

Databázová integrita

Jiří Zacpal



KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

KMI/DATAB Databáze

Integritní omezení

Příklad

```
CREATE TABLE Osoby
(
  osCislo VARCHAR(7),
  jmeno VARCHAR(20),
  prijmeni VARCHAR(30),
  stav VARCHAR(20),
  obor VARCHAR(30),
  rocnik SMALLINT,
  email VARCHAR(50),
  pohlavi VARCHAR(4),
  datum_zahajeni_studia DATE,
  datum_ukonzeni_studia DATE,
  kredity SMALLINT
);
```

Příklad



INSERT INTO Osoby VALUES

```
('R1245', 'Jan', 'Novák', 'studuje', 'MATAP', 2, 'jan.novak06@upol.cz', 'muž', '1.9.2017',  
null, 12),  
( 'R1246', null, null, 'studuje', 'MMA', 1, 'anna.valkova01@upol.cz', 'žena', '1.9.2018', n  
ull, 10),  
( 'R1247', 'Pavel', 'Holík', 'nestuduje', 'KMI', 2, 'pavel.holik02@upol.cz', 'muž', '1.9.2  
017', '31.7.2016', 34),  
( 'R1248', 'Pavel', 'Mareš', 'studuje', 'KMI', 3, 'pavel.mares02@upol.cz', 'muž', '1.9.201  
6', null, 67),  
( 'R1249', 'Hana', 'Holíková', 'studuje', 'KMI', 1, 'hana.holikova03@upol.cz', 'žena', '1.  
9.2016', null, 40),  
( 'R1250', 'Karel', 'Dlabal', 'nestuduje', null, 1, null, 'muž', '1.9.2017', null, -5),  
( 'R1251', 'Marie', 'Novotná', 'studuje', null, 3, null, 'žena', '1.9.2017', null, null),  
( 'R1251', 'Jan', 'Kinský', 'nestuduje', null, 1, null, 'muž', '1.9.2017', null, null);
```

Integritní omezení

- **NOT NULL** – atribut nemůže mít prázdnou (NULL) hodnotu
- **DEFAULT** – výchozí hodnota pro nový záznam
- **CHECK** – nastavení integritního omezení

Příklad

```
ALTER TABLE Osoby  
  ALTER COLUMN  
    jmeno SET NOT NULL;
```

```
ALTER TABLE Osoby  
  ALTER COLUMN  
    prijmeni SET NOT NULL;
```

- ERROR: column "jmeno" of relation "osoby" contains null values
- Musíme upravit záznam:

```
UPDATE Osoby SET jmeno='Alois', prijmeni='Moudrý' WHERE  
osCislo='R1246';
```

Příklad

```
ALTER TABLE Osoby  
    ALTER COLUMN  
    rocnik SET DEFAULT 1;
```

```
ALTER TABLE Osoby  
    ALTER COLUMN  
    kredity SET DEFAULT 0;
```

- Můžeme vložit záznam:

```
INSERT INTO Osoby(osCislo,jmeno,prijmeni) VALUES  
( 'R1253', 'Marie', 'Svobodová' );
```

Příklad

- Nastavíme omezení na tabulce Osoby:

```
ALTER TABLE Osoby
```

```
ADD CONSTRAINT check_rocnik CHECK (rocnik<=3 AND rocnik>=1);
```

```
ALTER TABLE Osoby
```

```
ADD CONSTRAINT check_pohlavi CHECK (pohlavi='muž' OR  
pohlavi='žena');
```

- Upravíme :

```
UPDATE Osoby SET rocnik=5 WHERE osCislo='R1246';
```

ERROR: new row for relation "osoby" violates check constraint "check_rocnik"

```
UPDATE Osoby SET pohlavi='X' WHERE osCislo='R1246';
```

ERROR: new row for relation "osoby" violates check constraint "check_pohlavi"

Příklad

- Odstraníme omezení z tabulky Osoby:

```
ALTER TABLE Osoby  
DROP CONSTRAINT check_rocnik;
```

```
ALTER TABLE Osoby  
DROP CONSTRAINT check_pohlavi;
```

- Upravíme :

```
UPDATE Osoby SET rocnik=5 WHERE osCislo='R1246';  
UPDATE Osoby SET pohlavi='X' WHERE osCislo='R1246';
```

Příklad

- Nastavíme omezení na tabulce Osoby:

```
ALTER TABLE Osoby
```

```
  ADD CONSTRAINT check_datum CHECK  
(datum_ukonzeni_studia > datum_zahajeni_studia);
```

ERROR: check constraint "check_datum" of relation "osoby" is violated by some row

- Musíme upravit n-tici:

```
UPDATE Osoby SET datum_ukonzeni_studia='31.7.2019' WHERE  
osCislo='R1247';
```

Úkol



Doplňte do tabulky **Predmety** tato integritní omezení:

- nenulové hodnoty pro sloupec **nazev**,
- výchozí hodnota 2021 pro sloupce **rok**,
- kontrolu, zda **datum_od** je starší než **datum_do**,
- nastavte omezení pro **kapacitu** 0 až 150.

```
CREATE TABLE Predmety
(
  id INT,
  nazev VARCHAR(30),
  katedra VARCHAR(4),
  zkratka VARCHAR(6),
  ucitel VARCHAR(30),
  rok SMALLINT,
  kapacita SMALLINT,
  budova VARCHAR(4),
  mistnost VARCHAR(6),
  datum_od DATE,
  datum_do DATE,
  kredity SMALLINT
)
```

Klíče

Příklad

```
CREATE TABLE Osoby
(
  osCislo VARCHAR(7),
  jmeno VARCHAR(20),
  prijmeni VARCHAR(30),
  stav VARCHAR(20),
  obor VARCHAR(30),
  rocnik SMALLINT,
  email VARCHAR(50),
  pohlavi VARCHAR(4),
  datum_zahajeni_studia DATE,
  datum_ukonzeni_studia DATE,
  kredity SMALLINT
);
```

Příklad



INSERT INTO Osoby VALUES

```
('R1245', 'Jan', 'Novák', 'studuje', 'MATAP', 2, 'jan.novak06@upol.cz', 'muž', '1.9.2017', null, 12),  
( 'R1246', null, null, 'studuje', 'MMA', 1, 'anna.valkova01@upol.cz', 'žena', '1.9.2018', null, 10),  
( 'R1247', 'Pavel', 'Holík', 'nestuduje', 'KMI', 2, 'pavel.holik02@upol.cz', 'muž', '1.9.2017', '31.7.2016', 34),  
( 'R1248', 'Pavel', 'Mareš', 'studuje', 'KMI', 3, 'pavel.mares02@upol.cz', 'muž', '1.9.2016', null, 67),  
( 'R1249', 'Hana', 'Holíková', 'studuje', 'KMI', 1, 'hana.holikova03@upol.cz', 'žena', '1.9.2016', null, 40),  
( 'R1250', 'Karel', 'Dlabal', 'nestuduje', null, 1, null, 'muž', '1.9.2017', null, -5),  
( 'R1251', 'Marie', 'Novotná', 'studuje', null, 3, null, 'žena', '1.9.2017', null, null),  
( 'R1251', 'Jan', 'Kinský', 'nestuduje', null, 1, null, 'muž', '1.9.2017', null, null);
```

- Můžeme vložit:

INSERT INTO Osoby VALUES

```
('R1245', 'Jan', 'Novák', 'studuje', 'MATAP', 2, 'jan.novak06@upol.cz', 'muž', '1.9.2017', null, 12);
```

- Už se nejedná o relaci.
- Co s tím?

Klíč



- R : relační proměnná nad A_1, \dots, A_n
- K : podmnožina $\{A_1, \dots, A_n\}$
- r : hodnota proměnné R

- K je **klíč**.

Příklad

- Vytvoříme klíč, který se skládá ze všech atributů:

```
ALTER TABLE Osoby
```

```
ADD CONSTRAINT vse_key
```

```
UNIQUE(osCislo,jmeno,prijmeni,stav,obor,rocnik,email,pohla  
vi,datum_zahajeni_studia,datum_ukonceni_studia,kredity);
```


Superklíč



- R : relační proměnná nad A_1, \dots, A_n
- K : podmnožina $\{A_1, \dots, A_n\}$
- r : hodnota proměnné R

- K je **superklíč** relace R pokud platí následující.
 - Pro každé dvě n -tice t_1, t_2 v těle relace r platí, že
 - pokud projekce t_1 na K se rovná projekci t_2 na K , pak t_1 se rovná t_2 .
- Volněji: Každá n -tice z těla relace r je určena hodnotami atributů z K .
- Je klíč `vse_key` superklíč?

Kandidátní klíč

- R: relační proměnná nad A_1, \dots, A_n
 - K: podmnožina $\{A_1, \dots, A_n\}$
 - r: hodnota proměnné R
-
- K je **kandidátní klíč** relace R pokud:
 1. K je superklíč.
 2. Žádná vlastní podmnožina K není superklíč.

Příklad

- Nejdříve odstraníme vse_key:

```
ALTER TABLE Osoby  
    DROP CONSTRAINT vse_key;
```

- Vytvoříme kandidátní klíč jmeno_key:

```
ALTER TABLE Osoby  
    ADD CONSTRAINT jmeno_key UNIQUE(jmeno,prijmeni);
```

Příklad

- Vytvoříme kandidátní klíč `cislo_key`:

```
ALTER TABLE Osoby
```

```
ADD CONSTRAINT cislo_key UNIQUE(oscislo);
```

```
could not create unique index "cislo_key,,
```

- Aktualizujeme n-tici:

```
UPDATE Osoby SET oscislo='R1252' WHERE prijmeni='Kinský'
```

- Vytvoříme kandidátní klíč `cislo_key`.

Primární klíč

- Jako primární klíč relační proměnné je vybrán jeden z kandidátních klíčů.
- Ostatní klíče relační proměnné se nazývají alternativní klíče.
- $\{A_1, \dots, A_n\}$: kandidátní klíč relační proměnné
- c : jméno (končí pkey)
- primární klíč:

CONSTRAINT c **PRIMARY KEY** (A_1, \dots, A_n)

Příklad

- Odebereme kandidátní klíče cislo_key:

```
ALTER TABLE Osoby DROP CONSTRAINT jmeno_key;
```

```
ALTER TABLE Osoby DROP CONSTRAINT cislo_key;
```

- Vytvoříme primární klíč:

```
ALTER TABLE Osoby
```

```
ADD CONSTRAINT cislo_pkey PRIMARY KEY(oscislo);
```

- Vytvoříme primární klíč jmeno_key.

```
ALTER TABLE Osoby
```

```
ADD CONSTRAINT jmeno_pkey PRIMARY KEY(jmeno,prijmeni);
```

```
ERROR: multiple primary keys for table "osoby" are not  
allowed
```

Úkol



Doplňte do tabulky **Predmety** vhodné klíče.

```
CREATE TABLE Predmety
(
  id INT,
  nazev VARCHAR(30),
  katedra VARCHAR(4),
  zkratka VARCHAR(6),
  ucitel VARCHAR(30),
  rok SMALLINT,
  kapacita SMALLINT,
  budova VARCHAR(4),
  mistnost VARCHAR(6),
  datum_od DATE,
  datum_do DATE,
  kredity SMALLINT
)
```

Referenční integritní omezení

Příklad



```
CREATE TABLE Osoby
(
  osCislo VARCHAR(7),
  jmeno VARCHAR(20),
  prijmeni VARCHAR(30),
  stav VARCHAR(20),
  obor VARCHAR(30),
  rocnik SMALLINT,
  email VARCHAR(50),
  pohlavi VARCHAR(4),
  datum_zahajeni_studia DATE,
  datum_ukonzeni_studia DATE,
  kredity SMALLINT,
  CONSTRAINT cislo_pkey PRIMARY
  KEY(oscislo)
);
```

```
CREATE TABLE Predmety
(
  id INT,
  nazev VARCHAR(30),
  katedra VARCHAR(4),
  zkratka VARCHAR(6),
  ucitel VARCHAR(30),
  rok SMALLINT,
  kapacita SMALLINT,
  budova VARCHAR(4),
  mistnost VARCHAR(6),
  datum_od DATE,
  datum_do DATE,
  kredity SMALLINT,
  CONSTRAINT id_pkey PRIMARY KEY(id)
)
```

Příklad



INSERT INTO Osoby VALUES

```
('R1245', 'Jan', 'Novák', 'studuje', 'MATAP', 2, 'jan.novak06@upol.cz', 'muž', '1.9.2017', null, 1),
('R1246', null, null, 'studuje', 'MMA', 1, 'anna.valkova01@upol.cz', 'žena', '1.9.2018', null, 10),
('R1247', 'Pavel', 'Holík', 'nestuduje', 'KMI', 2, 'pavel.holik02@upol.cz', 'muž', '1.9.2017', '31.7.2016', 34),
('R1248', 'Pavel', 'Mareš', 'studuje', 'KMI', 3, 'pavel.mares02@upol.cz', 'muž', '1.9.2016', null, 67),
('R1249', 'Hana', 'Holíková', 'studuje', 'KMI', 1, 'hana.holikova03@upol.cz', 'žena', '1.9.2016', null, 40),
('R1250', 'Karel', 'Dlabal', 'nestuduje', null, 1, null, 'muž', '1.9.2017', null, -5),
('R1251', 'Marie', 'Novotná', 'studuje', null, 3, null, 'žena', '1.9.2017', null, null),
('R1252', 'Jan', 'Kinský', 'nestuduje', null, 1, null, 'muž', '1.9.2017', null, null);
```

INSERT INTO Predmety

```
VALUES(402, 'Databáze', 'KMI', 'DB', 'Jiří Zacpal', 2018, 56, 'LP', '5006', '12.2.2019', '7.5.2019', 5),
(543, 'Základní software', 'KMI', 'ZSW', 'Jiří Zacpal', 2018, 36, 'LP', '5002', '18.9.2018', '11.12.2018', 4),
(224, 'Základy práce s PC L', 'KMI', 'ZPPCL', 'Jiří Zacpal', 2018, 18, 'LP', '5003', '13.2.2019', '8.5.2019', 3),
(403, 'Algoritmy 1', 'KMI', 'ALG1', 'Arnošt Večerka', 2019, 25, 'LP', '5002', '13.2.2020', '8.5.2020', 3);
```

Příklad

- Jak dodat informace o tom, že nějaký student studuje předmět?

ALTER TABLE Osoby

ADD COLUMN Predmet INT;

- Aktualizujeme n-tice v relaci Osoby:

UPDATE Osoby SET predmet=402 WHERE oscislo='R1245';

UPDATE Osoby SET predmet=402 WHERE oscislo='R1246';

UPDATE Osoby SET predmet=403 WHERE oscislo='R1247';

UPDATE Osoby SET predmet=403 WHERE oscislo='R1248';

UPDATE Osoby SET predmet=224 WHERE oscislo='R1249';

UPDATE Osoby SET predmet=224 WHERE oscislo='R1250';

Příklad

- Aktualizujeme n-tice v relaci Osoby:

```
UPDATE Osoby SET predmet=405 WHERE oscislo=,R1251';
```

- Tento předmět ale vůbec neexistuje!

Cizí klíč

- R_1 : relační proměnná nad A_1, \dots, A_n
- R_2 : relační proměnná nad B_1, \dots, B_m
- r_1 : hodnota proměnné R_1
- r_2 : hodnota proměnné R_2

- C_1, \dots, C_k : atributy ze záhlaví R_1
- D_1, \dots, D_k : kandidátní klíč R_2

- C_1, \dots, C_k je **cizí klíč** relační proměnné R_1 na atributy D_1, \dots, D_k relační proměnné R_2 pokud platí následující:
 - Pro každou n -tici t_1 v těle r_1 existuje n -tice t_2 v těle r_2 taková, že pro každé $1 \leq i \leq k$ je hodnota přiřazená n -ticí t_1 atributu C_i rovna hodnotě přiřazené n -ticí t_2 atributu D_i .

- Například:
- Atribut `child_name` je cizí klíč relační proměnné `parent` na atribut `name` relační proměnné `child`.

Cizí klíč jako integritní omezení

- c: jméno omezení (končí fkey)

- integritní omezení:

CONSTRAINT c **FOREIGN KEY** (C1, ..., Ck) **REFERENCES** R2 (D1, ..., Dk)

Příklad

- Přidáme cizí klíč:

```
ALTER TABLE osoby ADD CONSTRAINT osoby_predmet_fkey  
FOREIGN KEY (predmet) REFERENCES predmety (id);
```

- Nelze. Musíme změnit n-tici:

```
UPDATE Osoby SET predmet=403 WHERE oscislo='R1251';
```

Příklad



- Co když chceme chce jeden student mít zapsáno více předmětů?

```
INSERT INTO Osoby VALUES
```

```
('R1245', 'Jan', 'Novák', 'studuje', 'MATAP', 2, 'jan.novak06@upol.cz', 'm  
už', '1.9.2017', null, 12, 403);
```

- Nelze. n-tice s tímto atributem odcislo už existuje.
- Jak to vyřešit?

Příklad

- Vytvoříme relaci:

```
CREATE TABLE Zapsane_predmety  
(  
    id INT,  
    student VARCHAR(7),  
    predmet int  
);
```

- Odstraníme cizí klíč a atribut predmet:

```
ALTER TABLE Osoby  
DROP CONSTRAINT osoby_predmet_fkey;  
ALTER TABLE Osoby  
DROP predmet;
```

Příklad

- Nastavíme referenční integritu:

```
ALTER TABLE zapsane_predmety ADD CONSTRAINT osoby_fkey  
FOREIGN KEY (student) REFERENCES osoby (oscislo);  
ALTER TABLE zapsane_predmety ADD CONSTRAINT predmety_fkey  
FOREIGN KEY (predmet) REFERENCES predmety (id);
```

- Vytvoříme primární klíč:

```
ALTER TABLE zapsane_predmety  
ADD CONSTRAINT student_predmet_pkey PRIMARY KEY(student,predmet);
```

Příklad

- Můžeme přidávat n-tice:
`INSERT INTO zapsane_predmety VALUES`
`(1, 'R1245', 402),`
`(1, 'R1246', 402),`
`(2, 'R1247', 402),`
`(3, 'R1248', 403),`
`(4, 'R1249', 224);`

Bodovaný úkol



Pro relace v souboru `datab_07_sql_klice_ukol.sql` nastavte:

1. Primární klíče.
2. Cizí klíče tak, aby šlo půjčovat jen existující knihy existujícím čtenářům a kniha aby měla jen existujícího autora.
3. Dále nastavte tato omezení:
 - v relaci Autor, aby nebyly prázdné atributy Jmeno a Prijmeni,
 - v relaci Kniha, aby atribut rok byl v intervalu 0 až 2021 a atribut Nazev nebyl prázdný,
 - v relaci Ctenar, aby nebyly prázdné atributy Jmeno a Prijmeni.