

SQL

- SQL – Structured Query Language
- Deklarativni jezik koji objedinjuje:
 - upitni jezik
 - jezik za manipulaciju podacima (Data Manipulation Language)
 - jezik za definiciju podataka (Data Definition Language)

- SQL obezbeđuje realizaciju sledećih zadataka:
 - izražavanje upita (upitni jezik, naredba SELECT)
 - ažuriranje baze podataka (jezik za manipulaciju podacima, naredbe INSERT, UPDATE, DELETE)
 - realizacija implementacione šeme baze podataka i definisanje fizičke organizacije baze podataka (jezik za definiciju podataka, naredbe CREATE, DROP, ALTER)
 - realizacija pogleda (naredbe CREATE VIEW, DROP VIEW)
 - transakcijska obrada podataka (COMMIT, ROLLBACK, SAVEPOINT)
 - automatsko održavanje rečnika podataka, zaključavanje resursa (LOCK TABLE), zaštita podataka od neovlaštenog pristupa (GRANT, REVOKE), obezbeđenje proceduralnog načina obrade 'slog po slog'

Izražavanje upita i osnovna struktura naredbe SELECT

- Sve vrste upita u SQL-u izražavaju se putem naredbe SELECT
- Osnovna struktura:

```
SELECT <lista_obeležja>  
FROM <lista_tabela>  
WHERE <uslov_selekcije>
```
- <lista_obeležja> - lista obeležja nad kojima se formira upit
- <lista_tabela> - lista tabela potrebnih za izvršavanje postavljenog upita
- <uslov_selekcije> - uslov po kome se selektuju torke iz tabele

Primer

- Izlistati sadržaj svih tabela.

```
select * from radnik;
```

```
select * from projekat;
```

```
select * from radina;
```

Primer

- Izlistati imena i prezimena radnika.

```
select ime, prezime from radnik;
```

- Izlistati različita imena radnika.

```
select distinct ime from radnik;
```

<uslov_selekcije>

- Predstavlja logički izraz koji može biti sastavljen iz više predikatskih izraza povezanih logičkim operatorima AND, OR, NOT
- Predikatski izraz može biti:
 - relacioni izraz tipa: $x > y$, $x < y$, $x \leq y$, $x \geq y$, $x \neq y$, $x = y$
 - x IS NULL, x IS NOT NULL
 - x BETWEEN y AND z , x NOT BETWEEN y AND z
 - x LIKE y , x NOT LIKE y
 - x IN (<lista_vrednosti>), x NOT IN (<lista_vrednosti>)
 - $x \Theta$ ANY (<lista_vrednosti>), $\Theta \{<, >, \leq, \geq, \neq, =\}$
 - $x \Theta$ ALL (<lista_vrednosti>)
 - EXISTS (<lista_vrednosti>)
 - NOT EXISTS (<lista_vrednosti>)

<uslov_selekcije>

- <obelezje> LIKE <uzorak>, džoker znaci: %, _
- aritmetički izrazi se mogu pojaviti kao operandi u <lista_obelezja> i <uslov_selekcije>
- ROUND (x, y) - zaokruživanje vrednosti x na naznačeni broj decimala y

Primeri

- Izlistati mbr, ime i prz radnika koji imaju platu veću od 25000.

```
select mbr, ime, prezime from radnik  
where plata>50000;
```

- Izlistati mbr, ime, prz radnika koji nemaju šefa.

```
select mbr, ime, prezime from radnik  
where sef is null;
```

- Izlistati mbr, ime, prz radnika čija je plata između 20000 i 24000 dinara.

```
select mbr, ime, prezime from radnik  
where plata between 50000 and 54000;
```

- Izlistati mbr, ime, prz radnika čije prezime počinje na slovo M.

```
select mbr, ime, prezime from radnik  
where prezime like 'M*';
```

Primeri

- Izlistati godišnju platu svakog radnika.

```
select mbr, ime, prezime, plata*12 from radnik;
```

- Izlistati matične brojeve radnika koji rade na projektima sa šifrom 10, 20 ili 30.

```
select distinct mbr from radina  
where spr in (10,20, 30);
```

- Izlistati matične brojeve radnika koji rade na projektu sa šifrom 10 ili rade 2, 4, ili 6 sati.

```
select distinct mbr from radina  
where brc in (2, 4, 6) or spr=10;
```

- Izlistati matične brojeve radnika koji se zovu Ana ili Sanja.

```
select mbr, ime, prezime from radnik  
where ime in ('Ana', 'Sanja');
```

- Izlistati ime, prz, god radnika rođenih između 1973 i 1975.

```
select ime, prezime, datumr from radnik  
where datumr between #1/1/1998# and #1/1/2000#;
```

Order by

- Izlazni rezultat upita se može sortirati u rastućem ili opadajućem redosledu pomoću klauzule

```
ORDER BY <podlista obeležja>
```

- Redosled obeležja u <podlista_obeležja> određuje prioritet po kome se vrši uređivanje
- Iza svakog obeležja u <podlista_obeležja> može se navesti službena reč ASC ili DESC, kojom se zahteva rastući, odnosno opadajući prikaz podataka (default=ASC)

Primer

- Prikazati radnike koji imaju šefa sortirano po imenu rastuće, a po prezimenu opadajuće.

```
select mbr, ime, prezime, plata  
from radnik  
where sef is not null  
order by ime, prezime desc;
```

Skupovne funkcije

- SQL je skupovno orijentisan jezik, pa u njemu postoje funkcije čiji argument predstavlja skup podataka, a rezultat je vrednost izračunata na osnovu zadatog skupa podataka
 - MAX (<obeležje>) – vraća maksimalnu vrednost za obeležje, uzimajući u obzir sve selektovane torke
 - MIN (<obeležje>)
 - COUNT (*) – vraća ukupan broj selektovanih torki
 - COUNT (<obeležje>) – vraća ukupan broj selektovanih torki, za koje vrednost obeležja nije nula vrednost
 - COUNT (DISTINCT <obeležje>)
 - SUM (<obeležje>) – vraća zbir vrednosti datog obeležja za sve selektovane torke
 - SUM (DISTINCT <obeležje>)
 - AVG (<obeležje>) – vraća srednju vrednosti datog obeležja za sve selektovane torke
 - AVG (DISTINCT <obeležje>)

Primeri

- Koliko ima radnika?

```
select count(*) as brojRadnika from radnik;
```

- Koliko ima šefova?

```
SELECT count(sef)
```

```
FROM
```

```
(SELECT DISTINCT sef FROM radnik)
```

```
select count(distinct sef) broj_sefova  
from radnik;
```

- Prikazati minimalnu i maksimalnu platu radnika.

```
select min(plata) as minimalna, max(plata) as maksimalna  
from radnik;
```

- Prikazati prosečnu platu svih radnika zaokruženo na jednu decimalu.

```
select round(avg(plata), 1)  
from radnik;
```

- Prikazati broj radnika, prosečnu platu i ukupnu godišnju platu radnika.

```
select count(*), avg(plata), 12*sum(plata)  
from radnik;
```

Spajanje tabela

- Kada je potrebno da se za realizaciju upita koriste obeležja iz više tabela, te tabele se moraju međusobno povezati
- Tabele se povezuju preko zajedničkih obeležja (strani ključevi, referencijalni integritet)
- Isti nazivi obeležja se moraju kvalifikovati nazivom tabele

`tabela.obelezje`

Spajanje tabela

- Prikazati mbr, prz, ime, plt i brc angažovanja svih radnika koji rade na projektu sa šifrom 1.

```
select radnik.mbr, radnik.prezime, radnik.ime,  
       radnik.plata, radina.brc  
from radnik, radina  
where radina.spr=1 and radnik.mbr=radina.mbr;
```

```
select r.mbr, r.prezime, r.ime, r.plata, rp.brc  
from radnik as r, radina as rp  
where rp.spr=1 and r.mbr=rp.mbr;
```

- Prikazati mbr, ime, prz i plt radnika koji su rukovodioci projekata.

```
select distinct mbr, ime, prezime, plata  
from radnik, projekat  
where ruk=mbr;
```


Spajanje tabela

- Prikazati nazive projekata na kojima radi Milan Milić.

```
select p.naziv
```

```
from projekat as p, radnik as r,  
     radina as rp
```

```
where p.spr=rp.spr and rp.mbr=r.mbr and  
      r.ime='Milan' and r.prezime='Savic';
```

- Prikazati imena i prezimena radnika koji rade na projektu sa nazivom P3.

```
select ime, prezime
```

```
from projekat as p, radnik as r,  
     radproj as rp
```

```
where p.spr=rp.spr and rp.mbr=r.mbr and  
      p.naziv='P3';
```

Višestruka upotreba iste tabele

- Prikazati mbr, ime, prz, plt radnika koji zarađuju više od radnika sa matičnim brojem 401.

```
select r.mbr, r.prezime, r.ime, r.plata
from radnik as r, radnik as r1
where r.plata>r1.plata and r1.mbr=3;
```

- Prikazati imena, prezimena i plate radnika koji zarađuju bar 1000 dinara manje od rukovodioca projekta na kom radnik radi.

```
select r1.ime, r1.prezime, r1.plata, p.naziv
from radnik as r1, radnik as r2, projekat as p,
radproj as rp
where r1.mbr=rp.mbr and rp.spr=p.spr and
p.ruk=r2.mbr and r1.plata+1000<r2.plata;
```