



Преобразуване от Е/Р модел
към релационен модел

Въведение в релационния модел

- ▶ Докато E/R подходът е подход за моделирани, при който се описва структурата на данните, то релационния подход лесно представя тази структура в базата от данни
- ▶ Релационният модел е изключително полезен, защото представя данните в таблици. Той е и сравнително опростен, защото всичко в модела се свежда до релация (таблица)
- ▶ Друго предимство на релационния модел е че поддържа езици от високо ниво, посредством които могат да бъдат моделирани данните в базата от данни.
- ▶ Въпреки това не трябва да забравяме, че е добра практика дизайна на базата от данни да се направи, като се използва E/R модела и след това така направения модел да се сведе (преобразува) до релационен. Релациите могат да бъдат създадени, посредством езика от високо ниво – SQL в базата от данни и съответно попълнени с данни.

Основни понятия в реляционния модел

- ▶ **Реляционният модел** представя данните в двумерни таблици наречени релации.
- ▶ **Атрибути** на релация са имената на колоните. Те описват съдържанието на колоните.
- ▶ **Схема на релация** – наричаме името на релацията последвана от списък с атрибутите към релацията. Например $R(A_1, A_2, A_3, \dots, A_n)$.
- ▶ **Схема на базата от данни** – наричаме множество от всички реляционни схеми в базата от данни.
- ▶ **Кортежи** – Редовете на релацията се наричат кортежи. Кортежите имат по една стойност за всеки атрибут от релацията.
- ▶ **Домейн** наричаме множеството от допустими стойности за даден атрибут. Домейна е част от реляционната схема.
- ▶ Реляционният модел изисква всички стойности в кортежите да са атомарни. Не е позволена стойността да е съставна – множество, списък, масив, структура и други.

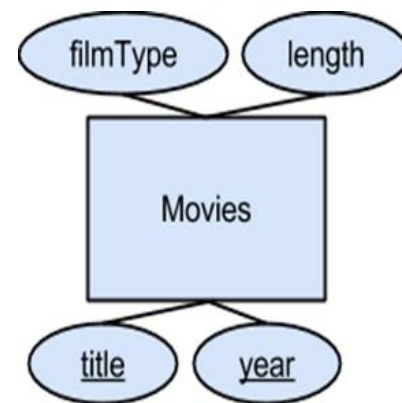
Преобразуване на E/R към релационен модел

- ▶ При преобразуването на E/R модел към релационен се спазват следните правила:
 - ▶ Множество от същности се свежда до релация
 - ▶ Връзка се свежда до релация
 - ▶ Слабо множество от същности не може директно да бъде сведено до релация
 - ▶ При 1:n връзка се изисква по специфично преобразуване
 - ▶ При връзки много-един, може да оптимизираме модела, като обединим двете релации за връзката и за множеството в което влиза стрелката един

Преобразуване на множества от същности

- ▶ Да разгледаме преобразуване на **множество от същности**
- ▶ В релационен модел множеството от същности се свежда до релация с име, името на множеството и атрибути, атрибутите на множеството.
- ▶ Ново-получената релация, няма да носи индикация за връзките, в които множеството участва.
- ▶ Ако един атрибут или група от атрибути е ключ за множеството от същности, то е ключ и за ново-получената релация. Ключът в релационната схема се подчертава.

Movies(title, year, length, filmType)



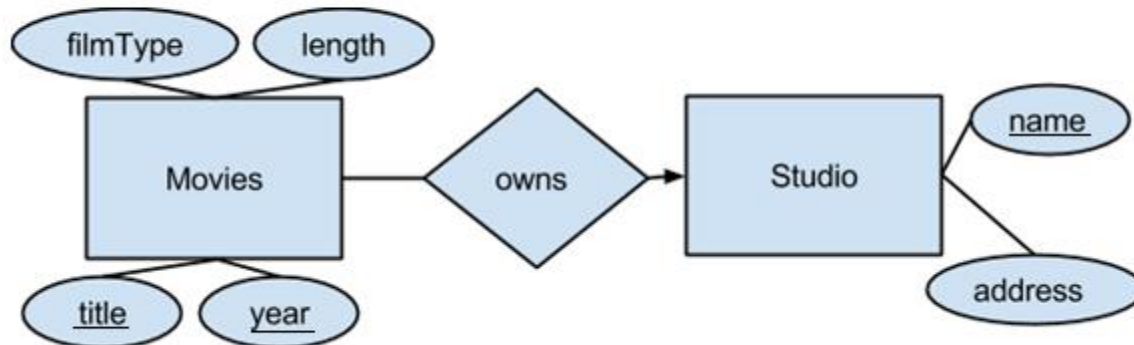
Преобразуване на връзки

- ▶ Да разгледаме преобразуването на **връзки**
- ▶ Връзките в релационен модел, също се представят чрез релация.
- ▶ Релацията за дадена връзка R има следните атрибути:
 - ▶ За всяко множество от същности, участващо във връзката R взимаме ключовите атрибути и те стават атрибути на релацията R
 - ▶ Ако връзката има атрибут, тогава те също са атрибути на релацията
 - ▶ Ако едно множество от същности участва няколко пъти но в различни роли, тогава неговите ключови атрибути, ще се появяват толкова пъти в релацията R , в колкото роли участва множеството.
 - ▶ Ако е необходимо, може да преименуваме атрибутите за да избегнем дублиране на имената им.

Преобразуване на връзки - пример

- ▶ Връзката Owns се преобразува до релационната схема:

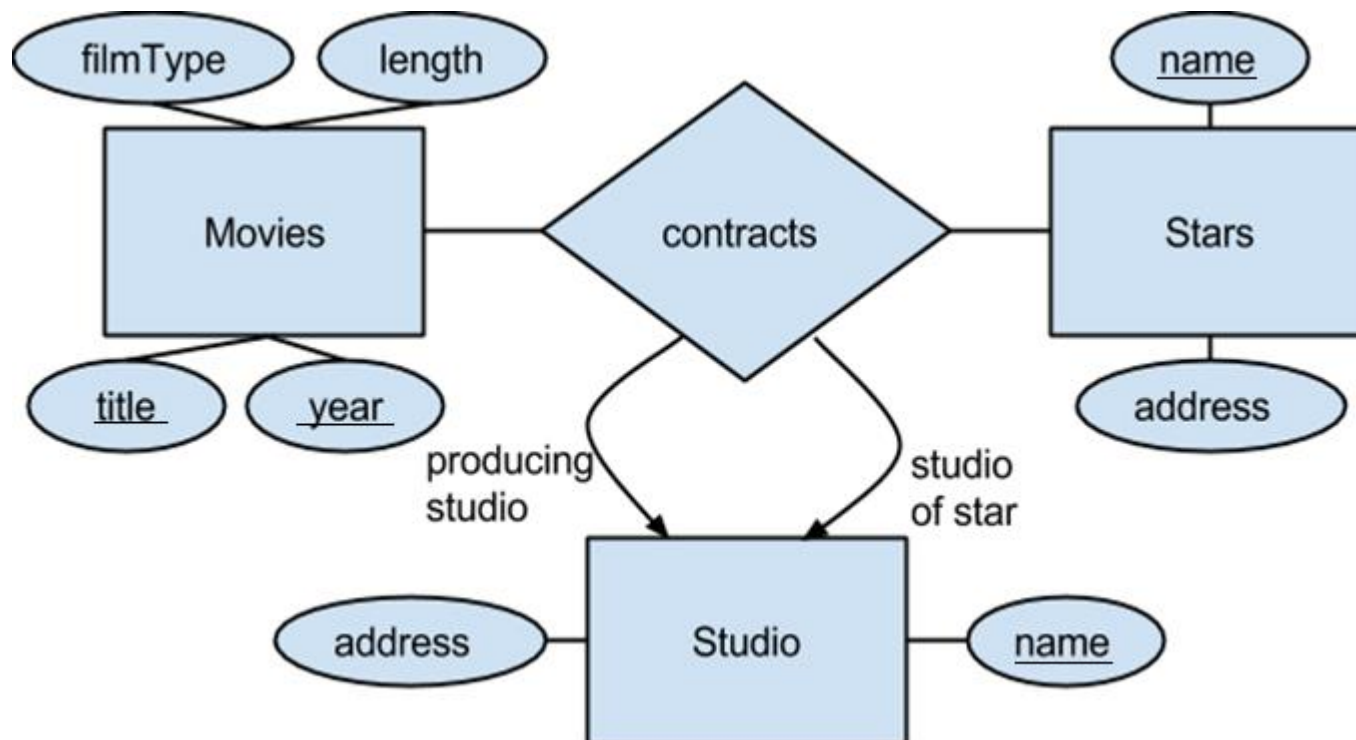
Owns(title, year, studioName)



Преобразуване на връзки с роли - пример

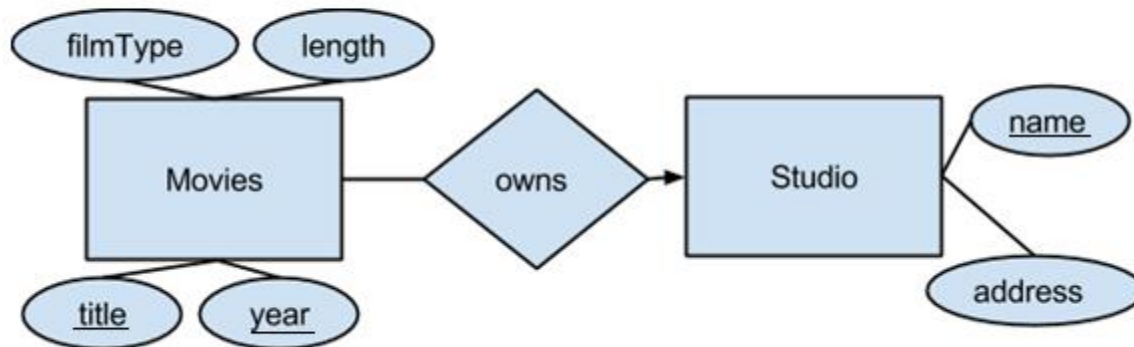
- Връзката Contracts се преобразува до релационната схема:

Contracts(title, year, starName, studioStar, studioProducing)



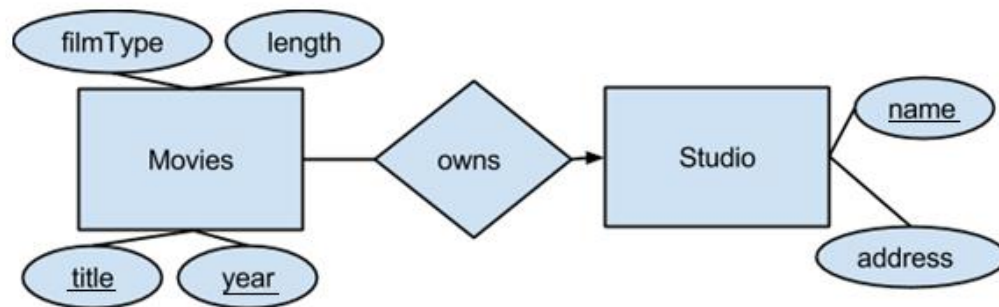
Комбиниране на релации

- ▶ Понякога релациите, които получаваме от преобразуването на множества от същности и връзки към релационен модел, могат да бъдат оптимизирани.
- ▶ Такива ситуации възникват, когато връзките са много-един или един – един
- ▶ Пример за комбиниране на релации



Комбиниране на релации - пример

- Задача: Преобразувайте E/R диаграмата към релационни схеми



- Решение:

Movies(title, year, length, filmType)

Studio(name, address)

Owns(title, year, studioName) X

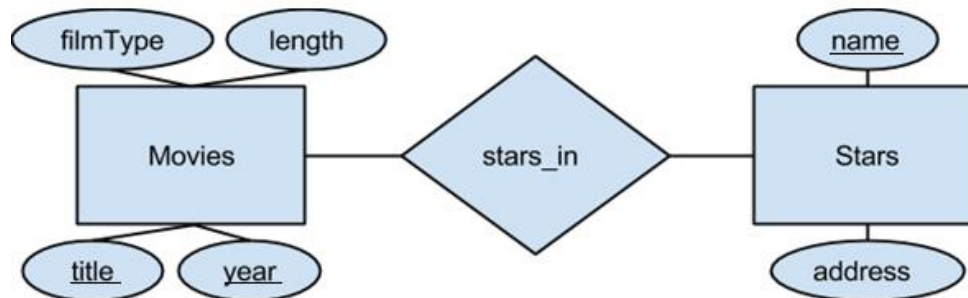
- Оптимизиран релационен модел

Movies(title, year, length, filmType, **studioName**)

Studio(name, address)

Комбиниране на релации

- ▶ Връзки много-много НЕ се оптимизират (не може!)
- ▶ Обединяването на такива релации води до излишества и неконсистентност на базата от данни

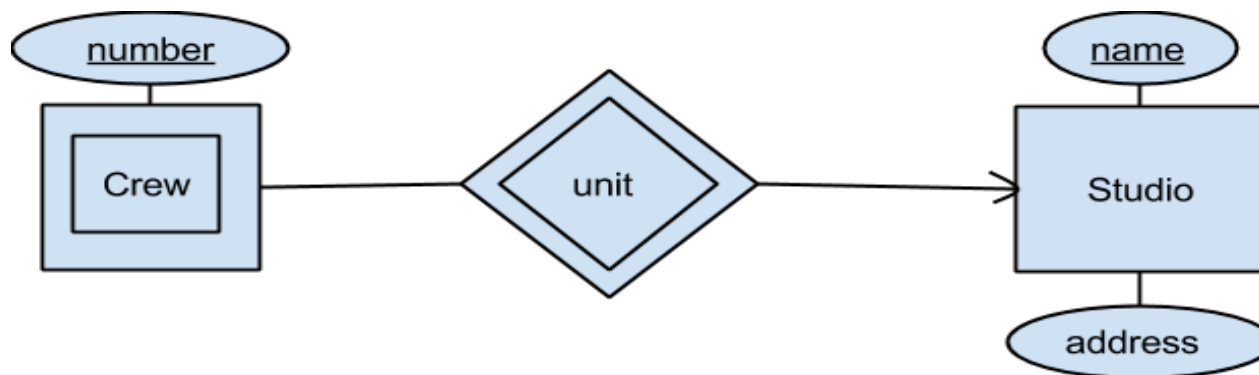


- ▶ Грешно !!!: **Movies**(title, year, *length*, *filmType*, *starName*)
- ▶ Ако в един филм играят няколко актьора, информацията за този филм ще бъде повторена. Имаме излишество.

Преобразуване на слаби множества от същности

- ▶ Ако имаме слабо множество от същности W в E/R диаграмата, алгоритъма за преобразуване е следният:
 - ▶ Релацията за слабото множество от същности W , не трябва да включва само атрибутите на W , но и ключовите атрибути на множествата от същности, които се достигат от W чрез поддържащи връзки (двоен ромб)
 - ▶ Релацията за всяка връзка (различна от поддържащите), в която участва слабото множество при преобразуване трябва да използва като ключ за W , всички ключови атрибути на W , включително и тези които се доставят от поддържащите множества
 - ▶ За поддържащите връзки, които свързват слабото множество W с другите множества от същности не се прави релация. Причината е, че поддържащите връзки винаги са от вида много-един и при преобразуване до релационен модел те ще съдържат ключа на W и ключа на поддържащото множество, което същност като атрибути са подмножество на атрибутите на релацията за W . Т.е двете релации могат да бъдат обединени.
 - ▶ Ако е необходимо, преименуваме имената на атрибутите за да избегнем дублиране

Преобразуване на слабо множество - пример

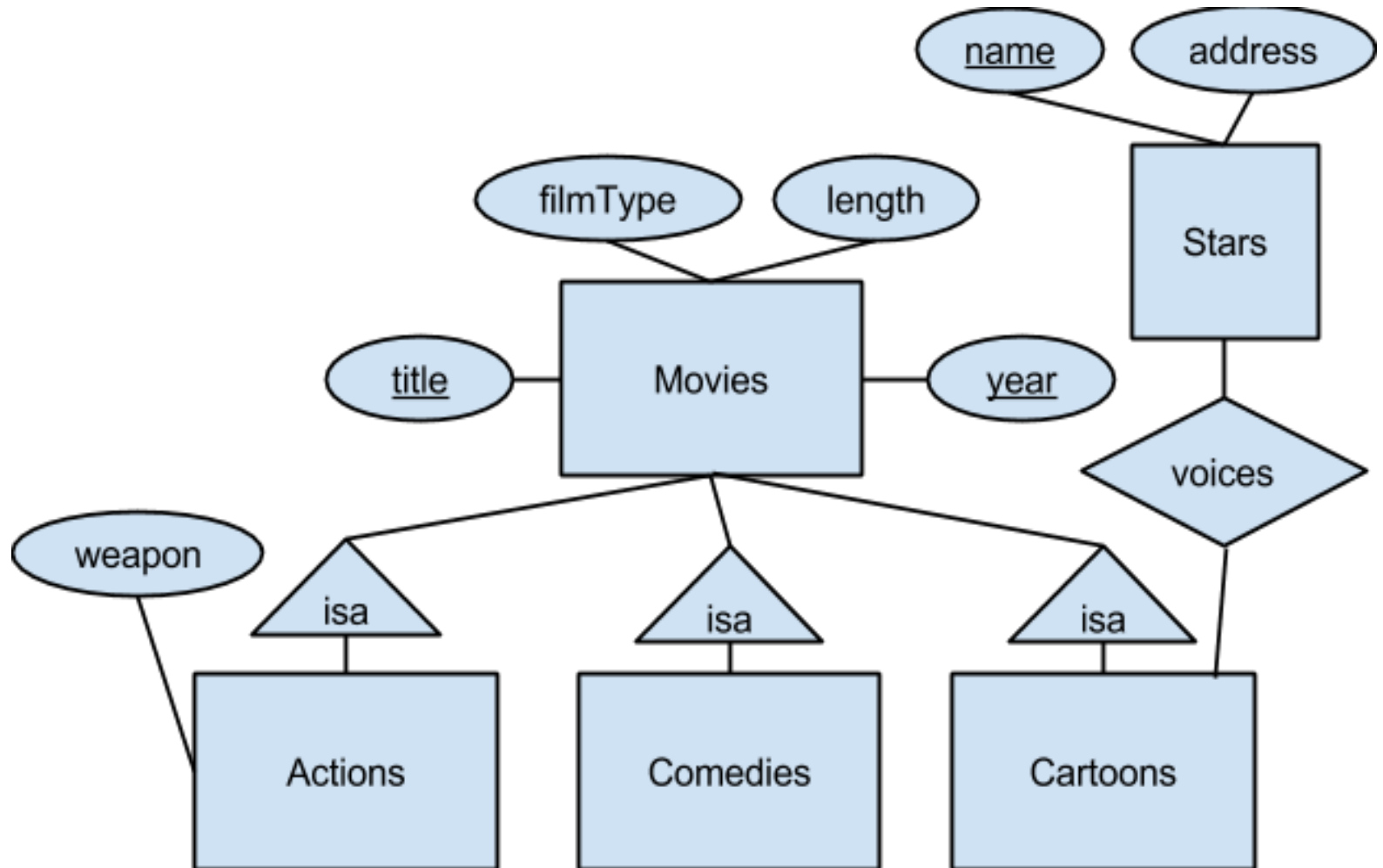


- ▶ Studio(name, address)
- ▶ Crews(number, studioName)
- ▶ Unit(number, studioName) X (не се прави релационна схема)

Преобразуване на йерархии от подкласове

- ▶ При множества от същности организирани в isa-йерархия имаме корен на йерархията. Това множество от същности (коренът) съдържа ключ, който уникално идентифицира всяка същност в йерархията.
- ▶ Когато имаме isa-йерархия (връзка) на множества от същности има три различни подхода за преобразуването на isa-връзката. Те са:
 - ▶ Релационен подход (E/R към REL)
 - ▶ Обектно-ориентиран подход (ООР)
 - ▶ NULL-подход (NULL)
- ▶ Релационният подход следва гледната точка на E/R модела разглеждана до сега
- ▶ Обектно-ориентираният подход, разглежда същностите като обекти от един и същи клас
- ▶ NULL – подхода използва null стойности.

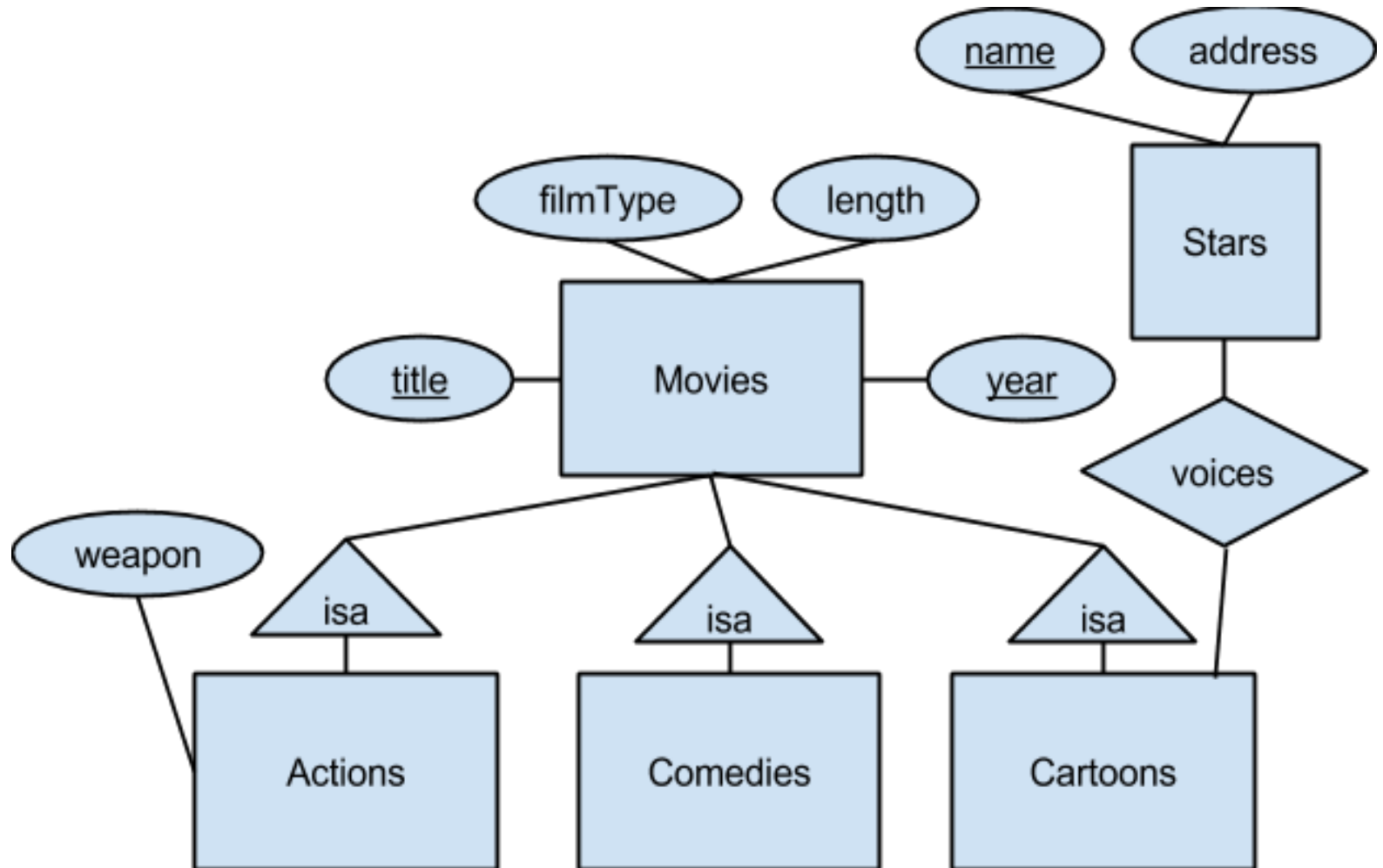
Задача



Е/Р към REL подход

- ▶ При този подход се създава релация за всяко множество от същности
- ▶ Ако множеството от същности не е корен на йерархията, то трябва да съдържа ключовите атрибути от корена и всички собствени атрибути.
- ▶ Ако множеството от същности участва във връзка, тогава при преобразуване на връзката ще използваме ключовите атрибути на множеството, за да идентифицират всяка същност от множеството във преобразуваната релация за връзката
- ▶ Връзката isa е специален тип връзка, която свързва компонентите на едно и също множество организирани в йерархия, а не различни множества. За isa връзка не се създава релация.

Задача



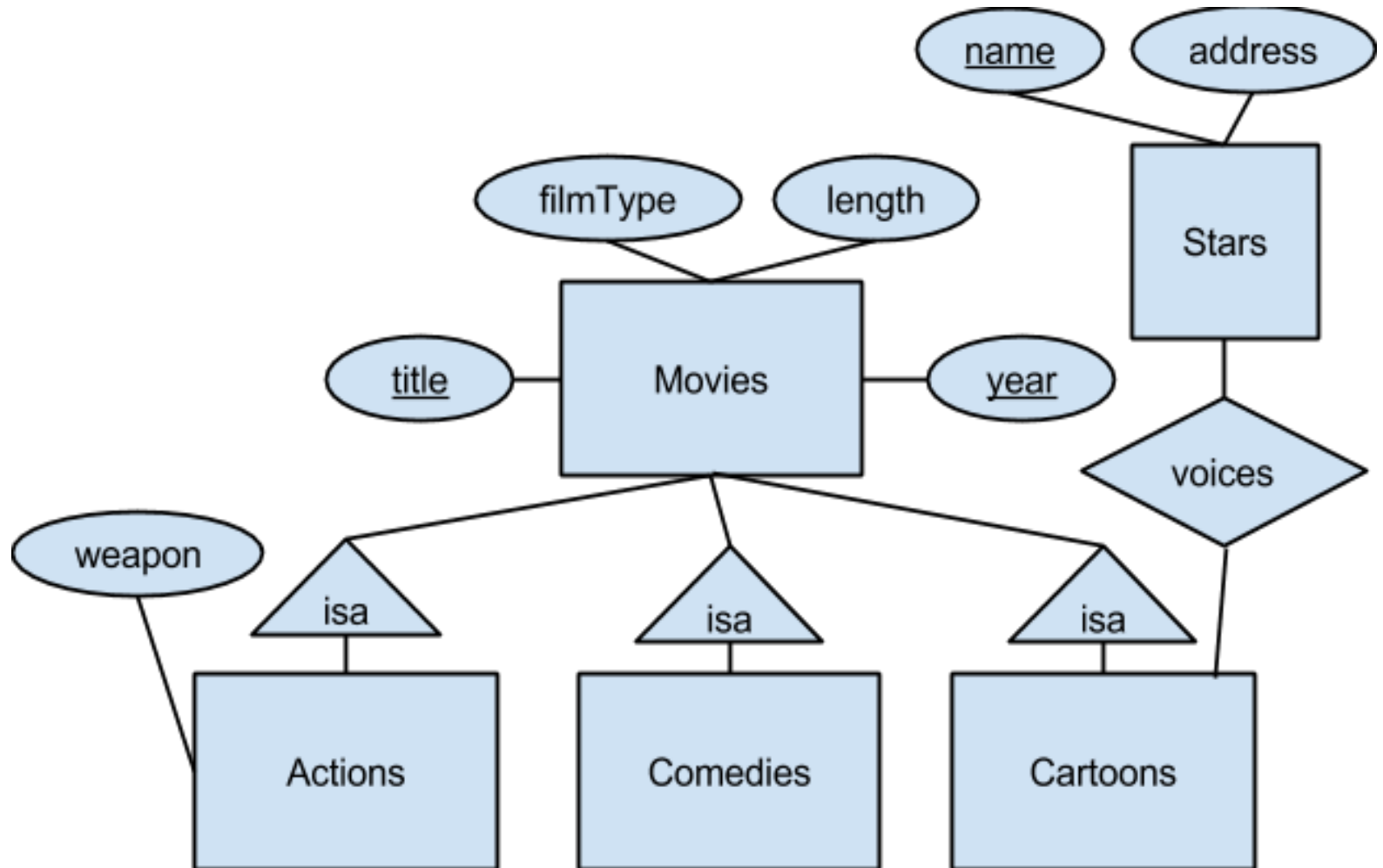
Е/Р към REL подход - Решение

- ▶ Movies (title, year, length, filmType)
 - ▶ Actions (title, year, weapon)
 - ▶ Comedies (title, year)
 - ▶ Cartoons (title, year)
 - ▶ Stars (name, address)
 - ▶ Voices(title, year, name)
-
- ▶ Ако един филм е едновременно и екшън и комедия и анимационен, то за него ще има четири записи във всяко едно от множествата същности
 - ▶ Въпреки че релацията Cartoons има схема, която е подмножество на релацията за връзката Voices, ние не може да я премахнем.
 - ▶ Ако Cartoons съдържа записи (кортежи) за неозвучен филм, те ще бъдат изгубени.

Обектно-ориентиран подход

- ▶ При този подход се създава релация за всяко едно възможно под-дърво от йерархията. Релацията ще съдържа всички атрибути на участващите в под-дървото множества.
- ▶ Подхода се нарича обектно-ориентиран, защото същностите се разглеждат като обекти, които принадлежат точно на един единствен клас (под-дърво от йерархията)
- ▶ Ако в йерархията от множества същности имаме $N+1$ множества от същности, включително и корена, то броя на резултатните релации при преобразуване на йерархията, чрез обектно-ориентирания подход ще бъде 2 на степен N

Задача



Обектно-ориентиран подход - Решение

- ▶ Броят на резултатните релации ще бъде 2 на степен 3 -та, т.е. 8 или при него имаме 8 възможни под-дървета

1. Само Movies
2. Само Movies и Cartoons
3. Само Movies и Comedies
4. Само Movies и Actions
5. Само Movies, Cartoons и Actions
6. Само Movies, Cartoons и Comedies
7. Само Movies, Comedies и Actions
8. Само Movies, Cartoons, Actions и Comedies

Обектно-ориентиран подход - Решение

Movies(title, year, length, filmType)

Movies_Cartoons(title, year, length, filmType)

Movies_Comedies(title, year, length, filmType)

Movies_Actions(title, year, length, filmType, weapon)

Movies_Cartoons_Actions(title, year, length, filmType, weapon)

Movies_Cartoons_Comedies(title, year, length, filmType)

Movies_Comedies_Actions(title, year, length, filmType, weapon)

Movies_Cartoons_Actions_Comedies(title, year, length, filmType, weapon)

Stars (name, address)

Voices(title, year, name)

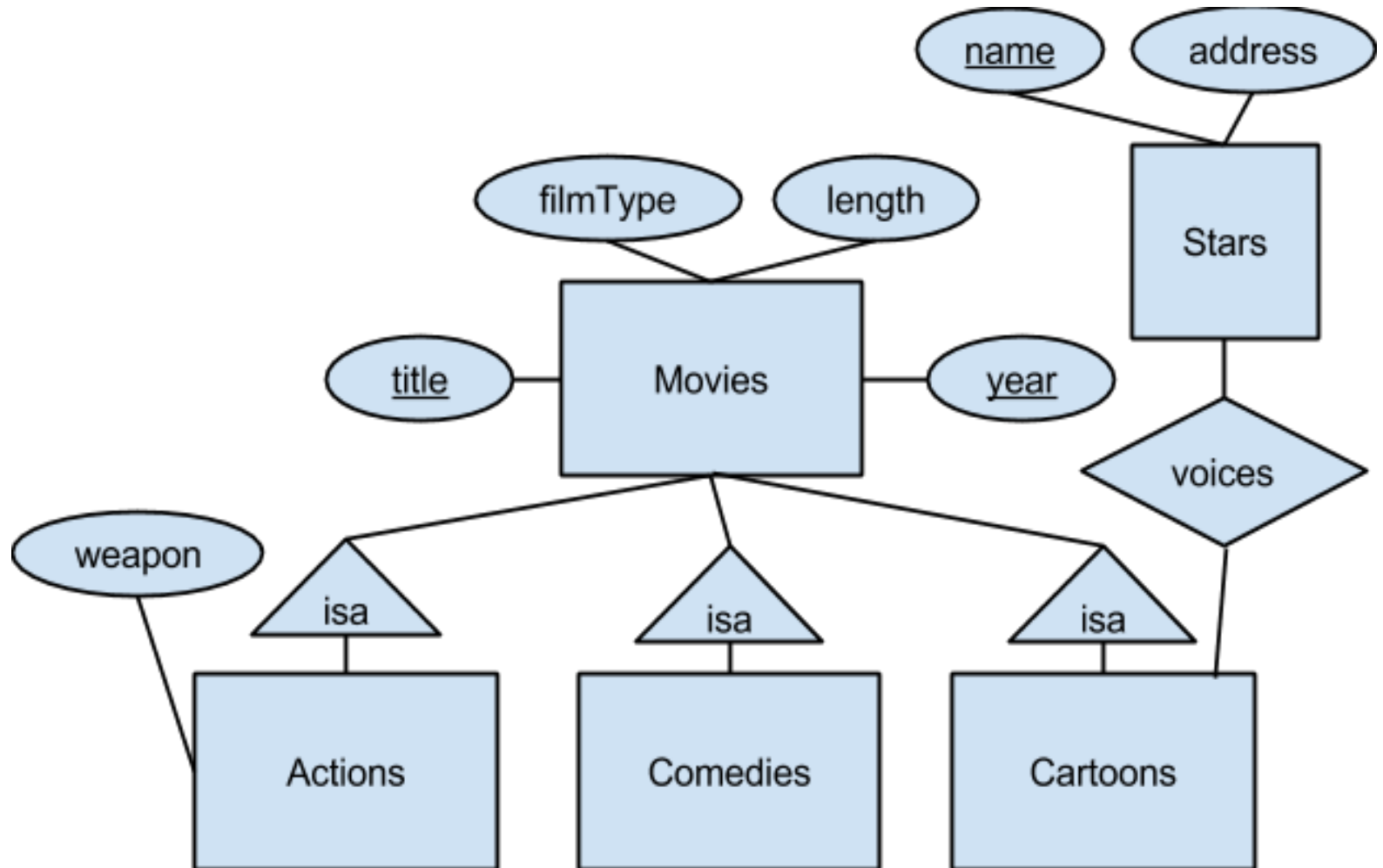
Обектно-ориентиран подход пояснения

- ▶ Въпреки че релациите по-долу имат еднакви атрибути, ние не можем да ги обединим в една релация
 - ▶ `Movies(title, year, length, filmType)`
 - ▶ `Movies_Cartoons(title, year, length, filmType)`
 - ▶ `Movies_Comedies(title, year, length, filmType)`
- ▶ Ако го направим, ще загубим информация за това кои филми са анимации и кои комедии.
- ▶ Както казахме при обектно-ориентирания подход всеки запис (същност) принадлежи точно на една релация. Така всички филми, които са анимации ще бъдат в таблицата `Movies_Cartoons(title, year, length, filmType)`
- ▶ Аналогично всички филми които са комедии ще бъдат в таблицата `Movies_Comedies(title, year, length, filmType)`
- ▶ При връзката `Voices` нещата стоят по същия начин както и при `E/R` подхода. Релацията за нея се състои от ключовите атрибути на релацията `Stars` и на релациите, които съдържат `Cartoons` – това са именно `name`, `title` и `year`

NULL подход

- ▶ При този подход, йерархията от под-класове се преобразува в една единствена релация.
- ▶ Тази релация съдържа всички атрибути на всички множества участващи в йерархията
- ▶ Ключът на релацията е ключът на множеството от същности, което е корен на йерархията
- ▶ Една същност от йерархията се представя като кортеж в релацията
- ▶ Ако даден кортеж няма подходяща стойност по някои от атрибутите там записваме **NULL**
- ▶ От тук идва и името на подхода

Задача



NULL подход - Решение

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

Stars (name, address)

Voices(title, year, name)

- ▶ Тези филми, които не са екшъни, ще имат NULL за атрибута weapon
- ▶ Ако един филм е едновременно и екшън и комедия, то за него ще има само един запис в таблицата Movies, за разлика от E/R подхода.