

Релационен модел на данните(тема 23)

Често пъти избираме релационния модел на представяне на данните, заради:

- простота му
- няма нужда да знаем вътрешното представяне на данните, ние ползваме абстрактното им релационно представяне
- съответства на мисленето на човека
- еднообразен начин за представяне на данните и връзките м/у тях във вид на таблица

Основни понятия

def **Релация**(*лаутан деф.*)

Релация представлява двумерна таблица, в която съхраняваме данните

Характеристики на релациите

0. Атрибути == имената на колоните на таблицата на релацията

1. Домейни

С всеки атрибут на релацията е свързан домейн, т.е. мн-во от допустими стойности.

Пример

```
Movies(title: string, year: integer, length: integer, filmType: string)
```

- всеки домейн на атрибут, т.е. $dom(A_i)$ е от **атомарен** тип, напр. integer или string
 - съставни типове като списъци и масиви не се позволяват

2. Схема(на релация) == името на релацията и множеството от атрибутите ѝ

- напр. Movies(title, year, length, filmType)

! забл. атрибутите в релационната схема са множество, но за определеност указваме наредба

3. Кортежи == редовете в таблицата(без заглавната част, разбира се)

def **База данни**

Множество от релации.

def **Схема на БД**

Множеството от всички схеми на релации в една БД наричаме схема на БД.

def **Релация(формална дефиниция)**

R е релация с релационна схема **R(A₁, A₂, ..., A_n)**. Тогава **r(R)** е математическа релация от степен **n** върху домейните **dom(A₁), ..., dom(A_n)**.

Т.е. **r** \subset **dom(A₁)** \times ... \times **dom(A_n)**.

Така **r** или **r(R)** е множество от n-tuples(кортежи):

r = {**t₁**, ..., **t_m**},

t_i = <**v₁**, ..., **v_n**> ,

v_j \in **dom(A_j)** или е NULL.

def **Екземпляр на релация(current instance)**

Множеството от кортежите за дадена релация наричаме екземпляр на релацията.

- **докато екземплярите са статични множества кортежи, релациите са нестатични(т.е. в някакъв смисъл променливи), те съхраняват текущ екземпляр**
 - т.е. можем да извършваме 2 типа промени:
 - insert, update, deletion of tuples
 - промени в схемата - по-рядко се случва

def **Кардиналност на екземпляр на релация**

Броят на кортежите в текущия екземпляр, т.е. |**r**|

Реализация на релационната база от данни

Обикновено първо проектираме E/R модел на БД, на базата на който определяме същността на данните и връзките между обектите, които ще съхраняваме, след което правим преход към релационното представяне на данните, като се опитваме да премахнем недостатъци и аномалии и на двете нива.

1. Определяне на данните, които ще се съхраняват
2. Определяне как ще представим обектите, т.е. атрибутите и същностите
3. Определяне връзките м/у обектите
4. Ограничения

- ключови атрибути, референциален интегритет, ограничения на домейните
- други ограничения в/у обектите

5. Отстраняване на недостатъци, аномалии, които могат да нарушат съгласуваността или напр. да доведат до допълнителен разход на памет

6. Реализация на БД

Видове операции върху релационната база от данни. Заявки към релационната БД

💡 Всяка **заявка** към базата данни се представя под формата на **израз** и тъй като говорим за релационна БД, този израз **съдържа операции от релационната алгебра**.

При релационната алгебра обектите, върху които са дефинирани операциите(операторите) по-долу, разбира се, са релации.

SQL е език за заявки към БД, но интерпретацията на заявките, изразени със синтаксиса на SQL, съвпада семантично именно с операторите на релационната алгебра.

! **забл.** при обработка на SQL израз/заявка, той може да бъде опростен и оптимизиран, което се извършва от query optimizer; т.е. транслирането не е гарантирано директно

Релационна алгебра

Нека имаме релациите:

- **R**

name	address	gender	birthdate
Carrie Fisher	123 Maple, Hollywood	f	9/9/99
Mark Hamil	456 Oak Rd. Brentwood	M	8/8/88

- **S**

name	address	gender	birthdate
Carrie Fisher	123 Maple, Hollywood	f	9/9/99
Harrison Ford	789 Palm Dr. Beverly Hills	M	7/7/77

Основни оператори

Основните оператори представляват базис за останалите оператори на релационната алгебра.

! забл. В зависимост от това дали разглеждаме релациите като множества или мултимножества, може да получим различни резултати при оценката на заявка, тъй като при ММ елементите имат кратност.

! забл. Теоретико-множествените оператори на релационната алгебра (\cup , $-$, \cap) са дефинирани върху релации, чиито схеми са съвместими.

def Нека $R(A_1, \dots, A_n)$ и $S(B_1, \dots, B_m)$. Казваме, че R и S са съвместими $\iff n = m \wedge (\forall i \in \{1, \dots, n\})[dom(A_i) = dom(B_i)]$

- Обединение - бинарна, комутативна и асоциативна операция

- бел.** $R \cup S$

напр.:

name	address	gender	birthdate
Carrie Fisher	123 Maple, Hollywood	f	9/9/99
Mark Hamil	456 Oak Rd. Brentwood	M	8/8/88
Harrison Ford	789 Palm Dr. Beverly Hills	M	7/7/77

- Разлика - бинарна операция

- $R - S$

name	address	gender	birthdate
Mark Hamil	456 Oak Rd. Brentwood	M	8/8/88

- Проекция - унарна операция;

- бел.** $\pi_{attrlst}(R)$, където $attrlst$ е списък от атрибути

Проекцията представлява хоризонтална рестрикция.

напр.:

$\pi_{name, birthdate}(R \cup S)$ като резултат дава:

name	birthdate
Carrie Fisher	9/9/99
Mark Hamil	8/8/88
Harrison Ford	7/7/77

- Селекция - унарна операция;
 - бел.** $\sigma_{predicate}(R)$, където *predicate* е условен израз
 - условният израз се състои от операции от вида:
 - <attribute> <op> <attribute>
 - <attribute> <op> <constant>
 - <contant> <op> <attribute>където $attribute \in R$ и $op \in \{=, \neq, <, >, \dots, \text{AND}, \text{OR}\}$
 - ??комутатативна - сигурно в някакъв тривиален смисъл е вярно

Селекцията представлява вертикална рестрикция.

напр.:

$\sigma_{gender='m'}(R \cup S)$ като резултат дава:

name	address	gender	birthdate
Mark Hamil	456 Oak Rd. Brentwood	M	8/8/88
Harrison Ford	789 Palm Dr. Beverly Hills	M	7/7/77

- Преименуване
 - бел.** $R1 := \rho_{R1(A_1, \dots, A_n)}(R)$
Операторът ρ дава нова схема на релацията.
- Декартово произведение - бинарна, комутативна, асоциативна операция
 - бел.** $R \times S$
 - Резултатната таблица има за редове всички кортежи, които се получават като конкатенация на кортеж от R с кортеж от S .
 - Схемата на резултата е обединение на схемите на R и S .

! забл. Схемата на резултата може да съдържа еднакви имена на атрибути. Затова може да префиксираме имената на едноименните атрибути, напр. $R.A$ или $S.A$

напр.:

Нека имаме релациите R и S .

- R

A	B
1	2
3	4

- S

B	C	D
2	5	6
4	7	8
9	10	11

Тогава $R \times S$ дава:

A	R.B	S.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

Допълнителни оператори

- сечение
 - $R \cap S$
 - $R \cap S = R - (R - S)$

- частно(не сме го споменавали на лекции)
 - обратната операция на \times

Нека $R(A_1, \dots, A_k, B_1, \dots, B_m)$ и $S(B_1, \dots, B_m)$
 $Q(A_1, \dots, A_k) = R \div S \iff Q \times S = R.$

- съединение, θ join
бел. $R \bowtie_{\theta} S$
 - изпълнение
 - $\sigma_{\theta}(R \times S)$
 - схемата на резултата е обединение на схемите на R и S
- екви-съединение
 - theta join, при което join condition включва само съвпадение по атрибутите
- естествено съединение, natural join
 - нека едноименните атрибути на R и S да са множеството C
 - $R \bowtie S := R \bowtie_{R.C=S.C} S$

напр. нека имаме релацията Movies:

title	year	length	filmType	studioName	starName
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

Тогава следната заявка би ни дала адресите на всички актьори(за които имаме информация), които са играли във филма **Star Wars**:

нека $ActorInfo := R \cup S$

$\pi_{address}(ActorInfo \bowtie_{starName=name} Movies)$

! забл. Т.к. представихме релациите като множества, повторения няма да има; ако работим с ММ предствяне, можем да приложим операторът δ , за да премахнем такива.

Приоритет на операторите

1. Унарни оператори - селекция, проекция и преименуване

2. Декартово произведение и съединение

3. Сечение

4. Обединение и разлика