



# Zadatci za vježbu

## #8

Previous step

Submit answer

Next step

Layout ▾

Tutorial content ▾

3/3

## : Implementacija vanjskih shema pomoću virtualnih relacija

U ovom koraku tutoriala koriste se relacije knjDjelo i autori. Navedene relacije ne postoje u bazi podataka i potrebno ih je prethodno kreirati.

U relaciji knjDjelo nalaze se podaci o knjigama koja se prodaju u gradskoj knjižari. Cijene knjiga navode se u različitim valutama, na primjer dolarima (USD) i kunama (HRK). Knjige u knjižari tiskane su na raznim jezicima, na primjer Hrvatski (HR), Engleski (ENG) ili Francuski (FR). Knjižara ima odjele koji se bave nabavkom knjiga, prodajom i računovodstvom.

knjDjelo

sifDjelo	sifautor	naslov	cijena	valuta	jezikTisak	prodanoPrimjeraka
4445	1	Fahrenheit 451	15	USD	ENG	2
4422	2	Na Cesti	12	USD	ENG	1
4111	3	Suite Francaise	95	HRK	FR	3
3245	4	Madame Bovary	150	HRK	FR	5
2345	5	O ljepoti	150	HRK	HR	0
3476	6	Snijeg	220	HRK	HR	1
2134	6	Zovem crvena	180	HRK	HR	2
4535	5	Bijeli zub	20	USD	ENG	11

autori

sifautor	imeAutor	prezimeAutor
1	Ray	Bradbury
2	Jack	Kerouac
3	Irene	Nemirovsky
4	Gustave	Flaubert
5	Zadie	Smith
6	Orhan	Pamuk

Za vježbu napišite SQL naredbe za definiranje sheme i sadržaja relacija naslov i knjiga.

rješenje

```
CREATE TABLE autor(  
    sifautor integer PRIMARY KEY  
    , imeAutor    VARCHAR(30) NOT NULL  
    , prezimeAutor VARCHAR(30) NOT NULL  
);  
  
CREATE TABLE knjDjelo  
    (sifDjelo INTEGER PRIMARY KEY  
    , sifautor INTEGER NOT NULL REFERENCES autor(sifAutor)  
    , naslov VARCHAR (300) NOT NULL  
    , cijena DECIMAL(7,2) NOT NULL  
    , valuta VARCHAR(3) NOT NULL  
    , jezikTisak VARCHAR(3) NOT NULL  
    , prodanoPrimjeraka INTEGER NOT NULL);  
  
INSERT INTO autor VALUES (1, 'Ray',      'Bradbury');  
INSERT INTO autor VALUES (2, 'Jack',     'Kerouac');  
INSERT INTO autor VALUES (3, 'Irene',    'Nemirovsky');  
INSERT INTO autor VALUES (4, 'Gustave',   'Flaubert');  
INSERT INTO autor VALUES (5, 'Zadie',    'Smith');  
INSERT INTO autor VALUES (6, 'Orhan',    'Pamuk');  
  
  
INSERT INTO knjDjelo VALUES      (4445, 1, 'Fahrenheit 451', 15      , 'USD', 'ENG', 2);  
INSERT INTO knjDjelo VALUES      (4422, 2, 'Na Cesti', 12      , 'USD', 'ENG', 1);  
INSERT INTO knjDjelo VALUES      (4111, 3, 'Suite Francaise', 95      , 'HRK', 'FR', 3);  
INSERT INTO knjDjelo VALUES      (3245, 4, 'Madame Bovary', 150      , 'HRK', 'FR', 5);  
INSERT INTO knjDjelo VALUES      (2345, 5, 'O ljepoti', 150      , 'HRK', 'HR', 0);  
INSERT INTO knjDjelo VALUES      (3476, 6, 'Snijeg', 220      , 'HRK', 'HR', 1);  
INSERT INTO knjDjelo VALUES      (2134, 6, 'Zovem se crvena', 180      , 'HRK', 'HR', 2);  
INSERT INTO knjDjelo VALUES      (4535, 5, 'Bijeli zub', 20      , 'USD', 'ENG', 11);
```

Potrebno je definirati nekoliko različitih vanjskih shema za različite kategorije korisnika.

Primjer 1.

Korisnik koji se bavi nabavkom knjiga na hrvatskom jeziku i čije su cijene izražene u kunama, želi imati popis takvih knjiga u knjižari.

rješenje

Stvorimo virtualnu relaciju vHRKnjige. U upitu koji definira virtualnu relaciju provjeravamo vrijednosti atributa valuta i jezik\_tisak. Valuta traženih knjiga mora biti Hrvatska kuna (HRK) i jezik mora biti Hrvatski (HR).

```
CREATE VIEW vHRKnjige AS  
SELECT imeAutor, prezimeAutor, naslov  
FROM autor  
NATURAL JOIN knjDjelo  
WHERE jezikTisak ='HR';
```

Izvršavanjem naredbe:

```
SELECT * FROM vHRKnjige
```

dobivamo: vHRKnjige

ime	prezime	naslov
Zadie	Smith	O ljepoti
Orhan	Pamuk	Snijeg
Orhan	Pamuk	Zovem se crvena

Primjer 2.

Korisnik koji se bavi nabavkom stranih knjiga, koje su tiskane na stranom jeziku ili čije su cijene izražene u nekoj stranoj valuti, želi imati popis takvih knjiga u knjižari.

rješenje

Stvorimo virtualnu relaciju vINOKnjige. U upitu koji definira virtualnu relaciju provjeravamo vrijednosti atributa valuta i jezik. Ukoliko jezik nije Hrvatski (HR) ili valuta nije HRK, tada smatramo da se radi o stranoj knjizi.

```
CREATE VIEW vINOKnjige AS
SELECT imeAutor, prezimeAutor, naslov
FROM autor
NATURAL JOIN knjDjelo
WHERE jezikTisak <> 'HR'
OR valuta <> 'HRK';
```

Izvršavanjem naredbe:

```
SELECT * FROM vINOKnjige
```

dobivamo:

vINOKnjige

ime	prezime	naslov
Ray	Bradbury	Fahrenheit 451
Jack	Kerouac	Na Cesti
Zadie	Smith	Bijeli zub
Irene	Nemirnovsky	Suite Francaise
Gustave	Flaubert	Madame Bovary

Primjer 3.

Osobe iz računovodstva zanimaju prihodi od prodaje knjiga po različitim valutama. Prihod se izračunava kao: broj prodanih knjiga \* jedinična cijena knjige. Prihodi u raznim valutama se ne mogu miješati, to jest moraju biti ispisani odvojeno.

rješenje

Stvorimo virtualnu relaciju vZaradaPoValutama. U upitu koji definira virtualnu relaciju potrebno je izvesti grupiranje po atributu valuta, kako bi dobili prihode po pojedinoj valuti. Prihod se izračuna kao jedinična cijena knjige \* broj prodanih knjiga.

```
CREATE VIEW vZaradaPoValutama (valuta, prihod) AS
SELECT valuta, SUM(cijena*prodanoPrimjeraka)
FROM knjDjelo
GROUP BY valuta;
```

Izvršavanjem naredbe:

```
SELECT * FROM vZaradaPoValutama
```

dobivamo:

vZaradaPoValutama

valuta	prihod
USD	262
HRK	1615

Primjer 4.

Osobe iz računovodstva zanimaju i prihodi od prodanih knjiga po jezicima (hrvatskom, engleskom i francuskom). S obzirom da knjige tiskane na istom jeziku mogu imati cijene u različitim valutama, prihod se grupirati i prema valuti cijene.

rješenje

Stvorimo virtualnu relaciju vZaradaPoJezicima. U upitu koji definira virtualnu relaciju grupiramo po atributu jezik\_tisak, te po atributu valuta kako bismo dobili sumu prihoda od prodanih knjiga prema jezicima i valuti koji su zastupljeni u bazi. Prihod se izračuna kao jedinična cijena knjige \* broj prodanih knjiga.

```
CREATE VIEW vZaradaPoJezicima (jezik, valuta, prihod) AS
SELECT jezikTisak, valuta, SUM(cijena*prodanoPrimjeraka)
FROM knjDjelo
GROUP BY jezikTisak, valuta;
```

Izvršavanjem naredbe:

```
SELECT * FROM vZaradaPoJezicima
```

dobivamo:

vZaradaPoJezicima

jezik	valuta	prihod
ENG	USD	262
HR	HRK	580
FR	HRK	1035

Primjer 5.

Vlasniku knjižare treba popis svih knjiga (bilo stranih ili domaćih) ali sa cijenom navedenom u kunama (za knjige sa cijenom u dolarima potrebno je cijenu konvertirati u kune po tečaju 5,5 HRK za 1 USD))

rješenje

Stvorimo virtualnu relaciju vKnjigeHRK. U upitu koji definira virtualnu relaciju povezujemo tablice autori i naslov preko zajedničkog ključa te provjeravamo cijenu knjige. Ako je cijena knjige naznačena u dolarima (USD) potrebno je pomnožiti tu vrijednost sa tečajem dolara (5,5 HRK za jedan dolar) da dobijemo približnu vrijednost knjige u kunama.

```
CREATE VIEW vKnjigeHRK (ime, prezime, naslov, cijena_u_kunama) AS
SELECT imeAutor, prezimeAutor, naslov
, CASE
    WHEN valuta='USD' THEN cijena*5.5
    ELSE cijena
  END AS prihod
FROM knjDjelo
NATURAL JOIN autor;
```

Izvršavanjem naredbe:

```
SELECT * FROM vKnjigeHRK
```

dobivamo:

vKnjigeHRK

ime	prezime	naslov	cijena_u_kunama
Ray	Bradbury	Fahrenheit 451 82,	50
Jack	Kerouac	Na Cesti	66
Irene	Nemirovsky	Suite Francaise	95
Gustave	Flaubert	Madame Bovary	150

Feedback

ime	prezime	naslov	cijena_u_kunama
Zadie	Smith	O ljepoti	150
Orhan	Pamuk	Snijeg	220
Orhan	Pamuk	Zovem se crvena	180
Zadie	Smith	Bijeli zub	110

Question

Playground

Ovaj zadatak odnosi se na bazu podataka studAdmin.

Kao u tablici dolje, ispisati po akademskim godinama najveći, najmanji i prosječan težinski prosjeka ocjena studenata. Težinski prosjek ocjena računati temeljem položenih ispita, a akademsku godinu kojoj ispit pripada odredite temeljem datuma ispitnog roka.

Uzmite u obzir da akademska godina počinje 1.10., a završava 31.9.

akGodina	najVTezProsjek	najMTezProsjek	prosTezProsjek
2017	5.0000000000000000	2.0000000000000000	3.4315314538396448
2018	5.0000000000000000	2.0000000000000000	3.6397169211860113
2019	5.0000000000000000	2.0000000000000000	3.5506029553890420
2020	5.0000000000000000	2.0000000000000000	3.3427362776380259

Napomena: dozvoljeno je zadatak riješiti pomoću više sql naredbi. Ako se rješenje sastoji od više naredbi potrebno ih je odvojiti s točka-zarezom.

rješenje

```
CREATE VIEW tezinskiProsjek (JMBAG, akGodina, tezProsjek) AS
SELECT JMBAG
      , EXTRACT(YEAR FROM datumRok - '9 mon'::interval) akGodina
      , SUM(ocjena*ECTSBod)/Sum(ECTSBod) AS tezinskipros0cj
FROM ispit
NATURAL JOIN predmet
WHERE ocjena > 1
GROUP BY JMBAG, akGodina;

SELECT akGodina
      , MAX(tezProsjek) najVTezProsjek
      , MIN(tezProsjek) najMTezProsjek
      , AVG(tezProsjek) prosTezProsjek
FROM tezinskiProsjek
GROUP BY akGodina
ORDER BY akGodina;
```

1

2

3

4

5

6

Run

Save