



Преобразуване от модел "Същност-  
връзки" 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 релационния модел

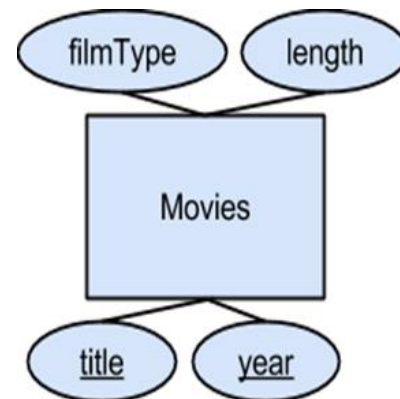
---

- ▶ **Релационният 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 релационни 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 релационната схема.
- ▶ Релационният 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 *isa* 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
- ▶ **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.

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



# Преобразуване на Е/Р към релационен модел

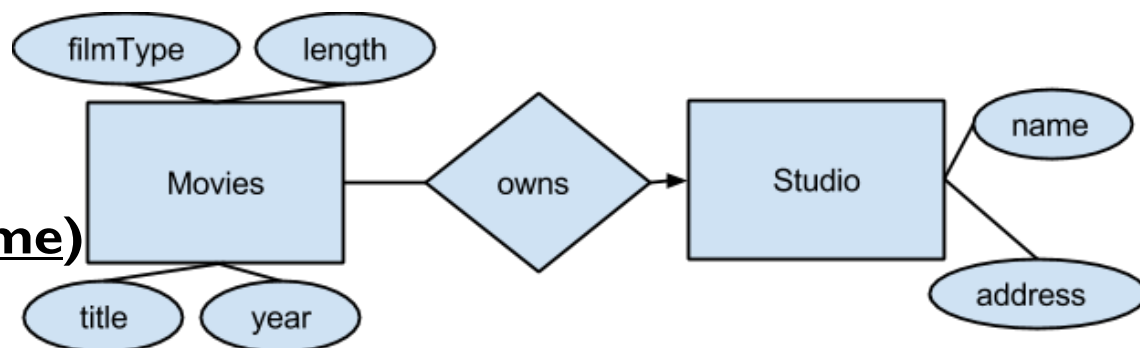
- ▶ **Да разгледаме преобразуване на връзки.**
- ▶ Връзките в релационен модел, също се представят чрез релация.
- ▶ Релацията за дадена връзка  $R$  има следните атрибути:
  - ▶ За всяко множество от същности, участващо във връзката  $R$  взимаме ключовите атрибути и те стават атрибути на релацията  $R$
  - ▶ 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.

- ▶ Пример за връзка

Връзката Owns

се преобразува до

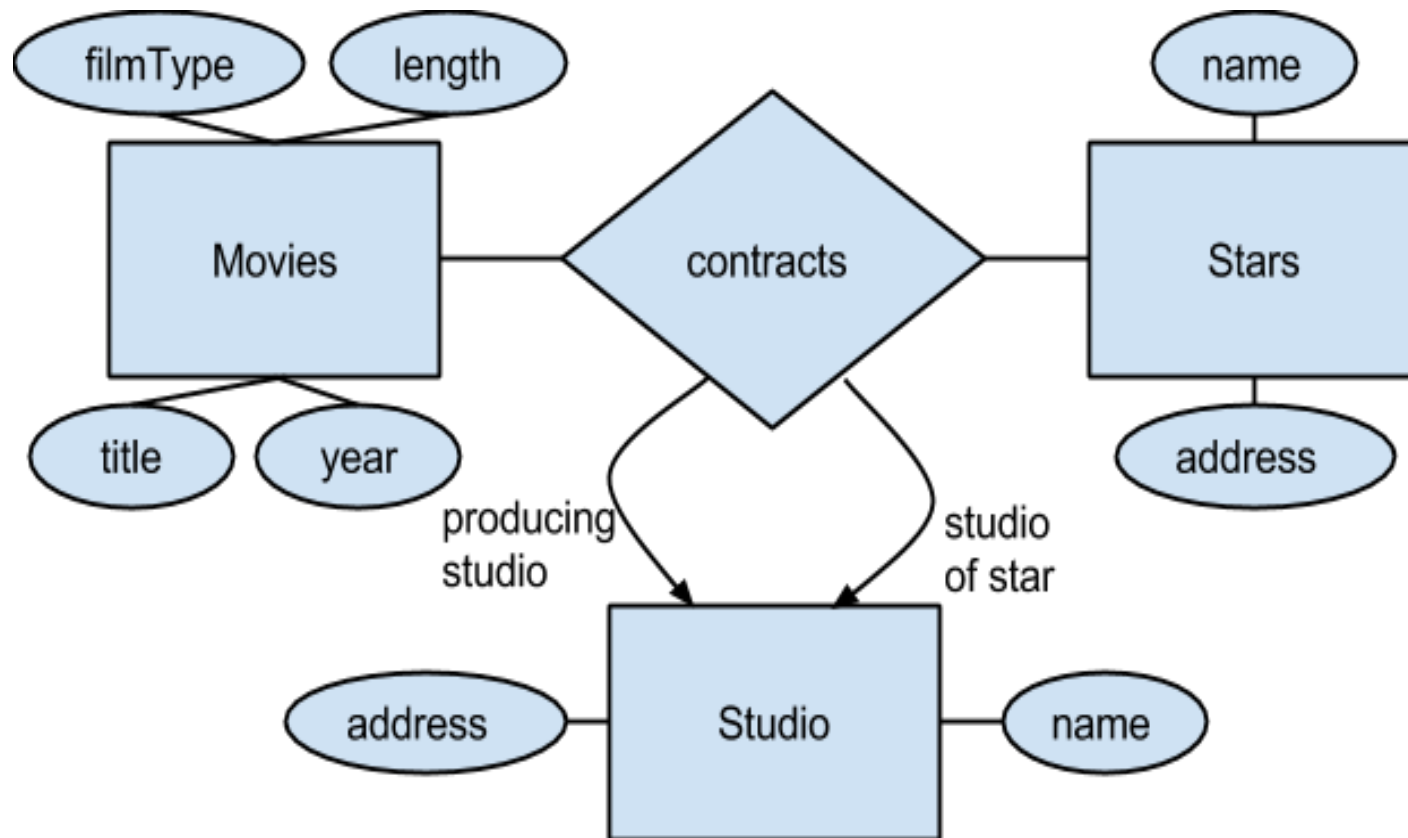
**Owns(title, year, studioName)**



# Преобразуване на Е/Р към релационен модел

- ▶ Пример за връзка с роли

**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
- ▶ Нека множеството от същности  $E$  е свързано с  $F$  посредством връзката  $R$ , която е много-един от  $E$  към  $F$ .
- ▶ И двете релации получени при преобразуването на  $E$  и  $R$ , ще съдържат ключовите атрибути на множеството от същности  $E$ , които уникално определят всяка същност в  $E$ . При релацията  $R$  ще имаме ключовите атрибути на  $E$ , ключовите атрибути на  $F$  и атрибутите на релацията  $R$ .
- ▶ Понеже връзката  $R$  е много-един, всички тези атрибути на връзката  $R$  ще имат стойности, които са уникално определени от ключовете за  $E$  и по тази причина може да обединим с релацията  $E$ .

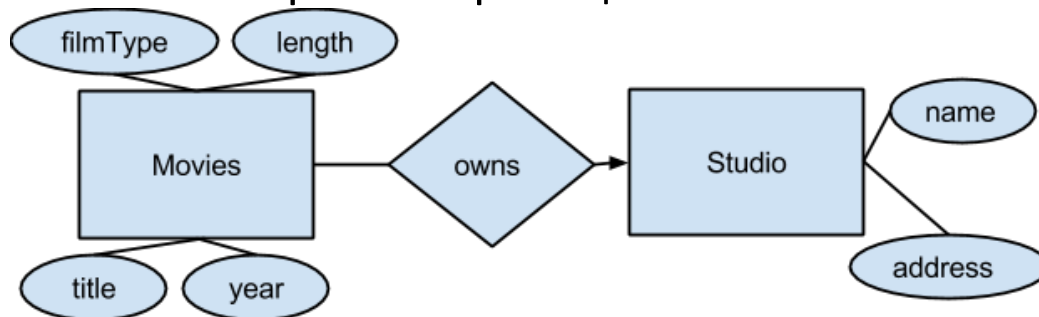
# Комбиниране на релации

## ► Правилата са следните:

1. В новата релация се включват всички атрибути на множеството E
2. Ключовите атрибути на множеството от същности F
3. Всички атрибути принадлежащи на връзката R
4. Именуваме новата релация с името на множеството от същности E

► За същностите от множеството E, които не са свързани с нито една същност от F, атрибутите по 2 и 3 точка ще имат null стойност в кортежите за E. Null стойностите се поставят в ситуации в които стойността липсва или не е известна. Null стойностите са неформална част в релационния модел.

## ► Пример за комбиниране на релации





# Комбиниране на релации

► Решение:

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

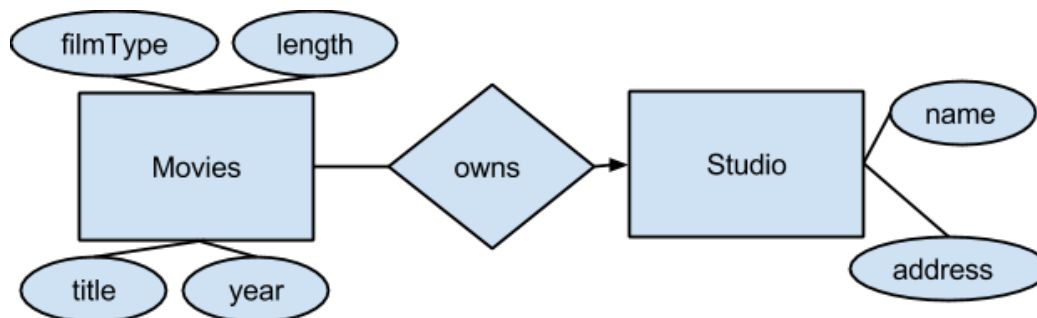
**Studio(name, address)**

**Owns(title, year, studioName) X**

► Оптимизиран релационен модел

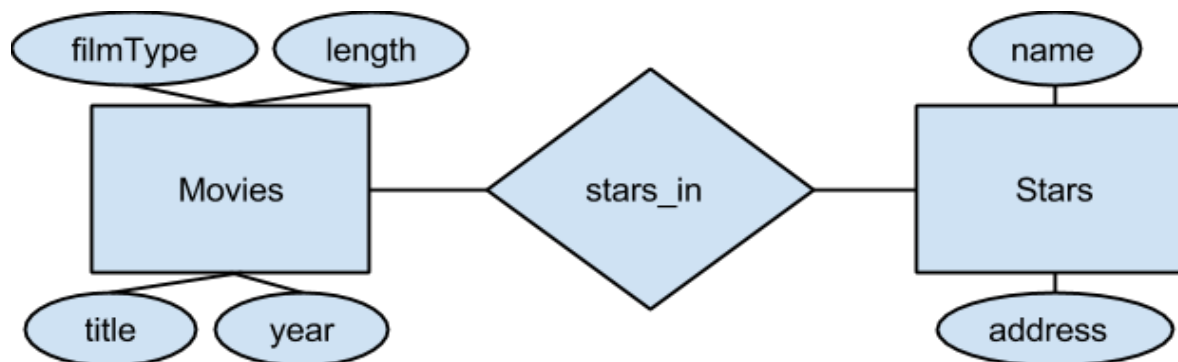
**Movies(title, year, length, filmType, studioName)**

**Studio(name, address)**



# Комбиниране на релации

- ▶ Не е добра идея да оптимизираме връзки много-много.
- ▶ Обединяването на такива релации води до излишества



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

**Stars(name, address)**

**Stars\_in(title, year, starName)**

- ▶ Оптимизиране на релационен модел (грешно !!!)

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

Ако в един филм играят няколко актьора, информацията за този филм ще бъде повторена. Имаме излишество

# Задача 1

---

Начертайте E/R диаграма за БД, съдържаща информация за една компания.

⇒ Employees: id, name phone

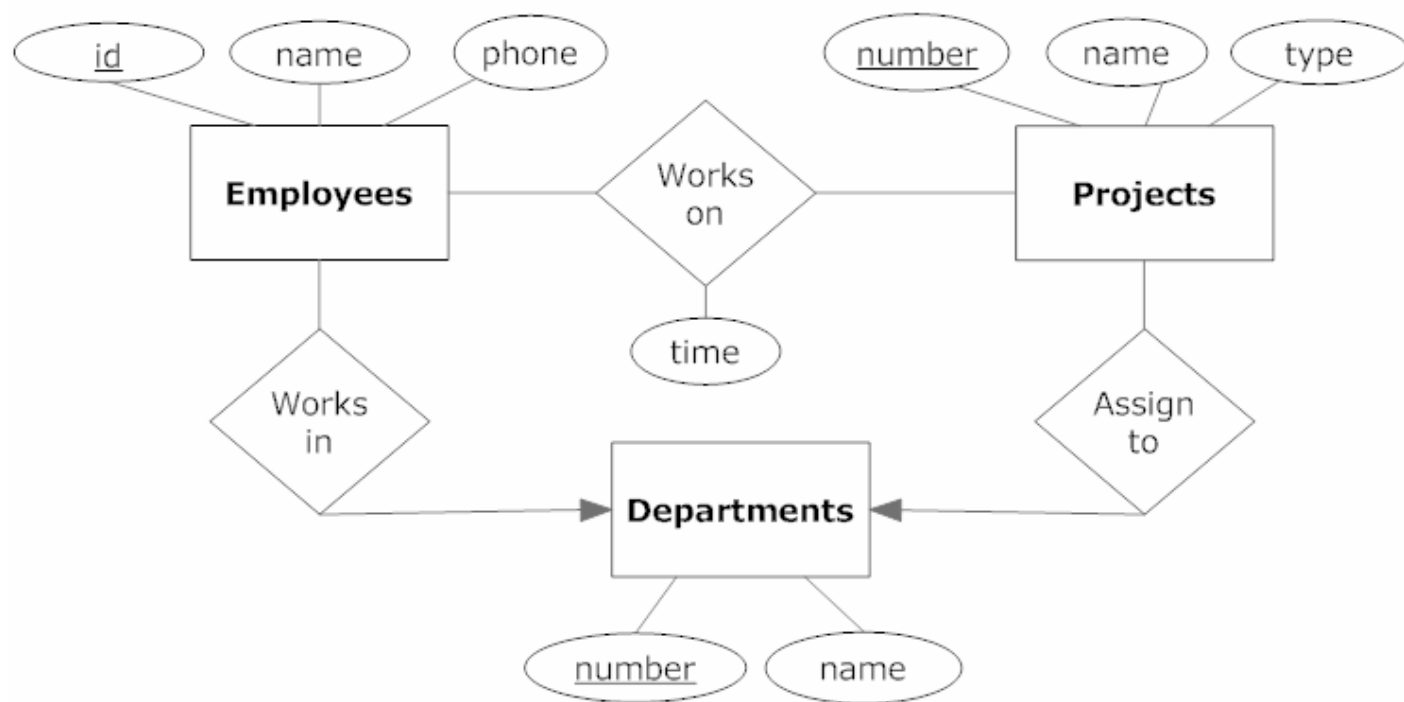
⇒ Projects: type, number, name

⇒ Department: number name

Моделирайте базата от данни и я преобразувайте към релационни схеми.

# Решение на Задача 1

---



# Решение на Задача 1

---

- ▶ Employees (id, name, phone);
- ▶ Projects (number, name, type);
- ▶ Departments (number, name);
- ▶ WorksOn (id, emplID, projectNumber, time);
- ▶ WorksIn (emplID, deptnNo);
- ▶ AssignTo (projectID, deptNo);

Оптимизиране на релационния модел

- ▶ Employees (id, name, phone, deptNo);
- ▶ Projects (number, name, type, deptNo);
- ▶ Departments (number, name);
- ▶ WorksOn (id, emplID, projectNumber, time);

## Задача 2

---

Начертайте E/R диаграма за БД, съдържаща информация за закупуване на билети.

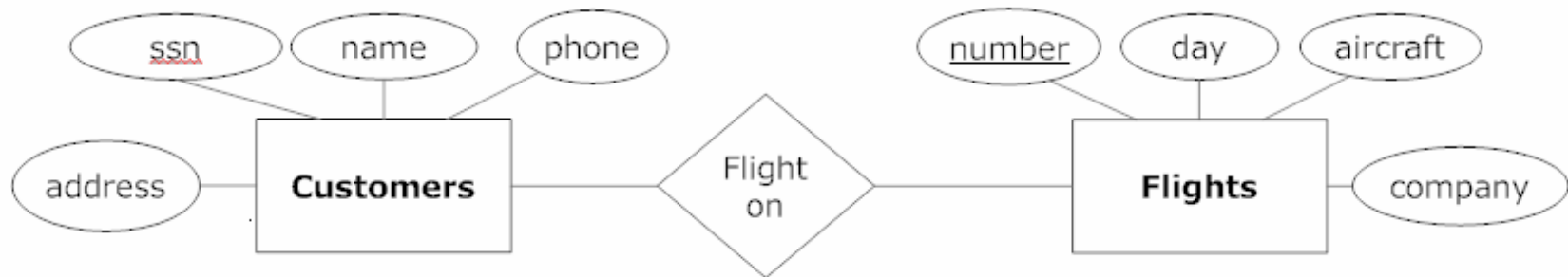
⇒ Customers: ssn, name, phone, address

⇒ Flights: number, day, aircraft, company

Моделирайте базата от данни и я преобразувайте към релационни схеми

## Решение на Задача 2

---



- ▶ **Customers** (ssn, name, phone, address)
- ▶ **Flights** (number, day, aircraft, company)
- ▶ **FightOn** (customerSSN, flightNumber)

## Задача 3

---

Да се направи E/R диаграма на база от данни съдържаща информация за „Библиотека”. В библиотеката се съдържа информация за

⇒ Книги: заглавие, isbn, година на издаване

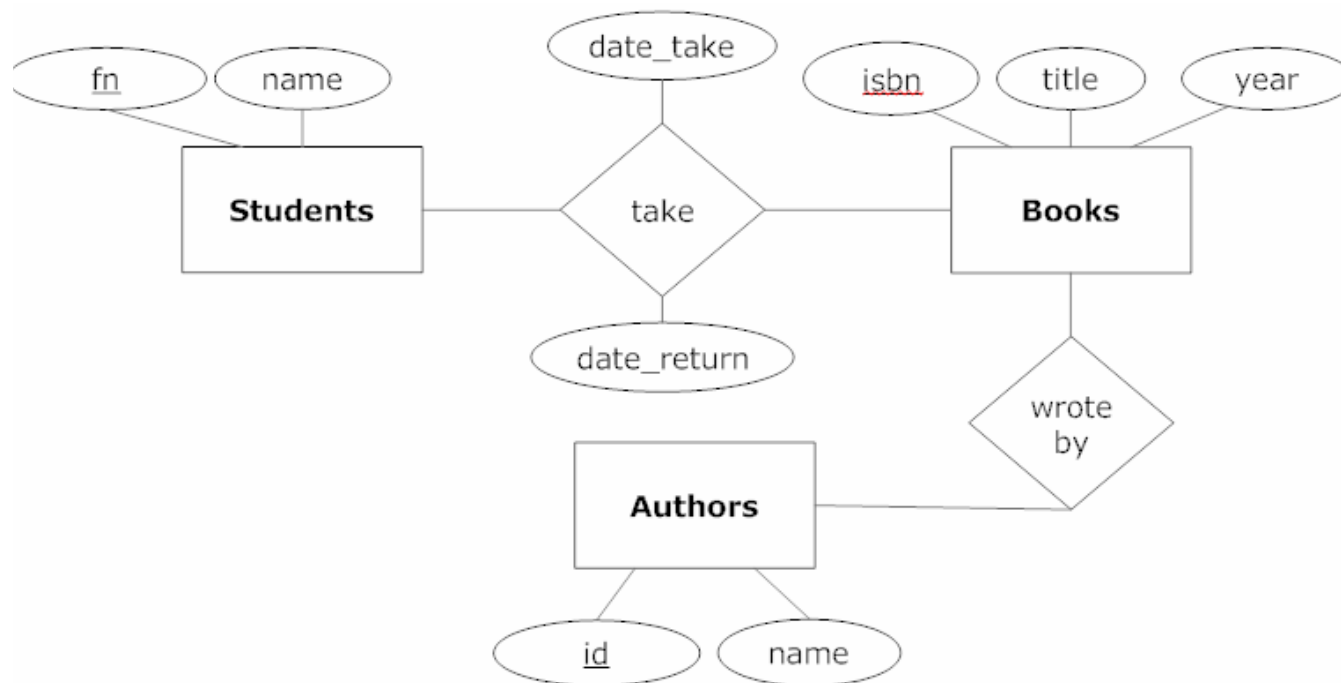
⇒ Автори: име

⇒ Студенти: име, факултетен номер.

Моделирайте базата от данни и я преобразувайте към релационни схеми.



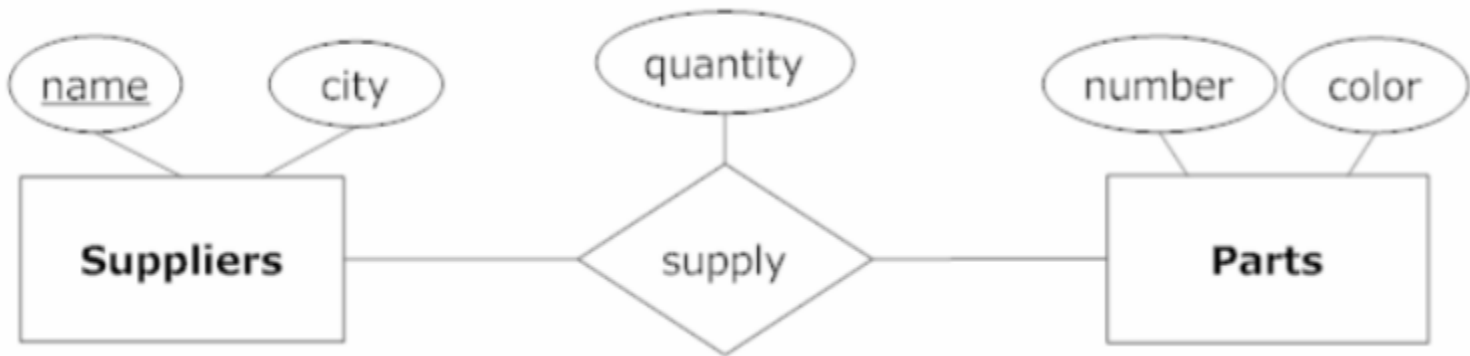
# Решение на Задача 3



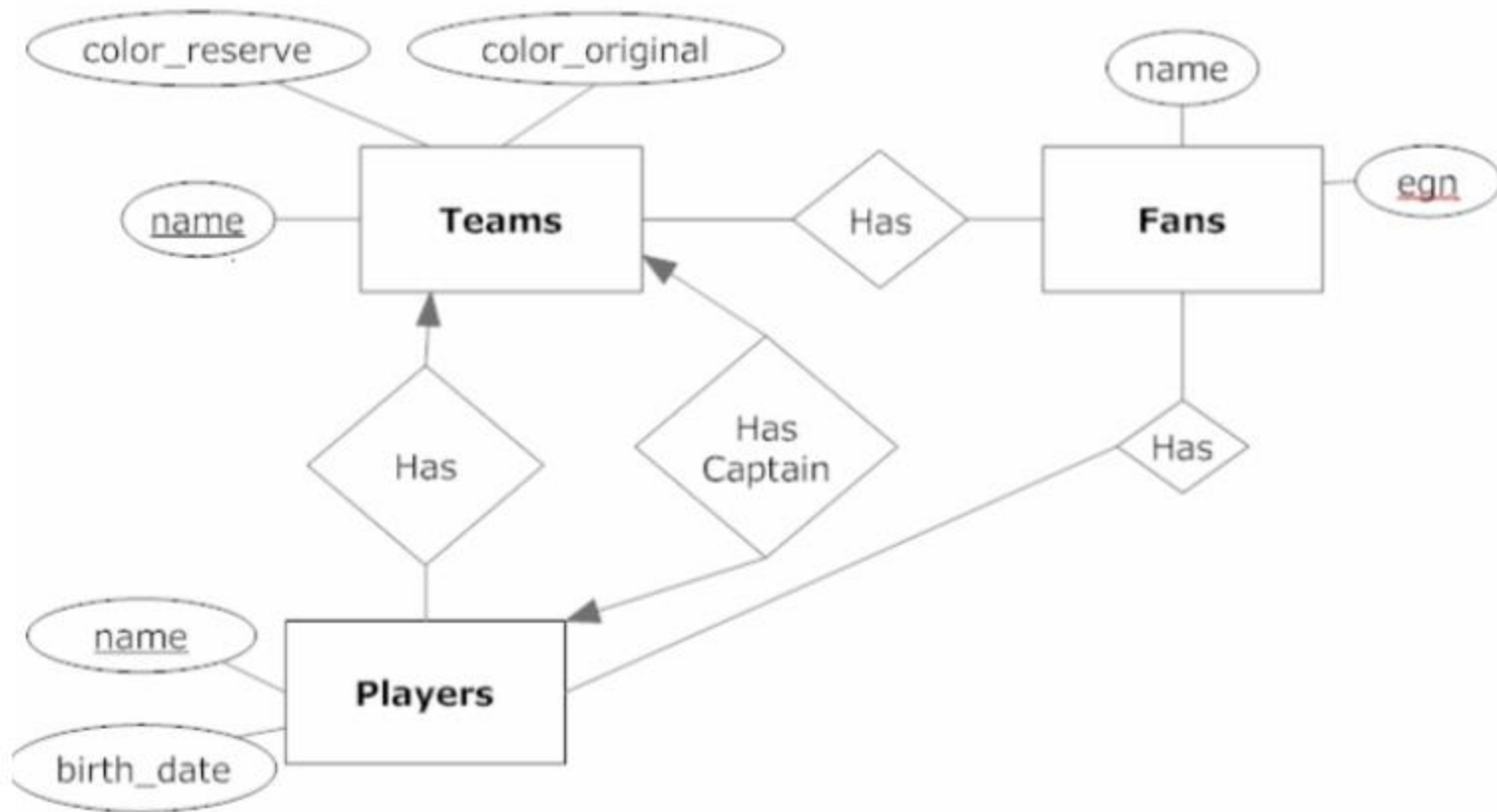
- ▶ **Students** (fn, name)
- ▶ **Books** (isbn, title, year)
- ▶ **Authors** (id, name)
- ▶ **Take** (studentFN, bookID, date\_take, date\_return);
- ▶ **WroteBy** (bookID, authorID);

## Задача 4

---



## Задача 5



## Задача 6

---

- ▶ Да се направи дизайн на база от данни за филми, която ще съдържа следната информация:
  - ▶ Movie: title, year, length, incolor
  - ▶ Studio: name, address, presC#
  - ▶ MovieStar: name, address, gender, birthdate
  - ▶ MovieExec: cert#, name, address, networth
- ▶ Да се преобразува в релационен модел, да се определят ключовете: primary key, foreign key и типа на данните (домейна на данните)

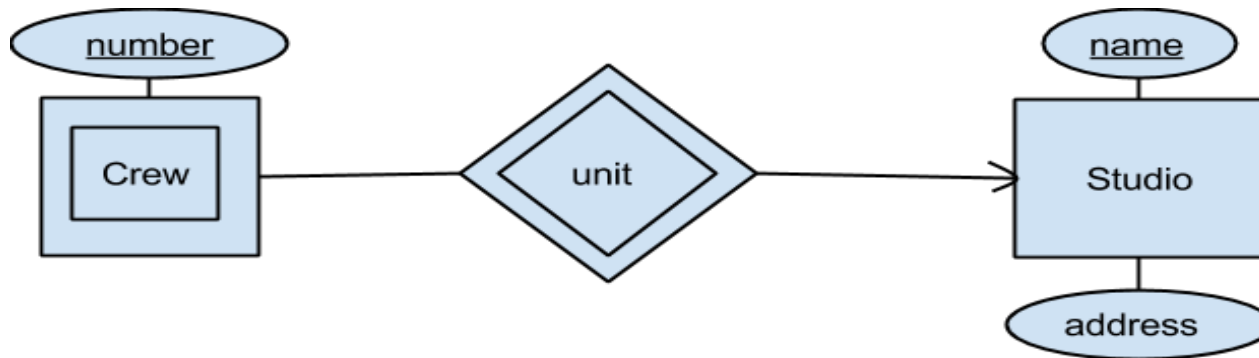
# Преобразуване на слаби множества от същности

---

- ▶ 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

# Пример за преобразуване на слабо множество

- ▶ **Studio(name, address)**
- ▶ **Crews(number, studioName)**
- ▶ **Unit(number, studioName)** X (излишно)



## Задача 6

---

Да се направи E/R диаграма за следната БД:

⇒ Курс: стая, номер

⇒ Катедра: име, ръководител катедра

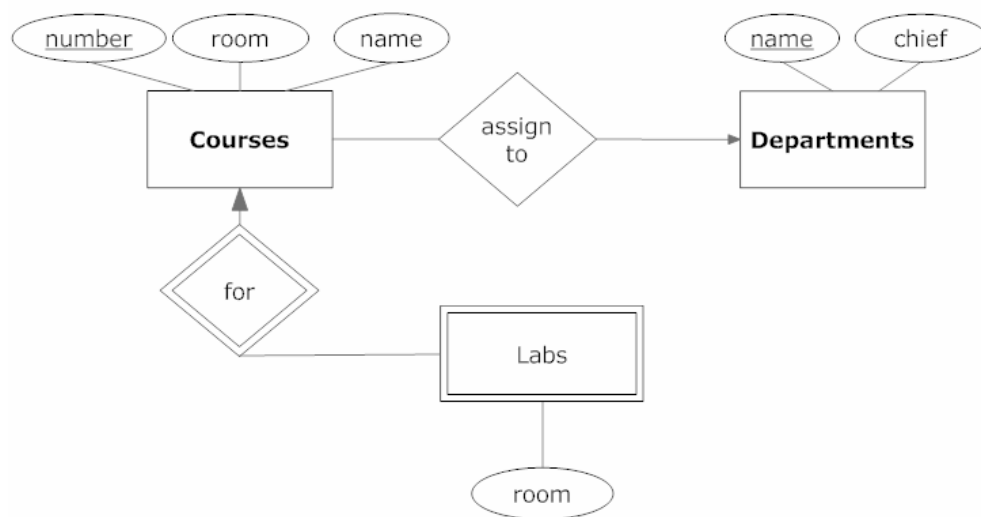
⇒ Лабораторни: компютърна зала

Между горе дефинираните множества същности са в сила следните връзки:

⇒ Всеки курс се предлага от точно една катедра.

Преобразувайте, E/R диаграмата към релационен модел

# Решение на Задача 6



- ▶ **Courses** (number, room, name);
- ▶ **Departments** (name, chief);
- ▶ **Labs**(courseNo, room);
- ▶ **AssignTo** (courseNo, deptName);

Оптимизация на реляционния модел

- ▶ **Courses** (number, room, name, deptName);
- ▶ **Departments** (name, chief);
- ▶ **Labs**(courseNo, room);



# Преобразуване на йерархии от подкласове

---

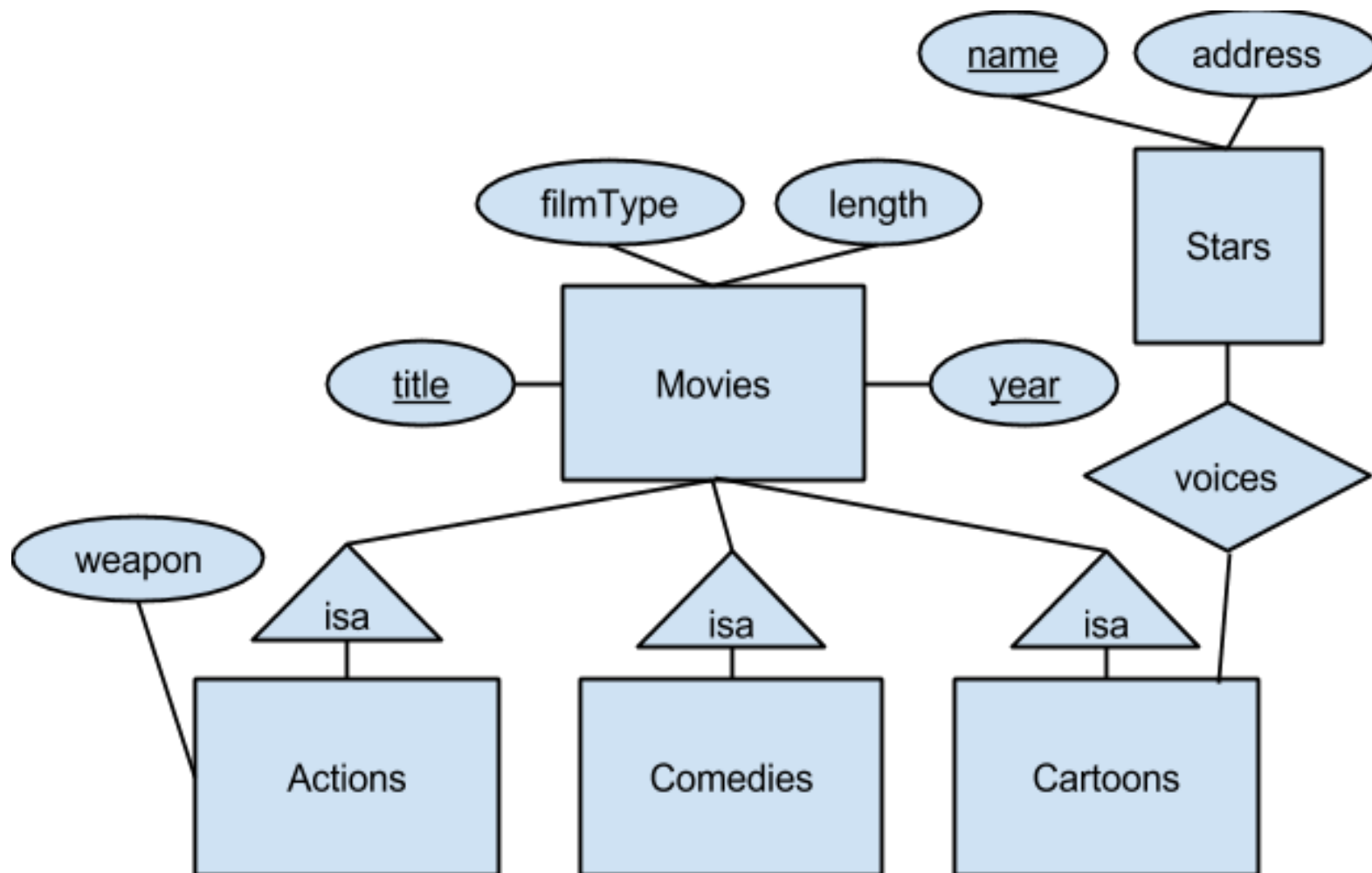
- ▶ При множества от същности организирани в isa-йерархия имаме корен на йерархията. Това множество от същности (коренът) съдържа ключ, който уникално идентифицира всяка същност в йерархията.
- ▶ Когато имаме isa-йерархия (връзка) на множества от същности има три различни подхода за преобразуването на isa-връзката. Те са:
  - ▶ Релационен подход
  - ▶ Обектно-ориентиран подход
  - ▶ NULL-подход
- ▶ 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.

## Е/Р 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.

# Подход Е/Р - Задача



## Е/Р подход - Решение

---

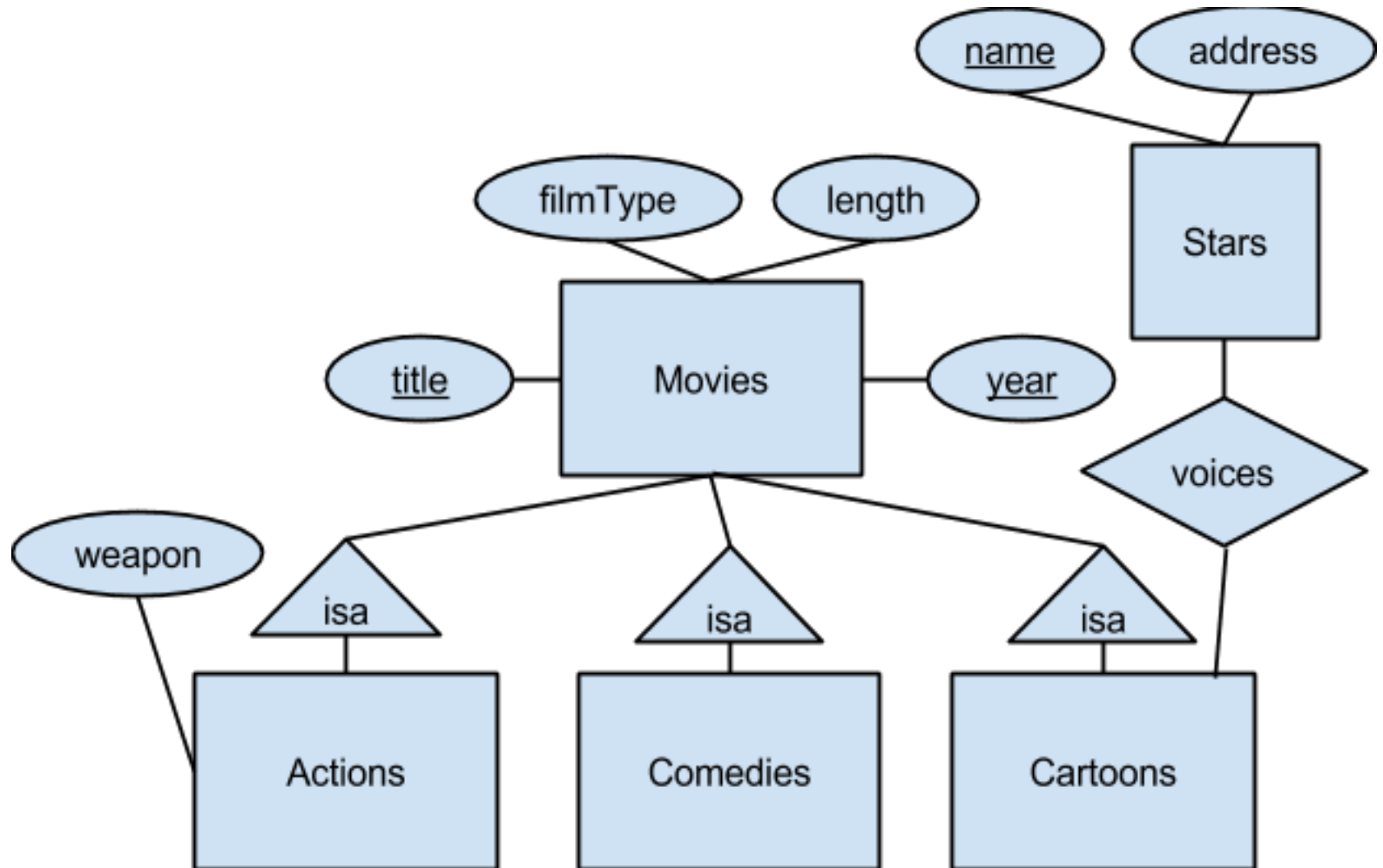
- ▶ **Movies (title, year, length, filmType)**
  - ▶ **Actions (title, year, weapon)**
  - ▶ **Comedies (title, year)**
  - ▶ **Cartoons (title, year)**
  - ▶ **Stars (name, address)**
  - ▶ **Voices(title, year, name)**
- 
- ▶ Ако един филм е едновременно и екшън и комедия и анимационен, то за него 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  $N$

# Обектно-ориентиран подход - 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

---

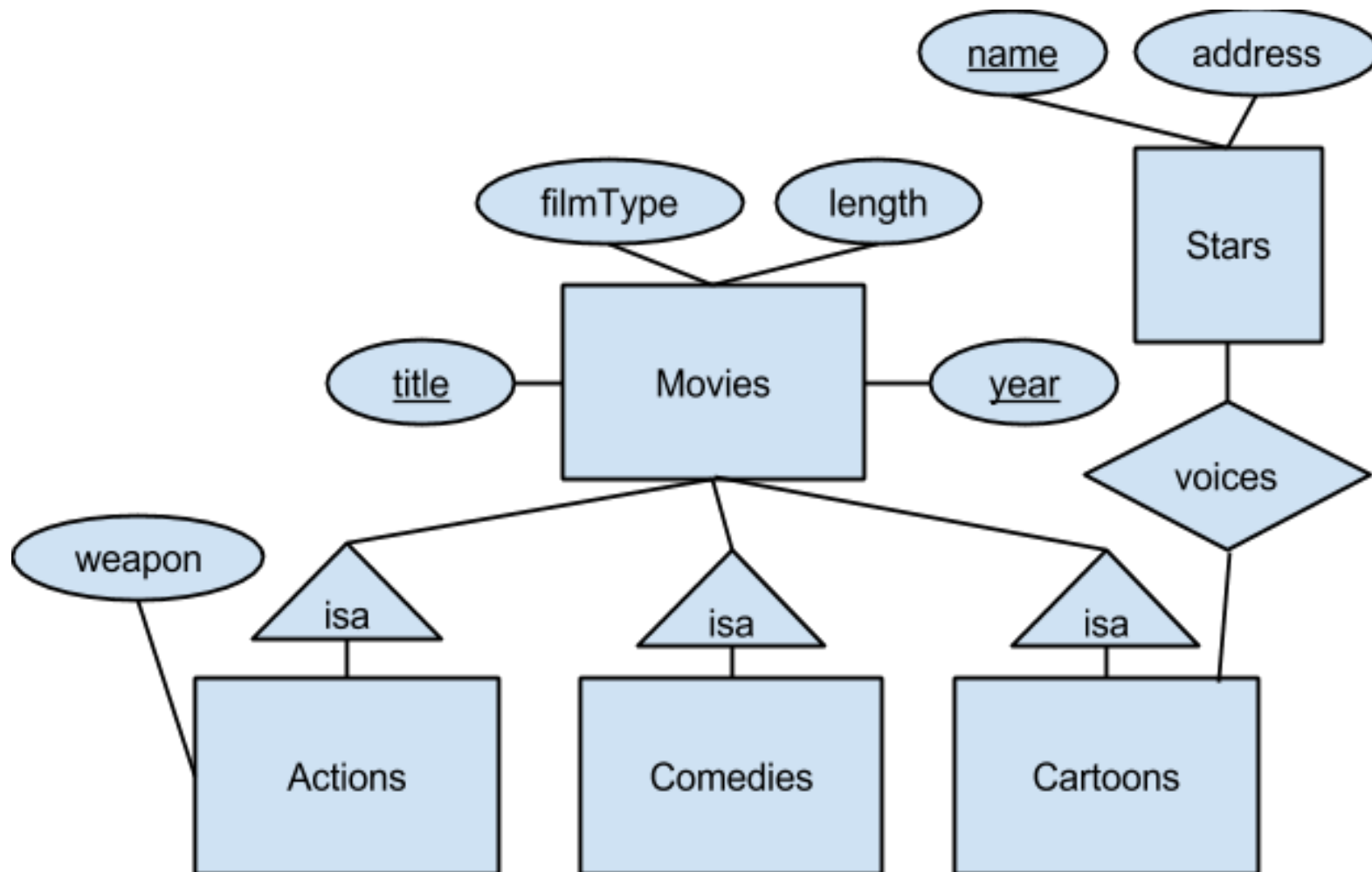
- ▶ Въпреки че релациите по-долу имат еднакви атрибути, ние не можем да ги обединим в една релация
  - ▶ `Movies(title, year, length, filmType)`
  - ▶ `Movies_Cartoons(title, year, length, filmType)`
  - ▶ `Movies_Comedies(title, year, length, filmType)`
- ▶ Ако го направим, ще загубим информация за 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

# NULL подход - Задача



## NULL подход - Решение

---

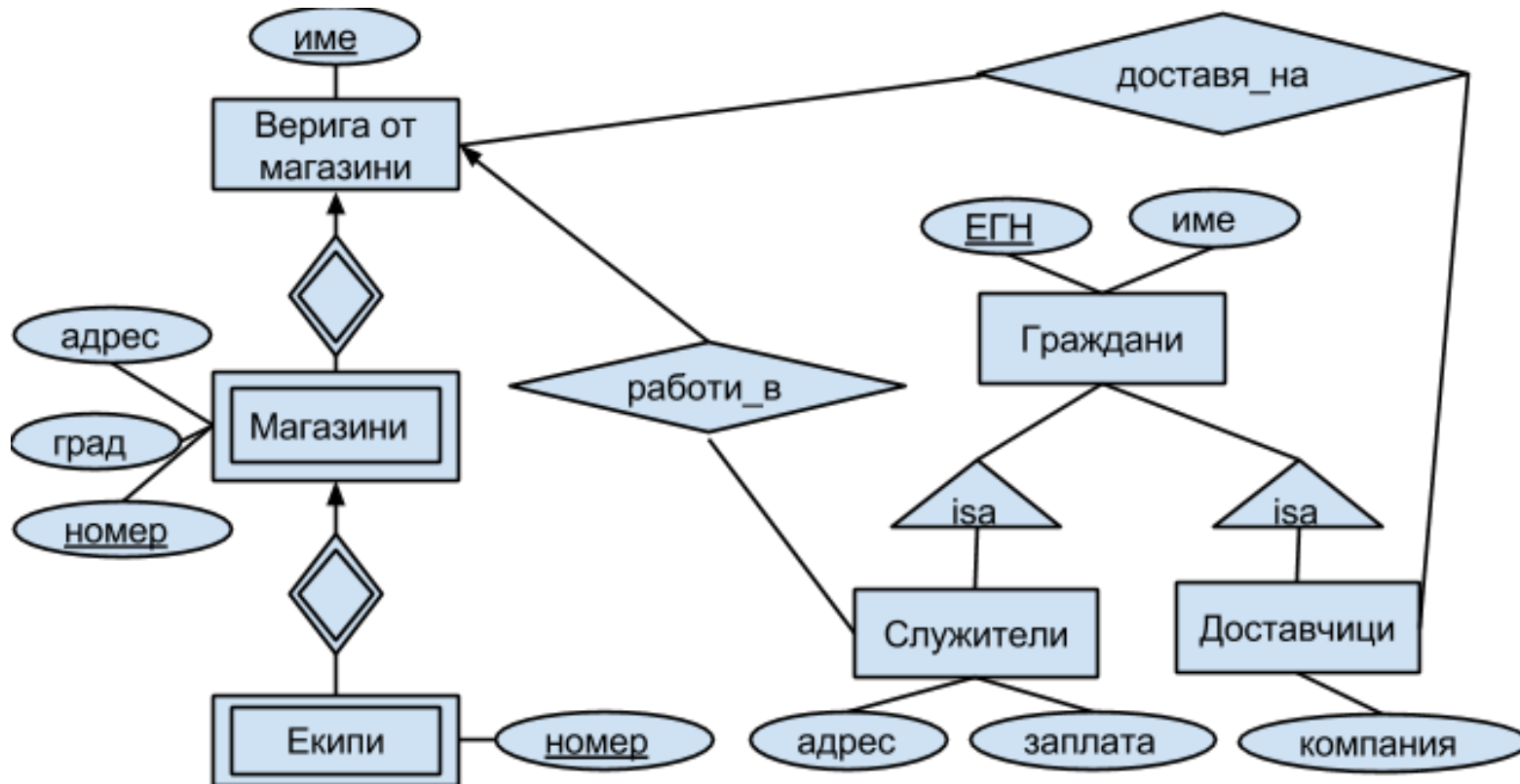
**Movies(title, year, length, filmType, weapon)**

**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.

# Задача 7



# Решение на Задача 7

---

- ▶ Верига\_от\_магазини (име)
- ▶ Магазин (име на верига, номер, град, адрес)
- ▶ Екип (име на верига, номер на магазин, номер)

**Вариант 1** - E/R подход за преобразуване на isa-йерархията:

- ▶ Граждани (ЕГН, име)
- ▶ Служители (ЕГН, адрес, заплата)
- ▶ Доставчици (ЕГН, компания)

**Вариант 2** - Обектно-ориентиран подход за преобразуване на isa-йерархията:

- ▶ Граждани (ЕГН, име)
- ▶ Граждани\_Служители (ЕГН, име, адрес, заплата)
- ▶ Граждани\_Доставчици (ЕГН, име, компания)
- ▶ Граждани\_Служители\_Доставчици (ЕГН, име, адрес, заплата, компания)

**Вариант 3** - NULL-подход за преобразуване на isa-йерархията:

- ▶ Граждани (ЕГН, име, адрес, заплата, компания)

# Решение на Задача 7

---

Ще използваме E/R подхода. Така решението придобива вида:

- ▶ Верига\_от\_магазини (име)
- ▶ Магазин (име на верига, номер, град, адрес)
- ▶ Екип (име на верига, номер на магазин, номер)
- ▶ Граждани (ЕГН, име)
- ▶ Служители (ЕГН, име, адрес, заплата)
- ▶ Доставчици (ЕГН, име, компания)
- ▶ Доставка\_на (ЕГН на доставчик, име на верига)
- ▶ Работи\_в (ЕГН на служител, име на верига)

**Забележка 1:** Връзката Работи\_в е много–едно и може да бъде оптимизирана.

# Решение на Задача 7

---

## Оптимизиран модел:

- ▶ Верига\_от\_магазини (име)
- ▶ Магазин (име на верига, номер, град, адрес)
- ▶ Екип (име на верига, номер на магазин, номер)
- ▶ Граждани (ЕГН, име)
- ▶ Служители (ЕГН, име, адрес, заплата, име на верига)
- ▶ Доставчици (ЕГН, име, компания)
- ▶ Доставя\_на (ЕГН на доставчик, име на верига)