
Tehnologije upravljanja podatkov

Univerzitetni študij

Bolonjski program



Matjaž Kukar, 2011

DML ukazi v SQL

- SELECT
- INSERT
- UPDATE
- DELETE

Primeri relacij (tabel)

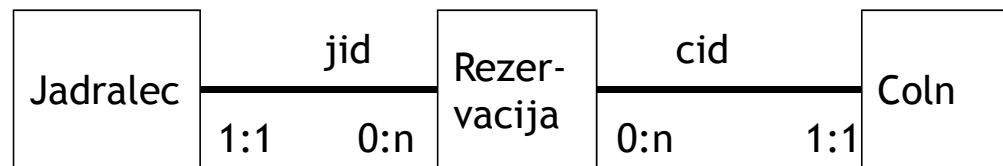
- Sheme za primere rel. algebre:

Jadralec(jid, ime, rating, starost)

Coln(cid, ime, dolzina, barva)

Rezervacija(jid, cid, dan)

- Pomen in povezava relacij:



Primeri relacij (tabel)

Jadralec:

| <u>jid</u> | ime | rating | starost |
|------------|---------|--------|---------|
| 22 | Darko | 7 | 45 |
| 29 | Borut | 1 | 33 |
| 31 | Lojze | 8 | 55.5 |
| 32 | Andrej | 8 | 25.5 |
| 58 | Rajko | 10 | 35 |
| 64 | Henrik | 7 | 35 |
| 71 | Zdravko | 10 | 16 |
| 74 | Henrik | 9 | 35 |
| 85 | Anze | 3 | 25.5 |
| 95 | Bine | 3 | 63.5 |

Rezervacija:

| <u>jid</u> | <u>cid</u> | <u>dan</u> |
|------------|------------|------------|
| 22 | 101 | 2006-10-10 |
| 22 | 102 | 2006-10-10 |
| 22 | 103 | 2006-10-08 |
| 22 | 104 | 2006-10-07 |
| 31 | 102 | 2006-11-10 |
| 31 | 103 | 2006-11-06 |
| 31 | 104 | 2006-11-12 |
| 64 | 101 | 2006-09-05 |
| 64 | 102 | 2006-09-08 |
| 74 | 103 | 2006-09-08 |

Coln:

| <u>cid</u> | ime | dolzina | barva |
|------------|-------------|---------|--------|
| 101 | Elan | 34 | modra |
| 102 | Elan | 34 | rdeca |
| 103 | Sun Odyssey | 37 | zelena |
| 104 | Bavaria | 50 | rdeca |

INSERT stavek...

```
INSERT INTO TableName [ (columnList) ]  
VALUES (dataValueList)
```

- Seznam columnList ni obvezen; če ga spustimo, interpreter pričakuje vrednosti za vse stolpce tabele, v vrstnem redu, kot so bili kreirani.
- Pri vnosu moramo vpisati najmanj vse obvezne vrednosti (not null), razen za stolpce, pri katerih je bila ob kreiranju določena privzeta vrednost (DEFAULT).

INSERT stavek...

- Seznam `dataValueList` mora ustrezati seznamu `columnList`:
 - Število elementov v seznamih mora biti enako;
 - Vrednost, ki se nanaša na nek stolpec, mora biti v seznamu `dataValueList` na istem mestu, kot je stolpec v seznamu `columnList`;
 - Podatkovni tip vrednosti, ki se nanaša na nek stolpec, mora biti enak kot podatkovni tip stolpca.

Primeri INSERT stavkov...

Jadralec(jid, ime, rating, starost)
Coln(cid, ime, dolzina, barva)
Rezervacija(jid, cid, dan)

- Vnos nove vrstice v tabelo rezervacija:

INSERT INTO rezervacija

- VALUES (74, 102, DATE'2011-11-20');

- Shema relacije rezervacija:

Rezervacija(jid, cid, dan)

Primeri INSERT stavkov...

Jadralec(jid, ime, rating, starost)
Coln(cid, ime, dolzina, barva)
Rezervacija(jid, cid, dan)

- Relacija jadralec(jid, ime, rating, starost)
- Vnos nove vrstice v tabelo jadralec – vnos samo obveznih vrednosti

```
INSERT INTO jadralec
```

```
VALUES (300, null, null, null)
```

Stolpci ime, starost in rating so neobvezni.

- ali

```
INSERT INTO jadralec(jid, ime, rating, starost)
```

```
VALUES (300, 'Patrik', 8, 20);
```


Primeri INSERT stavkov...

- Vnos več vrstic iz ene ali več drugih tabel...

```
INSERT INTO TableName [ (columnList) ]  
SELECT columnList ...
```

Primeri INSERT stavkov...

- Predpostavimo, da imamo tabelo stariJadralec z enako shemo kot tabela jadralec. Vanjo želimo vnesti oznake, imena in starost jadralcev, starejših od 40 let.

```
INSERT INTO StariJadralec(jid, ime, starost)
  SELECT jid, ime, starost
  FROM jadralec
  WHERE starost > 40;
```

Ratinga ne prenašamo:
postane NULL

UPDATE stavek...

UPDATE TableName

SET columnName1 = dataValue1

[, columnName2 = dataValue2...]

[WHERE searchCondition]

- TableName se lahko nanaša na ime osnovne tabele ali ime pogleda.
- Sklop SET določa nazive enega ali več stolpcev ter nove vrednosti teh stolpcev.

UPDATE stavek



- WHERE sklop je neobvezen:
 - Če ga spustimo, se v imenovane stolpce vpišejo nove vrednosti za vse vrstice v tabeli;
 - Če WHERE sklop določimo, se spremembe zgodijo zgolj za vrstice, ki ustrezajo WHERE pogojem.
- Nove podatkovne vrednosti morajo ustrezati podatkovnemu tipu stolpca.

Primeri UPDATE stavkov

Jadralec(jid, ime, rating, starost)
Coln(cid, ime, dolzina, barva)
Rezervacija(jid, cid, dan)

- Vsem starim jadralcem postavimo rating na 10.

```
UPDATE stariJadralec  
SET rating=10;
```

- Vsem starim jadralcem zmanjšaj rating za 10%.

```
UPDATE stariJadralec  
SET rating=0.9 * rating;
```

Primeri UPDATE stavkov

```
Jadralec(jid, ime, rating, starost)  
Coln(cid, ime, dolzina, barva)  
Rezervacija(jid, cid, dan)
```

- Vsem starim jadralcem pripišemo originalni rating iz tabele jadralec.

```
UPDATE stariJadralec sj  
SET rating = ( SELECT rating  
FROM jadralec  
WHERE jid=sj.jid );
```

Povezava z drugimi tabelami je možna samo preko uporabe vgnezdenih poizvedb.

DELETE stavek

DELETE FROM TableName
[WHERE searchCondition]

- TableName se lahko nanaša na ime osnovne tabele ali ime pogleda.
- WHERE sklop ni obvezen. Če ga spustimo, zbrisemo vse vrstice v tabeli. Tabela ostane.

Primeri DELETE stavkov

- Izbriši vse rezervacije pred letom 2006.

```
DELETE FROM rezervacija  
WHERE EXTRACT(YEAR FROM dan) < 2006;
```


Primeri DELETE stavkov

Jadralec(jid, ime, rating, starost)
Coln(cid, ime, dolzina, barva)
Rezervacija(jid, cid, dan)

- Izbriši vse rezervacije zelenih čolnov pred letom 2006.

```
DELETE FROM rezervacija
WHERE EXTRACT(YEAR FROM dan) < 2006
AND cid IN (SELECT cid
FROM coln
WHERE barva='zelena');
```

Povezava z drugimi tabelami je možna samo preko uporabe vgnezdenih poizvedb.

Stavek TRUNCATE TABLE

- S pomočjo stavka TRUNCATE TABLE izbrišemo vse vrstice tabele.

TRUNCATE TABLE TbName

- Mnogo hitrejša kot DELETE FROM TbName.
- Tabela ne sme imeti integritetnih ali referenčnih omejitev.
- Primer:

TRUNCATE TABLE jadralec

Stavki skupine SQL DDL...

- DDL skupina zajema SQL stavke za manipulacijo s strukturo podatkovne baze.
- Tabele in omejitve:
 - Podatkovni tipi, ki jih podpira SQL standard
 - Integritetne omejitve
 - Kako definirati omejitve z SQL-om
 - Uporaba integritetnih omejitev v CREATE in ALTER TABLE stavkih
- Pogledi:
 - Kako kreirati in brisati poglede z SQL-om?
 - Kako SUPB izvaja operacije nad pogledi?
 - Pod kakšnimi pogoji so pogledi spremenljivi?
 - Prednosti in slabosti pogledov

Kreiranje tabele

```
CREATE TABLE TableName (  
    ImeAtributa PodatkovniTip,  
    ImeAtributa PodatkovniTip,  
    ...  
    Omejitve  
    ...  
)
```

- Imena (različna!) in tipi atributov (standardni, definirani)
- Omejitve vrednosti atributov (celovitost vrednosti atributov)
- Referenčne omejitve (celovitost povezav)
- Integritetne omejitve (celovitost vrstic, poslovna pravila)

Standardni podatkovni tipi v ISO SQL

Table 6.1 ISO SQL data types.

| Data type | Declarations | | | |
|---------------------|------------------------|-------------|---------------------|----------|
| boolean | BOOLEAN | | | |
| character | CHAR | VARCHAR | | |
| bit | BIT | BIT VARYING | | |
| exact numeric | NUMERIC | DECIMAL | INTEGER | SMALLINT |
| approximate numeric | FLOAT | REAL | DOUBLE PRECISION | |
| datetime | DATE | TIME | TIMESTAMP | |
| interval | INTERVAL | | | |
| large objects | CHARACTER LARGE OBJECT | | BINARY LARGE OBJECT | |

Integritetne omejitve – celovitost podatkov

- Za zagotavljanje celovitosti podatkov SQL standard ponuja več vrst omejitev:
 - Obveznost podatkov
 - Omejitve domene (Domain constraints); podtipi atributov
 - Pravila za celovitost podatkov (Integrity constraints)
 - Celovitost entitet (Entity Integrity)
 - Celovitost povezav (Referential Integrity)
 - Števnost (Multiplicity)
 - Splošne omejitve (General constraints)
- Omejitve so lahko definirane v CREATE TABLE ali ALTER TABLE stavkih.

Integritetne omejitve – celovitost atributov

- Obveznost podatkov

ime VARCHAR(10) NOT NULL

- Omejitve domene

CHECK (rating >= 1 AND rating <= 10)

- Domena natančneje kot podatkovni tip določa množico dopustnih vrednosti (domena=podtip)
- Domene lahko eksplicitno definiramo; smiselno če nek podtip pogosto uporabljamo

Integritetne omejitve – celovitost atributov

- CREATE DOMAIN

```
CREATE DOMAIN DomainName [AS] dataType  
[DEFAULT defaultOption]  
[CHECK (searchCondition)]
```

Primer:

```
CREATE DOMAIN Tspol AS CHAR  
CHECK (VALUE IN ('M', 'Ž'));
```

Deklaracija atributa:

```
Spol Tspol NOT NULL
```


Integritetne omejitve – celovitost atributov

- searchCondition lahko vsebuje iskalno tabelo (lookup table):

```
CREATE DOMAIN sifraColna AS INTEGER  
CHECK (VALUE IN (SELECT cid FROM Coln));
```

- Domeno lahko ukinemo z uporabo stavka DROP DOMAIN:

```
DROP DOMAIN DomainName
```

```
[RESTRICT | CASCADE]
```

Kaj naj se zgodi, če je domena trenutno v uporabi

Integritetne omejitve - celovitost vrstic

- Primarni ključ tabele mora vsebovati enolično neprazno vrednost v vsaki vrstici tabele.
- ISO standard podpira primarne ključe s sklopom PRIMARY KEY v okviru CREATE in ALTER TABLE stavkov.

PRIMARY KEY(EMSO)

- Vsaka tabela ima lahko največ en primarni ključ. Enoličnost neosnovnih (niso del primarnega ključa) stolpcev zagotavljamo z uporabo UNIQUE

UNIQUE(davcna_stevilka)

Integritetne omejitve - celovitost povezav

- Implementacija povezav med tabelami s tujimi ključi.
- FOREIGN KEY (tuji ključ) je stolpec ali množica stolpcev, ki povezujejo vsako vrstico tabele A z vrstico referenčne tabele B, kjer se ujemajo vrednosti $A.FK = B.PK$.
- Celovitost povezav zagotavlja, da če ima tuji ključ neko vrednost, potem se ta vrednost nahaja v primarnem ključu povezane tabele.
- ISO standard omogoča definicijo tujih ključev s sklopom FOREIGN KEY v CREATE in ALTER TABLE

FOREIGN KEY(jid) REFERENCES jadralec(jid)

Integritetne omejitve - celovitost povezav

- Vsak INSERT/UPDATE stavek, ki skuša kreirati FK vrednost v tabeli, ne da bi ta vrednost obstajala kot PK v povezani tabeli, je zavrjen.
- Ob UPDATE/DELETE operacijah nad referencirano tabelo so možne naslednje akcije glede na originalno operacijo (ON UPDATE, ON DELETE):
 - CASCADE: briše ustrezne vrstice iz vseh tabel, ki referencirajo
 - SET NULL: postavi FK na NULL
 - SET DEFAULT: postavi FK na privzeto vrednost (če je definirana)
 - NO ACTION: zavrne update/delete operacijo

Integritetne omejitve - celovitost povezav

- Določimo z uporabo ON UPDATE, ON DELETE ON UPDATE SET NULL
- Primeri:

```
FOREIGN KEY (jid) REFERENCES jadralec(jid)  
ON DELETE SET NULL
```

```
FOREIGN KEY (cid) REFERENCES coln(cid)  
ON UPDATE CASCADE
```

Kreiranje podatkovnih objektov...

- SQL DDL omogoča kreiranje, spreminjanje in brisanje podatkovnih objektov, kot so: shema, domena, tabela, pogled in indeks.
- Glavni SQL DDL stavki so:

CREATE SCHEMA

DROP SCHEMA

CREATE/ALTER DOMAIN

DROP DOMAIN

CREATE/ALTER TABLE

DROP TABLE

CREATE VIEW

DROP VIEW

CREATE INDEX

DROP INDEX

Kreiranje podatkovnih objektov...

- Relacije in drugi podatkovni objekti obstajajo v nekem okolju.
- Vsako okolje vsebuje enega ali več katalogov, vsak katalog pa množico shem.
- Shema je poimenovana kolekcija povezanih podatkovnih objektov.
- Objekti v shemi so lahko tabele, pogledi, domene, trditve, dodelitve, pretvorbe in znakovni nizi. Vsi objekti imajo istega lastnika.
- Primera shem na pb.fri.uni-lj.si: pb, vaje

Kreiranje sheme

CREATE SCHEMA [Name |
AUTHORIZATION CreatorId]

DROP SCHEMA Name [RESTRICT | CASCADE]

- RESTRICT (privzeto): shema mora biti prazna, sicer brisanje ni možno.
- CASCADE: kaskadno se brišejo vsi objekti, povezani s shemo. Če katerokoli brisanje ne uspe, se zavrne celotna operacija.

Kreiranje tabele...



```
CREATE TABLE TableName
({colName dataType [NOT NULL] [UNIQUE]
  [DEFAULT defaultOption]
  [CHECK searchCondition] [,...]}
{[FOREIGN KEY (listOfFKColumns)
  REFERENCES ParentTableName [(listOfCKColumns)],
  [ON UPDATE referentialAction]
  [ON DELETE referentialAction ]] [,...]}
  {[CHECK (searchCondition)] [,...]}
[PRIMARY KEY (listOfColumns),]
{[UNIQUE (listOfColumns),] [...,]}
)
```

Primer kreiranja tabele...

Najprej kreiramo domene

```
CREATE DOMAIN sifraColna AS INTEGER  
CHECK (VALUE IN (SELECT cid FROM coln));
```

```
CREATE DOMAIN rezervDan AS DATE  
CHECK(VALUE BETWEEN DATE'1.1.1995'  
AND DATE'1.1.2200');
```

```
CREATE DOMAIN sifraJadralca AS INTEGER  
CHECK(VALUE BETWEEN 100 AND 999);
```

```
CREATE DOMAIN spolJadralca AS CHAR  
CHECK (VALUE IN ('M', 'Ž'));
```

Primer kreiranja tabele...

potem kreiramo tabelo



```
CREATE TABLE Rezervacija (  
    jid          sifraJadralca  NOT NULL,  
    cid          sifraColna    NOT NULL,  
    dan          rezervDan     NOT NULL DEFAULT date(),  
    CONSTRAINT PrevecRezervacijColna  
        CHECK (NOT EXISTS (SELECT cid, dan  
                            FROM rezervacija  
                            GROUP BY jid, dan  
                            HAVING COUNT(*) > 1))  
    PRIMARY KEY (jid, cid, dan),  
    FOREIGN KEY (jid) REFERENCES jadralec(jid)  
        ON DELETE SET NULL ON UPDATE CASCADE,  
    FOREIGN KEY (cid) REFERENCES coln(cid)  
        ON DELETE SET NULL ON UPDATE CASCADE,  
    ...);
```

ALTER TABLE stavek...

- S stavkom ALTER TABLE lahko:
 - Dodajamo ali ukinjamo stolpce v tabeli;
 - Dodajamo ali ukinemo omejitve tabele;
 - Za stolpce v tabeli določamo ali ukinjamo privzete vrednosti;
 - Spreminjamo podatkovne tipe stolpcev v tabeli;

Primeri ALTER TABLE stavkov...

- Spremeni tabelo rezervacija tako, da ukineš privzeto vrednost stolpca dan.

ALTER TABLE rezervacija

ALTER dan DROP DEFAULT;

Primeri ALTER TABLE stavkov...

- Spremeni tabelo rezervacija tako, da ukineš omejitev, da noben čoln ne sme imeti več kot eno rezervacijo na isti dan. V tabelo jadralec dodaj stolpec Spol.

ALTER TABLE rezervacija

DROP CONSTRAINT PrevecRezervacijColna;

ALTER TABLE jadralec

ADD spol spolJadralca NOT NULL DEFAULT = 'M';

Stavek DROP TABLE

- S pomočjo stavka DROP TABLE ukinemo tabelo. Obenem se zbrišejo vsi zapisi tabele.

DROP TABLE TbName [RESTRICT | CASCADE]

- Restrict: Ukaz se ne izvede, če obstajajo objekti, ki so vezani na tabelo, ki jo brišemo.
- Cascade: kaskadno se brišejo vsi vezani objekti.
- Primer:

DROP TABLE jadralec RESTRICT;