



Бази от данни

Модел на данни „Същност-връзки“

Моделът „Същност-връзки“ (E/R)

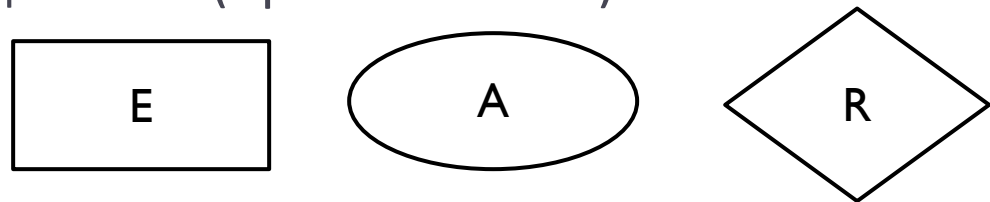
- ▶ Най-често използваният модел за абстрактно представяне на структурата на базата от данни е моделът „Същност-връзки“ (Entity-relationship, E/R)
- ▶ Силата на E/R модела е в неговата изразителна нотация. Чрез него може да бъде представена структурата на данните, но не и операции върху тях
- ▶ Моделът „Същност-връзки“ представя структурата на данните графично в диаграма – E/R диаграма
- ▶ Основните елементи на модела са:
 - ▶ Множество от същности
 - ▶ Връзки
 - ▶ Атрибути

Основни елементи на Е/Р модела

- ▶ **Същност** – наричаме абстрактен обект от някакъв вид. Например: студент, автомобил, филм и др.
- ▶ Съвкупност от такива същности се нарича множество от същности. Например: студенти, филми, автомобили и др.
- ▶ **Атрибути** – свойства (характеристики), които характеризират всяка същност от множеството от същности. Например: факултетен номер за студент, заглавие за филм, цвят за автомобил
- ▶ **Връзка** в модела е връзка (отношение) между две и повече множества от същности

Диаграма „Същност-връзки“

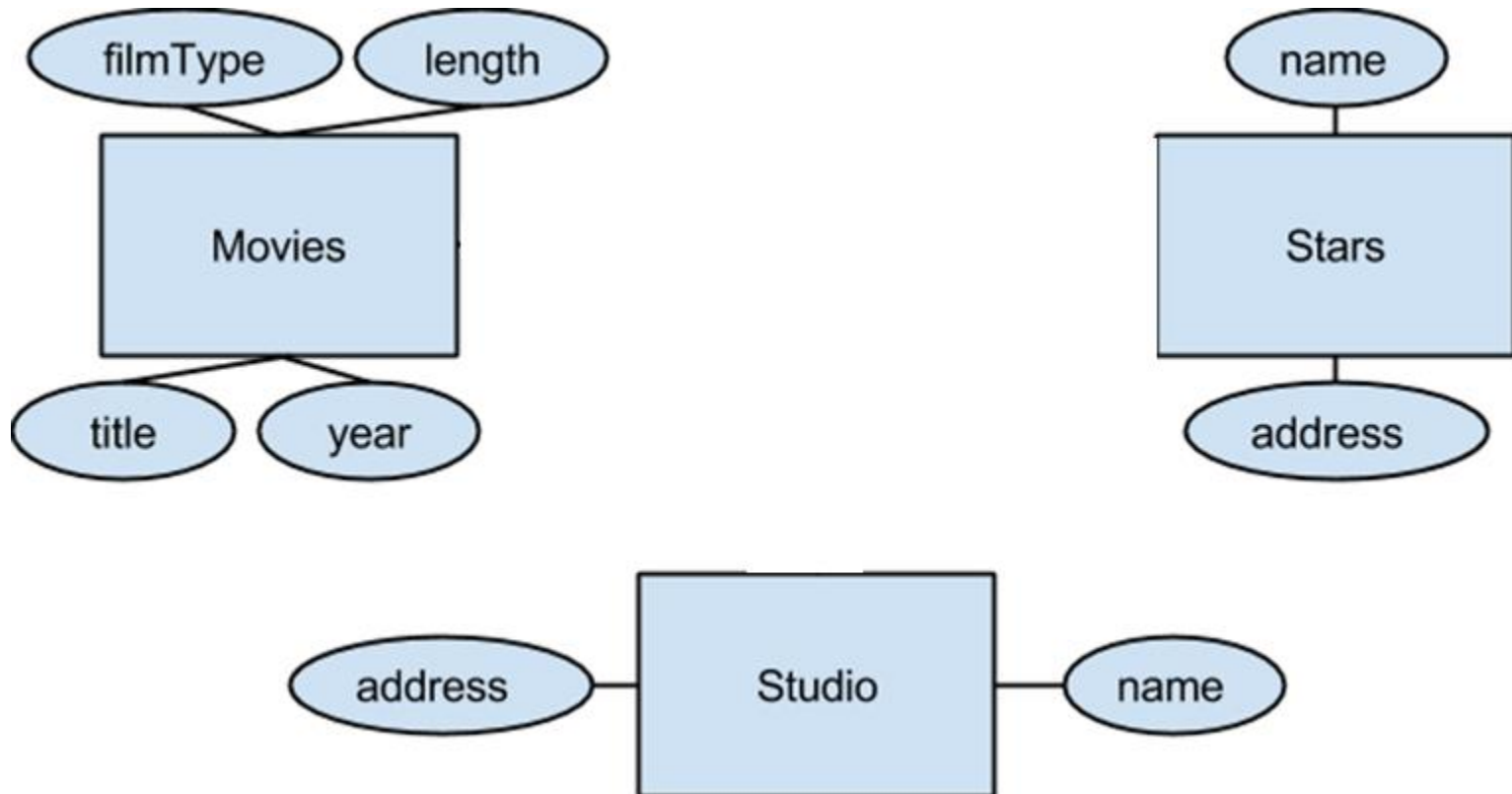
- ▶ Диаграмата „Същност-връзки“ (E/R диаграмата) е граф представящ множествата от същности с техните атрибути и връзките между тях.
- ▶ Нотацията която се използва е:
 - ▶ Множество от същности - (правоъгълник)
 - ▶ Атрибут - (овал)
 - ▶ Връзка – (ромб)
- ▶ Ребрата на графа свързват множеството от същности с неговите атрибути и връзките със съответстващите им множества от същности
- ▶ За примери в лекциите ще използваме базата от данни за филми (Movies)



База от данни Movies

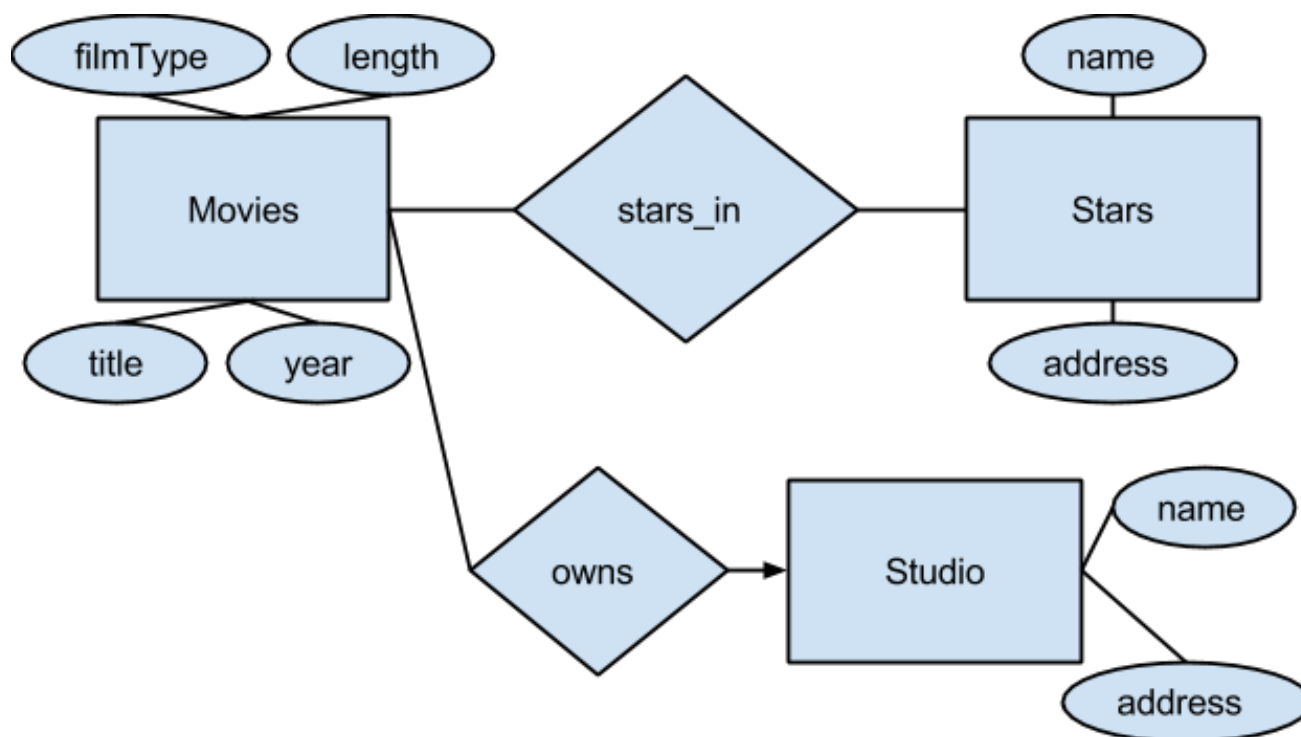
- ▶ **Продуценти:** номер на сертификат на продуцент, име, адрес, печалба от всички филми
- ▶ **Филми:** заглавие, година, тип, продължителност на филма в минути
- ▶ **Студио:** име, адрес, номер на сертификат на студио
- ▶ **Актьори:** име, адрес
- ▶ Един актьор може да участва в много филми и в един филм може да участват много актьори
- ▶ Един продуцент може да продуцира много филми, но всеки филм е продуциран точно от един продуцент
- ▶ Едно студио може да има много филми, но всеки филм е направен точно от едно студио

Множества от същности



Пример за Е/Р диаграма

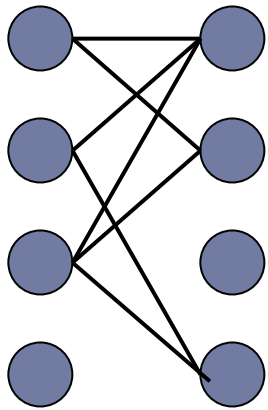
- ▶ **Филми:** заглавие, година, тип, дължина
- ▶ **Студио:** име, адрес
- ▶ **Актьори:** име, адрес



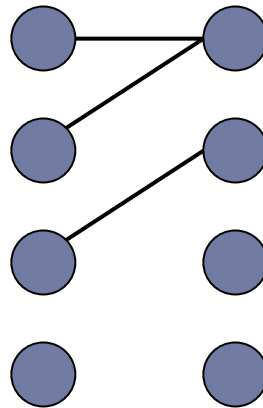
Видове връзки в E/R модела

- ▶ Връзките в E/R модела могат да бъдат бинарни и N-арни
- ▶ Бинарните връзки свързват две множества от същности, а N-арните свързват N множества от същности
- ▶ Често N-арните връзки се наричат и небинарни
- ▶ В зависимост от съответствието на същности от едно множество със същности от друго множество, връзките са много-един, много-много и едно – едно
- ▶ Нека R е връзка, която свързва множествата от същности E и F
 - ▶ Казваме, че връзката R е много към един от E към F, ако на една същност от множеството F и съответства 0, 1 или повече същности от E
 - ▶ Аналогично, казваме че връзката R е много към един от F към E, ако на една същност от множеството E и съответства 0, 1 или повече същности от F
 - ▶ Ако връзката R е едновременно много към един от E към F и много към един от F към E то тя е едно към едно, т.е. на една същност от множеството E съответства 0 или 1 същност от множеството F
 - ▶ Казваме, че връзката R е много към много от E към F, ако на много същности от множеството F и съответстват много същности от E

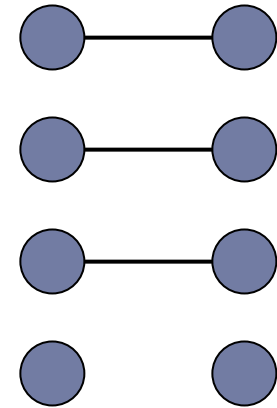
Видове бинарни връзки



Много-много



Много-един



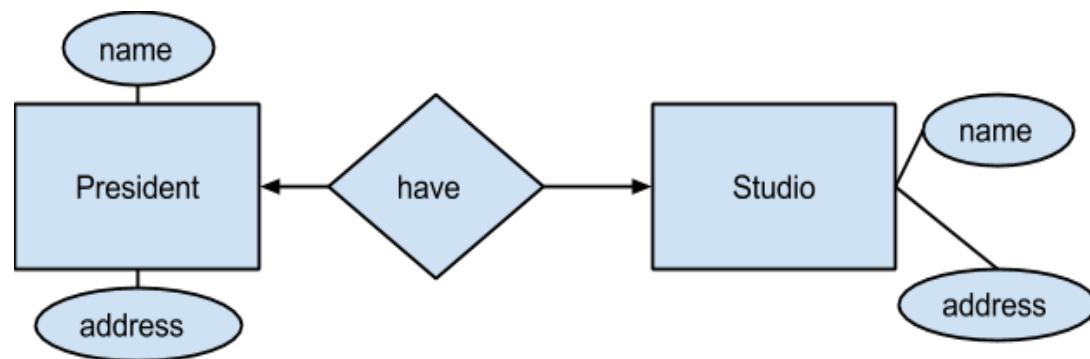
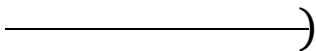
Едно-едно

Връзки едно-едно

- ▶ Един или 0

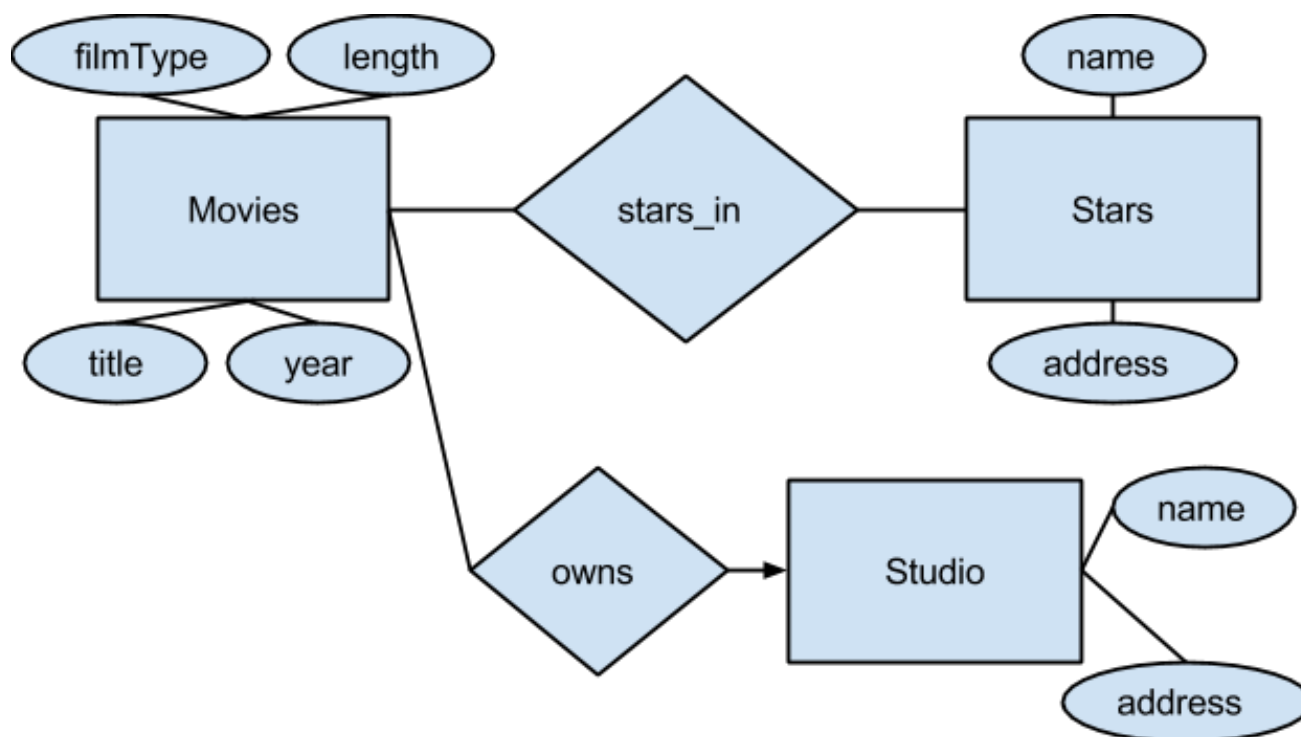


- ▶ Точно един



Пример за Е/Р диаграма

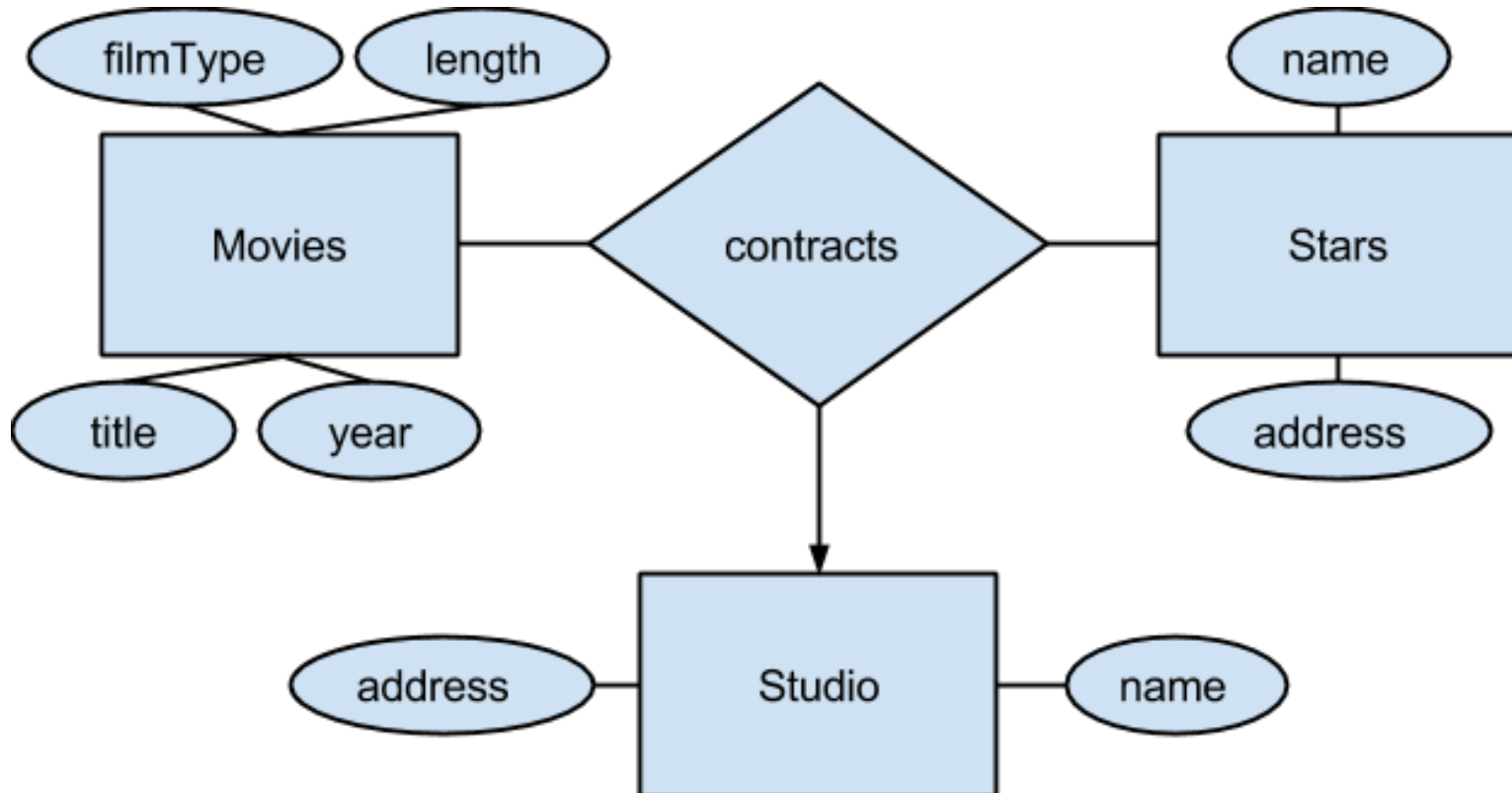
- ▶ **Филми:** заглавие, година, тип, дължина
- ▶ **Студио:** име, адрес
- ▶ **Актьори:** име, адрес



Небинарни връзки

- ▶ E/R моделът предоставя възможност и за дефиниране на връзки свързващи повече от две множества от същности
- ▶ На практика тринарни или връзки от по висока степен се срещат рядко.
- ▶ Небинарните връзки се представят чрез линии от ромба представящ връзката, към всяко едно от свързаните множества от същности
- ▶ Например връзката – Contracts свързва трите множества Stars, Movies и Studio

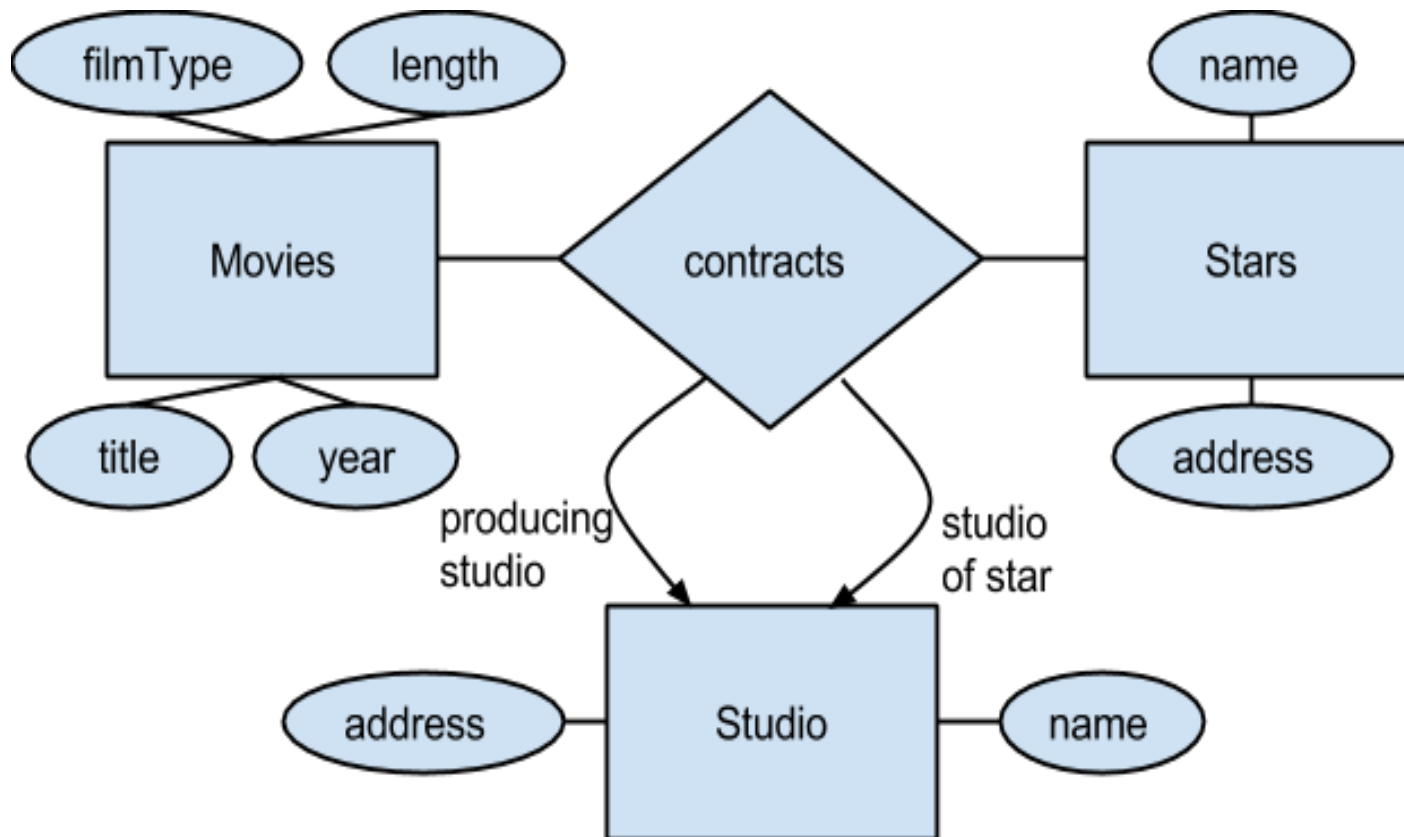
Пример за небинарна връзка



Роли при връзките

- ▶ В E/R модела връзките могат да имат роли
- ▶ Роля на връзката се прилага в случаите, когато едно множество от същности се използва два или повече пъти в една връзка, но в различен смисъл
- ▶ Тогава за всяка роля се поставя отделна дъга от връзката към множеството от същности
- ▶ Всяка дъга представя различна роля, която множеството от същности „играе“ в тази връзка
- ▶ За да се различават отделните роли, дъгите се именуват

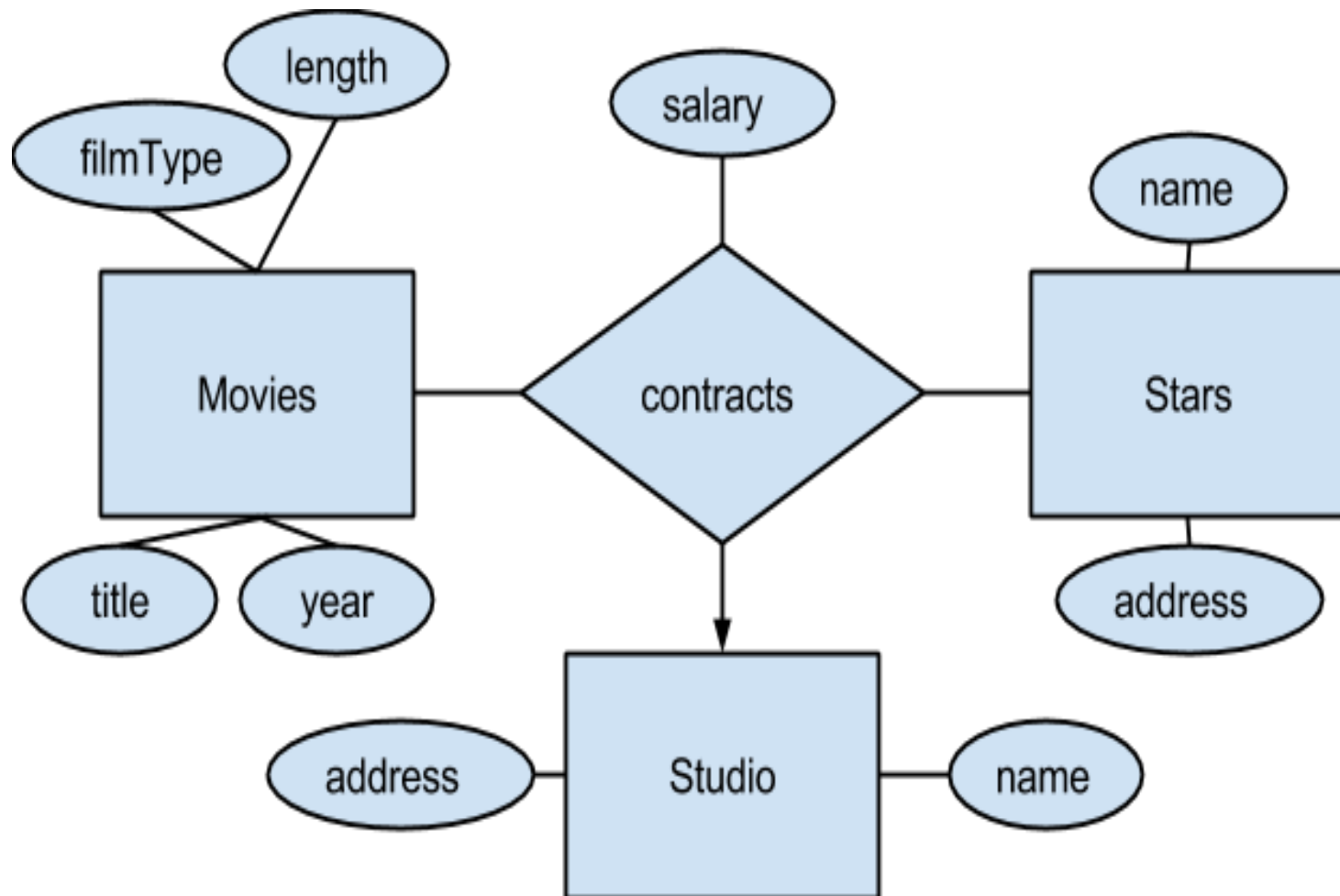
Пример за роли при връзките



Атрибути на връзките

- ▶ Атрибутът може да бъде не само характеристика на множество от същности, но и на връзка
- ▶ На една връзка се поставя атрибут в случаите, когато атрибутът е общ за всички множества от същности, които връзката свързва
- ▶ Например ако искаме да видим каква е сумата от продажбата на един филм направен от конкретно студио с участието на конкретна звезда и добавим такъв атрибут – salary, то salary не може да бъде атрибут на Studio, защото едно студио произвежда много филми. Атрибутът не може да бъде характеристика на Stars, защото един актьор играе в много филми. Така атрибутът може да бъде добавен само към връзката Contract

Пример за атрибути на връзките



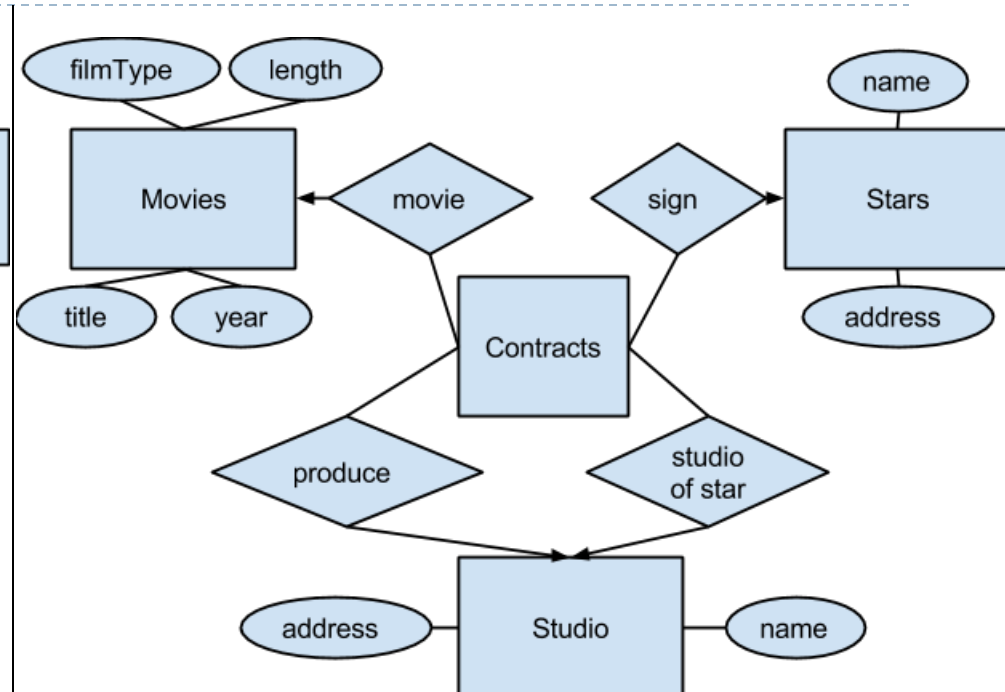
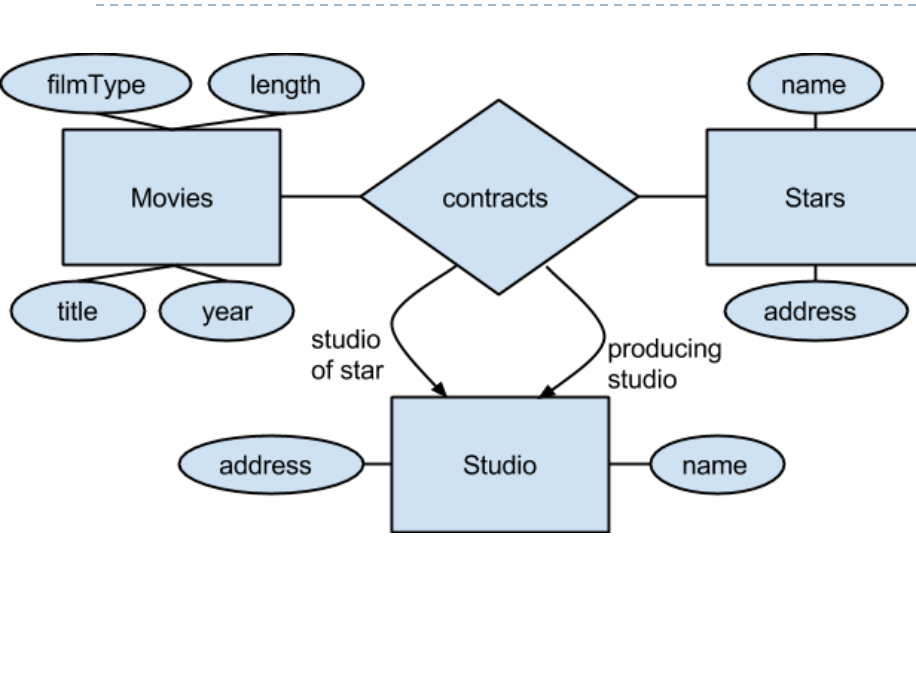
Преобразуване на небинарна връзка в бинарна

- ▶ Някои модели на данни, като Обектно-ориентирания модел на данни (ODL) поддържат само бинарни връзки
- ▶ В E/R модела това не е така
- ▶ Всяка небинарна връзка свързваща повече от две множества може да бъде сведена до бинарни връзки от вида много-един
- ▶ Алгоритмът за преобразуване на небинарна връзка към бинарна е следният:

Алгоритмът за преобразуване на небинарна връзка към бинарна

- ▶ Създаваме множество от същности на мястото на небинарната връзка с име, името на връзката
- ▶ Ако връзката има атрибути, те стават атрибути на новосъздаденото множество от същности
- ▶ За всяко множество от същности, което връзката свързва създаваме бинарна много-един връзка от новото множество от същности към свързаните множества от същности
- ▶ Ако дадено множество от същности участва с няколко роли към връзката, за всяка роля се прави отделна бинарна връзка много-един

Пример за преобразуване на небинарна връзка в бинарна



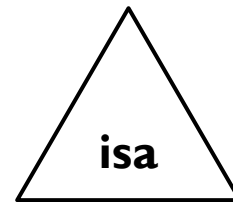
- ▶ Въвеждаме множество от същности на мястото на небинарната връзка с име, името на връзката
- ▶ Добавяме нови връзки много-един, които свързват новото множество с другите
- ▶ В случай на повече от една роля, за всяка роля се създава връзка много-един

Подкласове в модела „Същност-връзки“

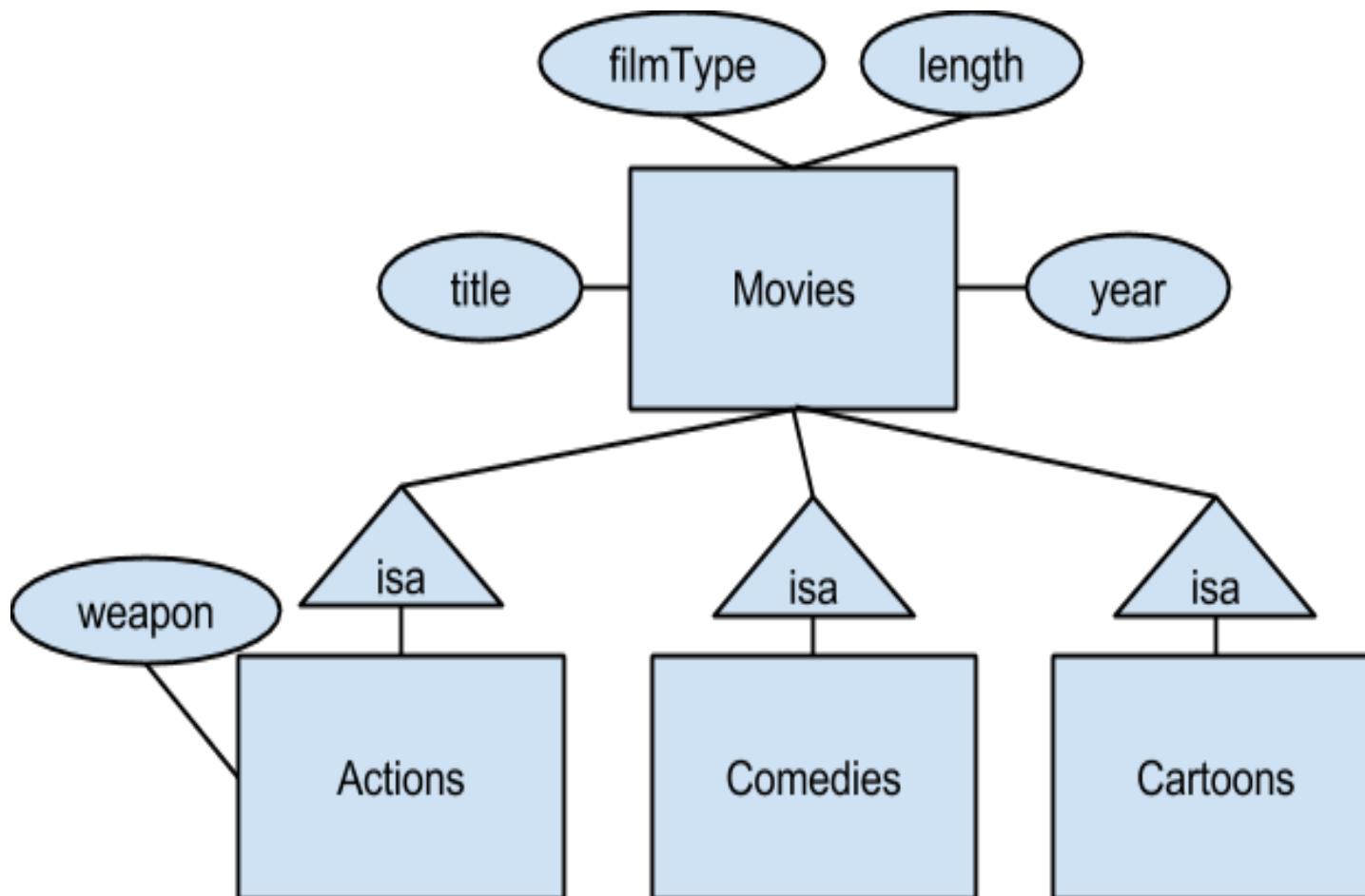
- ▶ Друг вид връзка освен горе-изброените е isa връзката
- ▶ Чрез нея могат да се изразяват йерархии от класове на множества от същности
- ▶ Често едно множество от същности съдържа същности със специфични характеристики, които са в сила само за конкретните същности, но не и за останалите същности в множеството
- ▶ В такива случай се дефинират подкласове от тези същности, които се свързват с базовото множество от същности чрез isa връзка
- ▶ Подкласовете на дадено множество от същности се определят от същности принадлежащи на даден клас, които освен основните си характеристики, притежават и допълнителни такива
- ▶ Множеството от същности, което е на върха на йерархията (супер-класа) съдържа общите атрибути за всички същности от йерархията

Подкласове в модела „Същност-връзки“

- ▶ Нотацията за isa връзка е триъгълник, като върхът на триъгълника сочи супер-класа
- ▶ Всяка isa-връзка е едно-едно
- ▶ Isa-връзките заедно със супер-класът и подкласовете се представят структурно като дърво, в което корен е най-общото множество от същности (супер-класа)
- ▶ Нека имаме йерархия от множества от същности свързани с isa връзка. Ако R е корен, а E и F са подкласове на R, то същностите от E и F имат същите атрибути като R и ново-добавени



Пример за подкласове в Е/Р модела

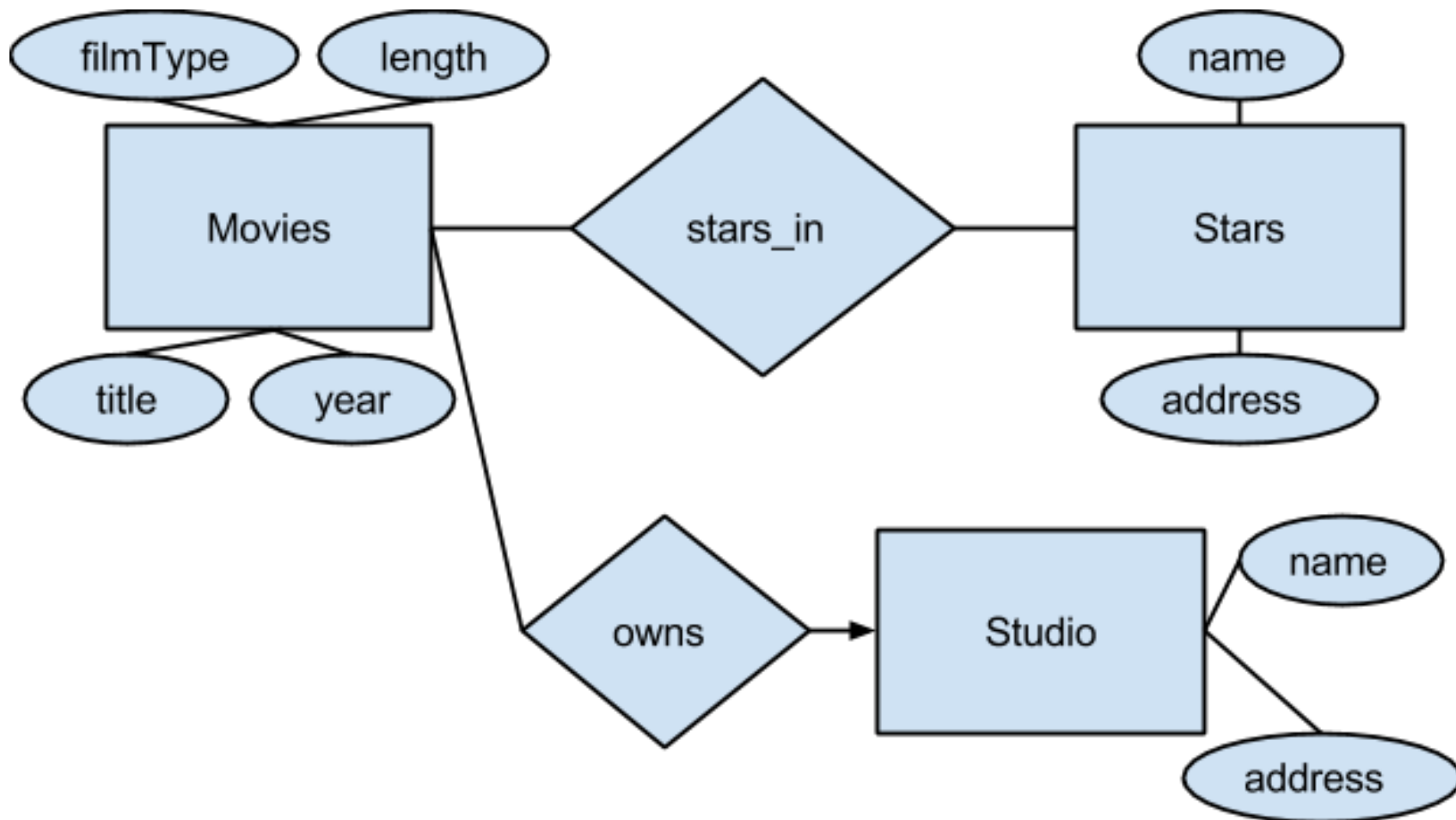


Подкласове E/R vs ООП

- ▶ Има разлика между класовете в E/R модела и в ООП
- ▶ При ООП даден елемент принадлежи точно на един клас
- ▶ В E/R модела дадена същност, може да принадлежи на няколко класа, например на супер-класа и на подкласа
- ▶ Това е в случаите при които дадената същност има повече от един атрибут по които съвпада с два или повече подкласа
- ▶ Например един филм, ако е едновременно и анимационен и комедия ще има записи за него и в трите множества от същности – Movies, Cartoons и Comedies

Е/Р диаграма

Е/Р диаграмата е начин за описание на схемата на БД

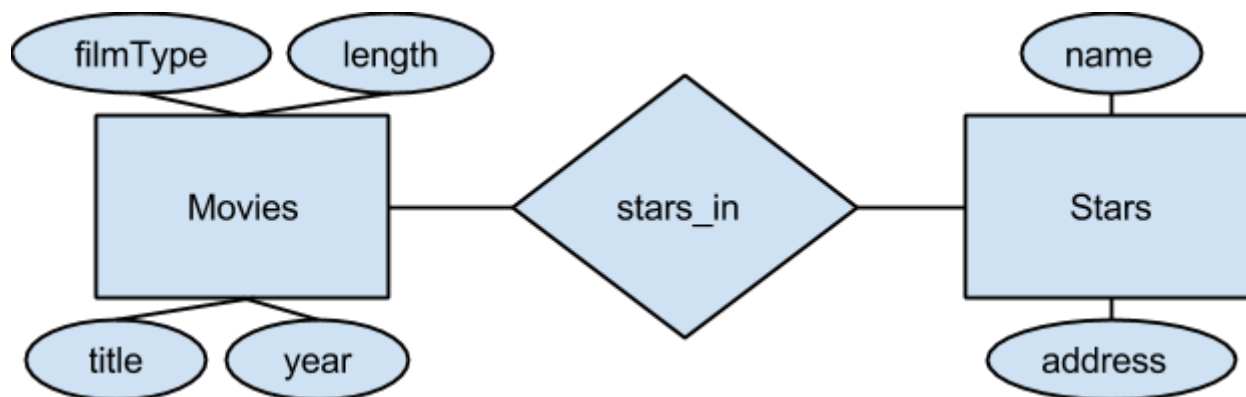


Екземпляр на E/R диаграма

- ▶ Екземпляр на E/R диаграма наричаме, конкретна инстанция на базата от данни
- ▶ Под конкретна инстанция на БД се има предвид конкретни същности от множествата от същности в E/R диаграмата с фиксирани стойности за атрибутите и връзките, които свързват съответните множества
- ▶ Например, ако връзката R свързва n множества от същности – $E_1, E_2, E_3, \dots, E_n$, инстанцията на R е множество от наредени n -торки от вида $(e_1, e_2, e_3, \dots, e_n)$, където e_i е конкретна същност от съответните множества $E_1, E_2, E_3 \dots E_n$

Проектиране на БД - принципи

- ▶ Изграждането на модел на базата от данни се базира на пет основни принципа:
 - ▶ Съответствие
 - ▶ Избягване на излишества
 - ▶ Простота
 - ▶ Избор на точните връзки
 - ▶ Избор на точните елементи

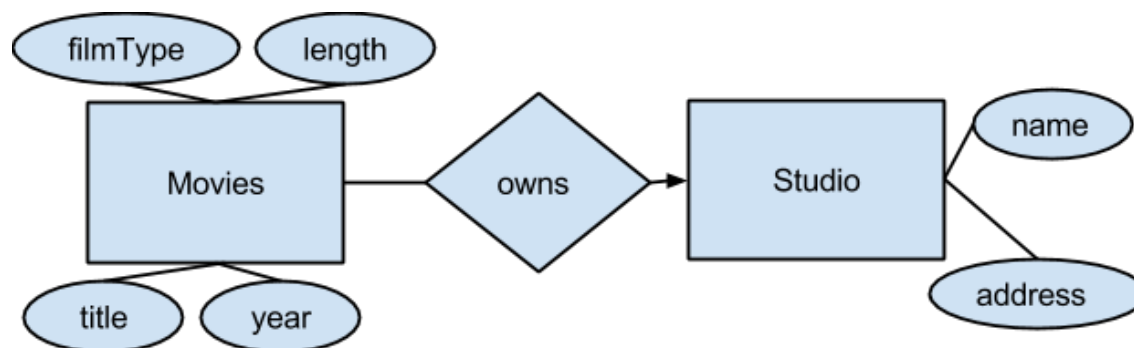


Съответствие

- ▶ Създаването на обектите или множествата трябва да отговаря на действителността
- ▶ Връзките, които се изграждат между тях също трябва да са смислени и да отразяват реалността
- ▶ Например връзката `stars_in` ни показва, че свързва `Stars` и `Movies` и тя задължително трябва да е много-много, защото E/R схемата е модел на реалния свят, а в него много звезди, могат да участват в много филми
- ▶ Понякога, реалният свят не ни дава достатъчна представа за това какъв трябва да е вида на връзката, тогава е необходима допълнителна информация за да се специфицира вида на връзката

Избягване на излишества

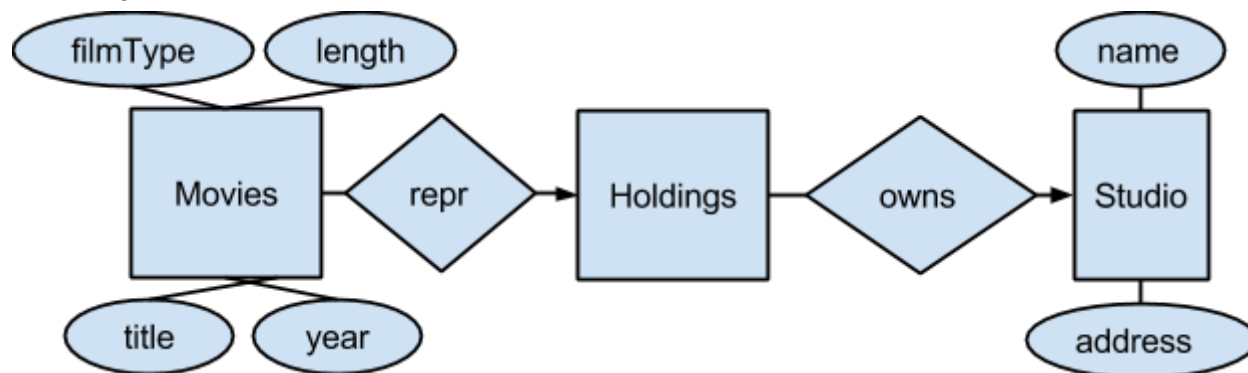
- ▶ Да се внимава да няма повтаряне на информация
- ▶ Например множествата от същности



- ▶ Ако добавим атрибут `studioName` към `Movies` ще получим излишество
 - ▶ Например ако за един филм сменим студиото, връзката `owns` ще свързва конкретният филм с новото студио
 - ▶ Ако забравим, обаче да променим `studioName` в `Movies`, ще получим невярна информация
 - ▶ Представянето на едно и също нещо на две места, ще заема и повече памет при съхраняването на данни

Простота

- ▶ Да се избягва употребата на повече от необходимите елементи
- ▶ Например, ако добавим ново множество от същности



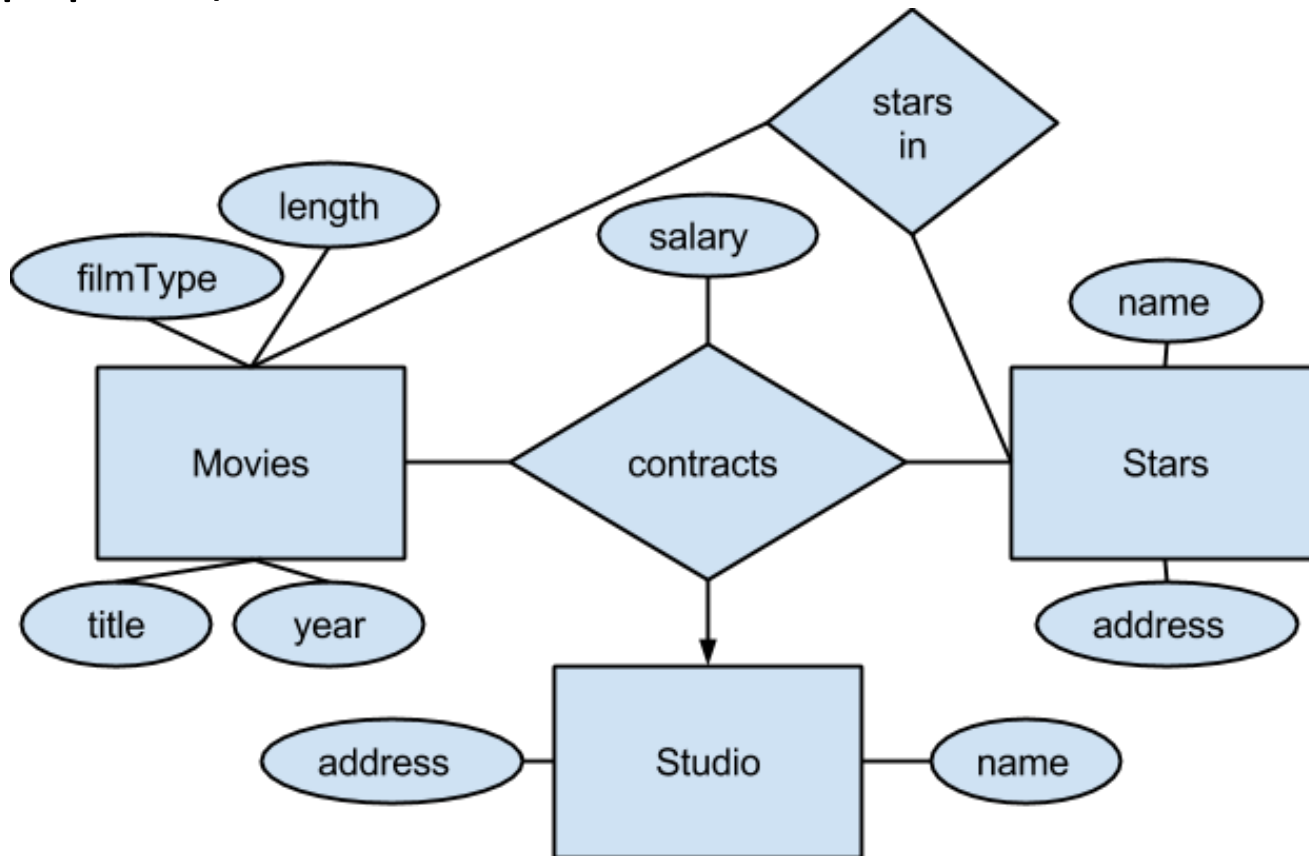
- ▶ В случая, диаграмата е коректна и отразява вярно действителността. Но същността Holdings е излишна и може без нея. Тя усложнява диаграмата, губи се памет и може да доведе до допускане на грешки
- ▶ Трябва да се целим към изчистен и опростен дизайн!

Избор на точните връзки

- ▶ Множествата от същности са свързани посредством връзки
- ▶ Но, добавянето на всевъзможни връзки към дизайна на схемата не винаги е добра идея
 - ▶ Това може да доведе до излишества
 - ▶ Резултатната БД ще изисква много повече място за да съхрани излишните елементи
 - ▶ Модифицирането на резултатната БД може да бъде твърдо сложно и да изисква твърде много промени по съхранените връзки
- ▶ Проблемите по естество са подобни като тези при излишество, но причината за проблемите са различни

Избор на точните връзки - Пример

- ▶ Например връзки, които представят една и съща информация



Избор на точните връзки - Пример

- ▶ Дали са необходими и двете връзки `contracts` и `stars_in` ?
- ▶ Истината е че зависи от нашите предположения
 - ▶ Ако актьор може да участва във филм само ако има договор със съответното студио за участие в съответния филм, тогава нямаме нужда от връзката `stars_in` . В този случай връзката `stars_in` може да се изведе от `contracts`
 - ▶ Ако обаче, един актьор може да работи по един филм без договор, тогава връзката `stars_in` е необходима

Избор на точните елементи

- ▶ Понякога имаме различни възможности при моделиране на дизайна на базата от данни и представянето на реалния свят
- ▶ Често трябва да направим избор дали да използваме връзка, атрибут или слабо множество за да отразим действителността.
- ▶ Например едно множество от същности може да бъде моделирано и като атрибут на друго множество от същности
- ▶ Изборът дали едно множество от същности да остане множество или да се включи като атрибут на множество или връзка, също трябва да се вземе предвид

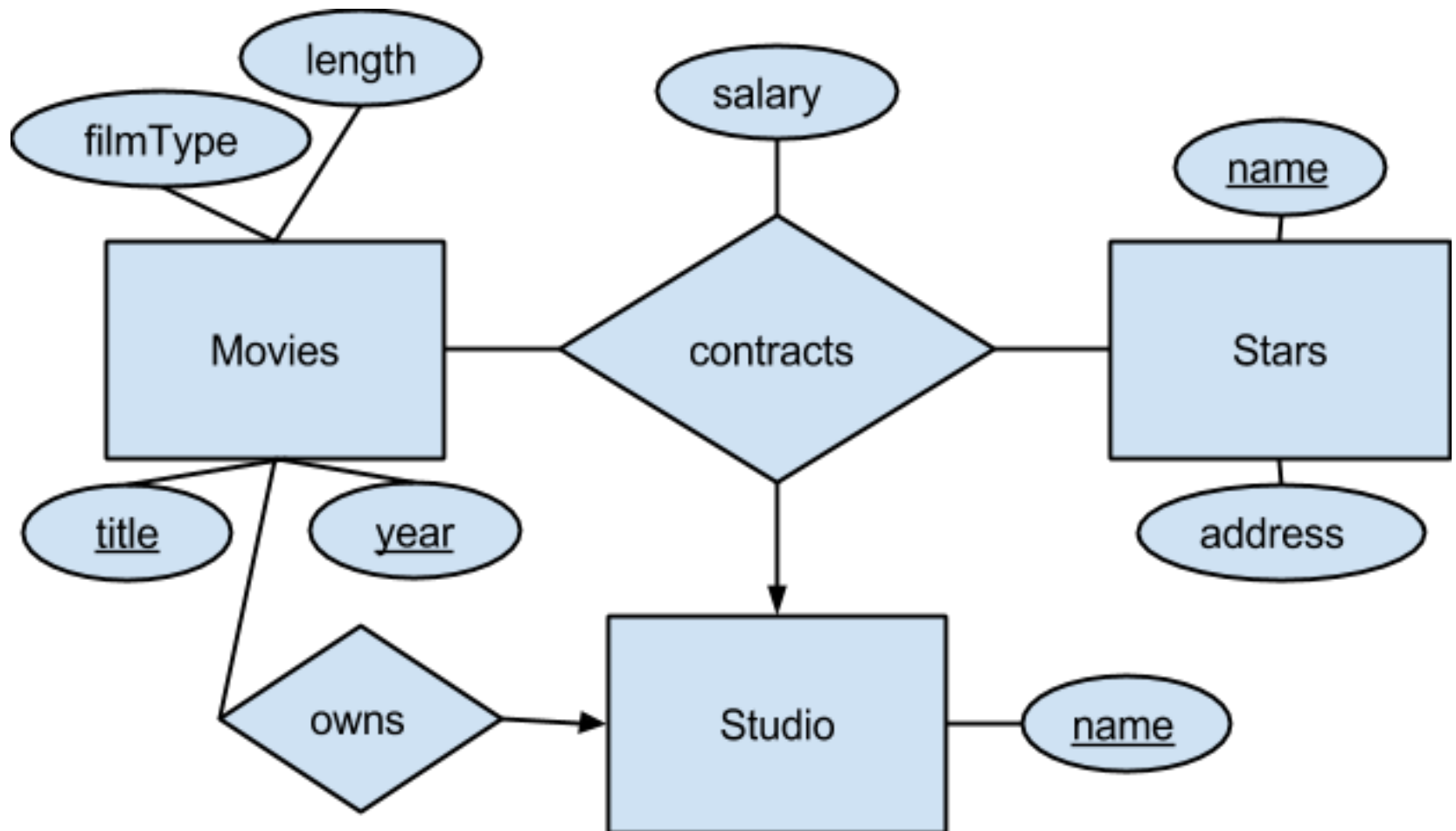
Избор на точните елементи - изисквания

- ▶ Не всяко множество от същности може да стане атрибут на множество или връзка. Ако **Е** е множество от същности изискванията са следните:
 - ▶ **Е** трябва да се сочи само от връзки много-един. Това ни гарантира, че няма да има излишно повтаряне на информация, ако **Е** стане атрибут
 - ▶ **Е** не трябва да участва повече от веднъж за дадена връзка (не трябва да участва в роли)
 - ▶ Всички атрибутите на **Е** трябва уникално да идентифицират всяка същност от множеството. Ако **Е** има един атрибут, то той трябва да е ключ. Ако **Е** има повече от един атрибут, то тогава нито един от тях не трябва да зависи от останалите

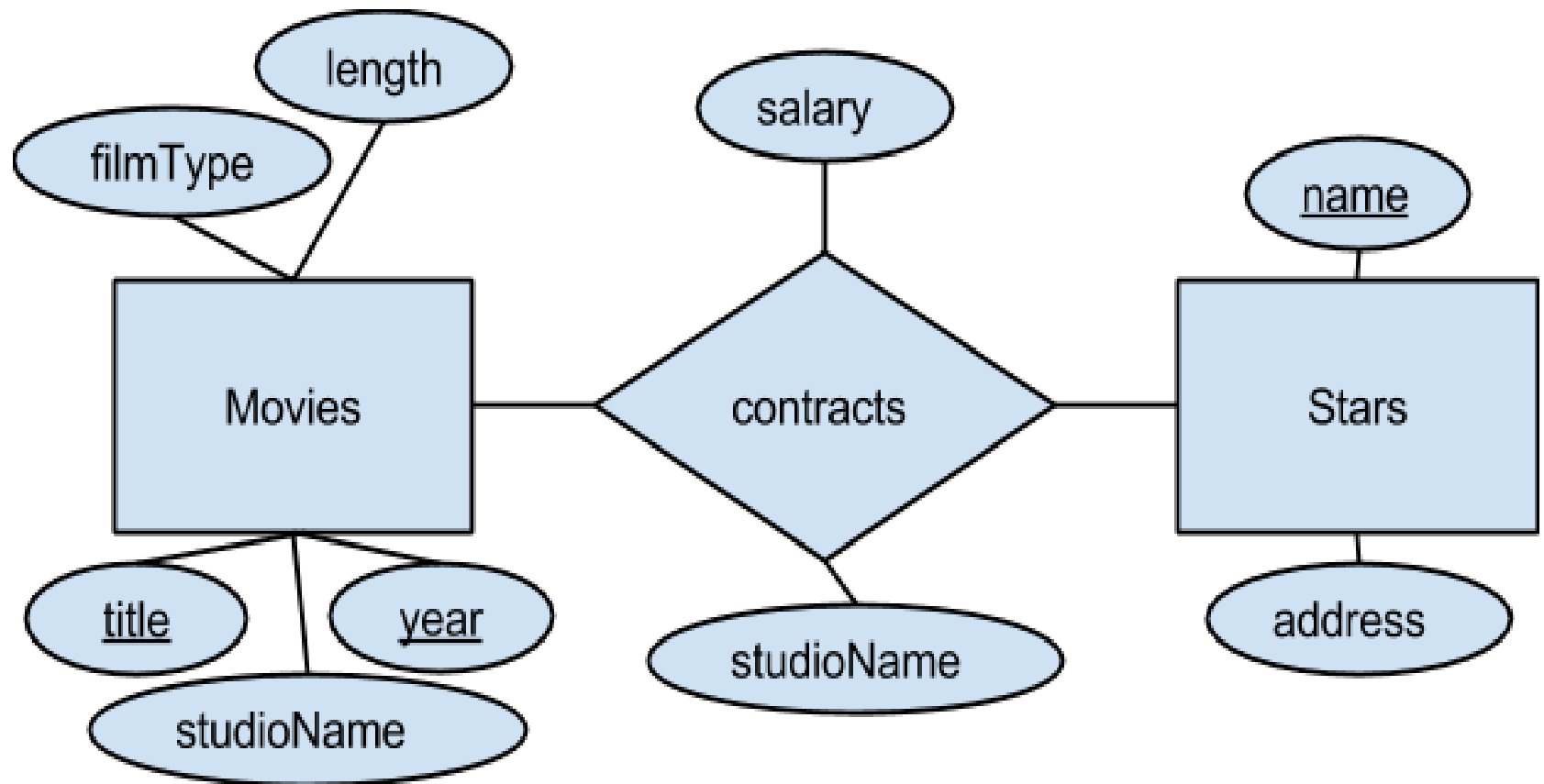
Избор на точните елементи - преобразуване

- ▶ Ако тези условия са изпълнение, тогава може да заменим множеството от същности E , както следва:
 - ▶ Ако E участва във връзка R много – един от множеството от същности F , тогава премахваме връзката R и атрибутите на E стават атрибути на F . Ако имаме еднакви имена на атрибути от E и F – преименуваме
 - ▶ Ако имаме небинарна връзка R със стрелка към E , правим атрибутите на E атрибути на връзката R и изтриваме дъгата от R към E

Избор на точните елементи - пример



Избор на точните елементи - решение



Моделиране на ограничения

- ▶ Проблеми като излишеството и неконсистентност могат да бъдат избегнати чрез ограничения
- ▶ Ограниченията са част от схемата на базата от данни
- ▶ Те могат да бъдат:
 - ▶ Ключове
 - ▶ Ограничения по единствена стойност
 - ▶ Ограничения по референтна цялостност
 - ▶ Общи ограничения
 - ▶ Ограничения по домейн
- ▶ В E/R модела има нотация само за изразяването на ключ и ограничение по референтна цялостност. Всички останали ограничения не могат да бъдат отразени върху диаграмата на E/R модела

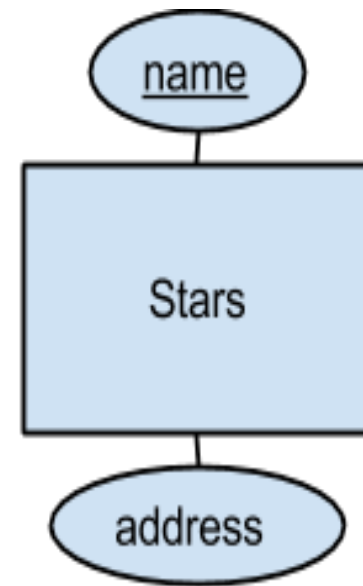
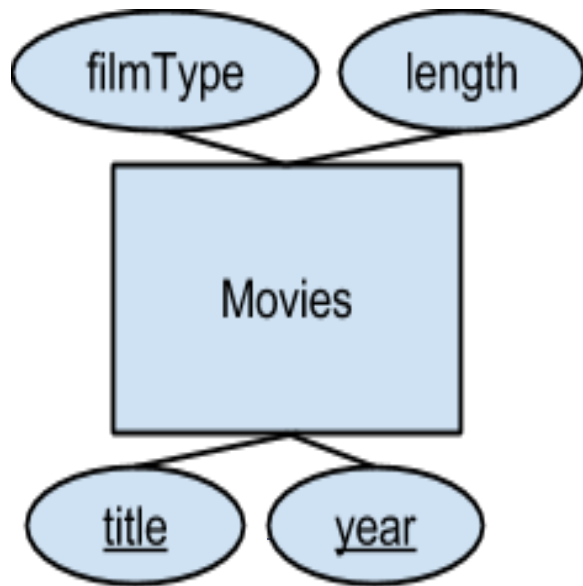
Ключ

- ▶ Ключ на множество от същности наричаме списък от атрибути, които уникално определят всяка същност от множеството
- ▶ Дефиниция на ключ в E/R модела – Ключ за едно множество от същности E е списък от един или повече атрибути K , такива че ако e_1 и e_2 са две различни същности в E , то e_1 и e_2 не съвпадат (не могат да имат еднакви стойности) по всички атрибути от K . Ако K съдържа повече от един атрибут, то тогава е възможно e_1 и e_2 да съвпадат по някои от атрибутите, но не и по всичките
- ▶ Ключът е важен защото може да идентифицира дадена същност
- ▶ В релационните бази от данни ключа ускорява достъпа до данните

Важно за ключовете

- ▶ Всяко множество от същности трябва да има ключ
- ▶ Един ключ може да съдържа повече от един атрибут
- ▶ Едно множество от същности може да има повече от един възможен ключ (кандидат ключ), но само един от атрибутите се избира за първичен ключ
- ▶ Когато едно множество от същности поражда *isa* йерархия, изисква се корена на йерархията да съдържа всички атрибути, които еднозначно да определят същностите в йерархията, т.е. ключът

Пример за ключ



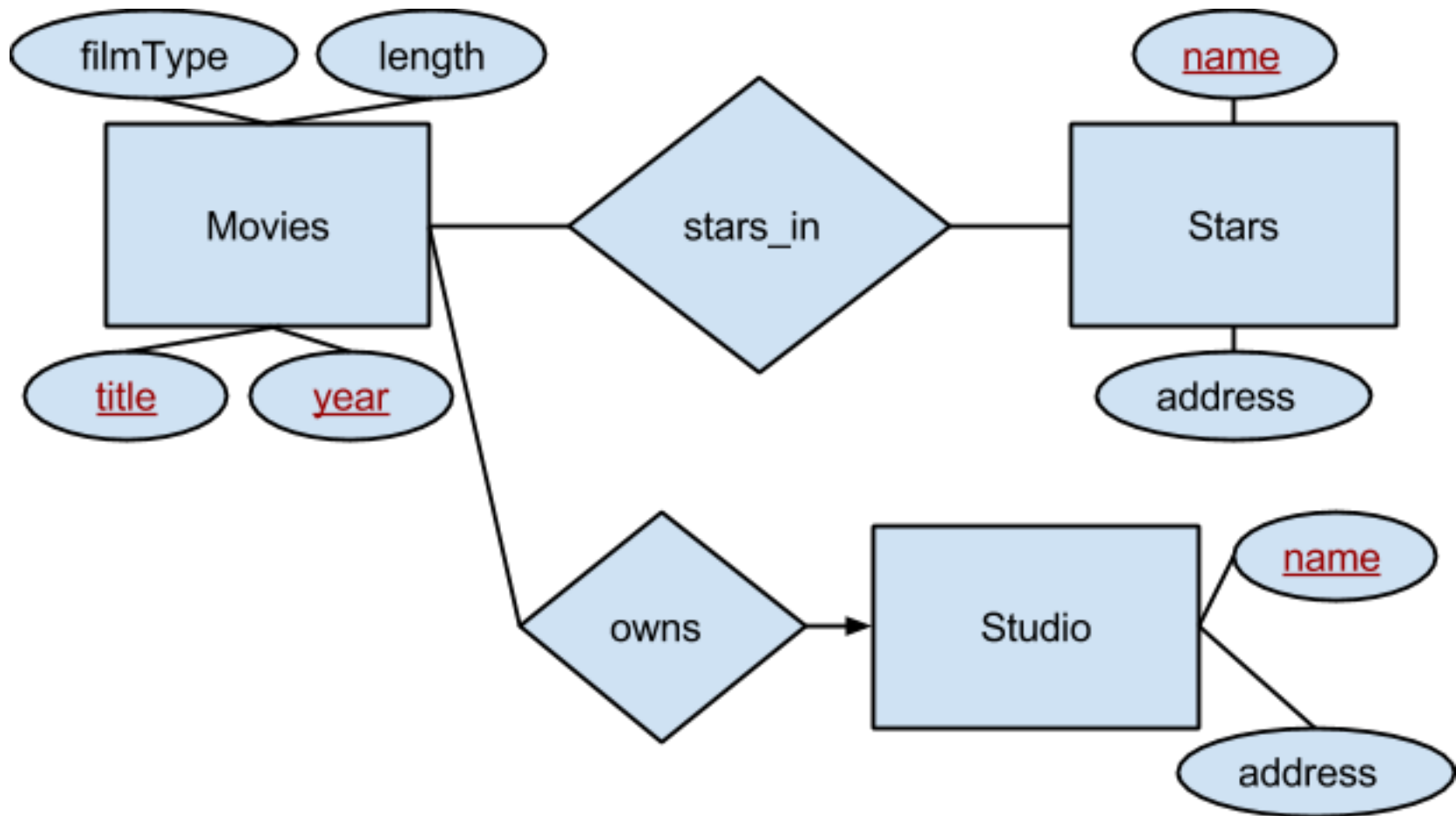
Ограниченията са част от схемата на БД

- ▶ Това дали един атрибут да бъде ключ или не се определя при дизайна на базата от данни, а не от моментното състояние на базата от данни (т.е. какви данни има въведени в базата към момента)
- ▶ Веднъж като се дефинира едно ограничение, вмъкване на данни или модифициране на данни в базата, които нарушават ограничението не могат да бъдат направени
- ▶ Ограниченията са в сила в сила за всички инстанции на базата от данни, а не за конкретна инстанция

Представяне на ключове в Е/Р модела

- ▶ Ключовете в Е/Р модела се подчертават
- ▶ Когато два или повече атрибути са подчертани, тогава всички те взети заедно формират ключа
- ▶ Няма нотация за изразяване на кандидат ключовете
- ▶ Подчертава се само първичният ключ
- ▶ Трябва да се внимава при слабите множества от същности, при които ключа се формира не само от собствените атрибути на множеството но и от атрибути на поддържащото множество

Представяне на ключове - пример



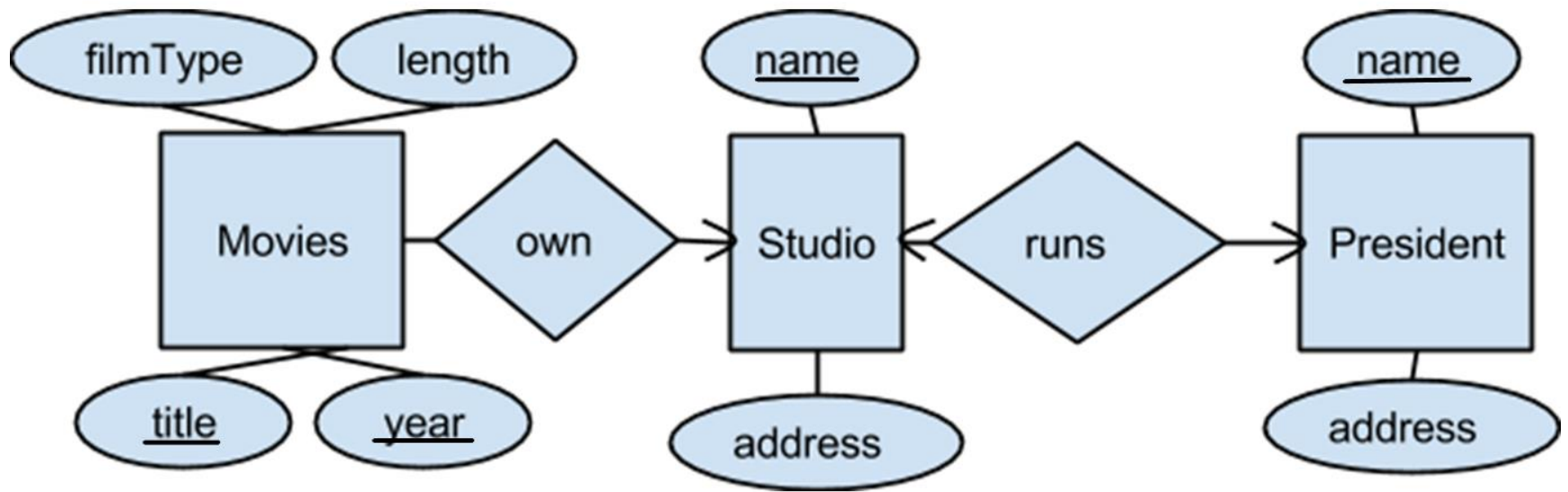
Ограничения по единствена стойност

- ▶ Ограниченията по единствена стойност гарантират уникалност и единственост на стойностите на атрибута
- ▶ Този тип ограничение се поражда от
 - ▶ Стойността на даден атрибут да е уникална.
 - ▶ Връзки от вида много-един, които изискват единственост на елемента към които сочи връзката, в случай че съществува
- ▶ Трябва да отбележим, че има случаи при които стойността за даден атрибут липсва. Тогава като стойност на атрибута се записва `null`
- ▶ В E/R модела няма нотация за `null`. Възможно е до атрибута да се постави нотация, която да отбелязва тези изисквания

Референтна цялостност

- ▶ Налага ограничението, че същността със сигурност съществува
- ▶ В смисъла на предното ограничение, референтната цялостност забранява null стойностите
- ▶ Обикновено те се използват при връзките много-един за да покажат съществуване и единственост на елемент към който връзката реферира
- ▶ Идеята на това ограничение е че то гарантира съществуването на елементите към които се реферира
- ▶ В зависимост от операциите вмъкване или изтриване от множеството което е реферирано са възможни два подхода:
 - ▶ Забраняват се такива операции
 - ▶ Операциите са каскадни

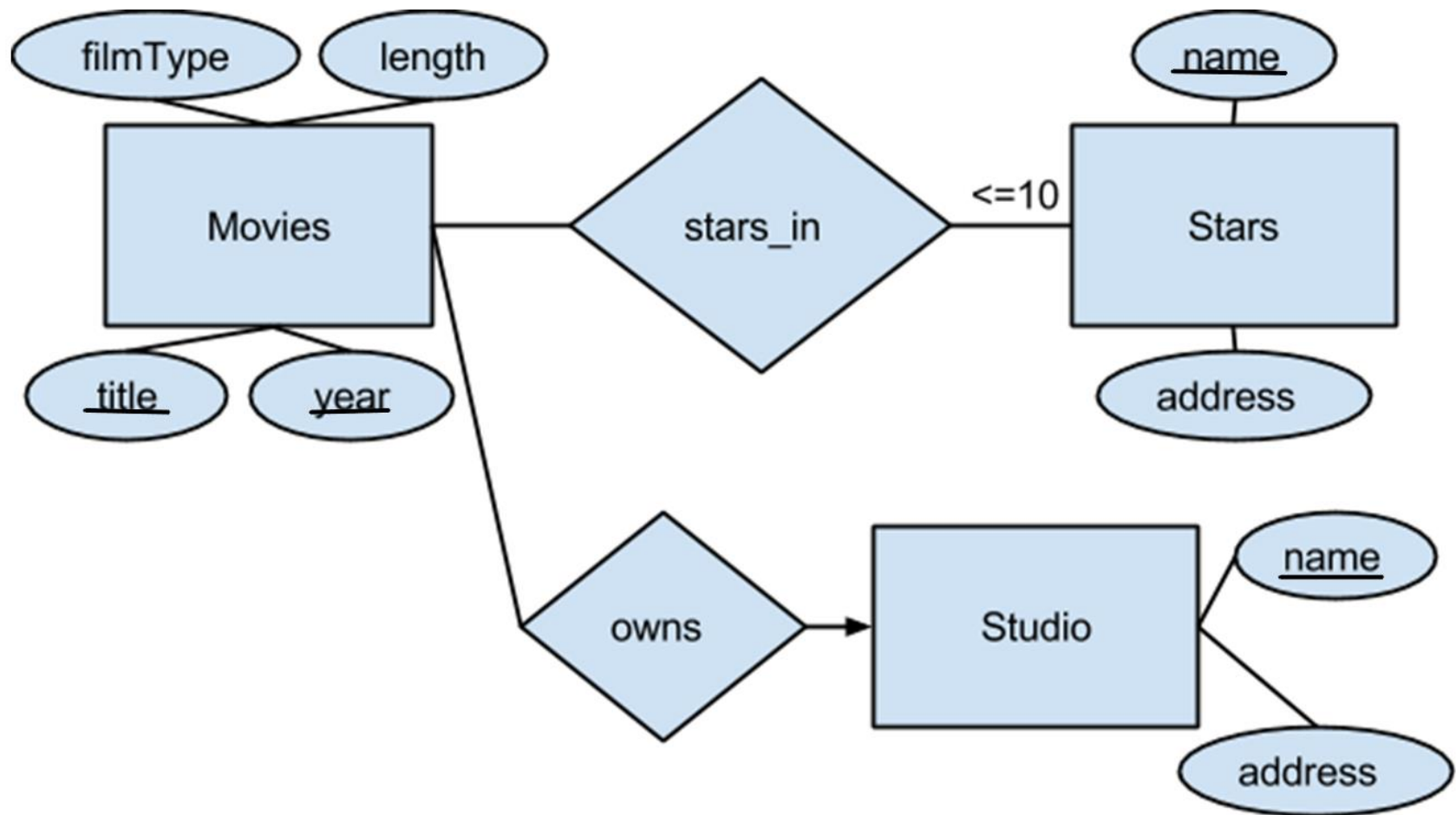
Референтна цялостност - представяне



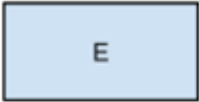
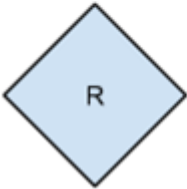


Други ограничения

- ▶ Ограничения по домейн – ограничават стойностите на атрибута да бъдат в дадена област. Например цели числа, низове, изброен тип и т.н.
 - ▶ Няма специална нотация за ограничение по домейн в E/R модела
- ▶ Ограничение на степента на връзката – с колко най-много същности да може да бъде свързана връзката
 - ▶ В E/R модела записваме числото над реброто свързващо двете същности

Други ограничения - пример



Е/R модел нотация

	Множество от същности, където с E е обозначено името на множеството от същности.
	Връзка, където с R е обозначено името на връзката.
	Атрибут, където с A е обозначено името на атрибута.
	Специална връзка, която се използва при моделиране на йерархии на наследяване.

Софтуер за Е/Р диаграми

- ▶ Draw IO - <http://draw.io/>
- ▶ Visio Professional 2013 - <http://office.microsoft.com/en-us/visio/>
- ▶ SmartDraw - <https://cloud.smartdraw.com/>
- ▶ CA ERwin Data Modeler - <https://erwin.com/products/erwin-data-modeler/>



Задача 1

Като използвате нотацията на E/R модела, направете дизайн на база от данни, която съдържа информация за:

- ▶ ⇒ Отбори – те се характеризират с име на отбор, оригинален цвят на фланелка и резервен цвят на фланелка;
- ▶ ⇒ Игрочи – те се характеризират с име на играч и рождена дата;

Да се вземат предвид следните ограничения:

- ▶ ⇒ Всеки отбор има уникално име;
- ▶ ⇒ Всеки играч има уникално име;
- ▶ ⇒ В един отбор може да играят много играчи;
- ▶ ⇒ Един играч може да играе само за един отбор;
- ▶ ⇒ Може да има играч в базата, който да не играе за нито един отбор;
- ▶ ⇒ Всеки отбор има точно един капитан, който също е играч от отбора; Не може да има отбор без капитан;
- ▶ ⇒ Един играч може да е капитан само на един отбор;

Задача 2

Като използвате нотацията на E/R модела, направете дизайн на база от данни, която съдържаща информация за:

⇒ Доставчици: име, град

⇒ Части: номер, цвят

В сила са следните ограничения:

⇒ Един доставчик може да доставя много части

⇒ Една част може да бъде доставяна от много доставчици

⇒ Всеки доставчик, доставя определено количество от частите

Задача 3

Като използвате нотацията на E/R модела, направете дизайн на база от данни, която съдържаща информация за:

⇒ Студенти: ЕГН, име, факултетен номер

⇒ Катедри: име, завеждащ катедра

⇒ Преподаватели: ЕГН, име, научно звание, научна степен

⇒ Курсове: номер на курс, заглавие на курса, описание

Да се вземат предвид следните ограничения:

⇒ Един студент може да посещава много курсове, но не повече от 20. Един курс, може да се посещава от много студенти

⇒ Всеки курс се води от точно един преподавател – титуляр на курса. Един преподавател може да бъде титуляр на много курсове.

⇒ В една катедра може да има много преподаватели. Един преподавател може да бъде точно в една катедра.

Задача 4

Като използвате нотацията на E/R модела, направете дизайн на база от данни, която съдържаща информация за:

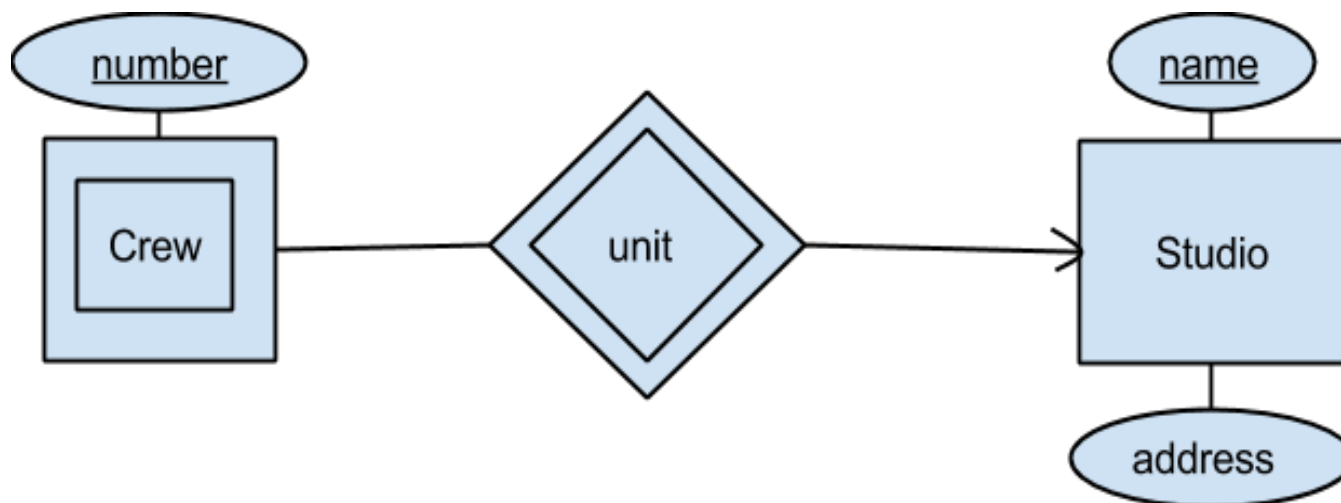
- ▶ ⇒ Отдели: номер на отдел, име на отдел
- ▶ ⇒ Проекти: номер на проект, име, локация
- ▶ ⇒ Служители: номер на служител, име, рождена дата, заплата
- ▶ Да се вземат предвид следните ограничения:
- ▶ ⇒ Всеки служител, работи към един отдел, а в един отдел могат да работят много служители; В базата от данни може да има и служители, които не работят в нито един отдел.
- ▶ ⇒ Всеки служител има шеф, който също е служител.
- ▶ ⇒ Всеки служител, може да работи по много проекти, а по един проект могат да работят много служители.
- ▶ ⇒ В един отдел може да има много проекти, а един проект е точно към един отдел;
- ▶ ⇒ Всеки отдел има ръководител на отдела, който също е служител. Един отдел може да има точно един ръководител на отдела. Не може да има отдел без ръководител. Един служител може да бъде ръководите само на един отдел.

Слаби множества от същности

- ▶ Има условия, при които ключа за едно множество от същности се формира от собствените атрибути на множеството и от атрибути на други множества. Такива множества от същности се наричат слаби
- ▶ Причините (източниците) за наличието на слаби множества от същности са две:
 - ▶ Множества от същности, които се намират в йерархия различна от isa – йерархията
 - ▶ Множества от същности получени в резултат на преобразуването на небинарна връзка към бинарни много-един връзки

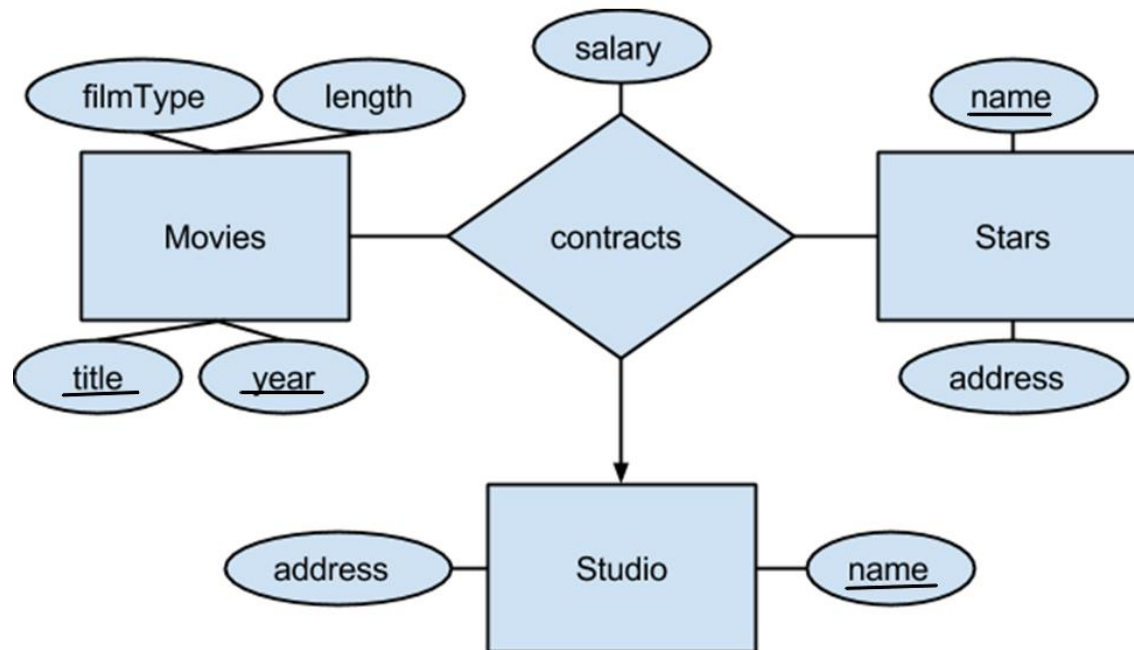
Множества от същности в йерархия

- ▶ Например, едно студио за филми има няколко филмови екипа. Екипите могат да бъдат обозначавани към съответното студио, като – Екип 1, Екип 2, Екип 3 и т.н. Друго студио може да има същите екипи - Екип 1, Екип 2, Екип 3
- ▶ Ето защо номер на екипа не може да бъде ключ за множеството Екипи. За да може уникално да се идентифицира екипа ни е необходимо и името на студиото



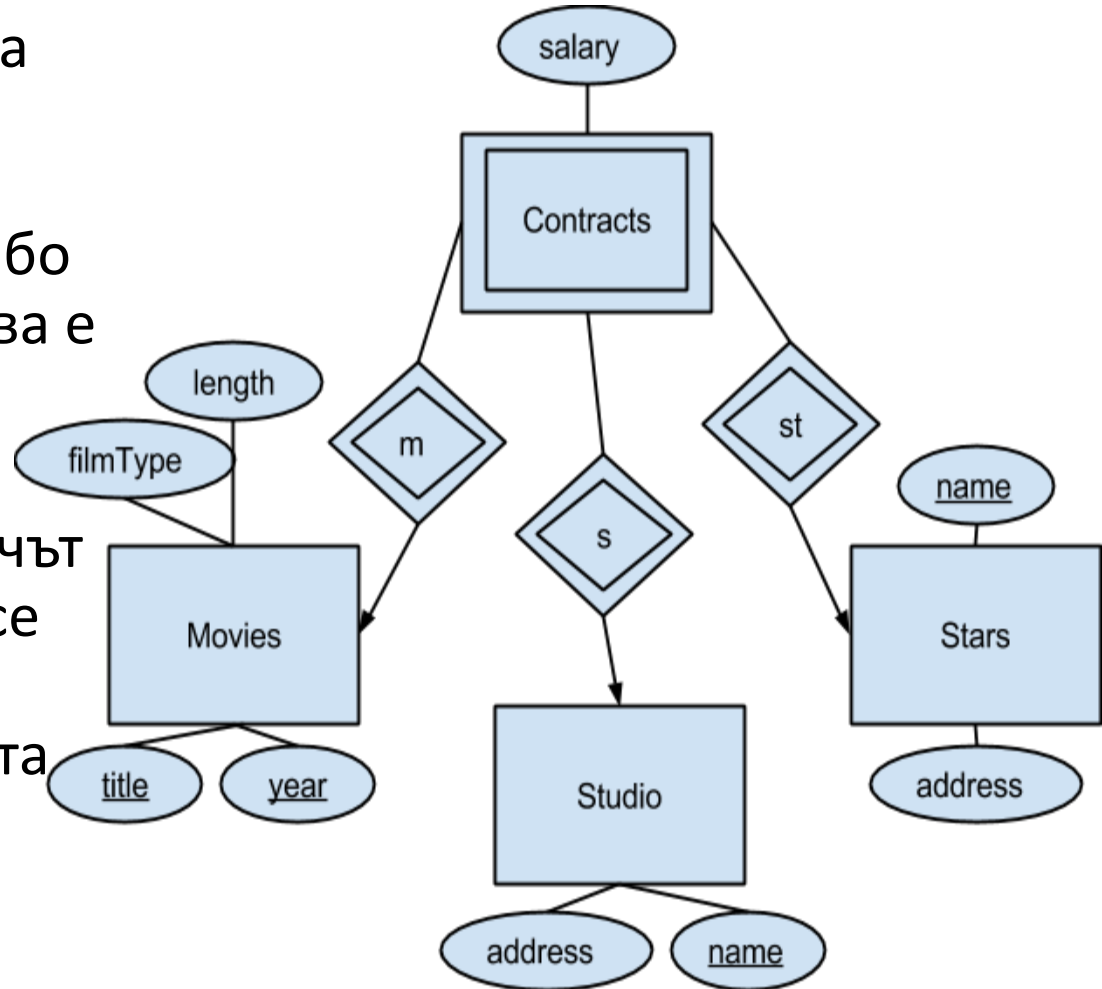
Преобразуване на небинарна връзка

- ▶ Нека разгледаме тринарната връзка Contracts
- ▶ Тя показва, че едно студио може да сключи договор за един или няколко филми в които могат да играят един или няколко актьора, с които също е сключен договор



Преобразуване на небинарна връзка

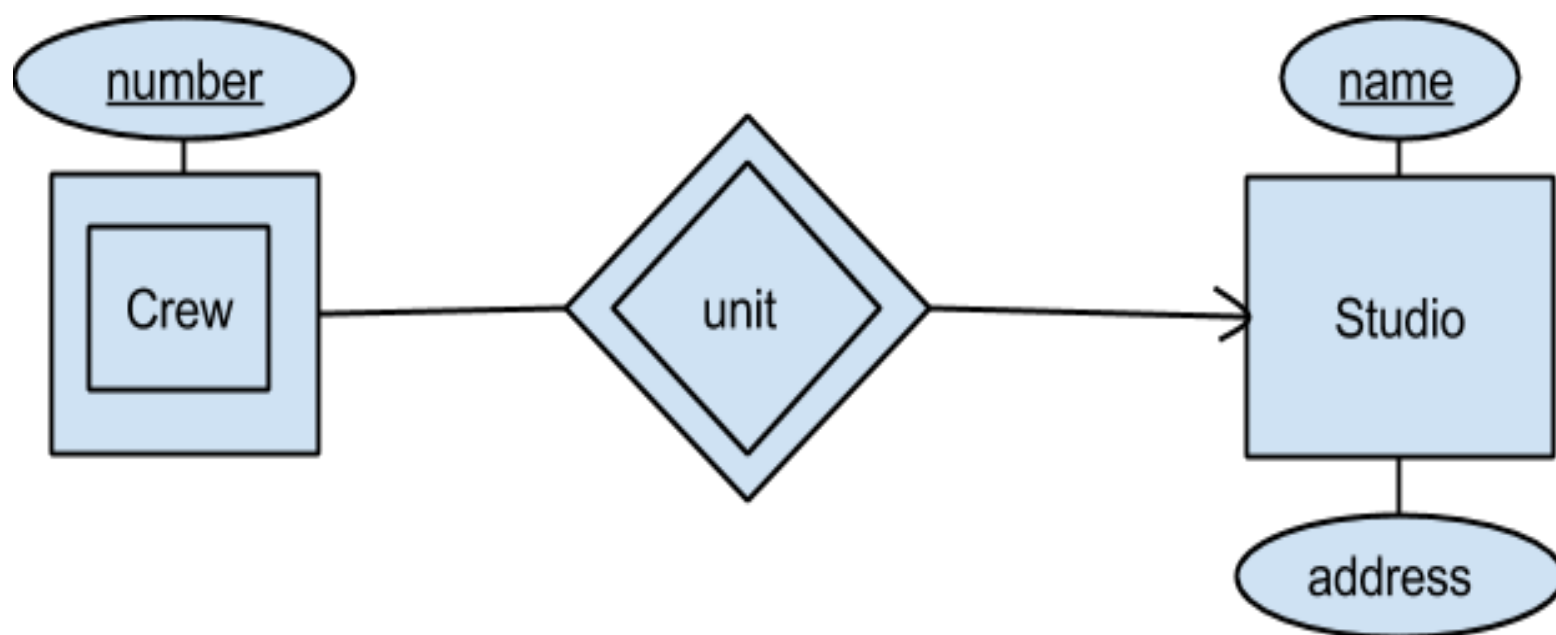
- ▶ При преобразуването на небинарна връзка към бинарни много-един връзки, се получава слабо множество. В случая това е новосъздаденото множество – Contracts
- ▶ При такива случаи, ключът на слабото множество се формира от ключовите атрибути на множествата същности, които небинараната връзка е свързвала



Изисквания към слабите множества от същности

- ▶ Ключът на едно слабо множество от същности се определя по следното правило:
 - ▶ Нула или повече от неговите собствени атрибути
 - ▶ Ключовите атрибути от множествата същности, които се достигат от E посредством много-един връзки. Тези много-един връзки се наричат поддържащи връзки за E
- ▶ Ако R е много-един връзка от E до някое множество от същности F , за да бъде R поддържаща връзка за E , трябва да бъдат изпълнени следните условия:
 - ▶ R трябва да е бинарна много-един връзка от E до F
 - ▶ R трябва да има референтна цялостност от E до F
 - ▶ Атрибутите, които F предоставя за ключ на E , трябва да са ключови за F
 - ▶ В случай, че F също е слабо множество, тогава рекурсивно взимаме ключовете по горните правила
 - ▶ Ако има няколко различни поддържащи връзки от E към F , тогава всяка връзка се използва да предостави копие на ключов атрибут от F

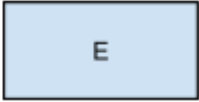
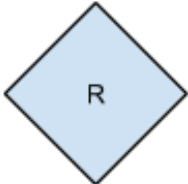

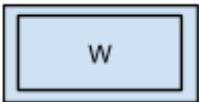
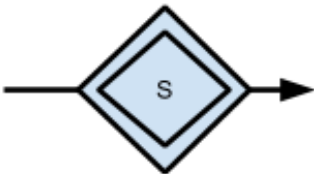

Пример



Нотация на слабите множества от същности

- ▶ Използва се следната нотация за означаване на слаби множества от същности и за дефиниране на неговите ключови атрибути:
 - ▶ Ако едно множество от същности е слабо, то изобразява като правоъгълник с двойни граници
 - ▶ Поддържащите връзки за слабото множество, също се изобразяват с двойни граници
 - ▶ Ако атрибут е част от ключа на слабото множество то, той се подчертава
- ▶ Ако обобщим – винаги, когато използваме правоъгълник с двойни граници, това е слабо множество от същности. Ромб с двойни граници се използва само за означение на поддържащи връзки. Възможно е да имаме и връзка много-един от слабото множество към друго множество и тази връзка да не е поддържаща.

Е/Р модел нотация

	Множество от същности , където с E е обозначено името на множеството от същности.
	Връзка , където с R е обозначено името на връзката.
	Атрибут , където с A е обозначено името на атрибута.
	Слабо множество , където с W е обозначено името на слабото множество.
	Поддържаща връзка , където с S е обозначено името на поддържащата връзка.
	Специална връзка , която се използва при моделиране на йерархии на наследяване.

Задача 6

Да се направи E/R диаграма за база от данни за верига от магазини. Известно е, че:

- ▶ Всяка **Верига от магазини** се характеризира с име – уникален идентификатор;
- ▶ Всяка верига от магазини има **Магазини** в различни градове, които магазини се характеризират с номер на магазин, град и адрес. Номерът на магазина не определя еднозначно всеки магазин. Например две различни вериги от магазини могат да имат магазини с едни и същи номера;
- ▶ Към всяка Верига от магазини работят **Служители**. Те се характеризират с ЕГН – уникален идентификатор, име, адрес и заплата;
- ▶ Всяка Верига от магазини е обслужвана от **Доставчици**. Те се характеризират с ЕГН – уникален идентификатор, име на доставчик и име на компания, в която работят;
- ▶ Във всеки Магазин от веригата Служителите работят на смени (в **Екипи**). Всеки Екип се характеризира с номер на екип. Номерът на екипа не определя еднозначно всеки екип. Например два различни магазина могат да имат екипи с един и същи номер.

В сила са следните ограничения:

- ▶ Към една Верига от магазини могат да работят много Служители. Един Служител може да работи само в една Верига от магазини;
- ▶ Към една Верига от магазини може да има много Доставчици. Един доставчик може да обслужва много Вериги от магазини;
- ▶ Служителите и Доставчиците са Граждани и имат общи характеристики: ЕГН и име.