

1. Za svakog nastavnika koji stanuje u mjestu s poštanskim brojem 20000 ispisati šifru, ime i prezime te oznake svih grupa kojima je taj nastavnik predavao akademske godine 2012/2013. Ispisati podatke i o nastavnicima koji akademske godine 2012/2013 nisu predavali niti jednoj grupi (ako stanuju u mjestu s poštanskim brojem 20000). Takvim nastavnicima kao oznaku grupe ispisati NULL.

Voditi računa da nastavnik istoj grupi može predavati više različitih predmeta, a u rezultatu se ne smiju pojavljivati iste n-torke.

Primjer rezultata

sifnastavnik	imenastavnik	prezimenastavnik	ozngrupa
56	Petar	Raduka	A-C
56	Petar	Raduka	A-G
56	Petar	Raduka	A-F
59	Davor	Šavor	(null)
87	Maden	Rajčić	A-E
...

```
SELECT DISTINCT nastavnik.sifNastavnik, imeNastavnik, prezimeNastavnik, oznGrupa
FROM nastavnik LEFT OUTER JOIN predmetGrupa ON
nastavnik.sifNastavnik=predmetGrupa.sifNastavnik AND akGodina = 2012
WHERE pBrStanNastavnik = 20000;
```

2. Svim grupama kojima je kao predavač evidentiran nastavnik Andrica Bajšanski postaviti umjesto njega nastavnika Darka Takšića.

Nakon toga iz relacije Nastavnik obrisati nastavnika Andricu Bajšanskog.

Općenita napomena: višestruke SQL naredbe odvajati znakom ';'.

UPDATE predmetGrupa

```
SET sifNastavnik = (SELECT DISTINCT sifNastavnik FROM nastavnik WHERE imeNastavnik = 'Darko' AND
prezimeNastavnik = 'Takšić')
```

```
WHERE sifNastavnik = (SELECT DISTINCT sifNastavnik FROM nastavnik WHERE imeNastavnik = 'Andrica'
AND prezimeNastavnik = 'Bajšanski');
```

DELETE FROM nastavnik

```
WHERE sifNastavnik = (SELECT DISTINCT sifNastavnik FROM nastavnik WHERE imeNastavnik = 'Andrica'
AND prezimeNastavnik = 'Bajšanski');
```

3. Za svakog nastavnika koji stanuje u mjestu s poštanskim brojem 34000 ispisati šifru, ime i prezime te oznake svih grupa kojima je taj nastavnik predavao akademske godine 2011/2012. Ispisati podatke i o nastavnicima koji akademske godine 2011/2012 nisu predavali niti jednoj grupi (ako stanuju u mjestu s poštanskim brojem 34000). Takvim nastavnicima kao oznaku grupe ispisati NULL.

Voditi računa da nastavnik istoj grupi može predavati više različitih predmeta, a u rezultatu se ne smiju pojavljivati iste n-torke.

Primjer rezultata

sifnastavnik	imenastavnik	prezimenastavnik	ozngrupa
56	Petar	Raduka	A-C
56	Petar	Raduka	A-G
56	Petar	Raduka	A-F
59	Davor	Šavor	(null)
87	Maden	Rajčić	A-E
...

```
SELECT DISTINCT nastavnik.sifNastavnik, imeNastavnik, prezimeNastavnik, oznGrupa
FROM nastavnik LEFT OUTER JOIN predmetGrupa ON
nastavnik.sifNastavnik=predmetGrupa.sifNastavnik AND akGodina = 2011
WHERE pBrStanNastavnik = 34000;
```

4. Ispisati naziv predmeta i oznaku grupa kojima je taj predmet bio predavan akademske godine 2012/2013. Ako je nekoj grupi taj predmet predavao nastavnik koji više nije u radnom odnosu, dodatno ispisati njegovo ime i prezime. U suprotnom, za ime i prezime nastavnika ispisati NULL vrijednosti. Smatra se da je nastavnik u radnom odnosu ukoliko mu nije evidentiran datum prestanka zaposlenja. Uzeti u obzir samo grupe kojima oznaka počinje slovom "A".

Primjer rezultata

naziv	oznGrupa	ime	prezime
Algoritmi i strukture podataka	A-F	NULL	NULL
Algoritmi i strukture podataka	A-G	Ibrahim	Palić
Baze podataka	A-A	NULL	NULL
Automati, formalni jezici i jezični procesori II	A-A	Marina	Pavlaković
Automati, formalni jezici i jezični procesori II	A-B	Aleksandra	Kavalari


```
SELECT nazPredmet, oznGrupa,
CASE
    WHEN datumZaposlenDo IS NOT NULL THEN nastavnik.imeNastavnik
    ELSE NULL
END AS ime,
CASE
    WHEN datumZaposlenDo IS NOT NULL THEN nastavnik.prezimeNastavnik
    ELSE NULL
END AS prezime
FROM predmetGrupa JOIN predmet on predmetGrupa.sifPredmet = predmet.sifPredmet
LEFT JOIN nastavnik ON predmetGrupa.sifNastavnik = nastavnik.sifNastavnik
WHERE akGodina = 2012 AND oznGrupa LIKE 'A%'
```

5. Potrebno je dodati novu organizacijsku jedinicu "Republika Hrvatska" sa šifrom 1 (nema nadređenu organizacijsku jedinicu). Zatim postaviti Republiku Hrvatsku kao nadređenu svim organizacijskim jedinicama koje dosad nisu imale nadređenu organizacijsku jedinicu (osim same Republike Hrvatske).

```
INSERT INTO orgJed VALUES(1, 'Republika Hrvatska', NULL);
```

```
UPDATE orgJed SET sifNadOrgJed=1
```

```
WHERE sifNadOrgJed IS NULL AND sifOrgJed <> 1;
```

6. Za sva mjesta čiji naziv počinje slovom "K" ispišite poštanski broj i naziv te matične brojeve (jmbag) studenata koji su u tom mjestu rođeni, ali u njemu trenutno ne stanuju. U ispisu se moraju pojaviti i podaci o mjestima u kojima nema takvih studenata (u tom slučaju vrijednost matičnog broja je NULL).

Primjer rezultata:

pbr	nazMjesto	rodjeniANeStanuju
44320	Kutina	0555001234
44320	Kutina	0555002345
48000	Koprivnica	0555003456
49000	Krapina	NULL


```
SELECT DISTINCT pBr, nazMjesto, JMBAG AS rodjeniANeStanuju
```

```
FROM mjesto LEFT JOIN student ON pBrRodStudent = mjesto.pBr AND pBrRodStudent <>
```

```
pBrStanStudent
```

```
WHERE nazMjesto LIKE 'K%';
```

7. Ispisati imena, prezimena i naziv mjesta stanovanja svih studenata koji **ne stanuju** u mjestu s nazivom "Zadar".

```
SELECT imeStudent, prezimeStudent, nazMjesto
```

```
FROM student JOIN mjesto on pBrStanStudent = pBr
```

```
WHERE nazMjesto <> 'Zadar'
```

8. Napisati upit koji će ispisati imena, prezimena i koeficijente svih nastavnika koji imaju koeficijent u intervalu [3, 5] i rade u organizacijskoj jedinici koja je **direktno podređena krovnoj** (nema nadređenu) organizacijskoj jedinici.

Obavezno koristiti BETWEEN.

Npr. ako imamo sljedeću hijerarhiju org.jedinica:

OrgJed1

```
| -OrgJed11
|         | -OrgJed111
|         | -OrgJed112
| -OrgJed12
|         | -OrgJed121
```

OrgJed2

```
| -OrgJed21
|         | -OrgJed211
|         | -OrgJed212
```

Za nju vrijedi da su OrgJed1 i OrgJed2 **krovne** (jer nemaju nadređenu) organizacijske jedinice, a OrgJed11, OrgJed12 i OrgJed21 su **direktno podređene krovnim** organizacijskim jedinicama (dok ostale nisu). Dakle, traže se djelatnici upravo iz takvih organizacijski jedinica kao što su OrgJed11, OrgJed12 i OrgJed21.

SELECT imeNastavnik, prezimeNastavnik, koef

FROM nastavnik JOIN orgJed

ON nastavnik.sifOrgJed = orgJed.sifOrgJed

WHERE koef BETWEEN 3 AND 5 AND sifNadOrgJed IN (SELECT DISTINCT sifOrgJed FROM orgJed WHERE sifNadOrgJed IS NULL);

9. Ispisati jmbag studenta, ime studenta, prezime studenta i nazive svih predmeta iz kojih je student dobio ocjenu 5. Ukoliko student iz niti jednog predmeta nije dobio ocjenu 5 za naziv predmeta ispisati NULL vrijednost. Uzeti u obzir samo one studente kojima prezime počinje slovom R.

Pretpostavlja se da za svaki upisani predmet postoji odgovarajući zapis u relaciji predmet.

Primjer rezultata

jmbag	ime	prezime	naziv
0555000011	Krešimir	Salopek	Telekomunikacijske mreže
0555000027	Andrej	Jurčić	NULL
0555000737	Marko	Horvat	Fizika 1
0555000737	Marko	Horvat	Baze podataka
0555000737	Marko	Horvat	Arhitektura računala 1

SELECT student.JMBAG, imeStudent, prezimeStudent, nazPredmet AS naziv

FROM student LEFT JOIN upisanPredmet ON student.JMBAG = upisanPredmet.JMBAG AND ocjena = 5

LEFT JOIN predmet ON upisanPredmet.sifPredmet = predmet.sifPredmet

WHERE prezimeStudent LIKE 'R%';

10. Za svakog studenta rođenog u siječnju ispišite jmbag studenta, akademsku godinu i prosječnu ocjenu svih predmeta koje je student upisao te akademske godine (nazovite stupac *prosjecna_ocjena*). Uzeti u obzir samo akademske godine nakon 2009/2010. Negativne ocjene se ne računaju u prosjek. Ispišite samo zapise za koje je prosječna ocjena veća od ili jednaka 3. Napomena: prosječnu ocjenu ne zaokruživati.

```
SELECT student.JMBAG, akGodina, AVG(ocjena) AS prosjecna_ocjena
FROM student JOIN upisanPredmet ON student.JMBAG = upisanPredmet.JMBAG
WHERE akGodina>2009
AND ocjena>1
AND MONTH(datumRod)=1
GROUP BY student.JMBAG, akGodina
HAVING AVG(ocjena)>=3;
```

11. Za organizacijsku jedinicu ispišite naziv te prosječan koeficijent nastavnika (nazovite stupac *prosjecni_koef*) koji su u toj organizacijskoj jedinici ikad bili zaposleni. Ispišite zapise samo za one organizacijske jedinice za koje je broj nastavnika koji su ikad bili zaposleni u toj organizacijskoj jedinici veći od 4.

```
SELECT nazOrgJed, AVG(koef) AS prosjecni_koef
FROM orgJed, nastavnik
WHERE orgJed.sifOrgJed = nastavnik.sifOrgJed
GROUP BY nazOrgJed
HAVING COUNT(sifNastavnik)>4;
```

12. Ispisati jmbag, ime i prezime svih studenata rođenih 1980. i 1981. godine. Dodatno, za svakog takvog studenta ispisati nazive predmeta koje je upisao u akademskoj godini 2010/2011. Ukoliko student nema upisanih predmeta u akademskoj godini 2010/2011, za naziv predmeta ispisati NULL vrijednost. Pretpostavlja se da za svaki upisani predmet postoji odgovarajući zapis u relaciji predmet. Primjer rezultata

jmbag	ime	prezime	naziv
0555000011	Krešimir	Salopek	Telekomunikacijske mreže
0555000027	Andrej	Jurčić	NULL
0555000737	Marko	Horvat	Fizika 1
0555000737	Marko	Horvat	Baze podataka
0555000737	Marko	Horvat	Arhitektura računala 1


```
SELECT student.JMBAG, imeStudent AS ime, prezimeStudent AS prezime, nazPredmet AS naziv
FROM student LEFT JOIN upisanPredmet ON student.JMBAG = upisanPredmet.JMBAG AND akGodina =
2010 LEFT JOIN predmet ON upisanPredmet.sifPredmet = predmet.sifPredmet
WHERE YEAR(datumRod) IN(1980, 1981);
```

13. Ispisati jmbag i ime svakog studenta za kojeg vrijedi da mu se ime sastoji od **parnog** broja znakova i da je **prvo** slovo imena **jednako zadnjem** slovu imena (naravno, bez obzira na velika i mala slova).

Na primjer: Neven, Ana, itd.

```
SELECT JMBAG, imeStudent
FROM student
WHERE MOD(CHAR_LENGTH(TRIM(imeStudent)),2)=0
AND SUBSTRING(LOWER(imeStudent) FROM 1 FOR 1) = SUBSTRING(imeStudent FROM
CHAR_LENGTH(TRIM(imeStudent)) FOR 1);
```

14. Ispisati JMBAG, prezime i inicijal imena studenata koji **ne stanuju** u istoj županiji u kojoj su i rođeni, te su akademske godine koja je za jedan manja od tekuće kalendarske godine iz bilo kojeg predmeta pohađali nastavu u grupi čija oznaka počinje slovom "B".

Pored JMBAG-a, prezimena i inicijala imena studenta ispisati i oznaku grupe u kojoj je student slušao predavanja.

Upit mora biti vremenski neovisan o trenutku izvođenja.

Stupce rezultata imenovati u skladu sa sljedećim predloškom:

JMBAG	prezimellme	grupa
0555000508	Matas.Z	B-C
0555001659	Pavić.D	B-E
0555002287	Čabrijan.M	B-K

U rezultatu se ne smiju pojavljivati jednake n-torke (duplikati). Poredak zapisa u rezultatu nije bitan.

```
SELECT DISTINCT student.JMBAG, TRIM(prezimeStudent) || '.' || SUBSTRING(imeStudent FROM 1 FOR 1)
AS prezimellme, oznGrupa AS grupa
FROM student JOIN upisanPredmet ON student.JMBAG = upisanPredmet.JMBAG
JOIN mjesto AS mjestoR ON pBrRodStudent = mjestoR.pBr
JOIN mjesto AS mjestoS ON pBrStanStudent = mjestoS.pBr
JOIN zupanija AS zupanijaR ON mjestoR.sifZupanija = zupanijaR.sifZupanija
JOIN zupanija AS zupanijaS ON mjestoS.sifZupanija = zupanijaS.sifZupanija
WHERE oznGrupa LIKE 'B%'
AND akGodina = YEAR(TODAY)-1
AND zupanijaS.sifZupanija <> zupanijaR.sifZupanija;
```

15. Ispisati prezimena i imena nastavnika zaposlenih u organizacijskim jedinicima s nazivom kraćim od naziva neposredno nadređene organizacijske jedinice. Pored prezimena i imena nastavnika ispisati i naziv organizacijske jedinice u kojoj je zaposlen.

Iz rezultata izbaciti n-torke u kojima se pojavljuju nastavnici kojima je prestao radni odnos.

Zapise poredati abecedno po nazivu organizacijske jedinice, te po prezimenu i imenu nastavnika.

```
SELECT prezimeNastavnik, imeNastavnik, org1.nazOrgJed
FROM nastavnik, orgJed AS org1, orgJed AS org2
WHERE nastavnik.sifOrgJed = org1.sifOrgJed
AND org1.sifNadOrgJed = org2.sifOrgJed
AND CHAR_LENGTH(TRIM(org1.nazOrgJed)) < CHAR_LENGTH(TRIM(org2.nazOrgJed))
AND datumZaposlenDo IS NULL
ORDER BY org1.nazOrgJed ASC, prezimeNastavnik ASC, imeNastavnik ASC;
```

16. Za svaki predmet čiji je ukupan broj sati nastave tjedno **manji** od prosječnog tjednog broja sati nastave svih predmeta ispisati naziv, broj ECTS bodova te koliko još **pored njega** ima predmeta sa istim brojem ECTS bodova.

Zapise u rezultatu poredati abecedno po nazivu predmeta

```
SELECT nazPredmet, ECTSBod, (SELECT COUNT(sifPredmet) FROM predmet AS pred WHERE
predmet.ECTSBod = pred.ECTSBod AND predmet.nazPredmet <> pred.nazPredmet) AS istoECTSBodova
FROM predmet
WHERE ukBrSatiTjedno < (SELECT AVG(ukBrSatiTjedno) FROM predmet)
ORDER BY nazPredmet ASC;
```

17. Svim nastavnicima čiji je koeficijent manji od prosječnog povećati koeficijent za 10%.

UPDATE nastavnik

SET koef = koef + koef/10

WHERE koef < (SELECT AVG(koef) FROM nastavnik);

18. Za relaciju **predmetgrupa** kreirati najmanji mogući broj indeksa koji će omogućiti efikasno obavljanje svih navedenih upita:

SELECT * FROM predmetgrupa WHERE sifpredmet = 1 AND akgodina = 2010;

SELECT * FROM predmetgrupa WHERE akgodina > 2010 AND akgodina < 2014 ORDER BY akgodina;

SELECT * FROM predmetgrupa WHERE sifnastavnik = 690 AND ozngrupa = 'D-A1';

SELECT * FROM predmetgrupa WHERE ozngrupa = 'D-A1' AND akgodina = 2010 AND sifpredmet > 20;

SELECT * FROM predmetgrupa ORDER BY akgodina DESC, sifpredmet;

SELECT * FROM predmetgrupa ORDER BY sifnastavnik;

SELECT * FROM predmetgrupa ORDER BY akgodina DESC, sifpredmet ASC, ozngrupa ASC;

CREATE INDEX in1 ON predmetGrupa (akGodina DESC, sifPredmet, oznGrupa);

CREATE INDEX in2 ON predmetGrupa (sifNastavnik, oznGrupa);

19. Osigurati entitetski integritet i integritet ključa u relaciji **mjesto**.

Napomena: isti naziv mjesta se može pojaviti više puta.

Osigurati da se u relaciji **nastavnik** može evidentirati samo mjesto stanovanja koje (ukoliko nije NULL) postoji u relaciji **mjesto**.

Napomena: višestruke SQL naredbe odvojiti znakom ";". Koristiti sintaksu za naknadnu izmjenu postojeće tablice: 'ALTER TABLE ime_tablice ADD CONSTRAINT opis_ogranicenja CONSTRAINT ime_ogranicenja'.

ALTER TABLE mesto ADD CONSTRAINT PRIMARY KEY(pBr);

ALTER TABLE nastavnik ADD CONSTRAINT FOREIGN KEY(pbrStanNastavnik) REFERENCES mesto(pBr);