



IT250 - BAZE PODATAKA

Naredba SELECT)-rad sa jednom tabelom

Lekcija 09

PRIRUČNIK ZA STUDENTE

IT250 - BAZE PODATAKA

Lekcija 09

NAREDBA SELECT)-RAD SA JEDNOM TABELOM

- ✓ Naredba SELECT)-rad sa jednom tabelom
- ✓ Poglavlje 1: Naredba SELECT nad jednom tabelom
- ✓ Poglavlje 2: WHERE klauzula sa operatorima poređenja
- ✓ Poglavlje 3: WHERE klauzula sa logičkim operatorima
- ✓ Poglavlje 4: WHERE klauzula sa ne-numeričkim atributima
- ✓ Poglavlje 5: Aritmetičke i sumarne funkcije u naredbi SELECT
- ✓ Poglavlje 6: Klauzule GROUP BY, HAVING, ORDER BY
- ✓ Poglavlje 7: Upotreba NULL vrednosti u naredbi SELECT
- ✓ Poglavlje 8: Pokazna vežba
- ✓ Poglavlje 9: Domaći zadatak
- ✓ Zaključak

Copyright © 2017 – UNIVERZITET METROPOLITAN, Beograd. Sva prava zadržana. Bez prethodne pismene dozvole od strane Univerziteta METROPOLITAN zabranjena je reprodukcija, transfer, distribucija ili memorisanje nekog dela ili čitavih sadržaja ovog dokumenta., kopiranjem, snimanjem, elektronskim putem, skeniranjem ili na bilo koji drugi način.

Copyright © 2017 BELGRADE METROPOLITAN UNIVERSITY. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of Belgrade Metropolitan University.

▼ Uvod

UVOD

Šta ćemo naučiti u ovoj lekciji?

U ovoj lekciji će biti reči o naredbi SELECT čija se primena pokazuje samo nad jednom tabelom.

Najpre je prikazano kako se u naredbom SELECT mogu izdvojiti samo pojedini atributi, a zatim i redovi tabele na osnovu primene aritmetičkih operatora selekcije i projekcije.

Izdvajanje redova tabele se vrši specificiranjem različitih uslova koji se mogu specificirati u klauzuli WHERE.

Tako se u predavanju govori o primeni:

1. operatora poređenja
2. logičkih operatora
3. operatora koji se koriste uz nenumeričke vrednosti atributa itd.

u klauzuli WHERE.

U naredbi SELECT se mogu primeniti u aritmetičke funkcije (+, -, : i *) kao i sumarne funkcije (MIN, MAX, AVG, COUNT). Još jedna važna klauzula koja se može koristiti u naredbi SELECT jeste i GROUP BY koja služi za grupisanje redova tabele koji imaju istu/iste vrednosti jednog ili više atributa.

▼ Poglavlje 1

Naredba SELECT nad jednom tabelom

ČEMU SLUŽI NAREDBA SELECT?

Za izveštavanje iz relacione baze podataka.

SELECT naredba u SQL-u se može sastojati od do šest klauzula, ali samo prve dve - SELECT i FROM - su obavezne. Upit se kao i svaka druga SQL naredba završava tačkom i zarezom. Klauzule su navedene u sledećem redosledu, a klauzule između uglastih zagrada [...] su opcione:

1. SELECT klauzula navodi atribut ili funkcije koje treba naći.
2. FROM klauzula specificira sve relacije (tabele) koje su potrebne u upitu, uključujući spojene (join-ovane) relacije, ali ne one u ugnježenim upitima.
3. WHERE klauzula specificira uslove za izbor torki iz ovih relacija, uključujući i uslove spajanja ako je potrebno.
4. GROUP BY specificira atribut grupisanja, dok HAVING specificira uslov za izbor grupa, a ne za pojedinačne torke. Ugrađene agregatne funkcije COUNT, SUM, MIN, MAX i AVG se koriste zajedno sa grupisanjem, ali one se takođe mogu primeniti na sve izabrane torke u upitu bez GROUP BY klauzule
5. ORDER BY specificira redosled prikazivanje rezultata upita.

```
SELECT <atribut i lista funkcija>
    FROM <lista tabela>
[ WHERE <uslov> ]
[ GROUP BY <atribut (i) za grupisanje> ]
[ HAVING <uslov grupisanja> ]
[ ORDER BY <lista atributa> ];
```

Navođenjem liste atributa u klauzuli SELECT zadaje se operacija PROJEKCIJE.

Kvalifikacionim izrazom u klauzuli WHERE zadaju se uslovi RESTRIKCIJE (SELEKCIJE) i SPAJANJA, odnosno iskazi slični iskazima u relacionom računu.

Upiti u SQL-u mogu biti veoma složeni. Osnovni oblik SELECT izraza, koji se ponekad naziva mapiranje ili blok za selekciju se formira od tri klauzule SELECT, FROM i WHERE i ima sledeći oblik:

```
SELECT <lista atributa>
    FROM <lista tabela>
WHERE <uslov>;
```

Uopšteno, isti upit u SQL-u se može napisati na više načina. Na primer, mnogi upiti mogu biti specifikirani sa uslovima spajanja (join-ovanja) u WHERE klauzuli, ili korišćenjem join-ovanih relacija u FROM klauzuli, ili sa nekim oblikom ugnježđenih upita i IN operatora poređenja. Ova fleksibilnost u određivanju upita ima prednosti i nedostatke.

TABELA ZA IZVRŠENJE UPITA NAD RELACIJOM RJ

Relacija RJ

Mogućnosti naredbe SELECT uz korišćenje klauzula za selekciju specifičnih n-torki relacije će biti ilustrovane na primeru relacionog modela sastavljenog od dve relacije: RADNIK i RJ (RADNA_JEDINICA), s tim što ćemo u ovoj lekciji SELECT naredbu primenjivati samo nad jednom od ove dve relacije.

Relacija RJ ima attribute

1. **S_RJ** (šifra radne jedinice) [prost ključ]
2. **NAZIV** i
3. **GRAD**.

Ovde je prikazana naredba CREATE TABLE za kreiranje tabele RJ a sadržaj same tabele nad kojom će se prikazati primena naredbe SELECT prikazan je na slici 1.1.

	S_RJ	Naziv	Grad
►	10	PRODAJA	BEOGRAD
	20	PRIZVODNJA	NIS
	30	PROJEKTOVANJE	BOR
	40	ISTRAZIVANJE	NOVI SAD

Slika 1.1 Sadržaj tabele RJ [Izvor: Autor]

```
CREATE TABLE `it350`.`RJ` (  
  `S_RJ` INT NOT NULL ,  
  `NAZIV` VARCHAR(45) NULL ,  
  `GRAD` VARCHAR(45) NULL ,  
  PRIMARY KEY (`S_RJ`) ,  
  UNIQUE INDEX `S_RJ_UNIQUE` (`S_RJ` ASC) )  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

TABELE ZA IZVRŠENJE UPITA NAD RELACIJOM RADNIK

Relacija RADNIK

Relacija **RADNIK** ima attribute

1. **S_RADNIK** (šifra radnika) [prost ključ] ,
2. **IME**,
3. **POSAO**,
4. **S_RUKOV** (šifra neposrednog rukovodioca),
5. **DAT_ZAP** (datum zaposlenja),
6. **LD** (lični dohodak),
7. **PREMIJA** i
8. **S_RJ** (šifra RJ-a u kojoj radnik radi).

Atribut S_RJ relacije RADNIK je spoljni, sekundarni ključ i omogućuje povezivanje te relacije sa relacijom R_J.

Ovde je prikazana naredba CREATE TABLE za kreiranje tabelle RADNIK, a sadržaj same tabelle nad kojom će se prikazati primena naredbe SELECT prikazan je na slici 1.2.

	S_RADNIK	IMEPREZIME	POSAO	S_RUKOV	DAT_ZAP	LD	PREMIJA	S_RJ
▶	3069	STEFAN S	ANALITACAR	33602	2001-12-17	80000	NULL	20
	3199	MILAN M	TRG PUTNIK	3398	2001-02-01	160000	30000	30
	3221	PETAR P	TRGPUTNIK	3398	2001-02-22	125000	50000	30
	3266	MARKO M	UKOVODILAC	3539	2001-04-02	297500	NULL	20
	3354	MARIJA M	TRG PUTNIK	3398	2001-09-28	125000	140000	30
	3398	ANA A	UKOVODILAC	3539	2001-05-01	285000	NULL	30
	3482	IVAN I	UKOVODILAC	3539	2001-06-09	245000	NULL	10
	3488	PAVLE P	SAVETNIK	3266	2001-11-09	300000	NULL	20
	3539	JOVAN J	PREDSEDNIK	NULL	2001-11-17	500000	NULL	10
	3544	GORAN G	TRG PUTNIK	3398	2001-09-08	150000	0	30
	3576	JELENA J	ANALITICAR	3488	2001-09-23	110000	NULL	20
	3600	JANKO J	ANALITICAR	3398	2001-12-03	95000	NULL	30
	3602	FILIP F	SAVETNIK	3266	2002-01-23	300000	NULL	20
	3634	DEJAN D	ANALITICAR	3482	2001-12-03	130000	NULL	10

Slika 1.2 Sadržaj tabelle RADNIK [Izvor: Autor]

```
CREATE TABLE `it350`.`RADNIK` (
  `S_RADNIK` INT NOT NULL ,
  `IMEPREZIME` VARCHAR(45) NULL ,
  `POSAO` VARCHAR(45) NULL ,
  `S_RUKOV` INT NULL ,
  `DAT_ZAP` DATE NULL ,
  `LD` INT NULL ,
  `PREMIJA` VARCHAR(45) NULL ,
  `S_RJ` INT NULL ,
  PRIMARY KEY (`S_RADNIK`) ,
  UNIQUE INDEX `S_RADNIK_UNIQUE` (`S_RADNIK` ASC) ,
  INDEX `SR_J` (`S_RJ` ASC) ,
  CONSTRAINT `SR_J`
    FOREIGN KEY (`S_RJ`)
    REFERENCES `it350`.`rj` (`S_RJ` )
```

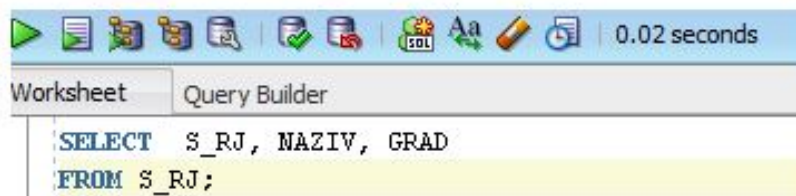
```
ON DELETE CASCADE  
ON UPDATE CASCADE)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

PRIKAZ NEIZMENJENOG SADRŽAJA TABELE

Primeri za prikaz neizmenjenog sadržaja tabele

Naredbom SELECT.....FROM se može prikazati neizmenjen sadržaj tabele ukoliko se u listi atributa navedu svi atributi tabele, a WHERE klauzulom nije specificiran nikakav uslov.

Primer 1: Prikazati šifre, nazive i lokacije svih radnih jedinica iz relacije R_J (slika 1. 3.)



Slika 1.3 SELECT naredba za prikaz sadržaja tabele S_RJ [Izvor: Autor]

Iz naredbe SELECT se dobija sledeći rezultat (slika 1.4.):

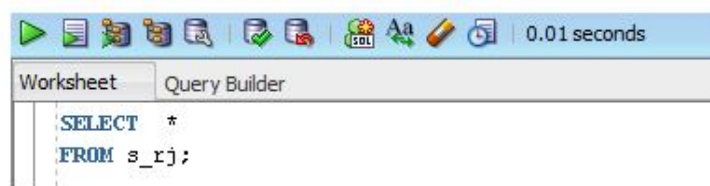
S_RJ	NAZIV	GRAD
10	Prodaja	Beograd
20	Proizvodnja	MIS
30	Projektovanje	BOR
40	Istrazivanje	NOVI SAD

Slika 1.4 Neizmenjeni sadržaj tabele S_RJ [Izvor: Autor]

U primeru je prikazano korišćenje naredbe SELECT za prikaz svih kolona iz tabele RADNA_JEDINICA, navođenjem njihovih imena i bez korišćenja WHERE uslova.

Kada se traže svi atributi neke relacije, umesto navođenja svih atributa pojedinačno, moguće je koristiti znak "*" sa istim dejstvom (slika 1. 5.)

Primer 2:



Slika 1.5 Drugi način za prikaz neizmenjenog sadržaja tabele S_RJ [Izvor: Autor]

Iz naredbe SELECT se dobija sledeći izlaz (slika 1.6.):

S_RJ NAZIV	GRAD
10 Prodaja	Beograd
20 Proizvodnja	MIS
30 Projektovanje	BOR
40 Istraživanje	NOVI SAD

Slika 1.6 Prikaz sadržaja svih atributa tabele [Izvor: Autor]

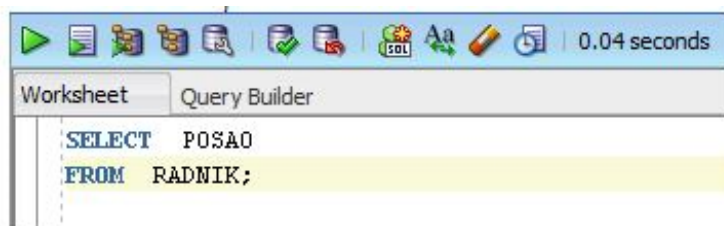
U primeru je prikazano korišćenje naredbe SELECT kojom se dobijaju svi atributi neke relacije korišćenjem znaka "*" i bez korišćenja WHERE klauzule

PRIKAZ SADRŽAJA SAMO NEKIH ATRIBUTA TABELE

Primeri za prikaz sadržaja samo nekih atributa tabele

Kada želimo da prikažemo samo neke attribute relacije, nakon reči SELECT navodimo listu tih atributa. (slika 1. 7 i slika 1. 8)

Primer: Prikazati sve poslove radnika.



Slika 1.7 Naredba SELECT za prikaz atributa POSAO [Izvor: Autor]

Iz naredbe SELECT se dobija sledeći izlaz:

```

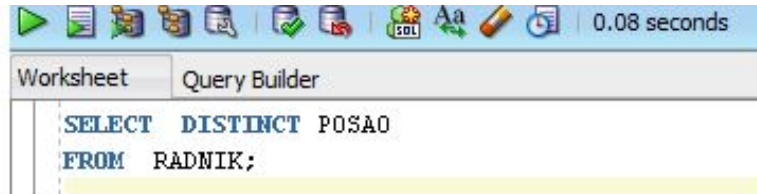
POSAO
-----
ANALITICAR
TRG PUTNIK
TRG PUTNIK
RUKOVODILAC
TRG PUTNIK
RUKOVODILAC
RUKOVODILAC
SAVETNIK
PREDSEDNIK
TRG PUTNIK
ANALITICAR
ANALITICAR
ANALITICAR
SAVETNIK
14 rows selected

```

Slika 1.8 Izlaz iz naredbe SELECT sa slike 1.7. [Izvor: Autor]

U primeru je prikazano korišćenje naredbe SELECT za prikaz samo nekih atributa relacije, navođenjem naziva tih atributa. Koristeći klausulu DISTINCT obezbeđujemo prikazivanje samo različitih vrednosti atributa (u ovom slučaju različitih poslova-slika 1. 9 i slika 1. 10).

Primer:



Slika 1.9 Korišćenje klauzule DISTINCT [Autor]

Iz naredbe SELECT se dobija sledeći izlaz:

```

POSAO
-----
PREDSEDNIK
SAVETNIK
ANALITICAR
TRG PUTNIK
RUKOVODILAC

```

Slika 1.10 Izlaz iz naredbe SELECT sa slike 1.9. [Izvor: Autor]

PRIKAZ SADRŽAJA SAMO NEKIH REDOVA TABELE

Moguće je korišćenjem klauzule WHERE u kojoj se mogu upotrebiti operatori poredjenja

SQL klauzula WHERE se koristi za specificiranje uslova prilikom preuzimanja torki (redova) iz relacije (tabele): prikazaće se samo oni redovi tabele koji zadovoljavaju zadati uslov u klauzuli WHERE.

Može se reći da WHERE klauzulu treba koristiti za filtriranje i prikazivanje samo neophodnih podataka.

Napomenimo i da WHERE klauzulu ne koristimo isključivo u sprezi sa SELECT naredbom, već se koristi i sa UPDATE, DELETE, itd. naredbama.

Za specificiranje kvalifikacionih izraza u WHERE klauzuli se mogu koristiti različite vrste operatora zavisno od toga da li treba da se primene nad numeričkim ili nenumeričkim vrednostima atributa.

Nad numeričkim vrednostima atributa mogu se primeniti:

1. operatori poredjenja
2. logički operatori

Nad nenumeričkim (karakter vrednostima) atributa mogu se primeniti specifični operatori kao što su IN, NOT IN, LIKE itd.

▼ Poglavlje 2

WHERE klauzula sa operatorima poređenja

OPERATORI POREĐENJA U WHERE KLAUZULI

Moguće je korišćenje klauzule WHERE u kojoj se mogu upotrebiti operatori poređenja

U WHERE klauzuli možemo koristiti operatore poređenja i to:

1. = (jednako)
2. != (nije jednako) -
3. <> (nije jednako) - drugi način predstavljanja !=
4. > (veće od)
5. < (manje od)
6. >= (veće od ili jednako)
7. <= (manje ili jednako)
8. !> (nije više od)
9. !< (nije manje od)

Primer: Neka su *a* i *b* dve celobrojene promenljive i neka je *a* = 10 i *b* = 20. Tada na osnovu navedenih operatora poređenja važi:

1. (*a* = *b*) - rezultat će biti false
2. (*a* != *b*) - rezultat će biti true
3. (*a* <> *b*) - rezultat će biti true
4. (*a* > *b*) - rezultat će biti false
5. (*a* < *b*) - rezultat će biti true
6. (*a* >= *b*) - rezultat će biti false
7. (*a* <= *b*) - rezultat će biti true
8. (*a* !> *b*) - rezultat će biti true
9. (*a* !< *b*) - rezultat će biti false

PRIMER 1. WHERE KLAUZULA SA OPERATOROM = I !=

Korišćenje operatora = i != u WHERE klauzuli

Primer 1.1: Prikazati sve podatke o radnicima koji rade u RJ 30

```
SELECT *
FROM RADNIK
WHERE S_RJ = 30;
```

Iz navedene naredbe SELECT se dobija sledeći rezultat (slika 2.1):

	S_RADNIK	IMEPREZIME	POSAO	S_RUKOV	DAT_ZAP	LD	PREMIJA	S_RJ
▶	3199	MILAN M	TRG PUTNIK	3398	0000-00-00	160000	30000	30
	3221	PETAR P	TRGPUTNIK	3398	0000-00-00	125000	5000	30
	3354	MARIJA M	TRG PUTNIK	3398	0000-00-00	125000	140000	30
	3398	ANA A	UKOVODILAC	3539	0000-00-00	285000	NULL	30
	3544	GORAN G	TRG PUTNIK	3398	0000-00-00	150000	0	30
	3600	JANKO J	ANALITICAR	3398	0000-00-00	95000	NULL	30

Slika 2.1 Izlaz iz SELECT za primer 1.1 [Izvor: Autor]

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 1.2: Prikazati sve podatke o radnicima koji ne rade u Rj 30

```
SELECT *
FROM RADNIK
WHERE S_RJ != 30;
```

Iz navedene naredbe SELECT se dobija sledeći rezultat (slika 2.2):

	S_RADNIK	IMEPREZIME	POSAO	S_RUKOV	DAT_ZAP	LD	PREMIJA	S_RJ
▶	3069	STEFAN S	ANALITICAR	33602	0000-00-00	80000	NULL	20
	3266	MARKO M	UKOVODILAC	3539	0000-00-00	297500	NULL	20
	3482	IVAN I	UKOVODILAC	3539	0000-00-00	245000	NULL	10
	3488	PAVLE P	SAVETNIK	3266	0000-00-00	300000	NULL	20
	3539	JOVAN J	PREDSEDIK	NULL	0000-00-00	500000	NULL	10
	3576	JELENA J	ANALITICAR	3488	0000-00-00	110000	NULL	20
	3602	FILIP F	SAVETNIK	3266	0000-00-00	300000	NULL	20
	3634	DEJAN D	ANALITICAR	3482	0000-00-00	130000	NULL	10

Slika 2.2 Izlaz iz SELECT za primer 1.2. [Izvor: Autor]

PRIMER 2: WHERE KLAUZULA SA OPERATORIMA VEĆE I MANJE

Upotreba operatora > i < u WHERE klauzuli

Primer 1.3 : Prikaži podatke o radnicima koji imaju premiju veću od ličnog dohotka

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

Naredbom SELECT se dobija sledeći rezultat (slika 2.3):

	S_RADNIK	IMEPREZIME	POSAD	S_RUKOV	DAT_ZAP	LD	PREMIJA	S_RJ
▶	3354	MARIJA M	TRG PUTNIK	3398	0000-00-00	125000	140000	30

Slika 2.3 Izlaz iz SELECT za primer 1.3 [Izvor: Autor]

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

Primer 1.4: Prikaži podatke o radnicima koji imaju premiju manju od ličnog dohotka

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

Naredbom SELECT se dobija sledeći rezultat (slika 2.4):

	S_RADNIK	IMEPREZIME	POSAD	S_RUKOV	DAT_ZAP	LD	PREMIJA	S_RJ
▶	3199	MILAN M	TRG PUTNIK	3398	0000-00-00	160000	30000	30
	3221	PETAR P	TRGPUTNIK	3398	0000-00-00	125000	5000	30
	3544	GORAN G	TRG PUTNIK	3398	0000-00-00	150000	0	30

Slika 2.4 Izlaz iz SELECT za primer 1.4 [Izvor: Autor]

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

PRIMER 3: WHERE KLAUZULA SA OPERATORIMA VEĆE JEDNAKO I MANJE JEDNAKO

Upotreba operatora >= i <=

Primer 1.5: Prikaži podatke o radnicima koji su zaposleni pre 09.06.2001. god., uključujući i taj datum

```
SELECT *
FROM RADNIK
WHERE DAT_ZAP <= '2001-06-09';
```

Naredbom SELECT se dobija sledeći rezultat (slika 2.5):

	S_RADNIK	IMEPREZIME	POSAD	S_RUKOV	DAT_ZAP	LD	PREMIJA	S_RJ
▶	3199	MILAN M	TRG PUTNIK	3398	2001-02-01	160000	30000	30
	3221	PETAR P	TRGPUTNIK	3398	2001-02-22	125000	50000	30
	3266	MARKO M	RUKOVIDILAC	3539	2001-04-02	297500	NULL	20
	3398	ANA A	RUKOVIDILAC	3539	2001-05-01	285000	NULL	30
	3482	IVAN I	RUKOVIDILAC	3539	2001-06-09	245000	NULL	10

Slika 2.5 Izlaz iz SELECT za primer 1.5 [Izvor: Autor]

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

Primer 1.6: Prikaži podatke o radnicima koji su zaposleni posle 09.06.2001. god., uključujući i taj datum

```
SELECT *
FROM RADNIK
WHERE DAT_ZAP >= '2001-06-09';
```

Naredbom SELECT se dobija sledeći rezultat (slika 2.6):

	S_RADNIK	IMEPREZIME	POSLO	S_RUKOV	DAT_ZAP	LD	PREMIJA	S_RJ
▶	3069	STEFAN S	ANALITACAR	33602	2001-12-17	80000	NULL	20
	3354	MARIJA M	TRG PUTNIK	3398	2001-09-28	125000	140000	30
	3482	IVAN I	RUKOVODILAC	3539	2001-06-09	245000	NULL	10
	3488	PAVLE P	SAVETNIK	3266	2001-11-09	300000	NULL	20
	3539	JOVAN J	PREDSEDNIK	NULL	2001-11-17	500000	NULL	10
	3544	GORAN G	TRG PUTNIK	3398	2001-09-08	150000	0	30
	3576	JELENA J	ANALITACAR	3488	2001-09-23	110000	NULL	20
	3600	JANKO J	ANALITACAR	3398	2001-12-03	95000	NULL	30
	3602	FILIP F	SAVETNIK	3266	2002-01-23	300000	NULL	20
	3634	DEJAN D	ANALITACAR	3482	2001-12-03	130000	NULL	10

Slika 2.6 Izlaz iz SELECT za primer 1.6 [Izvor: Autor]

Primerom je prikazano korišćenje operatora \geq u WHERE klauzuli naredbe SELECT.

▼ Poglavlje 3

WHERE klauzula sa logičkim operatorima

PRIMER 1: WHERE KLAUZULA SA OPERATORIMA AND I OR

Upotreba operatora AND i OR u WHERE klauzuli

U WHERE klauzuli se mogu koristiti logički operatori:

1. **AND** - omogućava višestruke uslove u SQL upitu, ali svi uslovi moraju biti ispunjeni
2. **OR** - koristi se za kombinovanje više uslova u upitu, tako da mora biti ispunjen barem jedan uslov
3. **NOT** - negacija, okreće značenje logičkih operatora sa koji se koristi

Primer 2.1: Prikaži ime, posao i LD svakog radnika iz Rj 20 koji zarađuje više od 200000 din

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

Naredbom SELECT se dobija sledeći rezultat (slika 3.1):

	IMEPREZIME	POSAO	LD
▶	MARKO M	UKOVODILAC	297500
	PAVLE P	SAVETNIK	300000
	FILIP F	SAVETNIK	300000

Slika 3.1 Izlaz iz SELECT primer 2.1 [Izvor: Autor]

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

Primer 2.2: Prikazati sve podatke o rukovodiocima radnih jedinica i predsedniku

```
SELECT *
FROM RADNIK
WHERE POSAO = 'UKOVODILAC'
OR POSAO = 'PRESEDNIK';
```

Naredbom SELECT se dobija sledeći rezultat (slika 3.2):

	S_RADNIK	IMEPREZIME	POSAO	S_RUKOV	DAT_ZAP	LD	PREMIJA	S_RJ
▶	3266	MARKO M	RUKOVODILAC	3539	2001-04-02	297500	NULL	20
	3398	ANA A	RUKOVODILAC	3539	2001-05-01	285000	NULL	30
	3482	IVAN I	RUKOVODILAC	3539	2001-06-09	245000	NULL	10
	3539	JOVAN J	PREDSEDNIK	NULL	2001-11-17	500000	NULL	10

Slika 3.2 Izlaz iz SELECT primer 2.2 [Izvor: Autor]

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

PRIMER 2: WHERE KLAUZULA SA OPERATOROM NOT

Primeri za operator NOT.

Primer 2.3: Prikaži ime, LD, posao i šifru radne jedinice za rukovodioce koji ne rade u radnoj jedinici 30 (korišćenjem logičkog operatora NOT)

```
SELECT IMEPREZIME, LD, POSAO, S_RJ
FROM RADNIK
WHERE POSAO = 'RUKOVODILAC'
AND NOT (S_RJ = 30);
```

Naredbom SELECT se dobija sledeći rezultat (slika 3.3):

	IMEPREZIME	LD	POSAO	S_RJ
▶	MARKO M	297500	RUKOVODILAC	20
	IVAN I	245000	RUKOVODILAC	10

Slika 3.3 Izlaz iz SELECT primer 2.3 [Izvor: Autor]

Primerom je prikazano korišćenje operatora **NOT**. Sledi ekvivalentan upit korišćenjem operatora poređenja **!=**.

```
SELECT IMEPREZIME, LD, POSAO, S_RJ
FROM RADNIK
WHERE POSAO = 'RUKOVODILAC'
AND (S_RJ != 30);
```

Primer 2.4: Prikaži ime, LD, posao i šifru PS-a za sve ne- rukovodioce u RJ -u 20 (korišćenjem samo NOT, AND i OR logičkih operatora)

```
SELECT IMEPREZIME, LD, POSAO, S_RJ
FROM RADNIK
WHERE ( NOT POSAO = 'RUKOVODILAC' )
AND NOT (S_RJ = 10 OR S_RJ = 30);
```

Naredbom SELECT se dobija sledeći rezultat (slika 3.4) :

	IMEPREZIME	LD	POSAO	S_RJ
►	STEFAN S	80000	ANALITICAR	20
	PAVLE P	300000	SAVETNIK	20
	JELENA J	110000	ANALITICAR	20
	FILIP F	300000	SAVETNIK	20

Slika 3.4 Izlaz iz SELECT primer 2.4 [Izvor: Autor]

Primerom je prikazano istovremeno korišćenje operatora NOT, AND i OR. Dati primer može se rešiti i primenom operatora poređenja:

```
SELECT IMEPREZIME, LD, POSAO, S_RJ
FROM RADNIK
WHERE POSAO != 'RUKOVODILAC'
AND S_RJ = 20;
```

PRIMER 3: WHERE KLAUZULA SA ISTOVREMENIM KORIŠĆENJEM VIŠE LOGIČKIH OPERATORA

Korišćenje AND i OR logičkih operatora

Primer 2.5: Prikaži ime, LD, posao i šifru PS-a za rukovodioca ili analitičare u RJ -u 10 ili RJ 30 (korišćenje zagrada da bi se definisao redosled ispitivanja uslova kod istovremene primene AND i OR logičkih operatora)

```
SELECT IMEPREZIME, LD, POSAO, S_RJ
FROM RADNIK
WHERE (POSAO = 'RUKOVODILAC'
OR POSAO = 'ANALITICAR')
AND (S_RJ = 10 OR S_RJ = 30);
```

Naredbom SELECT se dobija sledeći rezultat (slika 3.5) :

	IMEPREZIME	LD	POSAO	S_RJ
►	ANA A	285000	RUKOVODILAC	30
	IVAN I	245000	RUKOVODILAC	10
	JANKO J	95000	ANALITICAR	30
	DEJAN D	130000	ANALITICAR	10

Slika 3.5 Izlaz iz SELECT primer 2.5 [Izvor: Autor]

Primerom je prikazano istovremeno korišćenje operatora AND i OR.

▼ Poglavlje 4

WHERE klauzula sa ne-numeričkim atributima

PRIMER: WHERE KLAUZULA SA OPERATOROM IN

Operator IN

Izrazi nad nizom karaktera koji se najčešće koriste su: IN, NOT IN, LIKE

Operator IN služi za selekcijun-torki koje zadovoljavaju vrednost iz liste vrednosti.

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

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

Naredbom SELECT se dobija sledeći rezultat (slika 4.1):

	IMEPREZIME	POSAO	S_RJ
▶	MARKO M	RUKOVODILAC	20
	ANA A	RUKOVODILAC	30
	IVAN I	RUKOVODILAC	10
	JOVAN J	PREDSEDNIK	10

Slika 4.1 Izlaz iz SELECT [Izvor: Autor]

Primerom je prikazano korišćenje operatora NOT IN.

Isti uslov moguće je konceptualno iskazati i na sledeći način:

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

odnosno realizovati sledećom WHERE klauzulom:

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

PRIMER: WHERE KLAUZULA SA OPERATOROM LIKE I KARAKTEROM „%“

Operator LIKE uz korišćenje specijalnog karaktera '%'

Primer 3.1: Prikazati ime, posao i šifru RJ radnika čija imena počinju sa M.

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

Naredbom SELECT se dobija sledeći rezultat (slika 4.2):

	IMEPREZIME	POSAO	S_RJ
▶	MILAN M	TRG PUTNIK	30
	MARKO M	UKOVODILAC	20
	MARIJA M	TRG PUTNIK	30

Slika 4.2 Izlaz iz SELECT, primer 3.1 [Izvor: Autor]

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:

1. "%" predstavlja string od 0 ili više karaktera
2. "_" predstavlja poziciju jednog karaktera.

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

Primer 3.2:

... gde prezime počinje sa J (Primer 3.2)

```
SELECT IMEPREZIME, POSAO, S_RJ
FROM RADNIK
WHERE IMEPREZIME LIKE 'J%';
```

Naredbom SELECT se dobija sledeći rezultat (slika 4.3):

	IMEPREZIME	POSAD	S_RJ
▶	JOVAN J	PREDSEDNİK	10
	JELENA J	ANALITICAR	20
	JANKO J	ANALITICAR	30

Slika 4.3 Izlaz iz SELECT, primer 3.2 [Izvor: Autor]

PRIMER: WHERE KLAUZULA SA OPERATOROM LIKE I KARAKTEROM „_“

Operator LIKE uz korišćenje specijalnog karaktera '_'

Primer 3.3: ... gde je treći karakter imena R

```
SELECT IMEPREZIME, POSAD, S_RJ
FROM RADNIK
WHERE IMEPREZIME LIKE '__R%';
```

	IMEPREZIME	POSAD	S_RJ
▶	MARKO M	RUKOVODILAC	20
	MARIJA M	TRG PUTNIK	30
	GORAN G	TRG PUTNIK	30

Slika 4.4 Rezultat za SELECT, primer 3.3 [Izvor: Autor]

Primer 3.4: ... gde je ime dugačko 5 karaktera

```
SELECT IMEPREZIME, POSAD, S_RJ
FROM RADNIK
WHERE IMEPREZIME LIKE '_____';
```

	IMEPREZIME	POSAD	S_RJ
▶	ANA A	RUKOVODILAC	30

Slika 4.5 Rezultat za SELECT, primer 3.4 [Izvor: Autor]

Primer 3.5: .. gde ime nije dugačko 5 karaktera .

```
SELECT IMEPREZIME, POSAD, S_RJ
FROM RADNIK
WHERE IMEPREZIME NOT LIKE '_____ _';
```

	IMEPREZIME	POSAO	S_RJ
▶	STEFAN S	ANALITICAR	20
	MARIJA M	TRG PUTNIK	30
	ANA A	UKOVODILAC	30
	IVAN I	UKOVODILAC	10
	JELENA J	ANALITICAR	20

Slika 4.6 Rezultat za SELECT, primer 3.5 [Izvor: Autor]

Primer 3.6:.... gde je u imenu slovo G posle R

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

	IMEPREZIME	POSAO	S_RJ
▶	GORAN G	TRG PUTNIK	30

Slika 4.7 Rezultat za SELECT, primer 3.6 [Izvor: Autor]

OSTALI OPERATORI NAD KOLONAMA TIPAA CHARACTER

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

Najčešće funkcije nad kolonama tipa character koje se primenjuju su:

1. **String1 || string2** - spaja stringove karaktera
2. **LENGTH (str)** - nalazi dužinu stringa
3. **SUBSTR (str, spos, [, len])** - daje podstring od "len" karaktera, polazeći od pozicije "spos,,

Mogu se primeniti i funkcije:

1. 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

2. 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",

3. 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.

Sve ove funkcije primenjuju se nad atributima koji se nalaze iza SELECT klauzule.

PRIMER: OPERATOR || I LENGTH (STR) NAD KOLONAMA TIPa CHARACTER

Upotreba operatora || i LENGTH (str)

Primer 3.6: Prikaži imena radnika Rj 30 iz kojih neposredno treba da dođe posao koji obavlja. Sortirati rezultujuću tabelu u rastućem redosledu vrednosti atributa POSAO

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

Rezultat naredbe SELECT glasi (slika 4.8):

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

Slika 4.8 Rezultat za SELECT, primer 3.6 [Izvor: Autor]

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

Primer 3.7: Koliko su duga imena Rj-a ?

```
SELECT NAZIV, LENGHT(NAZIV) DUZINA  
FROM RJ;
```

Rezultat naredbe SELECT glasi (slika 4.9):

NAZIV	DUZINA
PRODAJA	7
PROIZVODNJA	11
PROJEKTOVANJE	13
ISTRAZIVANJE	12

Slika 4.9 Rezultat za SELECT, primer 3.7 [Izvor: Autor]

U primeru je prikazano korišćenje funkcije LENGHT nad nizom karaktera.

U MySQL funkcija LENGTH() se javlja tek od verzije 4.6.

PRIMER: OPERATOR DECODE NAD KOLONAMA TIPA CHARACTER

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

Primer 3.8: 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.

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

Rezultat naredbe SELECT glasi (slika 4.10):

	IMEPREZIME	POSAD	KLASA
►	STEFAN S	ANALITICAR	1
	MILAN M	TRG PUTNIK	2
	PETAR P	TRG PUTNIK	2
	MARKO M	RUKOVODILAC	3
	MARIJA M	TRG PUTNIK	2
	ANA A	RUKOVODILAC	3
	IVAN I	RUKOVODILAC	3
	PAVLE P	SAVETNIK	2
	JOVAN J	PREDSEDNİK	5
	GORAN G	TRG PUTNIK	2
	JELENA J	ANALITICAR	1
	JANKO J	ANALITICAR	1
	FILIP F	SAVETNIK	2
	DEJAN D	ANALITICAR	1

Slika 4.10 Rezultat za SELECT, primer 3.8 [Izvor: Autor]

MySQL alternativa bi glasila:

```
SELECT IMEPREZIME, POSAD,
CASE POSAD
  WHEN 'ANALITICAR' THEN 1
  WHEN 'RUKOVODILAC' THEN 3
  WHEN 'PREDSEDNİK' THEN 5
  ELSE 2
END AS KLASA
FROM RADNIK;
```


▼ Poglavlje 5

Aritmetičke i sumarne funkcije u naredbi SELECT

ARITMETIČKE FUNKCIJE

*Dobijaju se primenom aritmetički operatori +, -, *, /, POWER(broj, e), ROUND (broj [,d]), TRUNC (broj [,d]), SIGN (broj),...*

Aritmetičke funkcije u naredbi **SELECT** se koriste za izvršenje bilo koje aritmetičke operacije u upitu.

Sastavljene su od imena kolona i konstantnih vrednosti povezanih aritmetičkim operatorima ("+", "*", "-", "/"). U aritmetičkim izrazima se mogu koristiti i grupne funkcije.

Koristi se sledeća sintaksa:

SELECT aritmetička_funkcija;

Primer 4.1: *Koliko je srednje godišnje primanje trgovačkih putnika*

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

Iz ove SELECT naredbe se dobija sledeći izlaz (slika 5.1):

	AVG(LD+PREMIJA) * 12
▶	2340000.0000

Slika 5.1 Rezultat za SELECT primer 4.1 [Izvor: Autor]

Pored aritmetičkih operatora "+", "*", "-", "/" SQL podržava sledeće i aritmetičke funkcije:

1. **POWER (broj, e)** – diže broj na e-ti stepen
2. **ROUND (broj [,d])** – zaokrugljuje broj na d decimala
3. **TRUNC (broj [,d])** – odbacuje ostatak od d-tog decimalnog mesta
4. **ABS (broj)** – nalazi apsolutnu vrednost broja
5. **SIGN (broj)** – daje +1 ako je broj >0, 0 ako je broj = 0, -1 ako je broj <0.
6. **MOD (broj₁, broj₂)** – izračunava broj₁ po modulu broj₂
7. **SQRT (broj)** – nalazi pozitivan kvadratni koren broja

PRIMER: ARITMETIČKE FUNKCIJE PRIMENOM OPERATORA ROUND

Aritmetički operator ROUND (broj [,d]) zaokrugljuje broj na decimala

Primer 4.2: Koji radnici zarađuju više od 1000 dinara po satu? Zaradu po satu zaokrugliti na 2 decimale. (Podrazumeva se da postoje 22 radna dana u mesecu i 8 radnih sati u danu)

```
SELECT IMEPREZIME, ROUND(LD/(22*8),2)
"ZARADA PO CASU"
FROM RADNIK
WHERE LD/(22*8) > 1000;
```

Iz SELECT naredbe se dobija sledeći rezultat (slika 5.2):

	IMEPREZIME	ZARADA PO CASU
▶	MARKO M	1690.34
	ANA A	1619.32
	IVAN I	1392.05
	PAVLE P	1704.55
	JOVAN J	2840.91
	FILIP F	1704.55

Slika 5.2 Rezultat za SELECT primer 4.2 [Izvor: Autor]

Primer prikazuje još jedan način 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

To su: AVG (atribut), SUM (atribut), MIN (atribut), MAX (atribut), COUNT ()*

Funkciju za dobijanje sumarnih informacija u naredbi SELECT su:

1. AVG (atribut) - izračunava srednju vrednost
2. SUM (atribut) - izračunava ukupnu vrednost
3. MIN (atribut) - nalazi minimalnu vrednost
4. MAX (atribut) - nalazi maksimalnu vrednost
5. COUNT - prouzrokuje sažimanje sadržaja kolona nad kojima se primenjuje
 - COUNT (*) - nalazi broj n-torki u grupi
 - COUNT (atribut) - nalazi broj NOT-NULL vrednosti kolone
 - COUNT (DISTINCT atribut) - nalazi broj različite NOT-NULL vrednosti kolone

Sve funkcije za dobijanje sumarnih informacija se koriste za sumiranje informacija iz više torki u jednu sumarnu torku.

Funkcije SUM, MAX, MIN i AVG se mogu primeniti na skup ili numeričkih vrednosti i vraćaju, sumu, maksimalnu vrednost, minimalnu vrednost i prosek (srednja vrednost) tih vrednosti respektivno. Ove funkcije se mogu koristiti u SELECT klauzuli ili u HAVING klauzuli. Funkcije MAX i MIN se takođe mogu koristiti nad atributima koji imaju nenumeričke domene ako se vrednosti domena mogu međusobno upoređivati.

PRIMER 2: SUMARNE FUNKCIJE MIN, AVG, MAX, COUNT

Primeri sumarnih funkcija MIN, AVG, MAX, COUNT

Primer 4.3: Naći minimalni, srednji i maksimalni lični dohodak, kao i broj radnika u Rj 10 .

```
SELECT MIN(LD), AVG(LD), MAX(LD), COUNT(*)  
FROM RADNIK  
WHERE S_RJ = 10;
```

Naredbom SELECT se dobija sledeći rezultat (slika 4.3):

	MIN(LD)	AVG(LD)	MAX(LD)	COUNT(*)
▶	130000	291666.6667	500000	3

Slika 5.3 Rezultat za SELECT, primer 4.3 [Izvor: Autor]

Primer prikazuje korišćenje funkcija MIN, AVG, MAX, COUNT. Kao što se vidi, rezultujuća tabela se sastoji od samo jedne n-torke.

Primer 4.4: Naći ukupan LD i ukupnu premiju za trgovačke putnike

```
SELECT SUM(LD), SUM(PREMIJA)  
FROM RADNIK  
WHERE POSAO = 'TRG PUTNIK';
```

Naredbom SELECT se dobija sledeći rezultat (slika 5.4):

	SUM(LD)	SUM(PREMIJA)
▶	560000	220000

Slika 5.4 Rezultat za SELECT, primer 4.4 [Izvor: Autor]

Primer 4.5: Naći minimalni, srednji, maksimalni LD i LD + premiju, kao i broj Trgovačkih putnika u tabeli.

```
SELECT COUNT(LD) as 'Radnika',  
MIN(LD), AVG(LD), MAX(LD),
```

```
MIN(LD+PREMIJA), AVG(LD+PREMIJA),  
MAX(LD+PREMIJA)  
FROM RADNIK  
WHERE POSAO = 'TRG PUTNIK';
```

Naredbom SELECT se dobija sledeći rezultat (slika 5.5):

	RAdnika	MIN(LD)	AVG(LD)	MAX(LD)	MIN(LD+PREMIJA)	AVG(LD+PREMIJA)	MAX(LD+PREMIJA)
►	4	125000	140000.0000	160000	150000	195000.0000	265000

Slika 5.5 Rezultat za SELECT, primer 4.5 [Izvor: Autor]

▼ Poglavlje 6

Klauzule GROUP BY, HAVING, ORDER BY

KLAUZULA GROUP BY

Koristi se u slučajevima kada je potrebno izvršiti grupisanje redova tabela izdvajanje pojedinih grupa.

Klauzula **GROUP BY** omogućava dobijanje sumarne informacije za svaku različitu vrednost kolone ili grupe kolona po kojoj se vrši grupisanje. **Klauzula GROUP BY se gotovo uvek koristi uz neku funkciju za dobijanje sumarnih informacija (MIN, MAX, AVG, COUNT, SUM).**

Klauzula GROUP BY se koristi u slučajevima kada je potrebno izvršiti grupisanje redova tabela i izdvajanje pojedinih grupa.

Primer 5.1: Naći minimalni, srednji i maksimalni LD kao i ukupan broj radnika u svakom RJ.

Na osnovu do sada rečenog, ovaj upit bi se mogao rešiti formiranjem onoliko SQL SELECT naredbi koliko ima različitih PS-a. Pri tome se SELECT i FROM klauzule ne bi menjale a menjao bi se samo uslov selekcije u WHERE klauzuli (za RJ sa šifrom 10, 20 i 30). Rezultat svake SELECT naredbe bila bi jedna n-torka, tako da bi ukupno bile prikazane tri n-torke.

Identičan rezultat dala bi i sledeća naredba:

```
SELECT MIN(LD), AVG(LD), MAX(LD), COUN(*), S_RJ  
FROM RADNIK  
GROUP BY S_RJ;
```

Rezultat ove SELECT naredbe glasi (slika 1 . .):

	MIN(LD)	AVG(LD)	MAX(LD)	COUNT(*)	S_RJ
▶	130000	291666.6667	500000	3	10
	80000	217500.0000	300000	5	20
	95000	156666.6667	285000	6	30

Slika 6.1 Rezultata za SELECT, primer 5.1 [Izvor: Autor]

Pored funkcija, u SELECT listi se nalazi samo onaj prost atribut po kome se vrši grupisanje.

ATRIBUTI GRUPISANJA U GROUP BY KLAUZULI

GROUP BY klauzula deli relaciju na grupe pri čemu se svaka grupa sastoji od redova koji imaju istu vrednost nekih atributa, koji se nazivaju atribut (i) grupisanja.

Primer 5.2: Na slici 6.2. prikazan je sadržaj tabele EMPLOYEE koja sadrži podatke o radnicima (Fname, Minit, Lname, Ssn, Salary, Super_ssn) kao i šifru odeljenja u kojoj svaki radnik radi (Dno). Za svako odeljenje, pronađite broj odeljenja, broj zaposlenih u odeljenju i njihovu prosečnu platu.

U ovim slučajevima potrebno je podeliti relaciju u podskupove (ili grupe) koji se ne preklapaju. Svaka grupa (particija) će sastojati se od redova koji imaju istu vrednost nekih atributa, koji se nazivaju atribut (i) grupisanja. Primenom klauzule GROUP BY na svaku grupu možemo da primenimo sumarnu funkciju i da za svaku od njih dobijemo sumarne informacije. Atributi grupisanja treba da se pojave u SELECT klauzuli, tako da se vrednost koja proističe iz primene svake sumarne funkcije na grupu torki pojavljuje zajedno sa vrednošću atributa grupisanja.

SELECT naredba za primer 5.2:

```
SELECT Dno, COUNT (*), AVG (Salary)
FROM EMPLOYEE
GROUP BY Dno;
```

Torke iz EMPLOYEE su podeljene u grupe - svaka grupa ima istu vrednost za atribut grupisanja Dno. Dakle, svaka grupa sadrži zaposlene koji rade u istom odeljenju. Funkcije COUNT i AVG se primenjuju na svaku takvu grupu torki. Napomenimo da SELECT klauzula uključuje samo atribut grupisanja i sumarne funkcije koje se primenjuju na svaku grupu torki. Slika 6.2. ilustruje kako se vrši grupisanje u primeru 5.2, a takođe pokazuje i rezultat upita.

Fname	Minit	Lname	Ssn	...	Salary	Super_ssn	Dno
John	B	Smith	123456789		30000	333445555	5
Franklin	T	Wong	333445555		40000	888665555	5
Ramesh	K	Narayan	666884444		38000	333445555	5
Joyce	A	English	453453453		25000	333445555	5
Alicia	J	Zelaya	999887777		25000	987654321	4
Jennifer	S	Wallace	987654321		43000	888665555	4
Ahmad	V	Jabbar	987987987		25000	987654321	4
James	E	Bong	888665555		55000	NULL	1

Dno	Count (*)	Avg (Salary)
5	4	33250
4	3	31000
1	1	55000

Result of Q24

Slika 6.2 Grupisanje torki EMPLOYEE po vrednosti Dno [Izvor: NM-IT350-2020/2021.]

Ako u atributu grupisanja postoje NULL vrednosti, onda se kreira posebna grupa za sve torke sa NULL vrednostima u atributu grupisanja. Na primer, ako je tabela EMPLOYEE imala neke torke koje su imale NULL za atribut grupisanja Dno, postojala bi posebna grupa za te torke u rezultatu.

KLAUZULA GROUP BY-GRUPISANJE PO VIŠE KOLONA

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.

Primer 5.3: Izračunati broj radnika koji obavljaju različiti posao unutar svake RJ.

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

Iz ove SELECT naredbe se dobija sledeći rezultat (slika 6.3):

Primer prikazuje još jedno korišćenje klauzule GROUP BY.

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

Slika 6.3 Rezultat za SELECT primer 5.3 [Izvor: Autor]

KLAUZULA HAVING

Čemu služi klauzula HAVING?

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

Ponekad želimo da dođemo do vrednosti sumarnih funkcija samo za grupe koje zadovoljavaju određene uslove. Na primer, pretpostavimo da želimo prikazati koje poslove obavlja više od jednog radnika u svakoj RJ. SQL obezbeđuje HAVING klauzulu, koja se može pojaviti u vezi sa GROUP BY klauzulom, u tu svrhu.

HAVING-om se postavlja uslov nad sumarnim informacijama koji se odnosi na grupe torki formirane za svaku vrednost atributa grupisanja. Samo grupe koje zadovoljavaju uslove se dobijaju u rezultatu upita.

Primer 5.4: Prikazati koje poslove obavlja više od jednog radnika u svakoj RJ.

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

Iz ove SELECT naredbe se dobija sledeći rezultat (slika 6.4):

	S_RJ	POSAO	COUNT(*)
▶	20	ANALITICAR	2
	20	SAVETNIK	2
	30	TRG PUTNIK	4

Slika 6.4 Rezultat za SELECT primer 5.4 [Izvor: Autor]

KLAUZULA ORDER BY

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

Primenom 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 5.5: Prikaži ime, posao i LD radnika u PS-u 30 uređene u rastućem redosledu poslova i opadajućem redosledu ličnih dohodaka (Primer 5.4).

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

Iz ove SELECT naredbe se dobija sledeći rezultat (slika 6.5):

	IMEPREZIME	POSLO	LD
▶	JANKO J	ANALITICAR	95000
	ANA A	RUKOVODILAC	285000
	MILAN M	TRG PUTNIK	160000
	GORAN G	TRG PUTNIK	150000
	PETAR P	TRG PUTNIK	125000
	MARIJA M	TRG PUTNIK	125000

Slika 6.5 Rezultat za SELECT primer 5.5 [Izvor: NM IT250-2020/2021.]

U ovom primeru su poslovi radnika poredani po rastućem, a plata radika 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: GROUP BY, HAVING SA KORIŠĆENJEM ALIJASA

Primeri grupne funkcije GROUP BY i klauzule HAVING

Na slici 6.6 je prikazana tabela CUSTOMER sa primernim ključem Id.

CUSTOMER	
Id	PK
FirstName	
LastName	
City	
Country	
Phone	

Slika 6.6 Tabela CUSTOMER [Izvor: NM IT350-2020/2021.]

Zadatak 1: Nađi broj kupaca u svakoj zemlji. Izlistaj samo one zemlje kod kojih je broj kupaca veći od 10.

```
SELECT COUNT(Id), Country
FROM Customer
GROUP BY Country
HAVING COUNT(Id) > 10
```

Zadatak 2: Izlistaj broj kupaca u svakoj zemlji osim u USA, sortiran od najvećeg do najmanjeg. Uključi samo zemlje sa 9 ili više kupaca.

```
SELECT COUNT(Id), Country
FROM Customer
```

```
WHERE Country <> 'USA'
GROUP BY Country
HAVING COUNT(Id) >= 9
ORDER BY COUNT(Id) DESC
```

Opšta sintaksa za korišćenje alijasa je:

```
SELECT column-name AS alias-name
FROM table-name alias-name
WHERE condition
```

Zadatak 3: Izlistaj ukupan broj kupaca u svakoj zemlji. Rezultate prikaži tako da se lako razumeju zaglavlja kolona.

```
SELECT COUNT(C.Id) AS TotalCustomers, C.Country AS Nation
FROM Customer C
GROUP BY C.Country
```

TotalCustomers i Nation su alijasi kolona.

Alijas za tabelu (C) u ovom primeru nije posebno koristan.

Dobijeni rezultat iz naredbe SELECT je prikazan na slici 6.7.

TotalCustomers	Nation
3	Argentina
2	Austria
2	Belgium
9	Brazil
3	Canada
...	

Slika 6.7 Rezultat za SELECT za zadatak 4. [Izvor: NM IT350-2020/2021.]

ZADATAK 1.

1. zadatak upotrebe klauzule GROUP BY, HAVING i ORDER BY

Zadatak 1: 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.

Potrebna tabela i vrednosti;

```
CREATE TABLE `it350`.`proizvod` (
  `S_PROIZVOD` INT NOT NULL AUTO_INCREMENT,
```

```
`IME_PROIZVODA` VARCHAR(45) NULL,  
`KOLICINA` INT NULL,  
`MIN_CENA` INT NOT NULL,  
PRIMARY KEY (`S_PROIZVOD`)  
)ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

	S_PROIZVOD	IME_PROIZVODA	KOLICINA	MIN_CENA
►	1	BICIKL	2	900
	2	LOPTA	8	170
	3	REKET	7	200
	4	MAJKA	15	100
	5	MOTOCIKL	4	2000

Slika 6.8 Kreiranje tabele PROIZVOD [Izvor: Autor]

Upit kojim rešavamo traženi zadatak i njegov rezultat dati su na slici 7.

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

	IME_PROIZVODA	SUM(KOLICINA * MIN_CENA)
►	MOTOCIKL	8000
	BICIKL	1800

Slika 6.9 Rezultat za SELECT koji se traži u zadatku 1. [Izvor: Autor]

ZADATAK 2.

2. zadatak upotrebe klauzule GROUP BY, HAVING i ORDER BY

Zadatak 2: Kreirati izveštaj sa kolonama Ime_odeljenja, Lokacija, Ime_radnik iz tabele Radnik2 (Sifra_radnika, Ime_radnika, Ime_odeljenja, Lokacija, Zarada) u kojem su prikazani samo podaci o Radnicima koji ne rade u odeljenju PRODAJA 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.

Za kreiranje tražene tabele u zadatku potreban je sledeći SQL kod:

```
CREATE TABLE `it350`.`RADNIK2` (
  `SIFRA_RADNIKA` INT NOT NULL ,
  `IME_RADNIKA` VARCHAR(45) NULL ,
  `IME_ODELJENJA` VARCHAR(45) NULL ,
  `LOKACIJA` VARCHAR(45) NULL ,
  `ZARADA` INT NULL ,
  PRIMARY KEY (`SIFRA_RADNIKA`) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
```

Slika 6.10 prikazuje sadržaj tabele RADNIK2:

	SIFRA_RADNIKA	IME_RADNIKA	IME_ODELJENJA	LOKACIJA	ZARADA
▶	2250	ALEKSANDRA	PRERADA_MLEKA	HRVATSKA	5
	2276	DJORDJE	KUVAR	TUZLA	100000
	2415	BORIS	IT_DEPARTMAN	DORCOL	550
	5678	BOJAN	LEPLJENJE_ZICA	VISNJICA	15
	24001	VUK	LEMLJENJE_TABLI	ZARKOVO	850
	71556	ZAN-ZAK	TEGLJAC	FRANCUSKA	25

Slika 6.10 Sadržaj tabele RADNIK2 [Izvor: NM IT350-2020/2021.]

Upit za rešavanje zadatka 2.

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

Dobijeni rezultat je prikazan na slici 6.11:

	IME_ODELJENJA	LOKACIJA	IME_RADNIKA	ZARADA
▶	KUVAR	TUZLA	DJORDJE	100000

Slika 6.11 Rešenje zadatka 2. [Izvor: NM IT350-2020/2021.]

ZADATAK 3.

3. zadatak upotreba klauzula GROUP BY, HAVING i ORDER BY

Zadatak 3: 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).

Za kreiranje tražene tabele u zadatku potreban je sledeći SQL kod:

```
CREATE TABLE `it350`.`prodaja` (
  `SIFRA_PROIZVODA` INT NOT NULL ,
  `NAZIV_PROIZVODA` VARCHAR(45) NULL ,
  `IZNOS` INT NULL ,
  PRIMARY KEY (`SIFRA_PROIZVODA`) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
```

Na slici 6.12. prikazan je sadržaj tabele PRODAJA:

	SIFRA_PROIZVODA	NAZIV_PROIZVODA	IZNOS
▶	101	MARAMICE	30
	102	ZVAKE	15
	103	GRISINE	75
	104	COKOLADA	100
	105	LIZALICA	20

Slika 6.12 Sadržaj tabele PRODAJA [Izvor: NM IT350-2020/2021.]

```
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);
```

SELECT naredba za izvršenje upita u zadatku 3.

▼ Poglavlje 7

Upotreba NULL vrednosti u naredbi SELECT

PROVERA NULL VREDNOSTI

Vrši se uz pomoć klauzula NULL i NOT NULL

Ako posmatramo podatke o Goranu, trgovačkom putniku, u tabeli RADNIK, vidimo da u koloni PREMIJA ima vrednost NULL. 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. Provera NULL vrednosti vrši se na sledeći način:

Primer 6.1: Prikazati ime, posao i premiju radnika koji ne primaju premiju

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

Rezultat SELECT upita glasi (slika 7.1):

	IMEPREZIME	POS AO	PREMIJA
►	STEFAN S	ANALITICAR	NULL
	MARKO M	UKOVODILAC	NULL
	ANA A	UKOVODILAC	NULL
	IVAN I	UKOVODILAC	NULL
	PAVLE P	SAVETNIK	NULL
	JOVAN J	PRES EDNIK	NULL
	JELENA J	ANALITICAR	NULL
	JANKO J	ANALITICAR	NULL
	FILIP F	SAVETNIK	NULL
	DEJAN D	ANALITICAR	NULL

Slika 7.1 Rezultat za SELECT primer 6.1 [Izvor: NM IT250-2020/2021.]

Primer 6.2: Prikazati ime, posao i premiju radnika koji primaju premiju

```
SELECT IMEPREZIME; POSAO; PREMIJA
FROM RADNIK
WHERE PREMIJA IS NOT NULL;
```

Iz naredbe SELECT se dobija sledeći izlaz (slika 7.2):

	IMEPREZIME	POSAO	PREMIJA
▶	MILAN M	TRG PUTNIK	30000
	PETAR P	TRG PUTNIK	50000
	MARIJA M	TRG PUTNIK	140000
	GORAN G	TRG PUTNIK	0

Slika 7.2 Rezultat za SELECT primer 6.2 [Izvor: Autor]

UPOTREBA NULL VREDNOSTI U ARITMETIČKIM IZRAČUNAVANJIMA

NULL se ne koristi pri izračunavanju izraza i funkcija

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 (u Oracle ili ekvivalenti ove funkcije) koja privremeno menja NULL vrednost sa vrednošću za koju se sami odlučimo, tj. vrednošću koja je neutralna u odnosu na željenu operaciju.

Primer 6.3: Prikaži ukupnu mesečnu zaradu radnika u RJ 30

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

Izraz LD+NVL(PREMIJA,0) se izvršava se na sledeći način:

1. ukoliko je PREMIJA nedefinisana, NULL vrednost se zamenjuje sa nulom i sabira se sa ličnim dohotkom (LD),
2. 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 ne pojavljuje u rezultujućoj koloni jer je zbir konkretne vrednosti i nedefinisane vrednosti (nečeg što ne znamo šta je) uvek nedefinisana vrednost.

U MySQL-u, funkcija sa naziva COALESCE, mada možete koristiti i funkcije IFNULL koja se javlja u skoro svim dijalektima SQL-a.

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

Prethodnom naredbom SELECT se dobija rezultat prikazan na slici 7.3.

	IMEPREZIME	POSLO	LD	LD+COALESCE(PREMIJA, 0)
▶	MILAN M	TRG PUTNIK	160000	190000
	PETAR P	TRG PUTNIK	125000	175000
	MARIJA M	TRG PUTNIK	125000	265000
	ANA A	UKOVODILAC	285000	285000
	GORAN G	TRG PUTNIK	150000	150000
	JANKO J	ANALITICAR	95000	95000

Slika 7.3 Rezultat za SELECT primer 6.3 [Izvor: Autor]

PRIMER UPOTREBE NULL VREDNOSTI U ARITMETIČKIM IZRAČUNAVANJIMA

Ukoliko nisu eliminisane NULL vrednosti, uvek treba voditi računa o tome šta predstavlja rezultat aritmetičkog izraza ili funkcije

Primer 6.4: Za RJ 30 izračunaj srednji LD, srednju premiju, srednju mesečnu zaradu za sve radnike koji primaju premiju i srednju mesečnu zaradu za sve radnike

```
SELECT AVG(LD), AVG(PREMIJA), AVG(LD+PREMIJA),
       AVG(LD+COALESCE(PREMIJA,0))
FROM RADNIK
WHERE S_RJ = 30;
```

Naredba SELECT daje sledeći rezultat (slika 7.4):

	AVG(LD)	AVG(PREMIJA)	AVG(LD+PREMIJA)	AVG(LD+COALESCE(PREMIJA,0))
▶	156666.6667	55000.0000	195000.0000	193333.3333

Slika 7.4 Rezultat za SELECT primer 6.4 [Izvor: Autor]

Razmotrimo detaljnije vrednosti kolona rezultujuće tabele:

Vrednost 156666.667 kolone AVG(LD) dobijena je deljenjem sume ličnih dohodaka svih šest radnika na šest delova.

Vrednost 55000 kolone AVG(PREMIJA) dobijena je deljenjem sume premija onih radnika koji primaju premiju sa brojem radnika koji primaju premiju. To znači da je suma premija podeljena sa 4, jer radnici DRAGAN i GORAN imaju NULL vrednost u koloni PREMIJA i neće biti uključeni u dobijanje srednje vrednosti, odnosno n-torke koje odgovaraju tim radnicima biće eliminisane pre primene funkcije AVG.

Posebno treba obratiti pažnju na razliku rezultata funkcije

AVG((LD+PREMIJA) i AVG(LD+NVL(PREMIJA,0)).

Prva funkcija istu sumu deli na 4 dela (zbir ličnog dohotka i premije gde je vrednost premije nedefinisana je takođe nedefinisana, pa se te n-torke eliminišu pre primene funkcije AVG), a druga na 6 (nedefinisana vrednost premije se zamenjuje nulom i sabira sa ličnim dohotkom). Zato je rezultat prve funkcije veći od rezultata druge.

Ukoliko u fazi projektovanja nisu eliminisane NULL vrednosti, uvek treba voditi računa o tome šta predstavlja rezultat aritmetičkog izraza ili funkcije, odnosno koje n-torke su uključene u dobijanje tog rezultata.

▼ Poglavlje 8

Pokazna vežba

NAČIN ORGANIZACIJE INDIVIDUALNIH VEŽBI

Vežba je organizovana kroz uvod deo i deo za samostalni rad studenata

Vežba je organizovana kroz uvodni deo i deo za samostalni rad studenata.

1. U uvodnom delu pokaznih vežbi se daje pokazni primer koji studentima treba da pomogne u samostalnom rešavanju zadataka.
2. Zadatke koji su zadati za samostalni rad student samostalno rešava uz pomoć asistenta.

▼ 8.1 Pokazni primer - 1. deo

PRIKAZ TABELE STUDENTSKE BAZE DOSIJE - (5 MIN.)

Data je studentska baza i tabele DOSIJE, ISPIT

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

Slika 8.1.1 Sadržaj tabele DOSIJE [Izvor: NM IT350-2020/2021.]

Na osnovu primera sa predavanja, kreirati tabelu ISPIT.

PRIKAZ TABELE STUDENTSKE BAZE PREDMET I ISPITNI_ROK (5 MIN.)

Data je studentska baza i tabele PREDMET I 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 8.1.2 Sadržaj tabele Predmet [Izvor: NM IT350-2020/2021.]

Na osnovu primera sa predavanja, kreirati tabelu ISPITNI_ROK.

PRIMER UPITA BEZ WHERE KLAUZULE (5 MIN.)

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

Check All With selected: Change Delete Export

Number of rows: 25 Filter rows: Search this table

Slika 8.1.3 Primer 1-rešenje [Izvor: NM IT250-2020/2021.]

Primer 2. Napisati upit koji kao rezultat daje podatke o svim predmetima na fakultetu, od podataka izdvojiti id predmeta, naziv i broj bodova.

PRIMER UPITA SA WHERE KLAUZULOM (5 MIN.)

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_RODZENJA`="BEOGRAD"`

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	INDEKS	IME	PREZIME	GOD_RODZENJA	MESTO_RODZENJA
<input type="checkbox"/> Edit Copy Delete	20100021	Milos	Peric	1992	Beograd
<input type="checkbox"/> Edit Copy Delete	20100023	Sanja	Terzic	1991	Beograd

Check All With selected: Change Delete Export

Number of rows: 25 Filter rows: Search this table

Slika 8.1.4 Primer 3-rešenje [Izvor: NM IT350-2020/2021.]

Primer 4 . Napisati upit koji kao rezultat daje podatke o predmetima koji nosi izmedju 5 i 7 bodova, uključujući i njih.

PRIMERI UPITA UZ KORIŠĆENJE LOGIČKOG OPERATORA AND (5 MIN.)

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 8.1.5 Primer 1-rešenje [Izvor: NM IT350-2020/2021.]

Primer 2. Napisati upit kao rezultat daje podatke o položenim ispitima u januarskom 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 8.1.6 Primer 2-rešenje [Izvor: NM IT350-2020/2021.]

PRIMER UPITA UZ KORIŠĆENJE LOGIČKOG OPERATORA OR (5 MIN.)

Vežbanje korišćenja logičkih operatora OR

Primer 3. Napisati upit koji kao rezultat daje podatke o studentima koji se zovu Milos ili su iz Kraljeva.

```
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"/> Edit Copy Delete	20100021	Milos	Peric	1992	Beograd
<input type="checkbox"/> Edit Copy Delete	20100022	Marijana	Savkovic	1993	Kraljevo
<input type="checkbox"/> Edit Copy Delete	20100025	Ljubica	Savkovic	1991	Kraljevo

Check All With selected: Change Delete Export

Number of rows: 25 Filter rows: Search this table

Slika 8.1.7 Primer 3-rešenje [Izvor: NM IT350-2020/2021.]

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"/> Edit Copy Delete	20100021	1001	2011	jan	9	2020-01-20
<input type="checkbox"/> Edit Copy Delete	20100021	1021	2011	apr	7	2003-04-20
<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	20100023	1001	2011	jan	8	2020-01-20
<input type="checkbox"/> Edit Copy Delete	20100024	1001	2011	jan	10	2020-01-20
<input type="checkbox"/> Edit Copy Delete	20100025	1001	2011	jan	6	2020-01-20
<input type="checkbox"/> Edit Copy Delete	20100026	1001	2011	feb	7	2010-02-20
<input type="checkbox"/> Edit Copy Delete	20100026	1001	2011	jan	5	2020-01-20

Check All With selected: Change Delete Export

Slika 8.1.8 Primer 4-rešenje [Izvor: NM IT350-2020/2021.]

PRIMER UPITA UZ KORIŠĆENJE OPERATORA != (5 MIN.)

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 <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2001	Programiranje 1	8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2002	Programiranje 2	8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3001	Engleski jezik 1	5
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3002	Engleski jezik 2	5
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4001	Uvod u organizaciju racunara	5
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> 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 8.1.9 Primer 5-rešenje [Izvor: NM IT350-2020/2021.]

PRIMERI KORIŠĆENJA OPERATORA LIKE (5 MIN.)

Vežbanje operatora LIKE

```
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 8.1.10 Primer 1-rešenje [Izvor: NM IT350-2020/2021.]

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 8.1.11 Primer 2-rešenje [Izvor: NM IT350-2020/2021.]

JOŠ JEDAN PRIMER KORIŠĆENJA OPERATORA LIKE (5 MIN.)

Primer za operator 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

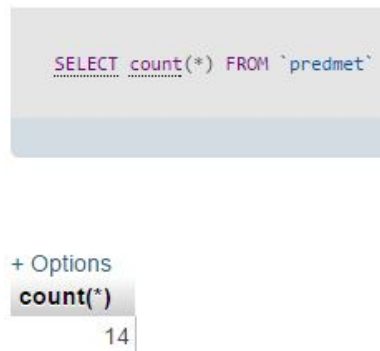
Slika 8.1.12 Primer 3-rešenje [Izvor: NM IT350-2020/2021.]

▼ 8.2 Pokazni primer - 2. deo

PRIMERI KORIŠĆENJA SUMARNIH FUNKCIJA (8 MIN.)

Vežbanje korišćenja sumarnih funkcija

Primer 1. Prebrojati koliko se predmeta na Fakultetu predaje.



Slika 8.2.1 Primer 1-rešenje [Izvor: NM IT350-2020/2021.]

Primer 2. Pronaći najmanji i najveći broj indeksa studenata upisanih na Fakultet.



Slika 8.2.2 Primer 2-rešenje [Izvor: NM IT350-2020/2021.]

PRIMER KORIŠĆENJA FUNKCIJE AVG (8 MIN.)

Vežbanje upita sa agregatnim funkcijama

Primer 3. Napisati upit koji izračunava prosečnu ocenu na premetu sa id-jem 2001 u januarskom roku 2011.



Slika 8.2.3 Primer 3-rešenje [Izvor: NM IT350-2020/2021.]

VIDEO

Objašnjenja naredbe SELECT - Consultas SQL (Select, Where, group by, having, order by, join)

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

PRIMERI KORIŠĆENJA KLAUZULE GROUP BY NA STUDENTSKOJ BAZI (13 MIN.)

Prvi primer: 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(*)
1001	6
2001	6
1021	4
3001	3

Slika 8.2.4 Primer 1-rešenje

[Izvor: NM IT350-2020/2021.]

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 manja 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 8.2.5 Primer 2-rešenje [Izvor: NM IT350-2020/2021.]

JOŠ JEDAN PRIMER KORIŠĆENJA KLAUZULE GROUP BY NA STUDENTSKOJ BAZI (8 MIN.)

Dat je primer 3

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 8.2.6 . Primer 3-rešenje [Izvor: NM IT350-2020/2021.]

PRIMER KORIŠĆENJA NULL VREDNOSTI (8 MIN.)

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_rođenja is null
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

INDEKS	IME	PREZIME	GOD_ROĐENJA	MESTO_ROĐENJA
20100024	Nikola	Vukovic	1992	NULL
20100027	Milena	Stankovic	0	NULL

Slika 8.2.7 Primer 1-rešenje [Izvor: NM IT350-2020/2021.]

VIDEO II

SQL Tutorial - 14: The SELECT Query

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

▼ 8.3 Zadaci za samostalni rad

ZADACI ZA SAMOSTALNI RAD - VREME IZRADE 75 MIN.

Samostalno rešite sledeće zadatke:

1. Napisati upit koji kao rezultat daje podatke o svim studentima na fakultetu, od podataka izdvojiti indeks, ime i prezime i mesto i datum rođenja.
2. Napisati upit koji kao rezultat daje podatke sve o ispitnim rokovima između 2011 i 2012 godine uključujući i njih.
3. Napisati upit koji kao rezultat daje podatke o predmetima koji u nazivu imaju "matematika" i istovremeno nose 6 ili manje bodova.
4. Napisati upit koji prikazuje id_predmeta, naziv i broj bodova za predmete kojima je broj poena različit od 5.
5. Prikazati podatke o predmetima koji u svom nazivu na bilo kom mestu imaju cifru 2 i nose 6 ili više bodova.

Za rešavanje prvih 5 zadataka potrebno je 30 minuta

6. Pronaći prvi sledeći u odnosu na najveći broj indeksa studenata upisanih na Fakultet.
7. Napisati upit koji izračunava najveću i prosečnu ocenu na predmetu "Programiranje 1" po svakoj školskoj godini i svakom ispitnom roku.
8. Napisati upit koji izračunava najveću i prosečnu ocenu po svakoj školskoj godini, po svakom ispitnom roku i svakom predmetu
9. Napisati upit koji prikazuje koliko je studenata položilo svaki od predmeta koji se na Fakultetu predaje po svakoj školskoj godini sortiran u rastućem redosledu po školskoj godini i opadajućem redosledu po broju studenata.
10. Napisati upit koji za predmete računa prosečnu ocenu studenata koji su taj predmet položili. Ne treba prikazati one predmete koje je položilo manje od 3 studenta.

Za rešavanje preostalih 5 zadataka potrebno je 45 minuta.

▼ Poglavlje 9

Domaći zadatak

DOMAĆI ZADATAK 9 - VREME IZRADE 90 MIN.

Zadatak snimate u dokument IT250-DZ09-Ime_Prezime_brIndexa.sql gde su Ime, Prezime i brIndexa vaši podaci. Dokument poslati predmetnom asistentu.

Napišite i izvršite tri od 15 navedenih upita. Baza podataka nad kojoj treba izvršavati upite je u prilogu lekcije. Redni brojevi zadatka koje student dobija se računaju po formulama:

1. zadatak = (Broj indeksa mod 30) + 1

2. zadatak = ((Broj indeksa + Redni broj 1. zadatka) mod 30) + 1

3. zadatak = ((Broj indeksa + Redni broj 2. zadatka) mod 30) + 1

Rešene zadatke sačuvati kao .SQL fajl, pod nazivom IT250-DZ09-Broj_indksa-Ime_Prezime.sql.

1. Napisati upit kojim se prikazuju informacije o svim studentima na fakultetu. Potrebno je prikazati broj indeksa, ime, prezime I smer studenta.

2. Napisati upit kojim se prikazuju informacije o predmetima koji se drže na fakultetu. Potrebno je prikazati ID predmeta, naziv predmeta I broj ESPB-a koji taj predmet nosi.

3. Napisati upit kojim se prikazuju informacije o profesorima koji predaju na fakultetu. Potrebno je prikazati ime I prezime profesora, kao I njegovo zvanje.

4. Napisati upit kojim se prikazuju informacije o smerovima na fakultetu. Prikazati samo naziv smera I opis.

5. Napisati upit kojim se prikazuju podaci o onim predmetima koji vrede više od 6 ESPB-a.

6. Napisati upit kojim se prikazuju podaci o onim predmetima koji vrede manje od 10 ESPB-a.

7. Napisati upit kojim se prikazuju podaci o onim predmetima koji vrede između 5 i 8 ESPB-a.

8. Napisati upit kojim se prikazuju podaci samo o onim predmetima koji vrede 5, 7 i 9 ESPB-a.

9. Napisati upit kojim se prikazuju podaci samo o zaposlenim predavačima koji su profesori.

10. Napisati upit kojim se prikazuju podaci o onim predavačima koji su ili saradnici ili asistenti

11. Napisati upit kojim se prikazuju podaci o onim predavačima koji nisu ili saradnici ili asistenti.

12. Napisati upit kojim se prikazuju podaci o predmetima koji imaju više od 2 časa vežbanja.

13. Napisati upit kojim se prikazuju podaci o predmetima koji imaju manje od 3 časa vežbanja.

14. Napisati upit kojim se prikazuju podaci samo o predmetima koji imaju 2 ili 3 časa vežbanja.

15. Napisati upit kojim se prikazuju podaci o studentima kojima je broj indeksa veći od 1200 I manji od 2500. Prikazati broj indeksa, ime, prezime, tip studiranja.

16. Napisati upit kojim se prikazuju podaci o studentima kojima je broj indeksa veći od 1001. Prikazati broj indeksa, ime, prezime, tip studiranja.

PREOSTALI ZADACI ZA DZ09

Dati su preostali zadaci za domaći rad

17. Napisati upit koji prikazuje id_predmeta, naziv i broj bodova za predmete kojima je broj poena različit od 5.
18. Napisati upit kojim se prikazuju podaci o studentima kojima je broj indeksa manji od 1500. Prikazati broj indeksa, ime, prezime, tip studiranja.
19. Napisati upit koji kao rezultat daje podatke o svim studentima na fakultetu, od podataka izdvojiti indeks, ime i prezime i mesto i datum rođenja.
20. Napisati upit koji kao rezultat daje podatke sve o ispitnim rokovima između 2011 i 2012 godine uključujući i njih.
21. Napisati upit koji kao rezultat daje podatke o predmetima koji u nazivu imaju "matematika" i istovremeno nose 6 ili manje bodova.
22. Prikazati podatke o predmetima koji u svom nazivu na bilo kom mestu imaju cifru 2 i nose 6 ili više bodova.
23. Napisati upit koji izračunava najveću i prosečnu ocenu na predmetu "Programiranje 1" po svakoj školskoj godini i svakom ispitnom roku.
24. Napisati upit koji izračunava najveću i prosečnu ocenu po svakoj školskoj godini, po svakom ispitnom roku i svakom predmetu
25. Napisati upit koji prikazuje koliko je studenata položilo svaki od predmeta koji se na Fakultetu predaje po svakoj školskoj godini sortiran u rastućem redosledu po školskoj godini i opadajućem redosledu po broju studenata.
26. Napisati upit koji za predmete računa prosečnu ocenu studenata koji su taj predmet položili. Ne treba prikazati one predmete koje je položilo manje od 3 studenata.
27. Napisati upit kojim se prikazuju podaci o Internet studentima. Prikazati indeks, ime, prezime i datum rođenja za svakog internet studenta.
28. Napisati upit kojim se prikazuju podaci o studentima rođenim u Beogradu. Prikazati indeks, ime, prezime, tip studiranja.
29. Napisati upit kojim se prikazuju podaci o svim studentima rođenim u Mladenovcu ili Kraljevu.
30. Napisati upit kojim se prikazuju podaci o studentima koju su rođeni u Beogradu i tip studiranja je Tradicionalni.

UPUTSTVO ZA REŠAVANJE ZADATKA

Treba se pridržavati sledećeg uputstva

Prilikom slanja domaćih zadataka, neophodno je da ispunite sledeće:

- Subject mail-a mora biti IT250-DZbr (u slučaju kada šaljete domaći za ovu nedelju to je IT250-DZ09)
- U prilogu mail-a treba da se nalazi arhiviran projekat koji se ocenjuje imenovan na sledeći način: IT250-DZbr-BrojIndeksa-Ime Prezime. Na primer, IT250-DZ09-1234-VeljkoGrkovic
- Telo mail-a treba da ima pozdravnu poruku

- Arhivu sa zadatkom poslati na adresu predmetnog asistenta:
milica.vlajkovic@metropolitan.ac.rs (studenti u Beogradu i online studenti) ili
tamara.vukadinovic@metropolitan.ac.rs (studenti u Nišu).

Svi poslati mail-ovi koji ne ispunjavaju navedene uslove NEĆE biti pregledani. Za sva pitanja ili nedoumice u vezi zadatka, možete se obratiti asistentu

▼ Zaključak

ZAKLJUČAK

Šta smo naučili u ovoj lekciji?

U ovoj lekciji se govorilo o različitim primerima primene naredbe SELECT koja se može koristiti za rad nad jednom ili više tabela. Z. O vde 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 govorilo 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 u radu sa bazama podataka.

LITERATURA:

Za pisanje ove lekcije korišćena je sledeća literatura

1. C. J. Date, An introduction to Database Systems, Addison-Wesley Publishing Company, 1990
2. https://www.w3schools.com/SQL/sql_select.asp
3. https://www.w3schools.com/SQL/sql_distinct.asp
4. https://www.w3schools.com/SQL/sql_where.asp
5. <https://www.w3resource.com/sql/boolean-operator/sql-boolean-operators.php>
6. <https://www.w3resource.com/sql/arithmetic-operators/sql-arithmetic-operators.php>
7. https://www.w3schools.com/SQL/sql_min_max.asp
8. https://www.w3schools.com/SQL/sql_count_avg_sum.asp
9. https://www.w3schools.com/SQL/sql_groupby.asp
10. https://www.w3schools.com/SQL/sql_having.asp
11. https://www.w3schools.com/SQL/sql_orderby.asp
12. https://www.w3schools.com/SQL/sql_isnull.asp