

Бази от данни

Модел “Същност-връзки”

доц. д-р Димитър Димитров

Въведение

- Стъпки при проектирането на приложение, свързани с данните:
 - Събиране на изисквания
 - Концептуален модел на данните
 - Абстрактно представяне на високо ниво
 - Независим от СУБД
 - Логически модел на данните
 - Добавя повече детайли
 - Физически модел на данните
 - Зависим от конкретното СУБД, което ще се използва

Какво е E/R модел

- Модел „Същност-връзки“ (Entity-relationship model или E/R model)
- Може да се използва за абстрактно представяне на структурата на базата от данни
- Представя структурата графично в диаграма – E/R диаграма
- Идеи → E/R → релационна схема

Защо E/R? (1)

- Лесно разбираем и от разработчици, и от нетехнически заинтересовани лица
- Формален подход, недвусмислен
- По-високо ниво на абстракция
- По-малко технически детайли
- Изразителна нотация
- Позволява указването на ограничения

Защо E/R? (2)

- Помага за правилното проектиране на БД
- Помага за осигуряването на нормализация на данните

Основни елементи

1. Същности (Entities)
 2. Връзки (Relationships)
 3. Атрибути (Attributes)
- Според някои автори – множества от същности (Entity sets)

1. Същност

- Реален обект, концепция, събитие...
 - Може да се опише със съществително име
- Съществува физически (продукт, автомобил) или логически (продажба)
- Искаме да съхраняваме данни за него

Още понятия

- Тип същности (Entity type)
 - Категория, като клас в ООП
- Същност
 - Инстанция на типа същности
 - Често тип същности е синоним на същност – различни мнения по въпроса
- Множество от същности (Entity set)
 - Съвкупност от същности от един и същ тип

2. Атрибути

- Свойства на същностите
- Характеризират всяка същност от множеството от същности
- Всички същности в множеството имат едни и същи свойства
- Пример: същност “продукт”, атрибути – “наименование”, “цена”

Видове атрибути

- Обикновено са атомарни
 - Числа, символни низове (напр. телефонен номер)
- Може и композитни (напр. адрес)
 - Необходимо е да се прецени как ще бъдат представени физически в същинската схема на БД
- Може и с много стойности (напр. телефонни номера на човек)
- Може и производни (може да бъдат изчислени от други атрибути в базата)

Пример

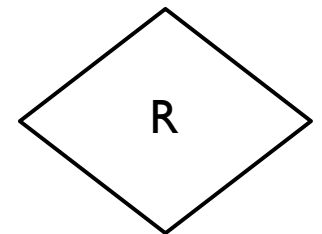
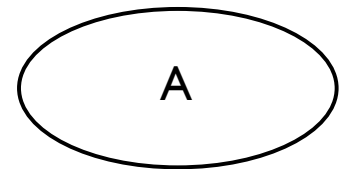
- Искаме да съхраняваме информация за филми и звезди, които играят в тях
- Същности:
 - Movie
 - Star
- Атрибути на филмовите звезди:
 - Name: “Harrison Ford” (simple, single-valued)
 - Address: “123 Main Str., LA, CA” (complex)
 - Birthdate: “1/1/50” (simple)
 - Age: 73 (може да се извлече от Birthdate и текущата година, напр. 2023)
- Типове същности:
 - Stars(Name,Address,Birthdate,Age)
 - Movies(Title,Year,Length,FilmType)

3. Връзка

- Асоциация, взаимоотношение между същности
- Може да се опише с глагол
- Пример: певец *изпълнява* песен
 - Същности: “певец”, “песен”
 - Връзка: “изпълнява”

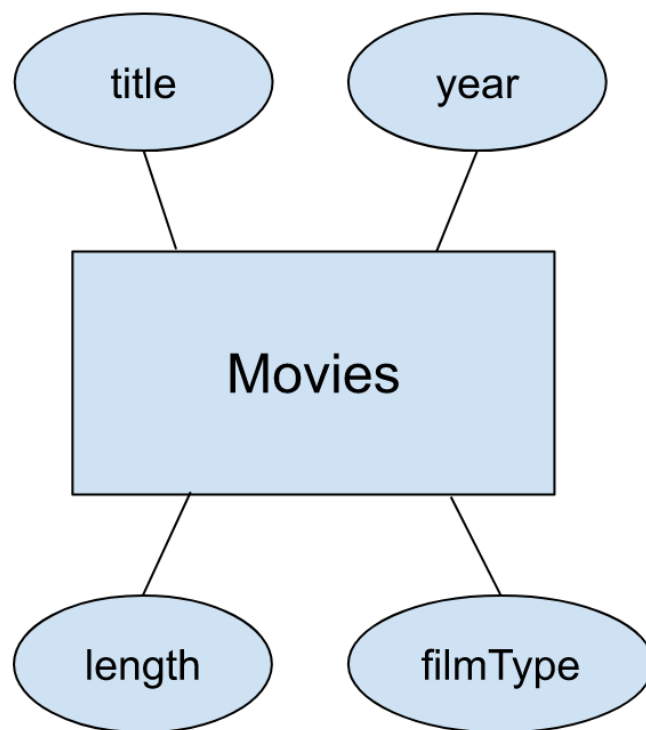
Е/R диаграма

- Графично представяне на множествата от същности, техните атрибути и връзките между тях
- Нотация на основните елементи:
 - Множество от същности – правоъгълник
 - Атрибут – овал
 - Множество от връзки - ромб



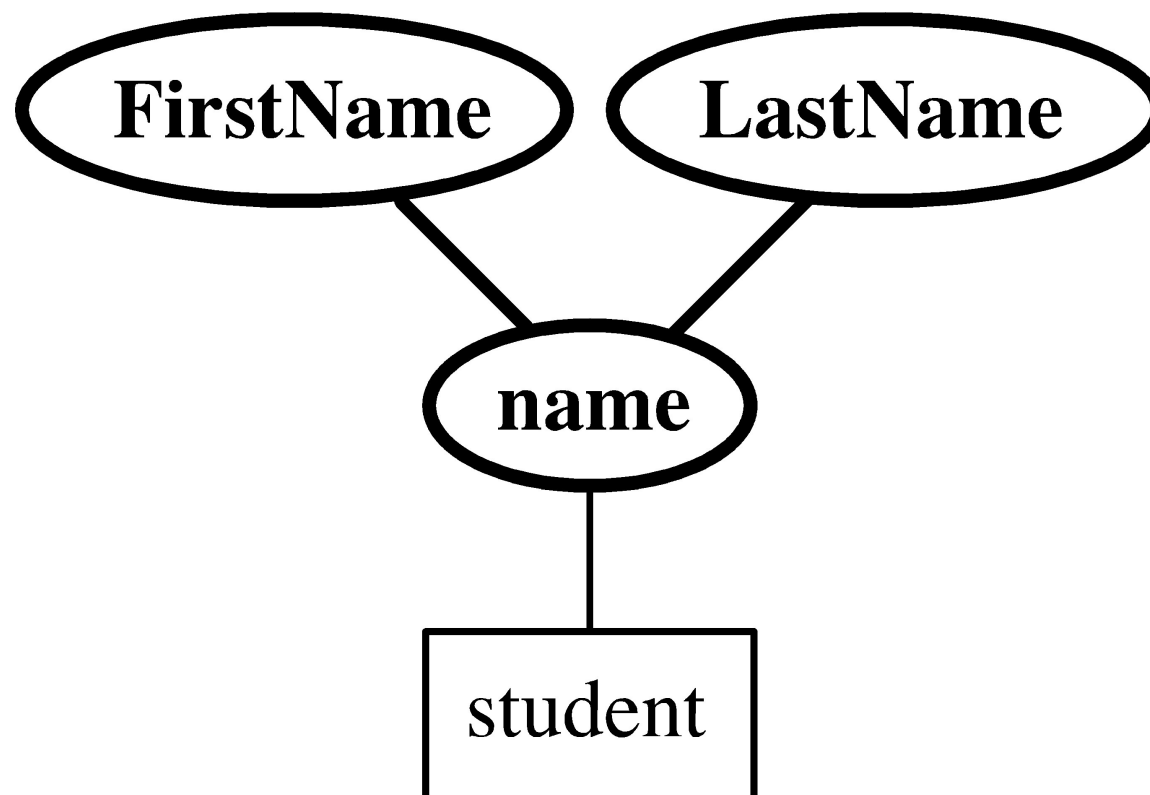
Пример

- Същност и атрибути



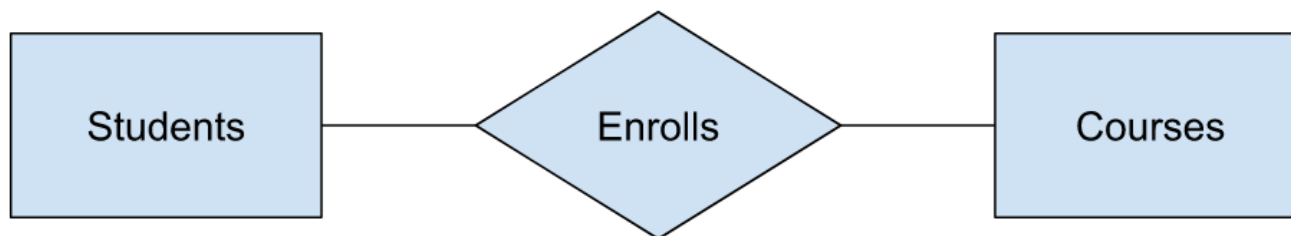
Пример

- Композитен атрибут



Връзка в E/R диаграма

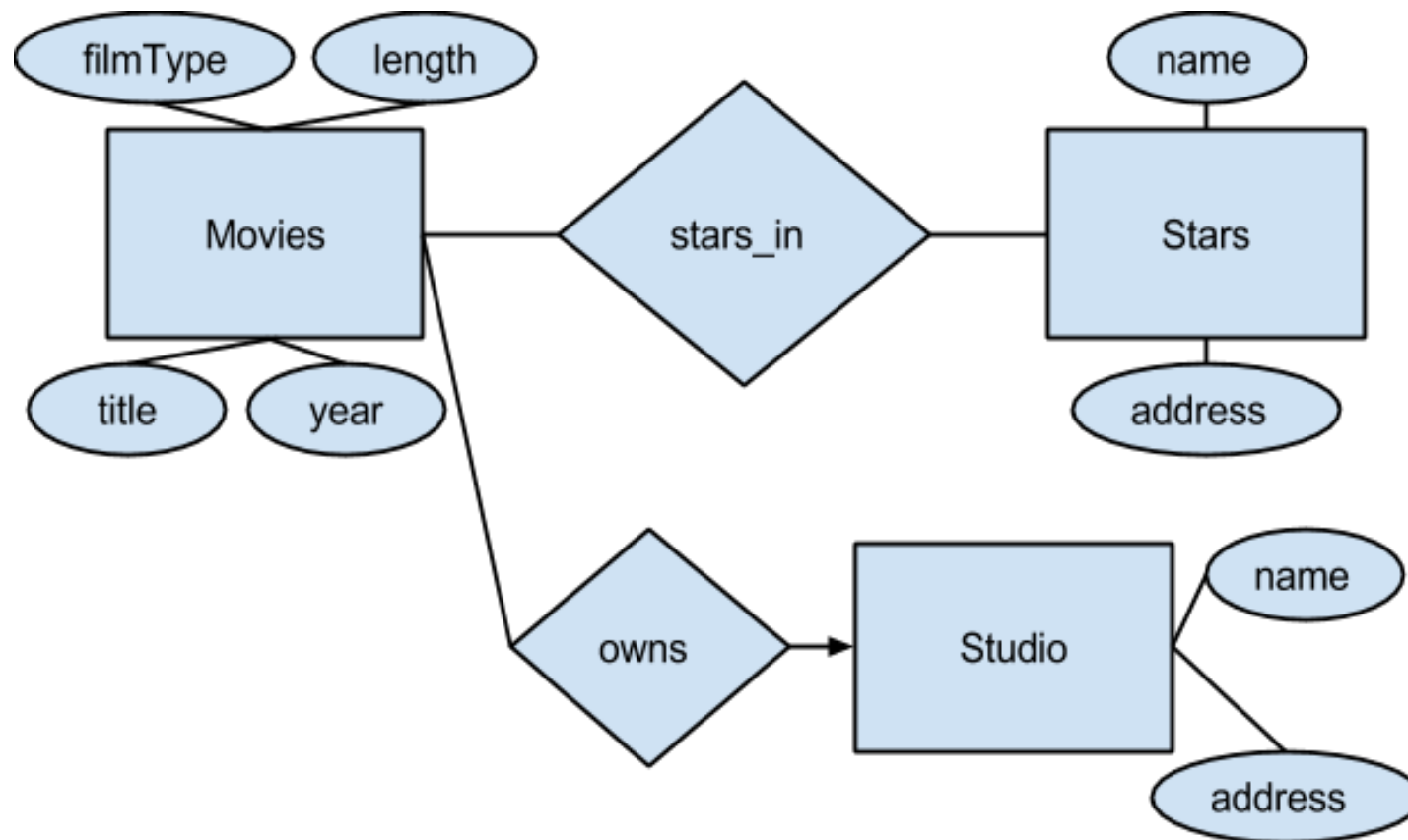
- Връзката свързва две или повече множества същности
 - Възможна е и връзка между същности от един тип
- Представя се чрез ромб и линии (и/или стрелки) към същностите от връзката



Пример: БД “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 от точно един продуцент
- Едно студио може да има много филми, но всеки филм е създаден от точно едно студио

Пример – E/R диаграма



- Следва да разберем по-подробно как се представят различните видове връзки

Екземпляр на E/R диаграма

- Конкретен краен набор от същности с конкретни значения за всеки атрибут
- Екземпляри на връзките, описани в диаграмите на БД

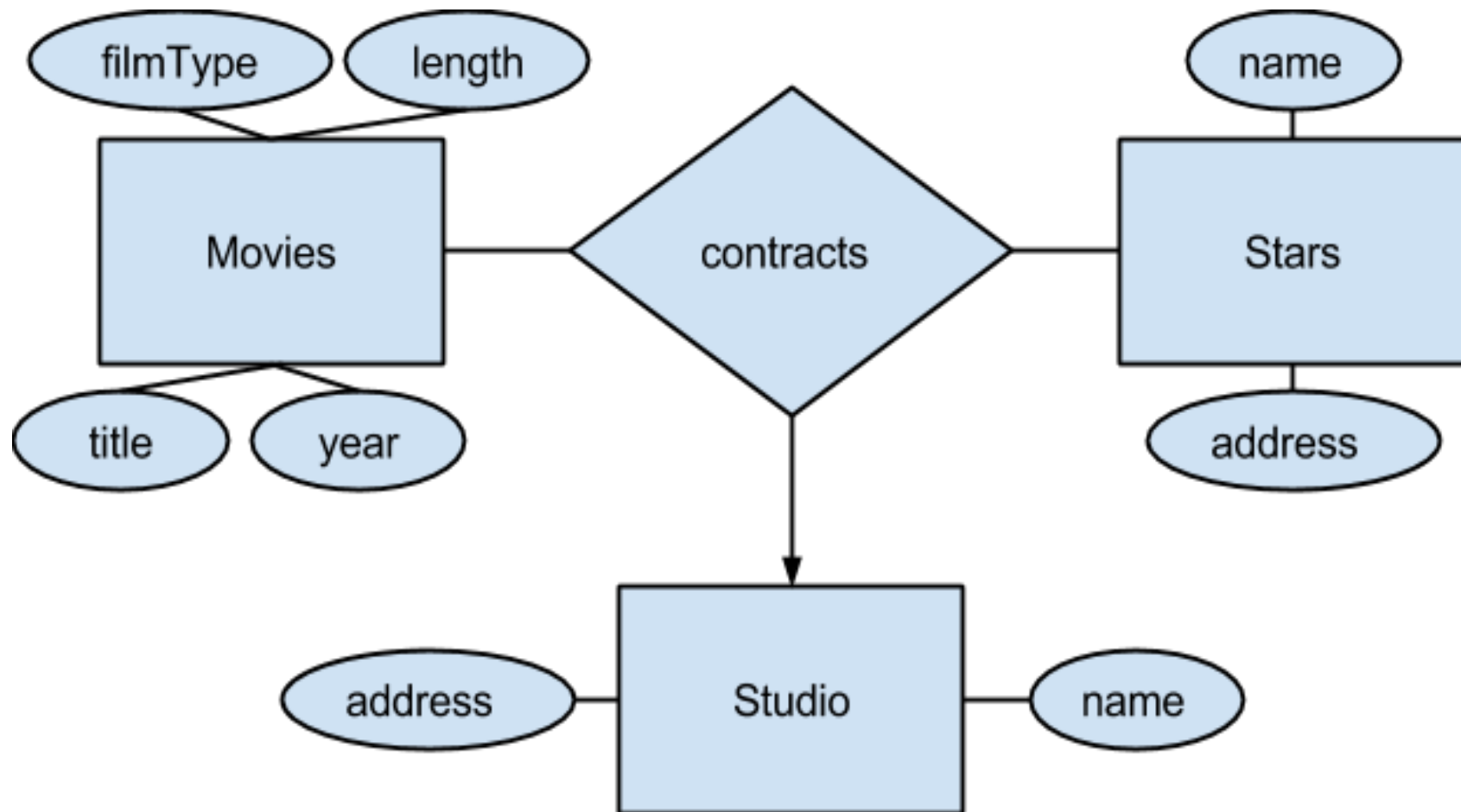
Свойства на връзките

- Име
- Множественост (Cardinality)
- Степен (Degree)
- Атрибути
- Роля
- Участие (Participation)

Степен на връзките

- Брой на участващите множества от същности
- Връзките в E/R модела могат да бъдат бинарни и N -арни
 - В частност, при $N=3$ - тернарни
- Бинарните връзки свързват две множества от същности, а N -арните – N множества от същности

Пример за небинарна връзка

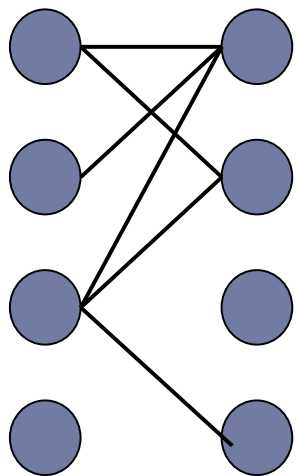


Множественост

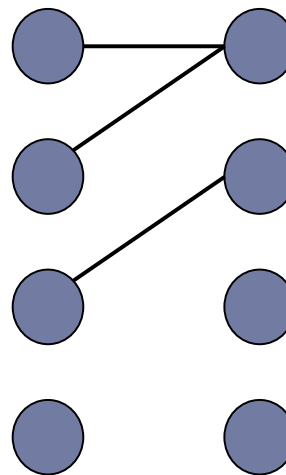
- Определя максималния брой на същностите, с които една същност може да участва във връзка
- Отговаря на въпроси като “В колко филма може да играе даден актьор?”
- Ще разгледаме множественост на бинарни връзки

Видове бинарни връзки

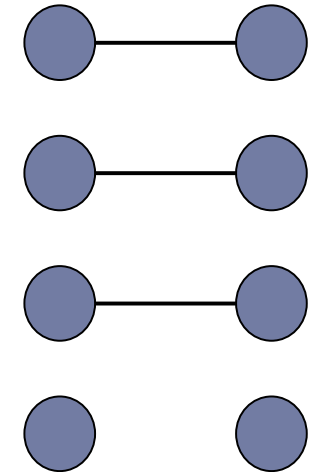
- В зависимост от съответствието на същности от едно множество със същности от друго множество, имаме:



Много към много
Many-to-many
M:N



Много към едно
Many-to-one
M:1

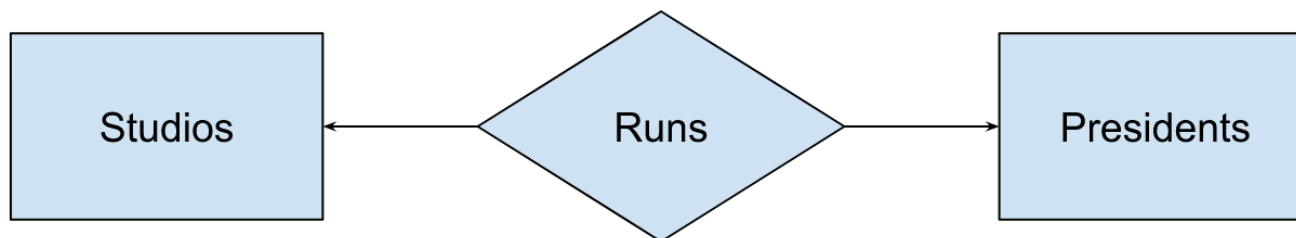


Едно към едно
One-to-one
1:1

Казано с думи...

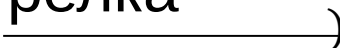
- Нека R е връзка, която свързва множествата от същности E и F
- R е *много към едно* от E към F , ако на една същност от F съответства 0, 1 или повече същности от E
- *Едно към много* – аналогично
- Примери

- R е *едно към едно*, ако на една същност от едното множество от същности съответстват 0 или 1 същности от другото множество
 - R е едновременно много към едно от E към F и много към едно от F към E
 - Примери:
 - Държави, градове, столици



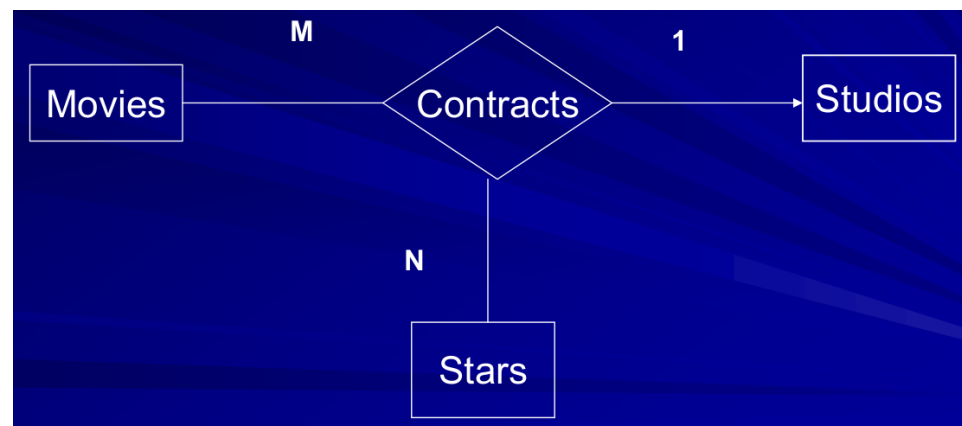
- *R* е *много към много*, ако същност от всяко едно от двете множества от същности може да се свързва с много същности от другото множество
 - Т.е. не се налага ограничение в броя
 - Пример

Представяне в диаграма

- Много към едно – стрелка към “едно”-страната
- Едно към едно – стрелки и към двете множества от същности
- Много към много – линии без стрелки
- Ако същност участва точно веднъж (а не 0-1 пъти), може да се използва заоблена стрелка 

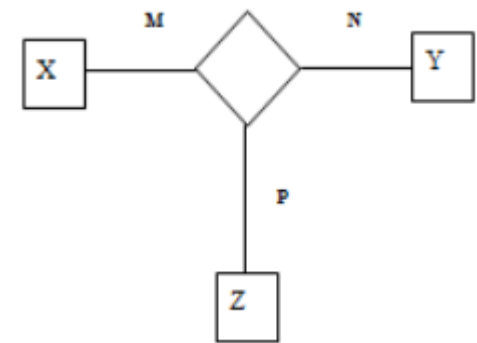
Или:

- Означения (1, M, N) върху линиите



Множественост при тернарните връзки

- 1:1:1
- 1:1:M
- 1:M:N
- M:N:P
- Тълкуване на примера: за всяка двойка (X,Y) съществуват P същности от Z

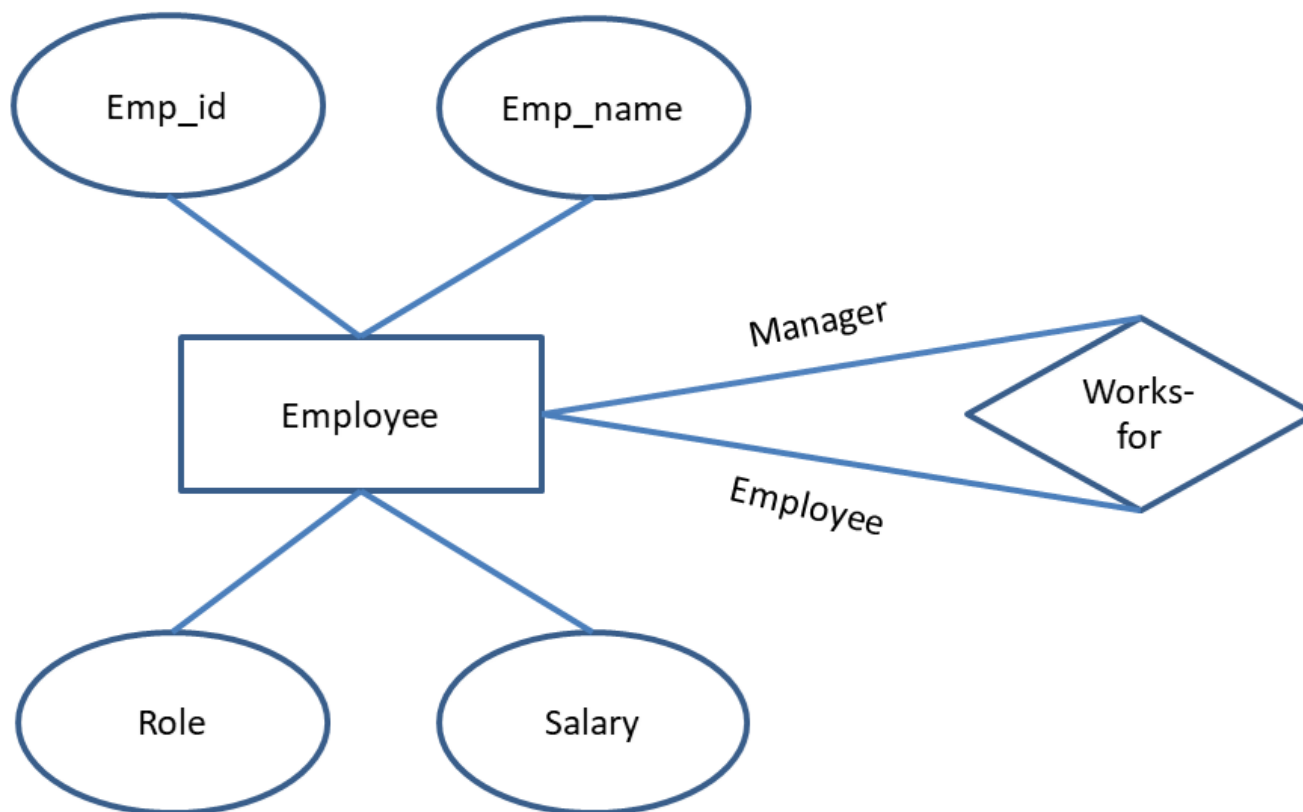


Роли

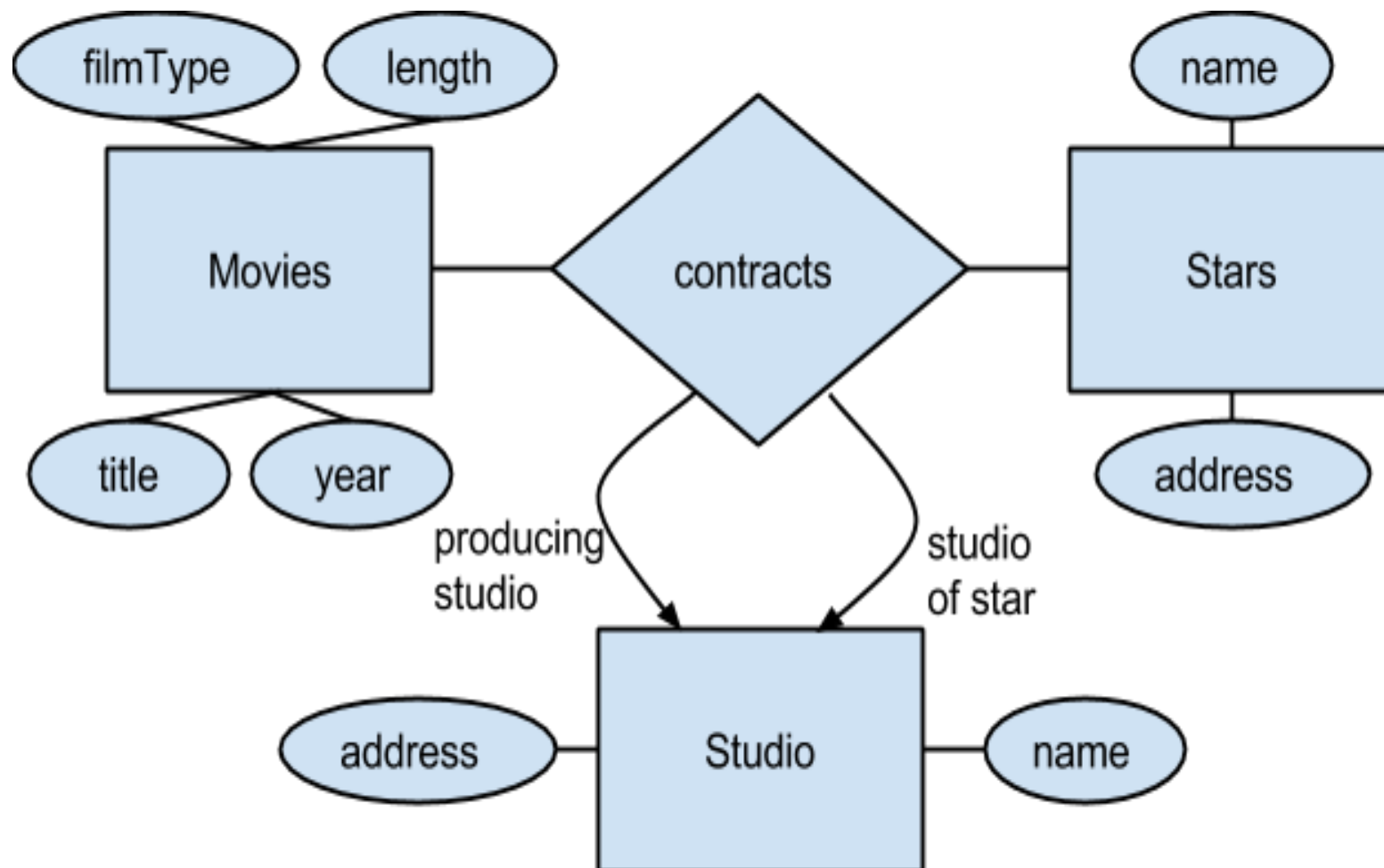
- Всяко множество от същности, което участва във връзка, изпълнява роля в тази връзка
- Едно множество от същности може да се използва два или повече пъти в една връзка, но в различен смисъл
 - Тогава за всяка роля се поставя отделна дъга от връзката към множеството от същности

Роли – пример №1

- За да се различават отделните роли, дъгите се именуват



Роли – пример №2



Атрибути на връзка (1)

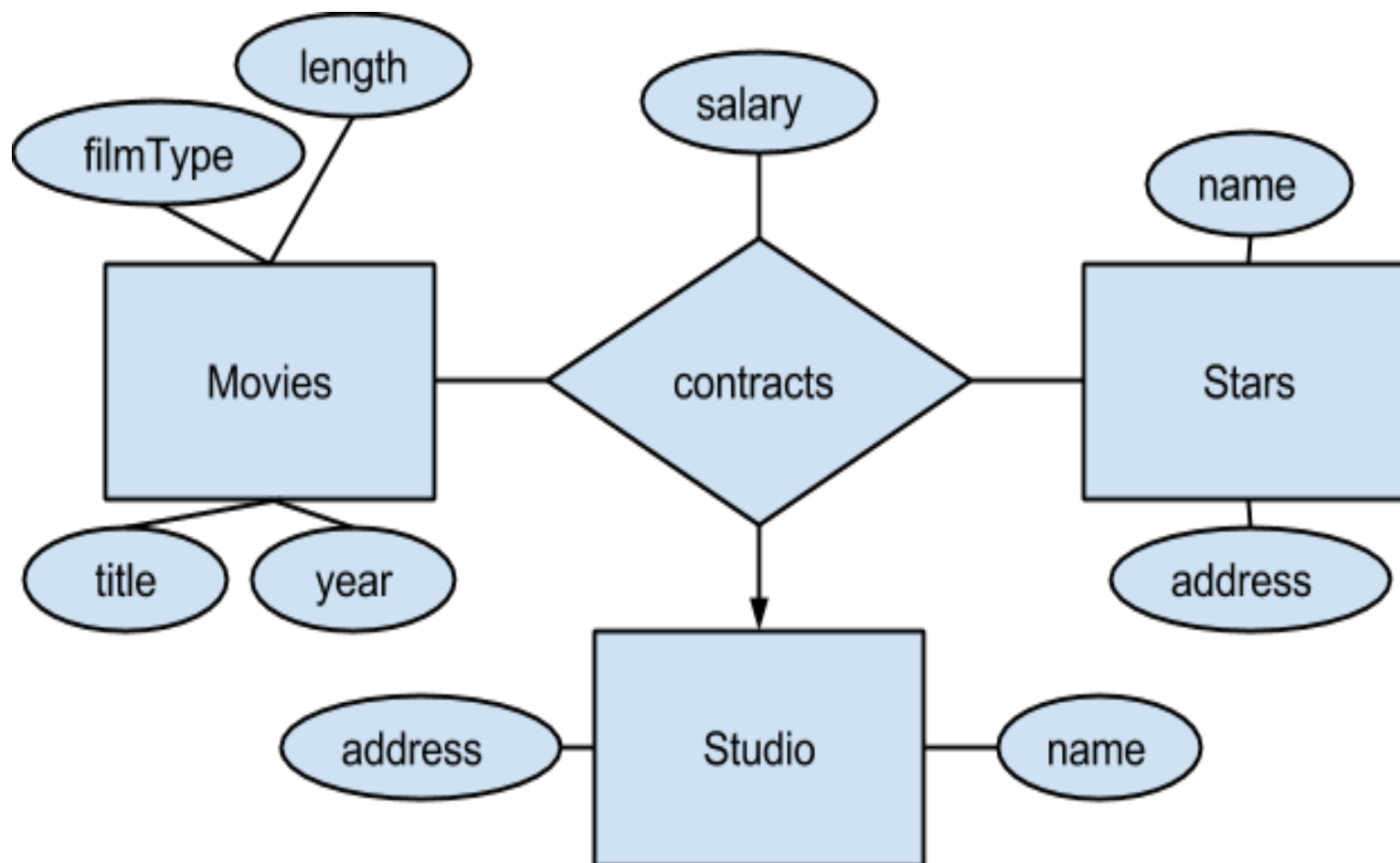
- Атрибутът може да бъде не само характеристика на множество от същности, но и на връзка
- Характеристика, която е специфична за конкретната връзка между същности, а не за самите същности

Атрибути на връзка (2)

- Типични случаи:
 - Исторически данни – напр. Employees, Projects, връзка WorksOn с атрибут startDate
 - Количествена информация – напр. количество закупени продукти в една поръчка
 - Качествена информация – напр. статус на проект в Manages връзка между мениджъри и проекти
- Пример: връзката Contracts може да има атрибут salary
 - salary не може да бъде атрибут на Studio, защото едно студио произвежда много филми
 - не може да бъде характеристика на Stars, защото един актьор играе в много филми

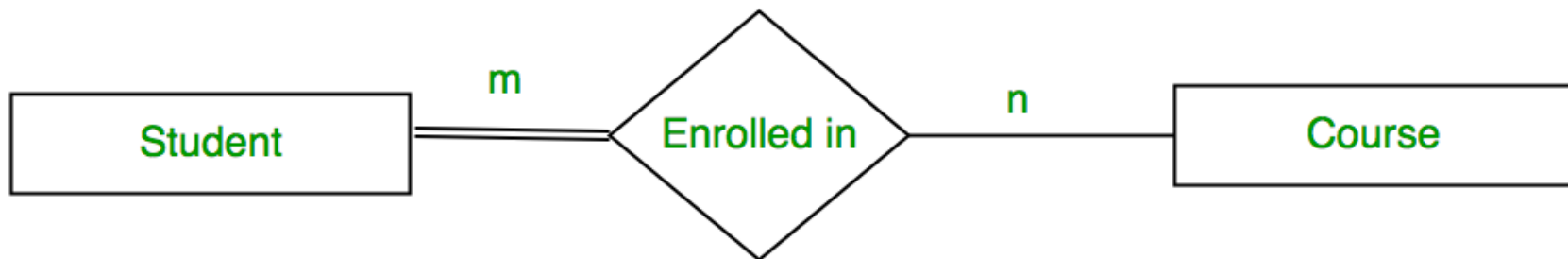
Атрибути на връзка (3)

- Представяне в Е/Р диаграма



Участие (Participation)

- Указва дали участието на същност в дадена връзка е задължително
- Може да се укаже поотделно за различните краища на връзката



Ограничения

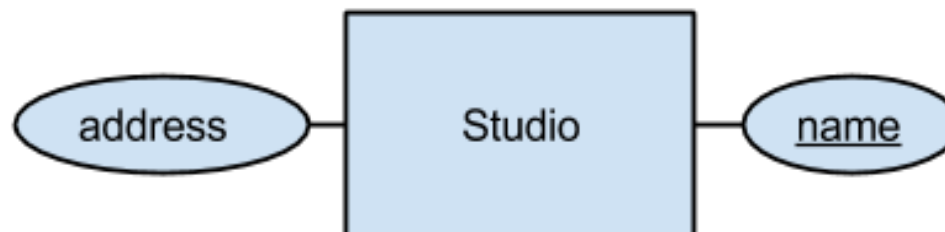
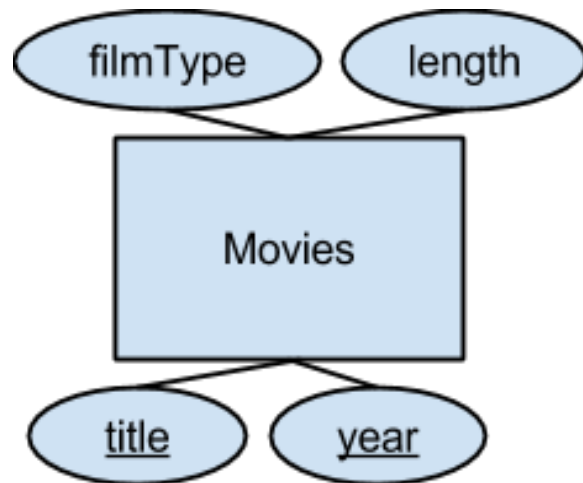
- Правила и условия, които осигуряват, че данните са структурирани според определени критерии
- Дотук разгледахме:
 - Множественост
 - Участие
- Ще видим още:
 - Ключове
 - Референтна цялостност
- Ще споменем и ограничения, за които няма нотация в E/R
- Ограниченията са част от схемата на релационните БД
 - Веднъж декларирани, СУБД няма да позволи модификация на данните, която би ги нарушила

Ключ (1)

- Ключ на множество от същности наричаме атрибут или списък от атрибути, които уникално определят всяка същност от множеството
- Ако имаме повече от един атрибут:
 - Никои две същности от множеството не съвпадат по всички атрибути на ключа
 - Две същности могат да съвпадат по част от атрибутите на ключа, но не и по всички
- Ключът е важен, защото може да идентифицира дадена същност
- Всяко множество от същности трябва да има ключ
 - Освен слабите, за които ще стане дума по-късно
- Едно множество от същности може да има повече от един възможен ключ (кандидат-ключ), но само един от атрибутите се избира за първичен ключ

Ключ (2)

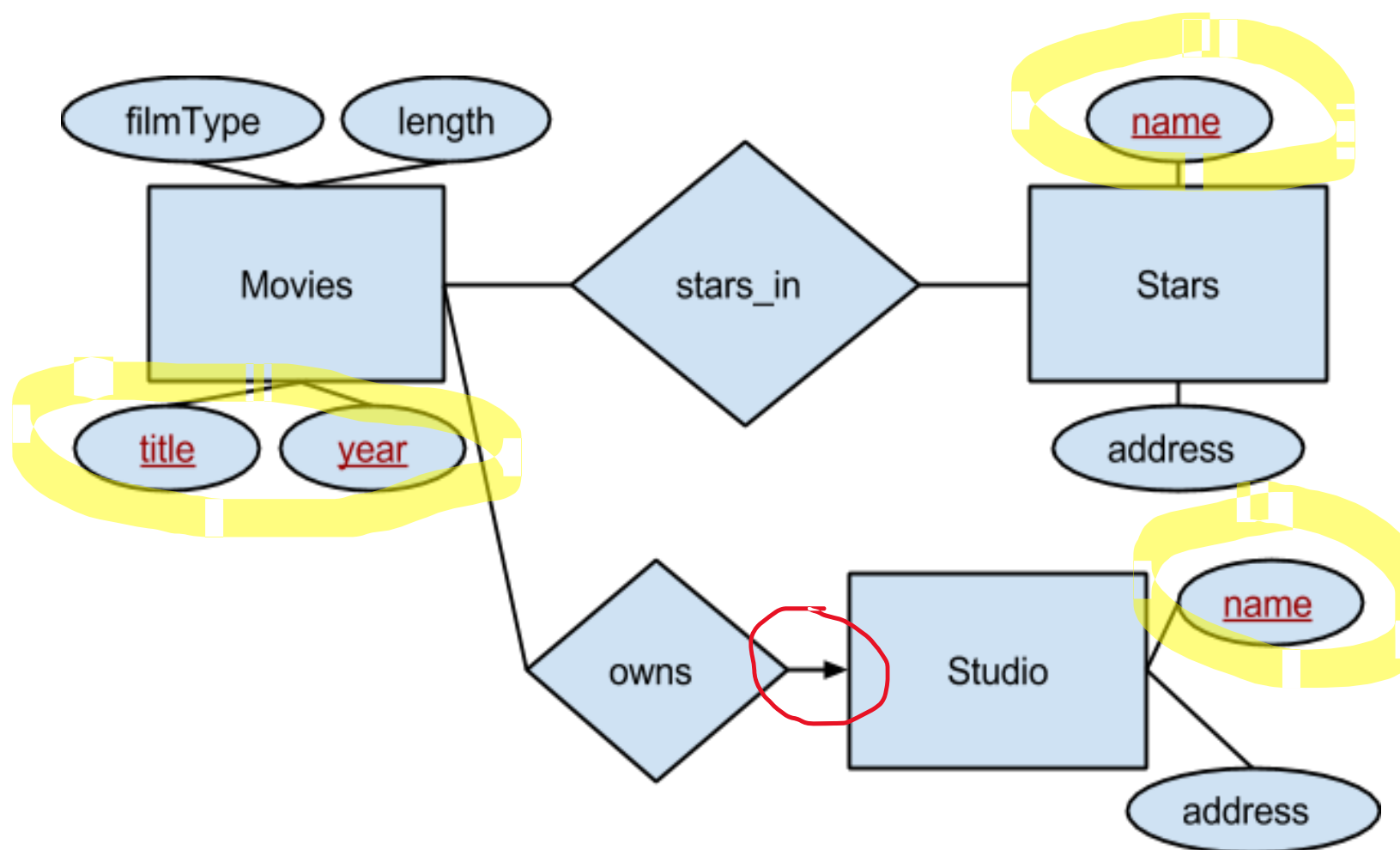
- Представяне в Е/Р диаграма – чрез подчертаване на всички атрибути, които го съставят
- Няма нотация за изразяване на кандидат-ключовете



Ограничения по единствена стойност

- Гарантирант 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 същността, към която сочи връзката в случай, че съществува
- Стойността за даден атрибут може да липсва – тогава като стойност се записва null
- В E/R диаграмата няма нотация за null, но може да се укаже по неформален начин

Пример



Референтна цялостност (Referential integrity)

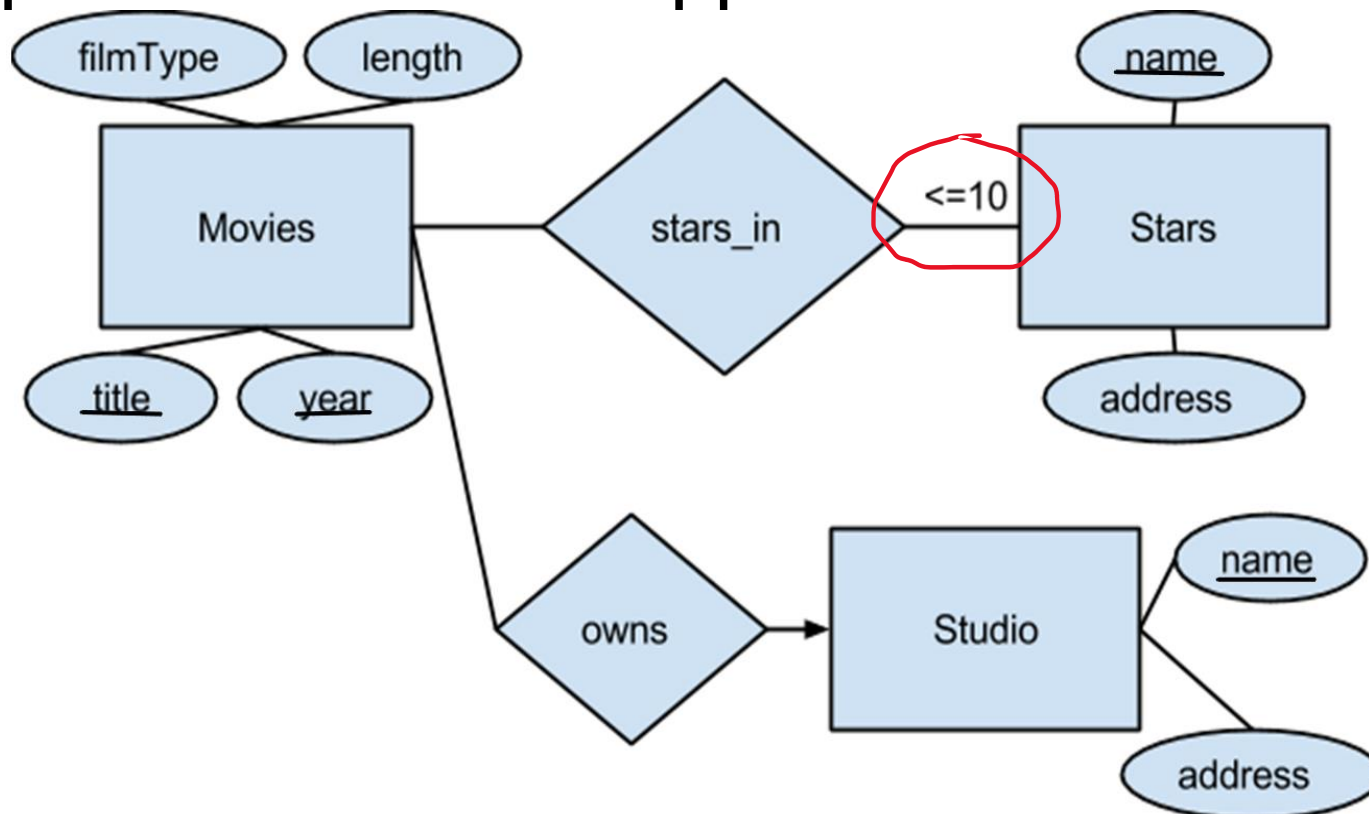
- Обикновено се използва при връзките много към един, за да покаже съществуване и единственост на същност, към която връзката реферира
- Гарантира съществуването на същностите, към които се реферира
- Няма специална нотация в диаграмите, но обикновено следва от използването на връзки, ключове и др.
- Важно ограничение, за което ще стане дума и в следващи лекции

Други ограничения

- Ограничения по домейн
 - Ограничават стойностите на атрибут да бъдат в дадена област (да принадлежат на дадено множество)
 - Например: цели числа, низове, дати...
 - Няма специална нотация в E/R диаграмите

Множественност – още ВЪЗМОЖНОСТИ

- Брой звезди в един филм не надвишава 10
- Над линията може да напишем 0..10 или:



Задача 1

Като използвате нотацията на E/R модела, определете множествата от същности и връзките според следното описание:

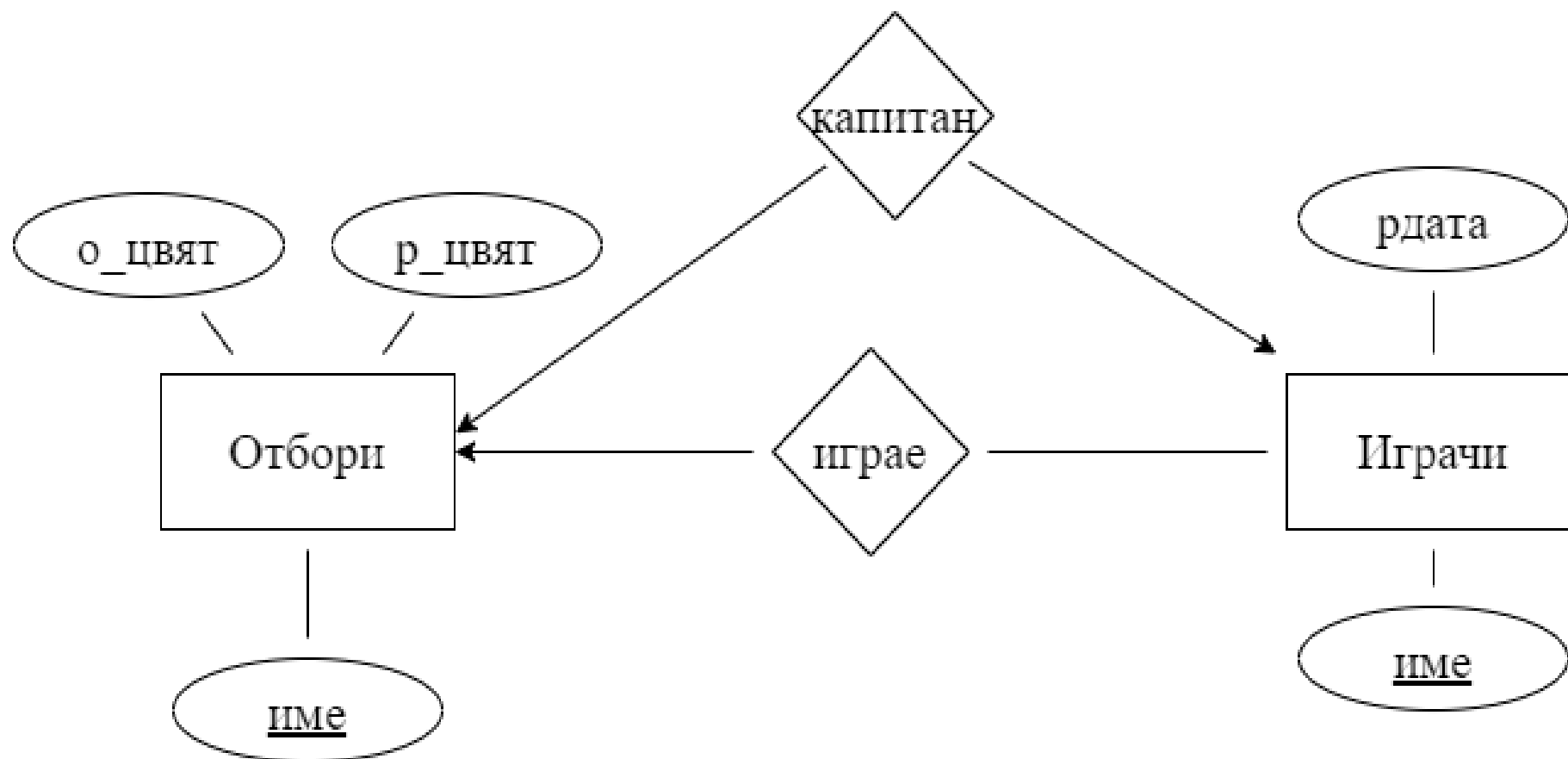
База от данни съхранява информация за отбори – те се характеризират с име на отбор, основен цвят на екип и резервен цвят на екип. Също така в базата се съхранява информация за играчи – те се характеризират с име на играч и рождена дата. В сила са следните ограничения:

- 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
- Името на отбора е уникално
- Името на играча е уникално

Софтуер за диаграми

- Огромно разнообразие от инструменти, например:
- draw.io - <https://app.diagrams.net/>

Решение на задача 1



Задача 2

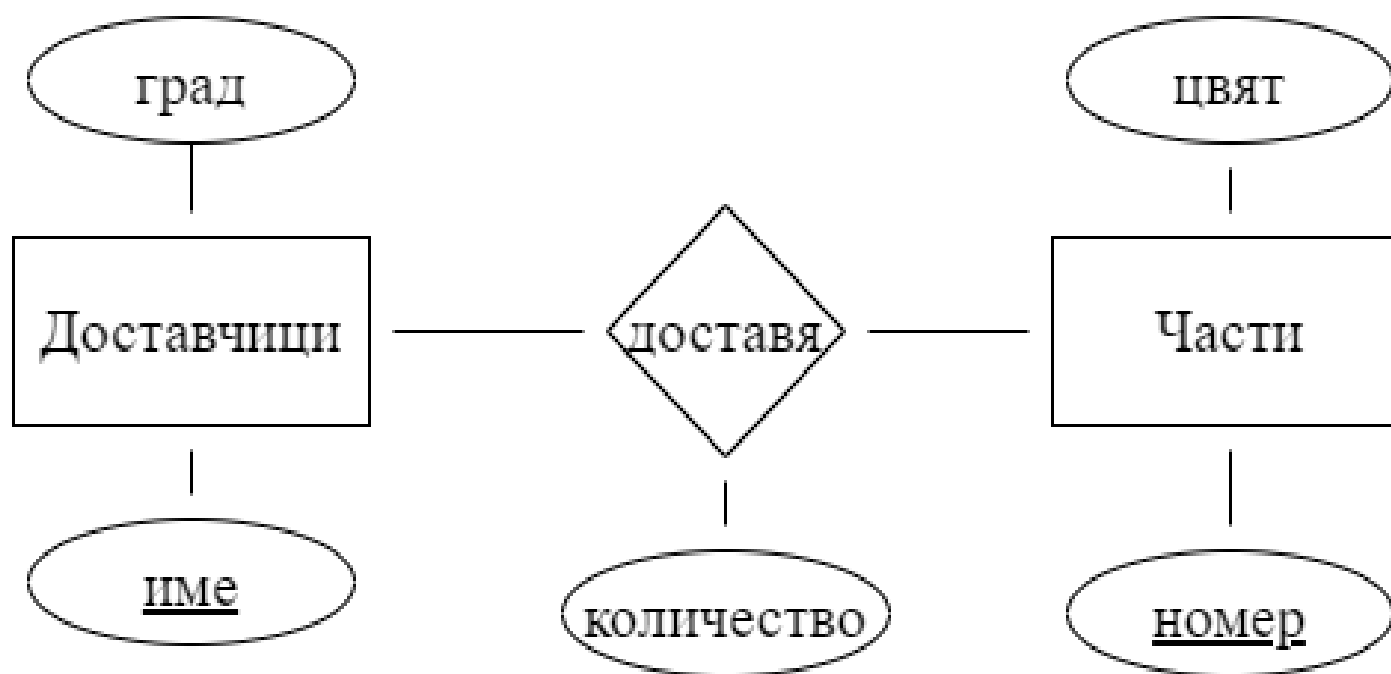
Като използвате нотацията на E/R модела, направете дизайн на база от данни, съдържаща информация за:

- 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

Решение на задача 2



Задача 3

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

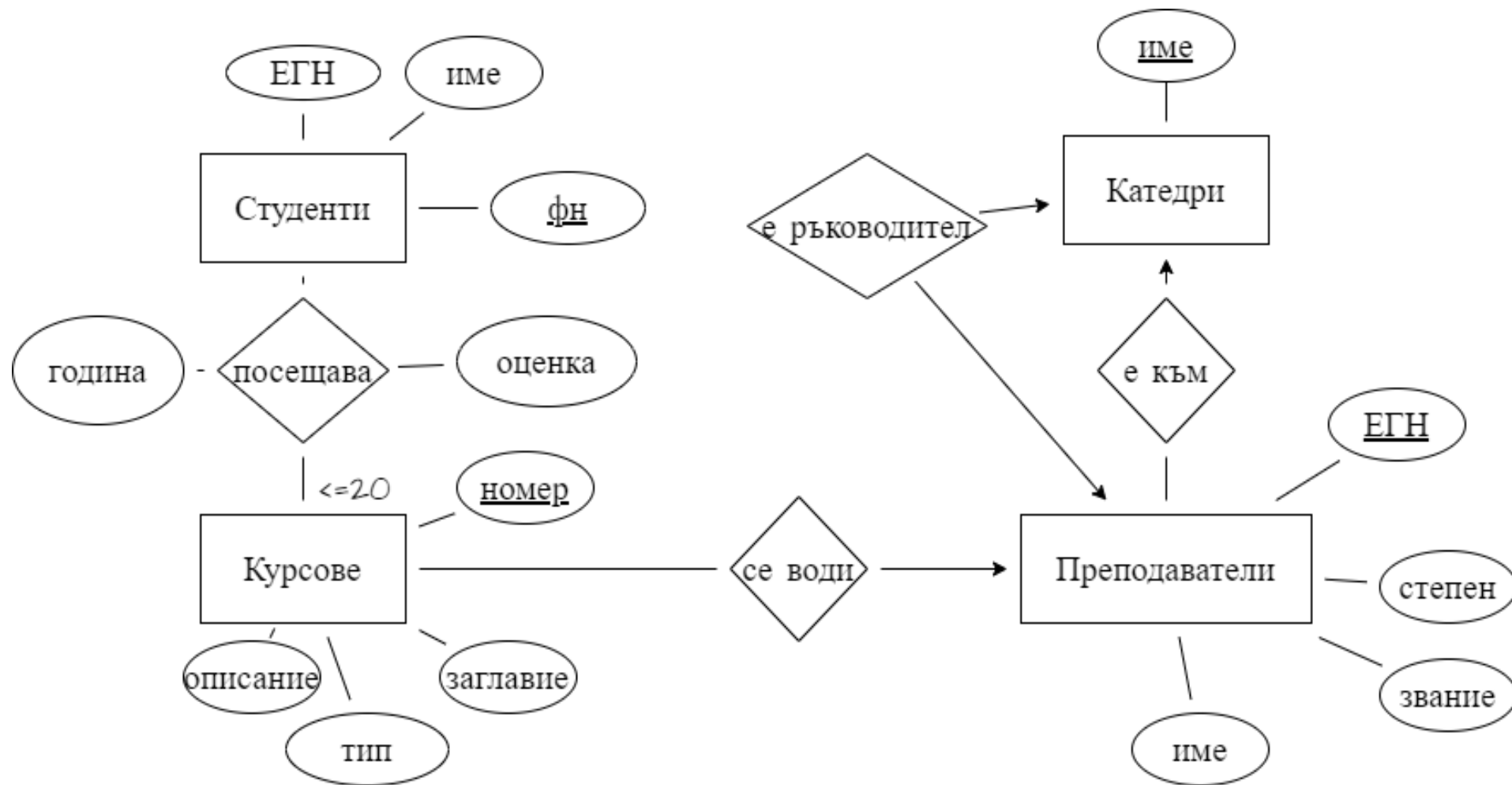
- Студенти: ЕГН, име, факултетен номер
- Катедри: име, завеждащ катедра
- Преподаватели: ЕГН, име, научно звание, научна степен
- Курсове: номер на курс, заглавие на курса, описание, тип

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
- Всеки преподавател работи в точно една катедра
- Всяка катедра се ръководи от точно един преподавател
- Един преподавател не може да ръководи повече от една катедра

При посещаването на даден курс през дадена година студентът получава оценка

Решение на задача 3



Задача 4

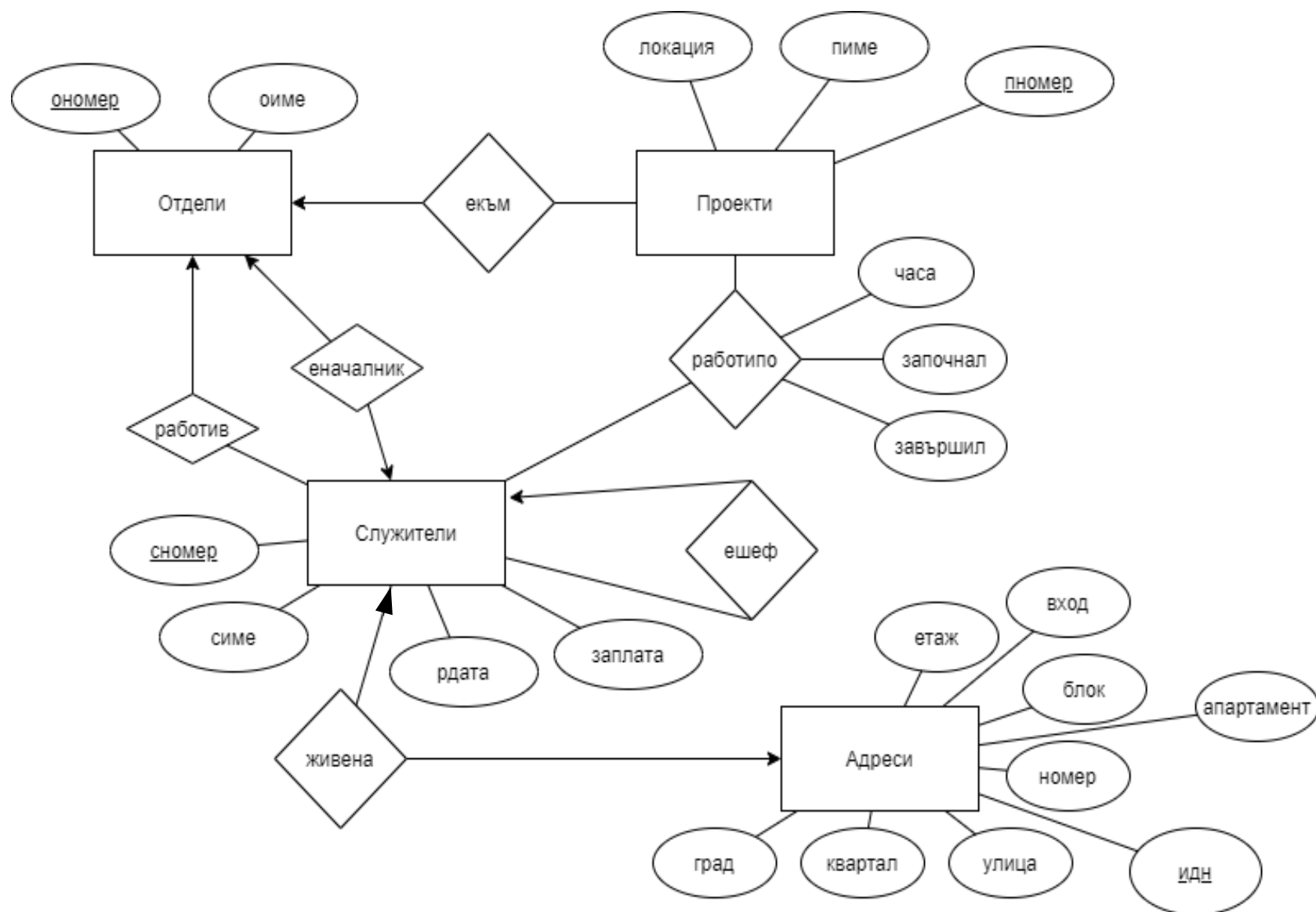
Направете дизайн на база от данни, съдържаща информация за:

- Отдели: номер на отдел, име на отдел
- Проекти: номер на проект, име, локация
- Служители: номер на служител, име, рождена дата, текуща заплата, адрес (град, ..., апартамент)

Да се вземат предвид следните ограничения:

- Всеки служител работи към един отдел, а в един отдел могат да работят много служители
- Може да има и служители, които не работят в нито един отдел
- Всеки служител може да работи по много проекти, а по един проект могат да работят много служители. За всяка двойка служител-проект в базата от данни съхраняваме начална и крайна дата, както и изработени часове
- Всеки служител има шеф, който също е служител
- Всеки отдел има ръководител на отдела, който също е служител
- Един отдел може да има точно един ръководител. Не може да има отдел без ръководител. Един служител може да бъде ръководител на точно на един отдел
- Всеки проект принадлежи на един отдел, а в един отдел може да се работи по много проекти

Решение на задача 4



Задача 4 – дискусия

- Как да представим адресите?
 - Композитен атрибут на Служители
 - + Опростено
 - Отделна таблица – позволява бъдещи разширения:
 - + Служител да има много адреси
 - + История на адресите
 - + Много служители на един адрес
 - Повече таблици

Подкласове

- Друг вид връзка: **is-a**
- Чрез нея могат да се изразяват йерархии от класове на множества от същности
- Понякога едно множество от същности съдържа същности със специфични характеристики, които са в сила само за конкретните същности
- В такива случай се дефинират подкласове от тези същности, които се свързват с базовото множество от същности чрез is-a 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 (суперкласът) съдържа общите атрибути за всички същности от йерархията

Подкласове - пример

- Суперклас: Movies(title, year, length, ...)
- Подкласове: Cartoons, Mystery, Science-Fiction
 - Не всички филми са такива, но някои са
- В допълнение към всички характеристики (атрибути и връзки) на movies, murder-mysteries имат и атрибут weapon

Подкласове – антипример (1)

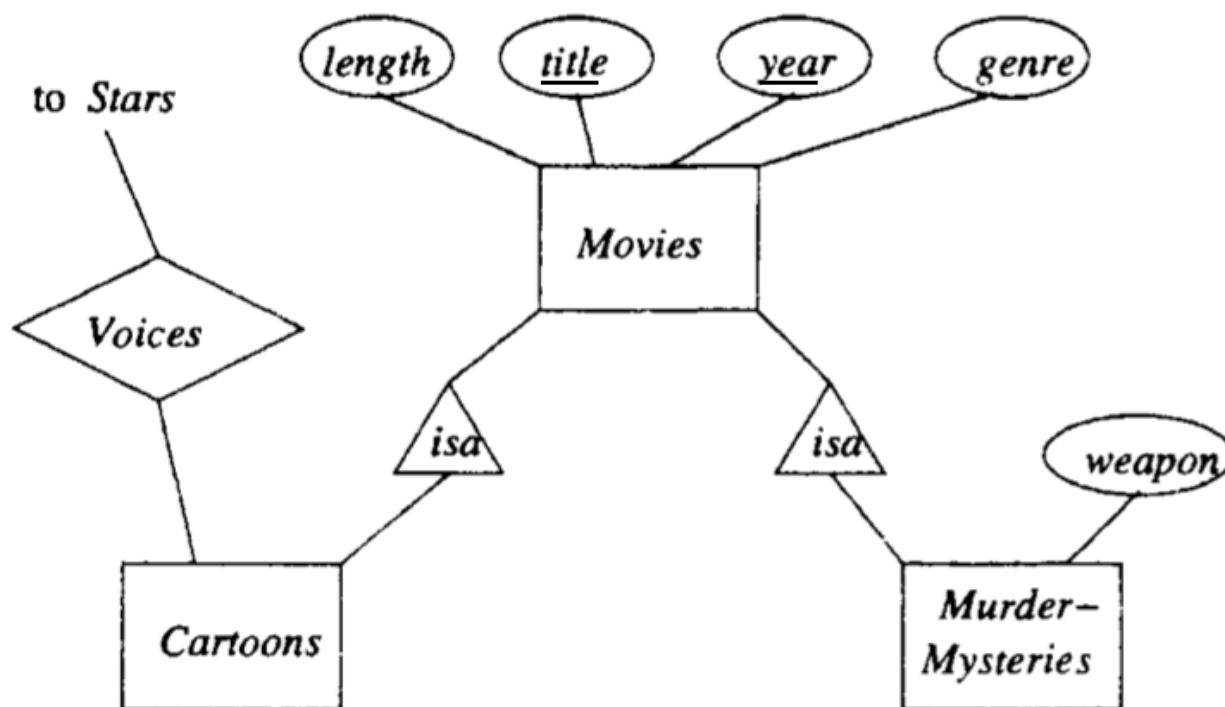
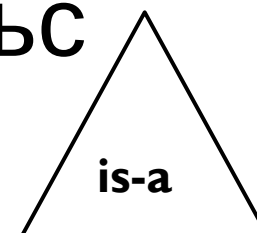
- Точка в равнината има координати x и y
- Окръжност има координати (x, y) на центъра и радиус
- Имаме общи атрибути
- Можем ли да използваме is-a?

Подкласове – антипример (2)

- Не, не трябва да използваме is-a
- Как да разберем, че не може?
- Задаваме си въпроса “точката окръжност ли е?”
 - Не е, следователно не може

Представяне в диаграма

- Връзката is-a свързва подклас със суперклас
- Триъгълник, сочещ суперкласа



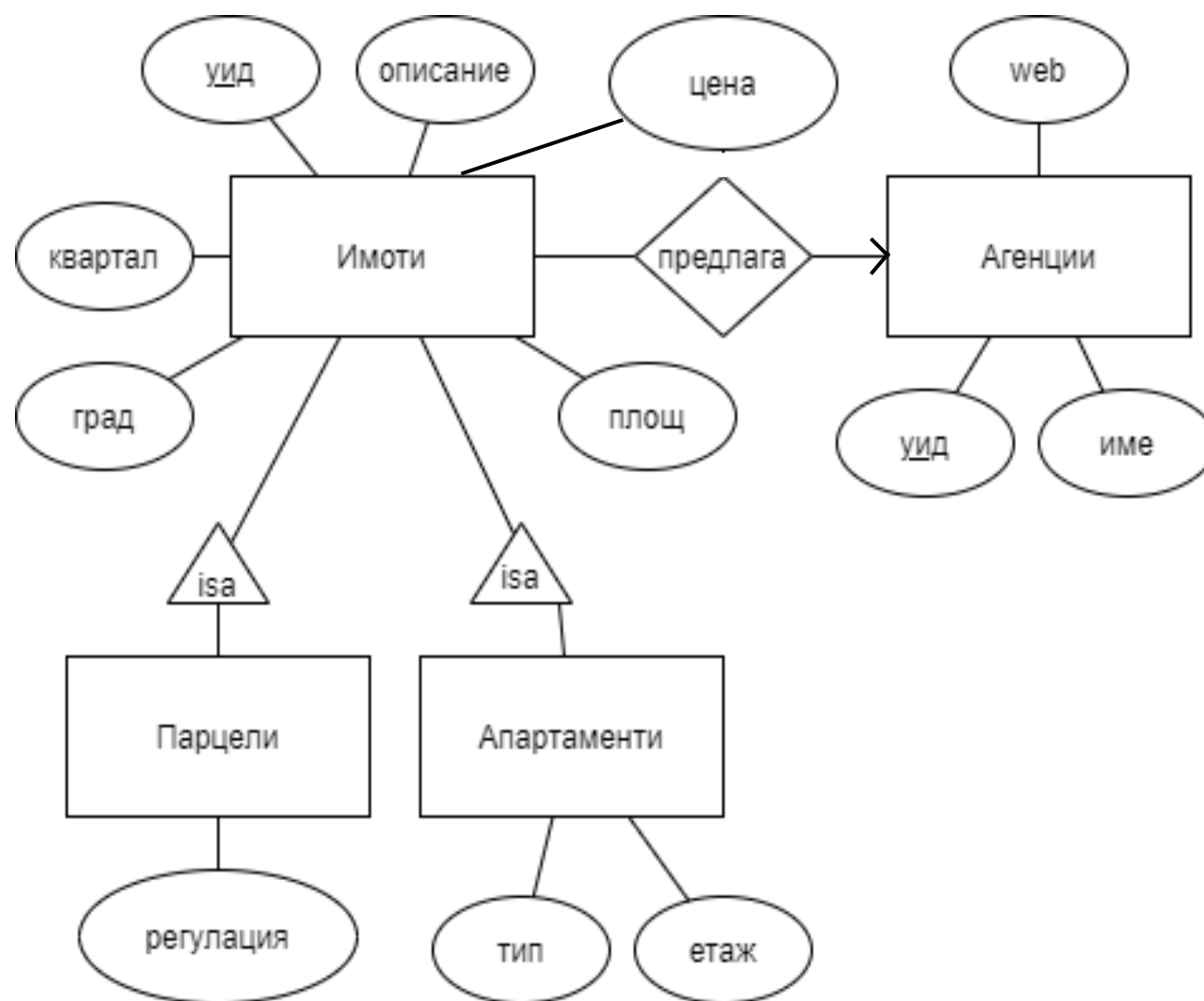
Подкласове и ключове

- Само кореновото множество от същности има ключ
- Този ключ се използва като ключ от всички множества от същности в йерархията

Задача 5

- Информационна система съхранява информация за обяви за продажби на недвижими имоти и за агенциите, които ги предлагат. Поддържат се два вида обяви - за апартамент и за парцел. За всяка обява задължително се съхранява следната информация: уникален идентификатор, описание, град, квартал, цена, площ в кв. м. За някои обяви е посочен и допълнителен ориентир (напр. “бул. Джеймс Баучър”, “НДК” и т.н.). Във всяка обява за апартамент допълнително са указани номер на етаж и брой стаи, а във всяка обява за парцел - дали е в регулация. Всяка обява се предлага от агенция. За всяка агенция задължително се съхранява уникален идентификатор, име и уеб сайт. Една агенция може да предлага много обяви. Един и същ имот може да се предлага от няколко агенции, но това се водят две различни обяви, в които всеки елемент (описание, цена и т.н.) може да са различни. Създайте E/R модел на БД, която съхранява гореописаната информация.

Решение на задача 5



Задача 6

- Информационна система за градски транспорт, подобен на този в град София, съхранява информация за закупените от гражданите електронни карти за пътуване. Предлагат се два вида карти — карти за определен брой пътувания и карти за определени линии (например карта, която важи едновременно за метро №2, за трамвай №2 и за трамвай №9). Картите за определени линии имат и период на валидност (начална и крайна дата). Линиите на градския транспорт имат номер и тип (метро, трамвай, тролейбус). Всяка карта има цена в левове. Електронните карти се издават на пластични носители (така наричаме пластиката, която гражданите носят със себе си и показват на контрольорите). Във всеки носител може да бъдат заредени няколко електронни карти, напр. в един и същ носител може да има карта за трамваи №1 и №5, валидна от 1.7.2024 г. до 31.7.2024 г., и карта за 10 пътувания. Всеки носител има уникален идентификатор. Някои носители имат и име на собственика, а другите са анонимни. Добавете сурогатен ключ там, където е необходимо. Да се създаде E/R модел на БД, която съхранява гореописаната информация. Да се начертае E/R диаграма на модела.

Слабо множество от същности (1)

- Не може да бъде уникално идентифицирано, ако използваме само неговите атрибути
 - Атрибутите му не са достатъчни за формиране на ключ
- Разчита на друго, силно множество от същности, с което е свързано
- За формиране на ключа му включваме ключовите атрибути на силното (идентифициращо/родителско) множество
 - Заедно с 0 или повече собствени атрибути на слабото множество
- Същност от слабото множество трябва да има поддържаща връзка с точно една същност от силното
 - Възможно е слабото множество да участва и в други връзки, които не са поддържащи

Слаби множества от същности (2)

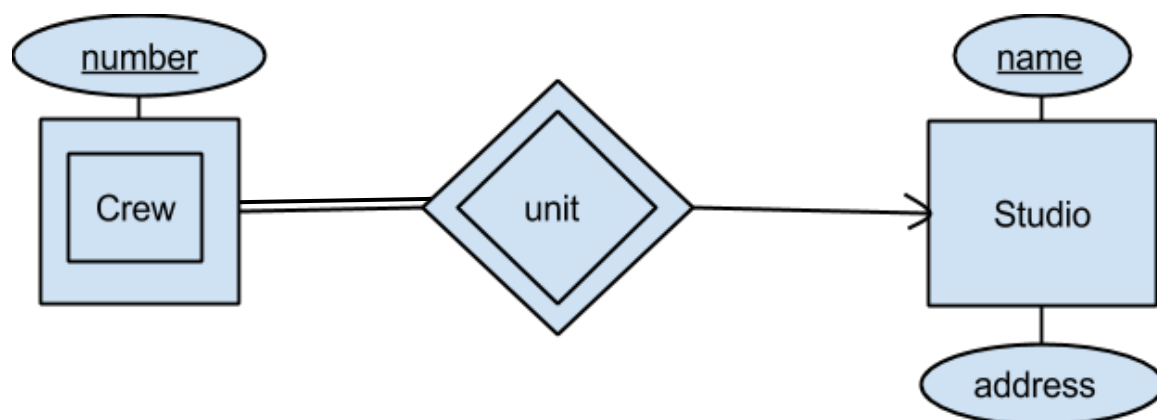
- Причини за възникване
 - Йерархии, различни от is-a
 - Въвеждане на свързващи множества от същности при елиминиране на многостепенни връзки
 - Новото множество е слабо и е свързано с връзки много към един с идентифициращите множества
- В реалните проекти често липсата на ключ се преодолява чрез генериране на уникални идентификатори
 - сурогатен ключ, ще го коментираме по-късно

Представяне в диаграма

Слабо множество от същности	Правоъгълник с двойна рамка
Поддържащи връзки	Ромбове с двойна рамка
Атрибут от слабото множество, който е част от ключа	Овал с подчертан текст

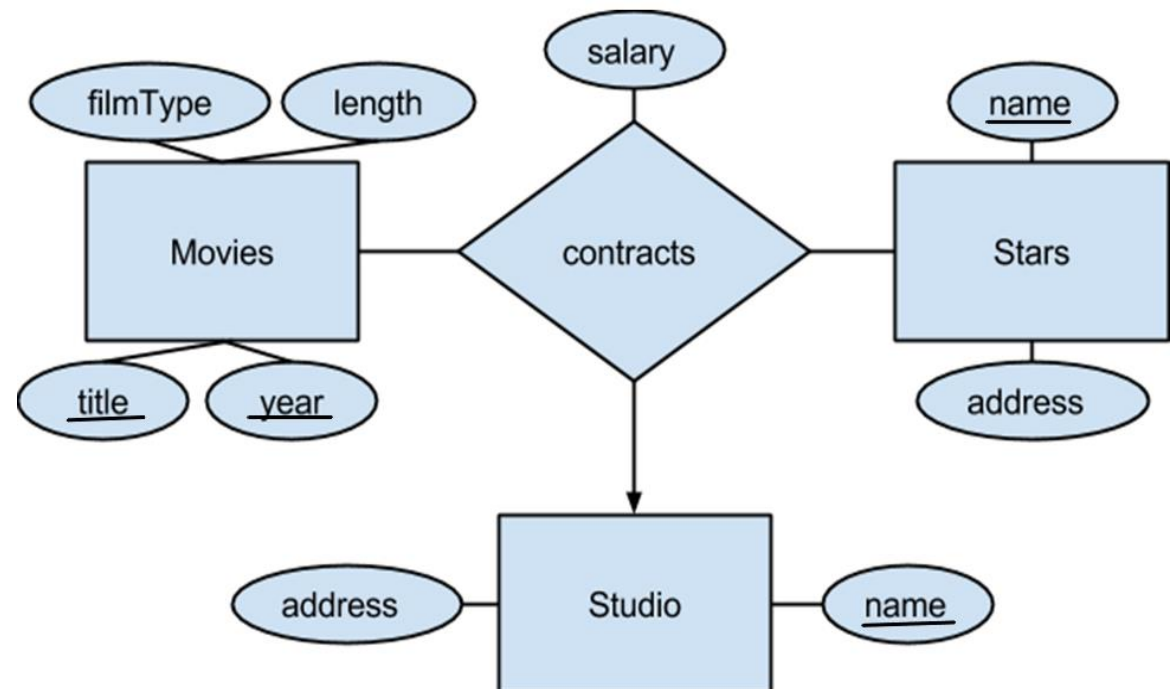
Пример

- Едно студио за филми има няколко филмови екипа
- Екипите могат да бъдат обозначавани към съответното студио като Екип 1, Екип 2 и т.н.
- Друго студио може да има същите екипи - Екип 1, Екип 2 и т.н.
- Номер на екипа не може да бъде ключ за множеството от същности “Екипи”
- Необходимо е и името на студиото

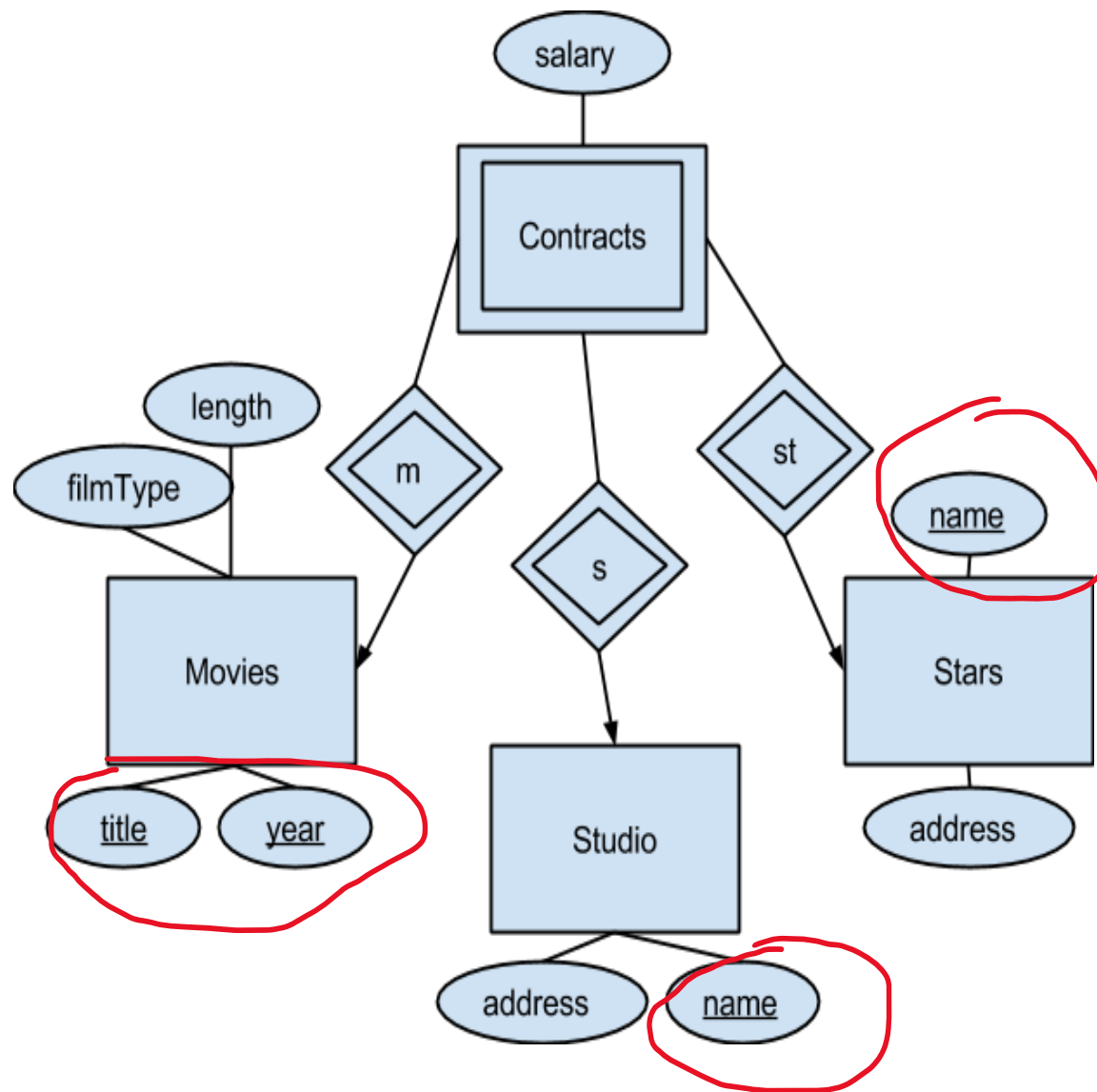


Преобразуване на небинарна връзка (1)

- Нека разгледаме тернарната връзка Contracts
- Едно студио може да сключи договор за един или няколко филма, в които могат да играят един или няколко актьори, с които също е сключен договор



Преобразуване на небинарна връзка (2)




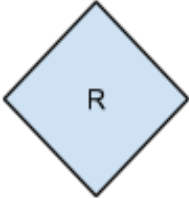

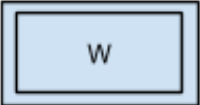
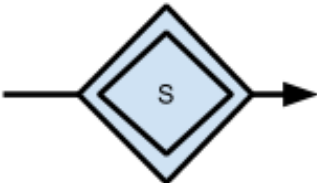

Задача 7







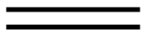




- Искаме да проектираме БД на информационна система за студентски карти за отстъпки
- Всяка карта за отстъпки има уникален номер и дата на валидност
- Всеки студент има факултетен номер, който е уникален в рамките на неговия университет, както и име и рождена дата
- Всеки университет има уникално име и град
- Ако един и същ човек е студент в няколко университета, в картата се посочва само един от тях, а за другите не се указва никаква информация
- Един студент може да има много карти за отстъпки

Задача 7 – насоки за решение

- Множеството от същности “Студенти” е слабо, тъй като може да има двама студенти с еднакъв ф.н. от различни университети
- Поддържащото множество е “Университети”
- Връзката между “Студенти” и “Университети” е M:1

Нотация на Е/Р модел – непълно обобщение

	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.

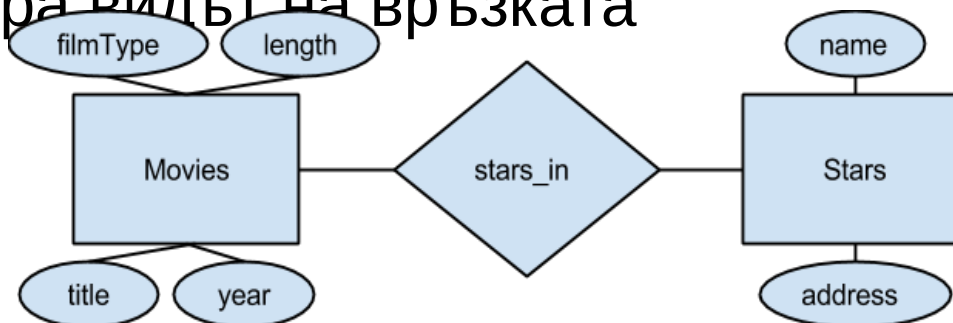
	Represents Entity
	Represents Attribute
	Represents Relationship
	Links Attribute(s) to entity set(s) or Entity set(s) to Relationship set(s)
	Represents Multivalued Attributes
	Represents Derived Attributes
	Represents Total Participation of Entity
	Represents Weak Entity
	Represents Weak Relationships
	Represents Composite Attributes
	Represents Key Attributes / Single Valued Attributes

Основни принципи на проектирането на БД

- Съответствие/достоверност
- Избягване на излишество
- Простота
- Избор на правилния вид елементи

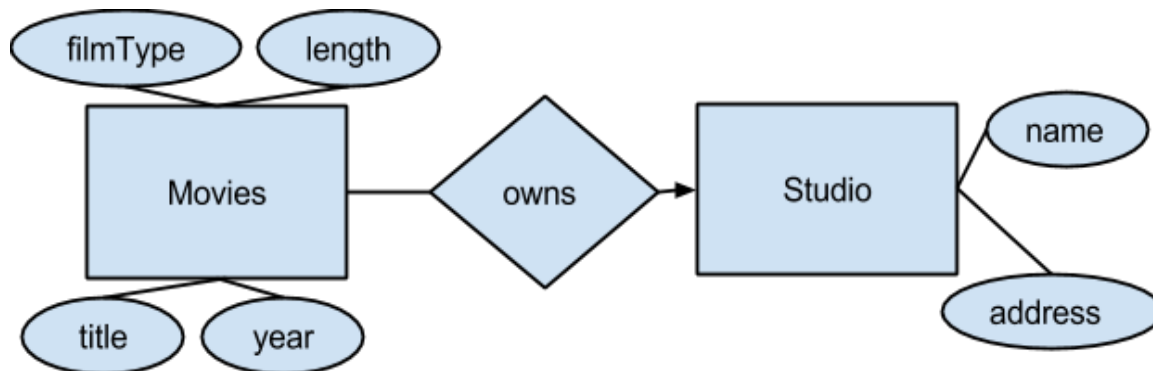
Съответствие/ Достоверност

- 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 към 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 видът на връзката. Тогава е необходима допълнителна информация, за да се специфицира видът на връзката



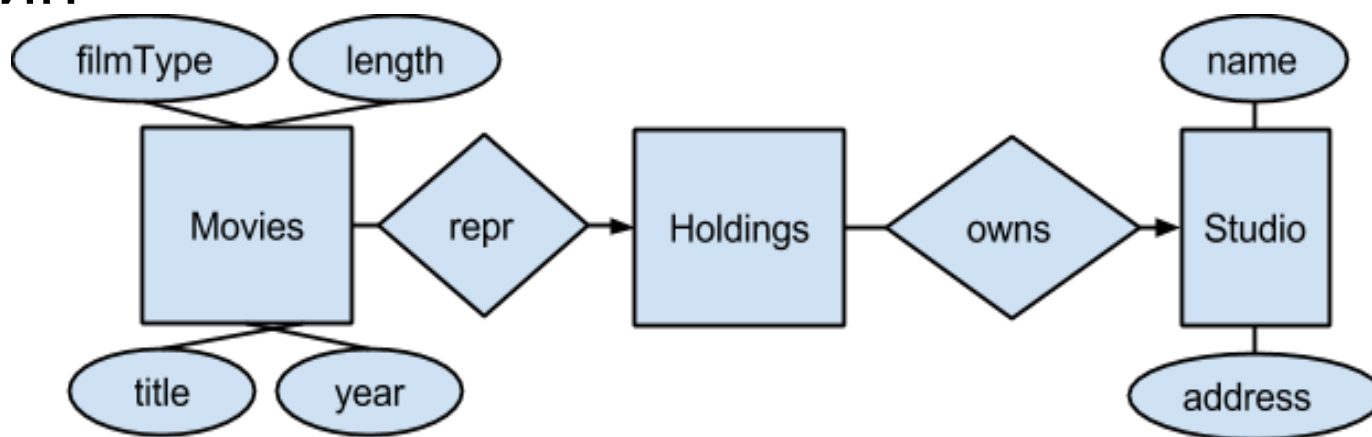
Избягване на излишество

- Една и съща информация, представена по различни начини
- Пример за излишество: атрибут studioName към Movies
 - Може да доведе до нарушение на съгласуваността (consistency):
 - Ако за един филм сменим студиото (връзката owns), но забравим да променим studioName
 - Допълнителен разход на памет



Простота

- Да се избягва употребата на повече от необходимите елементи
- Пример: ако добавим ново множество от същности Holdings
- В случая диаграмата е коректна и отразява вярно действителността, но същността Holdings е излишна
- Трябва да се целим към изчистен и опростен дизайн

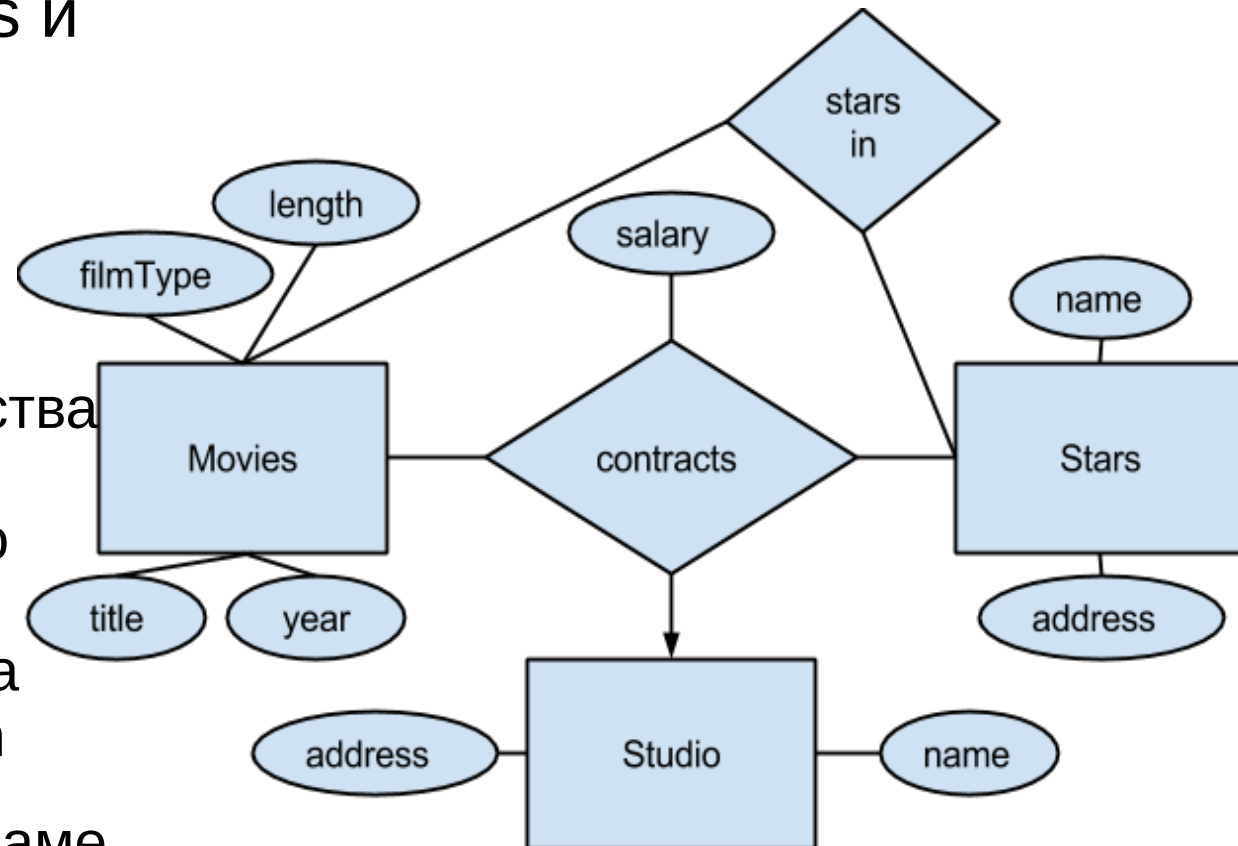


Избор на правилните връзки

- Не трябва да добавяме връзки, които следват от други връзки
- Би довело до същите проблеми като при излишеството

Пример

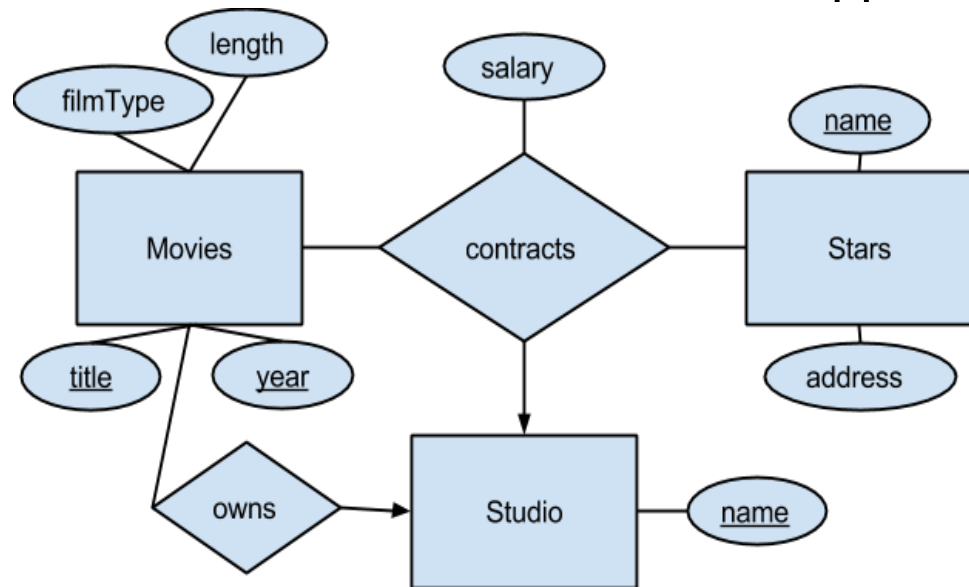
- Дали са необходими и двете връзки – contracts и stars in?
- Зависи от нашите предположения
 - Ако актьор може да участва във филм, само ако има договор със съответното студио за участие в съответния филм, тогава нямаме нужда от stars in
 - Ако обаче може или нямаме информация за договора, тогава имаме нужда



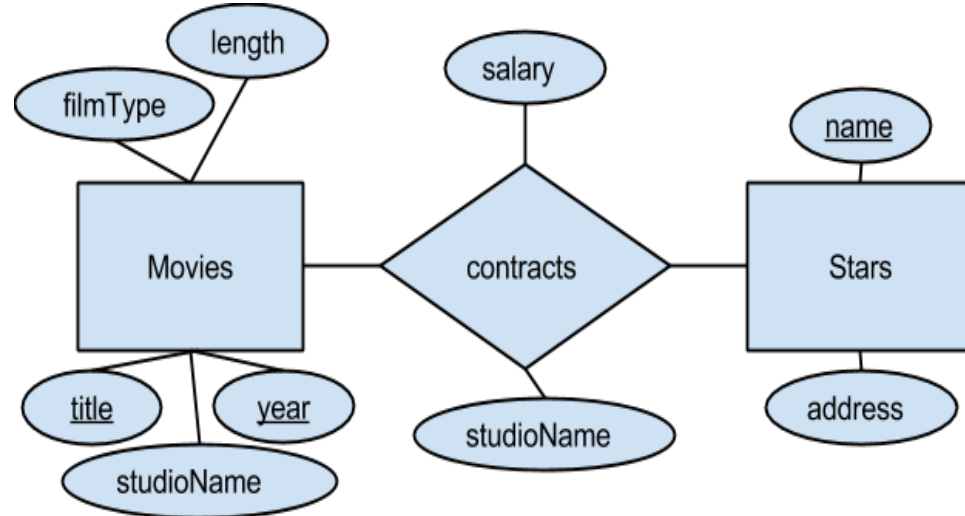
Избор на правилния вид елементи

- Понякога имаме различни възможности при моделиране на БД и представянето на реалния свят
- Връзка, атрибут или слабо множество?
- Например едно множество от същности може да бъде моделирано и като атрибут на друго множество от същности

- Множество от същности Studio или атрибути studioName
- Имаме само ключови атрибути
- Имаме само връзки много към 1, сочещи Studio, като няма множество с повече от една връзка към Studio



- + разделение на отговорностите
- + разширяемост в бъдеще
- entity set трябва да има и неключови атр.



- + простота
- + всеки entity set има и неключови атр.

Въпроси?