

OSNOVE BAZA PODATAKA

Standardni upitni jezik - SQL

Ulaganje upita


- ❑ SQL dozvoljava ulaganje upita nad jednom relacijom u upit nad drugom relacijom

Ulaganje upita

□ Primer:

- Upit koji prikazuje sve nastavnike koji predaju neki bilo koji predmet

```
select *  
from nastavnik  
where s_nas in (select s_nas from predaje);
```



SPOLJAŠNJI UPIT


UNUTRAŠNJI UPIT

Ulaganje upita

- Prilikom izvršavanja upita sa uloženim upitom, prvo se izvršava unutrašnji upit a zatim spoljašnji upit

```
select *  
from nastavnik  
where s_nas in (select s_nas from predaje);
```


SPOLJAŠNJI UPIT


UNUTRAŠNJI UPIT

Ulaganje upita

- ❑ Uloženi upit se može pojaviti u FROM klauzuli
- ❑ Primer:
 - Upit koji prikazuje imena i prezimena nastavnika

```
select nast.*  
from (select prezime_ime from nastavnik) nast;
```

- U ovom primeru je bilo neophodno imenovati unutrašnji upit (*nast*) kako bi bilo moguće vršiti selekciju iz rezultata tog upita

Ulaganje upita

- ❑ Uloženi upit se može pojaviti u FROM klauzuli
- ❑ Primer:
 - Upit koji prikazuje imena, prezimena i godišnje plate nastavnika

```
select nast.*  
from (select prezime_ime, plata * 12 as godPlata  
      from nastavnik) nast;
```

- Ukoliko unutrašnji upit sadrži izraze, tim izrazima moraju biti dodeljeni alijasi

Ulaganje upita

- ❑ Uloženi upit se može pojaviti u SELECT klauzuli
- ❑ Primer:
 - Upit koji za svakog nastavnika prikazuje njegovo ime, prezime, kao i udeo njegove plate u ukupnoj plati svih nastavnika

```
select prezime_ime, plata/(select sum(plata) from nastavnik)  
from nastavnik;
```

- U slučaju ulaganja upita u SELECT klauzuli, treba paziti da rezultat unutrašnjeg upita bude *samo jedna vrednost*

Ulaganje upita

- ❑ Uloženi upit se može pojaviti u WHERE klauzuli
 - Najčešći slučaj upotrebe
- ❑ Primer:
 - Upit koji prilazuje sve predmete koje predaje bar jedan nastavnik

```
select *  
from predmet  
where s_pred in (select s_pred from predaje);
```


Ulaganje upita

- ❑ Uloženi upit se može pojaviti u WHERE klauzuli
 - Najčešće se koristi u kombinaciji sa operatorima =, !=, IN, NOT IN, EXISTS, NOT EXISTS, ANY, ALL...
 - U slučaju upotebe operatora IN, ANY i ALL unutrašnji upit treba da vraća *listu*
 - ❑ *Lista* je tabela sa **jednom** kolonom

```
select *  
from predmet  
where s_pred in (select s_pred, s_nas from predaje);
```



NEISPRAVNO!

Zadatak

- ❑ Prikazati imena i prezimena svih nastavnika koji su direktori.

```
select prezime_ime  
from nastavnik  
where s_nas in (select s_dir from nastavnik);
```

Zadatak

- ❑ Prikazati šifre, imena, prezimena i zvanje svih nastavnika koji su imaju isto zvanje kao nastavnik Radović Nikola.

```
select s_nas, prezime_ime, zvanje
from nastavnik
where zvanje in (select zvanje
                  from nastavnik
                  where prezime_ime like 'Radovic Nikola');
```

Zadatak

- ❑ Prikazati prezime, ime, zvanje i platu docenata koji ne predaju ni jedan predmet.

```
select prezime_ime, zvanje, plata
from nastavnik
where zvanje = 'DOCENT'
      and
      s_nas not in (select s_nas from predaje);
```

Zadatak

- Pronađi nastavnika sa najvećom platom.

```
select s_nas, prezime_ime, zvanje, plata  
from nastavnik  
where plata = (select max(plata) from nastavnik);
```

- Primetiti upotrebu operatora jednakosti (=) u WHERE klauzuli
 - U slučaju kada očekujemo da unutrašnji upit vraća samo jednu vrednost, treba koristiti operator = umesto operatora IN

Ulaganje upita

- Dinamička zamena
 - Korišćenje tabele iz spoljašnjeg upita u unutrašnjem upitu („povezani“ upit)

Ulaganje upita

□ Dinamička zamena

■ Primer:

- Upit koji prikazuje prezime i ime svakog nastavnika, kao i broj predmeta koje taj nastavnik predaje

```
select prezime_ime,  
       (select count(s_pred)  
        from predaje p  
        where p.s_nas = n.s_nas)  
from nastavnik n;
```

Ulaganje upita

☐ Dinamička zamena

- Dinamička zamena najčešće zahteva definisanje alijasa za tabele koje se koriste u upitu
 - ☐ U prethodnom primeru, za tabelu NASTAVNIK je uveden alijas N, dok je za tabelu PREDAJE uveden alijas P
 - ☐ Ovo nam je omogućilo da razlikujemo obeležje S_NAS iz tabele NASTAVNIK od obeležja S_NAS iz tabele PREDAJE

Zadatak

- ❑ Prikazati šifre, prezimena, imena, plate i zvanja nastavnika čija je plata veća od prosečne plate nastavnika tog istog zvanja.

```
select s_nas, prezime_ime, plata, zvanje
from nastavnik n1
where plata > (select avg(plata)
               from nastavnik n2
               where n2.zvanje = n1.zvanje);
```

Ulaganje upita

☐ EXISTS operator

- Dinamička zamena se često se koristi u kombinaciji sa EXISTS operatorom
 - ☐ EXISTS se koristi u uslovu selekcije u WHERE klauzuli
- Uslov sa operatorom EXISTS je zadovoljen ako unutrašnji upit vraća neprazan skup n-torki
 - ☐ U suprotnom, EXISTS uslov nije zadovoljen

Ulaganje upita

❑ EXISTS operator

■ Primer:

- ❑ Upit koji prikazuje sve predmete koji se predaju

```
select naziv
from predmet pd
where exists (select *
              from predaje pj
              where pj.s_nas = pd.s_nas);
```

Ulaganje upita

□ EXISTS operator

- Za EXISTS operator nije bitno koja se obeležja selektuju u unutrašnjem upitu
- Bitno je samo da li rezultat izvršenja unutrašnjeg upita sadrži neke torke ili ne
- Postoji i NOT EXISTS varijanta

Zadatak

- Pronađi nastavnika sa najvećom platom (koristiti NOT EXISTS operator).

```
select n1.*  
from nastavnik n1  
where not exists (select *  
                  from nastavnik n2  
                  where n2.plata > n1.plata);
```

- Kako interpretirati gore-prikazani upit:
 - „Prikazati onog nastavnika koji je takav da **ne postoji** ni jedan drugi nastavnik sa platom većom od njegove.“

Ulaganje upita

☐ ANY operator

- Koristi se u WHERE klauzuli

- Primer:

```
select *  
from nastavnik  
where s_nas = any(1, 2, 3);
```

- ANY uslov je zadovoljen ako je obeležje jednako bar jednoj iz liste vrednosti

- ☐ Ekvivalentno IN operatoru

Ulaganje upita

☐ ALL operator

- Koristi se u WHERE klauzuli

- Primer:

```
select *  
from nastavnik  
where s_nas != all(1, 2, 3);
```

- ALL uslov je zadovoljen ako je obeležje različito od svih vrednosti u listi

- ☐ Ekvivalentno NOT IN operatoru

Ulaganje upita

☐ ANY i ALL operatori

- Razlika između ANY/ALL i IN/NOT IN jeste u tome što relacioni operator kod ANY/ALL ne mora uvek biti $=/!=$
 - ☐ Moguće je koristiti i $>$, $<$, $<=$, $>=$...

Zadatak

- Pronađi nastavnika sa najvećom platom (koristiti ALL operator).

```
select *  
from nastavnik  
where plata >= all(select plata  
                    from nastavnik);
```

Zadatak

- Prikazati prezimena i imena nastavnika koji predaju predmet sa šifrom 5, a ne predaju predmet sa šifrom 3.

```
select *  
from nastavnik  
where s_nas in (select s_nas  
                from predaje  
                where s_pred = 5)  
and  
s_nas not in (select s_nas  
              from predaje  
              where s_pred = 3);
```

- Zašto ovo nije moguće izvesti pomoću samo jednog unutrašnjeg upita?

Zadatak

- ❑ Prikazati šifru, prezime, ime i platu nastavnika koji predaju predmet „Programiranje RS”.

```
select s_nas, prezime_ime, plata
from nastavnik
where s_nas in
    (select s_nas
     from predaje
     where s_pred in
         (select s_pred
          from predmet
          where s_pred = 'Programiranje RS'));
```

Zadatak

- ❑ Prikazati šifru, prezime, ime i platu redovnih profesora čija je plata veća od prosečne plate docenata.

```
select s_nas, prezime_ime, plata
from nastavnik
where zvanje = 'R Prof' and plata >
    (select avg(plata)
     from nastavnik
     where zvanje = 'Docent');
```

Zadatak

- ❑ Prikazati različita zvanja nastavnika, i prosečnu platu po svakom zvanju.

```
select zvanje, avg(plata)
from nastavnik
group by zvanje;
```

Zadatak

- Doraditi prethodni zadatak tako da se prikaže samo ono zvanje za koje je prosečna plata svih nastavnika tog zvanja *najveća*.
 - Najlakši način da se ovo izvede jeste da se koristi ALL operator.

```
select zvanje
from nastavnik
group by zvanje
having avg(plata) >= all(select avg(plata)
                        from nastavnik
                        group by zvanje);
```

Zadatak

- ❑ Prikazati sva mesta u kojima se izvodi 2 ili više predmeta iz osmog semestra. Prikazati i ukupan broj predmeta koji se izvode u time mestima.

```
select mesto, count(s_pred)
from predmet
where semester = 8
group by mesto
having count(s_pred) >= 2;
```

Zadatak

- ❑ Prikazati podatke o nastavnicima koji *ne predaju ni jedan* predmet koji u svom nazivu sadrži reč „program“.

```
select s_nas, prezime_ime, plata
from nastavnik
where s_nas not in
    (select s_nas
     from predaje
     where s_pred in
        (select s_pred
         from predmet
         where s_pred like '%program%'));
```

Ovaj deo upita selektuje šifre svih nastavnika koji **predaju** predmete koji sadrže reč „program“. Spoljašnji upit selektuje sve nastavnike **sem tih**.

Zadatak

- Prikazati šifru i naziv onih predmeta za koje je ukupan broj časova veći od prosečnog broja časova svih predmeta.

```
select s_pred, naziv
from predmet
where s_pred in
    (select s_pred
     from predaje
     group by s_pred
     having sum(casova) > (select avg(casova)
                          from predaje));
```

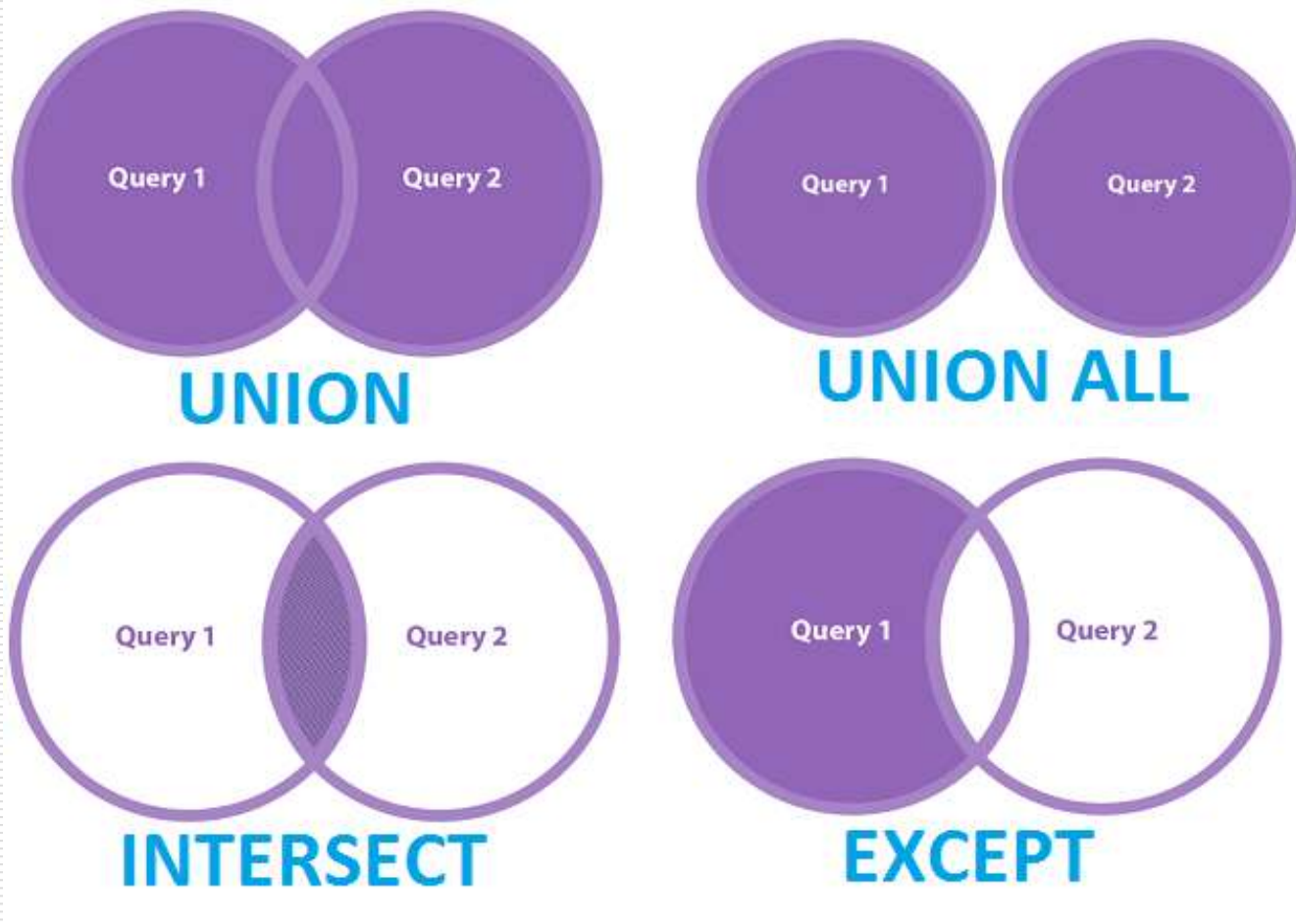
Skupovni operatori

- ❑ SQL dozvoljava upotrebu skupovnih operacija nad rezultatima upita.
 - Podržani su operatori unije, preseka i razlike
 - ❑ UNION, INTERSECT i EXCEPT

```
<upit 1>  
{ UNION | INTERSECT | EXCEPT }  
<upit 2>
```

- ❑ Podržan je i operator UNION ALL
 - Unija skupova bez uklanjanja duplikata n-torki

Skupovni operatori



Skupovni operatori

- Da bi navedene operacije mogle biti primenjene, rezultati upita SELECT blokova koji učestvuju u operaciji moraju:
 - Imati isti broj rezultujućih kolona i
 - Kolone moraju biti istih tipova

Zadatak

- Prikazati šifru, prezime, ime i platu svakog nastavnika. Plate direktora prikazati uvećane za 20%.

```
select s_nas, prezime_ime, plata * 1.20
from nastavnik
where s_nas in (select s_dir from nastavnik)
union
select s_nas, prezime_ime, plata
from nastavnik
where semester not in (select s_dir from nastavnik)
```

} direktori

} ostali...

Domaći

□ Kreirati tabele:

STUDENT (S_STUD, IME, PRZ, BR_INDEKSA)

POLAZE (S_STUD, S_PRED, BR_BODOVA, OCENA)

Obeležje	Značenje	Dozvoljene nedostajuće vrednosti
S_STUD	Šifra studenta (koristi se kao identifikator studenta)	NE
IME	Ime studenta	NE
PRZ	Prezime studenta	NE
BR_INDEKSA	Broj indeksa studenta	NE
S_PRED	Šifra predmeta koji je student polagao	NE
BR_BODOVA	Broj bodova koji je student osvojio na predmetu	DA
OCENA	Ocena studenta iz predmeta	DA

Domaći

□ Kreirati tabele:

STUDENT (S_STUD, IME, PRZ, BR_INDEKSA)

POLAZE (S_STUD, S_PRED, BR_BODOVA, OCENA)

```
create table student (  
    s_stud          smallint primary key,  
    ime            varchar(15) not null,  
    prz            varchar(20) not null,  
    br_indeksa     varchar(10) not null  
);  
  
create table polaze (  
    s_stud          smallint foreign key references student (s_stud),  
    s_pred          smallint foreign key references predmet (s_pred),  
    br_bodova       smallint,  
    ocena           smallint,  
    primary key (s_stud, s_pred)  
);
```

Domaći

- Popuniti tabele STUDENT i POLAZE podacima:

```
insert into student values (1000, 'Sava', 'Savic', 'SR1/2021');
insert into student values (2000, 'Mirko', 'Miric', 'SR2/2021');
insert into student values (3000, 'Marina', 'Maric', 'SR3/2021');
insert into student values (4000, 'Javor', 'Jelic', 'SR4/2021');
insert into student values (5000, 'Goran', 'Goric', 'SR5/2021');
insert into student values (6000, 'Jovan', 'Ilic', 'SR6/2021');
insert into student values (100, 'Marko', 'Markovic', 'SR7/2019');
```

```
insert into polaze values (1000, 2, 87, 9);
insert into polaze values (4000, 2, 40, 5);
insert into polaze values (6000, 2, 57, 6);
insert into polaze values (100, 2, 71, 8);
insert into polaze values (1000, 3, null, null);
insert into polaze values (4000, 3, 17, 5);
insert into polaze values (5000, 3, 77, 8);
insert into polaze values (1000, 1, 80, 8);
insert into polaze values (3000, 1, 97, 10);
```


Domaći

- ❑ Prikazati studente koji nisu polagali ni jedan predmet.
- ❑ Prikazati studente koji su polagali neki predmet kod nastavnika Ilić Jovana.
- ❑ Prikazati sve predmete koje je neko uspeo da *položi*.
 - Student je položio predmet ako ima bar ocenu 6.
- ❑ Prikazati sve predmete koje je Sava Savić uspeo da *položi*.

Domaći

□ Dodatni zadatak:

- Prikaži studenta koji je ostvario najveću ocenu na predmetu "Strukture i BP".

■ Rešenje:

```
select *  
from student  
where s_stud in  
  (select s_stud  
   from polaze  
   where s_pred = (select s_pred from predmet where naziv = 'Strukture i BP')  
                  and ocena >= (select max(ocena)  
                                from polaze  
                                where s_pred =  
                                (select s_pred from predmet where naziv = 'Strukture i BP')));
```