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'.



Odgovor ALTER TABLE mjesto ADD CONSTRAINT PRIMARY KEY (pbr) CONSTRAINT mjestoPK; ALTER TABLE nastavnik ADD CONSTRAINT FOREIGN KEY (pbrStanNastavnik) REFERENCES mjesto(pbr) CONSTRAINT nastavnikFKpbrStan;

Za relaciju student kreirati najmanji mogući broj indeksa koji će omogućiti efikasno obavljanje svih navedenih upita:

SELECT * FROM student WHERE datumrod > '01.01.1980' ORDER BY datumrod;

SELECT * FROM student WHERE prezimeStudent = 'Salopek' AND imeStudent = 'Andrej';

SELECT * FROM student WHERE prezimeStudent = 'Salopek';

SELECT * FROM student ORDER BY datumrod DESC;

SELECT * FROM student WHERE jmbg = '0101980554687';

SELECT * FROM student WHERE datumrod > '01.01.1980';

SELECT * FROM student ORDER BY datumrod, jmbag;



```
Odgovor

CREATE INDEX idx1 ON student(datumrod, jmbag);
CREATE INDEX idx2 ON student(prezimestudent, imestudent);
CREATE INDEX idx3 ON student(jmbg);
```

Napraviti 2 virtualne relacije:

1. Napraviti virtualnu relaciju **stanBr** sa shemom relacije **STANBR** = **pbrstan, broj** koja će omogućiti pregled broja studenata koji su položili barem jedan predmet u **2013/2014** akademskoj godini po **poštanskom broju stanovanja.**

Primjer rezultata:

SELECT * FROM stanBr:

pbrstan	broj
99999	1
(null)	5
11111	3

Primijetite mogućnost postojanja n-torke koja predstavlja studente čije mjesto stanovanja nije poznato.

2. Napraviti virtualnu relaciju **mjestoBr** sa shemom relacije **MJESTOBR** = **nazmjesto, broj** koja će po **SVIM mjestima** omogućiti pregled broja studenata koji su položili predmet u **2013/2014** akademskoj godini koji **stanuju** u tom mjestu. Uključiti i studente čije mjesto stanovanja nije poznato. **Virtualnu relaciju mjestoBr** <u>obavezno</u> napraviti pomoću relacije stanBr.

Primjer rezultata:

SELECT * FROM mjestoBr;

nazmjesto	broj
Zadar	325
(null)	267
Zagreb	355
Hum	(null)
Makarska	(null)

Primijetite mogućnost postojanja n-torke koja predstavlja studente čije mjesto stanovanja nije poznato i mogućnost postojanja n-torki koje predstavljaju mjesta za koja nema položenih predmeta u zadanoj akademskoj godini.

Odgovor CREATE VIEW stanbr (pbrstan, broj) AS SELECT pbrstanstudent, COUNT(DISTINCT student.JMBAG) FROM student JOIN upisanpredmet ON student.jmbag=upisanpredmet.jmbag WHERE upisanpredmet.akgodina=2013 AND upisanpredmet.ocjena>1 GROUP BY pbrstanstudent; CREATE VIEW mjestobr (nazmjesto, broj) AS SELECT nazmjesto, stanbr.broj FROM stanbr FULL OUTER JOIN mjesto ON stanbr.pbrstan=mjesto.pbr;

Svim predmetima čiji je broj sati tjedno veći od prosjeka povećati broj ECTS bodova za 20%. Nove ECTS bodove zaokružiti na cijeli broj.

Npr.

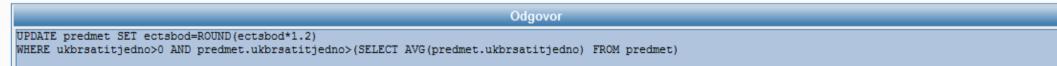
Prije:

sifPredmet	nazPredmet	ectsBod	ukBrSatiTjedno
1	Matematika	6.0	4.0
2	Fizika	6.0	4.0
3	Informatika	6.0	5.0

Poslije:

sifPredmet	nazPredmet	ectsBod	ukBrSatiTjedno
1	Matematika	6.0	4.0
2	Fizika	6.0	4.0
3	Informatika	7.0	5.0





< **1** >

Mogući broj bodova: 0,25 Ostvareni broj bodova: 0,25

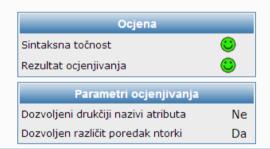


Primarni ključ u relaciji **predmetGrupa** je skup atributa: {sifPredmet, akGodina, oznGrupa}.**Jednom** SQL naredbom osigurajte entitetski integritet i integritet ključa u toj relaciji.

Osigurati entitetski integritet i integritet ključa u relaciji nastavnik.

Osigurati referencijski integritet među relacijama predmetGrupa i nastavnik.

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



Odgovor

alter table predmetgrupa add constraint primary key(sifpredmet, akgodina, ozngrupa) constraint prvo;

alter table nastavnik add constraint primary key(sifnastavnik) constraint drugo;

alter table predmetgrupa add constraint foreign key(sifnastavnik) references nastavnik(sifnastavnik) constraint trece;

Napraviti 2 virtualne relacije:

1. Napraviti virtualnu relaciju **polozeno** sa shemom relacije **POLOZENO** = **sifraPredmet, brojUspjesnihPolaganja** koja će omogućiti pregled šifri **svih** predmeta i broja uspješnih polaganja u školskoj godini **2011/2012**.

Primjer rezultata:

SELECT * FROM polozeno;

sifraPredmet	brojUspjesnihPolaganja
2	13
29	29
62	0

Primijetite moqućnost postojanja n-torke koja predstavlja predmete koje nijedan student nije uspješno položio navedene godine.

2. Napraviti virtualnu relaciju **polozenoNaziv** sa shemom relacije **POLOZENONAZIV**= **nazivPredmet, brojUspjesnihPolaganja** koja će omogućiti pregled naziva **svih** predmeta i broja uspješnih polaganja u školskoj godini **2011/2012**.

Virtualnu relaciju polozenoNaziv <u>obavezno</u> napraviti pomoću relacije polozeno.

Primjer rezultata:

SELECT * FROM polozenoNaziv;

nazivPredmet	brojUspjesnihPolaganja
Matematika 2	0
Logička Algebra	21
Fizika 1	0
Operacijski sustavi	16
Baze podataka	0



Odgovor

CREATE VIEW polozeno (sifraPredmet, brojUspjesnihPolaganja) AS

SELECT predmet.sifPredmet, (SELECT COUNT(jmbag)

FROM upisanPredmet

WHERE akGodina = 2011 AND ocjena > 1

AND upisanPredmet.sifPredmet = predmet.sifPredmet)

FROM predmet;

CREATE VIEW polozenoNaziv (nazivPredmet, brojUspjesnihPolaganja) AS

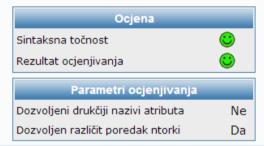
SELECT predmet.nazPredmet, polozeno.brojUspjesnihPolaganja

FROM predmet JOIN polozeno

ON predmet.sifPredmet = polozeno.sifraPredmet;

Svim grupama iz akademske godine 2010/2011 čiji kapacitet odgovara minimalnom kapacitetu svih grupa na oznaku dodati sufiks "-mala" (npr. grupa "D-H1" će postati "D-H1-mala").

(NAPOMENA: Vodite računa o mogućem postojanju praznih mjesta u originalnim nazivima grupa!)





Mogući broj bodova: 0,25 Ostvareni broj bodova: 0,25



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

SELECT * FROM **student** WHERE datumRod > '01.01.1980' ORDER BY datumRod;

SELECT * FROM **student** ORDER BY prezimeStudent, imeStudent, jmbg;

SELECT * FROM **student** WHERE prezimeStudent = 'Salopek';

SELECT * FROM student ORDER BY datumRod DESC;

SELECT * FROM student WHERE datumRod > '01.01.1980';

SELECT * FROM **student** ORDER BY datumRod, jmbag;

