

5. Kontrola integrity

verze z 22. října 2024

1 Primární klíč

Je dáno relační schéma S a podmnožina $K \subseteq S$.

Řekneme, že dvě n -tice t_1, t_2 nad S se **shodují** v K , pokud $t_1(K) = t_2(K)$.
Například n -tice:

- $\{\langle \text{title}, 'The Avengers' \rangle, \langle \text{year}, 2012 \rangle, \langle \text{length}, 143 \rangle\}$,
- $\{\langle \text{title}, 'The Avengers' \rangle, \langle \text{year}, 1998 \rangle, \langle \text{length}, 89 \rangle\}$

nad $\{\text{title}, \text{year}, \text{length}\}$ se shodují v $\{\text{title}\}$, ale neshodují v $\{\text{title}, \text{year}\}$.

Vezměme relaci R nad S . Schéma K nezveme **nadklíčem** relace R , pokud pro každé dvě n -tice $t_1, t_2 \in R$ platí, že jestliže se t_1 shoduje s t_2 v K , pak $t_1 = t_2$. Schéma S je vždy nadklíčem relace R . Například v relaci R :

title	year	length
The Matrix	1999	136
The Avengers	2012	143
The Avengers	1998	89
American Beauty	1999	122
Listen to Britain	1942	20
Khaneh siah ast	1963	20

(6 rows)

jsou $\{\text{title}, \text{year}\}$ a $\{\text{title}, \text{year}, \text{length}\}$ nadklíče, ale například $\{\text{title}\}$ nadklíčem není, protože n -tice:

- $t_1 = \{\langle \text{title}, 'The Avengers' \rangle, \langle \text{year}, 2012 \rangle, \langle \text{length}, 143 \rangle\}$,
- $t_2 = \{\langle \text{title}, 'The Avengers' \rangle, \langle \text{year}, 1998 \rangle, \langle \text{length}, 89 \rangle\}$

z R se shodují v $\{\text{title}\}$, ale $t_1 \neq t_2$.

Pro každou základní relaci *relation* uvažujeme množinu možných hodnot. Možné hodnoty jsou relace, které mohou být hodnotou relační proměnné *relation* — v současnosti nebo v budoucnosti mají charakteristickou vlastnost základní

relace *relation*. Pokud je K nadklíčem každé možné hodnoty základní relace *relation*, pak říkáme, že K je **nadklíčem** základní relace *relation*. Tedy například $\{\text{title}, \text{year}\}$ je nadklíčem základní relace *movie* nad $\{\text{title}, \text{year}, \text{length}\}$ dané: „Film *title* vydaný roku *year* je dlouhý *length* minut.“, kde předpokládáme, že každý film lze jednoznačně určit názvem a rokem vydání.

Nadklíč K základní relace *relation* se nazývá jejím **kandidátním klíčem**, pokud žádná vlastní podmnožina K není nadklíčem základní relace *relation*. Například u základní relace *movie* je $\{\text{title}, \text{year}\}$ kandidátní klíč, ale $\{\text{title}, \text{year}, \text{length}\}$ kandidátním klíčem není.

Pokud napíšeme, že y_1, \dots, y_n je kandidátním klíčem základní relace *relation*, myslíme tím, že $\{y_1, \dots, y_n\}$ je kandidátním klíčem základní relace *relation* a navíc udáváme pořadí atributů y_1, \dots, y_n v kandidátním klíči.

Základní relace může mít více kandidátních klíčů. Například základní relace *movie* nad $\{\text{movie_id}, \text{title}, \text{year}, \text{length}\}$ daná vlastností: „Vlastním film *movie_id* s názvem *title* vytvořený roku *year*, který má délku *length* minut.“ má dva kandidátní klíče: $\{\text{movie_id}\}$ a $\{\text{title}, \text{year}\}$.

Pokud nevíte proč, podívejte se na tuto možnou hodnotu:

movie_id	title	year	length
1	The Matrix	1999	136
2	The Avengers	2012	143
3	The Avengers	1998	89
4	American Beauty	1999	122
5	Listen to Britain	1942	20
6	Khaneh siah ast	1963	20

(6 rows)

Jeden z kandidátních klíčů základní relace vybereme a prohlásíme ho za **primární klíč**. Zbylé klíče se nazývají **alternativní klíče**. Například $\{\text{movie_id}\}$ může být zvolen primárním klíčem základní relace *movie* a $\{\text{title}, \text{year}\}$ klíčem alternativním.

1.1 Vyjádření nadklíče relačními operacemi

Následující věta charakterizuje nadklíč relace pomocí jistého zobrazení.

Věta 1. *Schéma K je nadklíčem relace R nad S , právě když $\{\langle t(K), t \rangle \mid t \in R\}$ je zobrazení $\pi_K(R)$ do R .*

Důkaz: Označme si množinu $\{\langle t(K), t \rangle \mid t \in R\}$ jako F .

Nejprve předpokládejme, že F je zobrazení množiny $\pi_K(R)$ do R a vezměme libovolné n -tice t_1, t_2 z R , které se shodují v K , což znamená, že $t_1(K) = t_2(K)$. Protože $t_1(K)$ i $t_2(K)$ jsou z $\pi_K(R)$, můžeme na ně použít zobrazení F a dostáváme

$t_1 = F(t_1(K)) = F(t_2(K)) = t_2$. Tedy t_1 a t_2 se rovnají. Ukázali jsme, že množina K je nadklíčem relace R .

Opačně předpokládejme, že K je nadklíčem R . Zvolme libovolné $t_1 \in \pi_K(R)$. Z definice projekce vyplývá, že musí existovat $t_2 \in R$ tak, že $t_2(K) = t_1$. Proto $\langle t_1, t_2 \rangle \in F$. Dále vezměme $\langle t_1, t_2 \rangle, \langle t_1, t'_2 \rangle \in F$. Z definice relace F vyplývá, že $t_2(K) = t_1 = t'_2(K)$. Tedy n -tice t_2 a t'_2 se shodují v K . Protože K je nadklíčem R , musí se n -tice t_2 a t'_2 rovnat. Ukázali jsme, že F je zobrazení množiny $\pi_K(R)$ do R .

□

Následující věta charakterizuje nadklíč relace pomocí relačních operací.

Věta 2. *Uvažujme relaci R nad S a zvolme relační schéma S' disjunktní s $S = \{y_1, \dots, y_n\}$, přejmenování atributů $h : S \rightarrow S'$ a $K = \{y_1, \dots, y_m\} \subseteq S$; tedy $m \leq n$. Položme θ rovno podmínce*

$$(y_1 = h(y_1)) \wedge \dots \wedge (y_m = h(y_m)) \wedge (\neg(y_{m+1} = h(y_{m+1})) \vee \dots \vee \neg(y_n = h(y_n))).$$

Pak platí, že K je nadklíčem R , právě když $\sigma_\theta(R \times \rho_h(R)) = \emptyset$.

Důkaz: Předpokládejme nejprve, že schéma K je nadklíč relace R . Dále předpokládejme, že existuje $t \in \sigma_\theta(R \times \rho_h(R))$. Položme $t_1 = t(S)$ a $t_2 = \rho_{h^{-1}}(t(S'))$. Obě n -tice t_1, t_2 jsou nad S . Z restrikce podle podmínky θ plyne, že $t(y_i) = t(h(y_i))$ pro každé $1 \leq i \leq m$ a že existuje číslo j tak, $m+1 \leq j \leq n$ a $t(y_j) = t(h(y_j))$. Z první části dostáváme, že t_1 a t_2 se shodují v K a protože K je nadklíč R , dostáváme, že t_1 se rovná t_2 . Z druhé části ale vyplývá, že t_1 je různé od t_2 , což je spor. Množina $\sigma_\theta(R \times \rho_h(R))$ tedy musí být prázdná.

Na druhou stranu předpokládejme, že $\sigma_\theta(R \times \rho_h(R)) = \emptyset$ a vezměme si t_1, t_2 z R , které se shodují v K . Položme t rovno $t_1 \rho_h(t_2)$. Jistě $t \in R \times \rho_h(R)$, ale $t \notin \sigma_\theta(R \times \rho_h(R))$. Tedy t nesplňuje podmínku θ . Protože t splňuje podmínku $(y_1 = h(y_1)) \wedge \dots \wedge (y_m = h(y_m))$ musí nespĺňovat podmínku $\neg(y_{m+1} = h(y_{m+1})) \vee \dots \vee \neg(y_n = h(y_n))$. To znamená, že splňuje její negaci $(y_{m+1} = h(y_{m+1})) \wedge \dots \wedge (y_n = h(y_n))$. Což znamená, že $t(y_i) = t(h(y_i))$ pro každé $1 \leq i \leq n$. Dostáváme, že t_1 se rovná t_2 . Ukázali jsme, že K je nadklíčem relace R .

□

Například v relaci:

TABLE movie;

title	year	length
The Matrix	1999	136
The Avengers	2012	143
The Avengers	1998	89
American Beauty	1999	122

(4 rows)

je {title, year} nadklíč:

```
# SELECT m1.title AS title1,
        m1.year AS year1,
        m1.length AS length1,
        m2.title AS title2,
        m2.year AS year2,
        m2.length AS length2
FROM   movie AS m1,
        movie AS m2
WHERE  m1.title = m2.title
AND    m1.year = m2.year
AND NOT ( m1.length = m2.length );
```

title1	year1	length1	title2	year2	length2
(0 rows)					

ale {title} nadklíčem není:

```
# SELECT m1.title AS title1,
        m1.year AS year1,
        m1.length AS length1,
        m2.title AS title2,
        m2.year AS year2,
        m2.length AS length2
FROM   movie AS m1,
        movie AS m2
WHERE  m1.title = m2.title
AND ( NOT ( m1.length = m2.length )
OR NOT ( m1.year = m2.year ) );
```

title1	year1	length1	title2	year2	length2
The Avengers	1998	89	The Avengers	2012	143
The Avengers	2012	143	The Avengers	1998	89
(2 rows)					

2 Omezení základní relace

Některé hodnoty základní relace můžeme vyloučit tak, že u základní relace uvedeme **omezení**. Omezení základní relace můžeme uvést při její deklaraci:

```
CREATE TABLE relation (
    attributes,
    constraints
);
```

kde *constraints* jsou omezení základní relace oddělená čárkou.

Omezení základní relace:

```
PRIMARY KEY (  $y_1, \dots, y_n$  )
```

deklaruje, že y_1, \dots, y_n je primárním klíčem základní relace. Omezení základní relace:

```
UNIQUE (  $y_1, \dots, y_n$  )
```

deklaruje, že y_1, \dots, y_n je alternativním klíčem základní relace. Například:

```
CREATE TABLE movie (  
    movie_id integer,  
    title text,  
    year integer,  
    length integer,  
    PRIMARY KEY ( movie_id ),  
    UNIQUE ( title, year )  
);
```

deklaruje, že {movie_id} je primárním a {title,year} alternativním klíčem základní relace movie. Do proměnné:

```
# TABLE movie;
```

```
movie_id | title | year | length  
-----+-----+-----+-----  
1 | The Matrix | 1999 | 136  
2 | The Avengers | 2012 | 143  
(2 rows)
```

nemůžeme přidat:

```
# INSERT INTO movie VALUES  
    ( 2, 'The Avengers', 1998, 89 );
```

```
ERROR: duplicate key value violates unique constraint "movie_pkey"  
DETAIL: Key (movie_id)=(2) already exists.
```

ani:

```
# INSERT INTO movie VALUES  
    ( 3, 'The Avengers', 2012, 89 );
```

```
ERROR: duplicate key value violates unique constraint "movie_title_year_key"  
DETAIL: Key (title, year)=(The Avengers, 2012) already exists.
```

Pokud se omezení týká jediného atributu, můžeme jej uvést rovnou v deklaraci atributu:

```
y scalar_type constraints
```

Omezení atributu píšeme za sebe a oddělujeme mezerou. Omezení atributu PRIMARY KEY prohlásí {y} za primární klíč a UNIQUE za alternativní klíč. Například:

```
CREATE TABLE movie (  
    movie_id integer PRIMARY KEY,  
    title text,  
    year integer,  
    length integer,  
    UNIQUE ( title, year )  
);
```

Jak bylo dříve zmíněno, řádek tabulky SQL pro zadaný sloupec nemusí obsahovat hodnotu. Říkáme pak, že obsahuje **null** hodnotu. V SQL ji zapisujeme NULL. Připomeňme si, že každá komponenta n -tice v relaci má vždy hodnotu. Omezení atributu NOT NULL zaručí, že ve sloupci nikdy nebude null hodnota. Například:

```
CREATE TABLE movie (  
    movie_id integer PRIMARY KEY,  
    title text NOT NULL,  
    year integer NOT NULL,  
    length integer,  
    UNIQUE ( title, year )  
);
```

Nelze přidat null hodnotu do sloupce title:

```
# INSERT INTO movie VALUES  
    ( 1, NULL, 1999, 136 ),  
    ( 2, 'The Avengers', 2012, 143 );
```

```
ERROR: null value in column "title" of relation "movie"  
        violates not-null constraint  
DETAIL: Failing row contains (1, null, 1999, 136).
```

V žádném sloupci uvedeném v primárním klíči nesmí být null hodnoty. Proto nelze:

```
# INSERT INTO movie VALUES  
    ( NULL, 'The Avengers', 1998, 89 );
```

```
ERROR: null value in column "movie_id" of relation "movie"  
        violates not-null constraint
```

Omezení PRIMARY KEY je ekvivalentní kombinaci omezení NOT NULL a UNIQUE.

Pokud základní relace má deklarovaný primární klíč a každý sloupec má omezení, že nesmí obsahovat null hodnoty, pak je zaručeno, že její hodnota bude relace. Například:

```
CREATE TABLE movie (
  movie_id integer PRIMARY KEY,
  title text NOT NULL,
  year integer NOT NULL,
  length integer NOT NULL,
  UNIQUE ( title, year )
);
```

3 Cizí klíč

Vezměme dvě relace R_1 nad schématem S_1 a R_2 nad schématem S_2 . Dále vezměme relační schémata $K_1 \subseteq S_1$ a $K_2 \subseteq S_2$ a přejmenování atributů $h: K_1 \rightarrow K_2$. Schéma K_1 nazýváme **cizím klíčem** relace R_1 na K_2 relace R_2 podle h , pokud pro každou n -tici $t_1 \in R_1$ existuje právě jedna n -tice $t_2 \in R_2$ taková, že $\rho_h(t_1(K_1)) = t_2(K_2)$.

Například pro relaci R_1 :

actor_name	movie_title	movie_year
Keano Reeves	The Matrix	1999
Keano Reeves	Dracula	1992
Laurence Fishburne	The Matrix	1999
Gary Oldman	Dracula	1992

(4 rows)

a relaci R_2 :

title	year
The Matrix	1999
Dracula	1992

(2 rows)

je $\{\text{movie_title}, \text{movie_year}\}$ cizím klíčem relace R_1 na $\{\text{title}, \text{year}\}$ relace R_2 podle $\{\langle \text{movie_title}, \text{title} \rangle, \langle \text{movie_year}, \text{year} \rangle\}$.

Vezměme základní relace *relation1* nad S_1 a *relation2* nad S_2 , pak K_1 je **cizím klíčem** základní relace *relation1* na K_2 základní relace *relation2*, pokud pro libovolnou možnou dvojici hodnot R_1 a R_2 základních relací *relation1* a *relation2* je K_1 cizím klíčem R_1 na K_2 relace R_2 podle h .

Pokud napíšeme, že y_1, \dots, y_n je cizím klíčem základní relace *relation1* na z_1, \dots, z_n základní relace *relation2*, pak tím myslíme, že $\{y_1, \dots, y_n\}$ je cizí klíč základní relace *relation1* na $\{z_1, \dots, z_n\}$ základní relace *relation2* podle přejmenování $h = \{\langle y_1, z_1 \rangle, \dots, \langle y_n, z_n \rangle\}$. Například *movie_title*, *movie_year* je cizím klíčem základní relace *movie_cast* na *title*, *year* základní relace relace *movie*. Nevyjádřené h je zde:

$$\{\langle \text{movie_title}, \text{title} \rangle, \langle \text{movie_year}, \text{year} \rangle\}.$$

Pokud navíc z_1, \dots, z_n je primární klíč základní relace *relation2*, stačí říci pouze, že y_1, \dots, y_n je cizím klíčem základní relace *relation1* na *relation2*.

3.1 Cizí klíč v SQL

Omezení základní relace *relation1*:

```
FOREIGN KEY (  $y_1, \dots, y_n$  ) REFERENCES relation2 (  $z_1, \dots, z_n$  )
```

deklaruje, že y_1, \dots, y_n je cizím klíčem základní relace *relation1* na z_1, \dots, z_n základní relace *relation2*. Schéma $\{z_1, \dots, z_n\}$ musí být deklarováno jako kandidátní klíč základní relace *relation2*. Například:

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

```
CREATE TABLE movie_cast (
    actor_name text,
    movie_title text,
    movie_year integer,
    PRIMARY KEY ( actor_name, movie_title, movie_year ),
    FOREIGN KEY ( movie_title, movie_year ) REFERENCES movies ( title, year )
);
```

Pokud:

```
# TABLE movie;
```

```

title      | year
-----+-----
The Matrix | 1999
Dracula    | 1992
(2 rows)
```



```
# TABLE movie_cast;
```

actor_name	movie_title	movie_year
Keano Reeves	The Matrix	1999
Keano Reeves	Dracula	1992
Laurence Fishburne	The Matrix	1999
Gary Oldman	Dracula	1992

(4 rows)

pak nelze odstranit:

```
# DELETE FROM movie WHERE title = 'Dracula';
```

```
ERROR: update or delete on table "movie" violates
foreign key constraint "movie_cast_movie_fkey" on table "movie_cast"
DETAIL: Key (title, year)=(Dracula, 1992) is still referenced
from table "casting".
```

ani změnit:

```
# UPDATE movie
SET title = 'Drákula'
WHERE title = 'Dracula';
```

```
ERROR: update or delete on table "movie" violates
foreign key constraint "movie_cast_movie_fkey" on table "movie_cast"
DETAIL: Key (title, year)=(Dracula, 1992) is still referenced
from table "casting".
```

ani přidat:

```
# INSERT INTO movie_cast VALUES
( 'Gary Oldman', 'Batman Begins', 2005);
```

```
ERROR: insert or update on table "casting" violates
foreign key constraint "movie_cast_movie_fkey"
DETAIL: Key (movie_title, movie_year)=(Batman Begins, 2005)
is not present in table "movies".
```

Pokud by cizí klíč obsahoval jediný atribut, můžeme použít omezení atributu y_1 :

```
REFERENCES relation2 (  $z_1$  )
```

Například:

```
CREATE TABLE movie (
    movie_id integer PRIMARY KEY,
    title text NOT NULL,
    year integer NOT NULL,
    UNIQUE ( title, year )
);
```

```
CREATE TABLE movie_cast (
    actor_name text,
    movie_id integer REFERENCES movies ( movie_id ),
    PRIMARY KEY ( actor_name, movie_id )
);
```

Pokud vynecháme odkazované atributy, doplní se primární klíč relace. Tedy

```
REFERENCES movie
```

znamená:

```
REFERENCES movie ( movie_id )
```

a podobně u dřívějšího příkladu omezení tabulky:

```
FOREIGN KEY ( movie_title, movie_year ) REFERENCE movie
```

znamená:

```
FOREIGN KEY ( movie_title, movie_year ) REFERENCES movie ( title, year )
```

3.2 Vyjádření cizího klíče relačními operacemi

Schéma K_1 je cizím klíčem relace R_1 na schéma K_2 relace R_2 podle h , právě když

$$\{\langle t_1, t_2 \rangle \mid t_1 \in R_1 \text{ a } t_2 \in R_2 \text{ a } \rho_h(t_1(K_1)) = t_2(K_2)\}$$

je zobrazení R_1 do R_2 .

Vezměme relace R_1 nad S_1 a R_2 nad S_2 , $K_1 \subseteq S_1$ a $K_2 \subseteq S_2$ a přejmenování atributů $h: K_1 \rightarrow K_2$. Schéma K_1 je cizím klíčem relace R_1 na K_2 relace R_2 podle h , právě když K_2 je nadklíčem R_2 a

$$\rho_h(\pi_{K_1}(R_1)) - \pi_{K_2}(R_2) = \emptyset.$$

Například pro

```
# TABLE movie_cast;
      actor_name | movie_title | movie_year
-----+-----+-----
Keano Reeves   | The Matrix  |      1999
Keano Reeves   | Dracula     |      1992
Laurence Fishburne | The Matrix  |      1999
Gary Oldman    | Dracula     |      1992
(4 rows)
```

```
# TABLE movie;
      title | year
-----+-----
The Matrix | 1999
Dracula    | 1992
(2 rows)
```

Je {movie_title,movie_year} cizím klíčem hodnoty movie_cast na {title,year} hodnoty movie:

```
# ( SELECT DISTINCT
      movie_title AS title,
      movie_year AS year
    FROM movie_cast )
EXCEPT
( SELECT DISTINCT
  title,
  year
  FROM movie );

title | year
-----+-----
(0 rows)
```

Porušíme cizí klíč:

```
INSERT INTO movie_cast VALUES
  ( 'Gary Oldman', 'Batman Begins', 2005);
```

Relace už není prázdná:

```
# ( SELECT DISTINCT
      movie_title AS title,
      movie_year AS year
    FROM movie_cast )
EXCEPT
( SELECT DISTINCT
```

```

        title,
        year
FROM    movie );

```

```

    title    | year
-----+-----
Batman Begins | 2005
(1 row)

```

Otázky a úkoly na cvičení

1. Určete všechny nadklíče následující relace.

name	born	nationality
Helena Bonham Carter	1966	England
Emma Watson	1990	English
Kyle MacLachlan	1959	USA
Hugh Laurie	1959	English

2. Vezměme relační proměnnou `actor` danou vlastností „Herec `actor_id` jménem `name` se narodil roku `born` a je národnosti `nationality`“. Předpokládejte, že nemůžou existovat dva herci stejného jména narození ve stejný rok. Určete nadklíče a kandidátní klíče proměnné `actor`. Který z kandidátních klíčů byste zvolili za primární klíč?
3. Vezměme relaci z prvního příkladu a označme si ji R_1 . Dále uvažujme relace R_2 :

actor	movie
Emma Watson	Little Women
Kyle MacLachlan	Dune
Hugh Laurie	101 Dalmatians

R_3 :

actor	movie
Helena Bonham Carter	The King's Speech
Emma Watson	Little Women
Kyle MacLachlan	Dune
Hugh Laurie	101 Dalmatians

a R_4 :

actor	movie
Helena Bonham Carter	The King's Speech
Emma Watson	Little Women
Emma Watson	Beauty and the Beast
Kyle MacLachlan	Dune
Hugh Laurie	101 Dalmatians

Rozhodněte, zda je $\{\text{name}\}$ cizím klíčem relace R_1 na $\{\text{actor}\}$ relace R_2 , R_3 a R_4 podle $h = \{\langle \text{name}, \text{actor} \rangle\}$.

4. Vezměme základní relaci **actor** danou vlastností „Herec **actor** narozený roku **born** žije ve městě **town** v zemi **country**.“ a základní relaci **town** danou vlastností „Město **name** nacházející se v zemi **country** má **population** obyvatel.“ Předpokládejte, že v žádné zemi neexistují dvě města stejného názvu. Je **town**, **country** cizím klíčem základní relace **actor** na **name**, **country** základní relace **town**?
5. Dokažte charakterizace cizího klíče pomocí zobrazení a relačních operací uvedených v části 3.2.