

# BAZE PODATAKA

---

Predavanje broj: 05

Nastavne teme:

- ✓ LIMITIRANJE REZULTATA
- ✓ AGREGATNE FUNKCIJE
- ✓ HAVING KLAUZULA
- ✓ POVEZIVANJE TABELA: PRIRODNO SPAJANJE

# LIMIT - Limitiranje rezultata

Klauzula **LIMIT** ograničava broj redova rezultata koje upit daje.

```
SELECT prezime, ime  
FROM radnik  
LIMIT 3
```

|   | prezime | ime        |
|---|---------|------------|
| ▶ | Vasić   | Petar      |
|   | Marić   | Aleksandar |
|   | Kondić  | Vanja      |

Klauzula **LIMIT** može se koristiti i sa definisanjem takozvanog offset. Ako se pomoću prethodnog upita žele učitati redovi od 4 do 6, to će se uraditi na sledeći način:

```
SELECT prezime, ime  
FROM radnik  
LIMIT 3, 3
```

|   | prezime | ime     |
|---|---------|---------|
| ▶ | Perić   | Jovan   |
|   | Mančić  | Janko   |
|   | Dimić   | Mirjana |

# Limitiranje rezultata

Kada se u klauzuli LIMIT zadaju **dva parametra (offset)**, prvi je red od kojeg počinje učitavanje, a drugi je maksimalan broj redova koji se želi učitati.

Kada postoji samo jedan parametar on predstavlja maksimalan broj redova koji se želi učitati. Kada se zadaje pomak, on počinje od 0.

Ako se želi da upit vrati redove od pomaka do kraja tabele za vrednost drugog parametra treba navesti neki jako veliki broj.

Klauzula LIMIT se najčešće koristi u kombinaciji sa odredbom ORDER BY da bi redosled redova u rezultatima upita imao određeni smisao.

Treba imati na umu da bez odredbe ORDER BY, redosled redova rezultata nije predvidiv.

# Funkcije agregacije

Funkcije agregacije su dobile naziv po tome što vrše agregaciju rezultata upita. Mogu se smatrati i višerednim ili grupnim funkcijama s obzirom da se primenjuju nad grupom redova da bi dale jedan rezultat po svakoj grupi.

- **AVG (kolona)** - izračunava srednju vrednost datog atributa
- **SUM (kolona)** - izračunava sumu svih vrednosti atributa
- **MIN (kolona)** - nalazi minimalnu vrednost atributa
- **MAX (kolona)** - nalazi najveću vrednost atributa
- **COUNT(\*)** - nalazi broj redova u tabeli (grupi)
- **COUNT(kolona)** - nalazi broj redova sa ne NULL vrednostima kolone
- **COUNT (DISTINCT kolona)** - nalazi broj redova sa različitim vrednostima zadate kolone.

# Funkcije agregacije

Funkcije grupe operišu nad grupom redova da bi dale jedan rezultat po redu. One uzimaju više redova kao ulaz, a onda vraćaju jedan red kao izlaz.

```
SELECT MAX(salary), MIN(salary), AVG(salary) FROM employees;
```

| MAX(SALARY) | MIN(SALARY) | AVG(SALARY) |
|-------------|-------------|-------------|
| 24000       | 2500        | 8775        |

```
SELECT MAX(salary)  
FROM employees;
```

| DEPT_ID | SALARY |
|---------|--------|
| 90      | 24000  |
| 90      | 17000  |
| 90      | 17000  |
| 60      | 9000   |
| 60      | 6000   |
| 60      | 4200   |
| 50      | 5800   |
| 50      | 3500   |
| 50      | 3100   |
| 50      | 2600   |
| 50      | 2500   |
| ...     | ...    |
| 60      | 11000  |
| 60      | 8600   |
|         | 7000   |
| 10      | 4400   |

| MAX (SALARY) |
|--------------|
| 24000        |

# Funkcije agregacije

- Zanemaruju null-vrednosti.
- **Ne mogu** se koristiti u klauzuli **WHERE**.
  - ~~SELECT \* FROM radnik WHERE plata > AVG(plata)~~
  - Error Code: 1111. Invalid use of group function
- **MIN, MAX i COUNT** se mogu koristiti sa **bilo kojim tipom podatka**.
- **SUM, AVG, STDDEV i VARIANCE** se mogu koristiti **samo sa numeričkim tipom podataka**.

# MIN(), MAX(), AVG()

Prikazati minimalnu, maksimalnu i prosečnu platu radnika koji su zaposleni posle 2000 godine.

```
SELECT min(plata) as 'Minimalna plata', max(plata) as  
'Maksimalna plata', avg(plata) as 'Prosečna plata'  
FROM radnik  
WHERE year(dat_zap)>2000
```

|   | Minimalna plata | Maksimalna plata | Prosečna plata |
|---|-----------------|------------------|----------------|
| ▶ | 900             | 3900             | 2100           |

# MIN(), MAX(), AVG()

Prikazati datum najkasnijeg zaposlenja, prezime radnika koje je na vrhu abecednog spiska i prezime radnika koje je na kraju spiska zaposlenih.

```
SELECT max(dat_zap) as 'Najkasniji datum zaposlenja',  
min(prezime) as 'Prvi na spisku', max(prezime) as 'Zadnji na spisku'  
FROM radnik
```

|   | Najkasniji datum zaposlenja | Prvi na spisku | Zadnji na spisku |
|---|-----------------------------|----------------|------------------|
| ▶ | 2004-05-20 00:00:00         | Bogovac        | Vuković          |

# COUNT(kolona\_za\_brojanie)

Često se može javiti potreba da utvrdite ukupan broj zapisa u nekoj tabeli. Jednostavno, nisu Vam potrebni nikakvi detalji, već samo ukupan broj zapisa, tj. redova ili linija jedne tabele.

- COUNT(kolona\_za\_brojanie) broji redove u kojima kolona\_za\_brojanie ima ne-null vrednost i koji zadovoljavaju WHERE uslov ako je dat.
- COUNT(\*) koristimo ako želimo da izbrojimo sve redove ili sve redove koji zadovoljavaju dati uslov ako je data klauzula WHERE.

| kol_1 | kol_2 | kol_3 | kol-4 |
|-------|-------|-------|-------|
| 3     | -     | aba   | -     |
| 6     | a     | -     | -     |
| -     | a     | -     | -     |
| 3     | a     | aca   | -     |

COUNT(kol\_1) = 3

COUNT(kol\_2) = 3

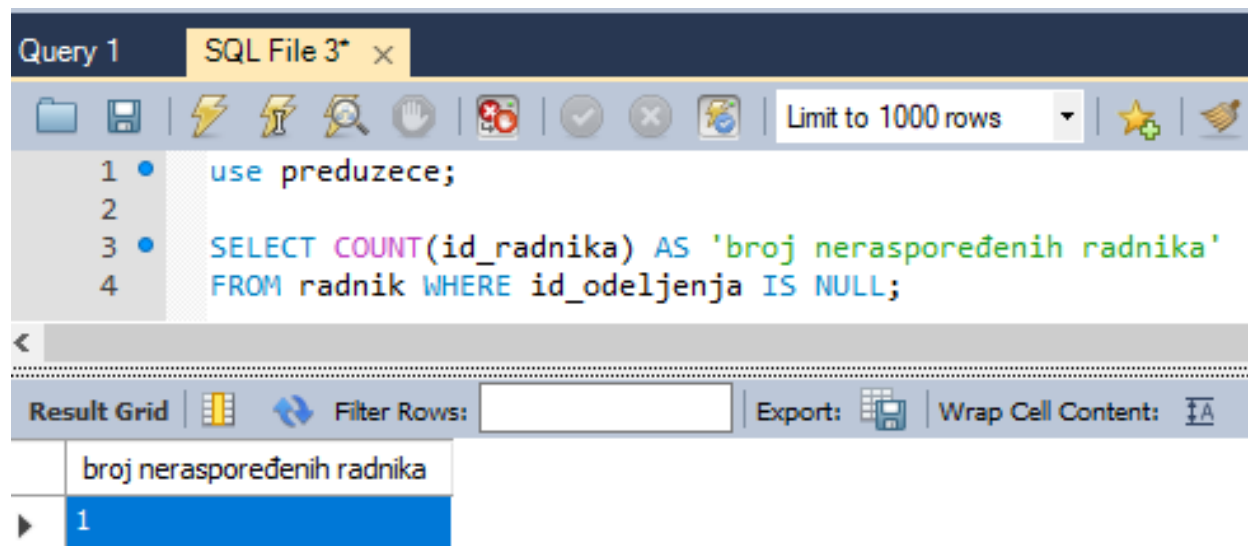
COUNT(kol\_3) = 2

COUNT(kol\_4) = null    COUNT(\*) = 4

# COUNT (kolona\_za\_brojanje)

Prikazati broj neraspoređenih radnika.

```
SELECT COUNT(id_radnika) AS 'broj neraspoređenih radnika'  
FROM radnik  
WHERE id_odeljenja IS NULL;
```



# COUNT (DISTINCT ime\_kolone)

Ukoliko je potrebno da utvrdimo koliko određenih jedinstvenih zapisa postoji, potrebno je koristiti ključnu reč DISTINCT odnosno želimo da izbrojimo samo različite vrednosti kolone **kolona \_za\_brojanje**:

COUNT(DISTINCT kolona\_za\_brojanje)

| kol_1 | kol_2 | kol_3 | kol-4 |
|-------|-------|-------|-------|
| 3     | -     | aba   | -     |
| 6     | a     | -     | -     |
| -     | a     | -     | -     |
| 3     | a     | aca   | -     |

COUNT(DISTINCT kol\_1) = 2    COUNT(kol\_1) = 3

COUNT(DISTINCT kol\_2) = 1    COUNT(kol\_2) = 3

COUNT(DISTINCT kol\_3) = 2

# COUNT (DISTINCT ime\_kolone)

| CD_NUMBER | TITLE                              | PRODUCER                      | YEAR |
|-----------|------------------------------------|-------------------------------|------|
| 90        | The Celebrants Live in Concert     | Old Town Records              | 1997 |
| 91        | Party Music for All Occasions      |                               |      |
|           | Back to the Shire                  | The Music Man                 | 2000 |
| 92        | Songs from My Childhood Carpe Diem | Middle Earth Records Old Town | 2002 |
| 93        | Here Comes the Bride Graduation    | Records                       | 1999 |
| 94        | Songbook Whirled Peas              | R & B Inc.                    | 2001 |
| 95        |                                    | The Music Man Tunes Are       | 2001 |
| 96        |                                    | Us                            | 1998 |
| 98        |                                    | Old Town Records              | 2000 |

```
SELECT COUNT(DISTINCT year) as "Years"  
FROM d_cds;
```

# IFNULL()

Za sve zaposlene prikazati njihove identifikacione brojeve, imena i ukupna primanja.

Ukupna primanja zaposlenog dobijemo kada saberemo platu i premiju. Pošto polje PREMIJA može imati Null vrednost, da je zadatak urađen na sledeći način, bili bi isključeni svi zaposleni koji ne primaju premiju iako oni primaju platu:

```
SELECT idbr, ime, plata+premija AS [ukupna primanja]  
FROM RADNIK;
```

```
SELECT idbr, ime, plata+ IFNULL(premija, 0) AS [Ukupna primanja]  
FROM RADNIK;
```

# AVG() i IFNULL()

Prikazati prosečnu vrednost premije zaposlenih.

| Result Grid        |          |          |            |        |             |                     |         |       |             |
|--------------------|----------|----------|------------|--------|-------------|---------------------|---------|-------|-------------|
| Filter Rows:       |          |          |            |        |             |                     |         |       |             |
| Edit:              |          |          |            |        |             |                     |         |       |             |
| Export/Import:     |          |          |            |        |             |                     |         |       |             |
| Wrap Cell Content: |          |          |            |        |             |                     |         |       |             |
| Id_radnika         | Ime      | Prezime  | Posao      | Kvalif | Rukovodilac | Dat_zap             | Premija | Plata | Id_odeljenj |
| 5519               | Vanja    | Kondić   | prodavac   | VKV    | 5662        | 1991-11-07 00:00:00 | 1300    | 1200  | 10          |
| 5652               | Jovan    | Perić    | električar | KV     | 5662        | 1980-05-31 00:00:00 | 500     | 1000  | 10          |
| 5662               | Janko    | Mančić   | upravnik   | VSS    | 6789        | 1993-08-12 00:00:00 | NULL    | 2400  | 10          |
| 5696               | Mirjana  | Dimić    | čistač     | KV     | 5662        | 1991-09-30 00:00:00 | 0       | 1000  | 10          |
| 5780               | Božidar  | Ristić   | upravnik   | VSS    | 6789        | 1984-08-11 00:00:00 | NULL    | 2200  | 20          |
| 5786               | Pavle    | Šotra    | upravnik   | VSS    | 6789        | 1983-05-22 00:00:00 | NULL    | 2800  | 30          |
| 5842               | Miloš    | Marković | direktor   | VSS    | NULL        | 1981-12-15 00:00:00 | NULL    | 3000  | 40          |
| 5867               | Svetlana | Grubač   | savetnik   | VSS    | 5842        | 1970-08-08 00:00:00 | NULL    | 2750  | 40          |
| 5874               | Tomislav | Bogovac  | električar | KV     | 5662        | 1971-04-19 00:00:00 | 1100    | 1000  | 10          |
| 5898               | Andrija  | Ristić   | nabavljač  | KV     | 5786        | 1980-01-20 00:00:00 | 1200    | 1100  | 30          |
| 5900               | Slobodan | Petrović | vozač      | KV     | 5780        | 2002-10-03 00:00:00 | 1300    | 900   | 20          |
| 5932               | Mitar    | Vuković  | savetnik   | VSS    | 5842        | 2000-03-25 00:00:00 | NULL    | 2600  | 20          |
| 5953               | Jovan    | Perić    | nabavljač  | KV     | 5786        | 1979-01-12 00:00:00 | 0       | 1100  | 30          |
| 6234               | Marko    | Nastić   | analitičar | VSS    | 5867        | 1990-12-17 00:00:00 | 3000    | 1300  | 30          |
| 6789               | Janko    | Simić    | upravnik   | VSS    | 5842        | 2003-12-23 00:00:00 | 10      | 3900  | 40          |
| 7890               | Ivan     | Buha     | analitičar | VSS    | 5867        | 2003-12-17 00:00:00 | 3200    | 1600  | 20          |
| 7892               | Luka     | Bošković | analitičar | VSS    | 5867        | 2004-05-20 00:00:00 | NULL    | 2000  | NULL        |

# AVG() i IFNULL()

Prikazati prosečnu vrednost premije zaposlenih.

```
SELECT AVG(Premija) FROM radnik;
```

|   |              |
|---|--------------|
|   | AVG(Premija) |
| ▶ | 1192.5       |

```
SELECT AVG(IFNULL(Premija,0))  
FROM radnik;
```

|                                                                                                                                                                                                                         |                        |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| Result Grid     Filter Rows: <input type="text"/> |                        |
|                                                                                                                                                                                                                         | AVG(IFNULL(Premija,0)) |
| ▶                                                                                                                                                                                                                       | 753.1578947368421      |

# GROUP BY() i funkcije agregacije

## Pravila

- Ako u klauzulu SELECT uključite grupnu funkciju (AVG, SUM, COUNT, MAX, MIN...) i još neku pojedinačnu kolonu, svaka tako navedena pojedinačna kolona mora biti navedena i u klauzuli GROUP BY.
- Alijasi kolona se ne mogu koristiti u klauzuli GROUP BY.
- Klauzula WHERE isključuje vrste (redove) pre njihove podele na grupe. Zato se grupne funkcije ne mogu koristiti u klauzuli WHERE.
- Za definisanje uslova selekcije na osnovu vrednosti grupnih funkcija koristi se klauzula HAVING

# GROUP BY() i funkcije agregacije

```
SELECT id_odeljenja AS 'šifra odeljenja',  
AVG(plata) AS 'prosečna plata'  
FROM radnik  
GROUP BY id_odeljenja
```

| šifra<br>odeljenja | prosečna plata     |
|--------------------|--------------------|
| NULL               | 2000               |
| 10                 | 1266.6666666666667 |
| 20                 | 1720               |
| 30                 | 1575               |
| 40                 | 3216.6666666666665 |

```
SELECT id_odeljenja AS 'šifra odeljenja',  
AVG(plata) AS 'prosečna plata'  
FROM radnik
```

NIJE DOBAR ODZIV!

| šifra<br>odeljenja | prosečna plata     |
|--------------------|--------------------|
| 20                 | 1797.3684210526317 |

# GROUP BY() i funkcije agregacije

```
SELECT id_odeljenja AS 'šifra odeljenja',  
AVG(plata) AS 'prosečna plata'  
FROM radnik  
WHERE id_odeljenja IS NOT NULL  
GROUP BY id_odeljenja
```

| šifra<br>odeljenja | prosečna plata     |
|--------------------|--------------------|
| 10                 | 1266.6666666666667 |
| 20                 | 1720               |
| 30                 | 1575               |
| 40                 | 3216.6666666666665 |

# GROUP BY() i funkcije agregacije

MySQL omogućava da se izabere redosled grupa kojim se prikazuju rezultati. Podrazumevani je rastući redosled. Sledeći upit je isti kao prethodni, ali se rezultati prikazuju opadajućim redosledom:

```
SELECT id_odeljenja AS 'šifra odeljenja', AVG(plata) AS 'prosečna plata'  
FROM radnik  
WHERE id_odeljenja IS NOT NULL  
GROUP BY id_odeljenja DESC
```

|   | šifra<br>odeljenja | prosečna plata     |
|---|--------------------|--------------------|
| ► | 40                 | 3216.6666666666665 |
|   | 30                 | 1575               |
|   | 20                 | 1720               |
|   | 10                 | 1266.6666666666667 |

# GROUP BY() i funkcije agregacije

---

Prikazati poslove radnika, prosečnu platu za svaki posao i broj izvršilaca.

# GROUP BY() i funkcije agregacije

```
SELECT posao, ROUND(AVG(plata),2) AS 'prosečna  
plata',  
COUNT(*) AS 'broj izvršilaca'  
FROM radnik  
GROUP BY posao
```

| posao      | prosečna<br>plata | broj<br>izvršilaca |
|------------|-------------------|--------------------|
| analitičar | 1633.33           | 3                  |
| čistač     | 1000.00           | 1                  |
| direktor   | 3000.00           | 1                  |
| električar | 1000.00           | 3                  |
| nabavljač  | 1100.00           | 2                  |
| prodavac   | 1200.00           | 1                  |
| savetnik   | 2675.00           | 2                  |
| upravnik   | 2825.00           | 4                  |
| vozač      | 1100.00           | 2                  |

# GROUP BY() i funkcije agregacije

---

Prikazati po odeljenjima kvalifikacije, ukupnu platu radnika sa tom kvalifikacijom i broj radnika.  
Rezultati treba da budu uređeni po kvalifikaciji u rastućem redosledu.

# GROUP BY() i funkcije agregacije

```
SELECT id_odeljenja AS 'šifra odeljenja', kvalif AS 'kvalifikacija',  
SUM(plata) AS 'ukupna plata', COUNT(*) AS 'broj radnika'  
FROM radnik  
GROUP BY id_odeljenja, kvalif  
ORDER BY kvalif
```

**Rezultat grupisanja po  
kvalifikaciji i šifri odeljenja;**

**Ukupna plata i broj izvršilaca za  
svaku podgrupu**

**Da li treba dodati još neki  
uslov???**

| šifra<br>odeljenja | kvalifikacija | ukupna<br>plata | broj<br>radnika |
|--------------------|---------------|-----------------|-----------------|
| 30                 | KV            | 2200            | 2               |
| 20                 | KV            | 2200            | 2               |
| 10                 | KV            | 4000            | 4               |
| 10                 | VKV           | 1200            | 1               |
| NULL               | VSS           | 2000            | 1               |
| 10                 | VSS           | 2400            | 1               |
| 20                 | VSS           | 6400            | 3               |
| 30                 | VSS           | 4100            | 2               |
| 40                 | VSS           | 9650            | 3               |

# HAVING klauzula

- Klauzulom **HAVING** (“koji imaju”) nameću se uslovi na grupe: vrši se filtracija rezultat nakon grupisanja.
- Ova klauzula zahteva da je prethodno izvršeno grupisanje klauzulom **GROUP BY**.
- Dok se klauzula **WHERE** primenjuje na sve n-torke pre nego što oni postanu delovi grupe, klauzula **HAVING** se primenjuje na već agregiranu vrednost za tu grupu.

Ukoliko se u upitu koriste sve tri klauzule onda se mora poštovati redosled:

1. **WHERE**,
2. **GROUP BY**,
3. **HAVING**.

**GROUP BY** može bez **HAVING** klauzule. **HAVING** klauzula **NE MOŽE** bez **GROUP BY**.

# HAVING klauzula

---

- Kao što se klauzula WHERE koristi da se ograniči izbor redova, tako se klauzula HAVING koristi za ograničen izbor grupa.
- Ako upit koristi klauzule GROUP BY i HAVING, prvo se grupišu redovi, zatim se izračunavaju grupne funkcije, a zatim se prikazuju samo one grupe koje su saglasne sa klauzulom HAVING.
- Klauzula WHERE se koristi da ograniči izbor redova koji se grupišu; klauzula HAVING ograničava izbor grupa dobijenih klauzulom GROUP BY.

# HAVING klauzula

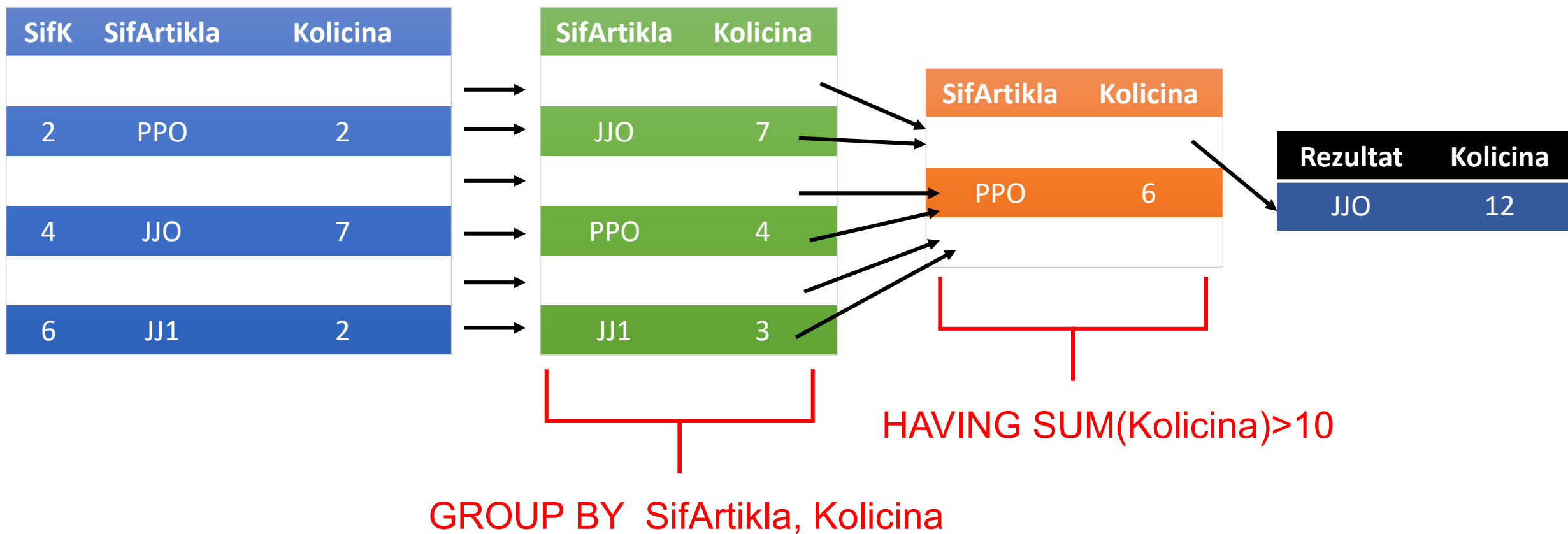
```
SELECT id_odeljenja, AVG(plata) AS 'prosečna plata'  
FROM radnik  
HAVING AVG(plata)>2000
```

|  | id_odeljenja | prosečna<br>plata |
|--|--------------|-------------------|
|--|--------------|-------------------|

```
SELECT id_odeljenja, AVG(plata) AS 'prosečna plata' FROM radnik HAVING AVG(plata)>2000  
0 row(s) returned 0.000 sec / 0.000 sec
```

Ne prijavljaj grešku u MySQL-u, ali ne vrati dobar rezultat!!!!

# HAVING klauzula



# HAVING klauzula

Prikazati minimalnu platu za odeljenja u kojima je minimalna plata veća od 1500.

```
SELECT id_odeljenja AS 'šifra odeljenja', MIN(plata) AS 'minimalna plata'  
FROM radnik  
WHERE id_odeljenja IS NOT NULL  
GROUP BY id_odeljenja  
HAVING MIN(plata)>1500
```

| šifra<br>odeljenja | minimalna<br>plata |
|--------------------|--------------------|
| 40                 | 2750               |

# HAVING klauzula

---

Prikazati odeljenja u kojima radi 1 ili 2 radnika VKV kvalifikacije.

# HAVING klauzula

```
SELECT id_odeljenja, kvalif, COUNT(*) AS 'broj radnika'  
FROM radnik WHERE kvalif='VKV'  
GROUP BY id_odeljenja, kvalif  
HAVING COUNT(*)>=1 OR COUNT(*)<=2;
```

| id_odeljenja | kvalif | broj<br>radnika |
|--------------|--------|-----------------|
| 10           | VKV    | 1               |

# HAVING klauzula

---

Prikazati maksimalnu platu odeljenja i broj zaposlenih za odeljenja u kojima radi od 2 do 5 radnika. Prikaz sortirati po maksimalnoj plati u opadajućem redosledu.

# HAVING klauzula

```
SELECT id_odeljenja, MAX(plata) AS 'maksimalna plata', COUNT(*) AS 'broj radnika'  
FROM radnik  
GROUP BY id_odeljenja  
HAVING COUNT(*) BETWEEN 2 AND 5  
ORDER BY MAX(plata);
```

| id_odeljenja | maksimalna<br>plata | broj<br>radnika |
|--------------|---------------------|-----------------|
| 20           | 2600                | 5               |
| 30           | 2800                | 4               |
| 40           | 3900                | 3               |

# OBAVEZNO ZAPAMTI

**SELECT** *kolona1, kolona2 , grupna funkcija1, grupna funkcija2, ...*

**FROM** *tabela*

**WHERE** *uslov na nivou reda – NE MOŽE AGEATNA (GRUPNA) FUNKCIJA* Bez alijasa

**GROUP BY** *kolona1, kolona2 , ...* Bez alijasa

**HAVING** *uslov na nivou grupe redova – može agregatna (grupna) funkcija* Bez alijasa

**ORDER BY** *kolona, funkcija ili izraz [ASC/DESC];* Bez alijasa

**LIMIT** *ofset (broj redova koji se prikazuje) – kod MySQL-a*

# ZADATAK 1

---

Prikazati brojeve odeljenja i srednju platu u svakom od njih. Iz proračuna isključiti analitičare i upravnike. Rezultate urediti po prosečnim primanjima u rastućem redosledu.

# ZADATAK 1

---

```
SELECT Id_odeljenja, AVG(plata) AS 'Prosečna plata'  
FROM radnik  
WHERE posao NOT IN ('analitičar','upravnik')  
GROUP BY Id_odeljenja  
ORDER BY AVG(plata)
```

## ZADATAK 2

---

Prikazati brojeve odeljenja i srednju platu u svakom od njih, samo za odeljenja u kojima je srednja plata veća od 2000.

## ZADATAK 2

---

```
SELECT Id_odeljenja, round(AVG(plata),2) AS  
'Prosecna plata'  
FROM radnik  
GROUP BY Id_odeljenja  
HAVING AVG(plata)>2000
```

## ZADATAK 3

---

Prikazati za svaki posao ukupnu platu radnika koji ga obavljaju, samo za poslove koje obavlja više od 2 radnika.

# ZADATAK 3

---

```
SELECT posao, SUM(plata) AS 'Ukupna plata'  
FROM radnik  
GROUP BY posao  
HAVING count(*)>2
```

# ZADATAK 4

---

Odrediti srednju godišnju platu unutar svakog odeljenja ne uzimajući u obzir plate direktora i upravnika.  
Ne prikazivati prosečnu godišnju platu neraspoređenih radnika.

# ZADATAK 4

---

```
SELECT Id_odeljenja, AVG(plata*12) AS 'Prosečna  
plata'  
FROM radnik  
WHERE posao NOT IN('direktor', 'upravnik')  
and Id_odeljenja IS NOT NULL  
GROUP BY Id_odeljenja
```

## ZADATAK 5

---

Za svako odeljenje prikazati broj radnika i ukupna primanja.

Ne prikazivati ukupna primanja neraspoređenih radnika.

## ZADATAK 5

---

```
SELECT Id_odeljenja, COUNT(*) AS 'Broj radnika',  
SUM(plata+IFNULL(premija,0)) AS 'Ukupna  
primanja'  
FROM RADNIK  
WHERE Id_odeljenja is NOT NULL  
GROUP BY Id_odeljenja
```

# Povezivanje podataka više tabela

---

Spajanje tabela predstavlja mogućnost istovremenog pretraživanja podataka iz više tabela i može se izvesti po jednoj od operacija poređenja (=, <, <=, >, >=, <>) nad uporedivim kolonama dve tabele.

Ova operacija učestvuje u izgradnji logičkog izraza u WHERE klauzuli, a sve tabele koje učestvuju u spajanju navode se u FROM klauzuli.

U slučaju istih imena kolona u dve tabele, njihova pojavljivanja moraju biti kvalifikovana navođenjem imena tabela iz kojih potiču.

# Povezivanje podataka više tabela

---

Spajanje tabela se koristi kada se informacije koje želimo da prikazemo (lista atributa u `SELECT` odredbi) nalaze u više tabela.

Pomoću spajanja se dobijaju informacije iz dve ili više tabela kao jedan skup rezultata (recordset, resultset).

Skup rezultata predstavlja “virtuelnu” tabelu koja postoji samo u memoriji računara (ali ne i na disku).

Kako se spajaju informacije iz više tabela u jednu, rezultat zavisi od toga kako zahtevamo da se spajaju podaci.

# Povezivanje podataka iz više tabela

Spajanjem (**JOIN**) se povezuje n-torke različitih tabela koristeći zajedničke (istoimene) attribute ili attribute definisane nad istim domenima.

U relacionoj teoriji postoje različite vrste spajanja:

- **Dekartov proizvod** – bezuslovno spajanje,
- **prirodno spajnje (ekvispajanje)** – spajanje po jednakosti
- **slobodno spajanje** – spajanje po proizvoljnom uslovu
- **relacijsko deljenje** - podupiti

# Povezivanje podataka iz više tabela

---

Spajanje se najčešće vrši preko zajedničkih atributa.

U slučaju da tabele u upitu nisu povezane ne postoji način da DBMS utvrdi koji su zapisi iz jedne tabele povezani sa zapisima iz druge tabele, pa kao odgovor vraća sve kombinacije zapisa.

Ovakav rezultat predstavlja **Dekartov** ili **Kartezijev proizvod** ("cross-product" ili "Cartesian product").

# Dekartov (kartezijanski) proizvod tabela

---

**Dekartov (kartezijanski) proizvod** tabela je bezuslovno spajanje dve tabele, kojim se dobijaju sve kombinacije svih redova jedne tabele sa redovima druge tabele. Zato ovakvo spajanje retko kad daje neku korisnu informaciju.

Ako bismo imali dve tabele sa po 10 zapisa, rezultat bi se sastojao iz 100 zapisa ( $10 \times 10$ ).

Upit se dugo izvršava i vraća rezultate bez ikakvog smisla.

# Prirodno spajanje - dve tabele

---

Za kombinovanje podataka iz dve tabele, potrebno je zadati uslov spajanja kojim se ograničava kombinovanje redova jedne tabele sa redovima druge tabele.

Kada se u upitu spajaju dve tabele, uslov ili grupa uslova pomoću kojih se povezuju dve tabele zove se **spojni uslov** što predstavlja vezu između tabela koja je definisana preko ključeva još prilikom definisanja strukture baze podataka.

# Prirodno spajanje - dve tabele

---

```
SELECT kolone_koje_nas_interesuju  
FROM table1, table2  
WHERE atribut_T1_FK= atribut_T2_PK;
```

Uslov spajanja, ili kombinacija uslova, zavisi od toga kakvu informaciju tražimo.

U **WHERE** klauzuli se navodi **neophodan uslov spajanja**:

**jednakost kolone spolnog ključa iz jedne tabele sa kolonom primarnog ključa druge tabele koju spolni ključ referencira.**

# Prirodno spajanje - dve tabele

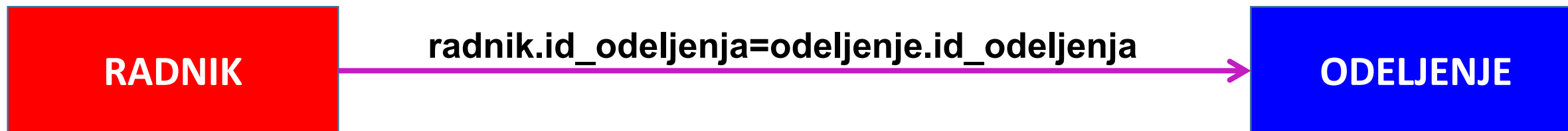
| Id_radnika | Ime        | Prezime  | Posao      | Kvalif | Rukovodilac | Dat_zap             | Premija | Plata | <b>Id_odeljenja</b> |
|------------|------------|----------|------------|--------|-------------|---------------------|---------|-------|---------------------|
| 5367       | Petar      | Vasić    | vozač      | KV     | 5780        | 1978-01-01 00:00:00 | 1900    | 1300  | 20                  |
| 5497       | Aleksandar | Marić    | električar | KV     | 5662        | 1990-02-17 00:00:00 | 800     | 1000  | 10                  |
| 5519       | Vanja      | Kondić   | prodavac   | VKV    | 5662        | 1991-11-07 00:00:00 | 1300    | 1200  | 10                  |
| 5652       | Jovan      | Perić    | električar | KV     | 5662        | 1980-05-31 00:00:00 | 500     | 1000  | 10                  |
| 5662       | Janko      | Mančić   | upravnik   | VSS    | 6789        | 1993-08-12 00:00:00 | NULL    | 2400  | 10                  |
| 5696       | Mirjana    | Dimić    | čistač     | KV     | 5662        | 1991-09-30 00:00:00 | 0       | 1000  | 10                  |
| 5780       | Božidar    | Ristić   | upravnik   | VSS    | 6789        | 1984-08-11 00:00:00 | NULL    | 2200  | 20                  |
| 5786       | Pavle      | Šotra    | upravnik   | VSS    | 6789        | 1983-05-22 00:00:00 | NULL    | 2800  | 30                  |
| 5842       | Miloš      | Marković | direktor   | VSS    | NULL        | 1981-12-15 00:00:00 | NULL    | 3000  | 40                  |

| <b>Id_odeljenja</b> | Ime_od          | Mesto        | Sef_odeljenja |
|---------------------|-----------------|--------------|---------------|
| 10                  | Komercijala     | Novi Beograd | 5662          |
| 20                  | Plan            | Dorćol       | 5780          |
| 30                  | Prodaja         | Stari Grad   | 5786          |
| 40                  | Direkcija       | Banovo Brdo  | 5842          |
| 50                  | Računski centar | Zemun        | NULL          |
| 60                  | Nabavka         | Rakovica     | NULL          |
| NULL                | NULL            | NULL         | NULL          |

# Prirodno spajanje - dve tabele

Kada se spaja  $n$  tabela, u većini slučajeva, biće potrebna po jedna veza između svakog para tabela, što znači da će postojati  $n-1$  spojnih uslova.

Spoj definisan u ovom primeru:



# Prirodno spajanje - dve tabele

---

U FROM klauzuli se navode tabele koje učestvuju u spoju.

U WHERE klauzuli se navodi spojni uslov u obliku:

**naziv\_tabele1.spoljni\_kljuc = naziv\_tabela2.primarni\_kljuc**

# Prirodno spajanje - dve tabele

```
SELECT ime, prezime, ime_od  
FROM radnik, odeljenje  
WHERE radnik.id_odeljenja = odeljenje.id_odeljenja
```

|   | ime        | prezime  | ime_od      |
|---|------------|----------|-------------|
| ► | Miloš      | Marković | Direkcija   |
|   | Svetlana   | Grubač   | Direkcija   |
|   | Janko      | Simić    | Direkcija   |
|   | Aleksandar | Marić    | Komercijala |
|   | Vanja      | Kondić   | Komercijala |
|   | Jovan      | Perić    | Komercijala |
|   | Janko      | Mančić   | Komercijala |
|   | Mirjana    | Dimić    | Komercijala |
|   | Tomislav   | Bogovac  | Komercijala |
|   | Petar      | Vasić    | Plan        |
|   | Božidar    | Ristić   | Plan        |
|   | Slobodan   | Petrović | Plan        |
|   | Mitar      | Vuković  | Plan        |
|   | Ivan       | Buha     | Plan        |
|   | Pavle      | Šotra    | Prodaja     |
|   | Andrija    | Ristić   | Prodaja     |
|   | Jovan      | Perić    | Prodaja     |

# Prirodno spajanje dve tabele bez uslova

```
SELECT ime, prezime, ime_od  
FROM radnik, odeljenje
```

Kao rezultat dobije se Dekartov proizvod.

Upit je vratio 114 redova (19x6).

**OBAVEZNO POVEZATI TABELE!!!**

|   | ime        | prezime | ime_od          |
|---|------------|---------|-----------------|
| ► | Petar      | Vasić   | Direkcija       |
|   | Petar      | Vasić   | Komercijala     |
|   | Petar      | Vasić   | Nabavka         |
|   | Petar      | Vasić   | Plan            |
|   | Petar      | Vasić   | Prodaja         |
|   | Petar      | Vasić   | Računski centar |
|   | Aleksandar | Marić   | Direkcija       |
|   | Aleksandar | Marić   | Komercijala     |
|   | Aleksandar | Marić   | Nabavka         |
|   | Aleksandar | Marić   | Plan            |
|   | Aleksandar | Marić   | Prodaja         |
|   | Aleksandar | Marić   | Računski centar |
|   | Vania      | Kondić  | Direkcija       |

# Kvalifikovanje kolona

---

Kada se pojavi isto ime kolone u obe tabele, onda ispred imena kolone mora da se doda i ime tabele kojoj pripada.

To se zove "**qualifying columns**" (**kvalifikovanje kolona**).

Kombinacijom imena tabele i imena kolone se eliminišu dvosmisleni nazivi kada dve tabele sadrže kolone sa istim imenom.

**radnik.id\_odeljenja = odeljenje.id\_odeljenja**

# Upotreba alijasa tabela

---

Prilikom rada sa više tabela potrebno je, osim naziva atributa, navesti i naziv tabele u kojoj se taj atribut nalazi.

- ako se u različitim tabelama nalaze atributi istog imena, a bar jedan se koristi u upitu.

Lakše je pridružiti tabeli neko kraće ime.

Umesto navođenja imena tabele u upitu poziva se dodeljeni alijas (sinonim).

# Upotreba alijasa tabela

```
SELECT R.ime, O.ime_od, O.mesto  
FROM ODELJENJE O , RADNIK R  
WHERE R.id_odeljenja=O.id_odeljenja
```

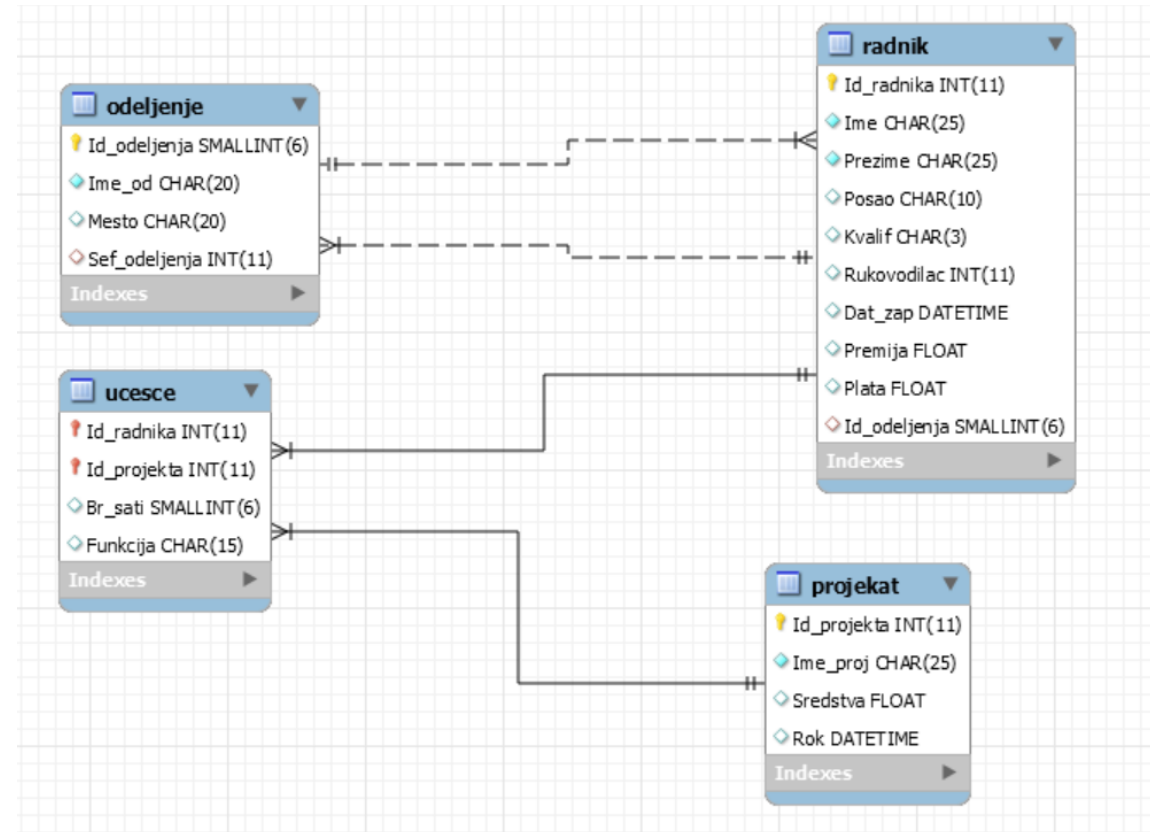
Kod upotrebe alijasa potrebno je održati doslednost.

U FROM odredbi navedi se puno ime tabele i dodeli skraćenica za tabelu koja se koristi u tom upitu.

Ako su uvedeni alijasi tabela, oni se moraju koristiti u svim kvalifikovanim nazivima kolona u klauzulama SELECT, ON, WHERE i dr.

# Upotreba spojeva u upitima koji obuhvataju više tabela

Kao što se vidi sa ER dijagrama baze preduzeće, podaci o učešću radnika na nekom projektu se nalaze u tabeli *ucesce*.



# Upotreba spojeva u upitima koji obuhvataju više tabela

```
SELECT r.prezime, r.ime, u.id_projekta  
FROM radnik r, ucesce u  
WHERE r.id_radnika=u.id_radnika
```

Izvršavanjem ovog upita dobijaju rezultati koji nam samo daju informaciju o šifri projekta na kome je radnik angažovan.

Podatak o nazivu projekta se nalazi u tabeli *projekta*.

|   | prezime  | ime        | id_projekta |
|---|----------|------------|-------------|
| ► | Marić    | Aleksandar | 400         |
|   | Perić    | Jovan      | 100         |
|   | Perić    | Jovan      | 300         |
|   | Mančić   | Janko      | 300         |
|   | Dimić    | Mirjana    | 200         |
|   | Dimić    | Mirjana    | 300         |
|   | Ristić   | Božidar    | 200         |
|   | Šotra    | Pavle      | 100         |
|   | Marković | Miloš      | 100         |
|   | Grubač   | Svetlana   | 200         |
|   | Ristić   | Andrija    | 200         |
|   | Petrović | Slobodan   | 100         |
|   | Vuković  | Mitar      | 100         |

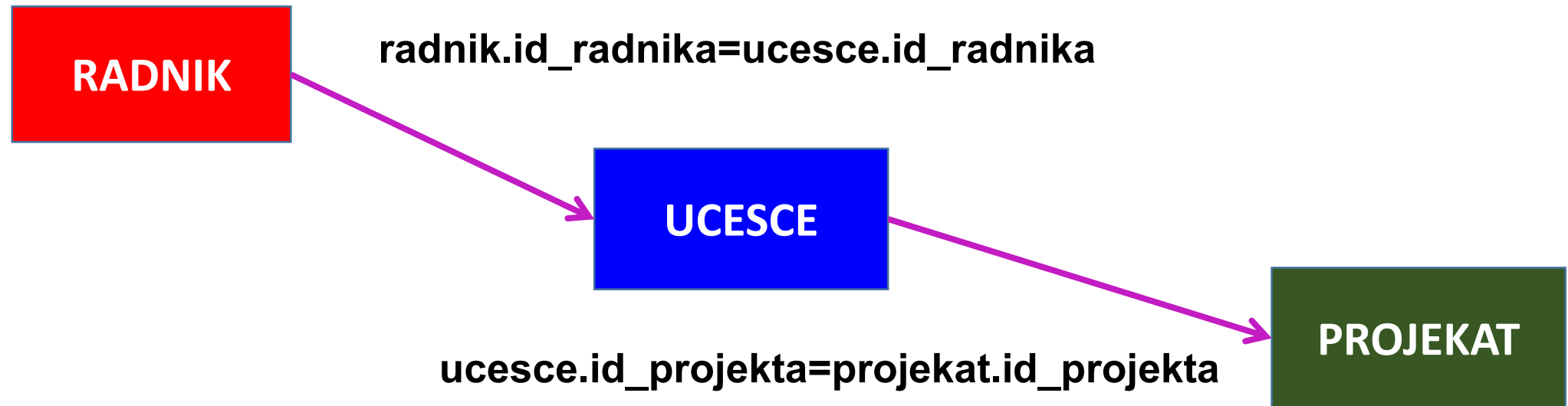
# Upotreba spojeva u upitima koji obuhvataju više tabela

U slučaju generisanja izveštaja o nazivima projekata na kojima su radnici angažovani neophodno je u spoju povezati tri tabele i to:

- **radnik**: da bi se prikazali podaci o **imenu** i **prezimenu**
- **ucesce**: da bi izdvojili redovi sa podacima o **šifri radnika** i **šifri projekta** na kome je angažovan
- **projekat**: da bi se izdvojili podaci o **nazivu projekata** na kome su radnici anagažovani.

# Upotreba spojeva u upitima koji obuhvataju više tabela

U FROM klauzuli se navode tri tabele (broj tabela u FROM klauzuli) to znači da će broj spojeva između tabela dva (broj tabela iz FROM klauzule -1)



# Upotreba spojeva u upitima koji obuhvataju više tabela

```
SELECT prezime, ime, ime_proj  
FROM radnik, ucesce, projekat  
WHERE radnik.id_radnika=ucesce.id_radnika AND  
ucesce.id_projekta=projekat.id_projekta
```

|   | prezime | ime        | ime_proj   |
|---|---------|------------|------------|
| ► | Dimić   | Mirjana    | izvoz      |
|   | Ristić  | Božidar    | izvoz      |
|   | Grubač  | Svetlana   | izvoz      |
|   | Ristić  | Andrija    | izvoz      |
|   | Vuković | Mitar      | izvoz      |
|   | Nastić  | Marko      | izvoz      |
|   | Simić   | Janko      | izvoz      |
|   | Perić   | Jovan      | plasman    |
|   | Mančić  | Janko      | plasman    |
|   | Dimić   | Mirjana    | plasman    |
|   | Vuković | Mitar      | plasman    |
|   | Perić   | Jovan      | plasman    |
|   | Nastić  | Marko      | plasman    |
|   | Buha    | Ivan       | plasman    |
|   | Marić   | Aleksandar | projekt... |
|   | Perić   | Jovan      | izvoz      |

# Upotreba spojeva za spajanje tabele sa samom sobom

---

Kao što se jedna tabela može spojiti sa drugom, tako se može spojiti i sa samom sobom.

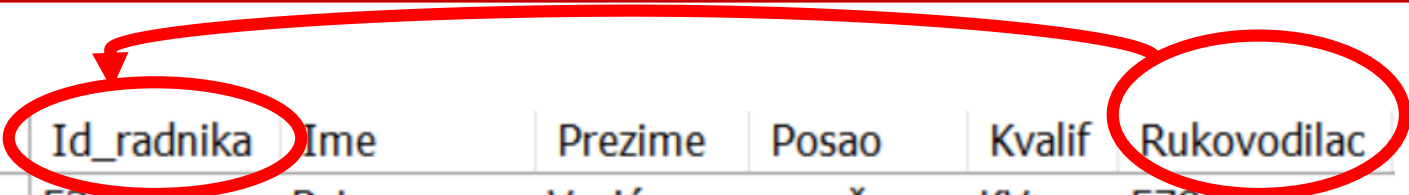
Ovakav način spajanja se koristi kada se traže veze između redova u istoj tabeli.

Poznat pod nazivom **SELF JOIN**.

# Upotreba spojeva za spajanje tabele sa samom sobom

U tabeli radnik definisan je atribut Rukovodilac koji po domenu vrednosti odgovara atributu Id\_radnika i označava šifru zaposlenog koji je rukovodilac nekom radniku.

Šifra rukovodioca je identifikacioni broj zaposlenog radnika.



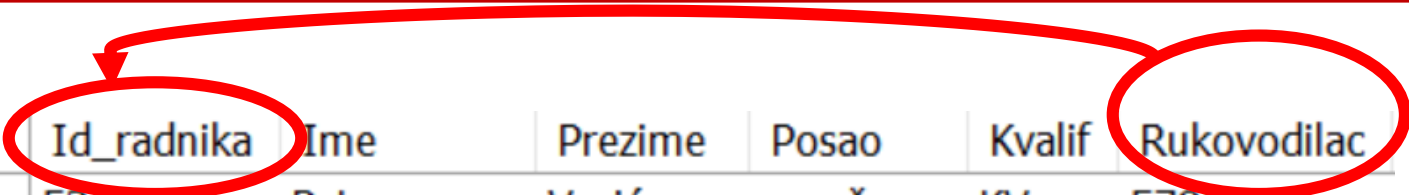
|   | Id_radnika | Ime        | Prezime  | Posao      | Kvalif | Rukovodilac |
|---|------------|------------|----------|------------|--------|-------------|
| ▶ | 5367       | Petar      | Vasić    | vozač      | KV     | 5780        |
|   | 5497       | Aleksandar | Marić    | električar | KV     | 5662        |
|   | 5519       | Vanja      | Kondić   | prodavac   | VKV    | 5662        |
|   | 5652       | Jovan      | Perić    | električar | KV     | 5662        |
|   | 5662       | Janko      | Mančić   | upravnik   | VSS    | 6789        |
|   | 5696       | Mirjana    | Dimić    | čistač     | KV     | 5662        |
|   | 5780       | Božidar    | Ristić   | upravnik   | VSS    | 6789        |
|   | 5786       | Pavle      | Šotra    | upravnik   | VSS    | 6789        |
|   | 5842       | Miloš      | Marković | direktor   | VSS    | NULL        |
|   | 5867       | Svetlana   | Grubač   | savetnik   | VSS    | 5842        |
|   | 5874       | Tomislav   | Bogovac  | električar | KV     | 5662        |
|   | 5898       | Andrija    | Ristić   | nabavljač  | KV     | 5786        |
|   | 5900       | Slobodan   | Petrović | vozač      | KV     | 5780        |
|   | 5932       | Mitar      | Vuković  | savetnik   | VSS    | 5842        |
|   | 5953       | Jovan      | Perić    | nabavljač  | KV     | 5786        |
|   | 6234       | Marko      | Nastić   | analitičar | VSS    | 5867        |

# Upotreba spojeva za spajanje tabele sa samom sobom

Za svakog radnika prikazati ime i prezime njegovog rukovodioca.

Svi podaci su u jednoj tabeli!

Zamisliti kao da postoje dve tabele **Tabela A** i **Tabela B** koje sadrže iste podatke o radnicima.



|   | Id_radnika | Ime        | Prezime  | Posao      | Kvalif | Rukovodilac |
|---|------------|------------|----------|------------|--------|-------------|
| ▶ | 5367       | Petar      | Vasić    | vozač      | KV     | 5780        |
|   | 5497       | Aleksandar | Marić    | električar | KV     | 5662        |
|   | 5519       | Vanja      | Kondić   | prodavac   | VKV    | 5662        |
|   | 5652       | Jovan      | Perić    | električar | KV     | 5662        |
|   | 5662       | Janko      | Mančić   | upravnik   | VSS    | 6789        |
|   | 5696       | Mirjana    | Dimić    | čistač     | KV     | 5662        |
|   | 5780       | Božidar    | Ristić   | upravnik   | VSS    | 6789        |
|   | 5786       | Pavle      | Šotra    | upravnik   | VSS    | 6789        |
|   | 5842       | Miloš      | Marković | direktor   | VSS    | NULL        |
|   | 5867       | Svetlana   | Grubač   | savetnik   | VSS    | 5842        |
|   | 5874       | Tomislav   | Bogovac  | električar | KV     | 5662        |
|   | 5898       | Andrija    | Ristić   | nabavljač  | KV     | 5786        |
|   | 5900       | Slobodan   | Petrović | vozač      | KV     | 5780        |
|   | 5932       | Mitar      | Vuković  | savetnik   | VSS    | 5842        |
|   | 5953       | Jovan      | Perić    | nabavljač  | KV     | 5786        |
|   | 6234       | Marko      | Nastić   | analitičar | VSS    | 5867        |

# Upotreba spojeva za spajanje tabele sa samom sobom

|   | Id_radnika | Ime        | Prezime  | Posao      | Kvalif | Rukovodilac         |
|---|------------|------------|----------|------------|--------|---------------------|
| ► | 5367       | Petar      | Vasić    | vozač      | KV     | 5780                |
|   | 5497       | Aleksandar | Marić    | električar | KV     | 5662                |
|   | 5519       | Vanja      | Kondić   | prodavac   | VKV    | 5662                |
|   | 5652       | Jovan      | Perić    | električar | KV     | 5662                |
|   | 5662       | Janko      | Mančić   | upravnik   | VSS    | 6789                |
|   | 5696       | Mirjana    | Dimić    | čistač     | KV     | 5662                |
|   | 5780       | Božidar    | Ristić   | upravnik   | VSS    | 6789                |
|   | 5786       | Pavle      | Šotra    | upravnik   | VSS    | 6789                |
|   | 5842       | Miloš      | Marković | direktor   | VSS    | <small>HULL</small> |
|   | 5867       | Svetlana   | Grubač   | savetnik   | VSS    | 5842                |
|   | 5874       | Tomislav   | Bogovac  | električar | KV     | 5662                |
|   | 5898       | Andrija    | Ristić   | nabavljač  | KV     | 5786                |
|   | 5900       | Slobodan   | Petrović | vozač      | KV     | 5780                |
|   | 5932       | Mitar      | Vuković  | savetnik   | VSS    | 5842                |
|   | 5953       | Jovan      | Perić    | nabavljač  | KV     | 5786                |
|   | 6234       | Marko      | Nastić   | analitičar | VSS    | 5867                |

U **tabeli A** potražimo podatak o **imenu, prezimenu radnika i šifri njegovog rukovodioca**.

Na osnovu šifre rukovodica, iz **tabele B** pronađemo **ime i prezime rukovodioca**.

# Upotreba spojeva za spajanje tabele sa samom sobom

---

```
SELECT A.prezime as 'Prezime radnik', A.ime as 'Ime radnika',  
B.prezime as 'Prezime rukovodioca', B.ime as 'Ime rukovodioca'  
  
FROM radnik A, radnik B  
  
WHERE A.rukovodilac=B.id_radnika
```

# Upotreba spojeva za spajanje tabele sa samom sobom

| Prezime radnik | Ime radnika | Prezime rukovodioca | Ime rukovodioca |
|----------------|-------------|---------------------|-----------------|
| Vasić          | Petar       | Ristić              | Božidar         |
| Marić          | Aleksandar  | Mančić              | Janko           |
| Kondić         | Vanja       | Mančić              | Janko           |
| Perić          | Jovan       | Mančić              | Janko           |
| Mančić         | Janko       | Simić               | Janko           |
| Dimić          | Mirjana     | Mančić              | Janko           |
| Ristić         | Božidar     | Simić               | Janko           |
| Šotra          | Pavle       | Simić               | Janko           |
| Grubač         | Svetlana    | Marković            | Miloš           |
| Bogovac        | Tomislav    | Mančić              | Janko           |
| Ristić         | Andrija     | Šotra               | Pavle           |
| Petrović       | Slobodan    | Ristić              | Božidar         |
| Vuković        | Mitar       | Marković            | Miloš           |
| Perić          | Jovan       | Šotra               | Pavle           |
| Nastić         | Marko       | Grubač              | Svetlana        |
| Simić          | Janko       | Marković            | Miloš           |

# ZADACI

1. Prikazati imena projekata i broj radnika na njima za sve projekte na kojima radi više od 3 radnika.
2. Prikazati imena odeljenja i ime i prezime šefova odeljenja.
3. Prikazati ime radnika i mesto odeljenja u kome radi za radnike čija je plata između 1500 i 2500 (uključujući i te vrednosti) i čije ime ne sadrži slove e. Rezultate sortirati po plati u opadajućem redosledu, a zatim po imenu u rastućem.
4. Prikazati ime i primanja radnika i njihovih rukovodilaca za radnike čija su primanja veća od primanja njihovih rukovodioca.

