

» Гл. ас. д-р Георги Чолаков

» Базы от данни

Релационен модел >

Релационен модел

- » Най-широко използвания модел за бази от данни;
- » Базира се на релационна алгебра и теория на множествата;
- » Предложен първоначално през 1970 год. от д-р Edgar Frank Codd;
- » Една от най-значимите разработки на релационния модел е System R, разработена от IBM в края на 70-те, замислена като „доказателство на концепцията“, показваща че РСУБД могат да бъдат създадени и да работят ефективно;



Релационен модел

- » Дава ход да множество разработки като Structured Query Language (SQL), който става стандарт за релационен език;
- » Релационният модел използва колекция от таблици, за да представя едновременно данните и взаимоотношенията между тях;
- » Таблиците са логически структури, не физически.



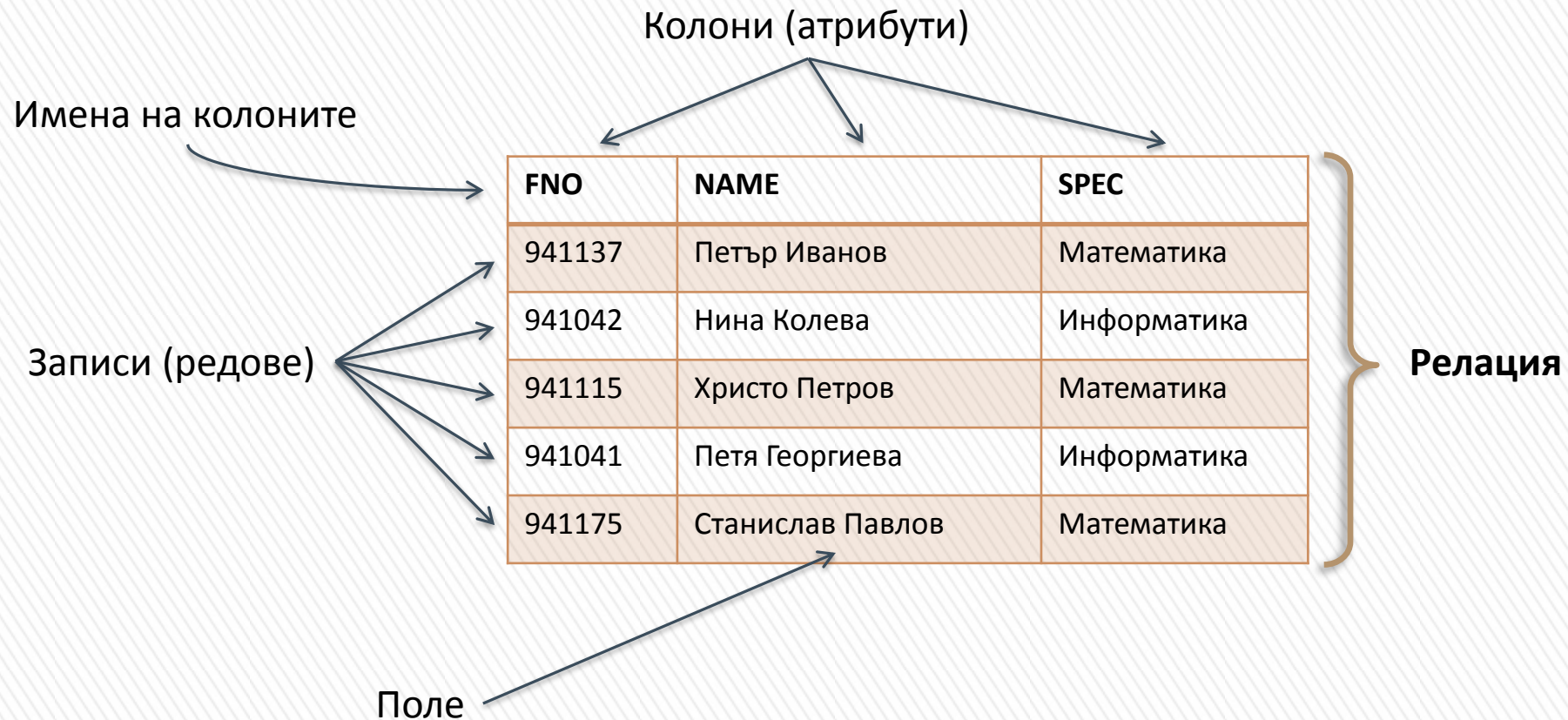
Релационен модел

Релационният модел се състои от 3 части:

- **Логическа структура за данни (релации)** – дефинира базата данни като колекция от релации, в които се съхраняват данните;
- **Правила за интегритет на данните** – множество от правила за валидност на данните, гарантиращи, че данните са консистентни във всеки момент;
- **Манипулативна част (релационна алгебра)** – множество от оператори за опериране с данните.



Основни понятия



Използвани термини

Релационен модел	Потребител
Релация	Таблица
Подредено множество (tuple)	Ред (запис)
Атрибут	Колона

Думата „релация“ се базира на математическата теория на множествата, използвана от Код за създаване на модела.

Тъй като релационният модел използва стойности на атрибути за представяне на взаимоотношения между таблици, често погрешно се смята, че „релация“ означава взаимоотношение.



Релационна таблица

Характеристики:

- » Таблицата се възприема като двумерна структура, съставена от редове и колони;
- » Всеки ред съдържа една инстанция на обекта, за който таблицата съхранява данни;
- » Всяка колона представя атрибут на обекта и има уникално име;
- » Сечението на ред с колона трябва да съдържа атомарна (скаларна, единична) стойност;
- » Всички стойности в една колона трябва да са от един и същ тип;
- » Подредбата на редовете и колоните в таблицата се смята за несъществена (те са множества, където наредба няма);
- » Всяка таблица трябва да има атрибут или комбинация от атрибути, идентифициращи уникално всеки ред.



Област (domain)

Най-малката семантична единица данни (наричат се още скаларни) в релационния модел е индивидуална стойност на данни – напр. ISBN номер на книга, факултетен номер на студент, ЕГН на гражданин.

От гледна точка на релационния модел тези стойности са неделими (atomic), защото ако (дори и да е възможно) бъдат декомпозирани на съставните им елементи, те ще загубят своя смисъл и информация, които носят.

Област - именувано множество от скаларни стойности, всички от един и същ тип.

Областта е множеството от всички възможни стойности, които дефинираните върху нея атрибути могат да имат.

Пример: областта на годините, в които са публикувани книги в библиотека – множество от цели числа между 1600 и 2019, т.е. в този диапазон стойността би имала смисъл, извън него годината на публикуване би била нереална.



Съставни области

Една съставна област се дефинира като Декартово произведение на множество от прости области.

Пример: съставната област DATE може да бъде дефинирана като Декартовото произведение на простите области DAY, MONTH, YEAR.

- ✓ Атрибут, дефиниран върху DATE областта, може да има 3 720 000 ($31 \cdot 12 \cdot 10000$) стойности;
- ✓ Не всички са валидни!

```
DAY    = ( 1, 2, ..., 31 )
MONTH  = ( 1, 2, ..., 12 )
YEAR   = ( 0, 1, ..., 9999 )

DATE   = ( 01 01 0001
           02 01 0001
           ...
           31 01 0001
           ...
           31 12 9999 )
```



Значение на областите

- » Ако атрибути от (евентуално) различни релации са дефинирани върху една и съща област, тогава сравнението им има значение;
- » В противен случай то е безсмислено и системата трябва да не го позволява. Например:

продукт.тегло > продукт.цена



Релация

» Една релация R върху множество от области D_1, D_2, \dots, D_n (не е задължително $D_i \neq D_j$) се състои от две части:

> заглавна част

> тяло



Релация

» **Заглавна част (heading)**: състои се от фиксирано множество от атрибути, които са подредени двойки във вид:

(име_атрибут : име_област),

т.е. $\{A_1 : D_1, A_2 : D_2, \dots, A_n : D_n\}$

такива, че всеки атрибут A_i е дефиниран върху точно един домейн D_i и всички A_i са различни ($i=1,2,\dots,n$).



Релация

» Тяло (body):

- > Състои се от множество от n-торки (наредени множества);
- > Всяка n-торка се състои от едно множество от двойки във вид:

(име_атрибут : стойност_атрибут)

т.е. $\{A_1 : v_{i1}, A_2 : v_{i2}, \dots, A_n : v_{in}\},$
 $i = 1, 2, \dots, m, v_i \in D_i$

n - степен на релацията;

m - мощност (кардиналност) на релацията.



Релация - пример

D_EMPID	D_NAME	D_DEPT_ID	D_SALARY
EMP_ID	NAME	DEPT_ID	SALARY
1	Петров	COMP_SYS	1350
2	Колева	SOFT_TECH	1800
3	Илиева	COMP_SYS	2240

Да проверим таблицата от този пример дали се вмести в дефиницията за релация.

1. Таблицата има 4 прилежащи домейна, върху които са дефинирани атрибутите ѝ.
2. Таблицата определено има две части – ред със имената на атрибутите и множество от редове с данни.



Релация - пример

Редът със заглавната част е представен във вид:

```
EMP_ID : D_EMP_ID ,  
NAME   : D_NAME ,  
DEPT_ID : D_DEPT_ID ,  
SALARY : D_SALARY
```

където първият компонент е атрибутът, а вторият е прилежащият домейн, върху който е дефиниран. По този начин можем да приемем, че това е представяне на заглавна част според дефиницията.



Релация - пример

3. Останалата част от таблицата определено се състои от множество от редове. Нека разгледаме един от тях:

(1, Петров, COMP_SYS, 1350)

Този ред представя следното множество от наредени двойки:

```
{ EMP_ID : 1,  
  NAME   : Петров,  
  DEPT_ID: COMP_SYS,  
  SALARY : 1350 }
```



Свойства на релациите

» Няма дублирани записи

Това следва от факта, че тялото на релацията е математическо множество, което по дефиниция не включва повтарящи се елементи.

Една таблица може да съдържа дублиращи се редове (ако е така конструирана), макар че такава ситуация е нежелателна, докато релацията не може да съдържа дублиращи се n -торки.



Свойства на релациите

» Записите са неподредени

Това също следва от факта, че тялото е математическо множество, в което елементите нямат подредба.

Затова не се говори за последователност при редовете или за позиционно адресиране.



Свойства на релациите

» Атрибутите са неподредени

Това свойство следва от факта, че заглавната част е математическо множество, в което не съществува подредба.

Така че отново и тук не съществува понятието последователност на атрибутите. Те обикновено се цитират по име, а не по позиция.



Свойства на релациите

» Стойностите на атрибутите са атомарни

Това свойство идва от факта, че стойностите на прилежащите домейни са атомарни. Т.е. всяко поле съдържа точно една стойност, а не колекция \Rightarrow релациите не съдържат повтарящи се групи от данни.



Някои видове релации

- » **Базови релации** – именувани релации, които са постоянно съхранени в базата данни – това са основните обекти, в които се съхраняват данните;
- » **Изгледи (views)** – именувани виртуални релации, базирани на базови релации. За тях се съхраняват само дефинициите им без реално да имат данни. Техните данни идват от релациите, върху които са дефинирани. Представяват логическа (концептуална) интерпретация на данните от базовите релации;



Някои видове релации

- » **Snapshots (моментни снимки)** - именувани дериватни релации, подобни на изгледите, но са реални, а не виртуални, т.е. имат собствени данни (materialized views). Те съдържат данни от момента на тяхното последно обновяване/създаване, като обикновено данните им се обновяват периодично;
- » **query results (резултати от заявки)** - неименувани дериватни релации, които се получават при изпълнение на заявка.

