

Zadaci za vježbu

(uz predavanje 4 - SQL)

Funkcije i operatori

Prvih 7 zadataka radit će se nad tablicom **upl_ispl** čija se skripta za stvaranje i punjenje nalazi u nastavku. Preostala tri zadatka odnose se na tablicu STUDENT iz baze STUDADMIN.

Tablica **upl_ispl** čuva podatke o računima koji su jedinstveno označeni svojom šifrom. Račun se kreira odnosno otvara na određeni datum te se uz njega vezuje određena uplata i određena isplata.

sifra_racuna	datum_otvaranja	upl_iznos	ispl_iznos
123456	2006.02.01	120,00	43,00
134567	2006.05.08	173,47	115,21
145678	2006.07.07	83,20	345,21
156789	2006.09.12	205,11	17,65
167890	2006.10.28	365,11	100,23
234567	2007.02.01	28,10	200,02
245678	2007.03.12	110,05	45,63

Skripta za kreiranje i punjenje gore navedene tablice (pokrenite redom dolje navedene naredbe):

```
CREATE TABLE upl_ispl(sifra_racuna INTEGER, datum_otvaranja DATE,  
upl_iznos DECIMAL(9,2), ispl_iznos DECIMAL(9,2));
```

```
INSERT INTO upl_ispl VALUES (123456,'1.2.2006', 120.00, 43.00);  
INSERT INTO upl_ispl VALUES (134567,'8.5.2006', 173.47,115.21);  
INSERT INTO upl_ispl VALUES (145678,'7.7.2006', 83.20,345.21);  
INSERT INTO upl_ispl VALUES (156789,'12.9.2006', 205.11,17.65);  
INSERT INTO upl_ispl VALUES (167890,'28.10.2006', 365.11,100.23);  
INSERT INTO upl_ispl VALUES (234567,'1.2.2007', 28.10,200.02);  
INSERT INTO upl_ispl VALUES (245678,'12.3.2007', 110.05, 45.63);
```

1. Napišite SELECT naredbu kojom će se ispisati šifra računa i apsolutna vrijednost prometa (razlika uplaćenog i isplaćenog iznosa) za svaki račun o kojem se unutar zadane tablice vodi evidencija.
2. Napišite SELECT naredbu koja će ispisati šifru računa i uplaćeni iznos, ali samo ako nije prošlo više od osamnaest mjeseci (540 dana) od trenutka otvaranja računa do danas. Upit mora biti neovisan o trenutku izvođenja.
3. Napišite SELECT naredbu koja će ispisati šifru računa te iznos uplate, ali tako da se broj uplaćenih kuna stavlja u zaseban stupac nazvan "uplaceno_kuna", a preostali iznos lipa u stupac "uplaceno_lipa".
4. Napišite SELECT naredbu kojom će se ispisati šifra računa, mjesec kad je račun otvoren, uplaćen i isplaćen iznos. Uplaćen i isplaćen iznos spojiti u jedinstven znakovni niz odvojen crticom bez suvišnih praznih mjesta na početku i završetku.
npr. "120.00-43.00".
5. Napišite SELECT naredbu koja će ispisati sve podatke o računima stvorenim u dane vikenda (subota i nedjelja) ako takvi računi postoje.
6. Što će se ispisati sljedećom SELECT naredbom?

```
SELECT DISTINCT
    DAY(datum_otvaranja) as dan
    ,MONTH(datum_otvaranja) as mjesec
    ,YEAR(datum_otvaranja) as godina
FROM upl_ispl
WHERE (SUBSTRING(sifra_racuna FROM 1 FOR 1)=2);
```

7. Što će se ispisati sljedećom SELECT naredbom?

```
SELECT sifra_racuna
    , datum_otvaranja - MDY(MONTH(datum_otvaranja),1,2007)
    AS dana_od_pocetka_mjeseca
FROM upl_ispl
WHERE YEAR(datum_otvaranja)=2007;
```

Napomena: Sljedeći se zadaci odnose na STUDADMIN bazu i to na tablicu STUDENT.

8. Što će se ispisati sljedećom SELECT naredbom?

```
SELECT FIRST 5 UPPER(ime)
    , LOWER(Prezime)
FROM student
WHERE YEAR(datrod)<1978;
```

9. Što će se ispisati sljedećom SELECT naredbom?

```
SELECT DISTINCT LOWER(ime)
FROM student
WHERE CHAR_LENGTH(TRIM(ime))<4;
```

10. Što će se ispisati sljedećom SELECT naredbom?

```
SELECT USER
      , prezime
      , YEAR(datrod)
FROM student WHERE SUBSTRING(jmbg FROM 5 FOR 3)='977';
```

Napomena: prvih 7 znamenaka JMBG-a pokazuje datum rođenja osobe čiji je JMBG, s time da se tisućljeće ne prikazuje. Tako se iz JMBG-a: 1207988..... može isčitati da je osoba rođena 12. srpnja 1988. godine.

Where clause, between, in, is null, like, select case

1. Napisati upit koji će vratiti imena i prezimena svih nastavnika koji su se zaposlili u intervalu [15.1.2002, 15.7.2002] i koji više nisu u radnom odnosu (datumzaposlendo je poznat, tj. nije NULL).
2. Napisati upit koji će vratiti sve nazive organizacijskih jedinica (svaki naziv samo jednom) u kojima je zaposlen barem jedan nastavnik (datumzaposlendo je nepoznat, tj. NULL) kojemu ime završava na "a" (npr. "Ivana", "Vanja").
3. Napisati upit koji će dohvatiti sve nazive predmeta koji sadrže niz 'teorij' bez obzira na velika i mala slova (npr. 'Teorija informacije', 'Informacijska teorija' ili 'TEORIJA INFORMACIJA').
4. Napisati upit koji će vratiti sve matične brojeve (svaki jmbag samo jednom) studenata koji su položili predmet koji u nazivu sadrži niz 'matematika' (bez obzira na velika i mala slova, npr. 'Matematika I', 'Statistika i matematika') i koji su iz tog predmeta dobili ocjenu dovoljan ili vrlo dobar (koristiti IN operator).
5. Napisati upit koji će za sve studente koji su upisali predmet koji u nazivu sadrži niz 'matematika' (bez obzira na velika i mala slova) ispisati matični broj, naziv predmeta i status. Status odrediti na sljedeći način:
 - ako je ocjena nedovoljan onda je status 'pad'
 - ako je ocjena pozitivna onda je status 'prolaz'
 - ako ocjena nije evidentirana (NULL) onda je status 'nepoznato'Svaki matični broj, naziv i status ispisati samo jednom (projekcija).
Npr. rezultat može izgledati:

jmbag	naziv	status
0036342145	Matematika I	prolaz
0036342145	Matematika II	pad

6. Što će se ispisati sljedećim upitima ?

upisanpredmet

jmbag	sifpredmet	akgodina	ozngrupa	datumocjena	ocjena
0555001157	26	2005	D-A5	(null)	(null)
0555001157	27	2005	D-A5	(null)	(null)
0555001115	2	2003	C-A2	02.02.2004	2
0555001211	2	2003	C-B2	02.02.2004	5

a)

```
SELECT *  
  FROM upisanpredmet  
 WHERE ocjena IN (NULL);
```

b)

```
SELECT *
  FROM upisanpredmet
 WHERE ocjena IS NULL;
```

c)

```
SELECT *
  FROM upisanpredmet
 WHERE ocjena IN (NULL, 5);
```

d)

```
SELECT *
  FROM upisanpredmet
 WHERE ocjena BETWEEN NULL AND 5;
```

Opišite riječima što će se ispisati sljedećim upitima:

e)

```
SELECT *
  FROM student
 WHERE ime BETWEEN 'I%' AND 'J%';
```

f)

```
SELECT *
  FROM student
 WHERE ime BETWEEN 'I' AND 'J';
```

g)

```
SELECT *
  FROM student
 WHERE ime >= 'I' AND ime <= 'J';
```

7. Što će ispisati sljedeći upiti:

student

mbr	Ime
100	Ivan
101	Ivana
102	Ivano

a)

```
SELECT *
  FROM student
 WHERE upper(ime) LIKE 'IVAN_';
```

b)

```
SELECT *
  FROM student
 WHERE upper(ime) LIKE 'IVAN%';
```

8. Što će ispisati sljedeći upiti:

mjerenja

velicina	iznos
Tlak	1012hPa
Temperatura	25 C
Vlažnost	80%
Vjetar	2-3 NW
CO2	(null)

a)

```
SELECT velicina, iznos,
       CASE
         WHEN(iznos LIKE '%#%' ESCAPE '#') THEN 'Postotak'
         ELSE 'Apsolutna vrijednost'
       END as tip,
       CASE
         WHEN NOT(iznos LIKE '%#%' ESCAPE '#') THEN 'Apsolutna
vrijednost'
         ELSE 'Postotak'
       END as tip2
FROM mjerenja;
```

b)

```
SELECT velicina, iznos,
       CASE
         WHEN(iznos LIKE '%8%' ESCAPE '8') THEN 'Postotak'
         ELSE 'Apsolutna vrijednost'
       END as tip,
       CASE
         WHEN NOT(iznos LIKE '%8%' ESCAPE '8') THEN 'Apsolutna
vrijednost'
         ELSE 'Postotak'
       END as tip2
FROM mjerenja;
```

c)

```
SELECT velicina, iznos,
       CASE
         WHEN(iznos LIKE '___%' ESCAPE '%') THEN 'Postotak'
         ELSE 'Apsolutna vrijednost'
       END as tip,
       CASE
         WHEN NOT(iznos LIKE '___%' ESCAPE '%') THEN 'Apsolutna
vrijednost'
         ELSE 'Postotak'
       END as tip2
FROM mjerenja;
```

9. Opišite riječima što će vratiti sljedeći SQL upit:

```
SELECT orgjed.*
FROM orgjed
JOIN orgjed as nadorgjed
ON orgjed.sifnadorgjed = nadorgjed.siforgjed
WHERE nadorgjed.sifnadorgjed IS NULL
AND LOWER(nadorgjed.nazorgjed) LIKE '%vele%';
```

10. Što će vratiti sljedeći SQL upit:

orgjed

siforgjed	nazorgjed	sifnadorgjed
1	FER	(null)
2	ZPR	1
3	ZOEEM	1
4	ZPM	1
5	PMF	(null)

```
SELECT orgjed.nazorgjed,
CASE
    WHEN(orgjed.sifnadorgjed IS NULL) THEN 'nema nadređenu'
    ELSE 'ima nadređenu, to je:' || nadorgjed.nazorgjed
END as nadorgjed
FROM orgjed, orgjed nadorgjed
WHERE orgjed.sifnadorgjed = nadorgjed.siforgjed;
```

UNION, NULL, OUTER, ORDER BY

U zadacima iz ovog poglavlja koriste se relacije iz baze podataka **studAdmin**. Detaljnije objašnjenje **studAdmin** baze podataka možete pronaći na web stranicama predmeta.

1. Ispisati sve podatke o studentima, te ih poredati po datumu rođenja studenata. Mlađi studenti se nalaze bliže početku liste, a stariji prema kraju.
2. Ispisati nazive svih upisanih predmeta i ocjene dobivene iz tih predmeta. Rezultate poredati po nazivu predmeta uzlazno, zatim po dobivenim ocjenama silazno. U ispisu se moraju pojaviti i upisani predmeti koji nemaju ocjenu.
3. Ispisati sve podatke o studentima kojima je nepoznata jedna od vrijednosti: datum rođenja, mjesto rođenja i mjesto stanovanja.
4. Napisati naredbu kojom ćete unijeti u relaciju student zapis o studentu za kojeg imate sljedeće podatke:
jmbag = 0036368145
ime = Ante
prezime = Dragan
spol = 'M'
datum rođenja = '12.01.989.'
5. Ispisati imena i prezimena petorice nastavnika koji se nalaze na početku liste svih nastavnika poredanih silazno po iznosu koeficijenta.
6. Napisati jednu SQL naredbu kojom će se ispisati imena svih studenata rođenih u Dubrovniku i imena svih nastavnika koji stanuju u Dubrovniku. (Ista imena se mogu pojavljivati više puta u rezultatu.)
7. Ispisati sve različite poštanske brojeve i pripadajuće nazive mjesta u kojima stanuje barem jedan student ili barem jedan nastavnik. Rezultate poredati abecedno po nazivu mjesta.
8. Ispisati oznake dvorana i njihov kapacitet, te uz te podatke dodatno ispisati akademsku godinu i oznaku grupa kojim se nastava održavala u tim dvoranama. Dvorane u kojima se nije izvodila nastava, moraju se svejedno pojaviti u ispisu, te u tom slučaju za podatke o grupama staviti NULL vrijednosti.
9. Ispisati šifre i nazive organizacijskih jedinica kojima je šifra nadređene organizacijske jedinice jednaka 36, te dodatno ispisati šifre nastavnika koji su u tim organizacijskim jedinicama zaposleni. Ukoliko u takvim

organizacijskim jedinicama nema zaposlenih nastavnika, svejedno se moraju pojaviti u ispisu, te u tom slučaju podaci o nastavnicima će biti NULL vrijednosti.

- 10.** Ispisati JMBAG studenta i sve ocjene koje je student dobio iz predmeta 'Inteligentni sustavi'. (Student može imati samo jednu pozitivnu ocjenu iz predmeta, ali može imati i negativnu ocjenu.) U ispisu se moraju pojaviti i oni studenti koji nisu nikad upisali taj predmet, te u tom slučaju za vrijednost ocjene postaviti NULL vrijednost. Rezultate poredati po JMBAG-u uzlazno, pa potom silazno po ocjeni studenta.
- 11.** Ispisati imena i prezimena svih studenata, naziv mjesta u kojem stanuju i naziv županije kojoj mjesto pripada. Moraju se u ispisu pojaviti svi studenti, bez obzira jesu li im evidentirani podaci o mjestu stanovanja, kao i županija mjesta stanovanja.
- 12.** Ispisati ime i prezime svih studenata rođenih u lipnju. Dodatno, za svakog takvog studenta ispisati akademsku godinu u kojoj je upisao predmet i naziv predmeta iz kojeg je dobio ocjenu 4 ili 5. Ukoliko student nema takvih upisanih predmeta za akademsku godinu, kao i naziv ispisati NULL vrijednost.
- 13.** Za sve studente ispisati ime studenta, prezime studenta i prezime svih studenata koji imaju jednako ime i rođeni su iste godine. Ukoliko nekakav student nema svog imenjaka rođenog iste godine, kao prezime ispisati NULL vrijednost. Rezultate poredati abecedno po imenu i prezimenu studenta.

Paraleno spajanje, reflektivno spajanje, GROUP BY, HAVING

U zadacima iz ovog poglavlja koriste se relacije iz baze podataka **studAdmin**. Detaljnije objašnjenje **studAdmin** baze podataka možete pronaći na web stranicama predmeta.

1. Napišite SQL naredbu kojom će se ispisati JMBAG te imena i prezimena onih studenata koji stanuju u županiji u kojoj su rođeni.
2. Napišite SQL naredbu kojom će za svakog studenta ispisati JMBAG, ime, prezime i datum rođenja te JMBAG, datum rođenja, prezimena i imena onih studenata koji su od njega stariji a inicijali imena i prezimena su im jednaki. Za studenta starijeg od svih ostalih nije potrebno ispisivati podatke.

3. Zadana je relacijska shema

ISPIT {sifPredmet, JMBAG, datumRok, sifNastavnikPismeni, ocjenaPismeni, sifNastavnikUsmeni, ocjenaUsmeni, datumOcjenaUsmeni}

u kojoj se prate izlasci studenta na ispit iz određenog predmeta. Vrijednosti atributa sifNastavnikPismeni, ocjenaPismeni, sifNastavnikUsmeni, ocjenaUsmeni, datumOcjenaUsmeni ne moraju uvijek postojati/biti poznate. Domene atributa relacije **ispit** jednake su domenama korespondentnih atributa relacije **upisanPredmet**.

Napišite SQL naredbe čijim izvođenjem će u bazi **studAdmin** biti kreirana relacija **ispit** gore zadane relacijske sheme sljedećeg sadržaja:

sif Predmet	JMBAG	datum Rok	sifNastavnik Pismeni	ocjena Pismeni	sifNastavnik Usmeni	ocjena Usmeni	datumOcjena Usmeni
5	0555000048	02.02.2007	271	2	268	1	05.02.2007
5	0555000032	02.02.2007	267	2	271	1	05.02.2007
14	0555000048	05.02.2007	267	3	268	4	08.02.2007
5	0555000032	02.02.2007	267	4	271	5	05.02.2007
40	0555000688	02.02.2007	268	4	268	5	05.02.2007

4. Napišite SQL naredbu kojom će se za svaki zapis iz relacije ispit ispisati redak sljedećeg sadržaja:

JMBAG	prezimeStudenta	imeStudenta	nazivPredmeta
	ocjenioPismenilme	ocjenioPismeniPrezime	ocjenaPismeni
	ocjenioUsmenilme	ocjenioUsmeniPrezime	ocjenaUsmeni

5. Napišite SQL naredbu kojom će se, za one ispite kojima su pismeni i usmeni dio ispita ocijenili različiti nastavnici koji stanuju u istom mjestu ispisati redak sljedećeg sadržaja:

JMBAG	prezimeStudenta	imeStudenta	nazivPredmeta
	ocjenaPismeni	ocjenaUsmeni	

6. Napišite SQL naredbu kojom će se ispisati JMBAG, imena i prezimena onih studenata koji su polagali ispit iz predmeta 'Baze podataka', koji ne stanuju u mjestu u kojem su rođeni ali im mjesto rođenja i mjesto stanovanja pripadaju istoj županiji. Rezultat mora biti relacija.

7. U relaciju **ispit** dodajte n-torku **t1** sljedećeg sadržaja:
 t1 = < 4, '0555000053', '02.02.2007', NULL, NULL, NULL, NULL, NULL>
 Izvedite ponovno naredbu koja predstavlja rješenje 2. zadatka. Zbog čega se u rezultatima ne pojavljuje redak za n-torku t1? Popravite rješenje 2. zadatka tako da se u rezultatima pojavljuje i redak za n-torku t1 (i njoj slične). Uništite relaciju **ispit**.
8. Napišite SQL naredbu kojom će se ispisati nazivi organizacijskih jedinica koje su neposredno podređene Fakultetu elektrotehnike i računarstva.
9. Napišite SQL naredbu kojom će se ispisati ime i prezime nastavnika, naziv organizacijske jedinice u kojoj je nastavnik zaposlen te naziv njoj neposredno nadređene organizacijske jedinice. U ispisu se moraju pojaviti svi nastavnici kojima do današnjeg dana nije prestao radni odnos.
10. Zadana je relacijska shema
OSOBA {sifOsoba, ime, prezime, sifOsobaUBraku}
 u kojoj se evidentiraju osobe i njihovi bračni partneri. Vrijednost atributa **sifOsobaUbraku** ne mora uvijek postojati/biti poznata. Domene atributa relacije **osoba** jednake su domenama korespondentnih atributa relacije **nastavnik**.
 Napišite SQL naredbe čijim izvođenjem će u bazi **studAdmin** biti kreirana relacija **osoba** gore zadane relacijske sheme sljedećeg sadržaja:

sifOsoba	ime	prezime	sifOsobaUBraku
238	Ivana	Martinec	249
249	Andrija	Salopek-Rabatić	238
252	Jure	Damiani-Einwalter	263
256	Milica	Balen	
257	Tatjana	Kancir	261
261	Salih	Jalšovec	257
262	Antun	Pajnić	
263	Vlasta	Zanchi	252

Napišite SQL naredbu kojom će se ispisati imena i prezimena osoba i njihovih bračnih partnera. U ispisu se ne trebaju pojaviti osobe koje nisu u braku. Uništite relaciju **osoba**.

11. Napišite SQL naredbu kojom će se ispisati naziv mjesta u kojem stanuje barem dvoje studenata.
12. Napišite SQL naredbu kojom će se ispisati akademska godina i naziv predmeta za koje u toj akademskoj godini postoji točno 10 nastavnih grupa.
13. Napišite SQL naredbu kojom će se za svaku organizacijsku ispisati naziv organizacijske jedinice i broj podređenih organizacijskih jedinica. Uz organizacijske jedinice koje nemaju podređenih ispisati za broj podređenih vrijednost 0.

RJEŠENJA

(uz predavanje 4 - SQL)

Funkcije i operatori

1.

```
SELECT sifra_racuna, ABS(upl_iznos-ispl_iznos) as promet FROM upl_ispl;
```

racun	promet
123456	77
134567	58,26
145678	262,01
156789	187,46
167890	264,88
234567	171,92
245678	64,42

2.

```
SELECT sifra_racuna, upl_iznos FROM upl_ispl  
WHERE TODAY-datum_otvaranja<=540;
```

racun	upl_iznos
156789	205,11
167890	365,11
234567	28,1
245678	110,05

Napomena: n-torke koje će se prikazati kao rezultat izvođenja upita ovisi o datumu na koji se upit izvodi. (Prikazan je rezultat za 4.3.2008.)

3.

```
SELECT sifra_racuna, ROUND(upl_iznos) AS uplaceno_kuna  
      , MOD(upl_iznos*100,100) AS uplaceno_lipa  
FROM upl_ispl;
```

racun	uplaceno_kuna	uplaceno_lipa
123456	120	0
134567	173	47
145678	83	20
156789	205	11
167890	365	11
234567	28	10
245678	110	5

4.

```
SELECT sifra_racuna, MONTH(datum_otvaranja)
, TRIM(upl_iznos || '-' || ispl_iznos)
FROM upl_ispl;
```

(expression)	(expression)	(expression)
123456	2	120.00-43.00
134567	5	173.47-115.21
145678	7	83.20-345.21
156789	9	205.11-17.65
167890	10	365.11-100.23
234567	2	28.10-200.02
245678	3	110.05-45.63

5.

```
SELECT * FROM upl_ispl
WHERE (WEEKDAY(datum_otvaranja)=0 OR WEEKDAY(datum_otvaranja)=6);
```

rbr	racun	datum	upl_iznos	ispl_iznos
5	167890	28.10.2006	365,11	100,23

6.

Ispisat će se dan, mjesec i godina svih računa koji počinju s znamenkom '2'. Korištenjem ključne riječi DISTINCT nema ponavljanja istih redaka.

dan	mjesec	godina
1	2	2007
12	3	2007

7.

Ispisat će se šifra računa te broj dana od prvog u mjesecu (mjesec kada je račun nastao) do nastanka samog računa i to za račune nastale u 2007. godini.

sifra_racuna	dana_od_pocetka_mjeseca
234567	0
245678	11

8.

Ispisat će se podaci za prvih pet studenata rođenih prije 1978. godine i to na sljedeći način: imena studenata velikim slovima a prezimena malim slovima.

(expression)	(expression)
HRVOJE	korkut
HRVOJE	buneta
TONEI	matković
LOBELL	ćubelić

KRISTIJAN	paškoto
-----------	---------

NAPOMENA: Ovo gore je samo primjer **moгуćeg** ispisa. Budući da nije upitom nije definiran željeni poredak zapisa (ORDER BY) nije moguće *a priori* odrediti kojih 5 studenata će se prikazati.

9.

Ispisat će se malim slovima, bez ponavljanja, sva imena studenata koja imaju manje od 4 znaka.

(expression)
ana
ani
eva
iva
ivo
jan
mia
tea
tin

10.

Ispisuje korisnika koji pristupa tablici, te prezimena i godine rođenja studenata koji su rođeni 1977. godine, s tim da se godina provjerava iz JMBG-a, a ispisuje preko atributa *datrod*.

(expression)	prezime	(expression)
badmin	Čubelić	1977
badmin	Paškoto	1977
badmin	Žuža	1977
badmin	Rodić	1977

Where clause, between, in, is null, like, select case

1.

```
SELECT ime, prezime
FROM nastavnik
WHERE datumzaposlenod BETWEEN '15.1.2002' AND '15.7.2002'
AND datumzaposlendo IS NOT NULL;
```

2.

```
SELECT DISTINCT orgjed.nazorgjed
FROM nastavnik, orgjed
WHERE nastavnik.siforgjed = orgjed.siforgjed
AND datumzaposlendo IS NULL
AND ime like '%a';
```

3.

```
SELECT naziv
FROM predmet
WHERE LOWER(naziv) LIKE '%teorij%';
```

Napomena: isto tako mogli smo koristiti i: UPPER(naziv) LIKE '%TEORIJ%'

4.

```
SELECT DISTINCT jmbag
FROM upisanpredmet, predmet
WHERE upisanpredmet.sifpredmet = predmet.sifpredmet
AND upper(naziv) LIKE '%MATEMATIKA%'
AND ocjena IN(2, 4);
```

5.

Napomena: u rješenju je dodan i status2 koji se ne traži u zadatku. status2 je dodan kako bi se pokazalo jedno od alternativnih rješenja.

```
SELECT DISTINCT    jmbag, naziv,
                   CASE
                     WHEN(ocjena = 1)                THEN 'pad'
                     WHEN(ocjena IN(2, 3, 4, 5)) THEN 'prolaz'
                     ELSE 'nepoznato'
                   END as status,
                   CASE
                     WHEN(ocjena = 1)                THEN 'pad'
                     WHEN(ocjena IS NULL) THEN 'nepoznato'
                     ELSE 'prolaz'
                   END as status2
FROM upisanpredmet
JOIN predmet
  ON upisanpredmet.sifpredmet = predmet.sifpredmet
AND upper(naziv) LIKE '%MATEMATIKA%';
```

6.

a)

jmbag	sifpredmet	akgodina	ozngrupa	datumocjena	ocjena
-------	------------	----------	----------	-------------	--------

b)

jmbag	sifpredmet	akgodina	ozngrupa	datumocjena	ocjena
0555001157	26	2005	D-A5	(null)	(null)
0555001157	27	2005	D-A5	(null)	(null)

c)

jmbag	sifpredmet	akgodina	ozngrupa	datumocjena	ocjena
0555001211	2	2003	C-B2	02.02.2004	5

d) neispravan upit – null se ne može koristiti na ovaj način

e) f) g) Treba imati na umu da znak % ima posebno značenje u kombinaciji s LIKE operatorom pa u ovom izrazu nikako **ne predstavlja bilo kakav znakovni niz**. Kao što je upit pod f) analogan upitu pod g) tako je i e) upit istovjetan upitu:

```
SELECT *  
FROM student  
WHERE ime >= 'I%' AND ime <= 'J%';
```

pa će vratiti sve studente kojima je ime po abecedi veće ili jednako 'I%' i manje ili jednako 'J%'. Budući da je ASCII vrijednost znaka '%' 37 sve znamenke i slova su veća od tog znaka (jer imaju veće ASCII vrijednosti), npr. 'I0' je veće od 'I%', 'IA' je veće od 'I%'. Korištenje ovog znaka u ovakvom kontekstu zapravo nema smisla.

Isprobajte ove upite nad relacijom student u bazi studAdmin.

7.

a)

mbr	ime
101	Ivana
102	Ivano

b)

mbr	ime
100	Ivan
101	Ivana
102	Ivano

Primijetite da '%' predstavlja znakovni niz duljine 0 ili više, a _ predstavlja točno jedan znak.

8.

a)

velicina	iznos	tip	tip2
Tlak	1012hPa	Apsolutna vrijednost	Apsolutna vrijednost
Temperatura	25 C	Apsolutna vrijednost	Apsolutna vrijednost
Vlažnost	80%	Postotak	Postotak
Vjetar	2-3 NW	Apsolutna vrijednost	Apsolutna vrijednost
CO2	(null)	Apsolutna vrijednost	Postotak

b) Primijetiti da se ESCAPE znak '8' **ne odnosi** na sadržaj znakovnog niza u tablici ('80%') već samo služi kako bi odredio na koji znak '%' se ESCAPE naredba odnosi. Ipak, bolje je izbjegavati ovakve potencijalne dileme i koristiti neki "specijalni" znak (npr. #, ^, \$, itd.) za koji znamo da ga nema u znakovnim nizovima kao što je to napisano u a).

velicina	iznos	tip	tip2
Tlak	1012hPa	Apsolutna vrijednost	Apsolutna vrijednost
Temperatura	25 C	Apsolutna vrijednost	Apsolutna vrijednost
Vlažnost	80%	Postotak	Postotak
Vjetar	2-3 NW	Apsolutna vrijednost	Apsolutna vrijednost
CO2	(null)	Apsolutna vrijednost	Postotak

c) Primijetiti da crveni znak % označava da zeleni znak % nema specijalno značenje te se traže znakovni nizovi duljine 3 koji završavaju na %. Tako će znakovni niz '80%' zadovoljiti taj uvjet, ali npr. '5%' neće.

LIKE '___%%' ESCAPE '%'

Ipak, bolje je izbjegavati ovakve potencijalne dileme i koristiti neki "specijalni" znak (npr. #, ^, \$, itd.) za koji znamo da ga nema u znakovnim nizovima kao što je to napisano u a).

velicina	iznos	tip	tip2
Tlak	1012hPa	Apsolutna vrijednost	Apsolutna vrijednost
Temperatura	25 C	Apsolutna vrijednost	Apsolutna vrijednost
Vlažnost	80%	Postotak	Postotak
Vjetar	2-3 NW	Apsolutna vrijednost	Apsolutna vrijednost
CO2	(null)	Apsolutna vrijednost	Postotak

9. Upit će vratiti sve organizacijske jedinice za koje vrijedi da njihova nadređena organizacijska jedinica:

a) nema nadređenu organizacijsku jedinicu

b) u imenu sadrži znakovni niz 'vele' bez obzira na velika i mala slova

Isprobajte upit na studAdmin bazi.

10.

nazorgjed	nadorgjed
ZPR	ima nadređenu, to je: FER
ZOEEM	ima nadređenu, to je: FER
ZPM	ima nadređenu, to je: FER

Organizacijske jedinice koje nemaju nadređenu se neće pojaviti u rezultatu zato jer ne ispunjavaju uvjet spajanja:

```
WHERE orgjed.sifnadorgjed = nadorgjed.siforgjed
```

Ukoliko bismo htjeli i njih uključiti u rezultat, trebalo bi koristiti vanjsko spajanje:

```
SELECT orgjed.nazorgjed,  
       CASE  
         WHEN(orgjed.sifnadorgjed IS NULL) THEN 'nema nadređenu'  
         ELSE 'ima nadređenu, to je:' || nadorgjed.nazorgjed  
       END as nadorgjed  
FROM orgjed  
LEFT OUTER JOIN nadorgjed  
  ON orgjed.sifnadorgjed = nadorgjed.siforgjed;
```

Onda bi rezultat bio:

nazorgjed	nadorgjed
FER	nema nadređenu
ZPR	ima nadređenu, to je: FER
ZOEEM	ima nadređenu, to je: FER
ZPM	ima nadređenu, to je: FER
PMF	nema nadređenu

Isprobajte oba upita na studAdmin bazi.

UNION, NULL, OUTER, ORDER BY

1.

```
SELECT * FROM student
ORDER BY datRod DESC
```
2.

```
SELECT naziv, ocjena
FROM upisanPredmet, predmet
WHERE predmet.sifpredmet = upisanPredmet.sifpredmet
ORDER BY naziv, ocjena desc
```
3.

```
SELECT *
FROM student
WHERE student.datRod IS NULL
OR student.pbrRod IS NULL
OR student.pbrStan IS NULL
```
4.

```
INSERT INTO student values ('0036368145', 'Ante',
'Dragan', NULL, 'M', '12.01.1989', NULL, NULL);
```
5.

```
SELECT FIRST 5 ime, prezime
FROM nastavnik
ORDER BY koef DESC
```
6.

```
SELECT imeStudent
FROM student, mjesto
WHERE student.pbrRodStudent = mjesto.pbr
AND mjesto.nazMjesto = 'Dubrovnik'
UNION ALL
SELECT imeNastavnik
FROM nastavnik, mjesto
WHERE nastavnik.pbrStanNastavnik = mjesto.pbr
AND mjesto.nazMjesto = 'Dubrovnik'
```
7.

```
SELECT pbrStan, nazMjesto
FROM student, mjesto
WHERE student.pbrStan = mjesto.pbr
UNION
SELECT pbrStan, nazMjesto
FROM nastavnik, mjesto
WHERE nastavnik.pbrStan = mjesto.pbr
ORDER BY nazMjesto
```

8.

```
SELECT dvorana.ozndvorana, kapacitet, akgodina, ozngrupa
FROM dvorana
LEFT OUTER JOIN predmetgrupa
ON dvorana.ozndvorana = predmetgrupa.ozndvorana
```

9.

```
SELECT orgjed.siforgjed, nazorgjed, sifnastavnik
FROM orgjed
LEFT OUTER JOIN nastavnik
ON orgjed.siforgjed = nastavnik.siforgjed
WHERE orgjed.sifnadorgjed = 36
```

10.

```
SELECT student.jmbag, ocjena
FROM predmet
INNER JOIN upisanPredmet
ON predmet.sifpredmet = upisanPredmet.sifpredmet
AND predmet.naziv = 'Inteligentni sustavi'
RIGHT OUTER JOIN student
ON student.jmbag = upisanpredmet.jmbag
ORDER BY jmbag, ocjena DESC
```

11.

```
SELECT ime, prezime, nazmjesto, nazzupanija
FROM student
LEFT OUTER JOIN mjesto
ON mjesto.pbr = student.pbrstan
LEFT OUTER JOIN zupanija
ON mjesto.sifzupanija = zupanija.sifzupanija
```

12.

```
SELECT ime, prezime, akgodina, naziv
FROM predmet
INNER JOIN upisanpredmet
ON upisanpredmet.sifpredmet = predmet.sifpredmet
RIGHT OUTER JOIN student
ON upisanpredmet.jmbag = student.jmbag
AND (ocjena = 5 OR ocjena = 4)
WHERE MONTH (datRod) = 6
```

13.

```
SELECT student.ime
       , student.prezime, studentImenjak.prezime
FROM student
      LEFT OUTER JOIN studentImenjak
        ON student.ime = studentImenjak.ime
        AND YEAR (student.datrod) =
            YEAR(studentImenjak.datrod)
        AND student.jmbag <> studentImenjak.jmbag
ORDER BY student.ime, student.prezime
```

Paraleno spajanje, reflektivno spajanje, GROUP BY, HAVING

DATABASE studAdmin;

1.

```
SELECT student.JMBAG
       , student.prezime
       , student.ime
  FROM student, mjesto mjestoR, mjesto mjestoS
 WHERE student.pbrRod = mjestoR.pbr
       AND student.pbrStan = mjestoS.pbr
       AND mjestoS.sifZupanija = mjestoR.sifZupanija;
```

2.

```
SELECT student.JMBAG
       , student.datRod
       , student.prezime
       , student.ime
       , studentStariji.JMBAG
       , studentStariji.datRod
       , studentStariji.prezime
       , studentStariji.ime
  FROM student, studentStariji
 WHERE student.JMBAG <> studentStariji.JMBAG
       AND student.datRod > studentStariji.datRod
       AND SUBSTRING (student.ime FROM 1 FOR 1) =
SUBSTRING(studentStariji.ime FROM 1 FOR 1)
       AND SUBSTRING(student.prezime FROM 1 FOR 1) =
SUBSTRING(studentStariji.prezime FROM 1 FOR 1);
```

3.

```
CREATE TABLE ispit (
  sifPredmet          INTEGER      NOT NULL
, JMBAG               CHAR(10)    NOT NULL
, datumRok           DATE         NOT NULL
, sifNastavnikPismeni INTEGER
, ocjenaPismeni       SMALLINT
, sifNastavnikUsmeni  INTEGER
, ocjenaUsmeni        SMALLINT
, datumOcjenaUsmeni   DATE
);
CREATE DISTINCT INDEX idx_ispit ON ispit(sifPredmet, JMBAG, datumRok);

INSERT INTO ispit VALUES (5 , '0555000048', '08.01.2007', 271, 2, 268,
1, '12.01.2007');
INSERT INTO ispit VALUES (5 , '0555000032', '08.01.2007', 267, 2, 271,
1, '12.01.2007');
INSERT INTO ispit VALUES (14, '0555000048', '05.02.2007', 267, 3, 268,
4, '08.02.2007');
INSERT INTO ispit VALUES (5 , '0555000032', '02.02.2007', 267, 4, 271,
5, '05.02.2007');
INSERT INTO ispit VALUES (40, '0555000688', '02.02.2007', 268, 4, 268,
5, '05.02.2007');
```

Kao posljedica izvođenja naredbe:

CREATE DISTINCT INDEX idx_ispit ON ispit(sifPredmet, JMBAG, datumRok);
u relaciju ispit neće biti moguće unijeti 2 n-torke s istom trojkom vrijednosti (sifPredmet, JMBAG, datumRok). Ukoliko se ne obavi ova naredba a nekoliko puta izvede neka od INSERT naredbi koje joj slijede rezultat neće biti u skladu s očekivanjima.

4.

```
SELECT ispit.JMBAG
      , student.prezime AS prezimeStudenta
      , student.ime AS imeStudenta
      , predmet.naziv as nazivPredmeta
      , nastavnikP.ime AS ocijenioPismeniIme
      , nastavnikP.prezime AS ocijenioPismeniPrezime
      , ispit.ocjenaPismeni
      , nastavnikU.ime AS ocijenioUsmeniIme
      , nastavnikU.prezime AS ocijenioUsmeniPrezime
      , ispit.ocjenaUsmeni
FROM ispit, student, predmet, nastavnik nastavnikP, nastavnik
nastavnikU
WHERE ispit.JMBAG = student.JMBAG
      AND ispit.sifPredmet = predmet.sifPredmet
      AND ispit.sifNastavnikPismeni = nastavnikP.sifNastavnik
      AND ispit.sifNastavnikUsmeni = nastavnikU.sifNastavnik;
```

5.

```
SELECT ispit.JMBAG
      , student.prezime AS prezimeStudenta
      , student.ime AS imeStudenta
      , predmet.naziv as nazivPredmeta
      , ispit.ocjenaPismeni
      , ispit.ocjenaUsmeni
FROM ispit, student, predmet, nastavnik nastavnikP, nastavnik
nastavnikU
WHERE ispit.JMBAG = student.JMBAG
      AND ispit.sifPredmet = predmet.sifPredmet
      AND ispit.sifNastavnikPismeni = nastavnikP.sifNastavnik
      AND ispit.sifNastavnikUsmeni = nastavnikU.sifNastavnik
      AND ispit.sifNastavnikPismeni <> ispit.sifNastavnikUsmeni
      AND nastavnikP.pbrStan = nastavnikU.pbrStan;
```

6.

```
SELECT ispit.JMBAG
      , student.prezime AS prezimeStudenta
      , student.ime AS imeStudenta
FROM ispit, student, predmet, mjesto mjestoR, mjesto mjestoS
WHERE ispit.JMBAG = student.JMBAG
      AND ispit.sifPredmet = predmet.sifPredmet
      AND student.pbrRod = mjestoR.pbr
      AND student.pbrStan = mjestoS.pbr
      AND student.pbrRod <> student.pbrStan
      AND mjestoS.sifZupanija = mjestoR.sifZupanija
      AND predmet.naziv = 'Baze podataka';
```

7.

```
INSERT INTO ispit VALUES (4, '0555000053', '02.02.2007', NULL, NULL,  
NULL, NULL, NULL);
```

Za t1 n-torku se ne pojavljuje redak u rezultatu jer se uvjeti spajanja

```
ispit.sifNastavnikPismeni = nastavnikP.sifNastavnik
```

i

```
ispit.sifNastavnikUsmeni = nastavnikU.sifNastavnik
```

evaluiraju kao UNKNOWN. Vrijednosti atributa ispit.sifNastavnikPismeni i ispit.sifNastavnikUsmeni su nepoznate - NULL pa se radi o usporedbi tipa

```
NULL = nastavnik.sifNastavnik.
```

Da bi n-torka bila kandidat za izlazni rezultat ovi uvjeti spajanja se moraju evaluirati kao TRUE.

Da bi se svaka n-torka iz relacije ispit pojavila u listi izlaznih rezultata prirodno spajanje relacije ispit s relacijom nastavnik treba zamijeniti vanjskim spajanjem.

```
SELECT ispit.JMBAG  
  , student.prezime AS prezimeStudenta  
  , student.ime AS imeStudenta  
  , predmet.naziv as nazivPredmeta  
  , nastavnikP.ime AS ocijenioPismeniIme  
  , nastavnikP.prezime AS ocijenioPismeniPrezime  
  , ispit.ocjenaPismeni  
  , nastavnikU.ime AS ocijenioUsmeniIme  
  , nastavnikU.prezime AS ocijenioUsmeniPrezime  
  , ispit.ocjenaUsmeni  
FROM ispit  
  INNER JOIN student  
    ON ispit.JMBAG = student.JMBAG  
  INNER JOIN predmet  
    ON ispit.sifPredmet = predmet.sifPredmet  
  LEFT OUTER JOIN nastavnik nastavnikP  
    ON ispit.sifNastavnikPismeni = nastavnikP.sifNastavnik  
  LEFT OUTER JOIN nastavnik nastavnikU  
    ON ispit.sifNastavnikUsmeni = nastavnikU.sifNastavnik;
```

Budući da se rezervirane riječi INNER i OUTER mogu izostaviti gornja SELECT naredba ekvivalentan je naredbi:

```
SELECT ispit.JMBAG  
  , student.prezime AS prezimeStudenta  
  , student.ime AS imeStudenta  
  , predmet.naziv as nazivPredmeta  
  , nastavnikP.ime AS ocijenioPismeniIme  
  , nastavnikP.prezime AS ocijenioPismeniPrezime  
  , ispit.ocjenaPismeni  
  , nastavnikU.ime AS ocijenioUsmeniIme  
  , nastavnikU.prezime AS ocijenioUsmeniPrezime  
  , ispit.ocjenaUsmeni  
FROM ispit  
  JOIN student  
    ON ispit.JMBAG = student.JMBAG  
  JOIN predmet
```



```

        ON ispit.sifPredmet = predmet.sifPredmet
LEFT JOIN nastavnik nastavnikP
        ON ispit.sifNastavnikPismeni = nastavnikP.sifNastavnik
LEFT JOIN nastavnik nastavnikU
        ON ispit.sifNastavnikUsmeni = nastavnikU.sifNastavnik;
DROP TABLE ispit;

```

8.

```

SELECT orgJedPod.nazOrgJed
FROM orgjed orgjedPod, orgjed
WHERE orgJedPod.sifNadOrgJed = orgJed.sifOrgJed
AND orgjed.nazOrgJed = 'Fakultet elektrotehnike i računarstva';

```

9.

```

SELECT nastavnik.ime, nastavnik.prezime, orgjed.nazOrgJed,
orgJedNad.nazOrgJed
FROM nastavnik
INNER JOIN orgjed
LEFT OUTER JOIN orgjed orgjedNad
ON orgjed.sifNadOrgJed = orgjedNad.sifOrgJed
ON nastavnik.sifOrgJed = orgjed.sifOrgJed
WHERE nastavnik.datumZaposlenDo IS NULL;

```

10.

```

CREATE TABLE osoba (
    sifOsoba          INTEGER          NOT NULL
, ime                NCHAR(25)       NOT NULL
, prezime            NCHAR(25)       NOT NULL
, sifOsobaUBraku     INTEGER
);
CREATE DISTINCT INDEX idx_osoba ON osoba (sifOsoba);

INSERT INTO osoba VALUES (238, 'Ivana' , 'Martinec' , 249);
INSERT INTO osoba VALUES (249, 'Andrija', 'Salopek-Rabatić' , 238);
INSERT INTO osoba VALUES (252, 'Jure' , 'Damiani-Einwalter', 263);
INSERT INTO osoba VALUES (256, 'Milica', 'Balen' , NULL);
INSERT INTO osoba VALUES (257, 'Tatjana', 'Kancir' , 261);
INSERT INTO osoba VALUES (261, 'Salih' , 'Jalšovec' , 257);
INSERT INTO osoba VALUES (262, 'Antun' , 'Pajnić' , NULL);
INSERT INTO osoba VALUES (263, 'Vlasta' , 'Zanchi' , 252);

```

Kao posljedica izvođenja naredbe:

```

CREATE DISTINCT INDEX idx_osoba ON osoba (sifOsoba);

```

u relaciju **osoba** neće biti moguće unijeti 2 n-torke s istom vrijednosti atributa sifOsoba. Ukoliko se ne obavi ova naredba a nekoliko puta izvede neka od INSERT naredbi koje joj slijede rezultat neće biti u skladu s očekivanjima.

```

SELECT osoba.ime, osoba.prezime, osobaUBraku.ime, osobaUBraku.prezime
FROM osoba, osoba osobaUBraku
WHERE osoba.sifOsobaUBraku = osobaUBraku.sifOsoba;

DROP TABLE osoba;

```

11.

```
SELECT mjesto.nazMjesto
FROM student, mjesto
WHERE student.pbrStan = mjesto.pbr
GROUP BY student.pbrStan, mjesto.nazMjesto
HAVING COUNT(student.jmbag) > 1;
```

12.

```
SELECT akGodina, predmet.naziv
FROM predmetGrupa, predmet
WHERE predmetGrupa.sifPredmet = predmet.sifPredmet
GROUP BY akGodina, predmet.sifpredmet, predmet.naziv
HAVING COUNT (oznGrupa) = 10;
```

13.

```
SELECT orgjed.nazOrgJed, COUNT(orgJedPod.sifOrgJed)
FROM orgjed
LEFT OUTER JOIN orgjed orgjedPod
ON orgjed.sifOrgJed = orgjedPod.sifNadOrgJed
GROUP BY orgjed.sifOrgJed, orgjed.nazOrgJed;
```

Pokušajte izvesti gornji upit tako da ne grupirate po šifri organizacijske jedinice nego samo po nazivu:

```
SELECT orgjed.nazOrgJed, COUNT(orgJedPod.sifOrgJed)
FROM orgjed
LEFT OUTER JOIN orgjed orgjedPod
ON orgjed.sifOrgJed = orgjedPod.sifNadOrgJed
GROUP BY orgjed.nazOrgJed;
```

Uočite da je broj n-torki koje ispisuje gornji upit veći od broja n-torki koje ispisuje donji upit. Naime, postoji nekoliko organizacijskih jedinica jednakog naziva npr. 'Ekonomski fakultet' u Zagrebu, Splitu, Rijeci i Osijeku i ukoliko se obavi grupiranje isključivo po nazivu ove 4 organizacijske jedinice će 'ući' u istu grupu. To naravno nije ono što se tražilo zadatkom.