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

Napomena: višestruke SQL naredbe odvojiti znakom ";". Koristiti sintaksu za naknadnu izmjenu postojeće tablice: 'ALTER TABLE ime\_tablice ADD CONSTRAINT ime\_ogranicenja opis\_ogranicenja'.

Check column mode: 4. PERMISSIVE: try 3 (to match by names); if not - try 2 (use column order).

```
ALTER TABLE upisanPredmet
ADD CONSTRAINT theKey PRIMARY KEY(jmbag, sifPredmet, akGodina, oznGrupa)
, ADD CONSTRAINT referenca FOREIGN KEY(sifPredmet,akGodina, oznGrupa) REFERENCES predmetGrupa(sifPredmet, akGodina, oznGrupa)

ALTER TABLE predmetGrupa
ALTER TABLE predmetGrupa
ADD CONSTRAINT keyGrupe PRIMARY KEY(sifPredmet, akGodina, oznGrupa);

Save
```

Result (count = 0, db = studAdmin04, @ Correct! Well done!):

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;
```

Check column mode: 4. PERMISSIVE: try 3 (to match by names); if not - try 2 (use column order).

```
CREATE INDEX i1 ON predmetgrupa(akgodina DESC, sifpredmet, ozngrupa);
CREATE INDEX i2 ON predmetgrupa(sifnastavnik, ozngrupa);

Save
```

Result (count = 0, db = studAdmin05, @ Correct! Well done!)

1. Napraviti virtualnu relaciju **stanBr** sa shemom relacije **STANBR = pbrstan, broj** koja će omogućiti pregled broja studenata koji su položili barem jedan predmet u **2012/2013** 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.

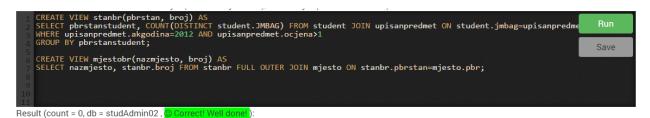
2. Napraviti virtualnu relaciju **mjestoBr** sa shemom relacije **MJESTOBR = nazmjesto, broj** koja će po **SVIM mjestima** omogućiti pregled broja studenata koji su položili predmet u **2012/2013** akademskoj godini koji **stanuju** u tom mjestu. Uključiti i studente čije mjesto stanovanja nije poznato. **Virtualnu relaciju mjestoBr obavezno 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.



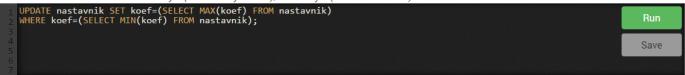
Svim nastavnicima čiji je koeficijent jednak minimalnom koeficijentu postaviti koeficijent na maksimalni koeficijent.

Npr.

Prije:		Poslije:	
sifNastavnik	koef	sifNastavnik	koef
1	3.0	1	8.5
2	7.9	2	7.9
3	8.5	3	8.5

Napomena: radi preglednosti iz primjera su izbačeni ostali atributi relacije nastavnik.

Check column mode: 4. PERMISSIVE: try 3 (to match by names); if not - try 2 (use column order).



Result (count = 0, db = studAdmin04, © Correct! Well done!):