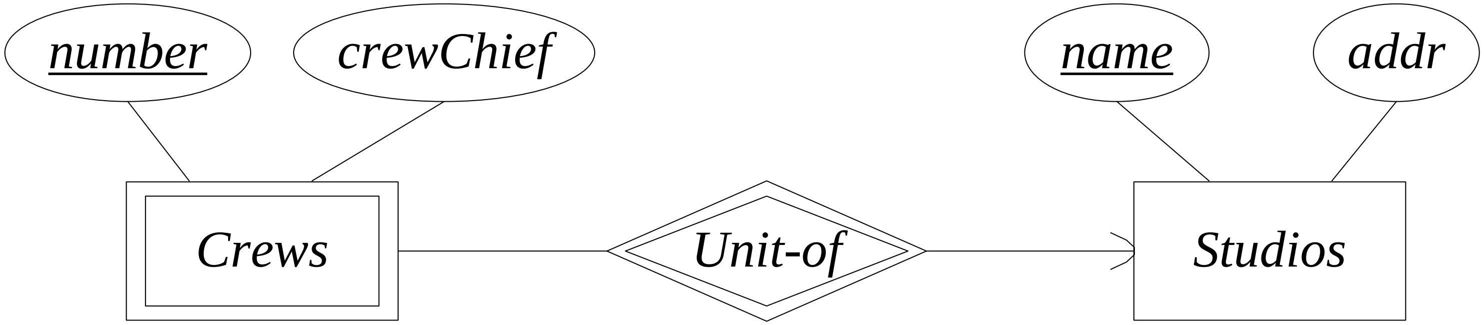
Studio (name, addr)

Crews (number, studioName, crewChief)

~~Unit-of (number, studioName, name)~~

Handling Weak Entity Sets

- Example



Studio (name, addr)

Crews (number, studioName, crewChief)

A supporting relationship needs no relation.

Handling Weak Entity Sets

- Modified rules
  - If W is a weak entity set, construct for *W* a relation whose schema consists of:
    1. All attributes of *W*
    2. All attributes of supporting relationships for *W*
    3. For each supporting relationships for *W*, say a many-one relationship from *W* to entity set *E*, all the key attributes of *E*

    Rename attributes, if necessary, to avoid name conflicts
  - Do not construct a relation for any supporting relationship for *W*

Outline

- From E/R Diagrams to Relational Designs
- Converting Subclass Structures to Relations

Converting Subclass Structures to Relations

- The principal conversion strategies

  1. Follow the E/R viewpoint

  2. Treat entities as objects belonging to a single class

  3. Use null values

E/R-Style Conversion

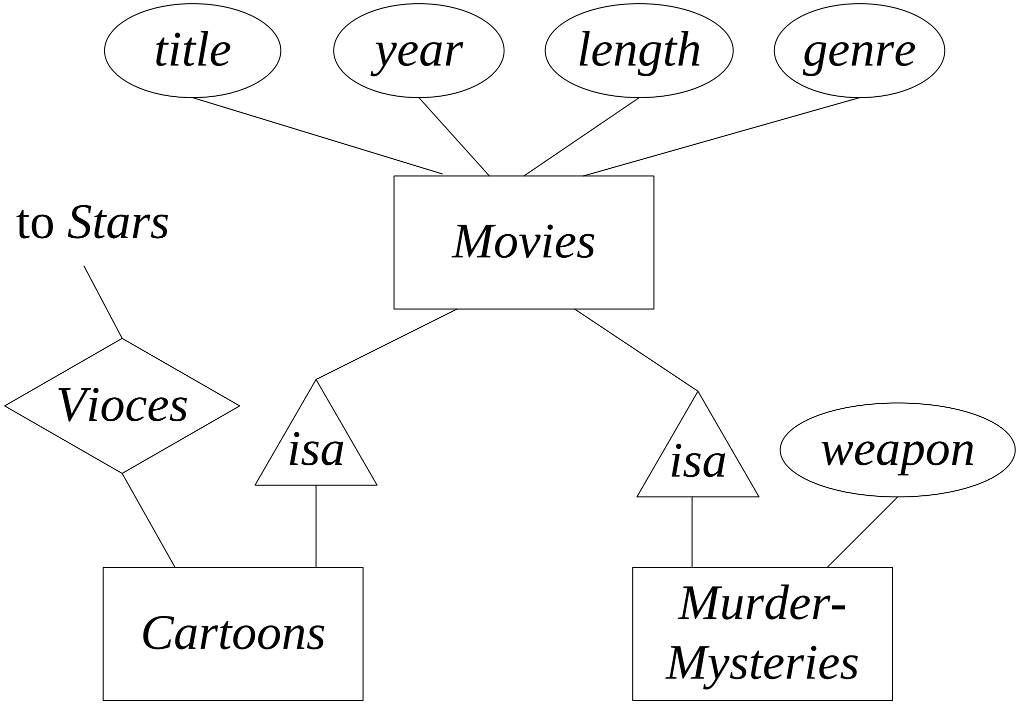
- The approach

1. Create a relation for each entity set, as usual.

2. If the entity set *E* is not the root of the hierarchy, then the relation for *E* will include the key attributes at the root, to identify the entity represented by each tuple, plus all the attributes of *E*.

E/R-Style Conversion

- Example



Movies (title, year, length, genre)

Voices (title, year, starName)

MurderMysteries (title, year, weapon)

Cartoon (title, year)

An Object-Oriented Approach

- The approach
  1. Enumerate all the possible subtrees that includes the root.

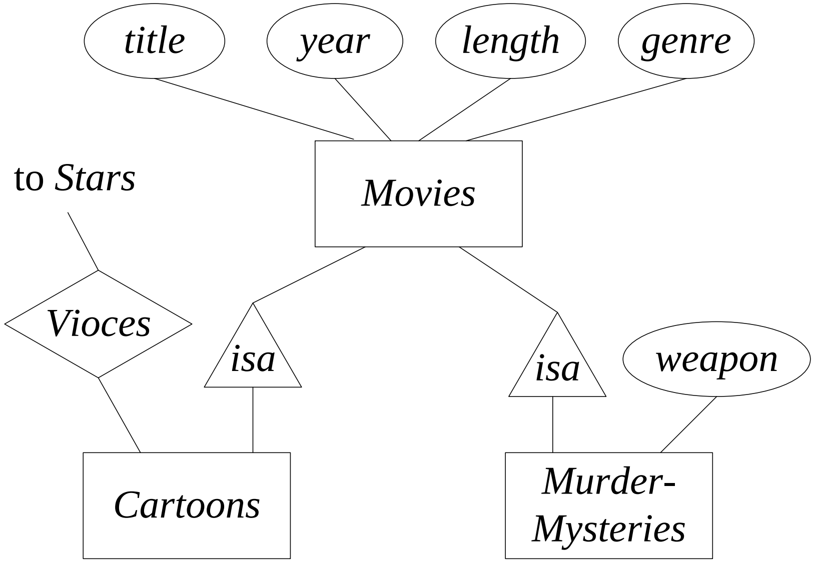
  2. For each, create one relation that represents entities having components in exactly that subtree.

  3. The schema for this relation has all the attributes of any entity set in the subtree.

  The assumption that entities are “objects” that belong to one and only one class.

An Object-Oriented Approach

- Example
  - – Four possible subtrees
    1. *Movies* alone
    2. *Movies* and *Cartoons*
    3. *Movies* and *Murder-Mysteries*
    4. All three entity sets



Movies (title, year, length, genre)

MoviesC (title, year, length, genre)

MoviesMM (title, year, length, genre, weapon)

MoviesCMM (title, year, length, genre, weapon)

Using Null Values to Combine Relations

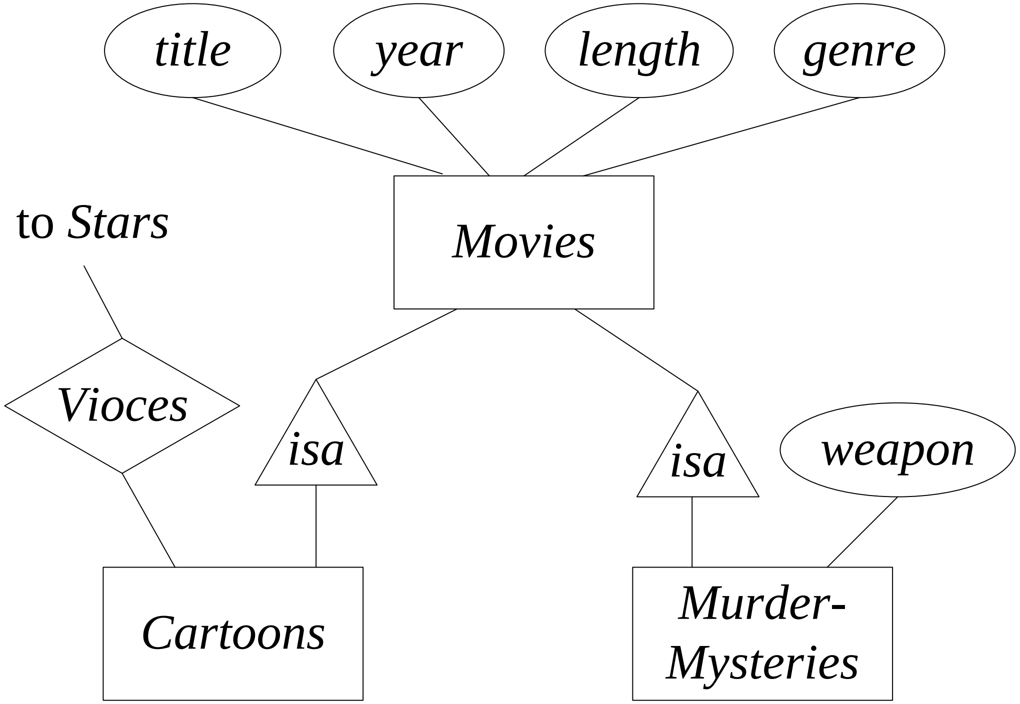
- The Approach

1. Create one relation with all the attributes of all the entity sets in the hierarch.

2. Each entity is represented by one tuple, and that tuple has a null value for whatever attributes the entity does not have.

Using Null Values to Combine Relations

- Example



Movies (title, year, length, genre, weapon)

Comparison of Approaches

- “straight-E/R”, “object-oriented”, and “nulls” (p169)
  1. Answering queries
  2. Not to use too many relations
  3. Minimize space and avoid repeating information

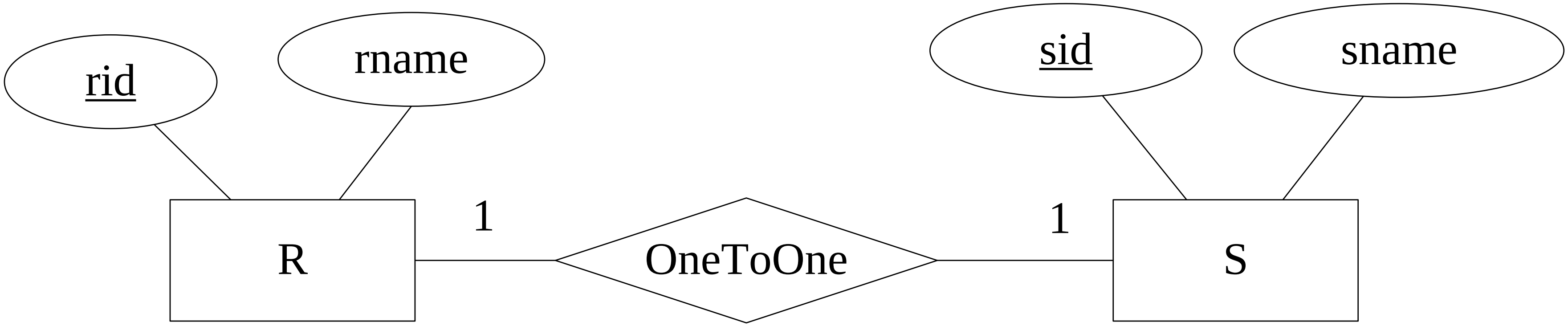
转换规则：ER模型→关系模型

关系模型

- 转换规则：ER模型→关系模型

1. 一个实体转换为一张表;
2. **一个“一对一”联系，可分配到实体表中（分配到哪个实体表都可以）；**
3. 一个“一对多”**联系**，既可分配到实体表中（要分配到“多”的实体表一边），又可用单独的关系表实现;
4. **一个“多对多”联系，必须用单独的关系表实现。两个实体表的主键合起来作关系表的主键。**

一对一



R

rid	rname	sid

S

sid	sname

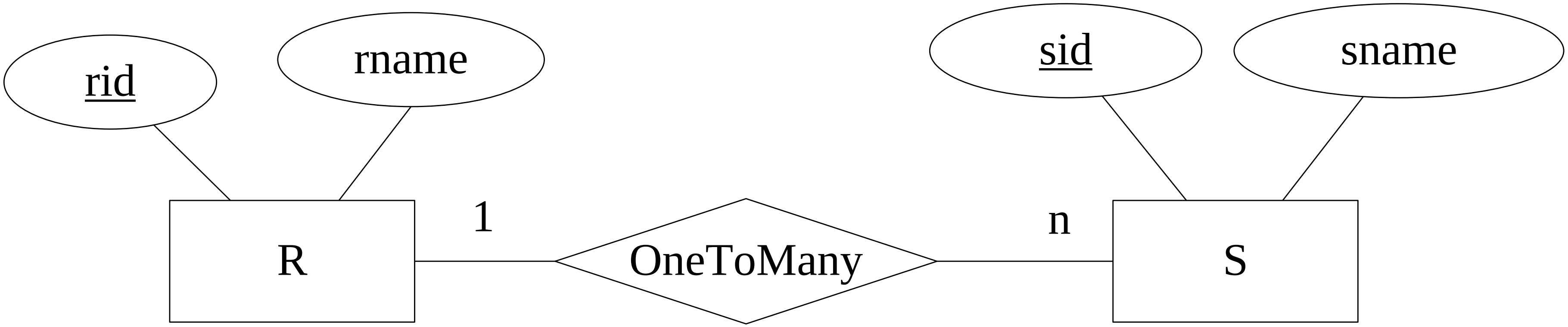
R

rid	rname

S

sid	sname	rid

一对多



R

rid	rname

S

sid	sname	rid

R

rid	rname

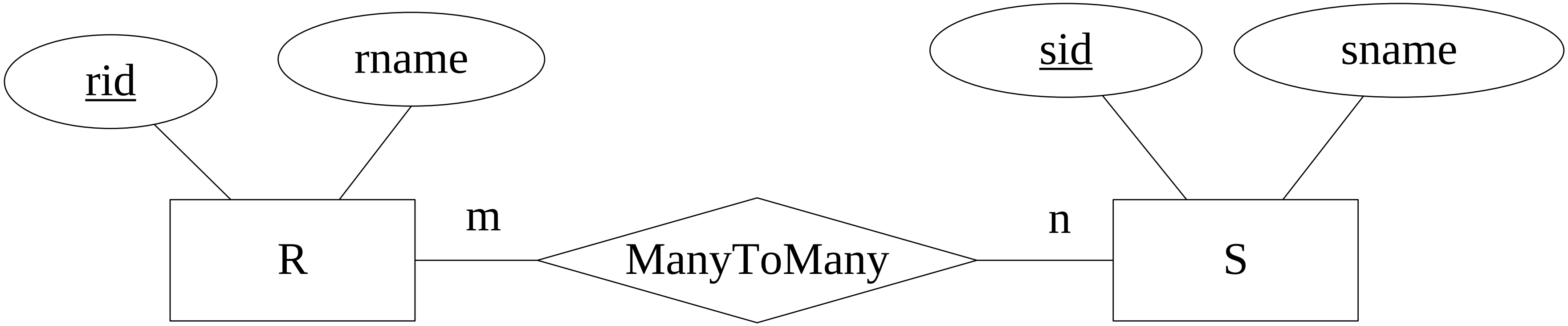
S

sid	sname

OneToMany

sid	rid

多对多



R

rid	rname

S

sid	sname

ManyToMany

rid	sid

转换规则：ER模型→关系模型

	表	属性（左表）	属性（右表）
实体	√	×	×
一对一	√	√	√
一对多	√	×	√
多对多	√	×	×

…The End of This Lecture…