

Fakultet tehničkih nauka, DRA, Novi Sad

Predmet:

# Struktura i organizacija baza podataka

Dr Slavica Aleksić,

Vladimir Dimitrieski,

Nikola Obrenović

# Primer

**radnik({Mbr, Ime, Prz, Sef, Plt, God, Pre},  
{Mbr}),**

**projekat({Spr, Ruk, Nap, Nar}, {Spr}),**

**radproj({Spr, Mbr, Brc}, {Spr + Mbr}),**

**radnik[Sef]  $\subseteq$  radnik[Mbr],**

**projekat[Ruk]  $\subseteq$  radnik[Mbr],**

**radproj[Mbr]  $\subseteq$  radnik[Mbr],**

**radproj[Spr]  $\subseteq$  projekat[Spr].**

# Tabela radnik

- **Mbr** - maticni broj radnika
- **Ime** - ime radnika
- **Prz** - prezime radnika
- **Sef** - maticni broj direktno nadređenog rukovodioca - radnika
- **Plt** - mesecni iznos plate radnika
- **God** - Datum rođenja radnika
- **Pre** – godišnja premija na platu radnika

Obeležja Mbr, Ime, Prz ne smeju imati null vrednost.  
Plata ne sme biti manja od 500

# Tabela projekt

- **Spr** - sifra projekta
- **Ruk** - rukovodilac projekta
- **Nap** - naziv projekta
- **Nar** - narucilac projekta

Obeležja Spr i Ruk ne smeju imati null vrednost, dok obeležje Nap mora imati jedinstvenu vrednost

# Tabela radproj

- **Spr** - sifra projekta
- **Mbr** - maticni broj radnika
- **Brc** - broj casova nedeljnog angazovanja na projektu

Sva tri obeležja ne smeju da imaju null vrednost

# Izražavanje upita i osnovna struktura naredbe SELECT

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

```
SELECT *| <lista_obeležja>  
FROM <lista_tabela>  
WHERE <uslov_selekcije>
```

<lista\_obeležja> sadrži obeležja nad kojima se formira rezultat upita, <lista\_tabela> sadrži nazive tabela potrebne za realizaciju upita, <uslov\_selekcije> izražava uslov selekcije podataka iz tabela koje su navedene iza službene reči FROM

# Upiti nad jednom tabelom

- Izlistati sadržaj svih tabela.

```
select * from radnik;
```

```
select * from projekat;
```

```
select * from radproj;
```

# Upiti nad jednom tabelom

- Prikazati imena i prezimena svih radnika.

```
select ime, prez  
from radnik;
```



# DISTINCT

```
SELECT [DISTINCT] <lista_obeležja>  
FROM <lista_tabela>  
WHERE <uslov_selekcije>
```

- Izlistati različita imena radnika.

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

# Aritmetički izrazi

```
SELECT *|[DISTINCT] <lista_obeležja>|izraz  
FROM <lista_tabela>  
WHERE <uslov_selekcije>
```

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

```
select mbr, ime, prz  
from radnik  
where plt>25000;
```

# Aritmetički izrazi

```
SELECT *|[DISTINCT] <lista_obeležja>|izraz  
FROM <lista_tabela>  
WHERE <uslov_selekcije>
```

- Izlistati godišnju platu svakog radnika.

```
select mbr, ime, prz, plt*12  
from radnik;
```

# Null vrednost

**x IS NULL – x je nula vrednost**

**x IS NOT NULL – x nije nula vrednost**

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

```
SELECT mbr, ime, prz from radnik  
where sef is null;
```

# Null vrednost

- Prikazati matične brojeve radnika, kao i plate, uvećane za NULL vrednost.

**SELECT Mbr, Plt + NULL FROM Radnik;**

- Prikazati matične brojeve radnika, kao i plate, uvećane za godišnju premiju.

**SELECT Mbr, Plt + Pre FROM Radnik;**

# Funkcija ISNULL(*izraz*, *konstanta*)

- Prikazati matične brojeve radnika, kao i plate, uvećane za godišnju premiju. Ukoliko za nekog radnika vrednost premije ne postoji, smatrati da ona iznosi 0.

```
SELECT Mbr, Plt + ISNULL(Pre, 0)  
FROM Radnik;
```

# BETWEEN

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

**select mbr, ime, prz**

**from radnik**

**where plt between 20000 and 24000;**

# BETWEEN

- Izlistati ime, prz, god radnika rođenih između 1953 i 1975.

**select ime, prz, god**

**from radnik**

**where god between '01-jan-1953' and '31-dec-1975';**



# NOT BETWEEN

- Izlistati ime, prz, god radnika koji nisu rođeni između 1953 i 1975.

**select ime, prz, god**

**from radnik**

**where god not between '01-jan-1953' and  
'31-dec-1975';**

# LIKE

**<obeležje> LIKE <uzorak>**

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

```
select mbr, ime, prz  
from radnik  
where prz like 'M%';
```

# NOT LIKE

- Izlistati mbr, ime, prz radnika čije ime ne počinje slovom A.

```
select mbr, ime, prz  
from radnik  
where ime not like 'A%';
```

# LIKE

- Izlistati mbr, ime, prz radnika čije ime sadrži slovo **A** na drugoj poziciji.

```
select mbr, ime, prz  
from radnik  
where ime like '_a%';
```

# LIKE

- Izlistati imena radnika koja počinju na slovo E.

```
select distinct ime  
from radnik  
where ime like 'E%';
```

# LIKE

- Izlistati radnike koji u svom prezimenu imaju slovo E.

```
select mbr, ime, prz  
from radnik  
where prz like '%e%';
```

# IN

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

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

# IN

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

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



# IN

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

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

# NOT IN

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

```
select mbr, ime, prz  
from radnik  
where ime not in ('Ana', 'Sanja');
```

# Uređivanje izlaznih rezultata

```
SELECT *| <lista_obeležja>  
FROM <lista_tabela>  
WHERE <uslov_selekcije>  
ORDER BY <podlista_obeležja>
```

**ORDER BY je uvek poslednja klauzula naredbe SELECT**

# ORDER BY

- Prikazati radnike koji imaju šefa sortirano po prezimenu.

```
select mbr, ime, prz, plt  
from radnik  
where sef is not null  
order by prz asc;
```

# ORDER BY

- Neki primeri upotrebe klauzule ORDER BY.

```
SELECT Mbr, Ime, Prz, Plt  
FROM Radnik ORDER BY Prz, Ime
```

```
SELECT Mbr, Prz, Ime, Plt  
FROM Radnik ORDER BY Prz ASC, Ime ASC
```

```
SELECT Mbr, Prz, Ime, Plt  
FROM Radnik ORDER BY Prz ASC, Ime DESC
```

# ORDER BY

- Neki primeri upotrebe klauzule ORDER BY.

```
SELECT Mbr, Prz, Ime  
FROM Radnik  
ORDER BY 2, 3, Plt
```

```
SELECT Mbr, Prz, Ime  
FROM Radnik  
ORDER BY 2, 3, Plt * 1.17
```

# ORDER BY

- Prikazati matične brojeve, imena, prezimena i plate radnika, po opadajućem redolsedu iznosa plate.

```
SELECT Mbr, Ime, Prz, Plt Plata  
FROM Radnik  
ORDER BY Plata DESC
```

# Uređivanje izlaznih rezultata

- Prikazati matične brojeve, konkatenirana imena i prezimena radnika, kao i plate, uvećane za 17%.

```
SELECT Mbr,  
Ime + ' ' + Prz "Ime i prezime",  
Plt * 1.17 Plata  
FROM Radnik;
```



# Funkcija count

- COUNT(\*) – vraća ukupan broj selektovanih torki
- COUNT(<obeležje>) – vraća ukupan broj selektovanih torki, za koje vrednost <obeležja> nije nula vrednost
- COUNT(DISTINCT <obeležje>) – vraća ukupan broj različitih torki, za koje vrednost <obeležja> nije nula vrednost

# Funkcija count

- Koliko ima radnika?

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

- Koliko ima šefova?

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

# Funkcije max i min

- $\text{MAX}(\langle \text{obeležje} \rangle)$  – vraća maksimalnu vrednost za  $\langle \text{obeležje} \rangle$ , uzimajući u obzir sve selektovane torke
- $\text{MIN}(\langle \text{obeležje} \rangle)$  – vraća minimalnu vrednost za  $\langle \text{obeležje} \rangle$ , uzimajući u obzir sve selektovane torke

# Funkcije max i min

- Prikazati minimalnu i maksimalnu platu radnika.

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

# Funkcija sum

- SUM(<obeležje>) – vraća zbir vrednosti datog <obeležja>, za sve selektovane torke, uključujući višestruko ponavljanje istih torke
- SUM(DISTINCT <obeležje>) – vraća zbir vrednosti datog <obeležja>, za sve različite selektovane torke

# Funkcija sum

- Prikazati broj radnika i ukupnu mesečnu platu svih radnika.

```
select count(*) "Broj radnika",  
sum(plt) "Mesečna plata"  
from radnik;
```

# Funkcija avg

- **AVG(<obeležje>)** – vraća srednju vrednost datog <obeležja>, za sve selektovane torke, uključujući višestruko ponavljanje istih torke
- **AVG(DISTINCT <obeležje>)** – vraća srednju vrednosti datog <obeležja>, za sve različite selektovane torke

# Funkcija avg

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

```
select count(*) “Broj radnika”,  
avg(plt) “Prosecna plata”,  
12*sum(plt) “Godisnja plata”  
from radnik;
```



# Funkcija round

- ROUND(<izraz>, <broj\_decimala>) – vraća zaokruženu vrednost datog <izraza> na dati <broj\_decimala>

# Funkcija round

- Prikazati prosečnu platu svih radnika zaokruženo na jednu decimalu.

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

# GROUP BY

- Koliko radnika radi na projektu i koliko je ukupno angažovanje na svakom projektu?

```
select spr, count(mbr), sum(brc)  
from radproj  
group by spr;
```

# HAVING

- Izlistati mbr radnika koji rade na više od dva projekta.

```
select mbr from radproj  
group by mbr  
having count(spr)>2;
```

```
select mbr, count(spr) from radproj  
group by mbr  
having count(spr)>2;
```

# Spajanje tabela

- Prikazati mbr, prz, ime, plt i brc angažovanja svih radnika koji rade na nekom od projekata.

```
select radnik.mbr, prz, ime, plt, brc  
from radnik, radproj  
where radnik.mbr = radproj.mbr;
```

# Spajanje tabela

- Prikazati mbr, prz, ime, plt i brc angažovanja svih radnika koji rade na projektu sa šifrom 10.

```
select radnik.mbr, prz, ime, plt, brc  
from radnik, radproj  
where spr=10 and  
radnik.mbr = radproj.mbr;
```

# Spajanje tabela

- Prikazati mbr, prz, ime, plt i brc angažovanja svih radnika koji rade na projektu sa šifrom 10.

```
select r.mbr, r.prz, r.ime, r.plt, rp.brc  
from radnik r, radproj rp  
where rp.spr=10 and r.mbr=rp.mbr;
```

# Spajanje tabela

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

```
select distinct mbr, ime, prz, plt  
from radnik, projekat  
where ruk=mbr;
```



# Primer

- Prikazati imena i prezimena rukovodilaca projekata i broj projekata kojima rukovode.

```
select prz, ime, count(spr)  
from radnik r, projekat p  
where ruk=mbr  
group by prz, ime;
```

# Primer

- Prikazati za svakog radnika mbr, prz, ime, ukupan broj projekata i ukupno mesečno angažovanje na projektima na kojima radi.

```
select r.mbr, r.prz, r.ime, count(*),  
sum(rp.brc)  
from radnik r, radproj rp  
where r.mbr=rp.mbr  
group by r.mbr, r.prz, r.ime;
```

# Primer

- Izlistati nazive projekata na kojima se radi više od 15 časova.

```
select nap  
from projekat p, radproj rp  
where p.spr=rp.spr  
group by p.spr, nap  
having sum(brc)>15;
```

# Primer

- Izlistati šifre i nazive projekata na kojima radi više od dva radnika.

```
select p.spr, p.nap  
from projekat p, radproj rp  
where rp.spr=p.spr  
group by p.spr, p.nap  
having count(mbr)>2;
```

# Upit sa višestrukom upotrebom iste tabele

- Prikazati mbr, ime, prz, plt radnika koji zarađuju više od radnika sa matičnim brojem 40.

```
select r.mbr, r.prz, r.ime, r.plt  
from radnik r, radnik r1  
where r.plt>r1.plt and r1.mbr=40;
```

# Upit sa višestrukom upotrebom iste tabele

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

```
select r1.ime, r1.prz, r1.plt, p.nap from  
radnik r1, radnik r2, projekat p, radproj rp  
where r1.mbr=rp.mbr and rp.spr=p.spr and  
p.ruk=r2.mbr and r1.plt+1000<r2.plt;
```

# Unutrašnje spajanje (INNER)

- Prikazati ime i prz radnika koji rade na projektu sa šifrom 30.

```
select ime, prz  
from radnik r inner join radproj rp  
on r.mbr=rp.mbr  
where spr=30;
```

# Spoljno spajanje (OUTER)

- Levo (LEFT)
- Desno (RIGHT)
- Potpuno (FULL)



# Spoljno spajanje (LEFT OUTER)

- Prikazati mbr, ime i prz radnika i šifre projekata na kojima rade. Prikazati, takođe, iste podatke i za radnike koji ne rade ni na jednom projektu, pri čemu za šifru projekta treba, u tom slučaju, prikazati nedostajuću vrednost.

```
select r.mbr,ime, prz, spr  
from radnik r left outer join radproj rp  
on r.mbr=rp.mbr;
```

# Spoljno spajanje (RIGHT OUTER)

- Prikazati nazive svih projekata i mbr radnika koji rade na njima. Ukoliko na projektu ne radi ni jedan radnik ispisati nulu.

```
select isnull(rp.mbr, 0) "Mbr radnika", nap  
from radproj rp right outer join projekat p  
on rp.spr=p.spr;
```

# Spoljno spajanje (FULL OUTER)

```
select isnull(rp.mbr, 0) "Mbr radnika",  
nap  
from radproj rp full outer join projekat p  
on rp.spr=p.spr;
```

# Nezavisni ugnježdjeni upiti

- SELECT unutar WHERE druge SELECT naredbe
- Predikatski izrazi:
  - ALL, IN i EXISTS
- SQL dozvoljava višestruko ugnježdavanje upita

# Nezavisni ugnježdjeni upiti

- Izlistati u rastućem redosledu plate mbr, ime, prz i plt radnika koji imaju platu veću od prosečne.

```
select mbr, ime, prz, plt  
from radnik  
where plt > (select avg(plt) from radnik)  
order by plt asc;
```

# Nezavisni ugnježdeni upiti

- Izlistati imena i prezimena radnika koji rade na projektu sa šifrom 30 (pomoću ugnježdenog upita).

```
select ime, prz from radnik  
where mbr in  
(select mbr from radproj where spr=30);
```

# Nezavisni ugnježdeni upiti

- Izlistati mbr, ime, prz radnika koji rade na projektu sa šifrom 20, a ne rade na projektu sa šifrom 10.

```
select mbr, ime, prz  
from radnik  
where mbr in  
(select mbr from radproj where spr=20)  
and mbr not in  
(select mbr from radproj where spr=10);
```

# Primer

- Izlistati imena, prezimena svih radnika osim rukovodioca projekta sa šifrom 10. (sa ugnježenim upitom)

```
select ime, prz, mbr  
from radnik  
where mbr != (select ruk from projekat  
              where spr=10);
```



# Primer

- Izlistati nazive projekata na kojima radi bar jedan radnik koji radi i na projektu sa šifrom 60.

```
select p.nap from projekat p  
where spr in (select spr from radproj  
where mbr in (select mbr from radproj  
where spr=60));
```

# Primer

- Prikazati imena i prezimena rukovodilaca projekata i broj projekata na kojima rade.

```
select prz, ime, count(distinct rp.spr) bp  
from radnik r, radproj rp  
where r.mbr=rp.mbr and  
r.mbr in (select ruk from projekat)  
group by prz, ime;
```

# Primer

- Izlistati nazive i šifre projekata na kojima je prosečno angažovanje veće od prosečnog angažovanja na svim projektima.

```
select p.spr, p.nap  
from projekat p, radproj rp  
where rp.spr=p.spr  
group by p.spr, p.nap  
having avg(brc)>(select avg(brc) from radproj);
```

# Ažuriranje baze podataka sa ugnježenim upitima

- Isključiti radnike sa onih projekta gde im je angažovanje na projektu manje od prosečnog angažovanja radnika na svim projektima

**delete from radproj**

**where brc < (select avg(brc) from  
radproj)**

# Ažuriranje baze podataka sa ugnježenim upitima

- Svim radnicima koji rade na projektu sa šifrom 10 povećati premiju za 500

**update radnik**

**set pre = isnull(pre,0) + 500**

**where mbr in (select mbr from radproj  
where spr = 10)**

# ALL

$x \Theta \text{ ALL } (<\text{lista\_vrednosti}> \text{ koju vraća } \text{select upit})$

$\Theta \in \{<, >, <=, >=, !=\}$

Primer:

$x != \text{ ALL } (<\text{lista\_vrednosti}>)$

$x$  je različito od svih vrednosti u  $<\text{listi\_vrednosti}>$

# Nezavisni ugnježdeni upiti (ALL)

- Izlistati ime, prz i god najstarijeg radnika.

```
select mbr, ime, prz, god  
from radnik  
where god <= all(select god from radnik);
```

```
select mbr, ime, prz, god  
from radnik  
where god = (select min(god) from  
radnik);
```

# Primer

- Izlistati nazive i šifre projekata na kojima je prosečno angažovanje veće od prosečnog angažovanja na svakom projektu (najveće prosečno angažovanje).

```
select p.spr, p.nap  
from projekat p, radproj rp  
where rp.spr=p.spr  
group by p.spr, p.nap  
having avg(brc)>=all(select avg(brc)  
from radproj group by spr);
```



# Povezani upiti

- Izlistati radnike koji nisu rukovodioci projekata.  
(ne postoji projekat kojim rukovodi taj radnik)

```
select mbr, ime, prz  
from radnik r  
where mbr not in  
(select ruk from projekat where mbr=ruk);
```

# Povezani upiti

- Prikazati mbr, ime, prz, plt radnika čiji je broj sati angažovanja na nekom projektu veći od prosečnog broja sati angažovanja na tom projektu.

```
select distinct r.mbr, ime, prz, plt, brc  
from radnik r, radproj rp1  
where r.mbr=rp1.mbr and  
rp1.brc>(select avg(brc) from radproj rp2  
where rp2.spr=rp1.spr);
```

# EXISTS

**EXISTS(<lista\_vrednosti>)** –  
<lista\_vrednosti> nije prazan skup  
vrednosti

**NOT EXISTS(<lista\_vrednosti>)** –  
<lista\_vrednosti> je prazan skup vrednosti

# EXISTS

- Izlistati radnike koji su rukovodioci projekata. (Postoji projekat kojim rukovodi taj radnik)

```
select mbr, ime, prz  
from radnik r  
where exists  
(select * from projekat where mbr=ruk);
```

```
select mbr, ime, prz  
from radnik r  
where mbr in  
(select ruk from projekat where mbr=ruk);
```

# EXISTS

- Izlistati radnike koji ne rade ni na jednom projektu. (ne postoji projekat na kom rade)

```
select mbr, ime, prz  
from radnik r  
where not exists  
(select * from radproj rp where r.mbr=rp.mbr);
```

```
select mbr, ime, prz  
from radnik r  
where mbr not in  
(select rp.mbr from radproj rp  
where r.mbr=rp.mbr);
```

# Kreiranje, izmena i brisanje definicije pogleda

**CREATE VIEW <naziv\_pogleda> [(alias  
[, alias]...)]**

**AS podupit;**

- Podupit koji se koristi za definisanje pogleda može biti kompleksan, ali ne sme da sadrži ORDER BY.
- ORDER BY se koristi tek kada se prikazuju podaci iz pogleda.

# Pogledi obezbeđuju sledeće prednosti

- ograničavaju pristup bazi podataka
- pojednostavljaju upite
- obezbeđuju nezavisnost podataka
- obezbeđuju višestruke poglede nad istim podacima
- mogu se brisati bez uklanjanja podataka u osnovnim tabelama.

# Modifikacija pogleda

- Pogledi se modifikuju pomoću ALTER VIEW naredbe (nova definicija zamenjuje staru).
- Dakle, pogled može biti izmenjen bez brisanja postojećeg pogleda.
- Na primer, mogu se dodati alijasi za kolone u pogledu.



# Kreiranje složenog pogleda

- Ukoliko se u upitu pomoću kog se kreira pogled nalaze skupovne funkcije (min, max, avg, sum, count) ili izrazi, u pogledu se moraju definisati alternativna imena za te kolone.

# DML operacije sa pogledima

- DML (Data Manipulation Language) operacije se mogu primenjivati na jednostavnim pogledima.
- Ako pogled sadrži **skupovne funkcije**, **group by** kaluzulu, **distinct** rezervisanu reč ili **rownum** rezervisanu reč, vrsta iz pogleda se ne može izbrisati.
- Isto važi i za modifikaciju podataka, s tim što dodatno važi i da se **kolone definisane izrazima** ne mogu modifikovati (npr, salary\*12).
- U pogled se ne mogu dodavati podaci ako pogled sadrži **skupovne funkcije**, **group by** kaluzulu, **distinct** rezervisanu reč, **rownum** rezervisanu reč, **kolonu koja je definisana izrazom**, ili **not null** kolonu u baznoj tabeli koja nije selektovana u pogledu.
- Dodavanjem vrednosti u pogled, one se dodaju direktno u baznu tabelu.

# Brisanje pogleda

**DROP VIEW pogled;**

# Pogled

- Kreirati pogled koji će prikazati mbr, ime i prz radnika i nazive projekata kojima rukovode.

**CREATE VIEW rukovodioci (Mbr, Ime, Prezime, Projekat) AS**

```
select r.mbr,ime, prz, nap  
from radnik r, projekat p  
where r.mbr=p.ruk
```

# Pogled

```
SELECT Ime, Prezime, Projekat  
FROM rukovodioci
```

# Pogled

- Kreirati pogled koji će prikazati mbr, ime i prz svih radnika i nazive projekata kojima rukovode. Ukoliko radnik ne rukovodi ni jednim projektom ispisati: ne rukovodi projektom.

```
CREATE VIEW rukovodioci_svi (Mbr, Ime, Prezime,  
Projekat) AS  
select r.mbr,ime, prz, isnull(nap, 'ne rukovodi  
projektom')  
from radnik r left outer join projekat p  
on r.mbr=p.ruk
```

# Pogled

- Napraviti pogled koji će za svakog radnika prikazati njihova imena i prezimena kao i prezimena njihovih šefova ako ih imaju. Ako nema šefa ispisati: nema sefa.

```
CREATE VIEW radnici_sefovi (Ime, Prezime, Sef) AS  
select r1.ime, r1.prz, isnull(r2.prz, 'Nema sefa')  
from radnik r1 left outer join radnik r2  
on r1.sef=r2.mbr
```

# Pogled

- Napraviti pogled koji će za sve radnike prikazati Mbr i ukupan broj sati angažovanja radnika na projektima na kojima radi.



# Pogled

```
CREATE VIEW angaz_po_radnicima (Mbr, SBrc) AS  
SELECT r.Mbr, ISNULL(SUM(rp.Brc), 0)  
FROM radnik r LEFT OUTER JOIN radproj rp  
ON r.Mbr = rp.Mbr  
GROUP BY r.Mbr
```

# Pogled

- Nakon toga, napraviti pogled koji će za svakog šefa (rukovodioca radnika) prikazati njegov matični broj, prezime, ime, ukupan broj radnika kojima rukovodi i njegovo ukupno angažovanje na svim projektima, na kojima radi. Koristiti prethodno definisani pogled.

# Pogled

```
CREATE VIEW angaz_sefova (Mbr, Prz, Ime, BrRad,  
BrSat) AS  
SELECT DISTINCT r.Sef, r1.Prz, r1.Ime, COUNT(*),  
a.SBrc  
FROM radnik r, radnik r1, angaz_po_radnicima a  
WHERE r.Sef = r1.Mbr AND r.Sef = a.Mbr  
GROUP BY r.Sef, r1.Prz, r1.Ime, a.SBrc
```

# Pogled

- Koliko je ukupno angažovanje svih šefova na projektima?

```
SELECT SUM(BrSat) AS UkAngSef  
FROM angaz_sefova
```

# Funkcije nad alfanumeričkim tipovima podataka

- **SUBSTRING**(niz karaktera, početni indeks, broj karaktera podniza) – vraća podniz karaktera izdvojen iz niza karaktera od početnog indeksa
- **CHARINDEX**(podniz karaktera, niz karaktera) – vraća indeks na kojem počinje podniz karaktera u nizu karaktera
- **LEN**(niz karaktera) – broj karaktera (dužina) u nizu karaktera
- **REPLACE**(niz karaktera, podniz karaktera, niz karaktera zamena) – vraća novi niz karaktera u kojem je podniz zamenjen sa “niz karaktera zamena”

# Funkcije nad alfanumeričkim tipovima podataka

- Za svaki projekat prikazati njegov naziv, prva tri karaktera u nazivu, dužinu naziva, indeks znaka space (razmak), kao i naziv projekta bez razmaka

```
SELECT nap "Naziv projekta",  
SUBSTRING(nap, 1, 3), LEN(nap),  
CHARINDEX(' ', nap), REPLACE(nap, ' ', '')  
FROM projekat
```

# Funkcije nad datumskim tipovima podataka

- **GETDATE()** – trenutno vreme i datum
- **DATENAME**(day | weekday | week | month | year, datum) – vraća dan, dan u nedelji, nedelju, mesec ili godinu iz datuma
- **DAY**(datum) – vraća dan iz datuma
- **MONTH**(datum) – vraća mesec iz datuma
- **YEAR**(datum) – vraća godinu iz datuma

# Funkcije nad datumskim tipovima podataka

- Za svakog radnika prikazati njegov matični broj, dan, mesec, godinu, dan u nedelji rođenja, kao i trenutno vreme i datum

```
SELECT DAY(god) Dan, MONTH(god) Mesec,  
DATENAME(year, god) Godina,  
DATENAME(weekday, god) "Dan u nedelji",  
GETDATE() FROM radnik;
```