

# 8. Лекция. Проектиране на релационни схеми

Category	Empty
Files	Empty
Created	May 28, 2023 11:22 AM
Reminder	Empty
Status	Open
URL	Empty
Updated	May 28, 2023 11:22 AM

---

## Проектиране на релационни схеми

### Аномалии

- излишество на данните
  - да не допускаме едни и същи данни да се записват на различни места, защото това освен излишество на памет води и до грешки, защото при всяка промяна трябва да променяме на всички места
- аномалии при добавяне на данните
- аномалии при промяна на данните
- аномалии при изтриване на данните

### Пример: Релационната схема Movies - аномалии

Повторение на информацията за типа и дължината на филма.



<b>Movies</b>	<u>title</u>	<u>year</u>	<u>length</u>	<u>filmType</u>	<u>studioName</u>	<u>starName</u>
	Star Wars	1977	124	color	Fox	Carry Fisher
	Mighty Ducks	1991	104	color	Disney	Emilio Estevez
	Wayne's World	1992	95	color	Paramount	Dana Carvey
	Star Wars	1977	124	color	Fox	Mark Hamill
	Star Wars	1977	124	color	Fox	Harrison Ford
	Wayne's World	1992	95	color	Paramount	Mike Meyers

Изтривайки информация за филм губим и информацията за актьор

Да въведем един филм в тази релация

Ключове: title, year, starName

Аномалия при въвеждане на информация - не можем да въведем информация за филм без информация за актьор

## Пример 2. Student-course database

Всеки път когато някой студент се яви в някой курс, то толкова пъти ще се запази и описанието на курса

Аномалия при въвеждането - не може да въведем курс преди да има поне един човек, който да го е записал и да го е взел

Аномалия при въвеждането - същото е и за студентите, не може да имаме студент без да е взел курс

Аномалия при изтриване - ако изтрием един курс, то ще изтрием и информация за студенти

Аномалия при изтриване - ако имаме един курс, в който само един студент се е явил и този студент напусне, то ще загубим информацията за курса

## Декомпозиция на релации

Декомпозиция на релацията  $R(A_1, \dots, A_n)$  представлява заместването ѝ с множеството релации  $R_1 \dots R_n$ , получени чрез проекции така, че

- $R$  и
- $R_1 \cup R_2 \cup \dots \cup R_n$

имат една и съща схема.

Хубаво е така да направим декомпозицията, че за дадена релация, всички атрибути трябва да зависят функционално само от ключа

## Boyce-Codd Normal Form (BCNF) - дефиниция

Релацията  **$R$  е в BCNF** тогава и само тогава, когато за всяка нетривиална зависимост  $A_1 A_2 \dots A_n \rightarrow B_1 B_2 \dots B_n$  от  $R$ , съответното м-во от атрибути  $\{A_1, A_2, \dots, A_n\}$  е суперключ за  $R$ .

Суперключът съдържа в себе си ключа



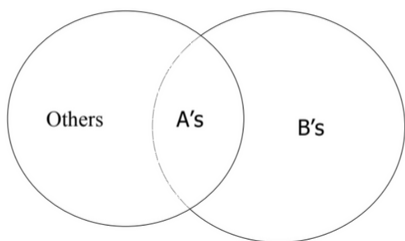
Като видим една релация първо трябва да мислим кой е ключът на тази релация

Трябва да обясним всичко, т.е. да покажем кои са нетривиалните функционални изисквания и да покажем, че удовлетворяват условието на BCNF

## Стратегия на декомпозиция

Ако атрибутите  $A$  определят функционално атрибутите  $B$ , то можем да ги изнесем в отделна релация

Също трябва да направим и друга релация за да съхраним връзката между атрибутите. Затова правим друга релация за others &  $A$



21 слайд

Името на студиото определя адресът на студиото

StudioName  $\rightarrow$  StudioAddr

- StudioName не е суперключ (title, year е ключът), затова е нарушена BCNF

Правим декомпозиция.

## Обобщение за релации с два атрибута

$R(A, B)$

Ако имаме едноатрибутен ключ  $A$ , то релацията е в BCNF

- $A \rightarrow B$  (аналогично за  $B$ -ключ)
- нямаме други функционални зависимости

Ако ключът е  $AB$

За да е релация една таблица, то трябва да имаме ключ, за да идентифицираме всеки отделен кортеж

## Задача - БД за резервиране на билети

БД за резервиране на билети за филми, прожектирани в различни киносалони

- всеки киносалон се намира в строго определен град
- един филм може да се прожектира в няколко града едновременно
- градовете обикновено имат повече от един киносалон
- има киносалони с няколко зали, в които могат да се прожектират едновременно няколко филма

Booking релация (заявки за билети) със следните атрибути

- title - име на филм
- theatre - име на киносалон
- city - град, в който се намира киносалона

Booking(title, city, theatre)

Всеки кортеж се състои от тройката (m,t,c)

Зададени са следните функционални зависимости

Theatre → city

Title, city → theatre

Booking(theatre, title, city)

Booking(title, city, theatre)

- има функционална зависимост, която е нетривиална и лявата ѝ част не е суперключ
  - това е theatre → city
- следвайки стратегията за декомпозиция създаваме две нови релации
  - при кръгчетата - A = theater, B = city

Ще разделим на релации:

Theatres(theatre, city)

Projections(theatre, title)

---

Това разделяне генерира грешно конструирани кортежи.

Не спазваме зависимостта, че title, city → theater

## 3 нормална форма (3-NF) , смекчава условието на BCNF - 3 NF

Релацията  $R$  е в **3 NF** тогава и само тогава, когато за **всяка нетривиална зависимост**  $A_1A_2...A_n \rightarrow B$  от  $R$ , съответното **м-во** от атрибути  $\{A_1, A_2, ..., A_n\}$  е суперключ за  $R$  или  **$B$  е част от ключ**.

Основанието е, когато имаме повече от един вариант за ключ

## Дефиниции

- Атрибут, който е част от ключа, се нарича **първичен атрибут**.
- Релацията  $R$  е в **3 NF** тогава и само тогава, когато за **всяка ФЗ**, или лявата страна е суперключ или дясната е първичен атрибут.

### Дефиниция на 1НФ :

- Всеки компонент на всеки кортеж съдържа атомарно значение.
  - имаме ли таблица, която по нашите дефиниции е релация, то тя е в 1НФ

### Дефиниция на 2 НФ :

- Релацията  $R$  е във 2 NF, когато е изпълнена 1НФ и всеки неключов атрибут е в пълна функционална зависимост от ключа, т.е зависи от целия ключ, а не от някакво подмножество на ключа
  - тук не се елиминират транзитивните зависимости като тази между `studioname` и `адрес`

Досега имаме знания за 4 нормални форми

1 се съдържа във втора,

втора се съдържа в трета,

трета се съдържа в BCNF

В реалните проекти се прави нормализация в 3-та нормална форма.