

7.6. Učitavanje podataka iz baze – SELECT sintaksa

Komanda SELECT ima sledeći opšti oblik:

```
SELECT kolone  
FROM tabele  
[WHERE uslovi]  
[ORDER BY kolone_za_sortiranje]  
[LIMIT broj];
```

Ovo nije potpuna sintaksa (potpuna sintaksa će biti objašnjena kasnije), ali ilustruje opšti oblik komande.

Komanda SELECT ima veliki broj neobaveznih odredaba. Ne moraju se uvek navoditi, ali ako se upotrebljavaju, moraju se zadavati redosledom koji je prikazan u opštem obliku komande.

7.6.1. Jednostavni upiti

Primer najjednostavnijeg oblika komande SELECT izgleda ovako:

```
SELECT * FROM firme;
```

Ako se ovaj upit izvrši sa podacima koji postoje u bazi podataka poslovanje trebalo bi da se dobiju rezultati nalik na sledeće:

firma	naziv_firme	mesto	adresa	telefon
1	BALKAN	Niš	Mokranjčeva 13	018-522-854
2	STIL	Beograd	Takovska 10	011-562-365
3	KOKOMAX	Niš	Dušanova 33	018-352-666
4	HELIO	Subotica	Nikole Tesle 55	027-555-125

Ovaj upit je učitao sve podatke iz zadate tabele – tj. vrednosti iz svih kolona za sve redove tabele *firme*.

Razume se, suština relacione baze podataka svakako nije u tome da daje sve podatke koji su u nju uneseni, već da omogući pronalaženje određenih podataka.

7.6.2. Učitavanje podataka iz određenih kolona

Znak * u prethodnom primeru upita znači „sve kolone tabele“. Umesto zvezdice mogu se zadati samo kolone iz kojih su potrebni podaci. To može biti samo jedna kolona, nekoliko kolona tabele ili čak sve kolone tabele poređane željenim redosledom. Nazive kolona treba zadati u obliku liste vrednosti razdvojenih zarezima.

Sledeći upit učitava samo vrednosti iz kolona *jed_mere* i *ime* za sve redove tabele *proizvodi*:

```
SELECT jed_mere, ime FROM proizvodi;
```

Ako se ovaj upit izvrši u bazi podataka poslovanje trebalo bi da se dobiju rezultati nalik na sledeće:

jed_mere	ime
kom.	Četka
lit.	Lak
m2	Stiropor
kg	Gips
kg	Destilovana voda
kom.	Šmirgla

Treba obratiti pažnju da su kolone prikazane redosledom koji je zadat u upitu, a ne redosledom kojim su definisane prilikom kreiranja tabele.

7.6.3. Apsolutna imena baza podataka i tabela

Dodatni oblik notacije koji bi korisnik trebalo da ima u vidu omogućava zadavanje apsolutnih imena baze podataka i tabele s kojom želi da radi. Kolonu *naziv_firme* tabele *firme* je moguće navesti u upitu kao *firme.naziv_firme*.

```
SELECT firme.naziv_firme
FROM firme;
```

Trebalo bi da rezultati ovog upita budu nalik na sledeće:

<i>naziv_firme</i>
BALKAN
STIL
KOKOMAX
HELIO

7.6.3. Alijasi

Kolonama i tabelama u komandi SELECT mogu se dodeliti drugačija imena, koja će se prikazivati u rezultatima. Na primer, može se upotrebiti sledeći upit:

```
SELECT naziv_firme AS naziv
FROM firme;
```

U ovom upitu kolona *naziv_firme* je preimenovana u *naziv*, ali samo u kontekstu ovog upita. Rezultati izvršavanja ovog upita u bazi podataka poslovanje izgledaju ovako:

<i>naziv</i>
BALKAN
STIL
KOKOMAX
HELIO

Kao što se može videti, sadržaj kolone *naziv_firme* sada je prikazan ispod zaglavlja *naziv*. Identifikator *naziv* poznat je kao alijas (*alias*).

Navedeni primer alijasa nije naročito koristan. Međutim, prava moć alijasa će biti shvaćena tek kad počnu da se pišu složeniji upiti i upiti u kojima se nešto izračunava.

Alijasi se mogu zadavati i za tabele, kao u sledećem primeru:

```
SELECT f.naziv_firme
FROM firme AS f;
```

Trebalo bi da rezultati ovog upita budu isti kao da je napisan bez upotrebe alijasa. Ovaj način notacije postaće koristan kada se kasnije počnu izvršavati upiti koji obuhvataju više tabele.

7.6.4. Upotreba odredbe WHERE za učitavanje samo određenih redova

Učitavanje samo određenih redova je korisno jer često treba da se iz jedne ili više tabele učitaju samo zapisi koji ispunjavaju određene uslove. Ta mogućnost postaje još važnija kada je potrebno učitati samo nekoliko traženih redova iz veoma obimne tabele.

To se može obaviti pomoću odredbe WHERE komande SELECT. Jednostavan primer bio bi sledeći:

```
SELECT naziv_firme, adresa, telefon
FROM firme
WHERE mesto = 'Niš';
```

Kao što se može videti i tekst upita se može rasporediti u više redova zbog čitljivosti. Rezultati izvršavanja ovog upita u bazi podataka poslovanje izgledaju ovako:

naziv_firme	adresa	telefon
BALKAN	Mokranjčeva 13	018-522-854
KOKOMAX	Dušanova 33	018-352-666

U odredbi WHERE bio je zadat uslov zbog kojeg se izdvajaju samo redovi tabele koji ga ispunjavaju – u ovom primeru to su firme iz Niša.

Treba obratiti pažnju da je u upitu bio kombinovan uslov sa kolonama koje su bile potrebne.

U ovom primeru u odredbi WHERE bilo je zadato ispitivanje jednakosti. U SQL-u znak = služi za ispitivanje jednakosti.

Postoji veliki broj funkcija koje se mogu zadati u odredbi WHERE i koje će kasnije biti detaljno objašnjene. Zasad će biti navedeni samo operatori koji se najčešće koriste:

- Jednakost, ili =
- Nejednakost (različitost), koja se piše kao != ili <>
- Sve kombinacije > (veće od), < (manje od), >= (jednako ili veće od) i <= (jednako ili manje od)
- IS NULL ili IS NOT NULL, pomoću kojih se ispituje da li određena vrednost jeste ili nije NULL
- Uobičajeni aritmetički operatori koji se najčešće koriste u kombinaciji sa operatorima za poređenje *neka_vrednost* > *neka_druga_vrednost* * 10
- Standardni logički operatori AND, OR i NOT, koji se koriste za povezivanje više uslova. Imaju niži prioritet od operatora za poređenje. Primer: *plata* > 30000 AND *plata* < 50000.

Osim operatora u nekim primerima biće korišćene i funkcije. Funkcija COUNT() omogućava prebrojavanje redova koje je upit učitao. Kod funkcija je neophodno da se otvorena zagrada stavi neposredno iza naziva funkcije. Na primer:

```
SELECT COUNT(*) FROM firme;
```

Rezultati izvršavanja ovog upita u bazi podataka poslovanje izgledaju ovako:

COUNT(*)
4

Ovaj upit prikazuje koliko redova sadrži tabela *firmе*.

Standardni redosled prioriteta izračunavanja delova izraza se može menjati tako što se grupišu između zagrada.

Sledeći primer je nešto složeniji upit u kome je zadata odredba WHERE:

```
SELECT * FROM fakture
WHERE ulaz_izlaz = 1 AND datum > '2006-10-23';
```

Rezultati izvršavanja ovog upita u bazi podataka poslovanje izgledaju ovako:

sifra_fakture	sifra_firme	datum	ulaz_izlaz
1	2	2006-10-25	1
2	4	2006-10-28	1
5	3	2006-10-25	1

Ovaj upit prikazuje podatke za sve ulazne fakture koje su formirane posle 23. oktobra 2006. godine.

Važno je istaći da u odredbi WHERE nije dozvoljena upotreba alijasa za kolone, već se mora navesti izvorno ime kolone.

7.6.5. Uklanjanje dupliranih vrednosti pomoću opcije DISTINCT

Pomoću rezervisane reči DISTINCT korisnik navodi da u rezultatima upita ne želi da vidi duplirane vrednosti. Izvršavanjem upita:

```
SELECT mesto FROM firme;
```

dobijaju se sledeći rezultati:

mesto
Niš
Beograd
Niš
Subotica

Treba obratiti pažnju da se podatak Niš pojavljuje dva puta. Upit je prikazao sve vrednosti iz kolone *mesto* tabele *firme*.

Upit formulisan na ovaj način:

```
SELECT DISTINCT mesto FROM firme;
```

daje sledeće rezultate:

mesto
Niš
Beograd
Subotica

U ovom primeru duplikati se ne pojavljuju.

U ovom slučaju razlika ne izgleda tako značajna. Razlika bi bila приметnija u tabeli sa velikim brojem podataka koji se ponavljaju.

Upit:

```
SELECT COUNT(mesto) FROM firme;
```

vraća broj vrednosti u koloni *mesto* tabele *firme* ne računajući NULL vrednosti:

COUNT(mesto)
4

Ako se želi videti broj različitih vrednosti u koloni *mesto* tabele *firme* treba napisati sledeći upit:

```
SELECT COUNT(DISTINCT mesto) FROM firme;
```

COUNT(DISTINCT mesto)
3

7.6.8. Sortiranje učitanih rezultata pomoću odredbe ORDER BY

Odredba ORDER BY omogućava sortiranje rezultujućih redova po jednoj ili više kolona. Redosled sortiranja može biti rastući, što se označava sa ASC, ili opadajući, što se označava sa DESC (od *descending*). Izvršavanjem upita:

```
SELECT *
FROM firme
ORDER BY mesto ASC, naziv_firme DESC;
```

dobijaju se sledeći rezultati:

firma	naziv_firme	mesto	adresa	telefon
2	STIL	Beograd	Takovska 10	011-562-365
3	KOKOMAX	Niš	Dušanova 33	018-352-666
1	BALKAN	Niš	Mokranjčeva 13	018-522-854
4	HELIO	Subotica	Nikole Tesle 55	027-555-125

Ovaj upit učitava vrednosti iz svih kolona za sve redove tabele *firme*. Rezultati će biti sortirani rastuće po vrednostima u koloni *mesto*, a ako postoje dve ili više firmi iz istog mesta one će biti sortirane opadajuće po nazivu firme.

Ako se za kolonu zada ORDER BY bez opcije ASC ili DESC, podrazumeva se ASC.

7.6.9. Ograničavanje broja redova rezultata pomoću odredbe LIMIT

Odredba LIMIT ograničava broj redova rezultata koje upit daje. Izvršavanjem upita:

```
SELECT *
FROM proizvodi
LIMIT 3;
```

dobijaju se sledeći rezultati:

maticni_broj	ime	jed_mere
1	Četka	kom.
2	Lak	lit.
3	Stiropor	m2

Ovaj upit daje prva tri reda koji ispunjavaju zadati uslov. U ovom slučaju to su prva tri reda učitana iz tabele *proizvodi*.

Može se za učitavanje zadati i podskup redova drugačiji od prvih n. Ako se pomoću prethodnog upita žele učitati redovi od 4 do 6, to će se uraditi na sledeći način:

```
SELECT *
FROM proizvodi
LIMIT 3, 3;
```

maticni_broj	ime	jed_mere
4	Gips	kg
5	Destilovana voda	kg
6	Šmirgla	kom.

Kada se u odredbi LIMIT zadaju dva parametra, prvi je relativni pomak (red od kojeg počinje učitavanje), a drugi je maksimalan broj redova koji se želi učitati. Kada postoji

samo jedan parametar on predstavlja maksimalan broj redova koji se želi učitati. Kada se zadaje pomak, on počinje od 0 (za četvrti red je zadat pomak 3, a pomak za prvi red je 0). Ako se želi da upit vrati redove od pomaka do kraja tabele za vrednost drugog parametra treba navesti neki jako veliki broj.

Odredba LIMIT se najčešće koristi u kombinaciji sa odredbom ORDER BY da bi redosled redova u rezultatima upita imao određeni smisao. Treba imati na umu da bez odredbe ORDER BY, redosled redova rezultata nije predvidiv.

Ova odredba je naročito korisna u Web ili GUI aplikacijama koje koriste MySQL jer omogućava jednostavan mehanizam podele rezultata na stranice.