

# BAZE PODATAKA

## SQL Rukovanje podacima



Neđeljko Lekić

Irena Orović

[www.etf.ac.me](http://www.etf.ac.me)

[www.elektronika.t-com.me](http://www.elektronika.t-com.me)

# U OVOJ LEKCIJI

- SQL
  - INSERT, UPDATE, i DELETE
  - Rječnik podataka
  - SQL SELECT
    - WHERE klauzule
    - SELECT iz više tabela
    - JOINS

# INSERT, UPDATE i DELETE

- **INSERT** – dodavanje reda (zapisa) u tabelu.
- **UPDATE** – izmjena podataka u zapisu (zapisima) tabele
- **DELETE** – brisanje zapisa iz tabele
- **UPDATE i DELETE** korite **WHERE** klauzulu kojom se specificira koje zapise izmijeniti ili ukloniti
- **BUDITE PAŽLJIVI** Netačnom **WHERE** klauzulom može se izgubiti puno podataka.

# INSERT

```
INSERT INTO  
  <table>  
  (col1, col2, ...)  
VALUES  
  (val1, val2, ...)
```

- Broj kolona i vrijednosti mora biti isti.
- Ako se dododaju vrijednosti u svaku kolonu, ne mora se navoditi lista sa imanima kolona
- SQL ne zahtijeva da svaki zapis bude različit (osim ako neko ograničenje to ne nameće).

# INSERT

Neka je polazna tabela sljedeća:

Student

ID	studIme	studAdresa	studGodina
1	Patar Marić	Slobode 23	1

```
INSERT INTO Student
```

```
(ID, studIme, studAdresa, studGodina)
```

```
VALUES (2, 'Marko Matić', 'Cvijeća 12', 3)
```

Student

ID	studIme	studAdresa	studGodina
1	Patar Marić	Slobode 23	1
2	Marko Matić	Cvijeća 12	3

# INSERT

```
INSERT INTO Student
(studIme, ID, studGodina)
VALUES ('Ana Tot', 3, 1)
```

Student

ID	studIme	studAdresa	studGodina
1	Patar Marić	Slobode 23	1
2	Marko Matić	Pobjede 12	3
3	Ana Tot		1

```
INSERT INTO Student
VALUES (4, 'Maja Šoć', 'Cetinjski put bb', 3)
```

Student

ID	studIme	studAdresa	studGodina
1	Patar Marić	Slobode 23	1
2	Marko Matić	Pobjede 12	3
3	Ana Tot		1
4	Maja Šoć	Cetinjski put bb	3

# UPDATE

```
UPDATE <table>
SET col1 = val1
    [, col2 = val2...]
[WHERE
    <condition>]
```

- U svim vrstama kod kojih je uslov zadovoljen postavljaju se zadate vrijednosti kolonama.
- Ako nije zadat uslov svi zapisi će biti promijenjeni, te **BUDITE PAŽLJIVI**
- Vrijednosti su konstante ili algebarski izrazi.

# UPDATE

```
UPDATE Student
SET studGodina = 2, studIme = 'Marina Šoć'
WHERE ID = 4
```

Student

ID	studIme	studAdresa	studGodina
1	Patar Marić	Slobode 23	1
2	Marko Matić	Pobjede 12	3
3	Ana Tot		1
4	Marina Šoć	Cetinjski put bb	2

```
UPDATE Student
SET studGodina = studGodina + 1
```

Student

ID	studIme	studAdresa	studGodina
1	Patar Marić	Slobode 23	2
2	Marko Matić	Pobjede 12	4
3	Ana Tot		2
4	Marina Šoć	Cetinjski put bb	3



# DELETE

- Ukljanja sve zapise koji zadovoljavaju uslov

**DELETE FROM**

**<table>**

**[WHERE**

**<condition>]**

- Ako neme uslova, onda će SVI zapisi biti obrisani – BUDITE PAŽLJIVI
- Neke verzije SQL-a imaju i naredbu **TRUNCATE TABLE <T>** koja je kao i **DELETE FROM <T>** ali je u nekim situacijama brža.

# DELETE

```
DELETE FROM  
Student  
WHERE studGodina = 2
```

Student

ID	studIme	studAdresa	studGodina
2	Marko Matić	Pobjede 12	4
4	Mrina Šoć	Cetinjski put bb	3

```
DELETE FROM Student  
or  
TRUNCATE TABLE Student
```

Student

ID	studIme	studAdresa	studGodina
----	---------	------------	------------

# SELECT

- SQL komanda koja se najčešće koristi.
  - Upit prema grupi tabela. Rezultat je takođe tabela.
  - Puno opcija. Upoznaćemo se sa mnogim od njih.
  - Obično postoji više načina za sastaviti bilo koji upit.
- SQL-ov SELECT je sličan ali se i razlikuje od selekcije  $\sigma$  u relacionoj algebri.
  - SELECT u SQL radi sve što i relacionalna algebra.
  - Ipak SQL se razlikuje od relacionog modela.

# SQL SELECT - PREGLED

**SELECT**

**[DISTINCT | ALL] <column-list>**

**FROM <table-names>**

**[WHERE <condition>]**

**[ORDER BY <column-list>]**

**[GROUP BY <column-list>]**

**[HAVING <condition>]**

- (*[ ] - optional, | - or*)

# JEDNOSTAVNI SELECT

```
SELECT <columns>  
FROM <table>
```

<columns> mogu biti

- jedna kolona
- lista kolona,  
razdvojena zarezima
- \* za 'sve kolone'

- Neka je data tabela Student sa kolonama:
  - studID
  - studIme
  - studAdresa
  - studGodina

# JEDNOSTAVNI SELECT

Selektovanje svih podataka tabele *Student*.

```
SELECT * FROM Student
```

<i>studID</i>	<i>studIme</i>	<i>studAdresa</i>	<i>studGodina</i>
1	Andrić	Slobode 15	1
2	Brandić	Njegoševa 2	3
3	Čengić	Petra I 13	1
4	Andelić	Lipa 12	1
5	Erić	Zabjelo G4	2
6	Franović	Centar 11	3
7	Granić	Centar 18	1
8	Hristić	Hercegovačka 11	1

# JEDNOSTAVNI SELECT

Selektovanje kolone *studIme* iz tabele *Student*.

```
SELECT studIme FROM Student
```

<i>studIme</i>
Andrić
Brandić
Čengić
Andelić
Erić
Franović
Granić
Hristić

# JEDNOSTAVNI SELECT

Selektovanje kolona *studIme* i *studAdresa*, iz tabele *Student*.

```
SELECT stuIme, stuAdresa  
FROM Student
```

<i>studIme</i>	<i>stuAdresa</i>
Andrić	Slobode 15
Brandić	Njegoševa 2
Čengić	Petra I 13
Anđelić	Lipa 12
Erić	Zabjelo G4
Franović	Centar 11
Granić	Centar 18
Hristić	Hercegovačka 1



# BUDITE PAŽLJIVI

- Kada se koristi DELETE i UPDATE
  - Traba biti siguran da je WHERE uslov u redu.
  - Poželjno ga je, prije, provjeriti pomoću SELECT upita sa istim WHERE uslovom.

## Primjer:

Prije izvršavanja:

```
DELETE FROM  
Student
```

```
WHERE Year = 3
```

provjeriti sa:

```
SELECT * FROM  
Student
```

```
WHERE Year = 3
```

# SEKVENCE

- Često želimo svakom redu (zapisu) dodijeliti jedinstveni broj
  - Može se upotrijebiti kao primarni ključ.
  - Upotreba cjelobrojnih vrijednosti (integer) je efikasnija.
  - Želimo da to radi DBMS.
- U više verzija SQL (pa i MySQL SQL-u) koristi se **AUTO INCREMENT** polje.
  - Detalji se razlikuju u različitim verzijama.
  - Obično se za prvi unos dodjeljuje 1, za sljedeći 2, i tako dalje.

# SEKVENCE: PRIMJER

```
CREATE TABLE Laboratorija.insekti (  
    id INT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    ime VARCHAR(30) NOT NULL, # vrsta insekta  
    datum DATE NOT NULL, # datum prikupljanja  
    porijeklo VARCHAR(30) NOT NULL # odakle je prikupljen  
) COLLATE = utf8_slovenian_ci ;
```

```
INSERT INTO insekti (id,ime,datum,porijeklo) VALUES  
(NULL,'kućna muva','2001-09-10','kuhinja'),  
(NULL,'stonoga','2001-09-10','kolovoz'),  
(NULL,'skakavac','2001-09-10','dvorište');
```

```
SELECT * FROM insekti ORDER BY id;
```

id	ime	datum	porijeklo
1	kućna muva	2010-07-01	kuhinja
2	stonoga	2010-07-01	kolovoz
3	skakavac	2010-07-01	dvorište

```
SELECT LAST_INSERT_ID();
```

# SEKVENCE: REOZNAČAVANJE ZAPISA

U slučaju brisanje više zapisa može se pojaviti potreba za ponovnom dodjelom AUTO INCREMENT vrijednosti.

Treba biti oprezan ukoliko je tabela povezana sa drugim tabelama.

Sljedeći primjer pokazuje kako se to može uraditi:

```
ALTER TABLE insekti DROP id;  
ALTER TABLE insekti  
ADD id INT UNSIGNED NOT NULL PRIMARY KEY  
AUTO_INCREMENT FIRST;
```

# SEKVENCE: START OD ZADATE VRIJEDNOSTI

MySQL će startovati sekvencu od 1, ali se može zadati bilo koji drugi broj  
Slijedi primjer gdje MySQL startuje sekvencu od 100.

```
ALTER TABLE insekti AUTO_INCREMENT = 100;
```

# ZA VJEŽBU

Ponoću SQL upita, kreirati sljedeće tabele:

Numere

cID	Num	Naziv	Traj	uID
1	1	Violent	239	1
1	2	Every Girl	410	1
1	3	Breather	217	1
1	4	Part of Me	279	1
2	1	Star	362	1
2	2	Teaboy	417	2

CD

cID	Naslov	Cijena
1	Mix	9.99
2	Kompilacija	12.99

Umjetnici

uID	Ime
1	Stellar
2	Cloudboy

Tabelu Numere referencirati na tabele CD i Umjetnici.

## ZA VJEŽBU

Dodati 2.50€ na cijenu svih CD-ova koji staju više od 10.00€.

Dodati novu kolonu, Zavr, u CD tabeli. Ova kolona treba da sadrži string do 100 karaktera, i ako žanr nije definisan, treba upisati "Unknown".

Dodati numeru pod imenom "Running" djelo umjetnik "Speed" koja traje 12 minuti i 27 sekundi na CD-u "Kompilacija". U ovom slučaju, može se smatrati da je sadržaj tabela upravo kao što je pokazano na prethodnom slajdu.

# SLJEDEĆA LEKCIJA

- SQL SELECT
  - WHERE klauzula
  - SELECT iz više tabela
  - Povezivanje tabela