

IV SQL

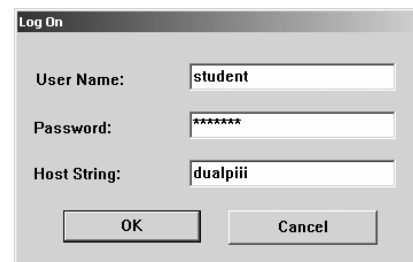
SQL (Structured Query Language) je jezik koji je Američki Institut za Nacionalne Standarde (ANSI - American National Standards Institute) prihvatio kao standardni jezik za relacione baze podataka. Takođe, prihvatila ga je i Internacionalna organizacija za standarde (ISO - International Standards Organization). Većina relacionih baza podataka (poput Oracle baze) jedino preko SQL-a dozvoljava pristup podacima. Ovo je i jedan od razloga potrebe da većina korisnika baza podataka upozna komande SQL-a. SQL omogućava korisnicima pristup podacima u sistemima za upravljanje relacionim bazama podataka (npr. Oracle, Sybase, Informix, Microsoft SQL Server).

SQL*Plus je jedan od ORACLE proizvoda koji služi za upravljanje podacima iz ORACLE baze podataka. U SQL*Plusu izvode se sve SQL i PL/SQL naredbe i izrazi. On predstavlja jedan "interface" između baze podataka i programa (ili izraza) koji manipuliše bazom. Takođe, osim SQL i PL/SQL naredbi, SQL*Plus poseduje i skup svojih naredbi koje omogućuju dodatne akcije nad bazom podataka.

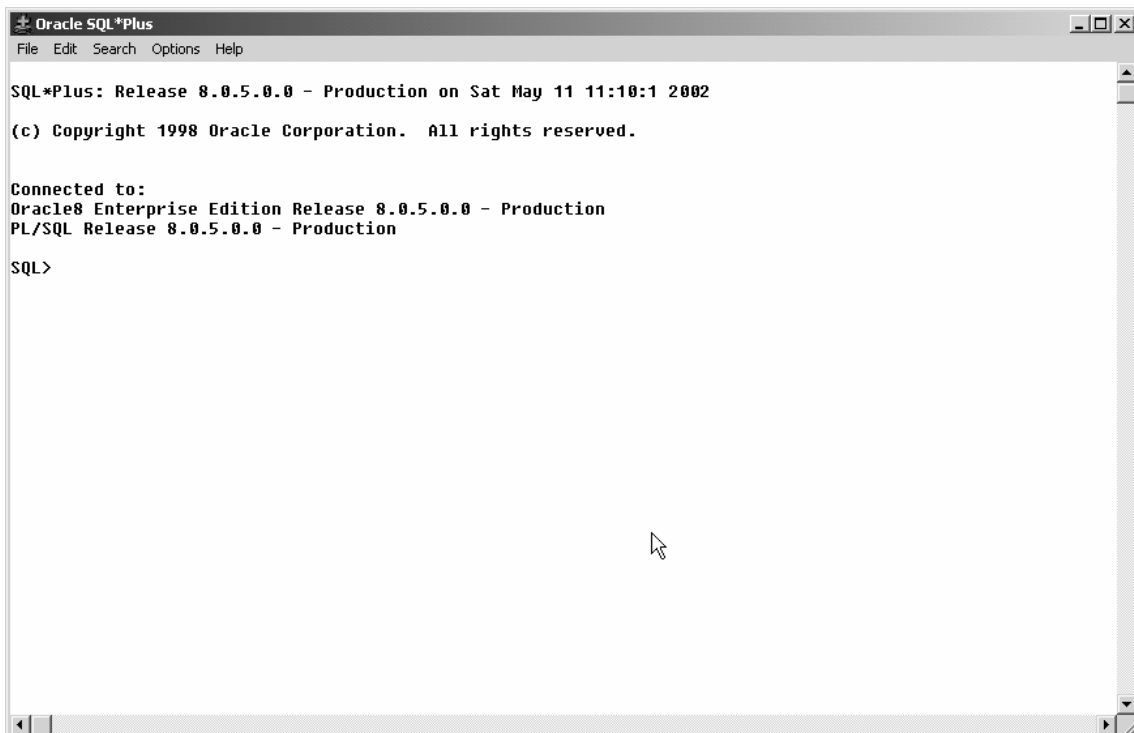


SQL Plus 8.0
Slika 1. SQL*Plus ikona

Pošto je SQL*Plus program, on se pokreće iz Windowsa izborom odgovarajuće ikone (Slika 1) ili iz Programs grupe. Po pokretanju programa sledi identifikacija korisnika (Slika 2). Unose se: ime korisnika (User name), šifra (Password) i naziv servera (Host String). Ime i šifra identični su onima koje dodeljuje administrator sistema korisnicima prijavljenim za rad sa bazom podataka. Kada korisnik na ovaj način uđe u SQL*Plus program on je automatski spojen sa bazom podataka.



Slika 2. Dijalog provere identifikacije korisnika



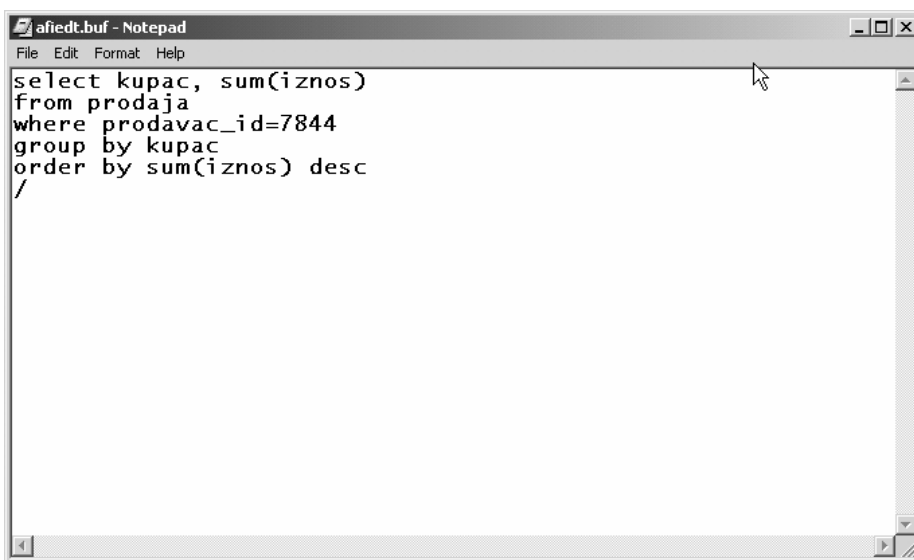
Slika 3. Prozor SQL*Plus programa

Kada je korisnik ispravno prijavljen za rad sa SQL*Plus programom, dobija "prompt" znak, koji mu daje mogućnost unosa naredbi i rada sa samim programom (Slika 3). Korisnik sada može videti sve objekte i upravljati njima (objekta nad kojima ima dodeljene privilegije). U SQL*Plus-u mogu se unositi i izvršavati sledeći skup naredbi:

- SQL naredbe,
- PL/SQL blokovi i
- SQL*Plus naredbe.

SQL*Plus poseduje editor za unos naredbi. Po završetku unosa jednog reda naredbe i pritiskom na taster Enter prelazi se u novi red naredbe. Svaka linija naredbe ima na svom početku redni broj. Svaka SQL naredba završava sa znakom ";". Kada SQL*Plus uoči ovaj znak, on počinje sa izvršavanjem izraza koji je unet. Kada se SQL izraz izvrši, rezultat se automatski prikazuje u nastavku.

SQL*Plus koristi i sistemski editor koji omogućuje lakši unos i izmenu naredbi (Slika 4). Zadnje upisan SQL izraz nalazi se u posebnom memorijskom području tzv. SQL buffer-u. Pozivanjem sistemskog editora naredbom edit iz linije SQL*Plus-a automatski se učitava poslednji SQL izraz (izraz smešten u SQL buffer-u). Najčešće, sistemski editor u Windows operativnom sistemu je Notepad program. Rad sa njim je dosta jednostavan i prikazan je ranije. SQL izrazi uneti preko sistemskog editora nemaju ";" kao znak za kraja. Umesto njega koristi se znak "/" (Slika 4). Posebno treba voditi računa da se između ovog znaka i poslednjeg reda SQL izraza ne nađe ni jedan drugi znak (prazan red npr.). Svaka izmena SQL izraza mora da se zapamti (Edit-> Save), izade iz Notepad-a (Edit-> Exit) i izvrši u SQL*Plus-u (unositi se run ili samo r).



```
afiedt.buf - Notepad
File Edit Format Help
select kupac, sum(iznos)
from prodaja
where prodavac_id=7844
group by kupac
order by sum(iznos) desc
/
```

Slika 4. SQL*Plus sistemski editor tj. Notepad program

1. OSNOVE SQL-a

SQL je neproceduralni jezik tj. severu su samo kaže šta da uradi, ali ne i kako to da uradi. Server baze podataka prevodi komande SQL-a u interne procedure koje obrađuju zahtev. SQL je jednostavan zato što skriva sve detalje obrade podataka. Oracle baze podataka sadrže i proceduralna proširenja SQL-a i tako objedinjuje prednosti oba tipa jezika. Proceduralno proširenje Oracle baze podataka naziva se PL/SQL.

Iskazi SQL jezika mogu se podeliti na četiri grupe. To su iskazi za:

1. **Pretraživanje** - iskazi samo omogućuju prikazivanje postojećih podataka u bazi. Iskazi uvek počinju ključnom rečju SELECT iza koje slede željeni podaci, iskazi i sl. Iskazi pretraživanja ne menjaju podatke u bazi.

2. **Manipulaciju** - jezik za manipulaciju podacima (DML - Data Manipulation Language). DML iskazi koriste se za menjanje podataka u bazi.
3. **Definisanje** - jezik za definisanje podataka (DDL - Data Definition Language). DDL iskazi koriste se za kreiranje ili brisanje objekata baze podataka.
4. **Kontrolu** - jezik za kontrolu podataka (DCL - Data Control Language). DCL iskazi koriste se za dodelu ili oduzimanje privilegija za pristup bazi podataka.

Za analitičke upite koriste se iskazi pretraživanja. U daljem tekstu biće zato prikazani osnovni elementi SQL izraza za pretraživanje baza podataka.

Komande SQL-a mogu se pisati i velikim i malim slovima.

```
SELECT PREZIME  
FROM RADNIK  
WHERE IME='FRANCIS'
```

ili

```
select prezime  
from radnik  
where ime='FRANCIS'
```

Ključne reči

U ovom primeru ključne reči su SELECT, FROM i WHERE.

U oba primera reč FRANCIS se piše velikim slovima. Za razliku od ključnih reči SQL-a, koje se mogu pisati i malim i velikim slovima, podaci se moraju napisati onako kako su dati u bazi podataka. Da bi se izbegle greške ovog tipa najbolje je da se korisnik najpre upozna sa načinom zapisa podataka u bazi.

Svaki upit mora da ima znak kojim se obeležava njegov završetak. U ovoj implementaciji znak za kraj upita je tačka-zarez(;).

Prvi upit

```
SQL> select * from posao;
```

Izlaz:

```
POSAO_ID FUNKCIJA  
-----  
        667 CLERK  
        668 STAFF  
        669 ANALYST  
        670 SALESPERSON  
        671 MANAGER  
        672 PRESIDENT
```

6 rows selected.

Primer 2.

```
SQL> select * from odelenje;
```

Izlaz:

ODELJENJE_ID	IME	LOKACIJA_ID
10	ACCOUNTING	122
20	RESEARCH	124
30	SALES	123
40	OPERATIONS	167
12	RESEARCH	122
13	SALES	122
14	OPERATIONS	122
23	SALES	124
24	OPERATIONS	124
34	OPERATIONS	123
43	SALES	167

11 rows selected.

U ovom SQL upitu znak "zvezdica" (*) kaže bazi podataka da vrati informacije iz svih kolona tabele navedene u FROM klauzuli. Baza podataka određuje u kom redosledu će biti vraćen sadržaj kolona.

U drugom primeru možemo primetiti da kolone 1 i 3 (ODELJENJE_ID i LOKACIJA_ID) imaju desno poravnanje, a da kolona 2 (IME) ima levo poravnanje. Po prihvaćenoj konvenciji numerički podaci imaju desno poravnanje a znakovni podaci levo poravnanje (ista konvencija važi i u MS Excelu).

Promena redosleda kolona

Posmatrajmo sledeća dva upita:

```
SQL> select * from posao;
```

i

```
SQL> select posao_id, funkcija from posao;
```

Na oba upita dobija se isti odgovor baze podataka:

PSAO_ID	FUNKCIJA
667	CLERK
668	STAFF
669	ANALYST
670	SALESPERSON
671	MANAGER
672	PRESIDENT

6 rows selected.

U drugom upitu iza ključne reči select navedeni su nazivi svih kolona u tabeli posao. Ovaj način postavljanja upita omogućuje korisniku kako da promeni redosled ispisa kolona, tako i izostavljanje neke kolone u odgovoru baze. Imena kolona međusobno su odvojena zarezima.

Primer 1.

```
SQL> select funkcija from posao;
```

Izlaz:

```
FUNKCIJA
-----
CLERK
STAFF
ANALYST
SALESPERSON
MANAGER
PRESIDENT
```

6 rows selected.

Primer 2.

```
SQL> select ime, prezime, zarada
      from radnik
```

Izlaz (deo):

IME	PREZIME	ZARADA
PAUL	ROSS	1300
FRANCIS	KING	5000
MARY	TURNER	1500
DIANE	ADAMS	1100
FRED	JAMES	950
JENNIFER	FORD	3000
GRACE	ROBERTS	2875
MICHAEL	DOUGLAS	800
BARBARA	MILLER	1300
ALICE	JENSEN	750
JAMES	MURRAY	750

32 rows selected.

U ovom primeru FROM klauzula je prenetu u drugi red. Rezultat upita je isti, jedino je način zapisa promenjen.

Izbor pojedinih kolona

Već smo u Primeru 2. videla da možemo izvršiti upit samo po pojedinim kolonama tabele. Ovaj način nam omogućuje i promenu redosleda ispisa kolona.

Izbor različitih tabela

Izbor tabele čiji će sadržaj biti prikazan određen je upotrebom kluzule FROM. Iza ove klauzule navodi se ime tabele.

Primer 1.

```
SQL> select * from lokacija;
```

Izlaz:

```
LOKACIJA_ID REGIONALNA_GRUPA
-----
122 NEW YORK
124 DALLAS
123 CHICAGO
167 BOSTON
```

U ovom primeru prikazan je sadržaj svih kolona tabele LOKACIJA.

Upit za eliminisanje duplikata

```
SQL> select drzava from kupac;
```

Izlaz:

```
DR
--
CA
CA
CA
CA
CA
CA
CA
CA
CA
MN
...
TX
MA
TX
NY
TX
TX
MA
TX
TX
NY
MA
NY
```

33 rows selected.

Poslednji red odgovora na upit (33 rows selected) kaže da je ispisano 33 redova tabele KUPAC. Iz odgovora se očigledno vidi da se pojedini elementi ponavljaju više puta. Upit možemo modifikovati tako da nam u rezultat upita bude prikazano samo po jedno pojavljivanje svakog elementa željene kolone. U tu svrhu koristi se ključna reč DISTINCT koja se navodi iza reči SELECT.

Primer 1.

```
SQL> select distinct drzava from kupac;
```

Izlaz:

```
DR
--
CA
MA
MN
NY
```

TX

Samo je pet različitih skraćenica za države uneto u tabeli KUPAC.

Primer 2.

```
SQL> select distinct ime from odeljenje;
```

Izlaz:

```
IME
-----
ACCOUNTING
OPERATIONS
RESEARCH
SALES
```

2. SQL izrazi

WHERE klauzula

Sintaksa WHERE klauzule je sledeća:

WHERE <Uslov pretraživanja>

SELECT, FROM i WHERE su tri najčešće korišćene klauzule u SQL-u. WHERE omogućuje selektivnost u našem upitau. Bez upotrebe WHERE klauzule biće prikazani svi podaci iz tabele. Na primer:

```
SQL> select * from odeljenje;
```

Izlaz:

ODELJENJE_ID	IME	LOKACIJA_ID
10	ACCOUNTING	122
20	RESEARCH	124
30	SALES	123
40	OPERATIONS	167
12	RESEARCH	122
13	SALES	122
14	OPERATIONS	122
23	SALES	124
24	OPERATIONS	124
34	OPERATIONS	123
43	SALES	167

11 rows selected.

Kod ovog upita prikazani su svi redovi tabele ODELJENJE. Ako bi smo želeli da vidimo samo podatke za Odeljenje prodaje (SALES), naš upit se mora proširiti uslovom da je ime odeljenja (IME) jednako SALES tj. IME = 'SALES'

```
SQL> select * from odeljenje where ime='SALES';
```

Izlaz:

ODELJENJE_ID	IME	LOKACIJA_ID
30	SALES	123
13	SALES	122
23	SALES	124
43	SALES	167

Reč IME može se pisati i malim slovima, međutim 'SALES' se piše onako kako je zapisan u bazi podataka (u našem slučaju velikim slovima). Kako upiti postaju složeniji preglednije ih je pisati u više redova. Na primer, predhodni upit može biti zapisan i kao:

```
SQL> select * from odeljenje  
where ime='SALES';
```

ili

```
SQL> select *  
from odeljenje  
where ime='SALES';
```