

BAZE PODATAKA

Jelena Žakula

Matematički fakultet, Beograd, Januar 2026.

Sadržaj

1 Rad sa tabelama i veze između njih	3
1.1 Uređenje polja i slogova u relaciji	3
1.2 Formatiranje podataka u tabeli	3
1.3 Opis referencijalnog integriteta	3
1.4 Izmene veze između tabela	3
2 Forme (obrasci)	3
2.1 Kreiranje formi (obrazaca) sa i bez čarobnjaka	3
2.2 Unos podataka pomoću formi	3
2.3 Dodavanje specijalnih kontrola formi (list-boksevi, kombo-boksevi, komandna dugmad i dr.)	3
2.4 Kreiranje multitabularnih formi	3
3 Pretraživanje i sortiranje	3
3.1 Traženje informacija u tabeli	3
3.2 Sortiranje, filtriranje i indeksiranje	3
4 Upiti	3
4.1 Osnovne SQL komande (definicione, kontrolne i manipulacione)	4
4.2 Kreiranje upita (sa i bez čarobnjaka)	4
4.3 Pregled rezultata upita	4
4.4 Kreiranje multitabularnih upita	7
5 Izveštaji	8
5.1 Kreiranje izveštaja (sa i bez čarobnjaka)	8
5.2 Pregled izveštaja	8
5.3 Postavljanje kontrola i izračunavanja u izveštajima	8
5.4 Kreiranje multitabularnih izveštaja	8
6 Vizuelizacija podataka iz baze	8
6.1 Komponente za povezivanje Windows-aplikacije sa bazom podataka	8
6.2 Vizuelne komponente za prikazivanje i modifikaciju podataka baze	8
6.3 Komponente za navigaciju	8

1. Rad sa tabelama i veze između njih

Podsećanja radi: Tabele predstavljaju osnovnu komponentu u relacionim bazama podataka. One omogućavaju organizaciju i skladištenje podataka na način koji olakšava upravljanje podacima i dobijanje korisnih informacija.

1.1. Uređenje polja i slogova u relaciji

1.2. Formatiranje podataka u tabeli

1.3. Opis referencijalnog integriteta

1.4. Izmene veze između tabela

2. Forme (obrasci)

2.1. Kreiranje formi (obrazaca) sa i bez čarobnjaka

2.2. Unos podataka pomoću formi

2.3. Dodavanje specijalnih kontrola formi (list-boksevi, kombo-boksevi, komandna dugmad i dr.)

2.4. Kreiranje multitabularnih formi

3. Pretraživanje i sortiranje

3.1. Traženje informacija u tabeli

3.2. Sortiranje, filtriranje i indeksiranje

4. Upiti

Upiti predstavljaju osnovni način komunikacije sa bazom podataka, omogućavajući korisnicima da pregledaju, dodaju, menjaju i brišu podatke prema svojim potrebama.

Upitni jezik SQL omogućava korisnicima da izvrše sve potrebne operacije nad bazom podataka, od kreiranja same baze i tabela u njoj, upisivanja, menjanja i pretraživanja podataka, do brisanja nekih podataka, tabela ili cele baze.

4.1. Osnovne SQL komande (definicione, kontrolne i manipulacione)

Komande jezika SQL se tradicionalno nazivaju upitima. Prva reč u svakom SQL upitu određuje vrstu akcije koju korisnik zahteva od sistema. Najčešće korišćeni upiti su:

- **Upiti SELECT**, koji služe da se iz baze pročitaju željeni podaci,
- **Upiti INSERT**, koji služe da se u bazu upišu novi podaci,
- **Upiti DELETE**, koji služe da se iz baze obrišu neki podaci, i
- **Upiti UPDATE**, koji služe da se neki podaci u bazi ažuriraju (izmene).

Od svih pomenutih tipova upita, za nas je najznačajniji upit **SELECT**, pomoću kojeg veoma brzo možemo da dođemo do podataka (*data*) koji su nam potrebni.

4.2. Kreiranje upita (sa i bez čarobnjaka)

Kreiranje upita u SQL-u može se obaviti ručno pisanjem upita ili korišćenjem alata poznatog kao čarobnjak (wizard). Čarobnjak pojednostavljuje formulisanje složenih upita, posebno kada je potrebno povezati više tabele ili primeniti filtere i sortiranja. Za naprednije korisnike, ručno pisanje upita omogućava veću fleksibilnost i preciznost u radu sa podacima.

4.3. Pregled rezultata upita

Rezultati izvršenih upita prikazuju se u tabelarnom obliku, što omogućava korisnicima da brzo analiziraju i interpretiraju podatke. Pregled rezultata može uključivati sortiranje, filtriranje i grupisanje podataka, što pomaže u identifikaciji važnih informacija i donošenju odluka.

S obzirom da je upravo **SELECT** upit osnovni način za čitanje podataka iz baze, sledeći primer prirodno ilustruje ovaj proces. **SELECT** upit određuje koje kolone i iz kojih tabela želimo da dobijemo podatke, a izvršavanjem upita dobijamo tabelarni prikaz koji možemo analizirati. Na ovaj način, veza između pisanja upita i pregleda rezultata postaje jasna i praktično primenljiva.

Svaki, pa i najjednostavniji upit **SELECT** mora da sadrži sledeće elemente:

```
SELECT kolone  
FROM tabele;
```

Ovim se sistemu nalaže da nam iz navedene tabele (ili više tabela) izdvoji sve podatke koji su dati u navedenim kolonama. Jezik nije osetljiv na razliku između velikih i malih slova (moguće je pisati i `select` i `from`), ali ćemo mi u skladu sa ustaljenom praksom, u nastavku sve ključne reči pisati velikim slovima. Sve beline u upitu se zanemaruju, pa samim tim nije bitno da li se upit piše u jednom ili više redova. Preglednosti radi, mi ćemo sve upite pisati u više redova.

Za početak ćemo uvek čitati podatke samo iz jedne tabele. Najjednostavniji slučaj je onaj u kome se želi čitanje svih podataka iz jedne tabele. Tada se umesto imena svih kolona može navesti samo simbol *. Razmotrimo sledeći primer. Tačka-zarez (;) može i ne mora da se doda na kraj upita.

Prikazati sve podatke o učenicima koji su upisani u bazi:

```
SELECT *
FROM ucenik;
```

Izvršavanjem upita dobija se sledeći rezultat:

id	ime	prezime	pol	datum_rodjenja	razred	odeljenje
1	Petar	Petrović	m	2006-07-01	1	1
2	Milica	Jovanović	ž	2006-04-03	1	1
3	Lidija	Petrović	ž	2006-12-14	1	1
4	Petar	Milovanović	m	2005-12-08	2	1
5	Ana	Pekić	ž	2005-02-23	2	1

Ovaj upit znači:

ODABERI sve kolone
IZ REDOVA tabele ucenik

On je funkcionalno ekvivalentan sledećem upitu, ali je od njega jednostavniji za pisanje:

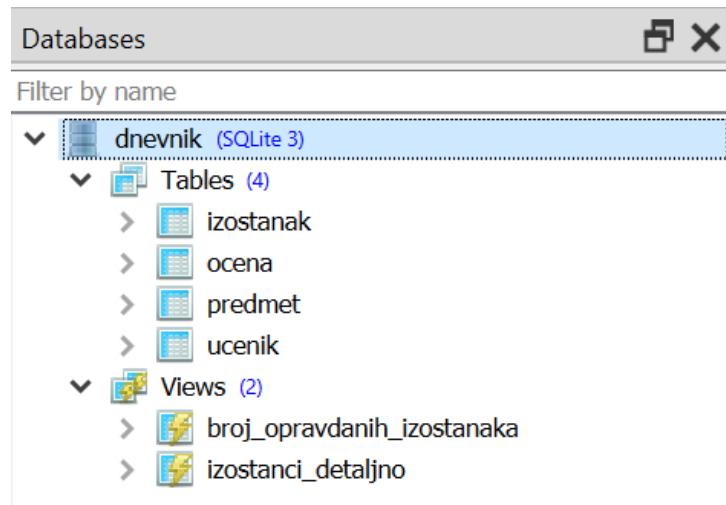
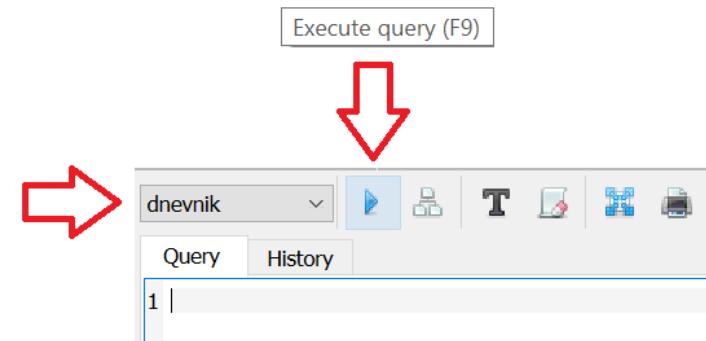
```
SELECT id, ime, prezime, pol, datum_rodjenja, razred, odeljenje
FROM ucenik;
```

Izvršavanjem upita dobija se sledeći rezultat:

id	ime	prezime	pol	datum_rodjenja	razred	odeljenje
1	Petar	Petrović	m	2006-07-01	1	1
2	Milica	Jovanović	ž	2006-04-03	1	1
3	Lidija	Petrović	ž	2006-12-14	1	1
4	Petar	Milovanović	m	2005-12-08	2	1
5	Ana	Pekić	ž	2005-02-23	2	1

Oblik upita **SELECT** koji smo upravo upoznali sadrži samo obavezne delove, pa je to najkraći mogući oblik ovog upita. Obradićemo različite složenije oblike upita **SELECT**, koji nam omogućavaju da od postojećih podataka odaberemo samo neke, da ih prebrojimo, da nađemo najmanji ili najveći podatak koji ispunjava neki uslov, da prikažemo jednostavne statistike po grupama podataka i slično.

U sistemu **SQLite Studio** se upiti pišu nakon što se klikne na kreiranu bazu **dnevnik** u prozoru **Databases** i potom izabere komanda menija **Tools → Open SQL Editor**. Kada se napiše upit, klikne se na dugme **Execute query (F9)** (plavi trouglica). Ukoliko se u prostoru za pisanje upita nalazi više njih, potrebno je obeležiti onaj koji želimo da pokrenemo. Ukoliko imamo više baza podataka, obavezno proveri da li je pored ovog dugmeta naziv baze u kojoj želiš da vršiš upite.



Često će nam kod upita biti potrebno da znamo i tačne nazive kolona. To možemo da vidimo za svaku tabelu pojedinačno tako što kliknemo na nju u prozoru **Databases**, pa se onda pojavi opis strukture tabele koji sadrži spisak svih kolona.

Spisak kolona možemo da vidimo i kada izvršimo osnovni **SELECT** upit:

	id	ime	prezime	pol	datum rodjenja	razred	odeljenje
1	1	Петар	Петровић	м	2006-07-01	1	1
2	2	Милица	Јовановић	ж	2006-04-03	1	1
3	3	Лидија	Петровић	ж	2006-12-14	1	1
4	4	Петар	Миловановић	м	2005-12-08	2	1
5	5	Ана	Пекић	ж	2005-02-23	2	1
6	6	Јован	Миленковић	м	2006-04-07	1	2

4.4. Kreiranje multitabularnih upita

Multitabularni upiti (multi-table queries) omogućavaju korisnicima da povežu podatke iz više tabela koristeći operacije kao što su **JOIN**, **INNER JOIN** i **LEFT JOIN**. Ovi upiti su posebno korisni kada je potrebno kombinovati informacije iz različitih delova baze podataka kako bi se dobio potpuniji uvid u podatke.

5. Izveštaji

- 5.1. Kreiranje izveštaja (sa i bez čarobnjaka)**
- 5.2. Pregled izveštaja**
- 5.3. Postavljanje kontrola i izračunavanja u izveštajima**
- 5.4. Kreiranje multitabularnih izveštaja**

6. Vizuelizacija podataka iz baze

- 6.1. Komponente za povezivanje Windows-aplikacije sa bazom podataka**
- 6.2. Vizuelne komponente za prikazivanje i modifikaciju podataka baze**
- 6.3. Komponente za navigaciju**