Бази от данни

Релационен модел

доц. д-р Димитър Димитров

Въведение

- Дотук разгледахме E/R модел
- Други модели:
  - Мрежови
  - Йерархичен
  - …
  - Релационен
- Ще разгледаме релационния модел
  - И преобразуване от E/R към релационен модел

Релационен модел (1)

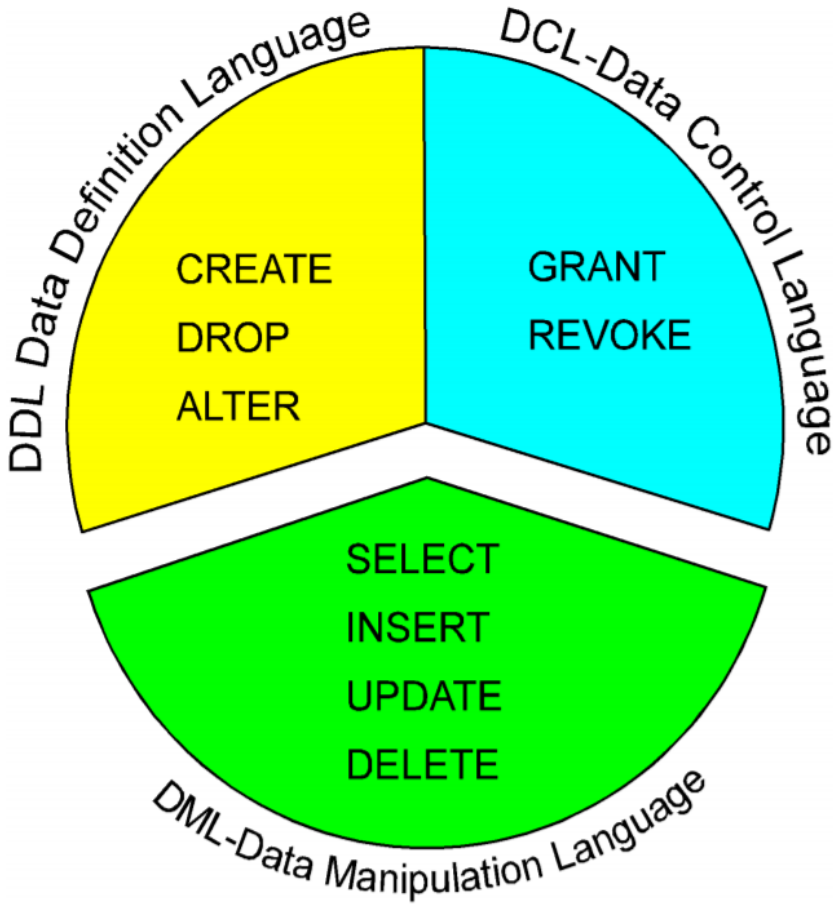
- Edgar Codd, 1969
- Данните се представят като наредени *n*-торки, групирани в релации
- Релациите се представят в релационните БД като (двумерни) таблици
- Простота на модела – само един елемент – релация
- Интуитивен – съответства на начина на човешкото мислене
- Еднообразен начин за представяне на данните и връзките
- Математическа основа

Релационен модел (2)

- E/R описва структурата на данните
- Релационният подход лесно представя тази структура в БД
- Добра практика е дизайнът на БД да се направи, използвайки E/R модела, а след това така направеният модел да се сведе (преобразува) до релационен

Релационен модел (3)

- Релационна алгебра
- SQL
  - Език от високо ниво, базиран на релационната алгебра
  - Декларативен език
  - Повечето диалекти имат някои отклонения от чистия релационен модел



Основни понятия (1)

- **Атрибутите** на една релация служат за имена на колони
  - Обикновено описват значението на елементите в съответните колони, напр. дължина на филм в минути
- **Схема на релация** се образува от нейното име и множеството от атрибутите й
  - Означава се с името, следвано от списък с атрибути, ограден в скоби
  - Movies (title, year, length, filmType)
  - Атрибутите на една релация са множество, а не списък, но обикновено указваме “стандартен” ред на атрибутите
- **Схема на (релационна) база от данни** е множеството от схемите на всички релации в дизайна на една релационна БД

Основни понятия (2)

- **Кортеж** (tuple): една наредена *n*-торка от релацията
  - Или: един ред, без заглавния (който съдържа имената на атрибутите)
  - Има по една стойност за всеки от атрибутите на релацията
  - Релацията е множество от кортежи
  - Пример: (Star Wars, 1977, 124, color) – спазваме реда, в който атрибутите са изброени в схемата на релацията
  - Релационният модел изисква всички стойности в кортежите да са атомарни (цяло число, низ), а не делими на по-малки елементи като списък, масив и т.н.
- **Домейн**
  - Асоциира се с всеки атрибут и определя допустимите стойноности, които може да приема всеки елемент
  - Пример: title – низ, year – цяло число, color – { color, blackAndWhite }
  - Елементите на всеки кортеж трябва да имат стойност, която принадлежи на домейна на съответния атрибут
  - Част от схемата на релацията, въпреки че няма нотация за него

Пример

Схема на релация:

MovieStar (name, address, gender, birthdate)

Атрибути:

name, address, gender, birthdate

Кортеж:

('Jane Fonda', 'Turner Av. ', 'F', '1977-07-07')

Домейни:

name: string

birthdate: date

MovieStar(name: string, address: string, gender: string, birthdate: date)

	NAME	ADDRESS	GENDER	BIRTHDATE
1	Jane Fonda	Turner Av.	F	1977-07-07
2	Alec Baldwin	Baldwin Av.	M	1977-07-06
3	Kim Basinger	Baldwin Av.	F	1979-07-05
4	Harrison Ford	Prefect Rd.	M	1955-05-05
5	Debra Winger ...	A way	F	1978-06-05
6	Jack Nicholson ...	X path	M	1949-05-05
7	Sandra Bullock ...	X path	F	1964-07-26

Ред на атрибутите и на кортежите

- Може да сменим реда на атрибутите и реда на кортежите и релацията ще остане същата
- Долните две таблици са две различни представяния на една и съща релация

name	address	gender	birthdate
Jane Fonda	Turner Av.	F	1977-07-07
Alec Baldwin	Baldwin Av.	M	1977-07-06

name	address	birthdate	gender
Jane Fonda	Turner Av.	1977-07-07	F
Alec Baldwin	Baldwin Av.	1977-07-06	M

Ключ на релация

- Ключ наричаме множеството от атрибути на релацията, които уникално определят всеки кортеж в релацията
- Пример: в релацията MovieStar ключът се състои от атрибута name

Екземпляр (инстанция) на релация

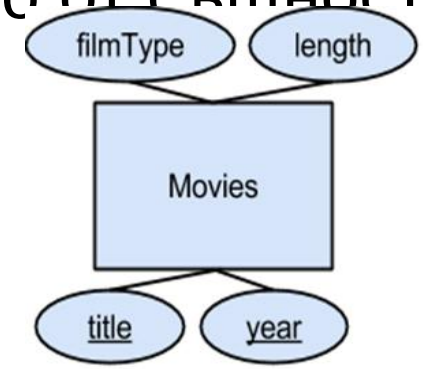
- Една релация не е статична, а се променя с времето
- Два типа промени:
  - Добавяне, изтриване и промяна на кортежи – често
  - Промени в схемата (напр. добавяне на атрибути) – рядко
- Екземпляр (инстанция) на релация: множеството от кортежи за дадена релация
- Инстанциите се променят с времето
- Множеството от кортежи, които в текущия момент се намират в релацията, се нарича текуща инстанция на релацията
- Схема ≠ Екземпляр

От E/R диаграми към релации

- Основни правила:
  - Всяко множество от същности се преобразува в отделна релация със същите атрибути
  - Всяка връзка се преобразува в релация, чиито атрибути са:
    - Ключовете на свързаните същности
    - Собствените атрибути на връзката (ако има такива)
  - В определени случаи комбинираме някои от релациите в една
- Специални случаи:
  - Слаби множества от същности
  - Is-a връзки и подкласове

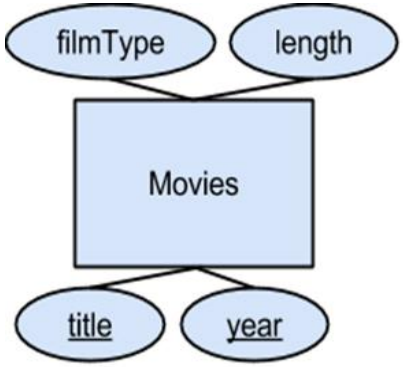
Преобразуване на множества от същности

- Преобразува се до релация с:
  - **Име**, съвпадащо с името на множеството от същности
  - **Атрибути**, съвпадащи с атрибутите на множеството
  - **Ключ**, образуван от ключовите атрибути на множеството
- Новата релация няма индикации за връзките, в които участва множеството от същности
- Пример:



Movies (<u>title</u>, <u>year</u>, length, filmType)

- Екземпляр/инстанция на релацията Movies (title, year, length, filmType):

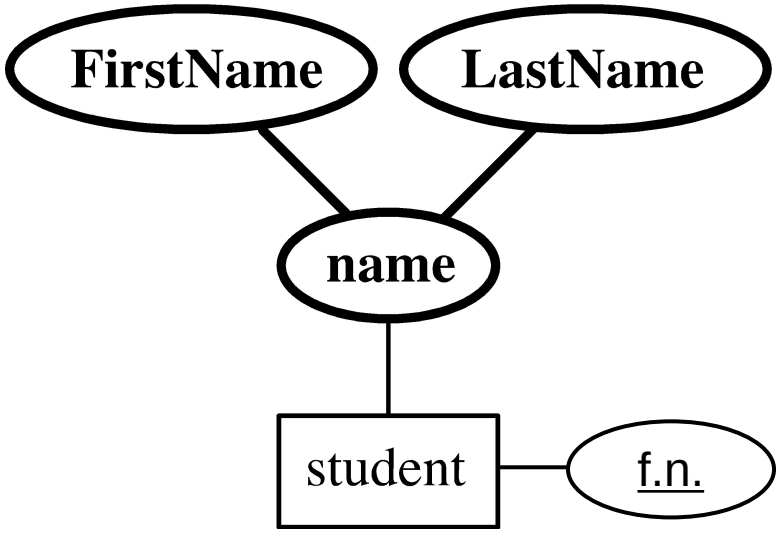


Movies

title	*year*	*length*	*filmType*
Star Wars	1977	124	color
Mighty Ducks	1991	104	color
Wayne's World	1992	95	color

Композитни атрибути

- При преобразуване на композитен атрибут се създават отделни атрибути за всеки податрибут



- Student (<u>fn</u>, firstName, lastName)

Производни атрибути

- Няма нужда да се представят в релацията
- Могат да бъдат изчислявани от заявките, с които извличаме данни
- Примери

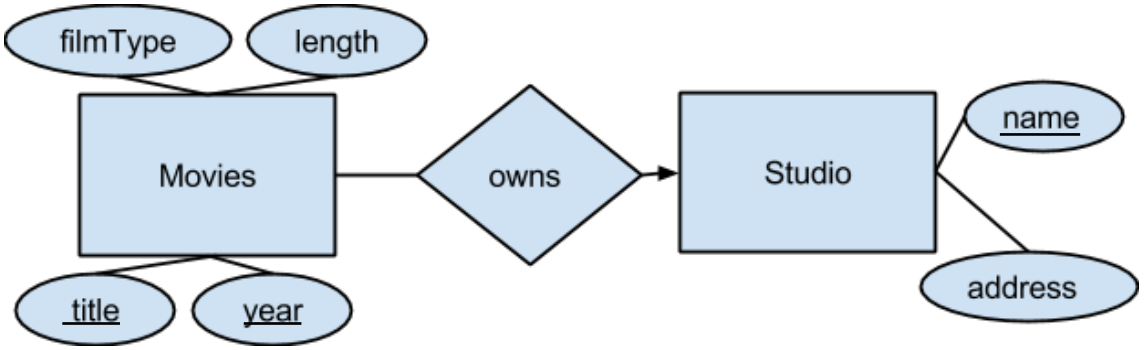
Преобразуване на връзки

- За всяко множество от същности, участващо във връзката R:
  - Взимаме ключовите атрибути и те стават атрибути на релацията R
  - Ако връзката има атрибути, 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, преименуваме атрибутите

Ключ на релацията, представяща връзка

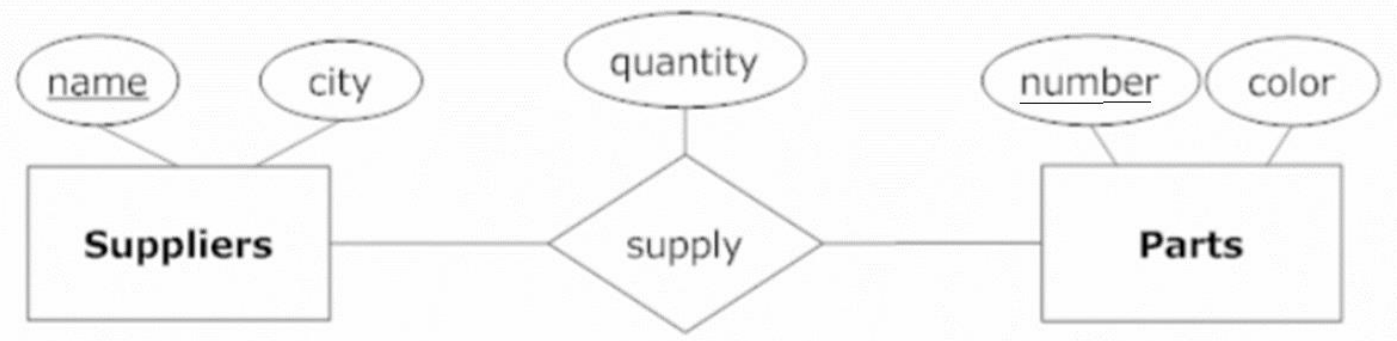
- Много към много
  - Ключовите атрибути на свързаните множества
- Много към едно от E към F
  - Ключовите атрибути само на E

Пример №1



- Owns (<u>title</u>, <u>year</u>, studioName)

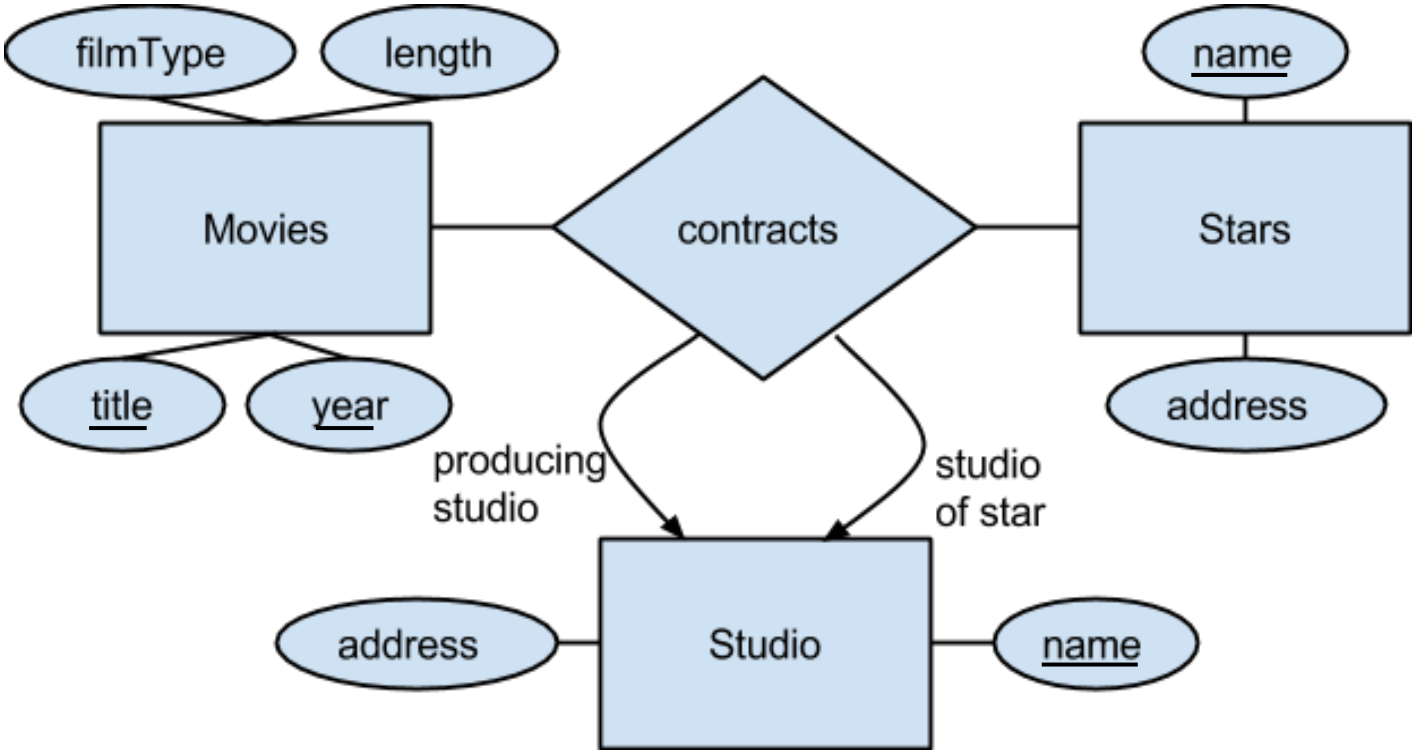
Пример №2



- Suppliers (name, city)
- Parts (number, color)
- **Supplies (supplierName, partNo, quantity)**

Пример 3 (1)

- Връзка с роли



- Contracts (<u>title</u>, <u>year</u>, <u>starName</u>, starStudio, producingStudio)

Пример 3 (2)

- Contracts

Movies		***Stars***	***<Stars-Studios>***	***<Movies-Studios>***
title	***year***	***StarName***	***studioOfStar***	***Prod.Studio***
Star Wars	1977	Carrie Fisher	Fox	Fox
Star Wars	1977	Mark Hamill	Col.Tristar	Fox
Star Wars	1977	Harrison Ford	Fox	Fox
Mighty Ducks	1991	Emilio Estevez	Warner Bros	Disney
Wayne's World	1992	Dana Carvey	Fox	Paramount
Wayne's World	1992	Mike Meyers	MGM	Paramount

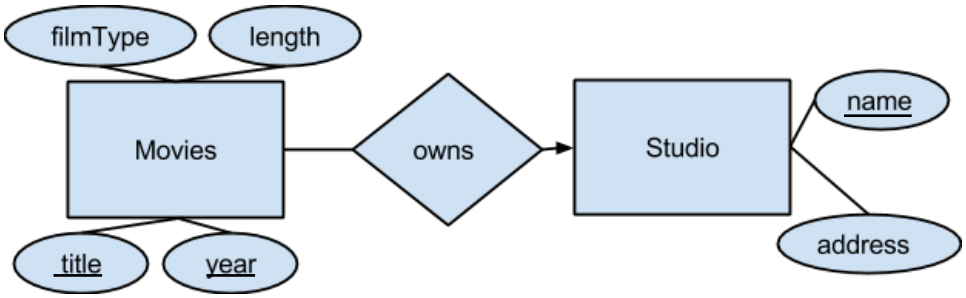
Комбиниране на релации

- В практиката обикновено не се създава отделна релация за връзките от тип M:1 и 1:1
- Нека множеството от същности E е свързано с F посредством връзката R, която е “много към едно” от E към F
- И двете релации, получени при преобразуването на E и R, ще съдържат ключовите атрибути на E, които уникално определят всяка същност в E
- В релацията за R ще имаме ключовите атрибути на E, ключовите атрибути на F и атрибутите на R
- Понеже R е “много към едно”, всички тези атрибути на R ще имат стойности, които са уникално определени от ключовете за E и по тази причина може да я обединим с релацията за E

Пример

Movies	*title*	*year*	*length*	*filmType*
	Star Wars	1977	124	color
	Mighty Ducks	1991	97	color
	Wayne's World	1992	109	color

Owns	*title*	*year*	*studioName*
	Star Wars	1977	Fox
	Mighty Ducks	1991	Disney

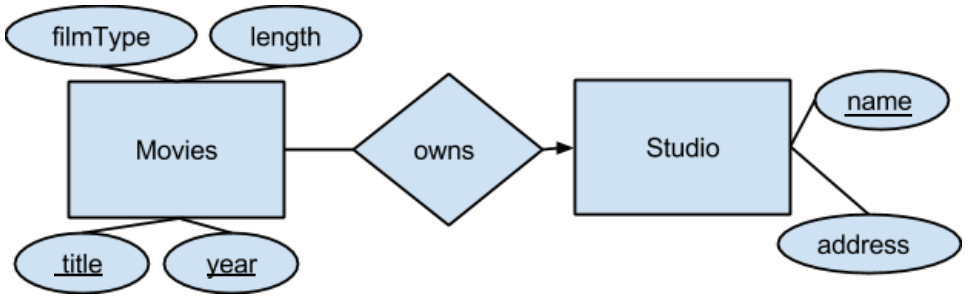


Movies	*title*	*year*	*length*	*filmType*	*studioName*
	Star Wars	1977	124	color	Fox
	Mighty Ducks	1991	97	color	Disney
	Wayne's World	1992	109	color	NULL

Правила за комбиниране на релации

- Имаме връзка R от тип “много към едно” от E към F
  - “много”-краят е към E
- Комбинираме релациите за E и R в релация със следните атрибути:
  1. Всички атрибути на E
  2. Ключовите атрибути на F
  3. Атрибутите на R
- Ключът на новата релация се състои само от ключовите атрибути на E
- Именуваме новата релация с името на множеството от същности E
- За същностите от E, които не са свързани с нито една същност от F, атрибутите по т. 2 и т. 3 ще имат стойност NULL в кортежите за E
- NULL стойностите се използват, когато стойността липсва или не е известна
  - NULL стойностите са неформална част на релационния модел
- Как би изглеждала новата релация, ако комбинираме R с F?

Пример



- Movies (<u>title</u>, <u>year</u>, length, filmType)
- Studio (<u>name</u>, address)
- ~~Owns (<u>title</u>, <u>year</u>, studioName)~~

Оптимизиран релационен модел:

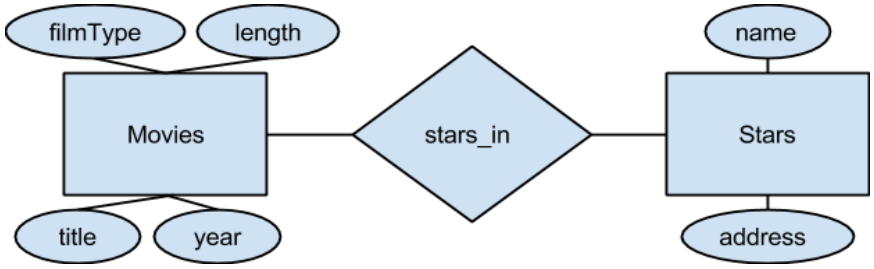
- Movies (<u>title</u>, <u>year</u>, length, filmType, studioName)
- Studio(<u>name</u>, address)

(Някои) предимства на оптимизирания модел

- Всички атрибути, които зависят от ключа на E, са обединени в една релация
- Ако има и други M:1 връзки от E към други множества от същности, техните ключови атрибути също ще бъдат част от релацията E
- Много по-ефективно е да се даде отговор на заявка, която включва атрибути от една релация, отколкото на заявка, която включва атрибути от две или повече релации

Много към много - оптимизиране

- Не е добра идея да оптимизираме връзки “много към много”
- Ще доведе до излишества
- Пример (правилен):
  - Movies (title, year, length, filmType)
  - Stars (name, address)
  - Stars_in (title, year, starName)



- Оптимизирана версия (неправилна!):
  - Movies (title, year, length, filmType, starName)
  - Stars (name, address)
- Ако в един филм играят няколко звезди, информацията за този филм ще бъде повторена – имаме излишество

Едно към едно – оптимизиране

- На пръв поглед е очевидно – прилагаме алгоритъма за M:1
  - Presidents (name, networth)
  - Studios (name, address)
  - ~~Runs (studioName, presidentName)~~

    →
  - Presidents (name, networth)
  - Studios (name, address, presidentName)
- Към коя от двете релации ще добавим ключа на другата?
  - Presidents (name, networth, studioName)
  - Studios (name, address)
- Може ли и към двете?
- А не може ли да обединим и трите релации в една?
  - Studios (name, address, presidentName, presidentNetworth)

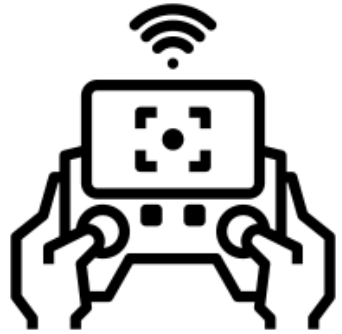
В едната или в другата релация?

- Идентифицираме коя релация е parent и коя – child (т.нар. parent-child relationship)
  - Добавяме ключа на parent в child, т.е. детето знае кой му е родител
  - По същия начин е и при връзки 1:M – един родител, много деца
  - Записите в коя таблица ще бъдат добавяни първи? Това ще бъде родителската таблица
  - Има ли край, в който връзката не е задължителна (partial participation)? Там ще е “детето”
    - Служители и незадължителни служебни лаптопи – лаптопите ще “знаят” кой ги ползва
- Друга стратегия, базирана на ефективността: добавяме външния ключ в таблицата с по-малко редове

За и против единствена релация при връзка 1:1

- Пример: обща релация за device и remote controller
- Предимство (важно): по-малко таблици – по-ефективни заявки за извличане на данни

- Недостатъци:
  - Семантичен проблем – не звучи добре характеристиките на устройство и на дистанционното да са в една таблица
  - Може да е трудно да се разграничи за коя същност се отнася даден атрибут
  - Много NULL стойности, когато в някой от краищата връзката е 0..1, а не 1
  - Възможно е кеширането да е неефективно, ако редовете са дълги
  - По-трудно осигуряване на различна степен на сигурност, тригери само за атрибутите на едната същност и др.

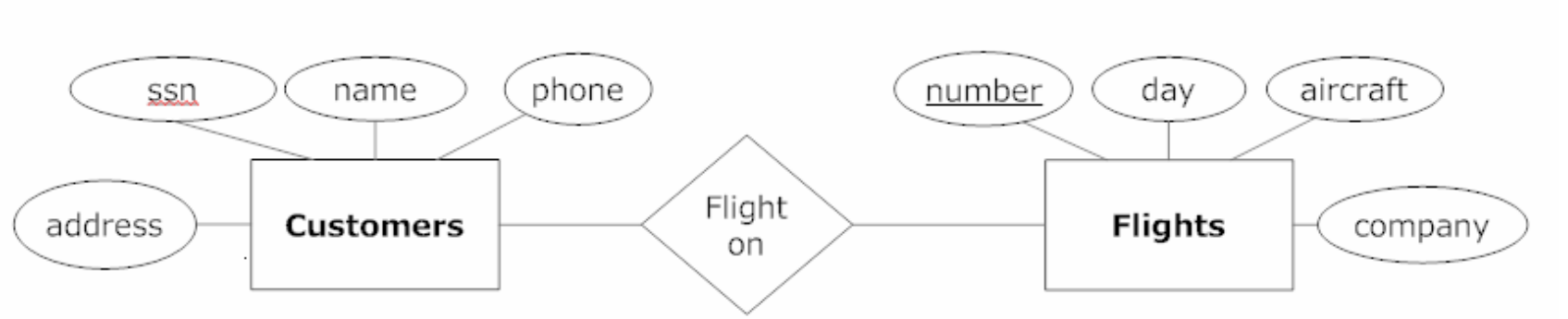


Преобразуване на връзки – обобщение

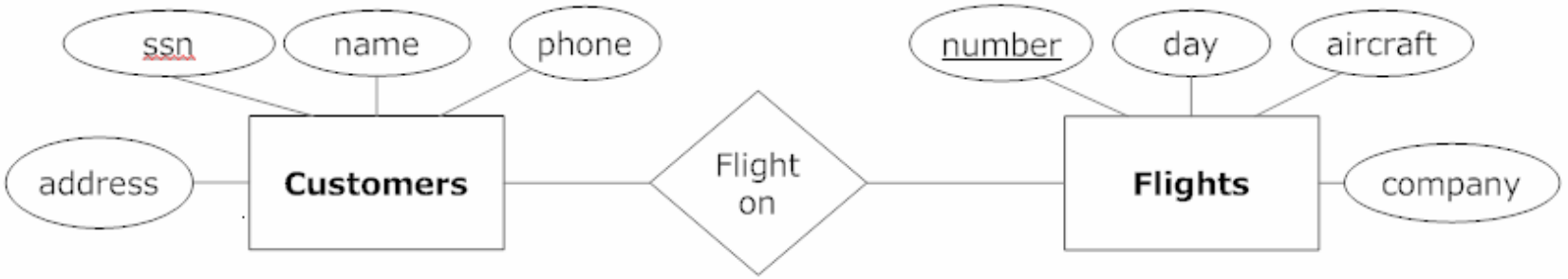
- Свойства на връзките (от предишната лекция)
  - Име
  - Множественост (Cardinality)
  - Степен (Degree)
  - Атрибути
  - Роля
  - Участие (Participation)
- Да обобщим: как се представя всяко от свойствата в релационния модел?

Задача 1

- Да се преобразува следният E/R модел до релационен:



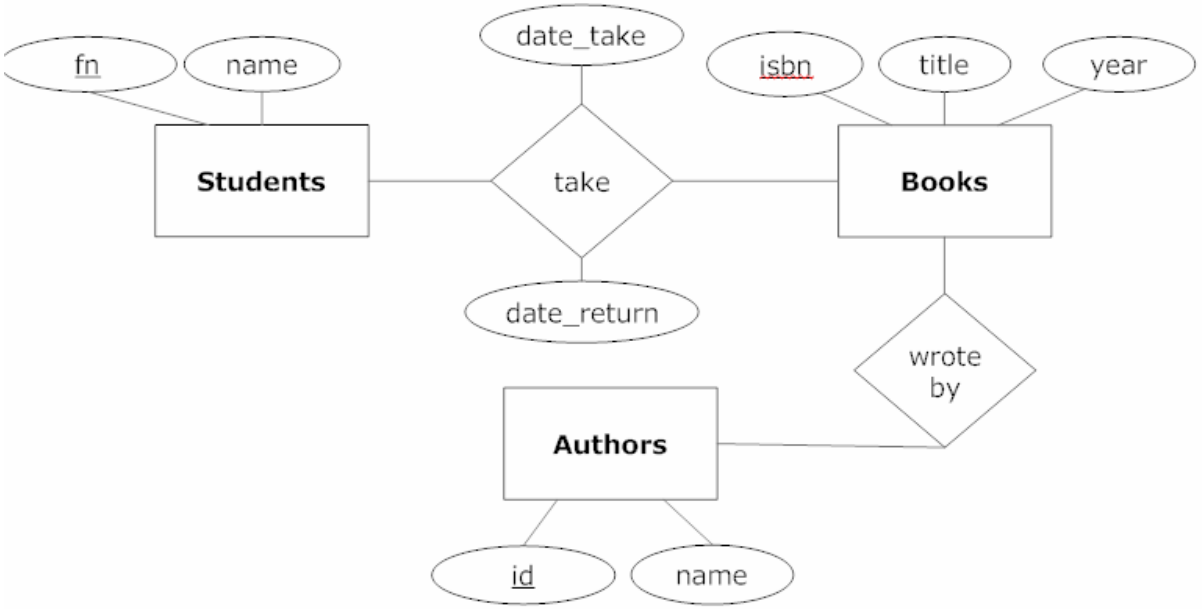
Решение на задача 1



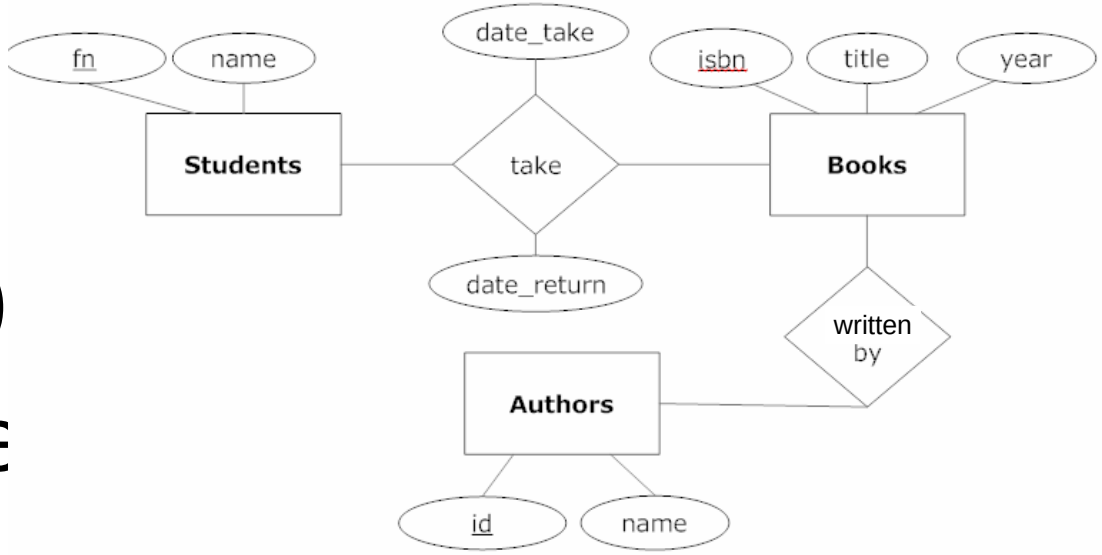
- Customers (ssn, name, phone, address)
- Fights (number, day, aircraft, company)
- FightOn (customerSSN, flightNumber)

- В син цвят са отбелязани външните ключове
  - По-късно ще бъде разгледано какво е външен ключ

Задача 2

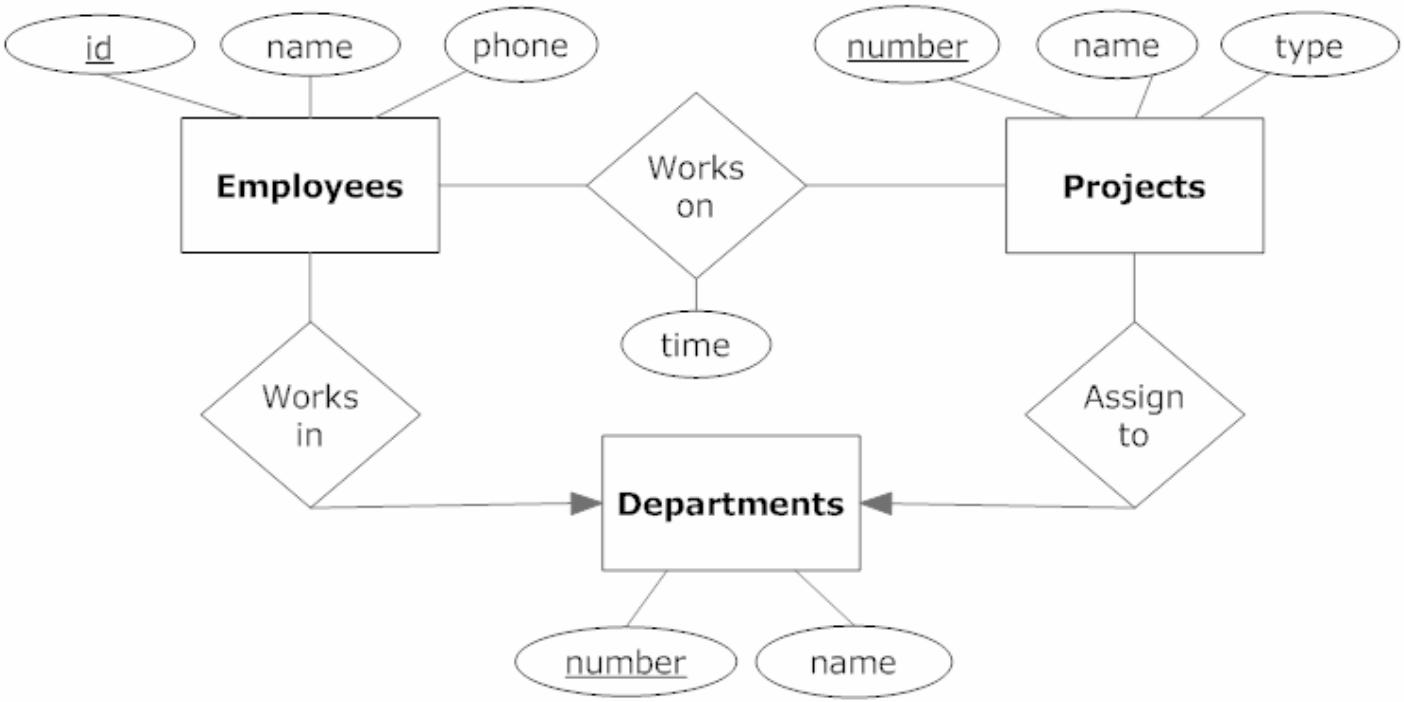

fn
name
date_take
isbn
title
year
Students
take
Books
date_return
wrote
by
Authors
id
name

Решение на задача 2



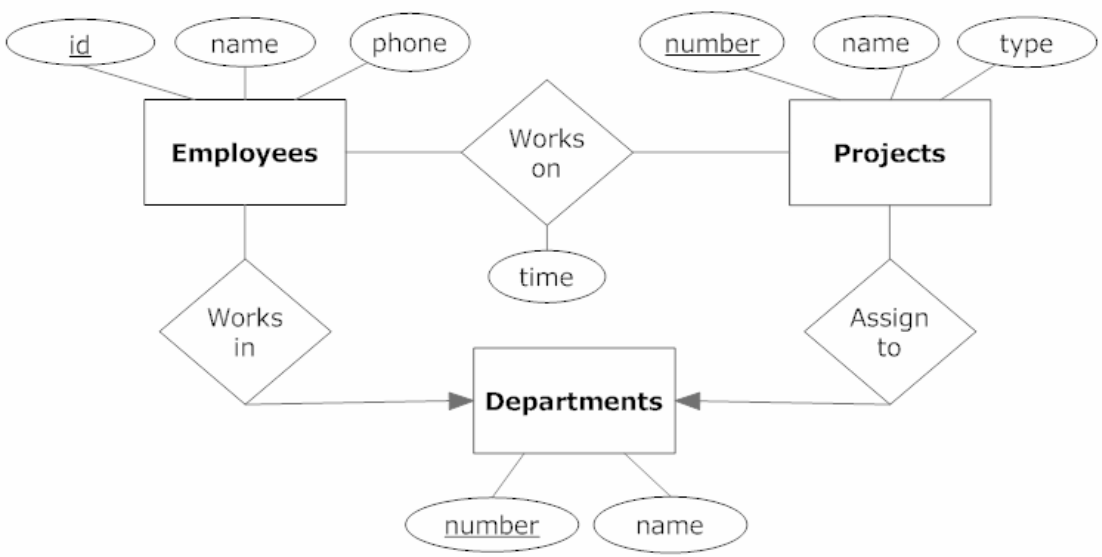
- Students (fn, name)
- Books (isbn, title, ye
- Authors (id, name)
- Take (studentFN, bookID, date_take, date_return);
- WrittenBy (bookID, authorID)

Задача 3


id
name
phone
number
name
type
Employees
Works on
Projects
time
Works in
Assign to
Departments
number
name

Решение на задача 3

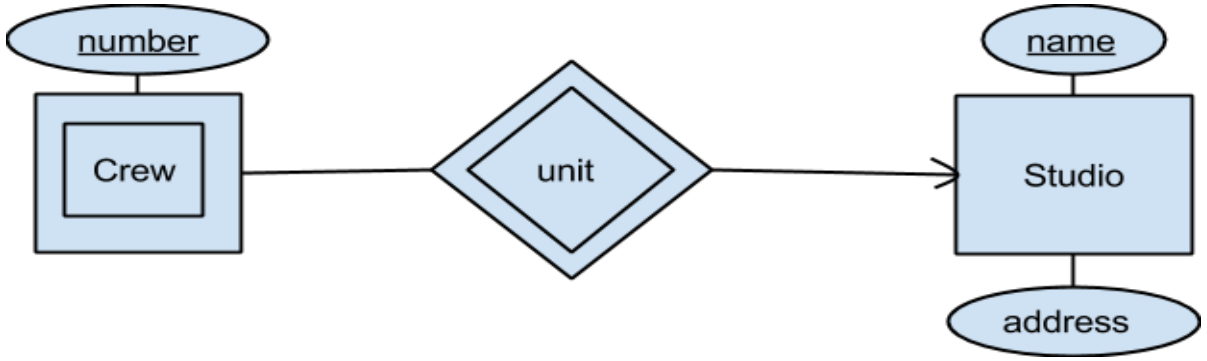
- Employees (id, name, phone)
- Projects (number, name, type)
- Departments (number, name)
- WorksOn (employee_id, project_number, time)
- WorksIn (employee_id, department_number)
- AssignedTo (project_number, department_number)



Оптимизиран релационен модел:

- Employees (empid, name, phone, dnumber)
- Projects (pnumber, name, type, dnumber)
- Departments (dnumber, name)
- WorksOn (empid, pnumber, time)

Преобразуване на слаби множества от същности (1)



- Нека W е слабо множество от същности
- Да си припомним: кои атрибути включва ключът на слабо множество?
- W се преобразува до релация, която включва както атрибутите на W, така и ключовите атрибути на поддържащите го множества

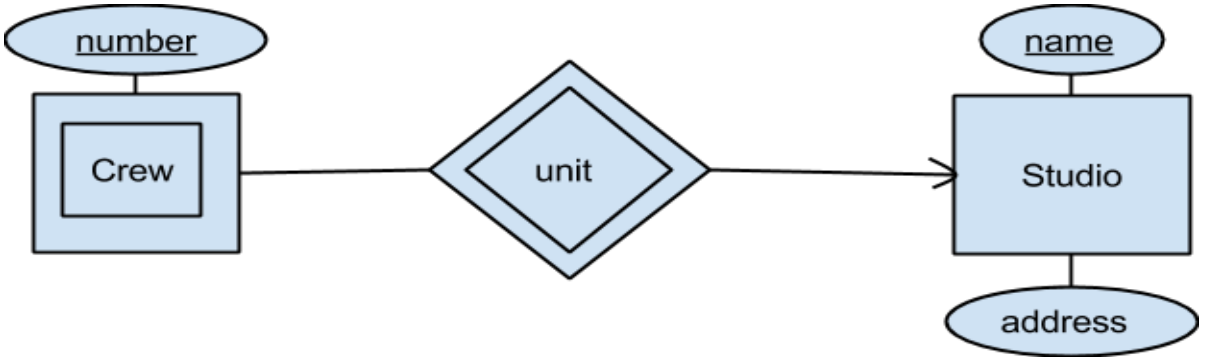
Преобразуване на слаби множества от същности (2)

- Поддържащата връзка НЕ се преобразува в отделна релация
  - Създава излишество
  - Атрибутите й биха съдържали ключовите атрибути на W и на поддържащото множество E
  - Връзката между W и E е “много към едно” и ключът на E е подмножество на ключа на W

Преобразуване на слаби множества от същности (3)

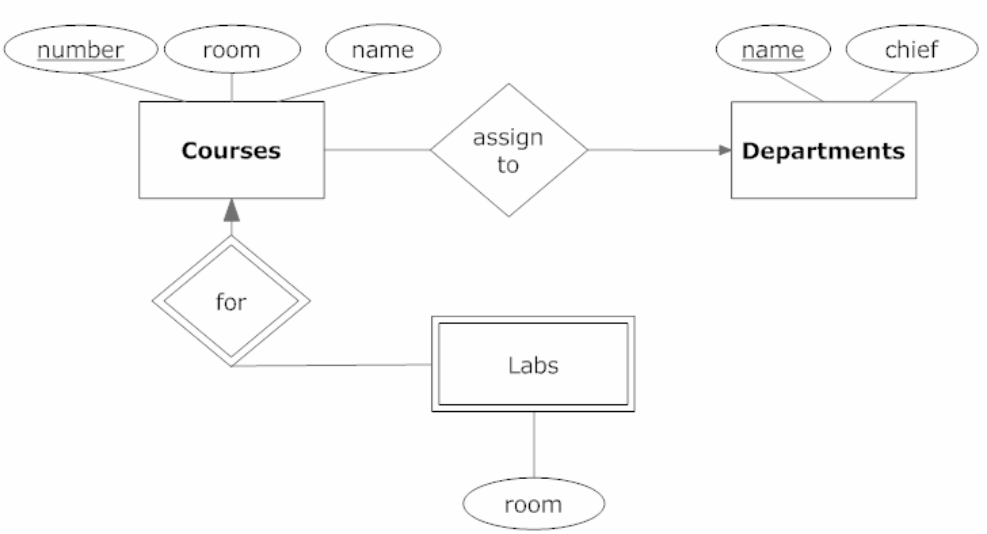
- Алгоритъм:
  - W се преобразува до релация със следните атрибути:
    - Всички атрибути на W
    - Всички атрибути на поддържащите връзки
    - Ключовите атрибути на всички поддържащи множества
  - Ключът й се състои от ключовите атрибути на W и ключовите атрибути на поддържащите го множества от същности
  - За поддържащите връзки не се създават релации
  - Ако W участва във връзки, различни от поддържащи, за ключ използваме атрибутите, описани по-горе
- Ако има атрибути с еднакви имена, ги преименуваме

Пример



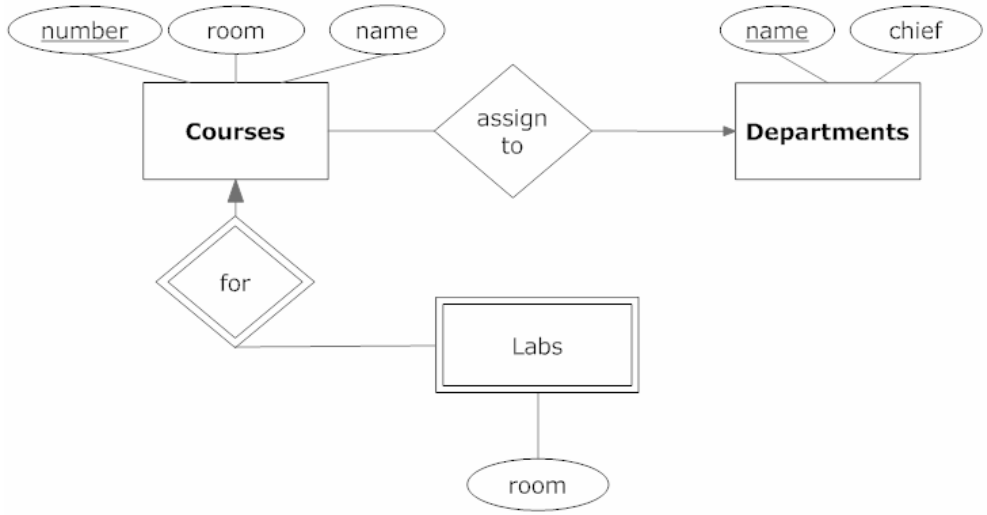
- Studio (name, address)
- Crews (number, studioName)

Задача 4


number
room
name
name
chief
Courses
assign
to
Departments
for
Labs
room

Решение на задача 4

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



Оптимизиран релационен модел:

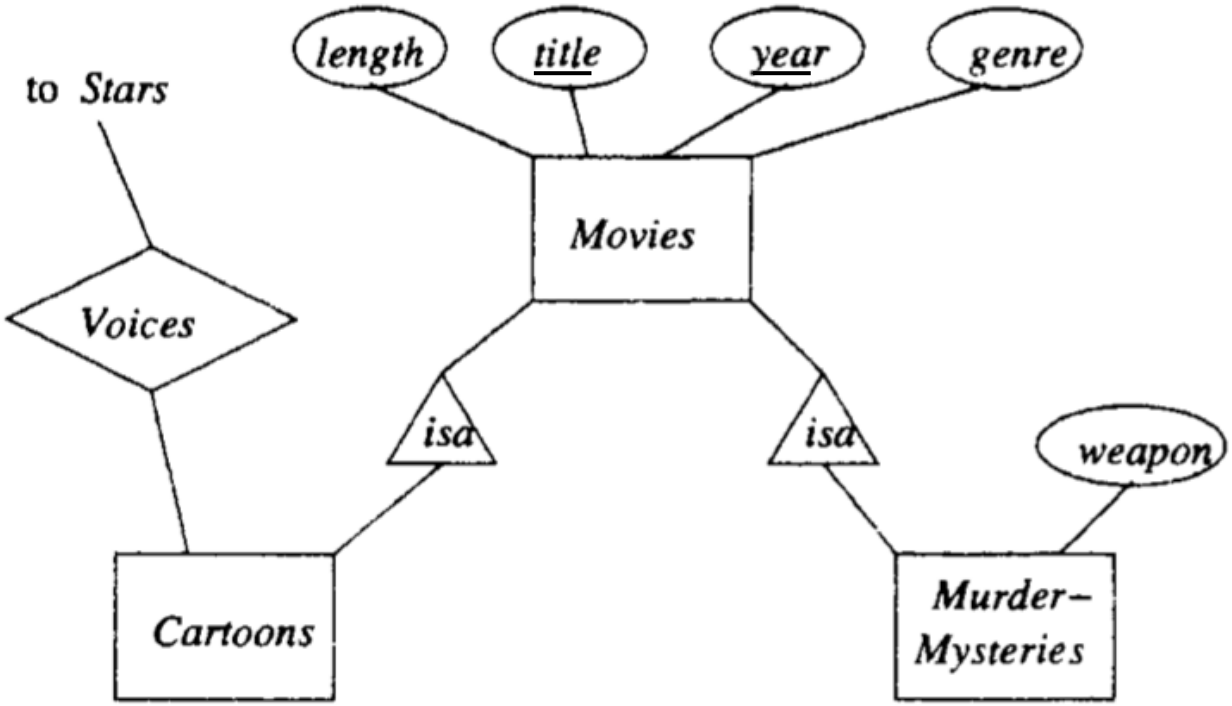
- Courses (number, room, name, deptName);
- Departments (name, chief);
- Labs (courseNo, room);

Преобразуване на йерархии от класове (1)

- При множества от същности, организирани в isa-йерархия:
  - съществува множество от същности – корен на йерархията
  - тази същност има ключ, който служи за идентифициране на всяка същност в йерархията

Преобразуване на йерархии от класове (2)

- Да си припомним



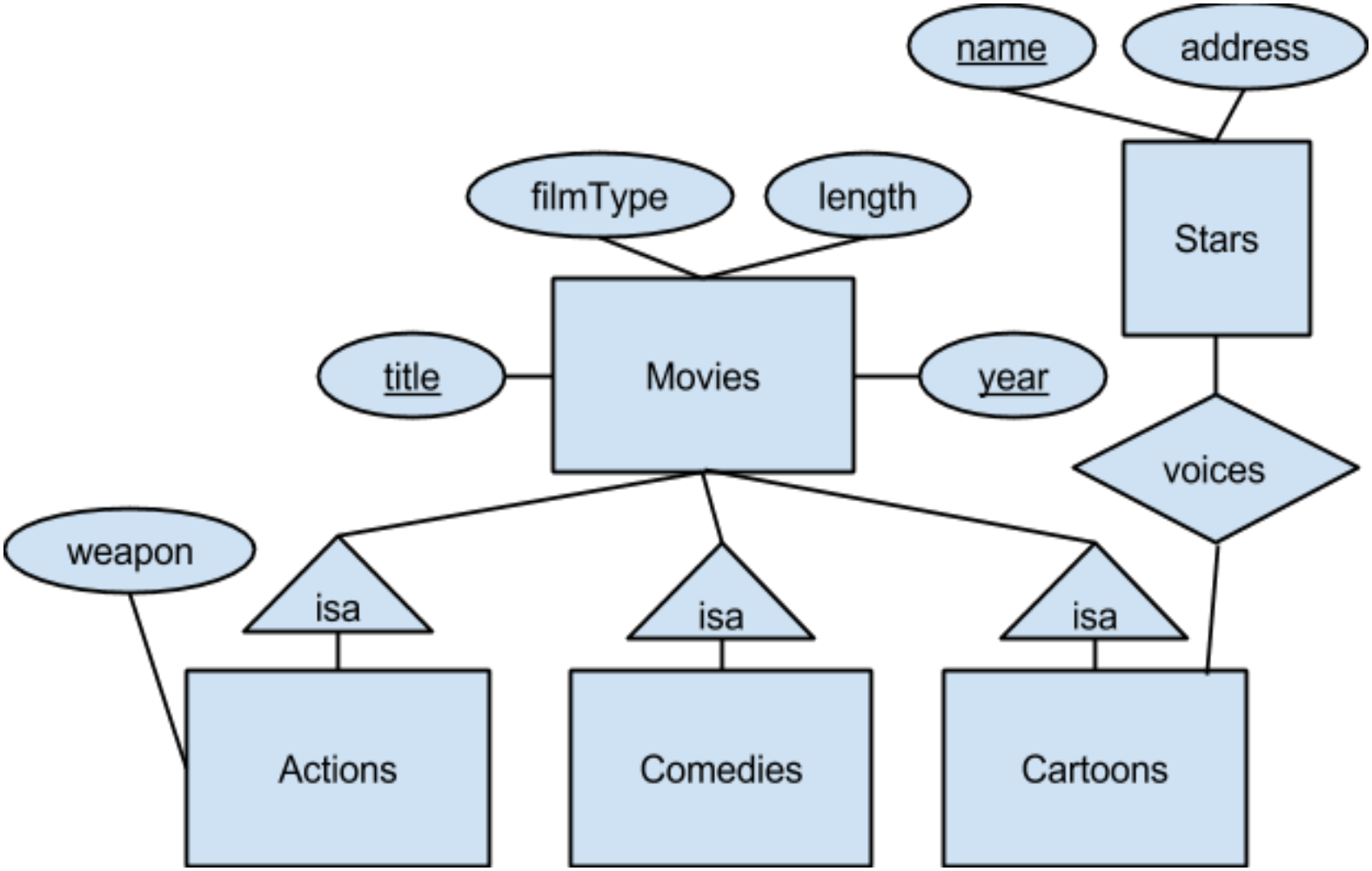
Преобразуване на йерархии от класове (3)

- Различни подходи
  - E/R подход: за всяко множество от същности E от йерархията създаваме релация, която включва атрибутите на E и ключа на корена
  - Обектно-ориентиран подход: за всяко възможно поддърво се създава релация с всички атрибути на всички множества от същности от поддървото
  - Null стойности: само една релация с всички атрибути на всички множества от същности от йерархията

E/R подход

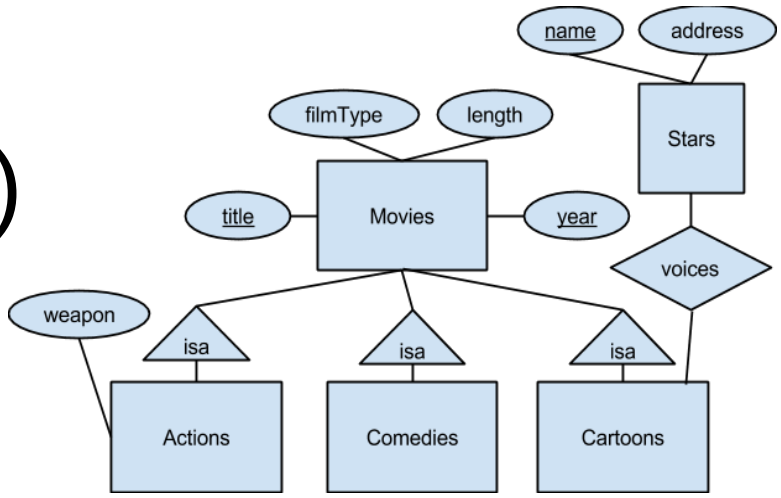
- Създава се релация за всяко от множествата от същности
- Ако множеството от същности не е корен на йерархията, то трябва да съдържа ключовите атрибути от корена и всички собствени атрибути
- За is-a връзката не се създава релация

Задача за E/R подход



Решение

- Movies (title, year, length, filmType)
- Actions (title, year, weapon)
- Comedies (title, year)
- Cartoons (title, year)
- Stars (name, address)
- Voices (title, year, name)



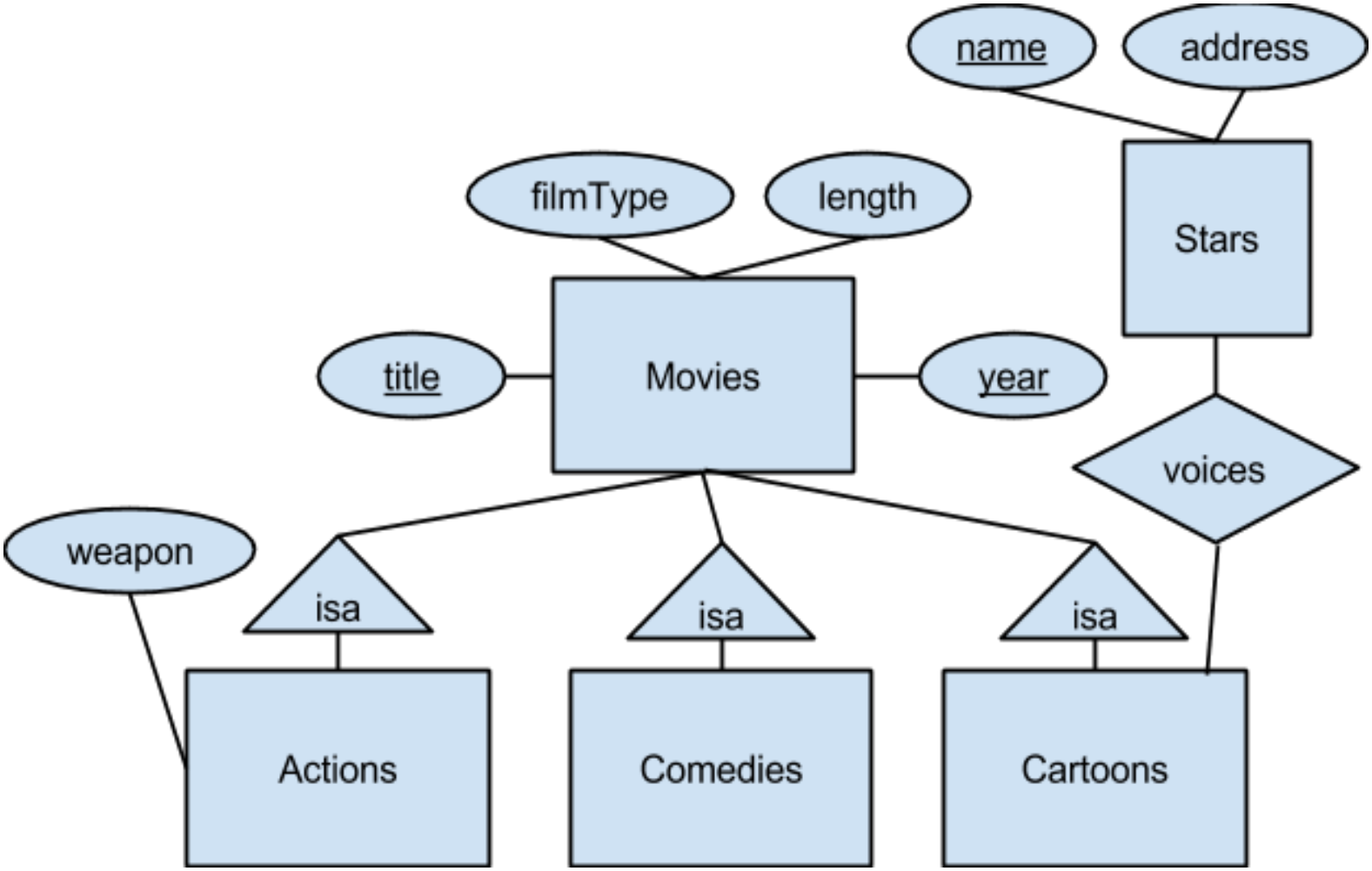
Решение – особености

- Ако един филм е едновременно и екшън, и комедия, и анимационен, то за него ще има по един запис във всяко едно от 4-те множества от 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

- Създава се релация за всяко едно възможно поддърво в йерархията
- Релацията ще съдържа всички атрибути на участващите в поддървото множества от същности
- Нарича се ОО, защото същностите се разглеждат като обекти, които принадлежат точно на един единствен клас
- Напр. даден филм може да бъде едновременно екшън и комедия, друг филм може да не бъде нито екшън, нито комедия, нито анимация

Задача за ОО 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)

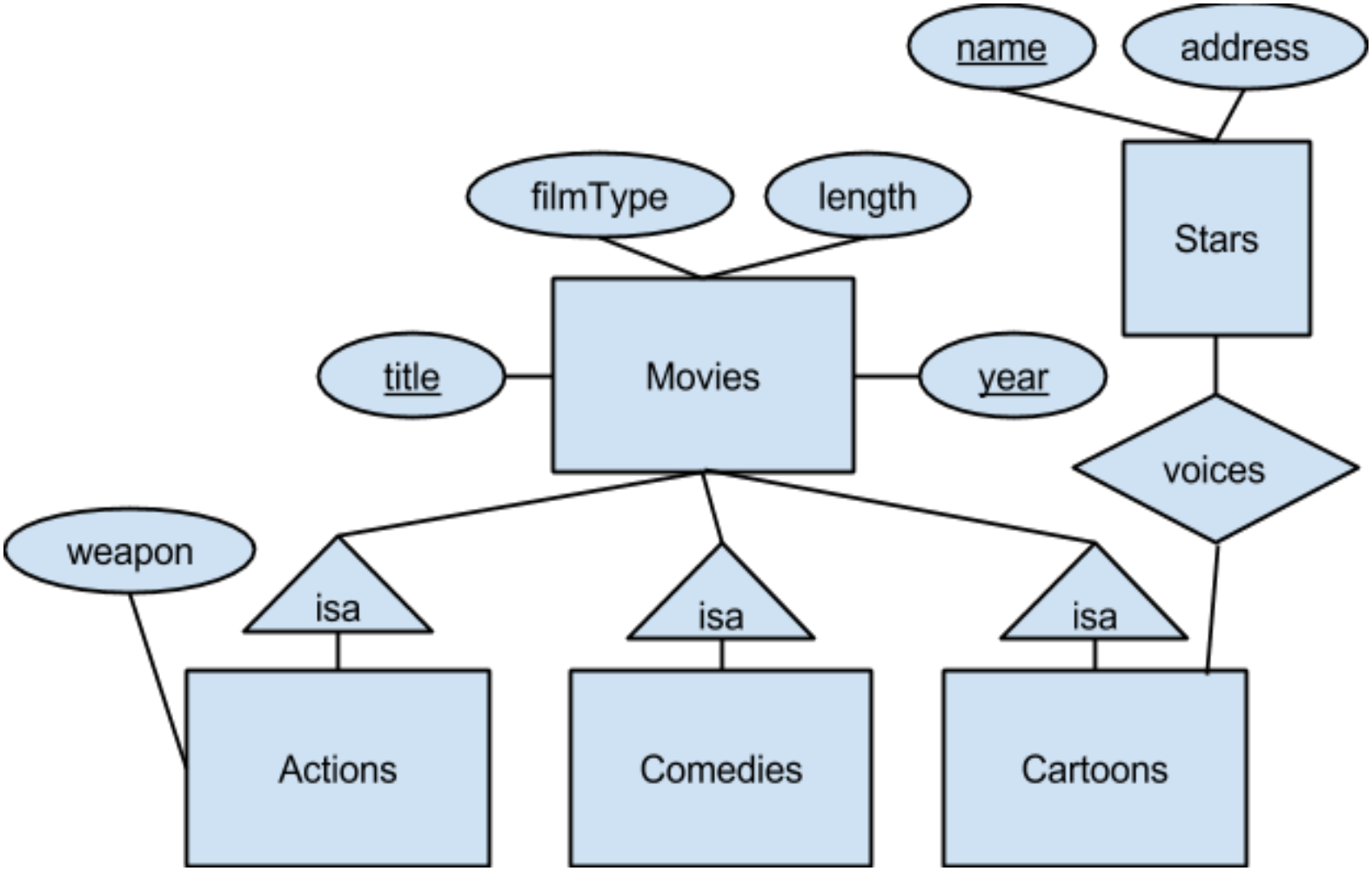
Решение - особености

- Броят на релациите е 8, т.е. 2 на степен 3 (брой подкласове)
- Релациите Movies, Movies_Cartoons и Movies_Comedies имат еднакви атрибути, но не можем да ги обединим в една
  - Няма да знаем дали даден филм е анимация или комедия

Null подход

- Йерархията от подкласове се преобразува в една единствена релация
- Релацията съдържа всички атрибути на всички множества от същности, 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
  - Напр. анимационен филм, който не е екшън, няма стойност за атрибута “оръжие”
- NULL в SQL се използва в два случая:
  - Липсва информация
  - Няма смислена стойност, която да използваме

Задача за Null подход



Решение

- Movies (title, year, length, filmType, weapon)
- Stars (name, address)
- Voices (title, year, name)

Решение - особености

- 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

Кой подход да изберем?

- Всеки от подходите има различни предимства и недостатъци и е подходящ за различни ситуации
- Какви заявки ще се изпълняват най-често?
  - При някои от подходите всички необходими данни биха били в една таблица и това ще направи заявките за извличане по-ефективни
- Може ли коренът да бъде инстанциран, напр. има ли филми, които не са нито комедии, нито анимации, нито екшъни?
  - Ако не, може да нямаме релация за корена
- Може ли един филм да бъде и екшън, и комедия?
- Очаква ли се да бъдат добавени нови подкласове в бъдеще?

Предимства и недостатъци на подходите (1)

- E/R подход

  + Чист дизайн, няма повторение на данни (освен ключовите стойности)

  + Всички филми са в една таблица

  – Атрибутите на даден екшън са разпръснати в няколко таблици

  – Ако искаме да няма филми, които не попадат в никой конкретен тип, или искаме да няма филми, които са едновременно екшъни и комедии, ще бъде трудно да реализираме ограничението

Предимства и недостатъци на подходите (2)

- ОО подход

  + Атрибутите на екшъните са в една таблица

  – Филмите са разпръснати в различни таблици

  – Трудно осигуряване на уникалност на даден общ атрибут (indexed view)

  – Дублирана информация за връзките, които са общи за всички филми

Предимства и недостатъци на подходите (3)

- Null подход

  + Всичко е в една таблица

  + Прост дизайн

  – Много NULL стойности, особено при атрибути, които са специфични за малко същности

  – Трудно добавяне на нов подклас, трудна промяна на атрибутите – засяга се цялата таблица

  – Труден при сложни йерархии

  – Трудно указване на ограничения, специфични за конкретен подклас

  – Въпреки че някои заявки биха се ускорили, други биха се забавили, ако имаме много редове, които са от подкласове, които не са “интересни” за конкретната заявка

  – Как ще разграничаваме екшън с неизвестно оръжие от обикновен филм?

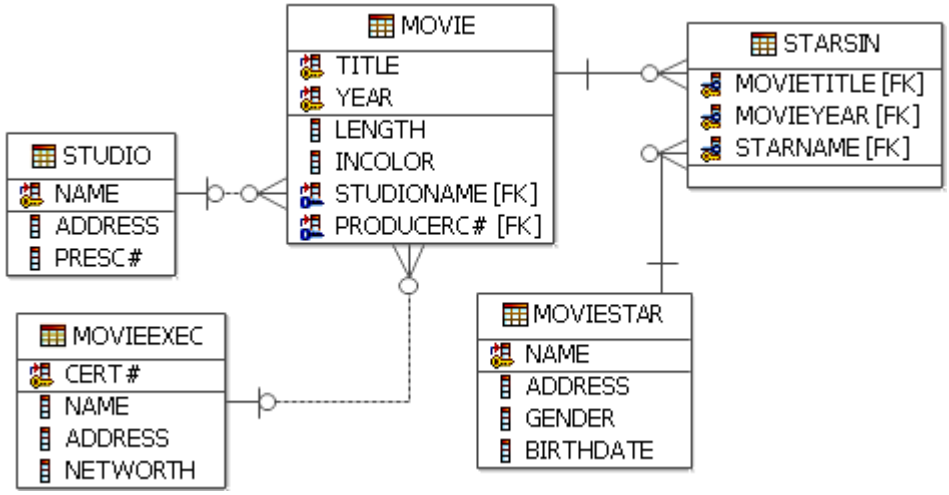
Задача

- Кой подход да изберем, ако често ще се изпълняват заявки като следната:
  - Кои филми от 1999 са по-дълги от 150 минути?
- А за следната:
  - Какви оръжия са използвани в анимации, по-дълги от 150 минути?

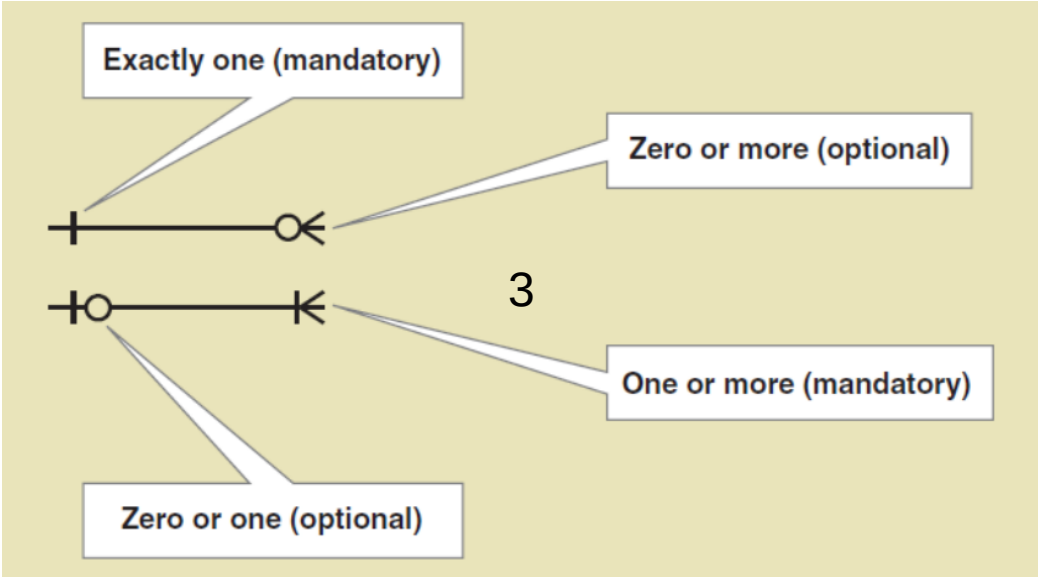
Сурогатен ключ

- Естествен (natural) ключ
  - често се състои от префикс и пореден №, напр. INV-1234
- Сурогатен (изкуствен) ключ
  - 1, 2, 3, …
  - UUID
- Ако естественият е редактируем, по-добре да добавим сурогатен
  - { title, year } vs movieId

Диаграми на схема на релационна БД



- Популярна нотация: кракът на враната (crow's foot)



Обобщение

- Релационният модел работи с едно единствено понятие – релация
- Преобразуване от E/R модел към релационен
  - Оптимизиране на връзки “много към едно”
  - Особености при преобразуване на слаби множества от същности
  - Три подхода за преобразуване на isa връзки

Въпроси?

В презентацията са използвани задачи и други материали, копирани от презентации на доц. д-р Радослава Христова