

BAZE PODATAKA

Predavanje broj: 04

Nastavne teme:

- ✓ SELECT ..FROM, WHERE, ORDER BY, GROUP BY
- ✓ OPERATORI, OPERACIJE;
- ✓ UGRAĐENE FUNKCIJE
- ✓ UPITI NAD JEDNOM TABELOM

Naredbe za rukovanje sa podacima - DML

NAREDBE ZA RUKOVANJE PODACIMA omogućavaju izveštavanje iz baze podataka (izdvajanje postojećih i izračunavanje novih informacija) i ažuriranje baze podataka u širem smislu.

Izveštavanje iz baze podataka:

- ❑ **SELECT** – pristup podacima, prikaz pronađenih ili izračunatih sadržaja iz baze podataka.

Ažuriranje baze podataka:

- ❑ **INSERT** – unošenje podataka, dodavanje novih redova u tabelu,
- ❑ **DELETE** – brisanje podataka, izbacivanje redova iz tabele,
- ❑ **UPDATE** – ažuriranje, izmena vrednosti podataka u koloni.

SELECT – kompletna sintaksa

```
SELECT ([ALL | DISTINCT] <lista_atributa1> ) | *  
FROM <lista_tabela>  
[ WHERE <lista_uslova1> ]  
[ GROUP BY <lista_atributa2> ]  
[ HAVING <lista_uslova2> ]  
[ ORDER BY <lista_atributa3> ]
```

SELECT koji atributi
FROM iz kojih tabela
WHERE uslov nad zapisima date tabele
GROUP BY grupisanje po atributima
HAVING uslovi za kreirane grupe
ORDER BY sortiranje po atributima

Termini

❑ Ključna reč

Ključna reč ukazuje na poseban SQL-iskaz. Ključne reči su: SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY

❑ Klauzula

Klauzula je deo SQL-rečenice. Na primer, klauzule su:

- SELECT id, plata
- WHERE plata=3000
- FROM radnici
- ORDER BY plata

❑ Rečenica (naredba)

Rečenica je jedna klauzula ili kombinacija dve ili više klauzula, kojom se izvršava neka operacija nad bazom podataka. Završava se znakom ;

- SELECT ime FROM radnici;
- SELECT * FROM radnici;
- SELECT id_radnik, ime, plata FROM radnici ORDER BY plata;

SELECT

U naredbi **SELECT** se:

- ❑ **Navode** atributi ($\langle \text{lista_atributa}_1 \rangle$) čije vrednosti želimo prikazati – odgovara operatoru **projekcije**,
- ❑ **Izdvajaju** tabele ($\langle \text{lista_tabela} \rangle$) u kojima se nalaze vrednosti tih atributa – odgovara operatoru **spajanja**,
- ❑ **Definišu uslovi** ($\langle \text{lista_uslova}_1 \rangle$, $\langle \text{lista_uslova}_2 \rangle$), koje podaci treba da zadovoljavaju – odgovara operatoru **selekcije** (**restrikcije**),
- ❑ **Definiše grupisanje** ($\langle \text{lista_atributa}_2 \rangle$),
- ❑ **Definše uređenje** ($\langle \text{lista_atributa}_3 \rangle$).

Osnovna anatomija SELECT rečenice

U najjednostavnijem obliku, **SELECT** rečenica mora da sadrži:

- **SELECT** klauzulu, koja specificira koja kolona se prikazuje
- **FROM** klauzulu, koja specificira tabelu kojoj pripada kolona navedena u **SELECT** klauzuli.

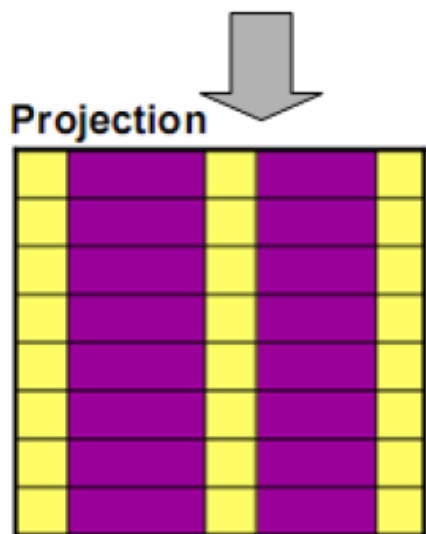
Osnovna anatomija SELECT rečenice

SELECT-rečenicom se izvršavaju operacije:

PROJEKCIJA je operacija koja iz skupa atributa izdvaja samo one koji su od interesa. Omogućava izbor kolona iz tabele!

→ klauzula

SELECT



SELEKCIJA je operacija koja izdvaja instance koje zadovoljavaju specificiran uslov. Omogućava izbor redova iz tabele (restrikcija).

→ klauzula

WHERE



Osnovna anatomija SELECT rečenice

Klauzula **SELECT** nam omogućava:

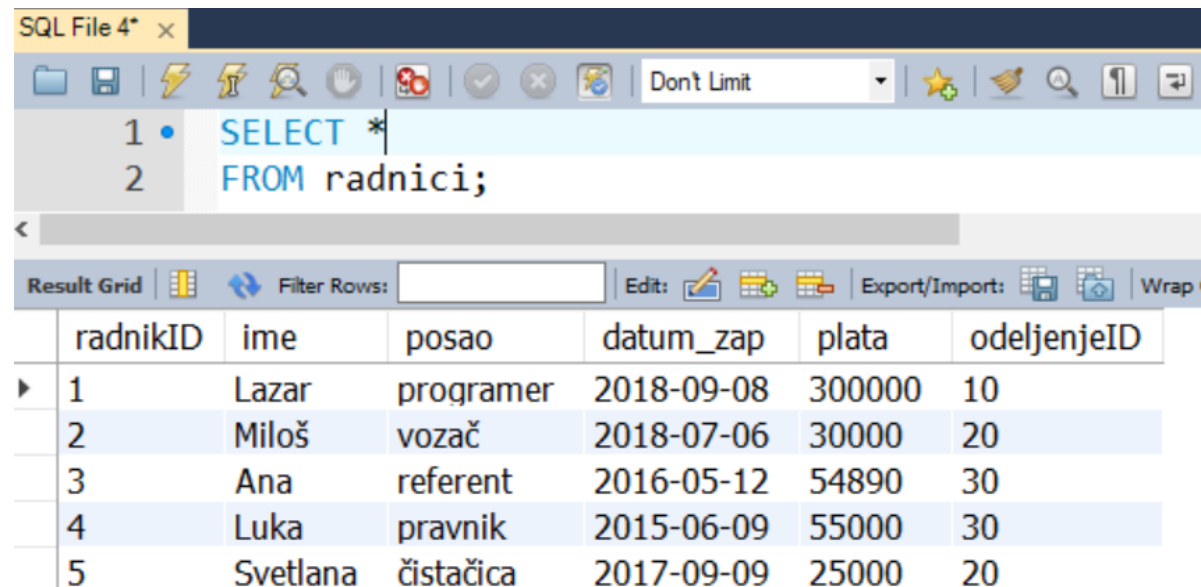
- ☐ Da izaberemo attribute čije vrednosti želimo dobiti
- ☐ Da definišemo redosled prikazivanja atributa u odgovoru (ovaj redosled se ne mora poklapati sa redosledom atributa u definiciji tabele)
- ☐ Da sprečimo pojavu identičnih instanci:
 - Opcija **DISTINCT** eliminiše identične vrednosti po kolonama iz rezultata.

SELECT – prikazivanje svih kolona tabele

Prikazivanje svih kolona podataka u tabeli može se izvršiti korišćenjem ključne reči SELECT sa zvezdicom(*).

Prikazati sve podatke o radnicima.

```
SELECT *  
FROM RADNICI
```



The screenshot shows a SQL IDE window titled 'SQL File 4*'. The query editor contains the following SQL statement:

```
1 • SELECT *  
2 FROM radnici;
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are shown in a table with the following columns: radnikID, ime, posao, datum_zap, plata, and odeljenjeID. The table contains 5 rows of data.

	radnikID	ime	posao	datum_zap	plata	odeljenjeID
▶	1	Lazar	programer	2018-09-08	300000	10
	2	Miloš	vozač	2018-07-06	30000	20
	3	Ana	referent	2016-05-12	54890	30
	4	Luka	pravnik	2015-06-09	55000	30
	5	Svetlana	čistačica	2017-09-09	25000	20

SELECT – navođenje kolona za prikaz

Prikazati ime i platu radnika

- Za dodavanje kolone plata u već kreiranu tabelu koristili smo klauzulu ALTER TABLE

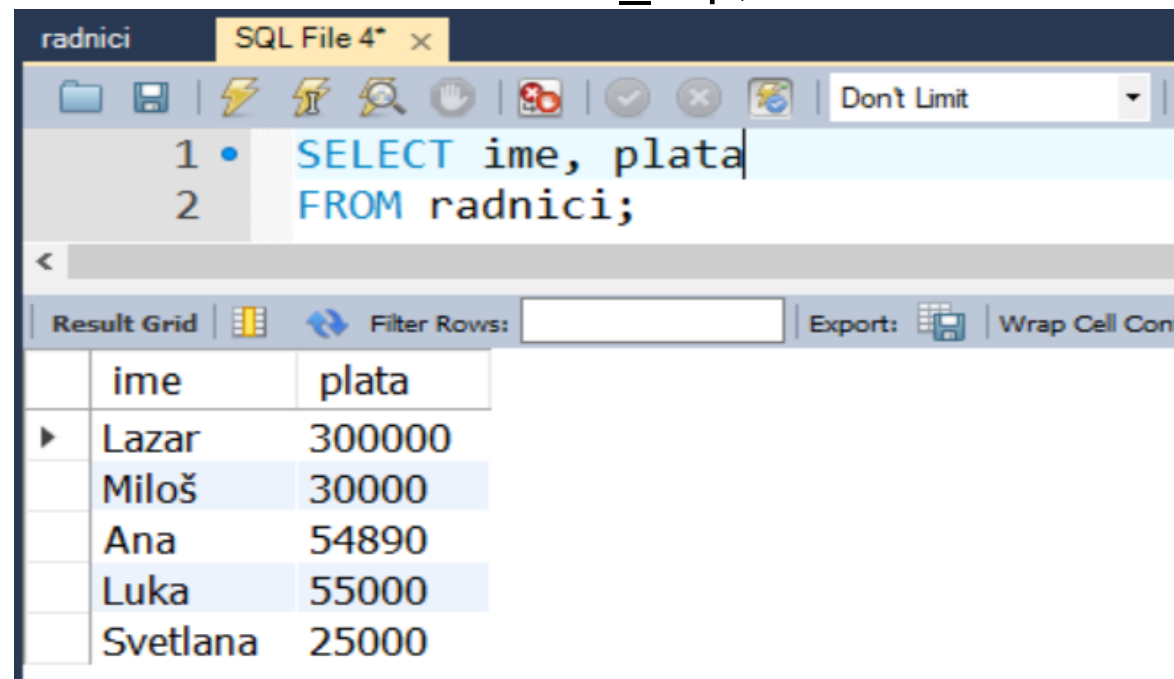
- ALTER TABLE radnici

ADD COLUMN plata FLOAT NOT NULL AFTER datum_zap;

- Upit za prikaz imena i plate radnika

SELECT ime, plata

FROM radnici



The screenshot shows a SQL IDE window titled 'radnici' and 'SQL File 4*'. The query editor contains the following SQL statement:

```
1 • SELECT ime, plata
2 FROM radnici;
```

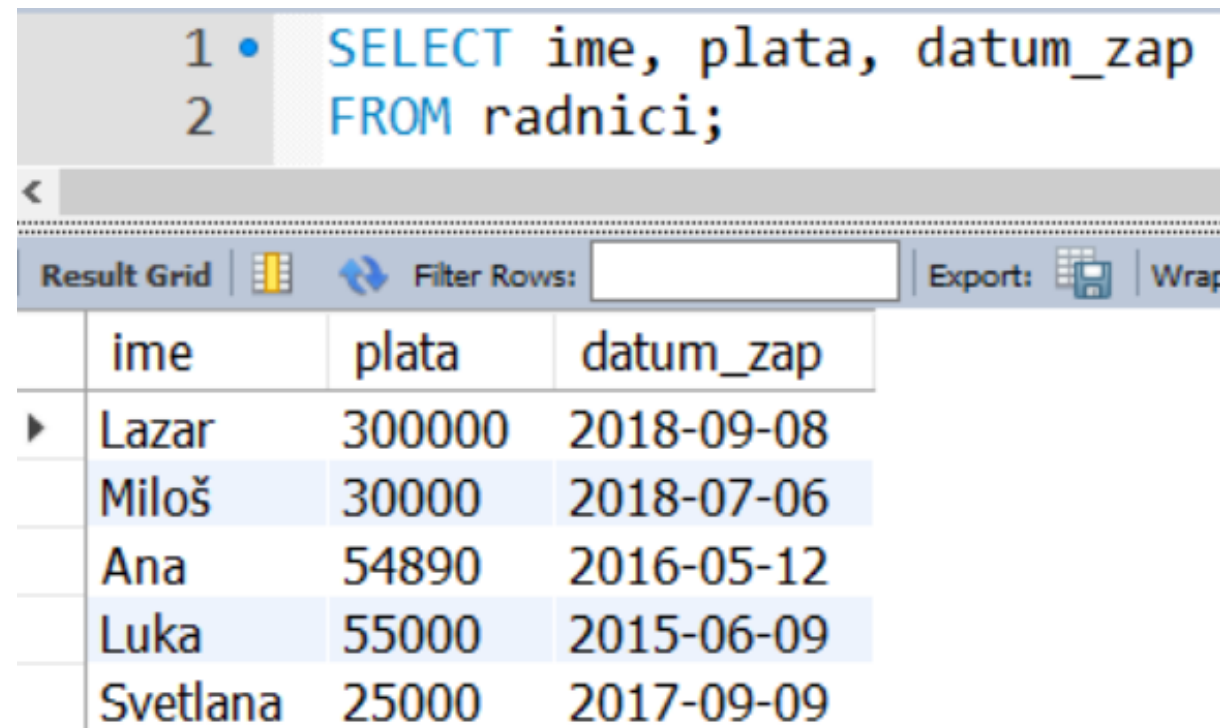
Below the query editor, the 'Result Grid' tab is active, displaying the results of the query in a table format. The table has two columns: 'ime' and 'plata'. The results are as follows:

ime	plata
Lazar	300000
Miloš	30000
Ana	54890
Luka	55000
Svetlana	25000

SELECT – navođenje kolona za prikaz

Prikazati ime, platu i datum zaposlenja radnika.

```
SELECT ime, plata, datum_zap  
FROM radnici;
```



The screenshot shows a database query interface. At the top, a text area contains the SQL query: `SELECT ime, plata, datum_zap FROM radnici;`. Below the text area is a horizontal scrollbar. Underneath the scrollbar is a toolbar with buttons for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap'. Below the toolbar is a table displaying the results of the query. The table has four columns: 'ime', 'plata', and 'datum_zap'. The first row is highlighted with a blue background. The data in the table is as follows:

	ime	plata	datum_zap
▶	Lazar	300000	2018-09-08
	Miloš	30000	2018-07-06
	Ana	54890	2016-05-12
	Luka	55000	2015-06-09
	Svetlana	25000	2017-09-09

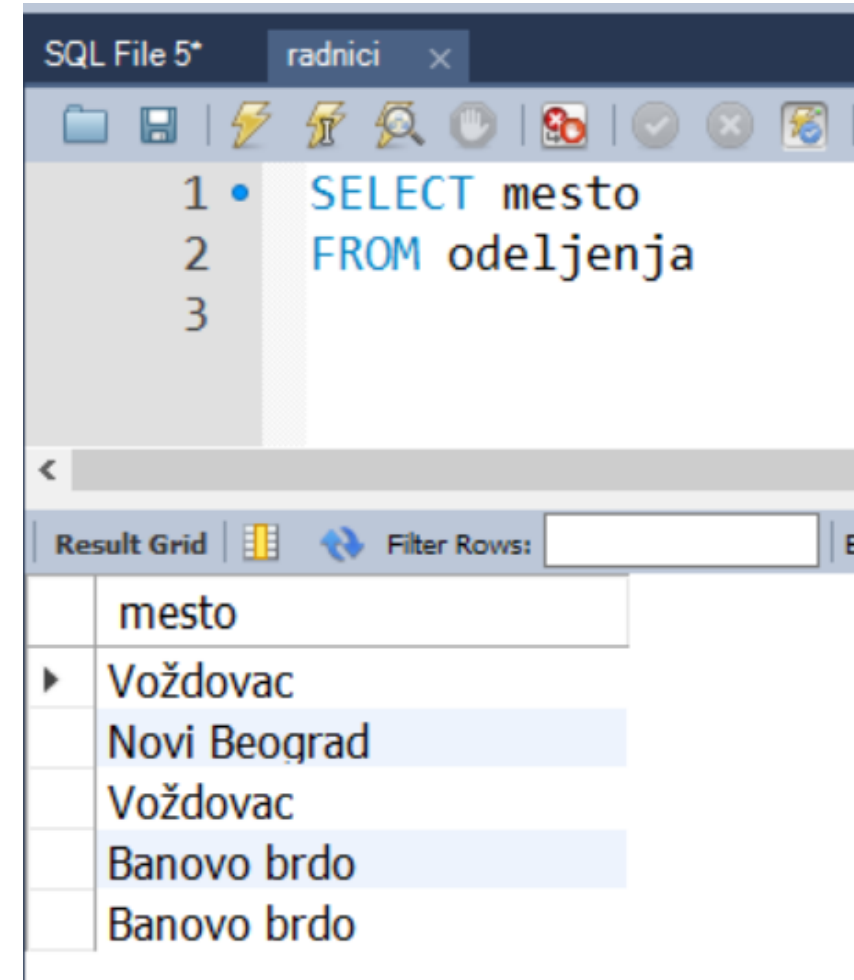
SELECT i ključna reč DISTINCT

SELECT rečenica koja se koristi za projekciju određenih kolona iz neke tabele, prikazuje sve vrednosti iz tih kolona, pri čemu može da se desi da se neke vrednosti ponavljaju

SELECT mesto **FROM** odeljenja;

Ova rečenica daje informaciju o mestima na kojima se nalaze odeljenja. Postoje 2 odeljenja koja se nalaze na Voždovcu i dva odeljenja na Novom Beogradu.

Zbog toga ovakav upit vraća redove koji se ponavljaju.



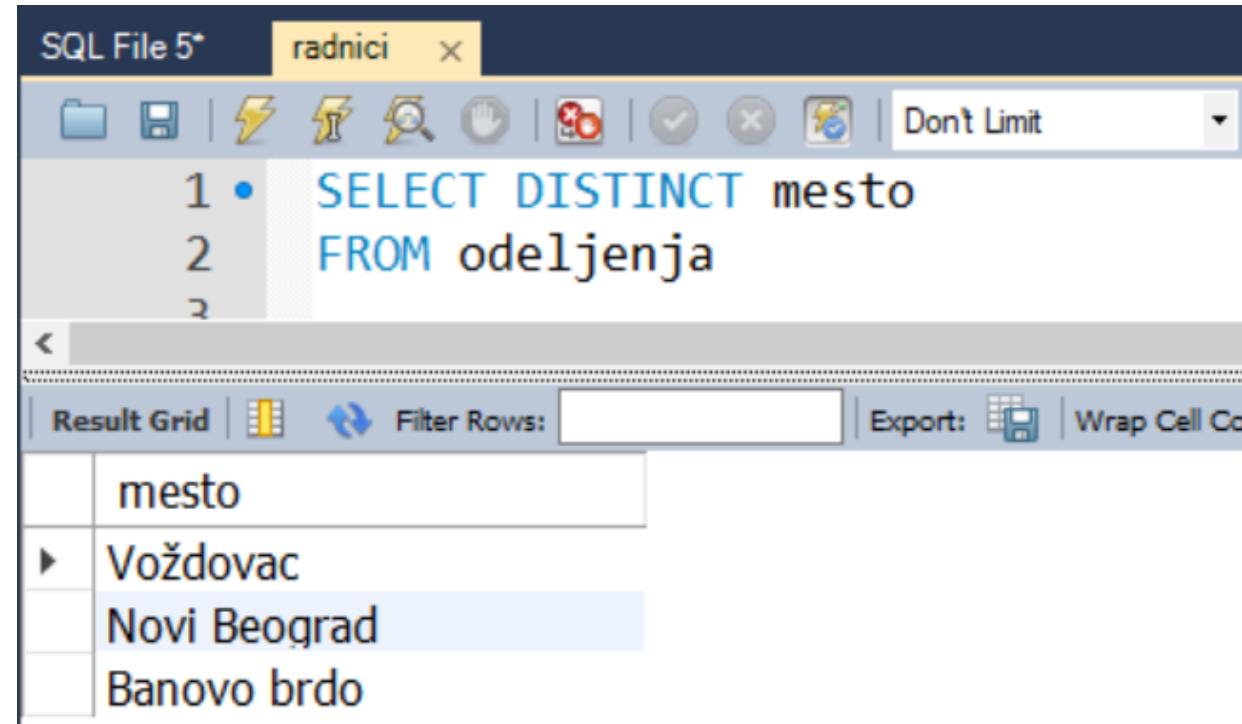
SELECT i ključna reč DISTINCT

Prikazati bez ponavljanja lokacije odeljenja

Ako je potrebno prikazati samo različite vredosti, a ne i koliko puta se pojavljuju, koristiće se ključna reč DISTINCT.

```
SELECT DISTINCT mesto  
FROM odeljenja
```

Sada su prikazana dva reda manje. Svi prikazani redovi imaju različite vrednosti.



SELECT – selekcija - restriktivan izbor redova

- ❑ **SELEKCIJA** je operacija koja izdvaja instance koje zadovoljavaju specificiran uslov.
- ❑ U naredbi **SELECT** uslov se zadaje klauzulom **WHERE**.
- ❑ U uslovu se kao operandi javljaju atributi ili ugnježdena naredba **SELECT** i **OPERATORI POREĐENJA**.
- ❑ Neki operatori porede datu kolonu ili izraz sa jednom vrednošću (**jednoredni operatori**), a drugi porede sa skupom ili intervalom vrednosti (**višeredni operatori**).

SELECT – selekcija – operatori poređenja

Operator	Opis operatora	Primer upotrebe
=, <, >, <>, <=, >=	Standardni relacijski operatori	ime='Petar' plata> 2000
AND, OR, NOT	Standardni logički operatori	ime='Petar' AND (OdeljenjeID =20 OR OdeljenjeID=30)
LIKE	Omogućava pronalaženje traženog teksta u nekom podatku znakovnog tipa.	ime LIKE 'M%'; ime LIKE '%M' ime LIKE '%M%'; Ime LIKE ' _ '
BETWEEN ... AND	Ispituje da li se vrednost datog izraza nalazi između dve zadate vrednosti uključujući i zadate vrednosti	Plata BETWEEN 1000 AND 5000
IN	Ispituje da li se vrednost datog izraza nalazi u specificiranoj listi	mesto IN ('Beograd', 'Zemun')
ALL	Ispituje da li se sve vrednost datog izraza nalazi u specificiranoj listi	radnikID = ALL (SELECT radnikID FROM ucesce WHERE finasije > 100000)
ANY	Ispituje da li se bar jedna vrednost datog izraza nalazi u specificiranoj listi	radnikID = ANY (SELECT radnikID FROM ucesce WHERE finasije > 100000)

WHERE klauzula

Kada se zahtevaju podaci iz baze podataka, možda nisu potrebni svi redovi iz tabele pa je neophodno ograničiti redove podataka koji će biti prikazani. Ovo se postiže WHERE klauzulom. WHERE klauzula sadrži uslov kojim mora da se ispuni i navodi se odmah posle FROM klauzule u SQL rečenici.

Alijasi i agregatne funkcije se ne mogu upotrebljavati u WHERE klauzuli!

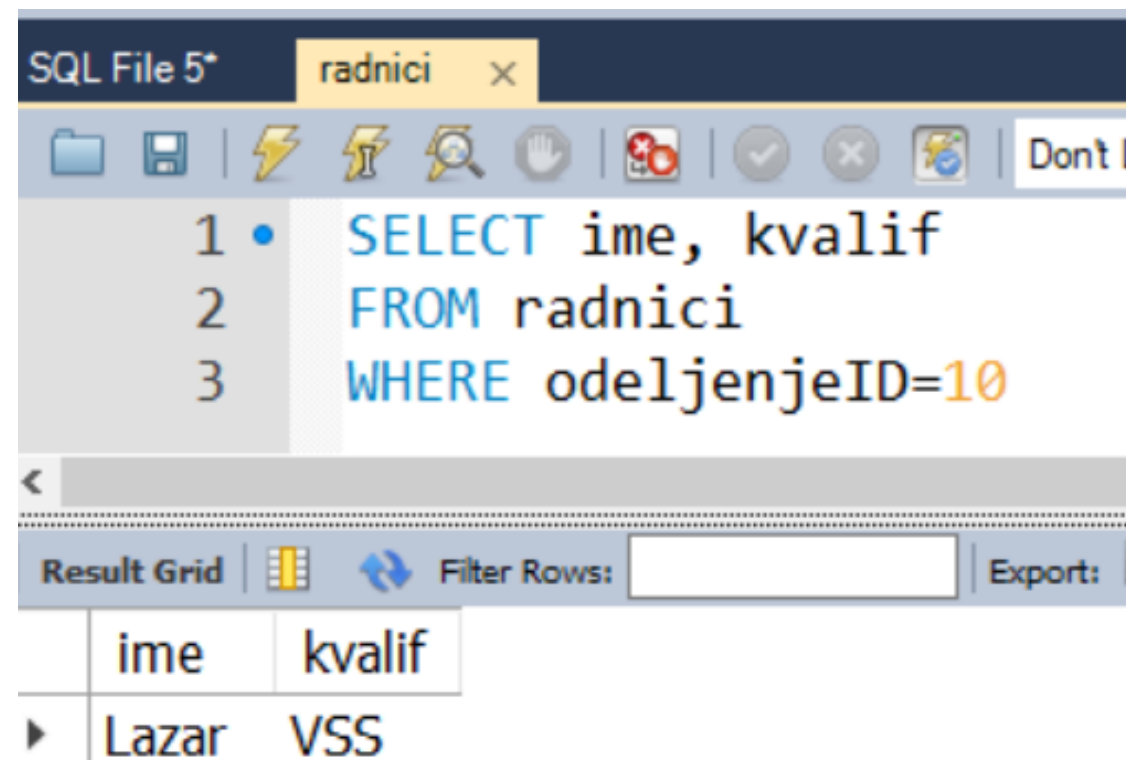
Sintaksa WHERE klauzule:

WHERE ime_kolone **uslov_poređenja** vrednost **||** lista_vrednosti

WHERE klauzula

Prikazati imena i kvalifikacije svih zaposlenih u odeljenju 10.

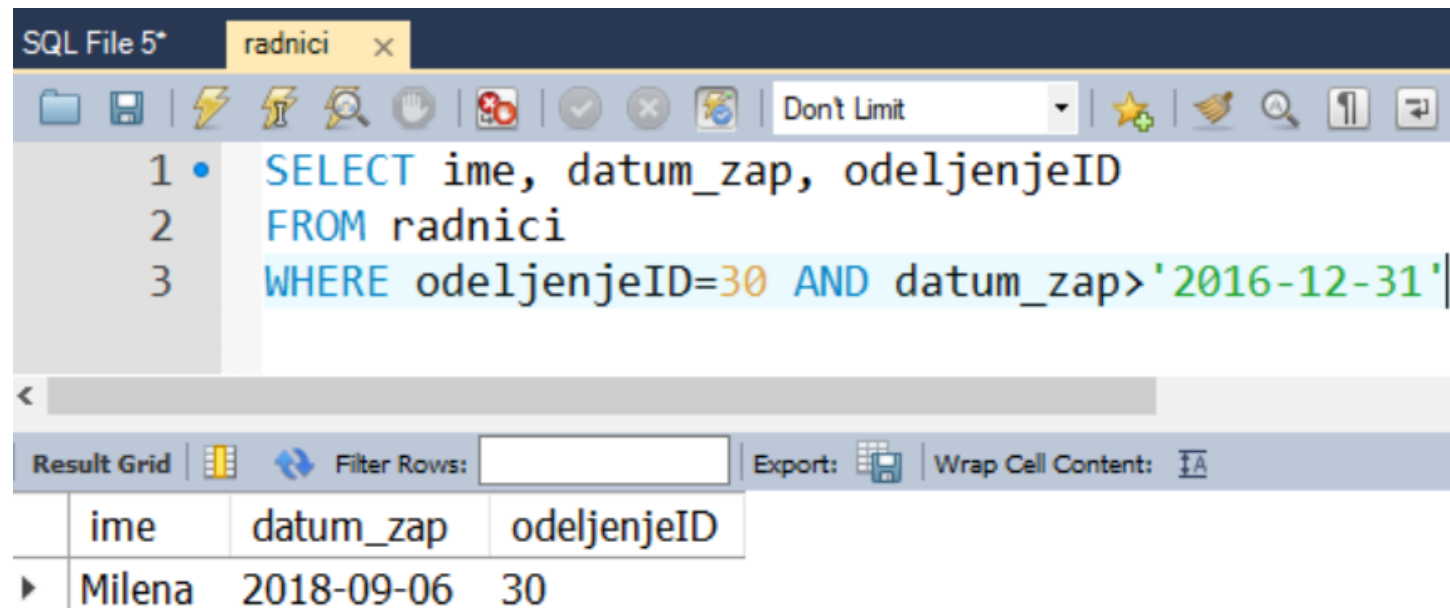
```
SELECT ime, kvalif  
FROM radnici  
WHERE odeljenjeID=10
```



AND operator

Prikazati imena i datume zaposlenja zaposlenih u odeljenju 30 koji se se zaposlili posle 31. 12. 2016. godine:

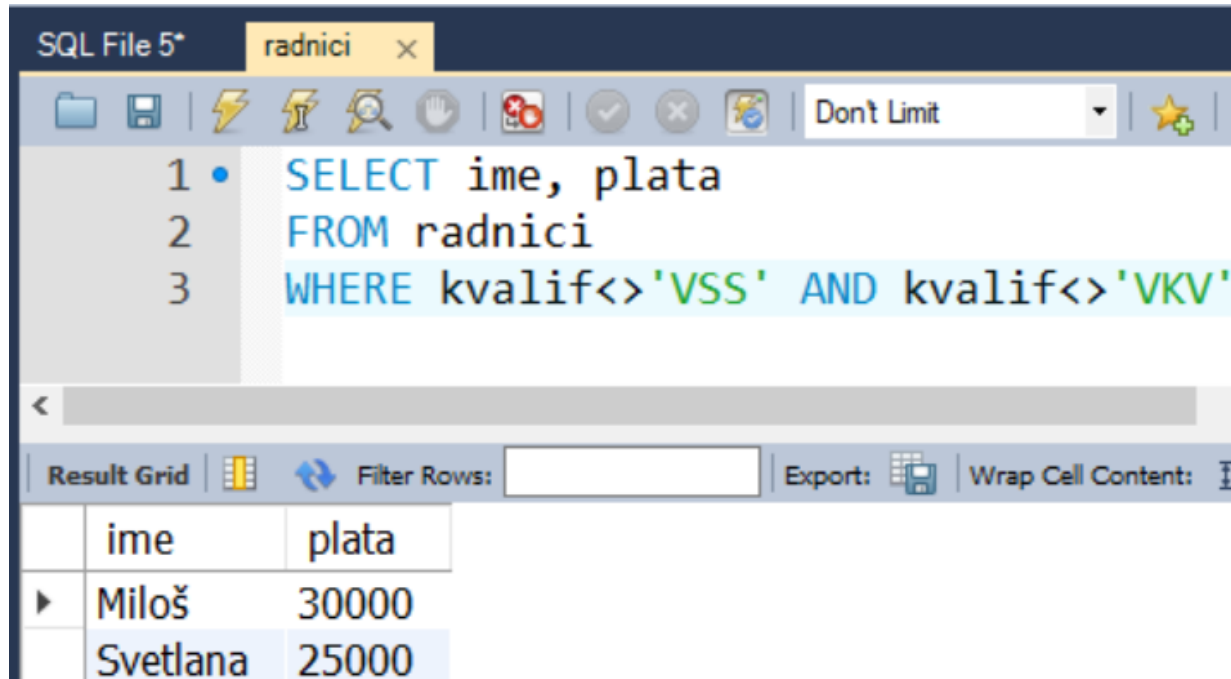
```
SELECT ime, datum_zap  
FROM radnici  
WHERE odeljenjeID=30 AND datum_zap>'2016-12-31'
```



AND i <> operator

Prikazati imena i plate zaposlenih koji nemaju visoku (VSS) ili visoko kvalifikovanu stručnu spremu (VKV).

```
SELECT ime, plata  
FROM radnici WHERE kvalif<>'VSS'  
AND kvalif<>'VKV'
```

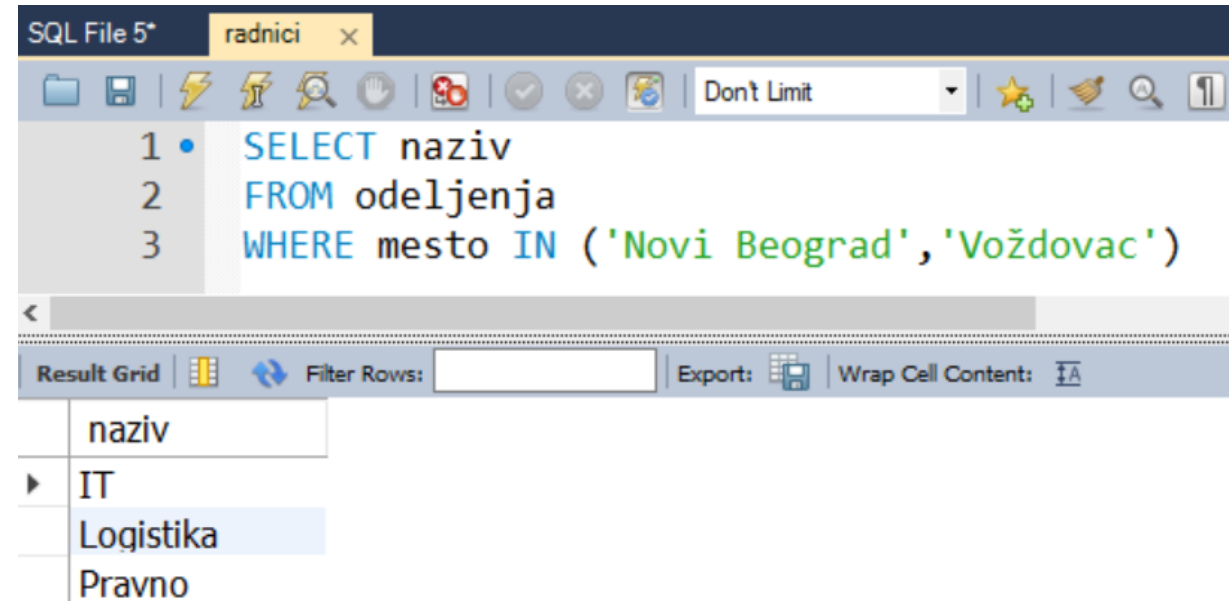


IN operator

Prikazati nazive odeljenja koja se nalaze na Novom Beogradu ili Voždovcu.

```
SELECT naziv FROM odeljenja  
WHERE mesto IN ('Novi Beograd','Voždovac')
```

Operator poređenja IN se koristi da bi se proverilo da li neka vrednost pripada skupu vrednosti.

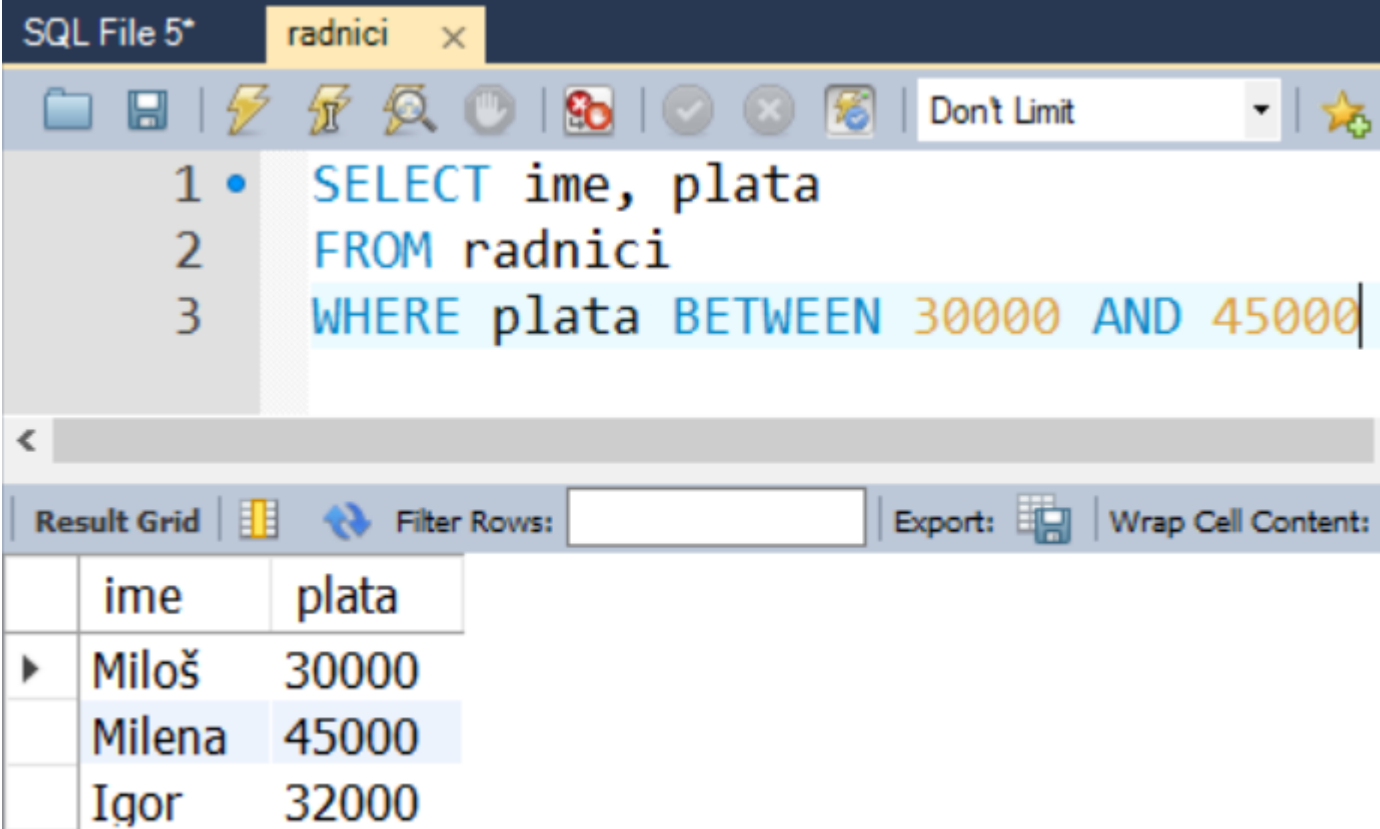


BETWEEN...AND operator

Prikazati imena i plate radnika čija je plata između 30000 i 45000 uključujući i te vrednosti.

```
SELECT ime, plata FROM radnici  
WHERE plata BETWEEN 30000  
AND 45000
```

Ne postoje neke posebne razlike u performansama kada se koristi BETWEEN...AND u odnosu na druge operatore komparacije. Operator BETWEEN...AND se koristi zbog jednostavnosti u čitanju koda i dobijanju rezultata iz baze podataka.



The screenshot shows a SQL IDE window titled "SQL File 5*" with a tab for "radnici". The query editor contains the following SQL code:

```
1 • SELECT ime, plata  
2 FROM radnici  
3 WHERE plata BETWEEN 30000 AND 45000
```

Below the query editor, the "Result Grid" displays the results of the query. The results are as follows:

	ime	plata
▶	Miloš	30000
	Milena	45000
	Igor	32000

BETWEEN...AND ili IN

Po čemu se razlikuju rezultati datih upita:

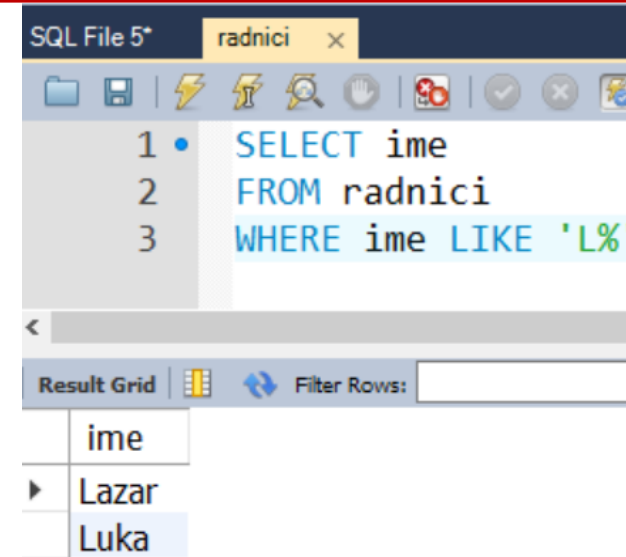
```
SELECT title, year  
FROM d_cds  
WHERE year BETWEEN 1999 AND 2001;
```

```
SELECT title, year  
FROM d_cds  
WHERE year IN (1999,2001);
```

LIKE operator

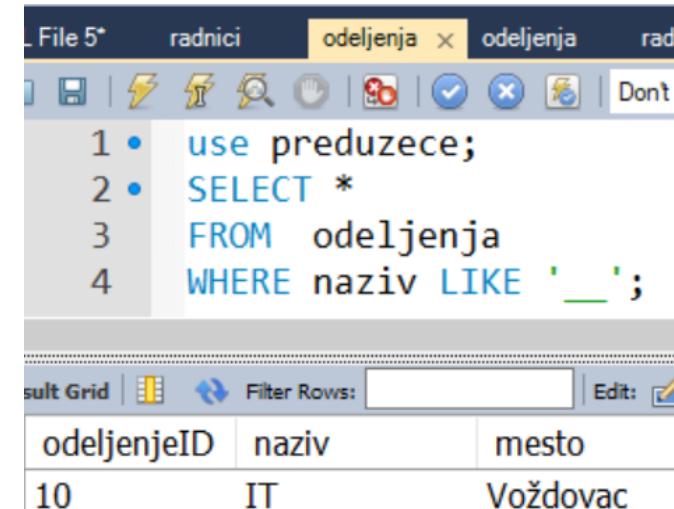
Prikazati imena radnika čije ime počinje sa slovom L

```
SELECT ime  
FROM radnici  
WHERE ime LIKE 'L%'
```



Prikazati odeljenja koja imaju 2 karaktera u nazivu odeljenja.

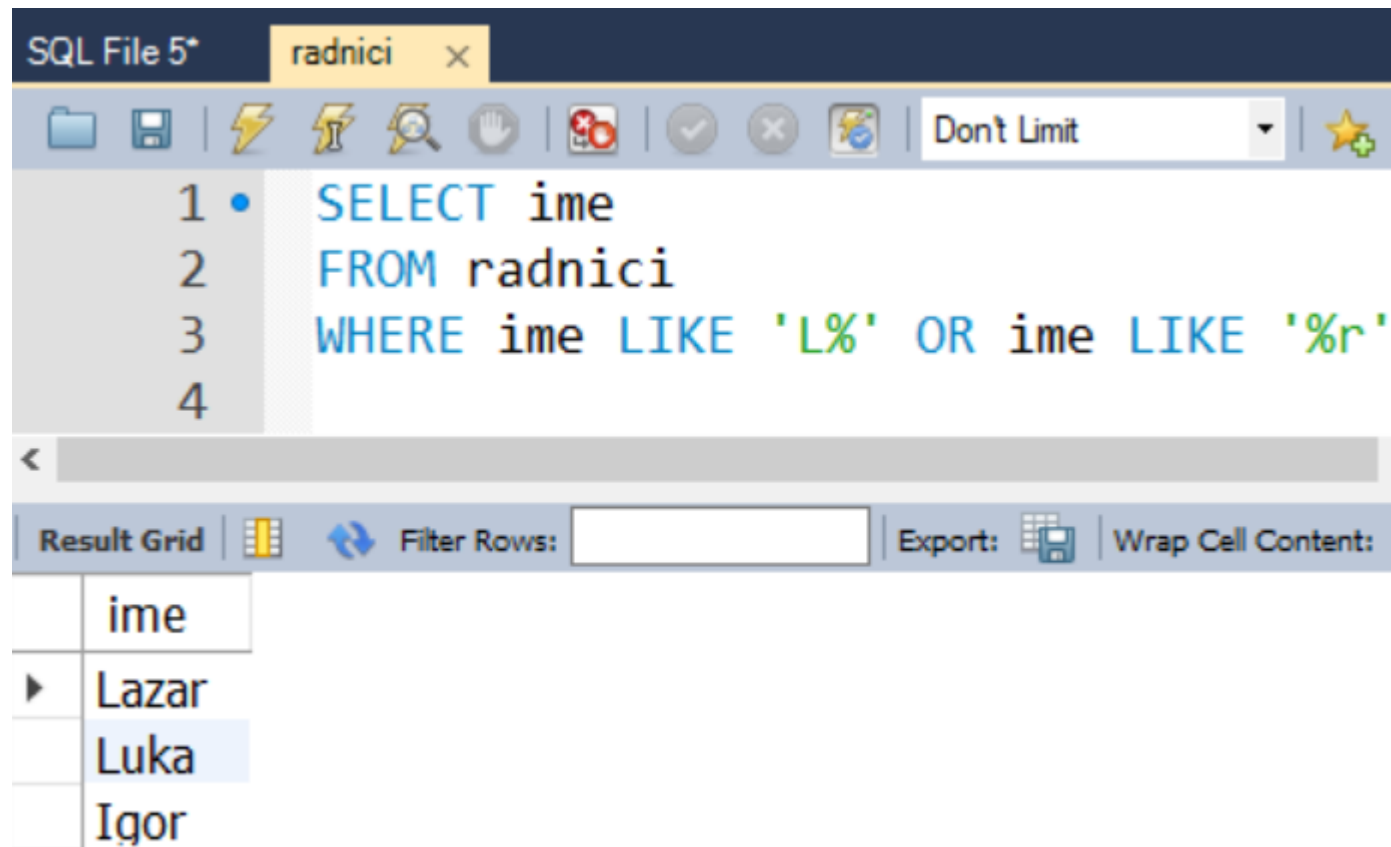
```
SELECT *  
FROM odeljenja  
WHERE naziv LIKE '__';
```



OR operator

Prikazati imena radnika čije ime počinje slovom L ili završava slovom r

```
SELECT ime  
FROM radnici  
WHERE ime  
LIKE 'L%' OR  
ime LIKE '%r'
```



SELECT – ALL i ANY

Operatori ALL i ANY se koriste na sledeći način:

kolona/izraz jednoredni_operator ALL (niz vrednosti)

kolona/izraz jednoredni operator ANY (nizvrednosti)

- ❑ Uslov sa ALL je ispunjen ako jednoredni operator vraća vrednost **True** za sve vrednosti iz datog niza vrednosti.
- ❑ Uslov sa ANY je ispunjen ako jednoredni operator vraća vrednost **True** za bar jednu vrednost iz datog niza vrednosti

Zadaci

1. Prikazati ime i mesto odeljenja za odeljenja koja sadrži slovo n u imenu.
2. Prikazati nazive različitih poslova u preduzeću.
3. Prikazati imena zaposlenih koji rade u odeljenju 10.
4. Prikazati imena i identifikacione brojeve svih zaposlenih u preduzeću za radnike koji imaju kvalifikaciju KV ili VKV ili VSS.
5. Prikazati ime, prezime i platu za radnike koji rade obavljaju posao analitičara ili imaju platu veću od 2000 i manju od 5000 uključujući i te vrednosti .
6. Prikazati posao zaposlenih koji rade u odeljenju 20 i zaposleni su posle 2000. godine.

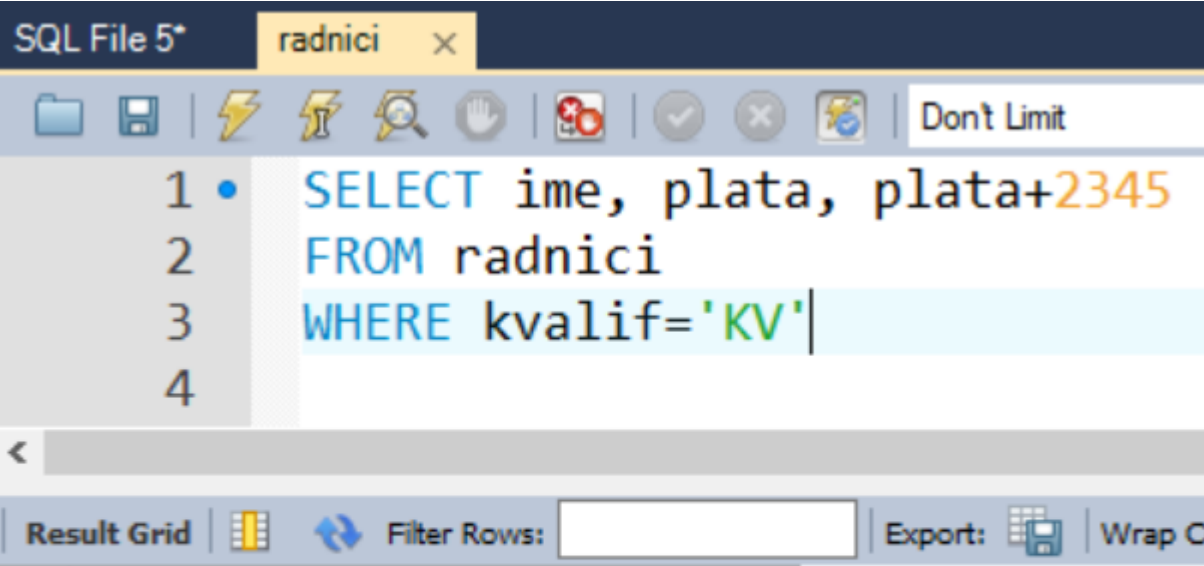
SELECT i aritmetički operatori

Primer primene aritmetičkih operatora:

```
SELECT ime, plata, plata+2345  
FROM radnici  
WHERE kvalif='KV'
```

Dati primer pokazuje upotrebu aritmetičke operacije sabiranja da bi se izračunala plata povećana za 2345 zaposlenima koji imaju KV kvalifikaciju.

Prikazana je kao nova kolona plata+2345.



The screenshot shows a SQL IDE window titled 'SQL File 5*' with a tab for 'radnici'. The query editor contains the following SQL statement:

```
1 • SELECT ime, plata, plata+2345  
2 FROM radnici  
3 WHERE kvalif='KV'  
4
```

Below the query editor, the 'Result Grid' displays the results of the query. The grid has four columns: 'ime', 'plata', and 'plata+2345'. There are two rows of data:

	ime	plata	plata+2345
▶	Miloš	30000	32345
	Svetlana	25000	27345

SELECT i aritmetički operatori

Prioritet aritmetičkih operacija:

Primeri primene aritmetičkih operatora:

1. ()
2. *, /
3. +, -

Operator Precedence

LAST_NAME	SALARY	12 * SALARY + 100
Hartstein	13000	156100
Fay	6000	72100
Higgins	12000	144100

Using Parentheses

LAST_NAME	SALARY	12 * (SALARY+100)
Hartstein	13000	157200
Fay	6000	73200
Higgins	12000	145200

SELECT – primeri selekcije i sortiranja-ORDER BY

- ❑ **UREĐIVANJE REZULTATA UPITA (SORTIRANJE)** po vrednosti izabranih atributa postiže se klauzulom **ORDER BY**.
- ❑ Klauzula **ORDER BY** eksplicitno određuje redosled instanci u rezultatu upita po nekom kriterijumu (po abecedi, po veličini, itd.) u rastućem (**ASC**) ili opadajućem (**DESC**) poretku.
- ❑ Klauzula **ORDER BY** je **uvek poslednja klauzula** u naredbi **SELECT** jer se instance najpre izdvajaju, a zatim uređuju.
- ❑ Sortiranje se može obavljati po više atributa.
- ❑ Rastući redosled se podrazumeva, pa je odrednica ASC izostavljena.

SELECT – primeri selekcije i sortiranja-ORDER BY

Prikazati sve podatke o radnicima sortirane po imenima u rastućem redosledu.

```
SELECT * FROM radnici  
ORDER BY ime
```

Prikazati sve podatke o radnicima sortirane po imenima u rastućem pa po plati u opadajućem redosledu.

```
SELECT * FROM radnici  
ORDER BY ime ASC, plata DESC
```

SELECT – selekcija i sortiranje-ORDER BY

Prikazati naziv fakulteta bez ponavljanja sortirane po opadajućem redosledu

```
SELECT DISTINCT naziv FROM fakultet  
ORDER BY naziv DESC;
```

Prikazati sve podatke o radnicima koji imaju kvalifikaciju KV sortirane po imenima u rastućem pa po plati u opadajućem redosledu.

```
SELECT * FROM radnici  
WHERE kvalif='KV'  
ORDER BY ime ASC, plata DESC
```

Rad sa NULL vrednostima

- ❑ Null je vrednost koja je nedodeljena, nepoznata ili neupotrebljiva.
- ❑ Null nije isto što i nula ili blanko znak. U SQL-u nula je broj, blanko je znak, a null znači ne postojanje bilo kakve vrednosti. Dakle, NULL znači da nema vrednosti.
- ❑ Ako je neko svojstvo neprimenljivo na većinu primeraka entiteta, onda taj atribut treba eliminisati iz tabele još u fazi projektovanja.
- ❑ Ako je vrednost bilo koje kolone u aritmetičkom izrazu null, rezultat je null (nepoznat).

Rad sa NULL vrednostima

- ❑ Npr., mogući atribut entiteta STUDENT je **nagrada**. Međutim, većina studenata u toku studija ne osvaja nikakve nagrade, pa bi za većinu studenata u toj rubrici stajala NULL vrednost. S druge strane, neki studenti osvajaju i više nagrada, pa taj podatak ne bi bio atomski. Stoga je ispravno uvesti novi entitet **NAGRADA<šifra_nagrade#, naziv_nagrade, broj_indeksa\$, godina>**. Tada bi se u odgovarajućoj tabeli registrovali samo nagrađeni studenti svojim brojem indeksa i nazivom nagrade koju su dobili.
- ❑ Mogući atribut entiteta RADNIK je **telefon**, ali je moguće da radnici nisu u obavezi da podatak o telefonskom broju učine dostupnim.
- ❑ Tabela RADNIK u prethodnim primerima je definisana tako da dopušta da postoje radnici za koje broj odeljenja nije poznat (odgovarajući atribut može uzeti vrednost NULL).

Rad sa NULL vrednostima

Vrednosti u koloni *Nova plata* izračunate su po formuli:

$$\text{Nova plata} = \text{Plata} * (1 + \text{Povećanje})$$

Prezime	Posao	Plata	Povećanje	Nova plata
King	AD-PRES	24000	2%	24480
Kochhar	AD-VP	17000		
Zlotkey	SA-MAN	10500	0%	10500
Abel	SA-REP	8600	5%	9030

Kochhar i Zlotkey nisu imali povećanje pa bi trebalo da ostanu na istoj plati, ali je Kochhar ostao bez plate. Zašto?

Rad sa NULL vrednostima

Rad sa NULL vrednostima u SQL-u podržan je operatorima:

- ❑ IS NULL (“je nula vrednost”)
- ❑ IS NOT NULL (“nije nula vrednost”)
- ❑ Prikazati sve podatke o radnicima koji nisu raspoređeni ni u jedno odeljenje.

```
SELECT *  
FROM RADNIK  
WHERE odeljenjeID IS NULL
```

- ❑ Prikazati imena i datume zaposlenja za sve zaposlene koji imaju premiju.

```
SELECT ime, datum_zap  
FROM RADNIK  
WHERE premija IS NOT NULL
```

Rad sa NULL vrednostima

Rad sa **NULL** vrednostima u **SQL-u** podržan je i funkcijom **ISNULL** u slučaju SQL Servera odnosno za MySQL **IFNULL** (<ime_atributa> , <vrednost>):

- ❑ Služi da u upitu sistematski dodeli datu vrednost poljima koja u datoj koloni imaju vrednost **NULL**. Ta dodela vrednosti je samo privremena i odnosi se na izvršavanje tog upita.
- ❑ S obzirom da je cilj dodele da se nad poljima sa **NULL** vrednostima obavi neka operacija, vrednost koja se dodeljuje je obično neutralna u odnosu na tu operaciju.
- ❑ Ako treba da sabiramo ili oduzimamo, vrednost treba da bude 0; ako treba da množimo ili delimo vrednost treba da bude 1; ako treba da konkatenujemo znakovne podatke vrednost treba da bude prazna znakovna konstanta.

Rad sa NULL vrednostima

Neka je tabela RADNIK dopunjena atributima ***plata*** i ***kredit***:

```
ALTER TABLE RADNIK  
ADD COLUMN plata INT,  
ADD COLUMN kredit INT
```

Sledeća naredba prikazuje identifikacione brojeve, prezimena, imena i primanja svakog radnika:

```
SELECT id_br, prezime, ime, plata+IFNULL(kredit,0)  
FROM RADNIK
```

ALIJASI KOLONE

- Alijas je način preimenovanja imena kolona na izlazu.
- Bez alijasa, kolone prikazane kao rezultat SQL –rečenice imaju ista imena kao kolone u tabeli ili ime koje pokazuje aritmetičku operaciju, kao $12 * (\text{plata} + 100)$.
- Nekada je poželjno da na izlazu bude prikazano ime kolone koje je lakše razumeti, ime koje je više „user friendly“. Alijasi kolona dozvoljavaju preimenovanje kolone na izlazu.

ALIJASI KOLONE

❑ Postoji nekoliko pravila kod korišćenja alijasa kolona.

Alijas kolone:

- menja naslov kolone,
- koristan je kod proračuna,
- navodi se odmah posle imena kolone,
- može imati opcionalnu ključnu reč AS između imena kolone i alijasa,
- zahteva znake navoda ako alijas sadrži blanko-znak, specijalne znake ili je osetljiv na velika i mala slova.

ALIJASI KOLONE

Primeri upotrebe alijasa:

SELECT ime, prezime, plata **AS** 'Mesečna plata zaposlenog'
FROM radnici;

SELECT ime, prezime, plata*12 **AS**
'Uvaćana plata zaposlenog'
FROM radnici;

The screenshot shows two SQL queries and their results in a database tool interface.

Query 1:

```
1 • SELECT ime, prezime, plata AS 'Mesečna plata zaposlenog'
2 FROM radnici;
```

Result Grid 1:

ime	prezime	Mesečna plata zaposlenog
Petar	Petrović	35000
Janko	Marković	39000
Marko	Marković	40000

Query 2:

```
1 • SELECT ime, prezime, plata*12 AS 'Uvaćana plata zaposlenog'
2 FROM radnici;
```

Result Grid 2:

ime	prezime	Uvaćana plata zaposlenog
Petar	Petrović	420000
Janko	Marković	468000
Marko	Marković	480000
Jasna	Pavlović	300000
Janko	Pavlović	348000
Zoran	Ilić	300000
Pavle	Marković	240000
Marko	Krunić	420000
Zorica	Marić	348000
Milan	Jovanović	324000
Milica	Simić	468000
Ljiljana	Radić	300000

Grupisanje instance - GROUP BY

- ❑Klauzula **GROUP BY** iza koje sledi lista atributa po kojima se grupisanje vrši omogućava **GRUPISANJE instanci** u rezultatu naredbe **SELECT**.
- ❑Klauzulom **GROUP BY** instance u rezultatu bivaju presložene tako da sve instance unutar grupe imaju jednake vrednosti atributa po kojima se grupisanje vrši.
- ❑Ako se navede više atributa po kojima se vrši grupisanje, prvo se ređaju instance sa jednakom vrednošću prvog atributa, zatim se unutar tih grupa preslažu prema vrednostima drugog atributa, itd.

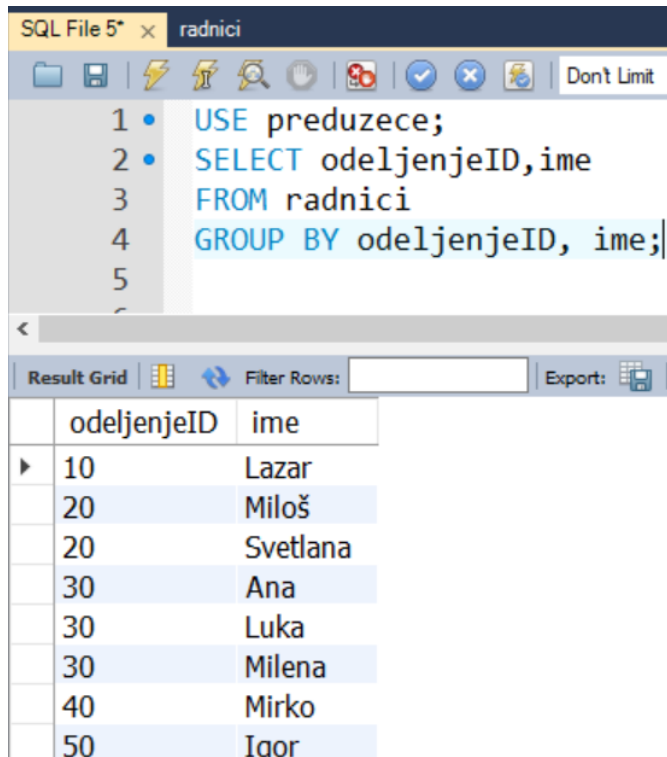
Grupisanje instance - GROUP BY

- ☐ Iako izgleda da postoji velika sličnost između klauzula **ORDER BY** i **GROUP BY** ova sličnost je samo prividna.
- ☐ Uređenje se ne mora vršiti po svim atributima navedenim u SELECT klauzuli, a grupisanje mora;
- ☐ Uređenje može biti i rastuće i opadajuće, a grupisanje samo rastuće;
- ☐ Agregatne funkcije, o kojima će kasnije biti reči, mogu biti primenjene na grupe podataka i u tome je upravo i prava uloga klauzule **GROUP BY**.

Grupisanje instance - GROUP BY

Prikazati imena radnika po odeljenjima.

```
SELECT odeljenjeID, ime FROM radnici  
GROUP BY odeljenjeID, ime;
```

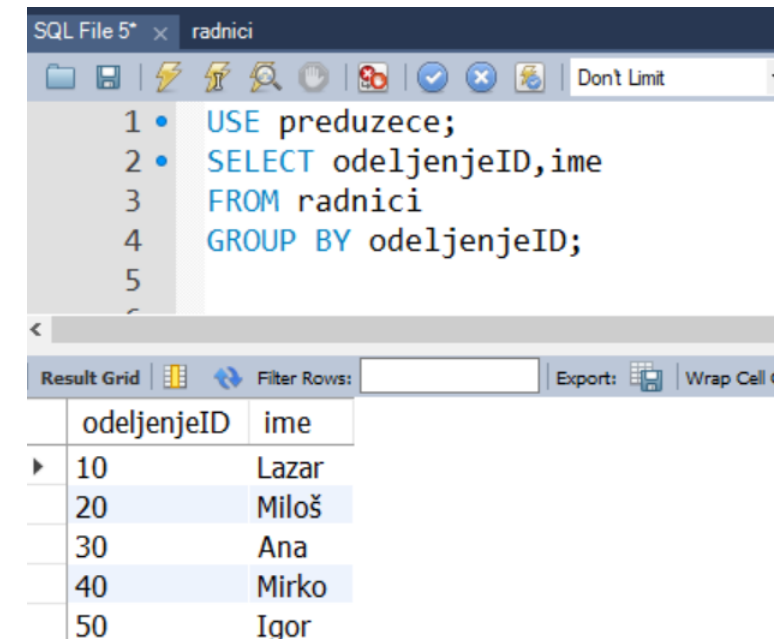


SQL File 5* x radnici

```
1 • USE preduzece;  
2 • SELECT odeljenjeID, ime  
3 FROM radnici  
4 GROUP BY odeljenjeID, ime;  
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell

	odeljenjeID	ime
▶	10	Lazar
	20	Miloš
	20	Svetlana
	30	Ana
	30	Luka
	30	Milena
	40	Mirko
	50	Igor



SQL File 5* x radnici

```
1 • USE preduzece;  
2 • SELECT odeljenjeID, ime  
3 FROM radnici  
4 GROUP BY odeljenjeID;  
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell

	odeljenjeID	ime
▶	10	Lazar
	20	Miloš
	30	Ana
	40	Mirko
	50	Igor

Operacije i funkcije

Operacije i funkcije omogućavaju da se na osnovu podataka prisutnih u bazi dođe do novih relevantnih informacija.

- ❑ Aritmetičke – +, -, *, /, % (ostatak celobrojnog deljenja),
- ❑ Znakovne – + (konkatenacija, spajanje stringova),
- ❑ Relacijske – =, <, >, <>, <=, >=,
- ❑ Logičke – NOT, AND, OR,

MySQL podržava veliki broj ugrađenih funkcija, kao i mogućnost uvođenja korisničkih funkcija. **Funkcije** mogu biti: **determinističke** i **nedeterminističke**. Determinističke funkcije vraćaju isti rezultat pri svakom pozivu nad istim skupom ulaznih vrednosti, a nedeterminističke ne.

Operacije i funkcije

Ugrađene funkcije u SQL-u su grupisane u desetak kategorija, a za nas su interesantne sledeće:

- ☐ Matematičke (numeričke) funkcije,
- ☐ Funkcije za rad sa znakovnim podacima,
- ☐ Funkcije za rad sa datumima,
- ☐ Agregatne funkcije.

Operacije i funkcije

Ime funkcije

NOW
CURDATE
CURTIME
DATE_ADD
DATE_SUB
DATEDIFF
DATE_FORMAT
FORMAT
TRIM
LOWER
UPPER
SUBSTRING
REVERSE
CONCAT

Opis funkcije

Vraća trenutni vremenski žig servera.
Vraća trenutni datum servera.
Vraća trenutno vreme servera.
Dodaje određeni vremenski interval na vrednost datuma.
Oduzima određeni vremenski interval od vrednosti datuma.
Računa razliku između dva datuma u danima.
Prikazuje datum u definisanom formatu.
Vraća string dobijen na osnovu definisanog formata.
Uklanja sve beline ispred i iza vidljivih karaktera u stringu.
Pretvara sva velika slova u mala u datom stringu.
Pretvara sva mala slova u velika u datom stringu.
Vraća deo stringa sa određene pozicije i broja karaktera.
Vraća okrenut string.
Vraća string dobijen spajanjem više tekstualnih vrednosti.

Operacije i funkcije

Ime funkcije

CONCAT_WS

SIN

COS

ASIN

ACOS

ATAN

TAN

ATAN2

COT

ABS

CEIL

FLOOR

MOD

ROUND

TRUNCATE

Opis funkcije

Vraća string spajanjem tekstualnih vrednosti separatorom.

Vraća rezultat trigonometrijske funkcije sin.

Vraća rezultat trigonometrijske funkcije cos.

Vraća rezultat trigonometrijske funkcije asin.

Vraća rezultat trigonometrijske funkcije acos.

Vraća rezultat trigonometrijske funkcije atan.

Vraća rezultat trigonometrijske funkcije tan.

Vraća rezultat trigonometrijske funkcije atan za 2 argumenta.

Vraća rezultat trigonometrijske funkcije ctan.

Vraća apsolutnu vrednost datog broja.

Vraća broj zaokružen na višu celobrojnu graničnu vrednost.

Vraća broj zaokružen na nižu celobrojnu graničnu vrednost.

Vraća ostatak pri deljenju dva broja.

Zaokružuje realnu vrednost na određeni broj decimala.

Odseca decimale iza određene pozicije, bez zaokurživanja.

Operacije i funkcije

Ime funkcije

Opis funkcije

CONV	Konvertuje zapis broja iz jedne osnovice (baze) u drugu.
DEGREES	Konvertuje ugao iz radijana u stepene.
RADIANS	Konvertuje ugao iz stepeni u radijane.
LN	Vraća vrednost prirodnog logaritma.
LOG	Vraća vrednost prirodnog logaritma.
LOG10	Vraća vrednost logaritma osnovice 10.
LOG2	Vraća vrednost logaritma osnovice 2.
SIGN	Vraća znak date vrednosti.
DIV	Koristi se za celobrojno deljenje.
EXP	Koristi se za stepenovanje.
POW	Vraća vrednost broja stepenovanog na određeni broj.
SQRT	Vraća kvadratni koren date vrednosti.
RAND	Vraća pseudo-nasumičnu realnu vrednost.
LENGTH	Vraća dužinu stringa u bajtovima.
ASCII	Vraća ASCII numeričku vrednost prvog karaktera.
CHAR	Vraća karakter koji predstavlja data vrednost.

Operacije i funkcije

Ime funkcije

SPACE
REPEAT
REPLACE
STRCMP
BIN
HEX
OCT
UNHEX
LPAD
RPAD
LTRIM
RTRIM
LEFT
RIGHT
CHAR_LENGTH

Opis funkcije

Vraća string sa definisanim brojem razmaka.
Umnožava dati string dati broj puta.
Menja pominjanje dela teksta u datom tekstualnom podatku.
Vrši poređenje dva tekstualna podatka.
Vraća string koji sadrži binarni prikaz određenog broja.
Vraća heksadecimalni prikaz datog broja.
Vraća oktalni zapis datog broja.
Vraća vrednost konvertovanu iz heksadecimalnog zapisa.
Vraća tekst određene širine dopunom simbolima levo.
Vraća tekst određene širine dopunom simbolima desno.
Uklanja beline ispred teksta.
Uklanja beline iza teksta.
Vraća dati broj karaktera sa početka datog stringa.
Uklanja beline iza teksta.
Vraća broj karaktera u datom stringu.

Operacije i funkcije

Ime funkcije

FROM_BASE64
TO_BASE64
MATCH
LIKE
RLIKE
ENCODE
DECODE
PASSWORD
SHA1
SHA2
INET_ATON
INET_NTOA
IS_IPV4
INET6_ATON
INET6_NTOA
IS_IPV6
UUID
UUID_SHORT
UUID_TO_BIN

Opis funkcije

Koristi se za dekodiranje teksta iz BASE64 kodiranog zapisa.
Koristi se za kodiranje teksta u BASE64 kodirani zapis.
Obavlja tekstualnu pretragu u datom tekstualnom podatku.
Obavlja uparivanje teksta po prostom šablonu.
Obavlja uparivanje teksta po regularnom izrazu.
Kodira string
Dekodira string kodiran naredbom ENCODE
Računa PASSWORD heš string datog teksta
Računa SHA1 kontrolni string
Računa SHA2 kontrolni string
Konvertuje tekstualni zapis IP adrese u broječanu vrednost
Konvertuje broječanu vrednost zapisa IP adrese u tekst
Utvrđuje da li je dati argument ispravna IPv4 adresa
Konvertuje tekstualni zapis IPv6 adrese u broječanu vrednost
Konvertuje broječanu vrednost zapisa IPv6 adrese u tekst
Utvrđuje da li je dati argument ispravna IPv6 adresa
Vraća jedinstveni univerzalni identifikator (UUID)
Vraća smraćeni jedinstveni univerzalni identifikator (UUID)
Konvertuje tekstualni zapis UUID u binarni

Operacije i funkcije

Neka je tabela RADNIK kreirana sledećom naredbom:

```
CREATE TABLE RADNICI (  
  id_radnik INT PRIMARY KEY,  
  CONSTRAINT id_radnik_provera CHECK (id_radnik>=1 AND id_radnik<=99999),  
  ime VARCHAR(15) NOT NULL,  
  prezime VARCHAR(20) NOT NULL,  
  kvalif CHAR(3),  
  datum_zap DATE,  
  posao VARCHAR(10),  
  plata REAL,  
  odeljenjeID int,  
  CONSTRAINT odeljenjeID_sk FOREIGN KEY(odeljenjeID)  
  REFERENCES ODELJENJA(odeljenjeID)  
  ON DELETE NO ACTION  
  ON UPDATE CASCADE);
```

Operacije i funkcije

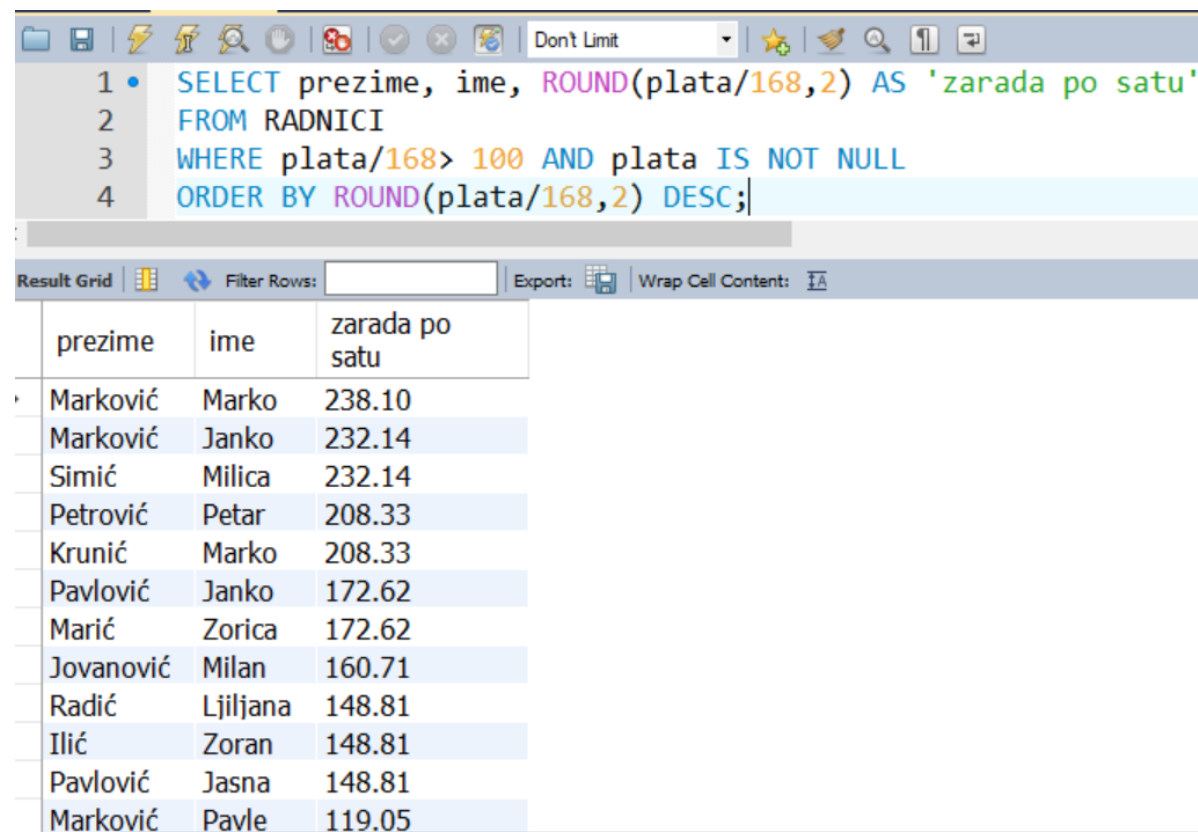
Popunjena na sledeći način:

id_radnik	ime	prezime	kvalif	datum_zap	posao	plata	odeljenjeID
11	Petar	Petrović	SSS	2000-03-21	knjigovođa	35000	50
12	Janko	Marković	VS	2001-10-19	sekretar	39000	50
22	Marko	Marković	VSS	2001-03-15	analitičar	40000	10
23	Jasna	Pavlović	KV	2008-04-04	spremačica	25000	20
30	Janko	Pavlović	KV	2011-04-19	kuvar	29000	20
32	Zoran	Ilić	NULL	2015-06-09	NULL	25000	10
33	Pavle	Marković	SSS	2016-09-19	magacioner	20000	10
42	Marko	Krunić	VSS	2018-12-14	referent	35000	30
43	Zorica	Marić	NULL	2017-05-23	NULL	29000	50
44	Milan	Jovanović	NULL	2018-07-25	NULL	27000	10
45	Milica	Simić	SSS	2018-09-24	analitičar	39000	50
51	Ljiljana	Radić	VK	2018-12-12	kuvar	25000	20

Operacije i funkcije

Prikazati prezimena, imena i zarade svih radnika čija je zarada po satu veća od 100 novčanih jedinica. Radnici su sortirani prema opadajućem redosledu svojih zarada po satu. Podrazumeva se da mesec ima 168 radnih sati.

```
SELECT prezime, ime,  
ROUND(plata/168,2) AS 'zarada po satu'  
FROM RADNIK  
WHERE plata/168 > 100 AND plata IS  
NOT NULL  
ORDER BY ROUND(plata/168,2) DESC
```



The screenshot shows a database query editor with the following SQL query:

```
1 • SELECT prezime, ime, ROUND(plata/168,2) AS 'zarada po satu'  
2 FROM RADNICI  
3 WHERE plata/168 > 100 AND plata IS NOT NULL  
4 ORDER BY ROUND(plata/168,2) DESC;
```

Below the query, the results are displayed in a table with the following columns: prezime, ime, and zarada po satu. The results are sorted in descending order of hourly wage.

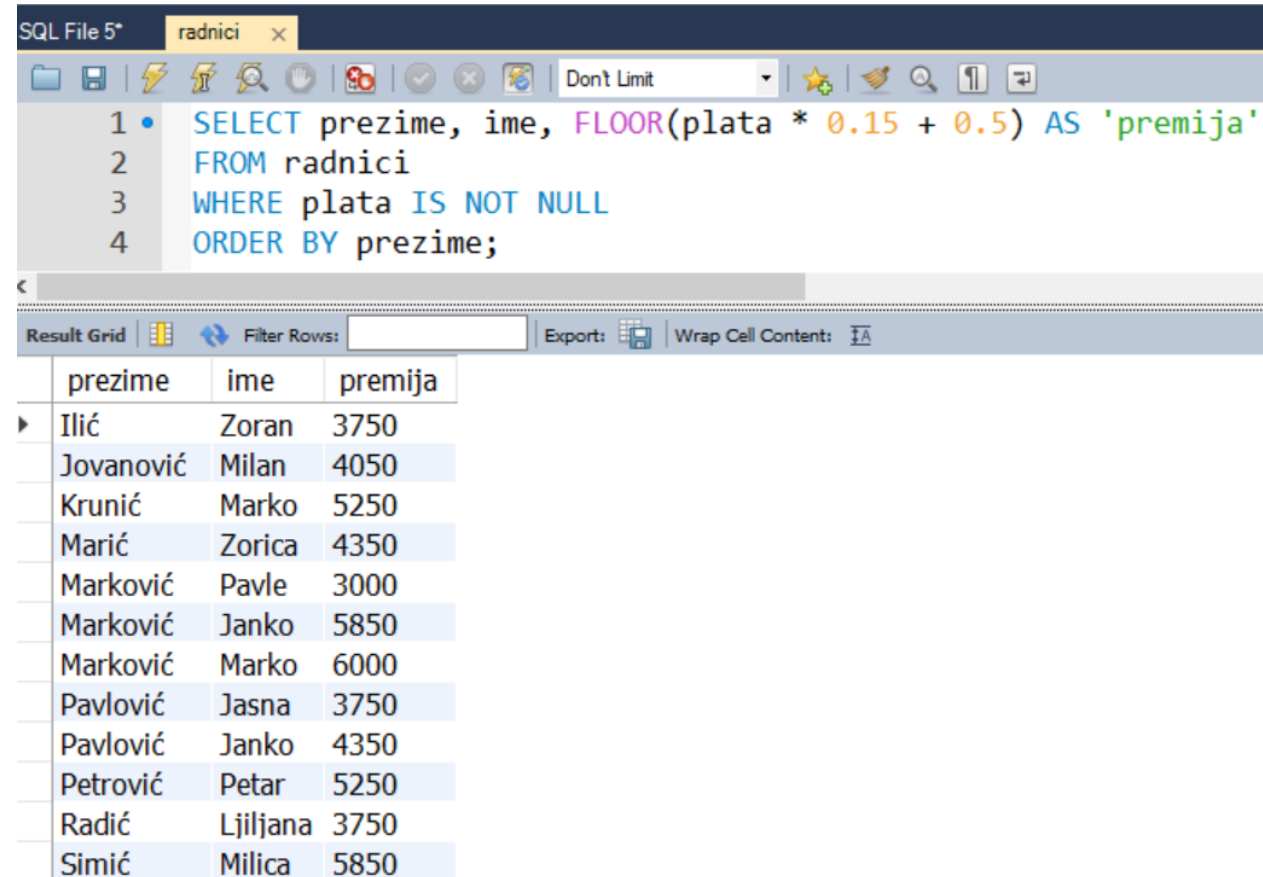
prezime	ime	zarada po satu
Marković	Marko	238.10
Marković	Janko	232.14
Simić	Milica	232.14
Petrović	Petar	208.33
Krunić	Marko	208.33
Pavlović	Janko	172.62
Marić	Zorica	172.62
Jovanović	Milan	160.71
Radić	Ljiljana	148.81
Ilić	Zoran	148.81
Pavlović	Jasna	148.81
Marković	Pavle	119.05

Operacije i funkcije

Prikazati prezimena, imena i premije svih radnika za koje podatak o plati nije NULL. Premija je 15% plate, zaokruženo na ceo broj. Radnici su sortirani po prezimenima.

```
SELECT prezime, ime,  
FLOOR(plata*0.15) AS 'premija'  
FROM RADNIK  
WHERE plata IS NOT NULL  
ORDER BY prezime
```

Isti efekat kao **FLOOR**(plata*0.15) ima i **ROUND**(plata*0.15,0)!



The screenshot shows a SQL IDE window titled 'SQL File 5*' with a tab for 'radnici'. The query editor contains the following SQL code:

```
1 • SELECT prezime, ime, FLOOR(plata * 0.15 + 0.5) AS 'premija'  
2 FROM radnici  
3 WHERE plata IS NOT NULL  
4 ORDER BY prezime;
```

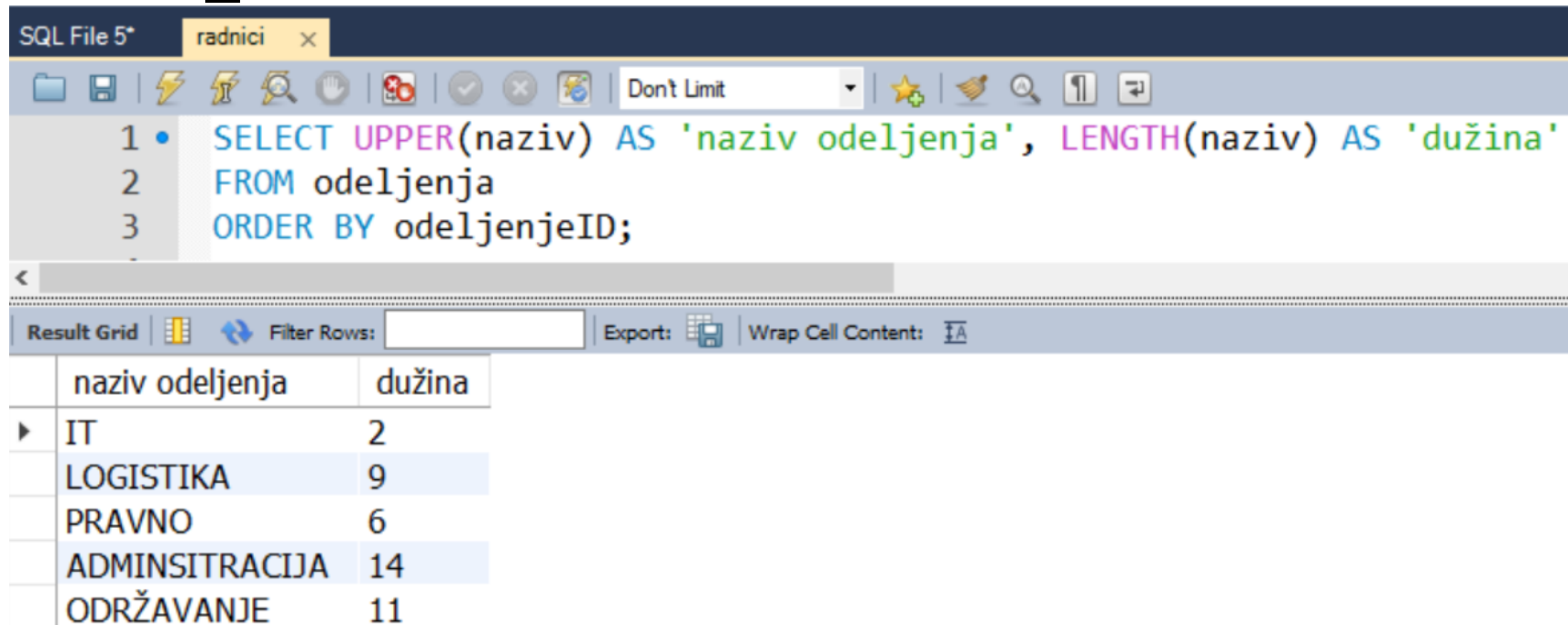
Below the query editor is a 'Result Grid' showing the results of the query. The grid has three columns: 'prezime', 'ime', and 'premija'. The results are sorted by 'prezime'.

prezime	ime	premija
Ilić	Zoran	3750
Jovanović	Milan	4050
Krunić	Marko	5250
Marić	Zorica	4350
Marković	Pavle	3000
Marković	Janko	5850
Marković	Marko	6000
Pavlović	Jasna	3750
Pavlović	Janko	4350
Petrović	Petar	5250
Radić	Ljiljana	3750
Simić	Milica	5850

Operacije i funkcije

Prikazati nazive odeljenja velikim slovima u abecednom poretku kao i broj karaktera u nazivu odeljenja.

```
SELECT UPPER(ime_od) AS 'naziv odeljenja', LENGTH(ime_od) AS 'dužina'  
FROM ODELJENJE  
ORDER BY ime_od
```



The screenshot shows a SQL IDE window titled 'SQL File 5*' with a tab 'radnici'. The query editor contains the following SQL code:

```
1 • SELECT UPPER(naziv) AS 'naziv odeljenja', LENGTH(naziv) AS 'dužina'  
2 FROM odeljenja  
3 ORDER BY odeljenjeID;
```

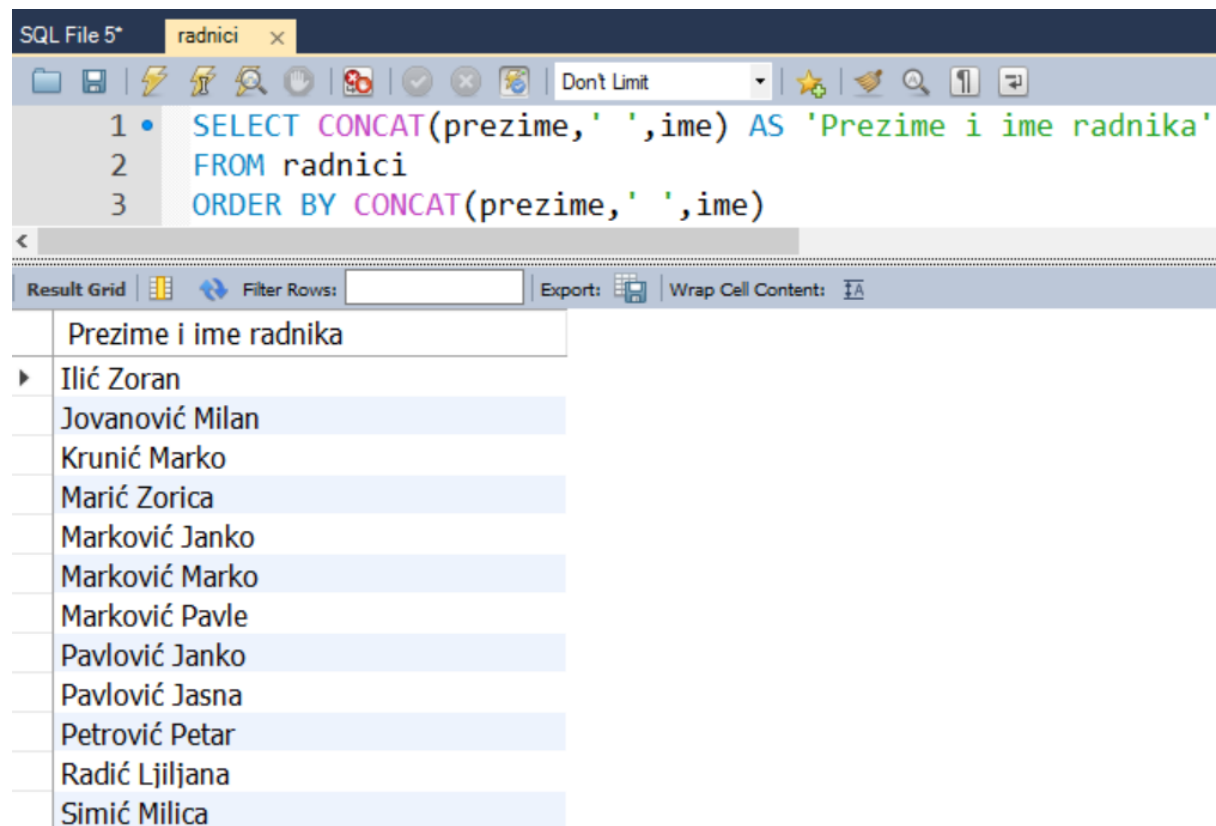
Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The grid has two columns: 'naziv odeljenja' and 'dužina'. The results are as follows:

naziv odeljenja	dužina
IT	2
LOGISTIKA	9
PRAVNO	6
ADMINSITRACIJA	14
ODRŽAVANJE	11

Operacije i funkcije

Prikazati prezimena i imena radnika u abecednom poretku, pri čemu su ime i prezime razdvojeni jednim blanko simbolom.

```
SELECT CONCAT(prezime,' ',ime) AS  
'Prezime i ime radnika'  
FROM radnici  
ORDER BY CONCAT(prezime,' ',ime)
```



The screenshot shows a SQL IDE window titled 'SQL File 5*' with a tab for 'radnici'. The query editor contains the following SQL code:

```
1 • SELECT CONCAT(prezime,' ',ime) AS 'Prezime i ime radnika'  
2 FROM radnici  
3 ORDER BY CONCAT(prezime,' ',ime)
```

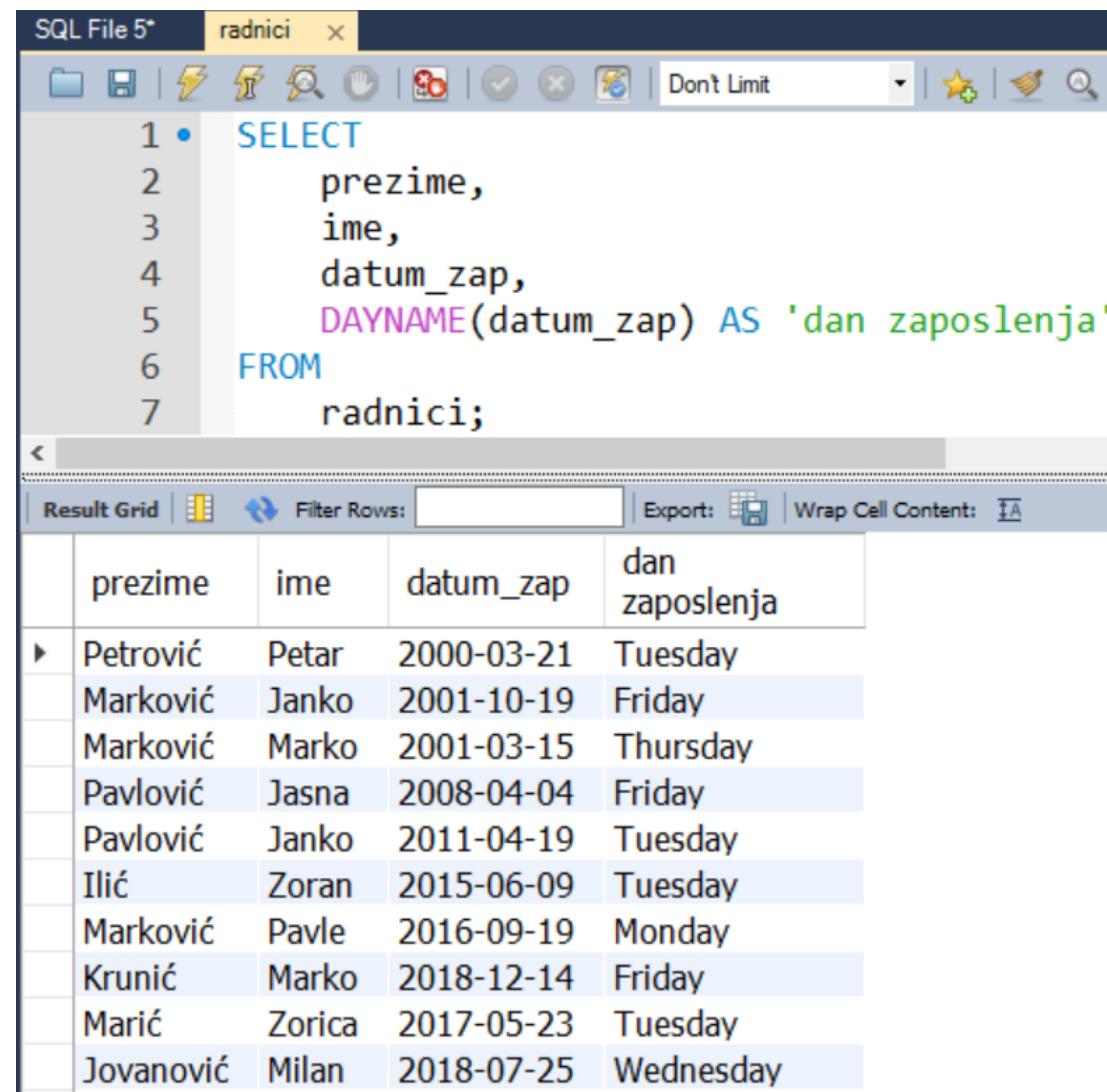
Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The results are sorted alphabetically by the concatenated name. The first row is expanded, showing the full name 'Ilić Zoran'.

Prezime i ime radnika
Ilić Zoran
Jovanović Milan
Krunić Marko
Marić Zorica
Marković Janko
Marković Marko
Marković Pavle
Pavlović Janko
Pavlović Jasna
Petrović Petar
Radić Ljiljana
Simić Milica

Operacije i funkcije

Prikazati prezimena, imena, datume zaposlenja i imena dana u sedmici kada su se radnici zaposlili.

```
SELECT prezime, ime, datum_zap,  
DAYNAME(datum_zap) AS 'dan zaposlenja'  
FROM radnici;
```



The screenshot shows a SQL IDE window titled 'SQL File 5*' with a tab for 'radnici'. The query editor contains the following SQL code:

```
1 SELECT  
2     prezime,  
3     ime,  
4     datum_zap,  
5     DAYNAME(datum_zap) AS 'dan zaposlenja'  
6 FROM  
7     radnici;
```

Below the query editor is the 'Result Grid' showing the results of the query. The grid has four columns: 'prezime', 'ime', 'datum_zap', and 'dan zaposlenja'. The results are as follows:

prezime	ime	datum_zap	dan zaposlenja
Petrović	Petar	2000-03-21	Tuesday
Marković	Janko	2001-10-19	Friday
Marković	Marko	2001-03-15	Thursday
Pavlović	Jasna	2008-04-04	Friday
Pavlović	Janko	2011-04-19	Tuesday
Ilić	Zoran	2015-06-09	Tuesday
Marković	Pavle	2016-09-19	Monday
Krunić	Marko	2018-12-14	Friday
Marić	Zorica	2017-05-23	Tuesday
Jovanović	Milan	2018-07-25	Wednesday

Operacije i funkcije

Neka radnik posle 20 meseci rada u preduzeću ima pravo na kredit. Prikazati prezimena, imena, datume zaposlenja i datume sticanja prava na kredit.

SELECT prezime, ime, datum_zap, **DATE_ADD**(datum_zap, interval 20 month) **AS** 'Datum sticanja prava na kredit' **FROM** radnici;

```
1 • SELECT
2     prezime,
3     ime, datum_zap,
4     DATE_ADD(datum_zap, interval 20 month) AS 'Datum sticanja prava na kredit'
5 FROM
6     radnici;
```

prezime	ime	datum_zap	Datum sticanja prava na kredit
Petrović	Petar	2000-03-21	2001-11-21
Marković	Janko	2001-10-19	2003-06-19
Marković	Marko	2001-03-15	2002-11-15
Pavlović	Jasna	2008-04-04	2009-12-04
Pavlović	Janko	2011-04-19	2012-12-19
Ilić	Zoran	2015-06-09	2017-02-09
Marković	Pavle	2016-09-19	2018-05-19

Operacije i funkcije

Prikazati prezimena, imena, datume zaposlenja i broj godina rada (Broj godina rada računati kao razliku između tekuće godine i godine zaposlenja).

```
SELECT prezime, ime, YEAR(CURDATE())-YEAR(datum_zap) AS 'Stož radnika'  
FROM radnici;
```

```
1 • SELECT  
2     prezime,  
3     ime,  
4     YEAR(CURDATE())-YEAR(datum_zap) AS 'Stož radnika'  
5 FROM  
6     radnici;
```

prezime	ime	Stož radnika
Petrović	Petar	19
Marković	Janko	18
Marković	Marko	18
Pavlović	Jasna	11
Pavlović	Janko	8
Ilić	Zoran	4
Marković	Pavle	3
Krunić	Marko	1
Marić	Zorica	2
Jovanović	Milan	1

Funkcije agregacije

Funkcije agregacije služe za dobijanje grupnih, sumarnih informacija.

Grupu može predstavljati čitava tabela, instance tabele koje zadovoljavaju postavljeni uslov (klauzula **WHERE**), grupe formirane po jednom ili više atributa (klauzula **GROUP BY**), instance grupe koje zadovoljavaju postavljeni uslov (klauzula **HAVING**). Sve agregatne funkcije su determinističke!

Agregatne funkcije	
Funkcija	Opis
AVG ([ALL DISTINCT] <i>numeričke vrednosti</i>)	nalazi prosek svih ili svih različitih vrednosti; NULL vrednosti se zanemaruju
SUM ([ALL DISTINCT <i>numeričke vrednosti</i>)	sabira vrednosti, NULL vrednost se zanemaruje

Agregatne funkcije

Agregatne funkcije	
Funkcija	Opis
COUNT (([ALL DISTINCT] <i>vrednosti</i>) *)	broj svih ili broj svih različitih vrednosti; ako je argument *, broje se i NULL vrednosti; rezultat je tipa INT
MAX ([ALL DISTINCT] <i>vrednosti</i>)	nalazi maksimum svih ili svih različitih vrednosti; NULL vrednosti se zanemaruju
MIN ([ALL DISTINCT] <i>vrednosti</i>)	nalazi minimum svih ili svih različitih vrednosti; NULL vrednosti se zanemaruju