

## Prijava na bazu podataka - SQLTools

Hostname: ???  
 Username : ???  
 Password: ???  
 Host string: **etflab**

© Emir Buza

## SQL komande

*SELECT* je SQL komanda koja omogućava vraćanje podataka iz baze podataka. *SELECT* komanda ne mijenja podatke u bazi podataka. Ona isključivo vraća podatke iz baze podataka. Aritmetičke operacije definisane nad kolonama tabele, u select iskazu nad kojom se radi select SQL komanda, neće promijeniti vrijednosti podataka te tabele u bazi podataka.

DML izkazi su SQL komande koje omogućavaju manipulaciju podataka u bazi podataka.

SQL iskazi se dijele na:

<i>DML</i> – Data Manipulation Language	su SQL komande koje omogućavaju manipulaciju sa podacima u bazi podataka; komandama kao što su : INSERT, UPDATE, DELETE.
<i>DDL</i> - Data Definition Language	su SQL komande koje omogućavaju kreiranje (create), mijenjanje (alter), i/ili brisanje objekata u bazi podataka sa komandama kao što su CREATE TABLE, DROP TABLE, ALTER TABLE, itd.
<i>DCL</i> – Data Control Language	su SQL komande koje omogućavaju spašavanje, vraćanje i/ili vraćanje stanja baze podataka nakon neke upotrebe DML SQL komande nad podacima u bazi od/do početka prijave na bazu podataka. Tu su komande COMMIT, ROLLBACK, SAVEPOINT.
<i>TC</i> – Transaction Control	su SQL komande koje vam omogućavaju dodjeljivanje prava nad objektima baze podataka. To su komande GRANT, REVOKE.

## SELECT iskaz – SELECT SQL komanda

Osnovni SQL iskaz može uključiti sljedeće:

<i>SELECT</i> klauzula	specificira kolone koje će se prikazati
<i>FROM</i> klauzula	specificira tabele koje sadrže kolone koje će se predstaviti predstavljene u <i>SELECT</i> klauzuli.

U sintaksi:

<i>SELECT</i>	je lista jedne ili više kolona
<i>DISTINCT</i>	ukidanje duplikata
*	odabiranje svih kolona
<i>Column</i>	odabir naziva kolone
<i>Alias</i>	ispisuje naziv kolone u zaglavlju definisan imenom alijasa
<i>FROM table</i>	specificira tabela u kojoj se nalaze kolone

## Preimenovanje kolona

Kao što je naglašeno u tekstu kod osnove strukture select iskaza, kolone koje su odabrane za prikaz će biti predstavljene na ispisu u zaglavlju istim imenom kao što se nazivaju i u datičnoj tabeli koja je predstavljena u select iskazu. Najčešće ovakav način prikaza kolona nije adekvatan, jer se kolone u tabeli predstavljaju nazivima koje u pojedinim situacijama ne moraju predstavljati neku logičnost u opisu. Iz tih razloga potrebno je date kolone preimenovati, nekim logičnim imenom – alijasom, tj. da se prilikom ispisa te kolone predstave imenima koje bolje opisuju datu kolonu u zaglavlju odabrane tabele. Za definisanje alijasa kolone u tabeli kod većine baza podataka potrebno je koristiti ključnu riječ «as», dok to kod Oracle baze podataka može biti i izostavljeno.

```
SQL> SELECT last_name AS prezime,
        first_name AS ime,
        salary AS plata
FROM employees;
```

U slučaju da se nazivi alijasa sastoje od dvije ili više riječi, tada je u tom slučaju naziv alijasa potrebno zatvoriti između dvostrukih znaka navoda.

```
SQL> SELECT last_name "Puno prezime zaposlenog"
FROM employees;
```

## Aritmetički izrazi

Upotreba aritmetičkih izraza u select iskazu je dozvoljana. Aritmetički izrazi napisani u select iskazu se ponašaju po pravilima aritmetičkih operacija. U slučaju kombinovanog aritmetičkog izraza koji sadrži: stepenovanja, korijenovanja, množenja, dijeljenja, sabiranja i oduzimanja, tj. da bi se postigao odgovorajući rezultat navedene aritmetičke izraze je potrebno zatvoriti u odgovarajuće zagrade u skladu sa očekivanim rezultatima aritmetičkog izraza. Aritmetičke operacije između kolona i konstanti su regularni dio aritmetičkih operacija u select iskazu i dozvoljen su bez obzira na broj i količina operacije koje je potrebno izvršiti.

Na primjer, predpostavimo da je potrebno uvećati platu svih zaposlenih za 10% i prikazati kao rezultat ime, prezime, platu i uvećanu platu za tih 10%.

```
SQL> SELECT last_name prezime,  
           first_name ime,  
           salary plata,  
           salary * 1.1 "Plata uvećana za 10% "  
FROM employees;
```

U slučaju da je potrebno izvršiti samo određenu funkciju i/ili aritmetički izraz tada se najčešće koristi DUAL tabela. Dual table će biti objašnjena poslije.

```
SQL> SELECT ((100 + 5.678) * (200 + 2.45678)) / 234.4567 total  
FROM dual;
```

## NULL vrijednost

*NULL* vrijednost znači odsustvo vrijednosti. Treba zapamtiti da ako neki broj sabiramo, oduzimamo, množimo ili dijelimo sa *NULL* kao rezultat te aritmetičke operacije će biti *NULL* vrijednost.

```
SQL> SELECT 100 + null "Null + 100"  
FROM dual;
```

Napomenimo da prilikom izbora kolona koje sadrže null vrijednost, tj. ispisa dotične kolone na ekranu, nakon izvršenja upita rezultati će se pojaviti sa odsustvom vrijednosti (prazno polje). Neki alati će u slučaju odsustva vrijednosti prikazati "(null)" kao odsustvo vrijednosti. Primjer kolone koja sadrži null vrijednost za pojedine zaposlene, u okviru tablele employees, je dodatak na platu (commission\_pct) u tabeli zaposlenih .

```
SQL> SELECT last_name,  
           first_name,  
           commission_pct "Dodatak na platu u procentima (%)"  
FROM employees;
```

## NVL funkcija

NVL funkcija mijenja null vrijednost u neku zadatu konkretnu vrijednost. U slučaju dodatka na platu, ako se želi ispisati 0 za one zaposlene koji nisu imali dodatak na platu, select će izgledati kako slijedi:

```
SQL> SELECT last_name prezime,  
           First_name ime,  
           NVL(commission_pct,0) "Dodatak na platu"  
FROM employees;
```

## Spajanje kolona u jedno kolonu

Da bi se spojio jedan ili više stringova ili kolona koriste se dvije uspravne linije «||». U primjeru koji slijedi izvršeno je spajanje prezimena i imena zaposlenog, a kolona je prikazana kao jedna kolona «Naziv zaposlenog».

```
SQL> SELECT last_name || ' ' || first_name "Naziv zaposlenog"
      FROM employees;
```

## ORDER BY klauzula

Order by klauzula se koristi za sortiranje jedne ili više kolona u upitu. Postoje dva načina sortiranja i to po rastućem (*ASC*-ending) i opadajućem (*DESC*-ending) poretku. U order by klauzuli mogu se navesti samo one kolone koje su navedene u select iskazu. Po default-u je *ASC* poredak sortiranja, tako da ako se ne kaže eksplicitno koji način sortiranja se traži od baze podataka, baza će za navedene kolone izvršiti sortiranje u *ASC* poretku.

```
SQL> SELECT last_name,
      first_name,
      FROM employees
      ORDER BY last_name ASC;
```

```
SQL> SELECT last_name,
      first_name,
      FROM employees
      ORDER BY last_name DESC;
```

## WHERE klauzula

Where klauzula se koristi za filtriranje podataka koji se dohvataju iz baze po nekom uslovu ili kriteriju pretrage u zavisnosti od potrebnog uslova pretraživanja podataka u bazi.

U sintaksi:

<i>WHERE</i>	je restrikcija nad upitom na osnovu koje će se izvršiti pretraživanje podataka u bazi podataka
<i>uslov</i>	se sastoji od naziva kolona, izraza, konstanti i operatora poređenja

U where klauzuli mogu se porediti vrijednosti u odnosu na kolonu/kolone, karakteristične vrijednosti (npr. konstantne vrijednosti), aritmetički izraze i/ili funkcije i slično. Svaka where klauzula sadrži tri elementa i to:

- naziv kolone
- operator poređivanja i
- naziv kolone, konstante i/ili liste vrijednosti.

Na primjer, neka je potrebno napisati upit koji će vratiti ime, prezime, broj telefona i šifru odjela za sve zaposlene koji rade na poslovima – IT\_PROG.

```
SQL> SELECT first_name,  
        last_name,  
        phone_number,  
        department_id  
FROM employees  
WHERE job_id = 'IT_PROG';
```

## LIKE operator

*LIKE* operator se koristi za pretraživanje vrijednosti kolona ili stringovnih konstanti kada se ne poznaju egzaktna vrijednosti kolona nad kojima se vrši pretraživanje. Dakle za slučaj operatora «=» vrši se povezivanje, odnosno pretraživanje poredbene vrijednosti i vrijednosti za upoređivanje u odnosu 1:1, dok se kod like operatora najčešće traži string ili niz stringova koji postoje u poredbenoj vrijednosti.

Kod sastavljanje stringovne vrijednosti *LIKE* operator za poređenje koristi dva predefinisana karaktera i to:

- **%** - mijenja sekvencu 0 ili više karaktera
- **\_** - mijenja tačno jedan karakter.

Tako na primjer, ako je potrebno napisati upit koji će prikazati sve zaposlene čiji ime počinje sa slovom A, upit bi izgledao kako slijedi:

```
SQL> SELECT first_name  
FROM employees  
WHERE first_name LIKE 'A%';
```

Sljedeći primjer prikazuje slučaj kada je potrebno napisati upit koji prikazuje sve zaposlene čije ime sadrži na bilo kom mjesto slovo «m» i završava se sa slovom «a».

```
SQL> SELECT first_name  
FROM employees  
WHERE first_name LIKE '%m%a';
```

## BETWEEN operator

Between je operator koji se koristi za poređenje vrijednosti kolone sa unaprijed predefinisanim opsegom namjanje i najveće vrijednosti. Na primjer, pretpostavimo da je potrebno prikazati sve one zaposlene koji dobivaju plata između 1000 i 3000.

```
SQL> SELECT first_name  
FROM employees  
WHERE salary BETWEEN 1000 AND 3000;
```

## IS NULL operator

Poređenje null vrijednosti u bazi podataka ne može se ostvariti preko operatora «=», nego preko ključnih riječi *IS NULL* ili *IS NOT NULL*. Tako na primjer, ako se trebaju prikazati svi zaposleni koji ne dobivaju dodatak na platu upit bi izgledao kako slijedi:

```
SQL> SELECT first_name  
      FROM employees  
      WHERE commission_pct IS NULL;
```

## Zadaci

1. Napisati upit koji će prikazati naziv zaposlenog i platu za sve zaposlene koji imaju platu veću od 2456.
2. Napisi upit koji će prikazati naziv zaposlenog i šifru odjela za šifru zaposlenog 102.
3. Napisati upit koji će prikazati sve zaposlene čija plata nije u rangu od 1000 do 2345.
4. Napisati upit koji će prikazati naziv zaposlenog (predstavljeno kao jedna kolona) «Zaposleni», posao i datum zaposlenja za sve zaposlene koji su počeli da rade u periodu od 11.01.1996 do 22.02.1997.
5. Napisati upit koji će prikazati naziv zaposlenog i šifru odjela za sve zaposlene iz odjela 10 i 30 sortirano po prezimenu zaposlenog.
6. Napisati upit koji će prikazati platu, ime, prezime i dodatak na platu za sve zaposlene koji imaju platu veću od 1500 i rade u odjelima 10 ili 30. Za labelu kolona uzeti respektovno: mjesečna plata, ime zaposlenog, prezime zaposlenog i dodatak na platu.
7. Napisati upit sve zaposlene koji su počeli da rade prije 1996 godine.
8. Napisati upit koji će prikazati naziv, platu i posao zaposlenog za sve zaposlene koji nemaju nadređenog.
9. Napisati upit koji će prikazati naziv zaposlenog, platu i dodatak na platu za sve one zaposlene koji su stekli dodatak na platu. Sortirati podatke u opadajućem poretку po plati i dodatku na platu.
10. Napisati upit koji će prikazati naziv zaposlenog za sve one zaposlene koji imaju dva slova «l» u nazivu (naziv se sastoji od imena i prezimena zaposlenog).
11. Napisati upit koji će prikazati naziv, platu i dodatak na platu za sve zaposlene čiji je iznos dodatka na platu veći od plate zaposlenog umanjene za 80%.