

Databáze ♦ poznámky k přednášce

1. Relace

verze z 18. září 2023

1 Relace a relační proměnné

Představme si, že chceme uchovávat informace o filmech, které vlastníme. Naše právě vznikající kolekce obsahuje pouhé tři filmy: The Avengers (1998), The Avengers (2012) a The Matrix (1999). Číslo v závorce uvedené za názvem filmu je rokem vydání filmu.

Vhodné řetězce písmen anglické abecedy a podtržítka budeme nazývat **atributy**. Například `title`, `year` nebo `movie_id` jsou atributy. Množinu všech atributů označíme \mathcal{Y} .

Každému atributu $y \in \mathcal{Y}$ je přiřazena spočetná množina D_y nazývaná jeho **doména**. Například D_{title} může být množina všech řetězců nad anglickou abecedou a D_{year} množina všech přirozených čísel.

Konečná podmnožina množiny atributů \mathcal{Y} se nazývá **relační schéma**. Například $\{\text{title}, \text{year}\}$ je relační schéma.

Mějme relační schéma R . Zobrazení $t: R \rightarrow \bigcup_{y \in R} D_y$, kde pro každé $y \in R$ je $t(y) \in D_y$, se nazývá **n -tice** nad R . Například $\{\langle \text{title}, \text{'The Matrix'} \rangle, \langle \text{year}, 1999 \rangle\}$ je n -tice nad $\{\text{title}, \text{year}\}$. Množinu všech n -tic nad relačním schématem R značíme $\text{Tupl}(R)$.

Konečná podmnožina množiny $\text{Tupl}(R)$ se nazývá **relace** nad R . Například

$$\begin{aligned} & \{ \langle \text{title}, \text{'The Avengers'} \rangle, \langle \text{year}, 1998 \rangle \}, \\ & \{ \langle \text{title}, \text{'The Avengers'} \rangle, \langle \text{year}, 2012 \rangle \}, \\ & \{ \langle \text{title}, \text{'The Matrix'} \rangle, \langle \text{year}, 1999 \rangle \} \end{aligned}$$

je relace nad $\{\text{title}, \text{year}\}$.

Relaci můžeme přehledně zadat tabulkou. Například předchozí relaci můžeme zadat tabulkou:

title	year
The Avengers	1998
The Avengers	2012
The Matrix	1999

Všimněte si, že tabulka oproti relaci navíc určuje pořadí n -tic a atributů. Tedy i tato tabulka určuje stejnou relaci:

year	title
1999	The Matrix
1998	The Avengers
2012	The Avengers

Prázdnou relací nad R rozumíme prázdnou množinu.

Nechť \mathcal{D} je relace nad R a $V(t)$ je výroková forma, kde proměnná t má obor hodnot $\text{Tupl}(R)$. Pak $V(t)$ se nazývá **charakteristická vlastnost** \mathcal{D} (zkráceně jen **vlastnost** určující \mathcal{D}), jestliže platí

$$t \in \mathcal{D}, \text{ právě když } V(t) \text{ je pravdivé;}$$

což lze zkráceně zapsat jako $\mathcal{D} = \{t \in \text{Tupl}(R) \mid V(t)\}$.

Například výroková forma „Vlastním film $t(\text{title})$ vytvořený roku $t(\text{year})$.“ je charakteristickou vlastností relace

title	year
The Avengers	1998
The Avengers	2012
The Matrix	1999

Charakteristické vlastnosti relací budeme zjednodušovat vynecháním proměnné t . Tedy předchozí vlastnost je zkráceně: „Vlastním film **title** vytvořený roku **year**.“

Databáze vždy zachycuje jen omezenou část reálného světa. Například udržujeme informace jen o vlastněných filmech. Pokud jasné určíme část světa, která nás zajímá, můžeme charakteristické vlastnosti relací zjednodušit.

Například místo:

„Vlastním film **title** vydaný roku **year**.“

uvedeme jen:

„Film **title** byl vydaný roku **year**.“

K označení prvku reálného světa používáme **identifikátor** (zkráceně **id**). Přičemž identifikátor musí prvek jednoznačně určovat. Jako identifikátor prvku se často používají přirozená čísla nebo náhodné řetězce. Pro účely databáze se prvek ztotožňuje se svým identifikátorem.

Například vlastnost „Film **movie_id** má název **title** a byl vydaný roku **year**.“ určuje relaci:

movie_id	title	year
1	The Avengers	1998
2	The Avengers	2012
3	The Matrix	1999

Relace modelují jak entity reálného světa (například filmy nebo herce), tak vztahy mezi nimi (například role herce ve filmu).

Například relace

actor_name	movie_title	movie_year
Keano Reeves	The Matrix	1999
Keano Reeves	Dracula	1992
Laurence Fishburne	The Matrix	1999
Gary Oldman	Dracula	1992
Anthony Hopkins	The Elephant Man	1980

určená vlastností

„Zajímá mě herec `actor_name`, který hrál ve mnou vlastněném filmu `movie_title` z roku `movie_year`.“

zachycuje vztah mezi entitami. Vlastnost lze zkrátit na: „Herec `actor_name` hrál ve filmu `movie_title` z roku `movie_year`.“

Relační proměnná je proměnná, jejíž hodnota je relace. **Typem** relační proměnné je relační schéma. Hodnotou relační proměnné může být pouze relace nad tímto schématem. K relační proměnné přísluší výroková forma, která je vždy charakteristickou vlastností její hodnoty. **Základní relace** je relační proměnná, jejíž hodnotu lze přímo měnit. Více o přímé změně hodnoty proměnné se dozvíte v některé z následujících přednášek.

Například základní relace `movie` nad `{title, year}`, určená vlastností „Film `title` je z roku `year`.“ má hodnotu:

title	year
The Avengers	1998
The Avengers	2012
The Matrix	1999

2 SQL

SQL tabulky jsou podobné relacím, ale liší se od nich tím, že určují pořadí sloupců i řádků, názvy sloupců nemusí být jedinečné, mohou obsahovat duplicitní řádky a buňka tabulky nemusí obsahovat hodnotu.

Jako relaci můžeme chápat SQL tabulku, kde

- názvy sloupců jsou jedinečné,
- nejsou duplicitní řádky,
- se nespolehneme na pořadí řádků,

- všechny buňky mají hodnotu.

Jiné tabulky používat nebudeme.

Skalární typ dává jméno množině, která může být doménou atributů. Například `integer` je skalárním typem, který dává jméno množině celých čísel od $-2\,147\,483\,648$ do $2\,147\,483\,647$. Skalární typ `text`¹ pojmenovává množinu všech řetězců. Další skalární typy PostgreSQL můžete najít v dokumentaci. Atribut (sloupec) SQL tabulky má doménu určenou skalárním typem.

Přijmeme následující omezení. Stejné atributy v různých SQL tabulkách musí být stejného skalárního typu.

Skalární typ atributů můžeme zobrazit v tabulce:

title text	year integer
The Avengers	1998
The Avengers	2012
The Matrix	1999

Příkaz

```
CREATE TABLE relation (
    attribute1 scalar_type1,
    attribute2 scalar_type2,
    :
);
```

deklaruje základní relaci *relation* nad $\{attribute1, attribute2, \dots\}$. Doména atributu *attributei* je dána skalárním typem *scalar_typei*. Hodnotou je prázdná relace.

Příklad:

```
CREATE TABLE movie (
    title text,
    year integer,
);
```

Příkaz

```
TABLE relation;
```

zobrazí hodnotu relační proměnné *relation*.

Příklad:

¹Není ve standardu SQL.

```
# TABLE movie;
title | year
-----+-----
(0 rows)
```

Následující příkaz slouží k přidání n -tic do relační proměnné. Necht *relation* je relační proměnná nad $R = \{y_1, \dots, y_n\}$ a hodnotou \mathcal{D} a t_1, \dots, t_m jsou n -tice nad R takové, že množiny \mathcal{D} a $\{t_1, \dots, t_m\}$ jsou disjunktní. Předpokládáme pořadí atributů y_1, \dots, y_n dané při deklaraci proměnné.

Příkaz

```
INSERT INTO relation VALUES
    (t1(y1), ..., t1(yn)),
    ⋮
    (tm(y1), ..., tm(yn));
```

nastaví hodnotu proměnné *relation* na $\mathcal{D} \cup \{t_1, \dots, t_m\}$

Například po vykonání

```
# INSERT INTO movie VALUES
    ( 'The Matrix', 1999 ),
    ( 'The Avengers', 2012 ),
    ( 'The Avengers', 1998 );
```

se změní hodnota proměnné na

```
# TABLE movie;
    title    | year
-----+-----
The Matrix   | 1999
The Avengers | 2012
The Avengers | 1998
(3 rows)
```

Pokud dále vykonáme příkaz:

```
INSERT INTO movie VALUES ('A Space Odyssey', 1968);
```

Proměnná získá hodnotu:

```
# TABLE movie;
      title      | year
-----+-----
The Matrix       | 1999
The Avengers     | 2012
The Avengers     | 1998
A Space Odyssey  | 1968
(4 rows)
```

Připomínám, že na pořadí n -tic se nelze spolehnout.

Příkaz

```
DROP TABLE relation;
```

zruší základní relaci *relation*.

Příklad:

```
DROP TABLE movie;
```