

Lekcija 02

SQL: naredba SELECT (nad jednom tabelom)

dr Svetlana Cvetanović



SQL: NAREDBA SELECT (NAD JEDNOM TABELOM)

Uvod

01

02

03

04

Uvod

**DML-SELECT-
klauzula WHERE**

**Logički operatori
u naredbi SELECT**

**Izrazi nad nizom
karaktera u
naredbi SELECT**

**Aritmetičke i
sumarne funkcije
u naredbi SELECT**

□ *Vežba: DML-
SELECT-klauzula
WHERE*

□ *Vežba: DML-
SELECT-klauzula
WHERE*

*Vežba: Izrazi nad
nizom karaktera u
naredbi SELECT*

*Vežba: Aritmetičke i
sumarne funkcije u
naredbi SELECT*

SQL: NAREDBA SELECT (NAD JEDNOM TABELOM)

05

**Vežba:Klauzule
GROUP BY ,
HAVING i ORDER
BY**

- ❑ *Primeri: Klauzule
GROUP BY ,
HAVING*
- ❑ *Vežba:Klauzule
GROUP BY ,
HAVING i ORDER
BY*

06

**Upotreba NULL
vrednosti u
naredbi SELECT**

*Vežba: Upotreba
NULL vrednosti u
naredbi SELECT*

07

Zaključak

08

09

UVOD

Šta ćemo naučiti u ovoj lekciji?

Naredbom SELECT se može dobiti modifikovan sadržaj jedne ili više tabela i to primenom **aritmetičkih funkcija** nad kolona numeričkog tipa ili primenom odgovarajućih funkcija nad nizom karaktera.

Modifikovan sadržaj tabele se može dobiti i primenom operacije restrikcije tj. upotrebom **klauzule WHERE** uz koju se mogu koristiti različiti operandi koji se odnose na numeričke i nenumeričke sadržaje vrednosti atributa.

Naročito je važna upotreba **sumarnih funkcija** koje se primenjuju najčešće uz korišćenje grupne funkcije GROUP BY. Klauzula GROUP BY se primenjuje za dobijanje srednjih, minimalnih, maksimalnih, sumarnih vrednosti na nivou grupa podataka u tabeli.

DML-SELECT-klauzula WHERE

□ *Vežba: Primeri kvalifikovanog pretraživanja*

01

SELECT- KLAUZULA WHERE ZA SELEKCIJU SPECIFIČNIH N-TORKI

Korišćenje operatora = i !=

Primer: Prikazati sve podatke o radnicima koji rade u RJ 30 (slika 3.)

```
SELECT *
FROM RADNIK
WHERE $RJ != 30;
```

Slika 3. Upit za izvršenje prethodnog primera

Iz naredbe SELECT se dobija sledeći izlaz (slika 4.):

	S_RADNIK	S_RJ	IME	POSAO	S_RUKO	DAT_ZAP	LD	PREMIJA
1	3199	30	MILAN	TRG PUTNIK	3398	20-FEB-01	160000	30000
2	3221	30	PETAR	TRG PUTNIK	3398	22-FEB-01	125000	50000
3	3354	30	MARIJA	TRG PUTNIK	3398	28-SEP-01	125000	140000
4	3398	30	ANA	RUKOVODILAC	3539	01-MAY-01	285000	(null)
5	3544	30	GORAN	TRG PUTNIK	3398	08-SEP-01	150000	0
6	3600	30	JANKO	ANALITICAR	3398	03-DEC-01	95000	(null)

Slika 4. Izlaz iz naredbe SELECT sa slike 3.

Ovim primerom je realizovana operacija relacione algebre SELEKCIJA (RESTRIKCIJA) korišćenjem operatora = .

U navedenom primeru bi mogao da se menja samo uslov selekcije n-torki specificiran WHERE klauzulom, dok bi podaci koji se prikazuju ostali isti.

Primer: Prikazati sve podatke o radnicima koji ne rade u RJ 30 (slika 5.)

```
SELECT *
FROM RADNIK
WHERE $RJ != 30;
```

Slika 5. Upit za izvršenje prethodnog primera

Iz naredbe SELECT se dobija sledeći izlaz (slika 6.):

S_RADNIK	S_RJ	IME	POSAO	S_RUKO	DAT_ZAP	LD	PREMIJA
3068	20	STEFAN	ANALITICAR	35602	17-DEC-00	80000	
3266	20	RAJKO	RUKOVODILAC	3539	02-APR-01	297500	
3482	10	IVAN	RUKOVODILAC	3539	09-JUN-01	245000	
348	20	IVANKA	SAVETNIK	3266	09-NOV-01	200000	
3539	10	JOVAN	POSREDOVAC		17-NOV-01	500000	
3576	20	PELENA	ANALITICAR	3488	23-SEP-01	110000	
3624	10	DEJAN	ANALITICAR	3482	23-FEB-02	130000	
3682	20	FILIP	SAVETNIK	3266	03-DEC-01	300000	

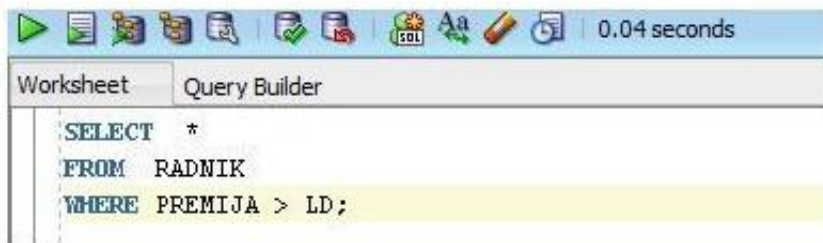
Slika 6. Izlaz iz naredbe SELECT sa slike 5.

Primerom je prikazano korišćenje operatora != u WHERE klauzuli naredbe SELECT.

SELECT- KLAUZULA WHERE ZA SELEKCIJU SPECIFIČNIH N-TORKI

Upotreba operatora > i <

Primer: Prikaži podatke o radnicima koji imaju premiju veću od ličnog dohotka (slika 7.)



```
SELECT *
FROM RADNIK
WHERE PREMIJA > LD;
```

Slika 7. Upit za izvršenje prethodnog primera

Naredbom SELECT se dobija sledeći izlaz (slika 8.):

S_RADNIK	S_RZ IME	POSLO	S_KODI DAT_TAP	ID PREMIJA
3354	30 RAPAZA	TRG POTNIK	3398 28-SEP-01	125000 140000

Slika 8. Izlaz iz naredbe SELECT sa slike 7.

Primerom je prikazano korišćenje operatora > u WHERE klauzuli naredbe SELECT.

Primer: Prikaži podatke o radnicima koji imaju premiju manju od ličnog dohotka (slika 9.)



```
SELECT *
FROM RADNIK
WHERE PREMIJA < LD;
```

Slika 9. Upit za izvršenje prethodnog primera

Naredbom SELECT se dobija sledeći izlaz (slika 10.):

S_RADNIK	S_RZ IME	POSLO	S_KODI DAT_TAP	ID PREMIJA
3199	30 BELAN	TRG POTNIK	3398 20-FEB-01	150000 30000
3221	30 PETAR	TRG POTNIK	3398 22-FEB-01	115000 50000
3544	30 GORAN	TRG POTNIK	3398 08-SEP-01	150000 0

Slika 10. Izlaz iz naredbe SELECT sa slike 9.

Primerom je prikazano korišćenje operatora < u WHERE klauzuli naredbe SELECT.

SELECT- KLAUZULA WHERE ZA SELEKCIJU SPECIFIČNIH N-TORKI

Upotreba operatora >= i <=

Primer: Prikaži podatke o radnicima koji su zaposleni posle 08.09.1981 .god., uključujući i taj datum

```
SELECT *  
FROM RADNIK  
WHERE DAT_ZAP >= '08-sep-81';
```

Slika 11. Upit za izvršenje prethodnog primera

Naredbom SELECT se dobija sledeći izlaz (slika 12.):

S_RADNIK	S_B3	IME	POSAD	S_PORO	DAT_ZAP	ID_PFERIJA	
3269	20	STEFAN	ANALITICAR	33602	17-DEC-00	80000	
3199	20	MILAN	TRG PUTNIK	3390	20-FEB-01	160000	30000
3221	30	PETAR	TRG PUTNIK	3390	22-FEB-01	125000	50000
3266	20	MARKO	ISKOVODILAC	3539	02-APR-01	297500	
3354	30	MARIJA	TRG PUTNIK	3390	20-SEP-01	125000	140000
3390	30	ANA	ISKOVODILAC	3539	01-MAY-01	285000	
3482	10	IVAN	ISKOVODILAC	3539	09-JUN-01	245000	
348	20	PAVLE	SAVETNIK	3266	09-NOV-01	300000	
3539	10	JOVAN	FRZ-SEKNIK		17-NOV-01	500000	
3544	30	GOVAN	TRG PUTNIK	3390	00-SEP-01	150000	
3576	20	TEKERA	ANALITICAR	3480	23-SEP-01	110000	
3600	30	JAKO	ANALITICAR	3390	03-DEC-01	95000	
3634	10	DEZAN	ANALITICAR	3482	23-JAN-02	130000	
3602	20	FILIP	SAVETNIK	3266	03-DEC-01	300000	

Slika 12. Izlaz iz naredbe SELECT sa slike 11.

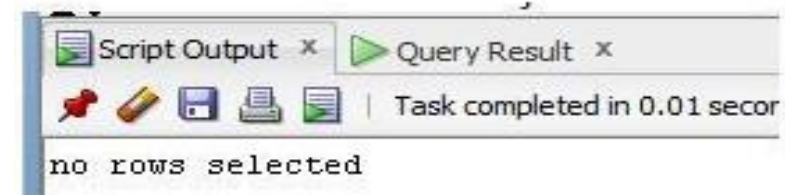
Primerom je prikazano korišćenje operatora >= u WHERE klauzuli naredbe SELECT.

Primer: Prikaži podatke o radnicima koji su zaposleni pre 08.09.1981 .god., uključujući i taj datum



Slika 13. Upit za izvršenje prethodnog primera

Naredbom SELECT se dobija sledeći izlaz (slika 14.):



Slika 14. Upit za izvršenje prethodnog primera

Primerom je prikazano korišćenje operatora <= u WHERE klauzuli naredbe SELECT. Izlaz iz ove SELECT naredbe pokazuje da nema podataka koji zadovoljavaju ovaj uslov

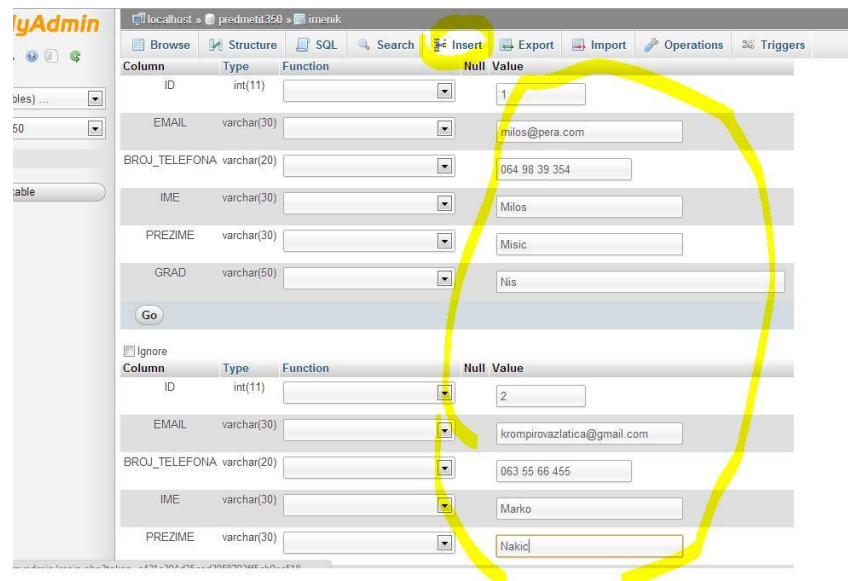
Vežba: Primeri kvalifikovanog pretraživanja

01

UNOS INICIJALNIH PODATAKA U PHPMYADMIN

Korišćenje naredbe INSERT za unos podataka u bazu kroz phpMyAdmin

Ukoliko želimo da unesemo više inicijalnih podataka u određenu tabelu to možemo izvršiti uz pomoć opcije **Insert** u phpMyAdmin alatu. Maksimalno po strani mogu se uneti podaci za 40 vrsti. Način na kome se unose podaci dat je na sledećoj slici:



Slika 1. Unos više inicijalnih podataka u bazu

PRIKAZ TABELE STUDENTSKE BAZE

Data je studentska baza i tabele DOSIJE, ISPIT, PREDMET I ISPITNI_ROK

	INDEKS	IME	PREZIME	GOD_RODJENJA	MESTO_RODJENJA
<input type="checkbox"/>	20100021	Milos	Peric	1992	Beograd
<input type="checkbox"/>	20100022	Marijana	Savkovic	1993	Kraljevo
<input type="checkbox"/>	20100023	Sanja	Terzic	1991	Beograd
<input type="checkbox"/>	20100024	Nikola	Vukovic	1992	
<input type="checkbox"/>	20100025	Ljubica	Savkovic	1991	Kraljevo
<input type="checkbox"/>	20100026	Zorica	Miladinovic	1993	Vranje
<input type="checkbox"/>	20100027	Milena	Stankovic	0	

Slika 2. Prikaz tabele DOSIJE

	indeks	id_predmeta	godina_rocka	oznaka_rocka	ocena	datum_ispita
<input type="checkbox"/>	20100021	1001	2011	jan	9	2020-01-20
<input type="checkbox"/>	20100021	1021	2011	apr	7	2003-04-20
<input type="checkbox"/>	20100021	2001	2011	jan	10	2025-01-20
<input type="checkbox"/>	20100021	3001	2011	jan	7	2027-01-20
<input type="checkbox"/>	20100022	1001	2011	jan	8	2020-01-20
<input type="checkbox"/>	20100022	1021	2011	apr	5	2003-04-20
<input type="checkbox"/>	20100022	2001	2011	jan	9	2025-01-20
<input type="checkbox"/>	20100023	1001	2011	jan	8	2020-01-20
<input type="checkbox"/>	20100023	1021	2011	apr	10	2003-04-20
<input type="checkbox"/>	20100023	2001	2011	jan	8	2025-01-20
<input type="checkbox"/>	20100023	3001	2011	jan	5	2027-01-20
<input type="checkbox"/>	20100024	1001	2011	jan	10	2020-01-20
<input type="checkbox"/>	20100024	1021	2011	apr	6	2003-04-20
<input type="checkbox"/>	20100024	2001	2011	jan	7	2025-01-20
<input type="checkbox"/>	20100024	3001	2011	jan	6	2028-01-20
<input type="checkbox"/>	20100025	1001	2011	jan	6	2020-01-20
<input type="checkbox"/>	20100025	2001	2011	feb	6	2010-02-20
<input type="checkbox"/>	20100025	2001	2011	jan	5	2025-01-20
<input type="checkbox"/>	20100026	1001	2011	feb	7	2010-02-20
<input type="checkbox"/>	20100026	1001	2011	jan	5	2020-01-20
<input type="checkbox"/>	20100026	1021	2011	apr	8	2003-04-20
<input type="checkbox"/>	20100026	2001	2011	feb	7	2010-02-20
<input type="checkbox"/>	20100026	3001	2011	jan	6	2028-01-20

Slika 3. Prikaz tabele ISPIT

PRIKAZ TABELE STUDENTSKE BAZE

Data je studentska baza i tabele DOSIJE, ISPIT, PREDMET I ISPITNI_ROK

```
SELECT * FROM `ispitni_rok`
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	godina_roka	oznaka_roka	naziv
<input type="checkbox"/> Edit Copy Delete	2011	apr	April 2011
<input type="checkbox"/> Edit Copy Delete	2011	feb	Februar 2011
<input type="checkbox"/> Edit Copy Delete	2011	jan	Januar 2011
<input type="checkbox"/> Edit Copy Delete	2011	jun	Jun 2011
<input type="checkbox"/> Edit Copy Delete	2011	okt	Oktober 2011
<input type="checkbox"/> Edit Copy Delete	2011	sep	Septembar 2011

Check All With selected: Change Delete Export

Number of rows: 25 Filter rows: Search this table

Slika 4. Prikaz tabele ISPITNI_ROK

```
SELECT * FROM `predmet`
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	id_predmeta	sifra	naziv	bodovi
<input type="checkbox"/> Edit Copy Delete	1001	M111	Matematika 1	6
<input type="checkbox"/> Edit Copy Delete	1002	M112	Matematika 2	6
<input type="checkbox"/> Edit Copy Delete	1003	M113	Statistika sa verovatnocom	6
<input type="checkbox"/> Edit Copy Delete	1021	M131	Racunarske mreze	6
<input type="checkbox"/> Edit Copy Delete	1101	M105	Diskretne strukture 1	6
<input type="checkbox"/> Edit Copy Delete	1102	M106	Diskretne strukture 2	6
<input type="checkbox"/> Edit Copy Delete	2001	P101	Programiranje 1	8
<input type="checkbox"/> Edit Copy Delete	2002	P102	Programiranje 2	8
<input type="checkbox"/> Edit Copy Delete	2003	P103	Objektno orijentisano programiranje	6
<input type="checkbox"/> Edit Copy Delete	2004	P104	Algoritmi i strukture podataka	6
<input type="checkbox"/> Edit Copy Delete	3001	S1	Engleski jezik 1	5
<input type="checkbox"/> Edit Copy Delete	3002	S2	Engleski jezik 2	5
<input type="checkbox"/> Edit Copy Delete	4001	R101	Uvod u organizaciju racunara	5
<input type="checkbox"/> Edit Copy Delete	4002	R102	Uvod u Veb i Internet tehnologije	5

Check All With selected: Change Delete Export

Number of rows: 25 Filter rows: Search this table

Slika 5. Prikaz tabele PREDMET

PRIMER UPITA BEZ WHERE NAREDBE

Vežbanje osnovnih SELECT upita

Primer 1.

Napisati upit koji kao rezultat daje podatke o svim studentima na fakultetu, od podataka izdvojiti indeks, ime i prezime za svakog studenta.

```
SELECT `INDEKS`, `IME`, `PREZIME` FROM DOSIJE
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	INDEKS	IME	PREZIME
<input type="checkbox"/>	20100021	Milos	Peric
<input type="checkbox"/>	20100022	Marijana	Savkovic
<input type="checkbox"/>	20100023	Sanja	Terzic
<input type="checkbox"/>	20100024	Nikola	Vukovic
<input type="checkbox"/>	20100025	Ljubica	Savkovic
<input type="checkbox"/>	20100026	Zorica	Miladinovic
<input type="checkbox"/>	20100027	Milena	Stankovic

Number of rows: 25 Filter rows: Search this table

Slika 6. Rezultat upita

Primer 2.

Napisati upit koji kao rezultat daje podatke o svim predmetima na fakultetu, od podataka izdvojiti id predmeta, naziv i broj bodova.

```
SELECT `id_predmeta`, `naziv`, `bodovi` FROM PREDMET
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	id_predmeta	naziv	bodovi
<input type="checkbox"/>	1001	Matematika 1	6
<input type="checkbox"/>	1002	Matematika 2	6
<input type="checkbox"/>	1003	Statistika sa verovatnocom	6
<input type="checkbox"/>	1021	Racunarske mreze	6
<input type="checkbox"/>	1101	Diskretne strukture 1	6
<input type="checkbox"/>	1102	Diskretne strukture 2	6
<input type="checkbox"/>	2001	Programiranje 1	8
<input type="checkbox"/>	2002	Programiranje 2	8
<input type="checkbox"/>	2003	Objektno orijentisano programiranje	6
<input type="checkbox"/>	2004	Algoritmi i strukture podataka	6
<input type="checkbox"/>	3001	Engleski jezik 1	5
<input type="checkbox"/>	3002	Engleski jezik 2	5
<input type="checkbox"/>	4001	Uvod u organizaciju racunara	5
<input type="checkbox"/>	4002	Uvod u Veb i Internet tehnologije	5

Number of rows: 25 Filter rows: Search this table

Slika 7. Prikaz rezultata

PRIMER UPITA SA WHERE NAREDBOM

Vežbanje *SELECT* upita sa *WHERE* naredbom

Primer 3.

Napisati upit koji kao rezultat daje podatke o studentima na fakultetu koji su rođeni u Beogradu.

```
SELECT * FROM `dosije` WHERE `MESTO_RODJENJA`="BEOGRAD"
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	INDEKS	IME	PREZIME	GOD_RODJENJA	MESTO_RODJENJA
<input type="checkbox"/>	20100021	Milos	Peric	1992	Beograd
<input type="checkbox"/>	20100023	Sanja	Terzic	1991	Beograd

Number of rows: 25 Filter rows: Search this table

Slika 8. Prikaz rezultata

Primer 4.

Napisati upit koji kao rezultat daje podatke o predmetima koji nosi izmedju 5 i 7 bodova, uključujući i njih.

```
SELECT * FROM `predmet` WHERE `bodovi`>=5 and `bodovi`<=7
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	id_predmeta	sifra	naziv	bodovi
<input type="checkbox"/>	1001	M111	Matematika 1	6
<input type="checkbox"/>	1002	M112	Matematika 2	6
<input type="checkbox"/>	1003	M113	Statistika sa verovatnocom	6
<input type="checkbox"/>	1021	M131	Racunarske mreze	6
<input type="checkbox"/>	1101	M105	Diskretne strukture 1	6
<input type="checkbox"/>	1102	M106	Diskretne strukture 2	6
<input type="checkbox"/>	2003	P103	Objektno orijentisano programiranje	6
<input type="checkbox"/>	2004	P104	Algoritmi i strukture podataka	6
<input type="checkbox"/>	3001	S1	Engleski jezik 1	5
<input type="checkbox"/>	3002	S2	Engleski jezik 2	5
<input type="checkbox"/>	4001	R101	Uvod u organizaciju racunara	5
<input type="checkbox"/>	4002	R102	Uvod u Veb i Internet tehnologije	5

Number of rows: 25 Filter rows: Search this table

Slika 9. Prikaz rezultata

Logički operatori u naredbi SELECT

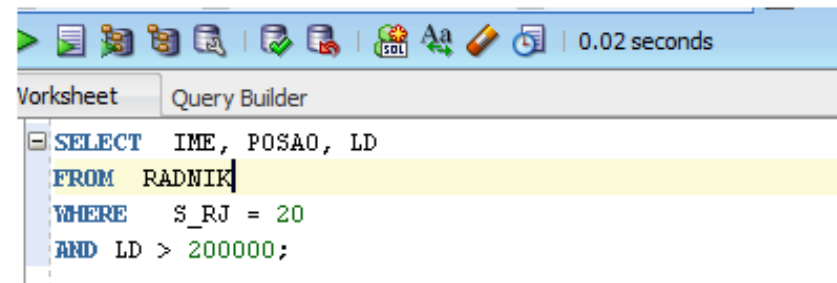
Vežba: *Logički operatori u naredbi SELECT*

02

SELEKCIJU N-TORKI KOJE ZADOVOLJAVAJU VIŠESTRUKU USLOVE

Upotreba operatora AND

Primer: Prikaži ime, posao i LD svakog radnika iz RJ 20 koji zarađuje više od 200000 din (slika 1.)



```
SELECT IME, POSAO, LD
FROM RADNIK
WHERE S_RJ = 20
AND LD > 200000;
```

Slika 1. Upit za izvršenje prethodnog primera

Naredbom SELECT se dobija sledeći izlaz (slika 2.):

IME	POSALO	LD
MARKO	RUKOVODILAC	297500
PAVLE	SAVETNIK	300000
FILIP	SAVETNIK	300000

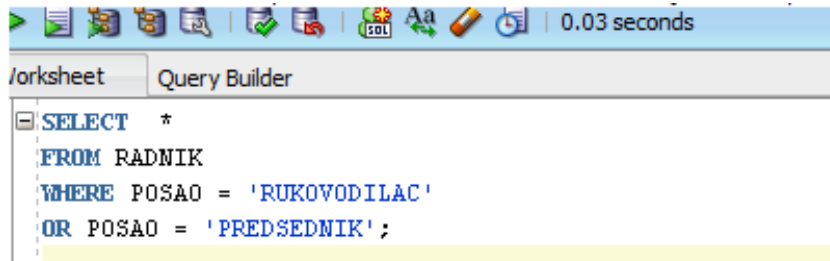
Slika 2. Izlaz iz naredbe SELECT sa slike 1.

Primerom je prikazano definisanje uslova uz korišćenje logičkog operatora AND (istovremeno se zadovoljava više uslova).

SELEKCIJU N-TORKI KOJE ZADOVOLJAVAJU BAR JEDAN OD VIŠE USLOVA

Operator OR

Primer: Prikazati sve podatke o rukovodiocima radnih jedinica i predsedniku (slika 3.)

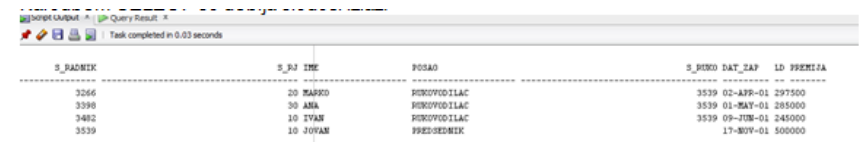


The screenshot shows a software interface with a toolbar at the top containing icons for execution, refresh, and other functions, along with a timer showing '0.03 seconds'. Below the toolbar is a tab labeled 'Worksheet' and a sub-tab 'Query Builder'. The main area contains a SQL query:

```
SELECT *  
FROM RADNIK  
WHERE POSAO = 'RUKOVODILAC'  
OR POSAO = 'PREDSEDNIK';
```

Slika 3. Upit za izvršenje prethodnog primera

Naredbom SELECT se dobija sledeći izlaz (slika 4.):



The screenshot shows a 'Query Result' window with a table of data. The table has five columns: S_RADNIK, S_PJ, IME, POSAO, and S_MERIDAT_ZAP. The data is as follows:

S_RADNIK	S_PJ	IME	POSAD	S_MERIDAT_ZAP	ID_PRENJIA
3266	20	MARCO	RUKOVODILAC	3539 02-APR-01	297500
3398	30	ANA	RUKOVODILAC	3539 01-MAY-01	285000
3482	10	IVAN	RUKOVODILAC	3539 09-JUN-01	245000
3539	10	JUVAN	PREDSEDNIK	17-NOV-01	500000

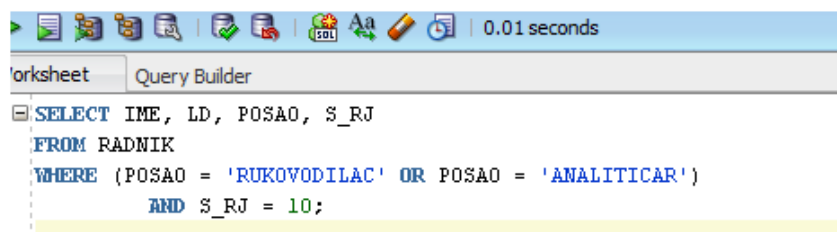
Slika 4. Izlaz iz naredbe SELECT sa slike 3.

Primerom je prikazano definisanje uslova uz korišćenje logičkog operatora OR (zadovoljava se bar jedan od više uslova).

SELEKCIJU N-TORKI ISTOVREMENIM KORIŠĆENJEM VIŠE LOGIČKIH OPERATORA

Korišćenje AND i OR logičkih operatora

Primer: Prikaži ime, LD, posao i šifru PS-a za rukovodioca i analitičare u RJ -u 10 (korišćenje zagrada da bi se definisao redosled ispitivanja uslova kod istovremene primene AND i OR logičkih operatora), slika 5.



```
SELECT IME, LD, POSAO, S_RJ
FROM RADNIK
WHERE (POSAO = 'RUKOVODILAC' OR POSAO = 'ANALITICAR')
AND S_RJ = 10;
```

Slika 5. Upit za izvršenje prethodnog primera

Naredbom SELECT se dobija sledeći izlaz (slika 6):

IME	LD POSAO	S_RJ
IVAN	245000 RUKOVODILAC	10
DEJAN	130000 ANALITICAR	10

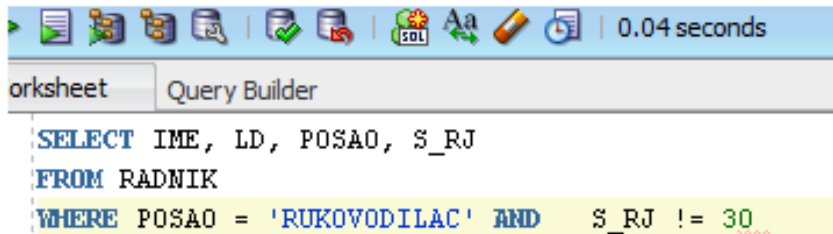
Slika 6. Izlaz iz naredbe SELECT sa slike 5.

Primerom je prikazano istovremeno korišćenje operatora AND i OR.

SELEKCIJU N-TORKI KOJE NE ZADOVOLJAVAJU ODREĐENE USLOVE

Operator NOT

Primer: Prikaži ime, LD, posao i šifru radne jedinice za rukovodioce koji ne rade u radnoj jedinici 30 (korišćenje operatora poređenja "!=" ili logičkog operatora NOT) (slika 7.)



The screenshot shows a software interface with a toolbar at the top containing various icons and a timer showing '0.04 seconds'. Below the toolbar is a tab labeled 'orksheet' and a 'Query Builder' window. The query text is as follows:

```
SELECT IME, LD, POSAO, S_RJ
FROM RADNIK
WHERE POSAO = 'RUKOVODILAC' AND S_RJ != 30
```

Slika 7. Upit za izvršenje prethodnog primera

Naredbom SELECT se dobija sledeći izlaz (slika 8.):

IME	LD POSAO	S_RJ
MARKO	297500 RUKOVODILAC	20
IVAN	245000 RUKOVODILAC	10

Slika 8. Izlaz iz naredbe SELECT sa slike 7.

Primerom je prikazano korišćenje operatora !=.

Vežba: Logički operatori u naredbi SELECT

02

PRIMER UPITA UZ KORIŠĆENJE LOGIČKOG OPERATORA AND

Vežbanje korišćenja logičkih operatora

Primer 1.

Napisati upit koji kao rezultat daje informacije o studentima na fakultetu koji su rođeni posle 1992. godine u Kraljevu

```
SELECT `IME`, `PREZIME` FROM `dosije` WHERE `GOD_RODJENJA`>1992 AND `MESTO_RODJENJA`="Kraljevo"
```

Number of rows: 25 Filter rows: Search this table

+ Options

IME	PREZIME
Marijana	Savkovic

Slika 1. Prikaz rezultata

Primer 2.

Napisati upit kao rezultat daje podatke o položenim ispitima u januarском roku.

```
SELECT * FROM `ispit` WHERE ocena>5 AND oznaka_roka="jan"
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	indeks	id_predmeta	godina_roka	oznaka_roka	ocena	datum_ispita
<input type="checkbox"/>	Edit Copy Delete	20100021	1001	2011 jan	9	2020-01-20
<input type="checkbox"/>	Edit Copy Delete	20100021	2001	2011 jan	10	2025-01-20
<input type="checkbox"/>	Edit Copy Delete	20100021	3001	2011 jan	7	2027-01-20
<input type="checkbox"/>	Edit Copy Delete	20100022	1001	2011 jan	8	2020-01-20
<input type="checkbox"/>	Edit Copy Delete	20100022	2001	2011 jan	9	2025-01-20
<input type="checkbox"/>	Edit Copy Delete	20100023	1001	2011 jan	8	2020-01-20
<input type="checkbox"/>	Edit Copy Delete	20100023	2001	2011 jan	8	2025-01-20
<input type="checkbox"/>	Edit Copy Delete	20100024	1001	2011 jan	10	2020-01-20
<input type="checkbox"/>	Edit Copy Delete	20100024	2001	2011 jan	7	2025-01-20
<input type="checkbox"/>	Edit Copy Delete	20100024	3001	2011 jan	6	2028-01-20
<input type="checkbox"/>	Edit Copy Delete	20100025	1001	2011 jan	6	2020-01-20
<input type="checkbox"/>	Edit Copy Delete	20100026	3001	2011 jan	6	2028-01-20

↑ Check All With selected: Change Delete Export

Number of rows: 25 Filter rows: Search this table

Slika 2. Prikaz rezultata

PRIMER UPITA UZ KORIŠĆENJE LOGIČKOG OPERATORA OR

Vežbanje korišćenja logičkih operatora

Primer 3.

Napisati upit koji kao rezultat daje podatke o studentima koji se zovu Milos ili su iz Vranja.

```
SELECT * FROM `dosije` WHERE `IME`="Milos" OR `MESTO_RODJENJA`="Kraljevo"
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	INDEKS	IME	PREZIME	GOD_RODJENJA	MESTO_RODJENJA
<input type="checkbox"/>	20100021	Milos	Peric	1992	Beograd
<input type="checkbox"/>	20100022	Marijana	Savkovic	1993	Kraljevo
<input type="checkbox"/>	20100025	Ljubica	Savkovic	1991	Kraljevo

Number of rows: 25 Filter rows: Search this table

Slika 3. Prikaz rezultata

Primer 4.

Napisati upit koji kao rezultat daje podatke o polaganju ispita koji imaju id 1001 ili ih je polagao student sa brojem indeksa 20100021.

```
SELECT * FROM `ispit` WHERE `indeks`= 20100021 OR `id_predmeta`=1001
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	indeks	id_predmeta	godina_roka	oznaka_roka	ocena	datum_ispita
<input type="checkbox"/>	20100021	1001	2011	jan	9	2020-01-20
<input type="checkbox"/>	20100021	1021	2011	apr	7	2003-04-20
<input type="checkbox"/>	20100021	2001	2011	jan	10	2025-01-20
<input type="checkbox"/>	20100021	3001	2011	jan	7	2027-01-20
<input type="checkbox"/>	20100022	1001	2011	jan	8	2020-01-20
<input type="checkbox"/>	20100023	1001	2011	jan	8	2020-01-20
<input type="checkbox"/>	20100024	1001	2011	jan	10	2020-01-20
<input type="checkbox"/>	20100025	1001	2011	jan	6	2020-01-20
<input type="checkbox"/>	20100026	1001	2011	feb	7	2010-02-20
<input type="checkbox"/>	20100026	1001	2011	jan	5	2020-01-20

Number of rows: 25 Filter rows: Search this table

Slika 4. Prikaz rezultata

PRIMER UPITA UZ KORIŠĆENJE OPERATORA !=

Vežbanje korišćenja operatora !=

Primer 5.

Napisati upit koji prikazuje id_predmeta, naziv i broj bodova za predmete kojima je broj poena različit od 6.

```
SELECT `id_predmeta`, `naziv`, `bodovi` FROM `predmet` WHERE bodovi!=6
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	id_predmeta	naziv	bodovi
<input type="checkbox"/> Edit Copy Delete	2001	Programiranje 1	8
<input type="checkbox"/> Edit Copy Delete	2002	Programiranje 2	8
<input type="checkbox"/> Edit Copy Delete	3001	Engleski jezik 1	5
<input type="checkbox"/> Edit Copy Delete	3002	Engleski jezik 2	5
<input type="checkbox"/> Edit Copy Delete	4001	Uvod u organizaciju racunara	5
<input type="checkbox"/> Edit Copy Delete	4002	Uvod u Veb i Internet tehnologije	5

Check All With selected: Change Delete Export

Number of rows: 25 Filter rows: Search this table

Slika 5. Prikaz rezultata

Izrazi nad nizom karaktera u naredbi SELECT

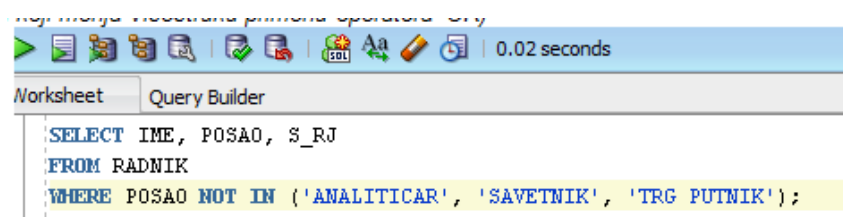
□ *Vežba: Izrazi nad nizom karaktera u naredbi SELECT*

03

SELEKCIJU N-TORKI KOJE ZADOVOLJAVAJU VREDNOST IZ LISTE VREDNOSTI

Operator in

Primer: Prikaži ime, posao i #PS radnika koji nisu analitičari, savetnici ili trgovački putnici (korišćenje operatora IN koji menja višestruku primenu operatora OR), slika 1.



```
SELECT IME, POSAO, S_RJ
FROM RADNIK
WHERE POSAO NOT IN ('ANALITICAR', 'SAVETNIK', 'TRG PUTNIK');
```

Slika 1. Upit za izvršenje prethodnog primera

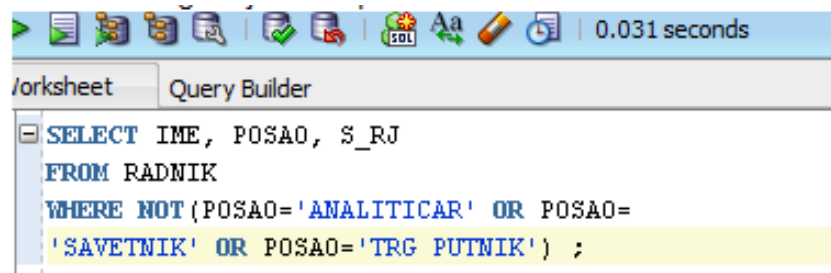
Naredbom SELECT se dobija sledeći izlaz (slika 2.):

IME	POSAO	S_RJ
MARKO	RUKOVODILAC	20
ANA	RUKOVODILAC	30
IVAN	RUKOVODILAC	10
JOVAN	PREDSEDNIK	10

Slika 2. Izlaz iz naredbe SELECT sa slike 1.

Primerom je prikazano korišćenje operatora NOT IN.

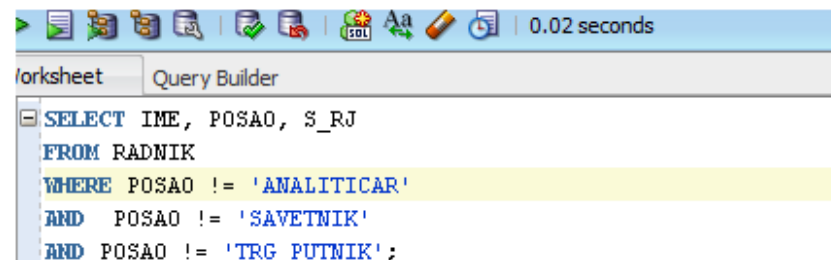
Isti uslov moguće je konceptualno iskazati i na sledeći način (slika 3.):



```
SELECT IME, POSAO, S_RJ
FROM RADNIK
WHERE NOT (POSAO='ANALITICAR' OR POSAO='SAVETNIK' OR POSAO='TRG PUTNIK');
```

Slika 3. Drugi način za upotrebu operatora IN

odnosno realizovati sledećom WHERE klauzulom:



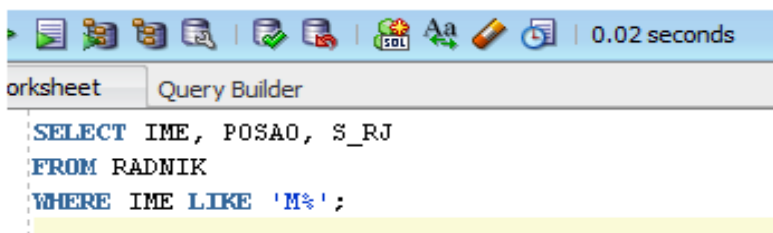
```
SELECT IME, POSAO, S_RJ
FROM RADNIK
WHERE POSAO != 'ANALITICAR'
AND POSAO != 'SAVETNIK'
AND POSAO != 'TRG PUTNIK';
```

Slika 4. Drugi način za upotrebu operatora IN

SELEKCIJU N-TORKI KOJE SADRŽE ODREĐENU KOMBINACIJU KARAKTERA

Operator like uz korišćenje specijalnog karaktera "%"

Primer: Prikazati ime, posao i šifru RJ radnika čija imena počinju sa M. (slika 5.)



```
SELECT IME, POSAO, S_RJ
FROM RADNIK
WHERE IME LIKE 'M%';
```

Slika 5. Upit za izvršenje prethodnog primera

Naredbom SELECT se dobija sledeći izlaz (slika 6.):

IME	POSAO	S_RJ
MILAN	TRG PUTNIK	30
MARKO	RUKOVODILAC	20
MARIJA	TRG PUTNIK	30

Slika 6. Izlaz iz naredbe SELECT sa slike 5.

Primerom je prikazano korišćenje operatora LIKE.

Klauzula LIKE omogućuje pretraživanje na osnovu "UZORKA" odnosno dobijanje informacija i kada ne znamo potpun naziv (tj. vrednost) određenog atributa tipa *character*. Ona koristi dva specijalna karaktera ("%","_") sa sledećim značenjem:

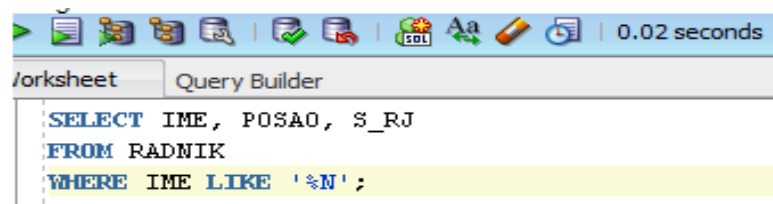
"%" predstavlja string od 0 ili više karaktera

"_" predstavlja poziciju jednog karaktera.

Ostali karakteri imaju uobičajeno značenje. Uslov u WHERE klauzuli navedenog upita kaže da IME treba da liči na uzorak naveden u jednostrukim navodnicima

Primeri:

... gde se ime završava sa N (slika 7.)



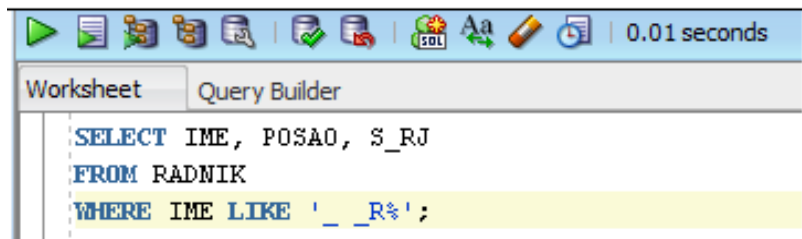
```
SELECT IME, POSAO, S_RJ
FROM RADNIK
WHERE IME LIKE '%N';
```

Slika 7. Upit za izvršenje prethodnog primera

SELEKCIJU N-TORKI KOJE SADRŽE ODREĐENU KOMBINACIJU KARAKTERA

Operator like uz korišćenje specijalnog karaktera „_”

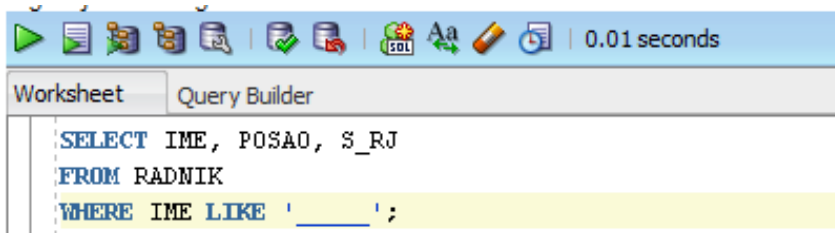
.. gde je treći karakter imena R (slika 8.).



```
SELECT IME, POSAO, S_RJ
FROM RADNIK
WHERE IME LIKE '_R%';
```

Slika 8. Upit za izvršenje prethodnog primera

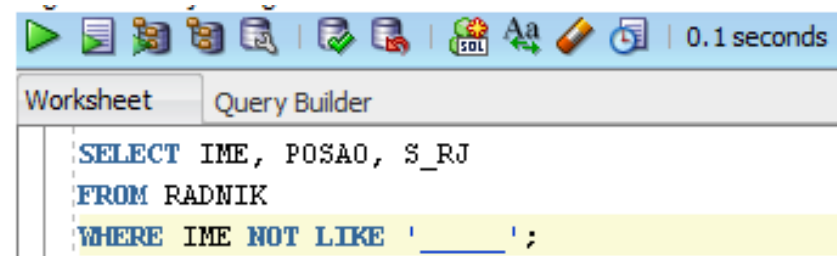
... gde je ime dugačko 5 karaktera (slika 9.)



```
SELECT IME, POSAO, S_RJ
FROM RADNIK
WHERE IME LIKE '_____';
```

Slika 9. Upit za izvršenje prethodnog primera

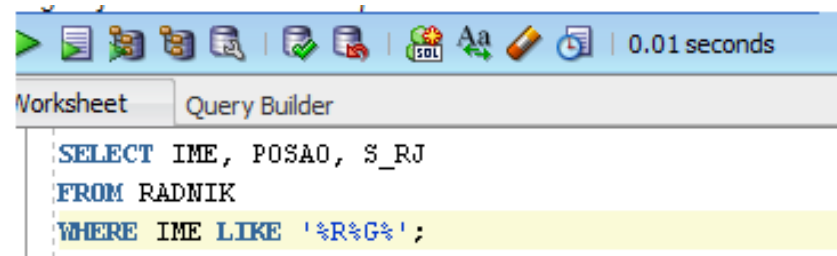
.. gde ime nije dugačko 5 karaktera (slika 10.).



```
SELECT IME, POSAO, S_RJ
FROM RADNIK
WHERE IME NOT LIKE '_____';
```

Slika 10. Upit za izvršenje prethodnog primera

.. gde je u imenu slovo G posle R (slika 11.).



```
SELECT IME, POSAO, S_RJ
FROM RADNIK
WHERE IME LIKE '%R%G%';
```

Slika 11. Upit za izvršenje prethodnog primera

NAJČEŠĆE FUNKCIJE NAD KOLONAMA TIPa CHARACTER

||, LENGTH (str), SUBSTR (str, spos, [, len]), INSTR (str, sstr [, spos]), UPPER (str), LOWER (str), TO_NUM (str), TO_CHAR (str), LPAD (str, len [, char]), RPAD (str, len [char])...

Najčešće funkcije nad kolonama tipa *character* koje se primenjuju su sledeće:

- a. **String1 || string2** – spaja stringove karaktera
- b. **LENGTH (str)** – nalazi dužinu stringa
- c. **SUBSTR (str, spos, [, len])**

– daje podstring od "len" karaktera, polazeći od pozicije "spos,,

Mogu se primeniti i funkcije:

- a. koje omogućuju kontrolu vrednosti atributa do nivoa pojedinačnog karaktera:

INSTR (str, sstr [, spos]) – traži podstring "sstr" u stringu "str" polazeći od pozicije "spos". Ako je nađen, vraća se njegova pozicija, inače 0.

UPPER (str) – menja sva mala slova u velika

- b. koje omogućuju očuvanje jednoobraznosti baze podataka:

LOWER (str) – menja sva velika slova u mala

TO_NUM (str) – pretvara niz karaktera (numeričkih) u broj

TO_CHAR (str) – pretvara broj u niz karaktera

LPAD (str, len [, char]) – popunjava levu stranu stringa "str" sa karakterom "char" u dužini od "len,,

- c. koje su pogodne za grafički prikaz relativnih odnosa rezultujućih vrednosti:

RPAD (str, len [char]) – popunjava desnu stranu stringa "str" sa karakterom "char" u dužini od "len"

NVL (str1, str2) – ako je str1 NULL vraća str2, inače vraća str1

DECODE (str, cs1, rst1, .., dft) – rezultat je rst, ako je str=cs.

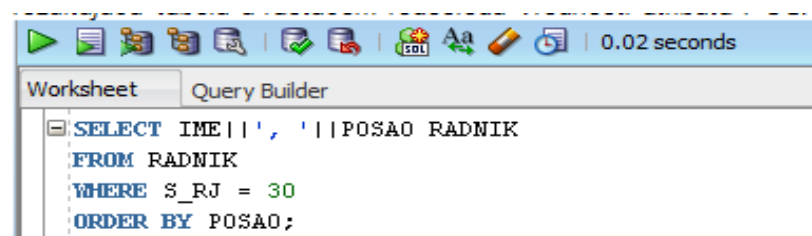
Poslednji argument je *default* vrednost.

Sve ove funkcije navode se iza SELECT klauzule.

PRIMER FUNKCIJA NAD KOLONAMA TIPA CHARACTER

Upotreba funkcije || i LENGTH (str)

Primer: Prikaži imena radnika RJ 30 iza kojih neposredno treba da dođe posao koji obavljaju. Sortirati rezultujuću tabelu u rastućem redosledu vrednosti atributa POSAO (slika 12.).

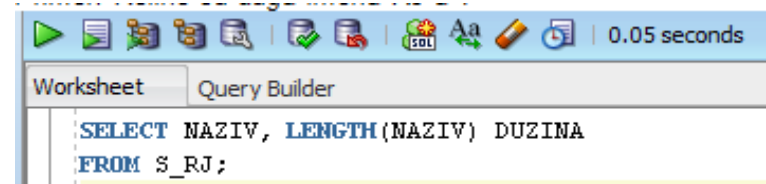


```
SELECT IME||', '||POSAO RADNIK
FROM RADNIK
WHERE S_RJ = 30
ORDER BY POSAO;
```

Slika 12. Upit za izvršenje prethodnog primera

Primer prikazuje spajanje tri stringa pri čemu se spojeni string naziva RADNIK. Prvi string je atribut IME, drugi string se sastoji od dva karaktera – zarez i jednog praznog mesta i treći string je atribut POSAO.

Primer: Koliko su duga imena RJ-a ? (slika 14.)



```
SELECT NAZIV, LENGTH(NAZIV) DUZINA
FROM S_RJ;
```

Slika 14. Upit za izvršenje prethodnog primera

Izlaz iz naredbe SELECT glasi (slika 13.):

RADNIK
JANKO, ANALITICAR
ANA, RUKOVODILAC
GORAN, TRG PUTNIK
PETAR, TRG PUTNIK
MILAN, TRG PUTNIK
MARIJA, TRG PUTNIK

Slika 13. Izlaz iz naredbe SELECT sa slike 12.

Izlaz iz naredbe SELECT glasi (slika 15.):

NAZIV	DUZINA
PRODAJA	7
PROIZVODNJA	11
PROJEKTOVANJE	13
ISTRAZIVANJE	12

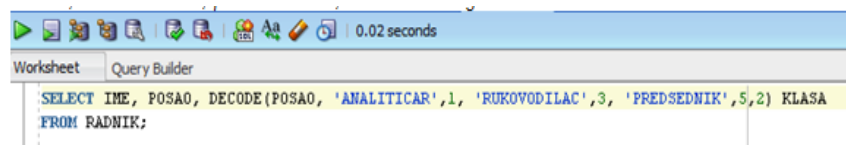
Slika 15. Izlaz iz naredbe SELECT sa slike 14.

U primeru je prikazano korišćenje funkcije LENGHT nad nizom karaktera.

PRIMER FUNKCIJA NAD KOLONAMA TIPAA CHARACTER

Upotreba funkcije DECODE (str, cs1, rst1, .., dft)

Primer: Koristeći kolonu POSAO formiraj kolonu KLASA tako što za posao analitičara vrednost klase treba da bude 1, rukovodioca 3, predsednika 5, a za svaki drugi 2 (slika 16.).



```
SELECT IME, POSAO, DECODE(POSAO, 'ANALITICAR',1, 'RUKOVODILAC',3, 'PRESEDNIK',5,2) KLASA
FROM RADNIK;
```

Slika 16. Upit za izvršenje prethodnog primera

Izlaz in naredbe SELECT glasi (slika 17.):

IME	POSAO	KLASA
STEFAN	ANALITICAR	1
MILAN	TRG PUTNIK	2
PETAR	TRG PUTNIK	2
MARKO	RUKOVODILAC	3
MARIJA	TRG PUTNIK	2
ANA	RUKOVODILAC	3
IVAN	RUKOVODILAC	3
PAVLE	SAVETNIK	2
JOVAN	PRESEDNIK	5
GORAN	TRG PUTNIK	2
JELENA	ANALITICAR	1
JANKO	ANALITICAR	1
DEJAN	ANALITICAR	1
FILIP	SAVETNIK	2

Slika 17. Izlaz iz naredbe SELECT sa slike 16.

Vežba: Izrazi nad nizom karaktera u naredbi SELECT

03

PRIMER KORIŠĆENJA OPERATORA LIKE

Vežbanje operatora like

Primer 1.

Napisati upit koji kao rezultat daje podatke o studentima koji imaju inicijale M.S.

```
SELECT indeks, ime, prezime FROM `dosije` WHERE ime like 'M%' and prezime like 'S%'
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

indeks	ime	prezime
20100022	Marijana	Savkovic
20100027	Milena	Stankovic

Number of rows: 25 Filter rows: Search this table

Slika 1. Prikaz rezultata

Primer 2.

Prikazati podatke o predmetima koji u svom nazivu imaju cifru 1.

```
SELECT * FROM `predmet` WHERE `naziv` like '%1'
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	id_predmeta	sifra	naziv	bodovi
<input type="checkbox"/> Edit Copy Delete	1001	M111	Matematika 1	6
<input type="checkbox"/> Edit Copy Delete	1101	M105	Diskretne strukture 1	6
<input type="checkbox"/> Edit Copy Delete	2001	P101	Programiranje 1	8
<input type="checkbox"/> Edit Copy Delete	3001	S1	Engleski jezik 1	5

Check All With selected: Change Delete Export

Slika 2. Prikaz rezultata

PRIMERI KORIŠĆENJA OPERATORA LIKE

Vežbanje operatora like

Primer 3.

Prikazati podatke o predmetima koji u svom nazivu imaju reč „jezik“.

```
SELECT * FROM `predmet` WHERE `naziv` like "% jezik %"
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

		id_predmeta	sifra	naziv	bodovi
<input type="checkbox"/>	Edit Copy Delete	3001	S1	Engleski jezik 1	5
<input type="checkbox"/>	Edit Copy Delete	3002	S2	Engleski jezik 2	5

Check All With selected: Change Delete Export

Number of rows: 25 Filter rows: Search this table

Slika 3. Prikaz rezultata

Aritmetičke i sumarne funkcije u naredbi SELECT

□ *Vežba: Aritmetičke i sumarne funkcije u naredbi
SELECT*

04

ARITMETIČKE FUNKCIJE

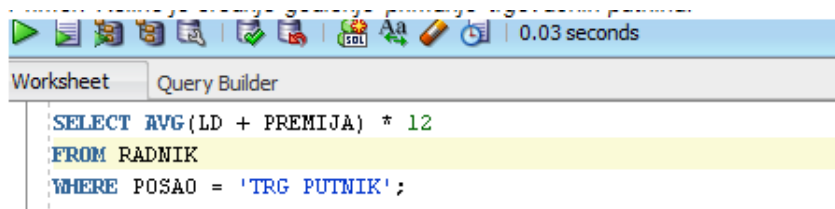
Aritmetički operatori "+", "", "-", "/", POWER (broj, e), ROUND (broj [,d]), TRUNC (broj [,d]), SIGN (broj)....*

Aritmetički izrazi se koriste za izvršenje bilo koje aritmetičke operacije u upitu. Sastavljeni su od imena kolona i konstantnih vrednosti povezanih **aritmetičkim operatorima** ("+", "*", "-", "/"). U aritmetičkim izrazima se mogu koristiti i grupne funkcije.

Koristi se sledeća sintaksa:

```
SELECT numerički_izraz as IME OPERACIJE;
```

Primer: Koliko je srednje godišnje primanje trgovačkih putnika (slika 1.).



```
SELECT AVG(LD + PREMIJA) * 12
FROM RADNIK
WHERE POSAO = 'TRG PUTNIK';
```

Slika 1. Upit za izvršenje prethodnog primera

Iz ove SELECT naredbe se dobija sledeći izlaz (slika 2.):

```
AVG(LD+PREMIJA)*12
-----
                2340000
```

Slika 2. Izlaz iz naredbe SELECT sa slike 1.

SQL podržava i sledeće aritmetičke funkcije:

POWER (broj, e) – diže broj na e-ti stepen

ROUND (broj [,d]) – zaokružuje broj na d decimala

TRUNC (broj [,d]) – odbacuje ostatak od d-tog decimalnog mesta

ABS (broj) – nalazi apsolutnu vrednost broja

SIGN (broj) – daje +1 ako je broj >0, 0 ako je broj = 0, -1 ako je broj <0.

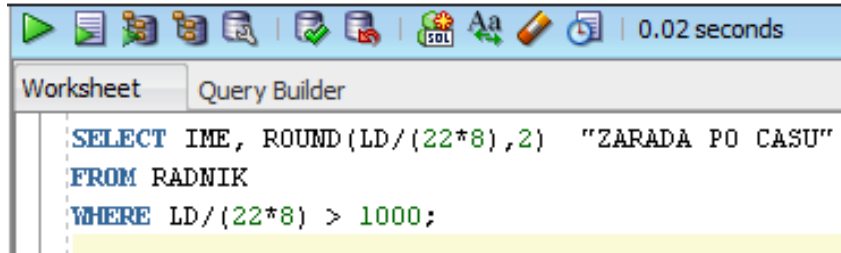
MOD (broj1, broj2) – izračunava broj1 po modulu broj2

SORT (broj) – nalazi pozitivan kvadratni koren broja

ARITMETIČKE FUNKCIJE

Aritmetički operator ROUND (broj [,d])

Primer: Koji radnici zarađuju više od 1000 dinara po satu. Zaradu po satu zaokružiti na 2 decimale. (Podrazumeva se da postoje 22 radna dana u mesecu i 8 radnih sati u danu), slika 3.



Slika 3. Upit za izvršenje prethodnog primera

Iz SELECT naredbe se dobija sledeći izlaz (slika 3.):

```
-----  
MARKO                1690.34  
ANA                   1619.32  
IVAN                  1392.05  
PAVLE                 1704.55  
JOVAN                 2840.91  
FILIP                 1704.55
```

Slika 4. Izlaz iz naredbe SELECT sa slike 3.

Iz primera se vidi još jedan primer korišćenja aritmetičkog izraza u naredbi SELECT.

Vidi se da je zaglavlje rezultujuće kolone ROUND(LD/(22*8),2) privremeno zamenjeno izrazom pod duplim navodnicima "ZARADA PO ČASU". Rezultujuću kolonu i izraz pod duplim navodnicima razdvaja jedno prazno mesto (blanko). Ukoliko se izraz sastoji samo od jedne reči nije ga potrebno stavljati pod navodnike.

FUNKCIJA ZA DOBIJANJE SUMARNIH INFORMACIJA

AVG (atribut), SUM (atribut), MIN (atribut), MAX (atribut), COUNT (*)

Funkciju za dobijanje sumarnih informacija su:

AVG (atribut) – izračunava srednju vrednost

SUM (atribut) – izračunava ukupnu vrednost

MIN (atribut) – nalazi minimalnu vrednost

MAX (atribut) – nalazi maksimalnu vrednost

Ove funkcije se definišu nad numeričkim kolonama.

Funkcija **COUNT** definisana je nad kolonama bilo kog tipa. Ona ima tri oblika:

COUNT (*) – nalazi broj n-torki u grupi

COUNT (atribut) – nalazi NOT-NULL vrednosti kolone

COUNT (DISTINCT atribut) – nalazi različite NOT-NULL vrednosti kolone

Sve se ove COUNT funkcije navode iza SELECT klauzule i prouzrokuju sažimanje sadržaja kolona nad kojima se primenjuju.

Primer: Aritmetičke i sumarne funkcije u naredbi SELECT

04

PRIMERI KORIŠĆENJA SUMARNIH FUNKCIJA

Vežbanje korišćenja sumarnih funkcija

Primer 1.

Prebrojati koliko se predmeta na Fakultetu predaje.

```
SELECT count(*) FROM `predmet`
```

+ Options
count(*)
14

Slika 1. Prikaz rezultata

Primer 2.

Pronaći najmanji i najveći broj indeksa studenata upisanih na Fakultet.

```
SELECT min(indeks), max(indeks) FROM dosije
```

Show all Number of rows: 25 ▼ Fi

Sort by key: None ▼

+ Options

min(indeks)	max(indeks)
20100021	20100027

Slika 2. Prikaz rezultata

PRIMER KORIŠĆENJA FUNKCIJE AVG

Vežbanje upita sa agregatnim funkcijama

Primer 3.

Napisati upit koji izračunava prosečnu ocenu na premetu sa id-jem 2001 u januarском roku 2011.

```
SELECT avg(ocena) from ispit where `id_predmeta`=2001 and `oznaka_rock`="jan" and `godina_rock`=2011 and ocena>5
```

Number of rows: 25 ▾ Filter rows:

+ Options

avg(ocena)
8.5000

Slika 3. Prikaz rezultata

Klauzule GROUP BY, HAVING, ORDER BY

□ *Primeri: Klauzule GROUP BY , HAVING*

□ *Vežba: Klauzule GROUP BY , HAVING i ORDER BY*

05

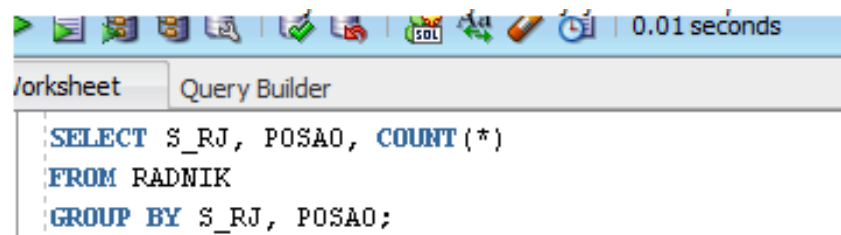
KLAUZULA GROUP BY

Grupisanje po više kolona

Grupisanje se može vršiti po više kolona. **Ukoliko se grupisanje vrši po N kolona, tada svaka različita n-torka čini grupu.**

Iz ovog primera se vidi da je dejstvo GROUP BY klauzule identično višestrukom pisanju SELECT naredbi sa različitim uslovima u WHERE klauzuli.

Primer: Izračunati broj radnika koji obavljaju različiti posao unutar svake RJ (slika 3.)



```
SELECT S_RJ, POSAO, COUNT(*)
FROM RADNIK
GROUP BY S_RJ, POSAO;
```

Slika 3. Upit za izvršenje prethodnog primera

Iz ove SELECT naredbe se dobija sledeći izlaz (slika 4.):

S_RJ	POSAO	COUNT(*)
10	RUKOVODILAC	1
30	TRG PUTNIK	4
20	RUKOVODILAC	1
10	PREDSEDNIK	1
20	ANALITICAR	2
30	RUKOVODILAC	1
20	SAVETNIK	2
30	ANALITICAR	1
10	ANALITICAR	1

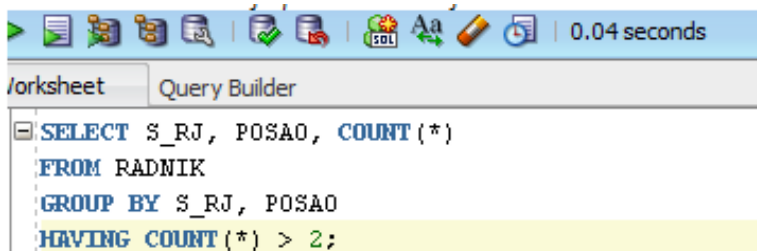
Slika 4. Izlaz iz naredbe SELECT sa slike 3.

Primer prikazuje još jedno korišćenje klauzule GROUP BY.

KLAUZULA HAVING

Klauzula HAVING određuje kriterijume za selekciju grupa pošto su grupe već formirane sa GROUP BY klauzulom.

Primer: Prikazati koje poslove obavlja više od 2 radnika u svakoj RJ (slika 5.).



```
SELECT S_RJ, POSAO, COUNT (*)  
FROM RADNIK  
GROUP BY S_RJ, POSAO  
HAVING COUNT (*) > 2;
```

Slika 5. Upit za izvršenje prethodnog primera

Iz ove SELECT naredbe se dobija sledeći izlaz (slika 6.):

S_RJ	POSAD	COUNT (*)
30	TRG PUTNIK	4

Slika 6. Izlaz iz naredbe SELECT sa slike 5.

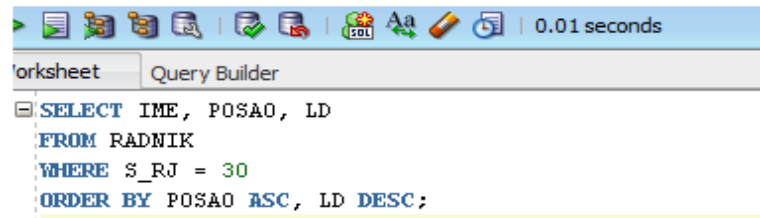
KLAUZULA ORDER BY

Rezultujuću tabelu je moguće sortirati po jednom ili više atributa u rastućem ili opadajućem redosledu.

Korišćenjem klauzule **ORDER BY** je rezultujuću tabelu moguće sortirati po jednom ili više atributa u rastućem ili opadajućem redosledu.

Za specifikaciju rastućeg redosleda koristi se klauzula **ASC**, a za specifikaciju opadajućeg redosleda klauzula **DESC**. Rastući redosled se podrazumeva, pa klauzulu **ASC** nije neophodno navoditi, za razliku od klauzule **DESC** koju uvek treba navesti kada se sortira u opadajućem redosledu. **ORDER BY** je uvek **poslednja** klauzula u **SELECT** bloku.

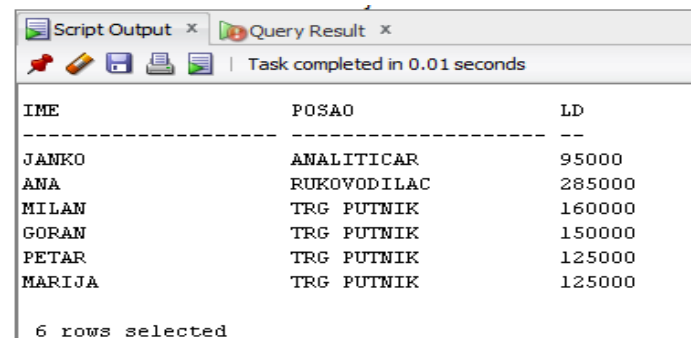
Primer: Prikaži ime, posao i LD radnika u PS-u 30 uređene u rastućem redosledu poslova i opadajućem redosledu ličnih dohodaka (slika 7.).



```
SELECT IME, POSAO, LD
FROM RADNIK
WHERE S_RJ = 30
ORDER BY POSAO ASC, LD DESC;
```

Slika 7. Upit za izvršenje prethodnog primera

Iz ove **SELECT** naredbe se dobija sledeći izlaz (slika 8.):



IME	POSAO	LD
JANKO	ANALITICAR	95000
ANA	RUKOVODILAC	285000
MILAN	TRG PUTNIK	160000
GORAN	TRG PUTNIK	150000
PETAR	TRG PUTNIK	125000
MARIJA	TRG PUTNIK	125000

6 rows selected

Slika 8. Izlaz iz naredbe **SELECT** sa slike 7.

U ovom primeru su poslovi radnika poređani po rastućem a plata radnika po opadajućem redosledu.

Kada se sortiranje vrši po koloni sa **NULL** vrednostima, n-torke sa **NULL** vrednostima su uvek na početku rezultujuće tabele, bez obzira da li je sortiranje u rastućem ili opadajućem redosledu.

PRIMERI: Klauzule GROUP BY, HAVING,
ORDER BY

05

PRIMER 1

Upotreba klauzula GROUP BY, HAVING i ORDER BY

Kreirati izveštaj sa kolonama **Ime i Ukupno** (gde je ukupno suma proizvoda količine i minimalne cene) na osnovu tabele **PROIZVOD** (S_PROIZVODA, IME_PROIZVODA, KOLICINA, MIN_CENA) koji će sadržati samo proizvode čije ime sadrži reč **BICYCLE** i za koje je Ukupno između 1500 i 30000. Dobijene podatke sortirati po opadajućim vrednostima za Ukupno.

```
SELECT IME_PROIZVODA, SUM (KOLICINA * MIN_CENA)
      FROM PROIZVOD
 WHERE IME_PROIZVODA LIKE „%BICYCLE”
 GROUP BY IME_PROIZVODA
 HAVING SUM (KOLICINA * MIN_CENA) BETWEEN 1500 AND
 30000
 ORDER BY SUM (KOLICINA * MIN_CENA) DESC;
```

S_PROIZVODA	IME_PROIZVODA	KOLICINA	MIN_CENA
1	BICYCLE	2	900
2	RED_BICYCLE	5	170
3	BALL	6	200
4	MAN_BICYCLE	10	2500
5	SHIRTS	1	1000

```
1 SELECT IME_PROIZVODA, SUM(KOLICINA * MIN_CENA)
2   FROM PROIZVOD
3  WHERE IME_PROIZVODA LIKE "%BICYCLE"
4  GROUP BY IME_PROIZVODA
5  HAVING SUM(KOLICINA * MIN_CENA) BETWEEN 1500 AND 30000
6  ORDER BY SUM(KOLICINA * MIN_CENA) DESC
```

IME_PROIZVODA	SUM(KOLICINA * MIN_CENA)
MAN_BICYCLE	25000
BICYCLE	1800

PRIMER 2

Upotreba klauzula GROUP BY, HAVING i ORDER BY

Kreirati izveštaj sa kolonama Ime_odeljenja, Lokacija, Ime_radnik iz tabele Radnik (Sifra_radnika, Ime_radnika, Ime_odeljenja, Lokacija, Zarada)u kojem su prikazani samo podaci o Radnicima koji ne rade u odeljenju SALES i čija je zarada veća ili jednaka od prosečne zarade. Izveštaj sortirati po Imenu odeljenja, po Lokaciji, pa po Zaradi od najveće do najmanje vrednosti.

```
SELECT IME_ODELJENJA, LOKACIJA, IME_RADNIKA,
ZARADA
FROM RADNIK
WHERE IME_ODELJENJA != 'SALES' AND ZARADA >=
(SELECT AVG (ZARADA) FROM RADNIK)
ORDER BY IME_RADNIKA, LOKACIJA, ZARADA;
```

The screenshot shows a database query interface. At the top, the SQL query is displayed: `SELECT * FROM `radnik``. Below the query, there are controls for the number of rows (set to 25) and a search filter. The main table displays the following data:

	SIFRA_RADNIKA	IME_RADNIKA	IME_ODELJENJA	LOKACIJA	ZARADA
<input type="checkbox"/>	2250	ALEKSANDRA	PRERADA_MLEKA	HRVATSKA	5
<input type="checkbox"/>	2278	CVARKOV_DJORDJE	KUVAR	TUZLA	100000
<input type="checkbox"/>	2415	BORIS	IT_DEPARTMENT	DORCOL	500
<input type="checkbox"/>	5873	BOJAN	LEPLJENJE_ZICA	MALI MOKRI LUG	10
<input type="checkbox"/>	24601	VUK	LEMLJENJE_TABLI	KRNJACA	1000
<input type="checkbox"/>	78563	ZAN VALZAN	GURAC_BRODOVA	FRANCUSKA	20

Below the table, there are controls for the number of rows (set to 25) and a search filter. The summary table shows the following data:

IME_ODELJENJA	LOKACIJA	IME_RADNIKA	ZARADA
KUVAR	TUZLA	CVARKOV_DJORDJE	100000

PRIMER 3

Upotreba klauzula GROUP BY, HAVING i ORDER BY

Kreirati izveštaj sa kolonama Naziv_proizvoda, Broj prodaja i Ukupna vrednost (Suma iznosa) na osnovu tabele PRODAJA (Sifra_proizvoda, Naziv_proizvoda, Iznos) za sve proizvode čija šifra počinje sa 10 i čiji je broj prodaja veći od maksimalnog broja prodaja. Podatke urediti po broju prodaja od najvećeg do najmanjeg, pa po ukupnoj vrednosti (od najveće do najmanje).

```
SELECT NAZIV_PROIZVODA, COUNT (*), SUM (IZNOS)
FROM PRODAJA
WHERE SIFRA_PROIZVODA LIKE '10%'
GROUP BY NAZIV_PROIZVODA
HAVING COUNT (*) >
(SELECT COUNT(*) as br FROM PRODAJA GROUP BY
SIFRA_PROIZVODA order by 1 desc limit 1)
ORDER BY COUNT (*), SUM (IZNOS);
```

The screenshot displays a database management tool interface. At the top, a SQL query is entered: `SELECT * FROM `prodaja``. Below the query, there are controls for the number of rows (set to 25) and a filter box. A 'Sort by key' dropdown is set to 'None'. The main area shows a table with columns: SIFRA_PROIZVODA, NAZIV_PROIZVODA, and IZNOS. The table contains five rows of data. Each row has a checkbox and icons for Edit, Copy, and Delete. Below the table, there are options to 'Check All', 'With selected: Change, Delete, Export'. At the bottom, there is another row count and filter box, and a SQL query editor with the following code:

```
1 SELECT NAZIV_PROIZVODA, COUNT(*), SUM (IZNOS)
2 FROM PRODAJA
3 WHERE SIFRA_PROIZVODA LIKE '10%'
4 GROUP BY NAZIV_PROIZVODA
5 HAVING COUNT(*) >
6 (SELECT COUNT(*) as br FROM PRODAJA GROUP BY SIFRA_PROIZVODA
7 order by 1 desc limit 1)
8 ORDER BY COUNT (*), SUM (IZNOS);
```

At the very bottom, there are buttons for 'SELECT *', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', and 'Clear'.

Vežba: Klauzule GROUP BY, HAVING, ORDER BY

05

PRIMER KORIŠĆENJA GROUP BY OPERATORA NA STUDENTSKOJ BAZI

Vežbanje operatora group by

Primer 1.

Napisati upit koji prikazuje koliko je studenata položilo svaki od predmeta koji se na Fakultetu predaje sortiran u opadajućem redosledu po broju studenata koji su položili.

```
SELECT `id_predmeta`, count(*) from ispit where `ocena`>5 group by `id_predmeta` order by count(*) desc
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

id_predmeta	count(*)	1
1001	6	
2001	6	
1021	4	
3001	3	

Slika 1. Prikaz rezultata

Primer 2.

Napisati upit koji za predmete računa prosečnu ocenu studenata koji su taj predmet položili. Ne treba prikazati one predmete kojima je prosečna ocena veća od 9,00.

```
SELECT `id_predmeta`, avg(`ocena`) from ispit where `ocena`>5 group by `id_predmeta` having avg(ocena)<9
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

id_predmeta	avg('ocena')
1001	8.0000
1021	7.7500
2001	7.8333
3001	6.3333

Slika 2. Prikaz rezultata

PRIMER KORIŠĆENJA GROUP BY OPERATORA NA STUDENTSKOJ BAZI

Vežbanje operatora group by

Primer 3.

Napisati upit koji za predmete računa prosečnu ocenu studenata koji su taj predmet položili. Ne treba prikazati one predmete koje je položilo manje od 2 studenta.

```
SELECT `id_predmeta`, avg(`ocena`) from ispit where `ocena`>5 group by `id_predmeta` having count(*)>2
```

Number of rows: 25

Filter rows: Search this table

Sort by key: None

+ Options

id_predmeta	avg(`ocena`)
1001	8.0000
1021	7.7500
2001	7.8333
3001	6.3333

Slika 3. Prikaz rezultata

Upotreba NULL vrednosti u naredbi SELECT

Vežba: Upotreba NULL vrednosti u naredbi SELECT

06

ŠTA OZNAČAVA NULL VREDNOST

Vrednosti koje su nedefinisane

NULL može označavati vrednosti koje su nedefinisane. Između NULL vrednosti i vrednosti nula postoji značajna semantička razlika.

Ako posmatramo podatke o Goranu, trgovačkom putniku, vidimo da u koloni PREMIJA ima vrednost nula. To znači da "nije bio vredan" i nije zaradio nikakvu premiju, međutim ukoliko bude vredniji moći će da zaradi premiju pa će njegova ukupna primanja biti jednaka zbiru ličnog dohotka (LD) i premije.

Za razliku od njega Jovan, predsednik, u koloni premija nema nikakvu vrednost. Ma koliko bio vredan to polje za njega će uvek ostati prazno, nedefinisano, jer je PREMIJA neprimenljivo svojstvo za sve radnike sem za trgovačke putnike. Tako će svi radnici sem trgovačkih putnika u koloni PREMIJA imati nedefinisanu vrednost, odnosno NULL vrednost. NULL vrednosti (kao rezultat neprimenljivog svojstva) treba eliminisati iz baze podataka, i to je moguće i potrebno učiniti još u fazi projektovanja informacionog sistema.

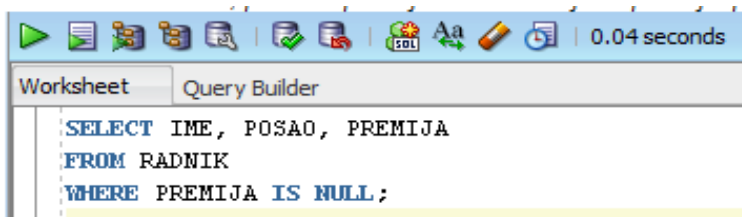
Drugi tip NULL vrednosti je još nepoznata vrednost i kao takva, ona je dozvoljena. O radniku Jovan, možemo uneti sve podatke sem šifre neposrednog rukovodioca (#RUKOV) jer nju još uvek ne znamo. Kada je budemo saznali unećemo je i NULL vrednost će biti eliminisana.

Bez obzira o kom tipu NULL vrednost se radi, određene kolone se na NULL vrednost mogu testirati pomoću dve specijalne klauzule: IS NULL ili IS NOT NULL za šta se koriste operatori poređenja.

PROVERA NULL VREDNOSTI

Vrši se uz pomoć klauzula *NULL* i *NOT NULL*

Primer: Prikazati ime, posao i premiju radnika koji ne primaju premiju (slika 1.)



```
SELECT IME, POSAO, PREMIJA
FROM RADNIK
WHERE PREMIJA IS NULL;
```

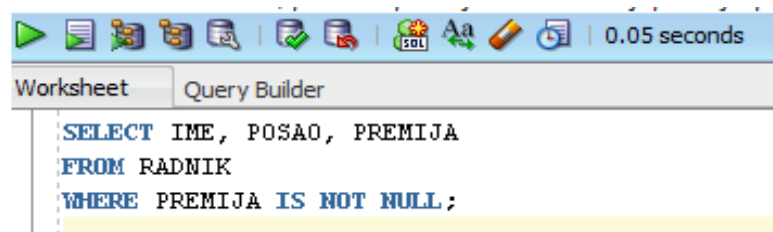
Slika 1. Upit za izvršenje prethodnog primera

Izlaz iz ove SELECT rečenice glasi (slika 2.):

IME	POS AO	PREMIJA
STEFAN	ANALITICAR	
MARKO	RUKOVODILAC	
ANA	RUKOVODILAC	
IVAN	RUKOVODILAC	
PAVLE	SAVETNIK	
JOVAN	PREDSEDNIK	
JELENA	ANALITICAR	
JANKO	ANALITICAR	
DEJAN	ANALITICAR	
FILIP	SAVETNIK	

Slika 2. Izlaz iz naredbe SELECT sa slike 1.

Primer: Prikazati ime, posao i premiju radnika koji primaju premiju (slika 3.).



```
SELECT IME, POSAO, PREMIJA
FROM RADNIK
WHERE PREMIJA IS NOT NULL;
```

Slika 3. Upit za izvršenje prethodnog primera

Iz naredbe SELECT se dobija sledeći izlaz (slika 4.):

IME	POS AO	PREMIJA
MILAN	TRG PUTNIK	30000
PETAR	TRG PUTNIK	50000
MARIJA	TRG PUTNIK	140000
GORAN	TRG PUTNIK	0

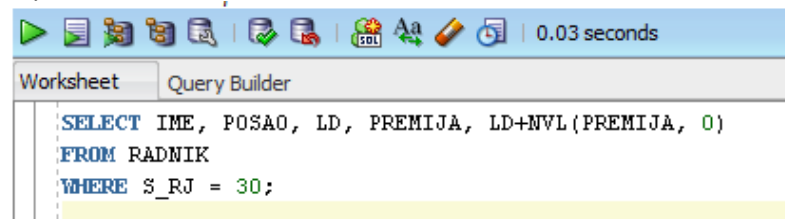
Slika 4. Izlaz iz naredbe SELECT sa slike 3.

UPOTREBA NULL VREDNOSTI U ARITMETIČKIM SRAČUNAVANJIMA

NULL se ne koristi pri izračunavanju izraza i funkcija, upotreba klauzule NVL

NULL vrednost je nedefinisana vrednost i zbog toga se ne koristi pri izračunavanju izraza i funkcija. Da bi se izračunavanje ipak omogućilo, koristi se NVL funkcija koja privremeno menja NULL vrednost sa vrednošću za koju se sami odlučimo, tj. vrednošću koja je neutralna u odnosu na željenu operaciju.

Primer: Prikaži ukupnu mesečnu zaradu radnika u RJ 30 (slika 5.).



```
SELECT IME, POSAO, LD, PREMIJA, LD+NVL(PREMIJA, 0)
FROM RADNIK
WHERE S_RJ = 30;
```

Slika 5. Upit za izvršenje prethodnog primera

Naredbom SELECT se dobija sledeći izlaz (slika 6):

IME	POSAO	LD	PREMIJA	LD+NVL(PREMIJA,0)
MILAN	TRG PUTNIK	160000	30000	190000
PETAR	TRG PUTNIK	125000	50000	175000
MARIJA	TRG PUTNIK	125000	140000	265000
ANA	RUKOVODILAC	285000		285000
GORAN	TRG PUTNIK	150000	0	150000
JANKO	ANALITICAR	95000		95000

Slika 6. Izlaz iz naredbe SELECT sa slike 5.

Izraz LD+NVL(PREMIJA,0) se izvršava se na sledeći način: ukoliko je PREMIJA nedefinisana, NULL vrednost se zamenjuje sa nulom i sabira se sa ličnim dohotkom (LD), inače se uzima konkretno definisana vrednost premije i sabira sa ličnim dohotkom.

Da nismo koristili NVL funkciju, odnosno da je izraz bio oblika LD+PREMIJA radnici DRAGAN i GORAN bi u rezultujućoj koloni koja odgovara navedenom izrazu imali nedefinisanu NULL vrednost. NULL vrednost se pojavljuje u rezultujućoj koloni jer je zbir konkretne vrednosti i nedefinisane vrednosti (nečeg što ne znamo šta je) uvek nedefinisana vrednost.

Vežba: Upotreba NULL vrednosti u naredbi SELECT

06

PRIMER KORIŠĆENJA NULL-A

Vežbanje pisanja upita sa NULL vrednošću

Primer 1.

Napisati upit kojim se prikazuju podaci o studentima kojima je mesto rođenja NULL.

```
SELECT * FROM `dosije` WHERE mesto_rodjenja is null
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	INDEKS	IME	PREZIME	GOD_RODJENJA	MESTO_RODJENJA
<input type="checkbox"/> Edit Copy Delete	20100024	Nikola	Vukovic	1992	NULL
<input type="checkbox"/> Edit Copy Delete	20100027	Milena	Stankovic	0	NULL

Slika 1. Prikaz rezultata

Zaključak

ZAKLJUČAK

Šta smo naučili u ovoj lekciji?

U ovoj lekciji se govori o različitim primerima primene naredbe SELECT koja se može koristiti za rad nad jednom ili više tabela. Ovde je korišćenje naredbe SELECT ograničeno samo na rad sa jednom tabelom. Razrađen je veliki broj primera ove naredbe sa različitim mogućnostima korišćenja klauzule WHERE kojom se primenjuje funkcija restrikcije.

U predavanju se takođe govori i o aritmetičkim naredbama koje se koriste za dobijanje izmenjenog sadržaja tabele, sumarnim funkcijama i grupnoj funkciji GROUP BY čija je primena vrlo česta u praksi sa bazama podataka.