

OSNOVE BAZA PODATAKA

Standardni upitni jezik - SQL

Ponavljanje

□ GROUP BY

■ Primer:

- Za svako zvanje, ispisati prosečnu platu nastavnika tog zvanja.

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

Ponavljjanje

□ Ulaganje upita

■ Primer:

- Za svako zvanje, ispisati prosečnu platu nastavnika tog zvanja.

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

Ponavljjanje

□ Skupovne operacije nad torkama

- UNION, INTERSECT, EXCEPT

- Primer:

- Prikazati prezime, ime, zvanje i plate svih docenata i redovnih profesora. Plate redovnih profesora prikazati uvećane za dodatak.

```
select prezime_ime, zvanje, plata + isnull(dodatak, 0)
from nastavnik
where zvanje = 'R PROF'
union
select prezime_ime, zvanje, plata
from nastavnik
where zvanje = 'DOCENT';
```

Povezivanje više relacija

- U svim prethodnim primerima SELECT naredbe rezultat pretraživanja formiran je iz samo jedne relacije.
 - Kako doći do rezultata pretraživanja koji uključuje podatke iz 2 ili više relacija?
- Povezivanje 2 ili više relacija omogućava kombinovanje podataka iz tih relacija, sa prikazom rezultata u jednoj relaciji
 - JOIN operacija

Povezivanje više relacija

□ Primer:

- Upit koji prikazuje prezime i ime svakog nastavnika, platu, kao i šifru predmeta koje on predaje.

```
select prezime_ime, plata, s_pred  
from nastavnik, predaje  
where nastavnik.s_nas = predaje.s_nas;
```

uslov spajanja 2 relacije

Povezivanje više relacija

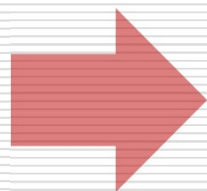
1 `select prezime_ime, plata, s_pred
from nastavnik, predaje
where nastavnik.s_nas = predaje.s_nas;`

relacija NASTAVNIK

S_NAS	PREZIME_IME	PLATA
1	Radovic Nikola	14500
2	Petrovic Petar	11500
3	Peic Petar	11500

relacija PREDAJE

S_NAS	S_PRED	CASOVA
1	3	3
1	5	2
2	1	2



relacija NASTAVNIK

S_NAS	PREZIME_IME	PLATA
1	Radovic Nikola	14500
1	Radovic Nikola	14500
1	Radovic Nikola	14500
2	Petrovic Petar	11500
2	Petrovic Petar	11500
2	Petrovic Petar	11500
3	Peic Petar	11500
3	Peic Petar	11500
3	Peic Petar	11500

relacija PREDAJE

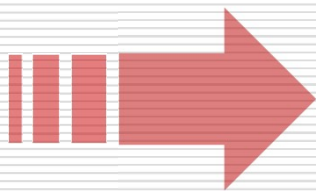
S_NAS	S_PRED	CASOVA
1	3	3
1	5	2
2	1	2
1	3	3
1	5	2
2	1	2
1	3	3
1	5	2
2	1	2

Dekartov proizvod n-torki dve relacije
(svaka n-torka iz prve relacije spojena sa svakom n-torkom iz druge relacije)

Povezivanje više relacija

```
select prezime_ime, plata, s_pred  
from nastavnik, predaje
```

2 where nastavnik.s_nas = predaje.s_nas;



relacija NASTAVNIK

S_NAS	PREZIME_IME	PLATA
1	Radovic Nikola	14500
1	Radovic Nikola	14500
1	Radovic Nikola	14500
2	Petrovic Petar	11500
2	Petrovic Petar	11500
2	Petrovic Petar	11500
3	Peic Petar	11500
3	Peic Petar	11500
3	Peic Petar	11500

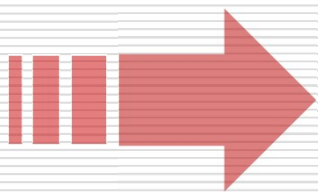
relacija PREDAJE

S_NAS	S_PRED	CASOVA
1	3	3
1	5	2
2	1	2
1	3	3
1	5	2
2	1	2
1	3	3
1	5	2
2	1	2

Povezivanje više relacija

```
select prezime_ime, plata, s_pred  
from nastavnik, predaje
```

2 where nastavnik.s_nas = predaje.s_nas;



relacija NASTAVNIK

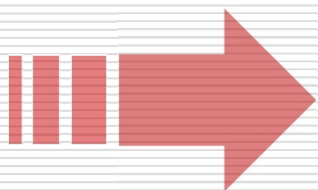
S_NAS	PREZIME_IME	PLATA
1	Radovic Nikola	14500
1	Radovic Nikola	14500
2	Petrovic Petar	11500

relacija PREDAJE

S_NAS	S_PRED	CASOVA
1	3	3
1	5	2
2	1	2

Povezivanje više relacija

3 `select prezime_ime, plata, s_pred
from nastavnik, predaje
where nastavnik.s_nas = predaje.s_nas;`



relacija NASTAVNIK relacija PREDAJE

PREZIME_IME	PLATA	S_PRED
Radovic Nikola	14500	3
Radovic Nikola	14500	5
Petrovic Petar	11500	1

Povezivanje više relacija

- Ukoliko se izostavi uslov spajanja u WHERE klauzuli, rezultat će biti Dekartov proizvod n-torki iz relacija
 - Svaka torka iz jedne relacije povezana sa svakom torkom iz druge
 - U najvećem broju slučajeva, to *nije* željeno ponašanje
- Za svaku novu relaciju koja se pojavljuje u FROM klauzuli, potreban je dodatni uslov spajanja u WHERE klauzuli
 - U slučaju spajanja 3 relacije, potrebna su 2 uslova spajanja itd


Zadatak

- ❑ Prikazati šifre svih nastavnika koji drže nastavu u osmom semestru.

```
select s_nas
from predmet, predaje
where predmet.s_pred = predaje.s_pred
      and semestar = 8;
```

```
select s_nas
from predmet pd, predaje pj
where pd.s_pred = pj.s_pred
      and semestar = 8;
```

Sa ciljem da skratimo
pisanje, uvodimo
alijase za tabele u
FROM klauzuli



Zadatak

- ❑ Prikazati imena i prezimena svih nastavnika koji predaju predmet "Programiranje RS". Prikazati i broj časova koji oni drže na ovom predmetu.

```
select prezime_ime, casova
from predmet pd, predaje pj, nastavnik n
where pd.s_pred = pj.s_pred
      and pj.s_nas = n.s_nas
      and naziv = 'Programiranje RS';
```

Povezivanje više relacija

- Moguće je spajati istu relaciju više puta u FROM klauzuli
 - SELF JOIN operacija
 - U ovom slučaju je neophodno definisati alijase za svaku instancu relacije u FROM klauzuli

Zadatak

- Za svakog nastavnika prikazati njegovo ime i prezime, kao i ime i prezime njegovog direktora.

```
select n.prezime_ime as "Nastavnik",  
       d.prezime_ime as "Direktor"  
from nastavnik n, nastavnik d  
where n.s_dir = d.s_nas;
```

```
select n.prezime_ime as "Nastavnik",  
       d.prezime_ime as "Direktor"  
from nastavnik n inner join nastavnik d on  
                                n.s_dir = d.s_nas;
```

Spajanje koje smo do sada koristili naziva se *unutrašnje spajanje* (INNER JOIN). Postoji i alternativna sintaksa za zapisivanje ovog tipa spoja (primetiti da je uslov spajanja premešten u FROM klauzulu).

Povezivanje više relacija

- U prethodnom zadatku se tražilo da prikazemo **sve** nastavnike i njihove direktore
 - Da li smo to uspešno izveli?
 - Primetiti da u rezultatu nedostaju neki nastavnici
 - Nastavnici Radović Nikola, Savić Milan i Petrić Janko nemaju direktora
 - Kod ovih nastavnika obeležje `s_dir` ima vrednost NULL, pa za njih uslov spajanja `n.s_dir = d.s_nas` nikada nije zadovoljen
 - Ovi nastavnici neće biti prikazani u rezultatu upita
 - Kada je potrebno raditi spoj i prikazati **sve** n-torke relacije, može se koristiti *spoljno spajanje* (OUTER JOIN)

Povezivanje više relacija

□ Spoljno spajanje – OUTER JOIN

■ Postoje 3 tipa:

- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN

■ Na primer, u slučaju upotrebe LEFT OUTER JOIN operacije:

- U rezultatu će biti sadržane sve n-torke iz *leve* relacije
- Torke iz leve relacije za koje postoji odgovarajuća n-torka u desnoj relaciji, biće spojene sa tim odgovarajućim n-torkama iz desne relacije
- Torke iz leve relacije za koje *ne postoji* odgovarajuća n-torka u desnoj relaciji biće zadržane u rezultatu, ali će za vrednosti obeležja iz desne relacije biti prikazane NULL vrednosti
 - Svaka torke iz leve relacije za koju ne postoji odgovarajuća torke u desnoj relaciji, biće prikazana u rezultatu **samo jedanput**

Zadatak

- Za **svakog** nastavnika prikazati njegovo ime i prezime, kao i ime i prezime njegovog direktora.

```
select n.prezime_ime as "Nastavnik",  
       d.prezime_ime as "Direktor"  
from nastavnik n left outer join nastavnik d on  
                                n.s_dir = d.s_nas;
```

Zadatak

- ❑ Prikazati sve predmete, kao i imena i prezimena nastavnika koji predaju te predmete. Za predmete koje niko ne predaje ispisati „nema nastavnika“ umesto imena nastavnika.

```
select s_pred, isnull(prezime_ime, 'nema nastavnika')
from
    (predmet pd left outer join predaje pj on
        pd.s_pred = pj.s_pred)
    left outer join nastavnik n on
        pj.s_nas = n.s_nas;
```

Zadatak

- Za svakog nastavnika prikazati njegovu šifru, ime, prezime i broj predmeta koje on predaje. Prikazati i nastavnike koji ne predaju ni jedan predmet.

```
select s_nas, prezime_ime, count(s_pred)
from nastavnik n left outer join predaje p on
                                     n.s_nas = p.s_nas
group by s_nas, prezime_ime;
```

Za potrebe ispisivanja imena i prezimena nastavnika, neophodno je dodati ovo obeležje u GROUP BY klauzulu. Primetiti da to ne remeti kreiranje grupa po obeležju `s_nas` (jer je za jednu šifru nastavnika uvek isto ime i prezime)

Zadatak

- ❑ Prikazati šifru nastavnika, prezime, ime, zvanje, i broj predmeta koje nastavnik predaje. Prikazati samo samo one nastavnike koji predaju bar jedan predmet, i imaju platu veću od prosečne plate svog zvanja.

```
select n.s_nas, prezime_ime, zvanje, count(s_pred)
from nastavnik n, predaje p
where n.s_nas = p.s_nas and
      plata > (select avg(plata)
               from nastavnik nn
               where nn.zvanje = n.zvanje)
group by n.s_nas, prezime_ime
having count(s_pred) >= 1;
```

Zadatak

- Za svaki predmet prikazati šifru, naziv, mesto u kome se održava, broj nastavnika koji drže nastavu iz tog predmeta i ukupan broj časova tog predmeta. Za predmete koji se ne drže, prikazati 0 za broj nastavnika i ukupan broj časova.

ISNULL funkcija nije neophodna kod prikazivanja broja nastavnika. Zašto?

```
select pd.s_pred, naziv, mesto,  
       count(s_nas) as "Broj nastavnika",  
       isnull(sum(casova), 0) as "Uk. broj časova"  
from predmet pd left outer join predaje pj on  
                                     pd.s_pred = pj.s_pred  
group by pd.s_pred, naziv, mesto;
```

Zadatak

- ❑ Prikazati šifre, imena i prezimena svih direktora, kao i broj nastavnika koji su im podređeni.

```
select d.s_dir, dd.prezime_ime,  
       count(d.s_nas) as "Broj podređenih nastavnika"  
from nastavnik d, nastavnik dd  
where d.s_dir = dd.s_nas  
group by d.s_dir, dd.prezime_ime;
```

Iz tabele nastavnik moguće
je iščitati šifre direktora
(s_dir) i šifre podređenih
(s_nas) nastavnika

Tabela NASTAVNIK je još
jedanput spojena kako bismo
prikazali ime i prezime
direktora

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 su polagali predmete, sa ocenama koje imaju iz tih predmeta.
- ❑ Prikazati studente koji su polagali predmet "Strukture i BP"
- ❑ Za **svaki** predmet prikazati njegov naziv i prosečnu ocenu na tom predmetu.
- ❑ Za svakog studenta koji je polagao neki predmet, prikazati njegovu najbolju ocenu.
 - Na primer:

s_stud	ime	prz	Najbolja ocena
100	Marko	Markovic	8
1000	Sava	Savic	9
...			

Domaći

□ Dodatni zadatak:

- Za svakog studenta koji je polagao neki predmet, prikazati njegovu najbolju ocenu.
 - Prikazati i naziv predmeta iz koga je ta ocena.

■ Rešenje:

```
select s.s_stud, ime, prz, ocena, naziv
from student s, polaze p, predmet pd,
     (select s_stud, max(ocena) maxOcena from polaze group by s_stud) x
where s.s_stud = p.s_stud and p.s_pred = pd.s_pred and
      x.maxOcena = p.ocena and x.s_stud = p.s_stud;
```

Spoj 2 tabele preko 2 uslova spajanja ☺
(u nekim slučajevima je i to moguće)