

## Zadaci za vježbu

(uz predavanja 11 - Privremene i virtualne relacije)

### 1. Privremene i virtualne relacije

Zadatak se odnosi na bazu **studAdmin**. U bazi studAdmin postoje relacije **student** i **upisanPredmet** čije su relacijske sheme definirane na sljedeći način:

STUDENT = {jmbag, ime, prezime, jmbg, spol, datrod, pbrrod, pbrstan}  
PK<sub>student</sub> = {jmbag}

UPISANPREDMET = {jmbag, sifpredmet, akgodina, ozngrupa, datumocjena, ocjena}  
PK<sub>upisanpredmet</sub> = {jmbag, sifpredmet, }

Želja je korisnika izdvojiti jmbag, prezime i ime studenata koji su u akademskoj godini 2006 imali prosjek veći ili jednak 4,1.

Budući da planira neko vrijeme analizirati podatke o tim studentima, korisnik ne želi stalno postavljati SQL upit

```
SELECT upisanpredmet.jmbag, student.prezime, student.ime,  
AVG(upisanpredmet.ocjena)  
FROM upisanpredmet, student  
WHERE akgodina=2006 AND student.jmbag=upisanpredmet.jmbag  
GROUP BY upisanpredmet.jmbag, student.prezime, student.ime  
HAVING AVG(ocjena)>=4.1;
```

Umjesto toga korisnik će stvoriti:

- a) privremenu relaciju
- b) virtualnu relaciju

**Zadatak 1.1.** Stvorite privremenu relaciju **jakoDobri** i napunite je podacima na način da odgovara gornjem SQL upitu odnosno opisu korisnikove želje. Neka je relacijska shema zadana na sljedeći način:

JAKODOBRI = {matbr, prezime, ime, prosjek}.

**Zadatak 1.2.** Stvorite virtualnu relaciju **vrhunskiStudenti** i napunite je podacima na način da odgovara gornjem SQL upitu odnosno opisu korisnikove želje. Neka je relacijska shema relacije zadana na isti način kao i za privremenu relaciju **jakoDobri**:

VRHUNSKISTUDENTI = {matbr, prezime, ime, prosjek}.

### **Zadatak 1.3.**

- a) Iz privremene relacije **jakoDobri** ispišimo prezime studenata i njihov prosjek, počevši od onog s najboljim prosjekom (ukoliko studenti imaju jednaki prosjek, poredajmo ih po abecedi).
- b) Iz virtualne relacije **vrhunskiStudenti** ispišimo prezime studenata i njihov prosjek, počevši od onog s najboljim prosjekom (ukoliko studenti imaju jednaki prosjek, poredajmo ih po abecedi). Dakle, u ispisu se traži isto što se tražilo u slučaju A.
- c) Usporedimo ispise u slučaju a) i slučaju b). Razlikuju li se?

**Zadatak 1.4.** U međuvremenu se naknadno upisao student Bill Clinton i položio ispit iz Fizike 1 dobivši ocjenu 5.

Administrator baze studAdmin dodaje podatke o novom studentu:

```
INSERT INTO student VALUES ('0036444444', 'Bill', 'Clinton',  
'1908946339999', 'M', '19.08.1946', NULL, NULL);
```

```
INSERT INTO upisanpredmet VALUES ('0036444444', 34, 2006, 'A-H',  
'01.03.2007', 5);
```

Što se događa sa sadržajem privremene odnosno virtualne relacije nakon ažuriranja? Izvršimo ispise na isti način kao u slučaju a) i b) u Zadatku 1.3!

**Zadatak 1.5.** U međuvremenu je još i studentica Jakovljević (vidjeti rješenja Zadatka 1.3 i Zadatka 1.4) izašla na ispit iz Baza podataka i dobila ocjenu 3:

```
INSERT INTO upisanpredmet VALUES ('0555005455', 34, 2006, 'A-H',  
'30.05.2007', 3);
```

Izvršimo ispis nad sadržajem virtualne relacije na isti način kao u slučaju b) u Zadatku 1.4 i promotrimo je li došlo do promjena u odnosu na Zadatak 1.4.!

**Zadatak 1.6.** Prilikom rada nad privremenom relacijom **jakoDobri** odnosno virtualnom relacijom **vrhunskiStudenti** došlo je do ispadanja električne mreže. Hoće li to imati utjecaja na rad korisnika (pretpostavimo da nam računala nisu imala bateriju)?

## **2. Migrirajuće n-torke**

**Zadatak 2.1.** Želimo promatrati koliko su ukupno ECTS bodova ostvarili studenti studenti slušanjem svih predmeta od početka studija do danas. Načinimo dvije virtualne relacije čija je relacijska shema zadana na sljedeći način:

ZBROJECTS = {mbr, prezime, ukupnobodova}.

Pritom je *mbr* matični broj studenta (odgovara atributu *jmbag* u relacijama **student** i **upisanPredmet**), *prezime* je prezime studenta (odgovara atributu *prezime* u relaciji **student**), a *ukupnobodova* je suma svih bodova koje je student ostvario tijekom svog studiranja, od početka do danas i mora se dobiti korištenjem agregacijske funkcije

Potrebno je načiniti dvije virtualne relacije koje odgovaraju relacijskoj shemi ZBROJECTS. Virtualna relacija **odslusaliMalo** sadrži studente čija suma ECTS bodova je manja od 25. Virtualna relacija **odslusaliSrednje** sadrži studente čija suma ECTS bodova je veća ili jednaka 25 i manja od 50. Ispišite sadržaj ovih relacija, relacije **odslusaliMalo** od najvećeg broja ukupnih bodova prema manjem, a relacije **odslusaliSrednje** od manjeg prema većem.

**Zadatak 2.2.** U međuvremenu je student Zekanović (*jmbag* 0555005754) u akademskoj godini 2006. odslušao predmet Informacijske mreže (*sifpred*=30) u grupi A-B. Položio ga je 1.3.2007. ocjenom 4. Registriramo li ovaj događaj izvršavanjem upita

```
INSERT INTO upisanpredmet  
VALUES ('0555005754', 30, 2006, 'A-B', '01.03.2007', 4);
```

Događaju li se kakve promjene nad sadržajem virtualnih relacija **odslusaliMalo** i **odslusaliSrednje**? Izvršimo ponovno ispis sadržaja ovih relacija na način jednak onom u Zadatku 2.1.

**Zadatak 2.3.** Pokušajmo studentima koji su odslušali predmeta u ukupnoj vrijednosti 24 ECTS boda promijeniti prezime u vrijednost *UNKNOWN*, na način da to učinimo izravno preko virtualne relacije.

**Zadatak 2.4.** Načinimo virtualnu relaciju **punoBodova** koja ima pregled nad šifrom predmeta, nazivom predmeta i brojem ECTS bodova iz relacije **predmet** (atributi neka se zovu redom *sifra*, *naziv* i *ects*), ali samo za predmete koji nose 7 ili više ECTS bodova.

Pokušajmo dodati novi predmet (*sifra*=111, *naziv*=Novi, *ects*=4) izravno preko virtualne relacije. Ispišimo sadržaj relacije **predmet** za sve šifre veće od 100.

**Zadatak 2.5.** Načinimo virtualnu relaciju **novihPunoBodova** koja ima pregled nad šifrom predmeta, nazivom predmeta i brojem ECTS bodova iz relacije **predmet** (atributi neka se zovu redom *sifra*, *naziv* i *ects*), ali samo za predmete koji nose 7 ili više ECTS bodova. Zabranimo migraciju n-torki tj. unos, promjenu ili brisanje n-torki putem ove virtualne relacije ukoliko se te n-torke ne nalaze u toj relaciji.

Pokušajmo dodati novi predmet (*sifra*=111, *naziv*=Novi, *ects*=4) izravno preko virtualne relacije. Pokušajmo isto učiniti za predmet (*sifra*=112, *naziv*=Najnoviji, *ects*=4).

### 3. Implementacija eksternih shema pomoću virtualnih relacija

Zadane su tablice Naslov i Autori. U prvoj tablici nalaze se podaci o djelima određenih autora, koja se prodaju u gradskoj knjižari. Cijene knjiga ne navode se u istoj valuti već u nekoliko njih, točnije u dolarima (USD) i kunama (HRK). Sve knjige u knjižari, nisu tiskane na hrvatskom jeziku, već se neke prodaju na originalnim jezicima (engleskom (ENG) i francuskom (FR)). Knjižara zapošljava ljude koji se bave nabavkom knjiga, što stranih što domaćih, te računovodstvom.

Tablica: Naslov

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

Tablica: Autori

sifautora	ime	prezime
1	Ray	Bradbury
2	Jack	Kerouac
3	Irene	Nemirovsky
4	Gustave	Flaubert
5	Zadie	Smith
6	Orhan	Pamuk

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

- 3.1. Korisnik koji se bavi nabavkom knjiga na hrvatskom jeziku (čije su cijene izražene u kunama) želi imati popis svih informacija o takvim knjigama u knjižari
- 3.2. Korisnik koji se bavi nabavkom knjiga na stranim jezicima (čije su cijene izražene u nekoj stranoj valuti) želi imati popis svih informacija o takvim knjigama u knjižari

- 3.3. Osobe iz računovodstva zanimaju prihodi od prodaje knjiga po različitim valutama
- 3.4. Osobe iz računovodstva zanimaju prihodi od prodanih knjiga po jezicima (hrvatskom, engleskom i francuskom)
- 3.5. 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 (množenjem sa pet))

## RJEŠENJA:

### Zadatak 1.1:

Stvorimo privremenu relaciju **jakoDobri** prema zadanoj relacijskoj shemi. Kako bismo znali definirati tipove atributa u privremenoj relaciji, moramo znati kakve će se vrijednosti u nju spremati (atribut *matbr* odgovara atributu *jmbag* studenta i sl.).

```
CREATE TEMP TABLE jakoDobri (  
    matbr CHAR(10)  
    , prezime NCHAR(25)  
    , ime NCHAR(25)  
    , prosjek DECIMAL(3,2));
```

Potom privremenu relaciju napunimo željenim podacima.

```
INSERT INTO jakoDobri  
    SELECT upisanpredmet.jmbag, student.prezime, student.ime,  
    AVG(upisanpredmet.ocjena)  
    FROM upisanpredmet, student  
    WHERE akgodina=2006 AND student.jmbag=upisanpredmet.jmbag  
    GROUP BY upisanpredmet.jmbag, student.prezime, student.ime  
    HAVING AVG(ocjena)>=4.1;
```

### Zadatak 1.2.

Kod izrade virtualne relacije ne koriste se odvojene naredbe CREATE i INSERT, već se „posao“ naredbe INSERT obavi u naredbi CREATE:

```
CREATE VIEW vrhunskiStudenti (matbr  
    , prezime  
    , ime  
    , prosjek) AS  
    SELECT upisanpredmet.jmbag, student.prezime, student.ime,  
    AVG(upisanpredmet.ocjena)  
    FROM upisanpredmet, student  
    WHERE akgodina=2006 AND student.jmbag=upisanpredmet.jmbag  
    GROUP BY upisanpredmet.jmbag, student.prezime, student.ime  
    HAVING AVG(ocjena)>=4.1;
```

### Zadatak 1.3.

a) ispis iz privremene relacije

```
SELECT prezime, prosjek  
FROM jakoDobri  
ORDER BY prosjek DESC, prezime;
```

### jakoDobri

prezime	prosjeck
Viola	5
Smiljanić	4,8
Maletić	4,4
Ratoša	4,4
Jakovljević	4,2
Mandić	4,2

b) ispis iz virtualne relacije

```
SELECT prezime, prosjek  
FROM vrhunskiStudenti  
ORDER BY prosjek DESC, prezime;
```

### vrhunskiStudenti

prezime	prosjeck
Viola	5
Smiljanić	4,8
Maletić	4,4
Ratoša	4,4
Jakovljević	4,2
Mandić	4,2

c) ispisi su identični.

## Zadatak 1.4.

Izvršimo ponovno upite

```
SELECT prezime, prosjek  
FROM jakoDobri  
ORDER BY prosjek DESC, prezime;
```

odnosno

```
SELECT prezime, prosjek  
FROM vrhunskiStudenti  
ORDER BY prosjek DESC, prezime;
```

U slučaju privremene relacije nema nikakvih promjena u odnosu na trenutak kad smo je načinili. Privremena relacija ne registrira naknadne promjene nad sadržajem izvorne relacije iz koje je izvukla podatke.

U virtualnu relaciju, naprotiv, dodan je novi redak, budući da student Clinton ima prosjek veći od 4,1.

### jakoDobri

prezime	prosjeak
Viola	5
Smiljanić	4,8
Maletić	4,4
Ratoša	4,4
Jakovljević	4,2
Mandić	4,2

### vrhunskiStudenti

prezime	prosjeak
Clinton	5
Viola	5
Smiljanić	4,8
Maletić	4,4
Ratoša	4,4
Jakovljević	4,2
Mandić	4,2

### Zadatak 1.5.

Polaganjem novog ispita, studentici Jakovljević je prosjek pao na 4,0 tj. manje od 4,1. Izvođenjem upita

```
SELECT prezime, prosjek  
FROM vrhunskiStudenti  
ORDER BY prosjek DESC, prezime;
```

Dobiva se sljedeći rezultat:

### vrhunskiStudenti

prezime	prosjeak
Clinton	5
Viola	5
Smiljanić	4,8
Maletić	4,4
Ratoša	4,4
Mandić	4,2

Vidimo da se broj studenata u relaciji smanjio za jedan redak (tj. nema više studentice Jakovljević).

### Zadatak 1.6.

Prekidom napajanja (pretpostavljamo da računala nisu imala bateriju) računala su se ugasila pa je time automatski došlo je do prekida SQL-sjednice.



Slučaj A: prekidom SQL-sjednice privremena relacija se izbrisala. Korisnik mora nanovo stvoriti privremenu relaciju.

Slučaj B: virtualna relacija i dalje postoji; nema efekata.

## ***Migrirajuće n-torke***

### **Zadatak 2.1.**

Naredbe za stvaranje ovih dviju virtualnih relacija su sljedeće:

```
CREATE VIEW odslusaliMalo (mbr
                        , prezime
                        , ukupnobodova) AS
    SELECT upisanpredmet.jmbag, student.prezime, SUM(predmet.ectsbod)
    FROM predmet, upisanpredmet, student
    WHERE predmet.sifpredmet=upisanpredmet.sifpredmet AND
    upisanpredmet.jmbag=student.jmbag
    GROUP BY upisanpredmet.jmbag, student.prezime
    HAVING SUM(predmet.ectsbod)<25;
```

```
CREATE VIEW odslusaliSrednje (mbr
                        , prezime
                        , ukupnobodova) AS
    SELECT upisanpredmet.jmbag, student.prezime, SUM(predmet.ectsbod)
    FROM predmet, upisanpredmet, student
    WHERE predmet.sifpredmet=upisanpredmet.sifpredmet AND
    upisanpredmet.jmbag=student.jmbag
    GROUP BY upisanpredmet.jmbag, student.prezime
    HAVING SUM(predmet.ectsbod)>=25 AND SUM(predmet.ectsbod)<50;
```

Nakon izvršavanja naredaba za ispis

```
SELECT * FROM odslusaliMalo ORDER BY ukupnobodova DESC;
SELECT * FROM odslusaliSrednje ORDER BY ukupnobodova;
```

ispisuje se 73 retka virtualne relacije **odslusaliMalo** i 133 retka virtualne relacije **odslusaliSrednje**.

Početak ispisa za obje relacije dan je u nastavku:

#### **odslusaliMalo (ZBROJECTS)**

mbr	prezime	ukupnobodova
0555001510	Palle	24
0555004918	Krizmanić	24
0555004351	Stamenković	23
0555005754	Zekanović	23
0555002698	Kostelenski	22
0555004388	Đuričić	22
0555003008	Gavrić	22

0555005231	Jurić	22
0555005114	Marčetić	22
0555005210	Šubara	22
0555005226	Germanović	22
0555005289	Šarić	22
0555002752	Jusufagić	22

Napomena: slijedi još određen broj redaka

#### odslusaliSrednje (ZBROJECTS)

mbr	prezime	ukupnobodova
0555000604	Splait	25
0555003013	Čiković	25
0555001872	Kučan	25
0555002320	Vrcelj	25
0555000529	Đukić	25
0555000560	Stanković	25
0555003968	Srdoč	29
0555004143	Flego	29
0555004276	Cupan	29
0555003926	Bobinac	29
0555005497	Otulić	29
0555005572	Runko	29
0555003898	Stojanović	29
0555003814	Štifter	29
0555003856	Brandić	29
0555003765	Rodić	29
0555003835	Polumirac	29
0555004073	Kovacic	29
0555004122	Perković	29
0555004229	Šaban	29
0555005679	Šegulja	29
0555003947	Krmpotić	29

Napomena: slijedi još određen broj redaka

#### Zadatak 2.2.

Student Zekanović je prije upisivanja informacijskih mreža ukupno ostvario 23 ECTS boda. Budući da Informacijske mreže nose 6 bodova, imat će 29 bodova. Zbog toga se njegovi podaci više neće nalaziti u virtualnoj relaciji **odslusaliMalo**: ona ima 72 retka umjesto ranijih 73. Međutim, Student Zekanović je ušao u kategoriju studenata čiji podaci se nalaze u virtualnoj relaciji **odslusaliSrednje**. Uistinu, u njoj se nalazi redak više. Izvršimo li naredbe

```
SELECT * FROM odslusaliMalo ORDER BY ukupnobodova DESC;
SELECT * FROM odslusaliSrednje ORDER BY ukupnobodova;
```

Vidimo da se u u virtualnoj relaciji **odslusaliMalo** na 4. mjestu više ne nalazi student Zekanović. On se pojavljuje kao novi, dodatni redak u virtualnoj relaciji **odslusaliSrednje**.

**odslusaliMalo (ZBROJECTS)**

mbr	prezime	ukupnobodova
0555001510	Palle	24
0555004918	Krizmanić	24
0555004351	Stamenković	23
0555002698	Kostelenski	22
0555004388	Đuričić	22
0555003008	Gavrić	22
0555005231	Jurić	22
0555005114	Marčetić	22
0555005210	Šubara	22
0555005226	Germanović	22
0555005289	Šarić	22
0555002752	Jusufagić	22

Napomena: slijedi još određen broj redaka

**odslusaliSrednje (ZBROJECTS)**

mbr	prezime	ukupnobodova
0555000604	Splait	25
0555003013	Čiković	25
0555001872	Kučan	25
0555002320	Vrcelj	25
0555000529	Đukić	25
0555000560	Stanković	25
0555003968	Srdoč	29
0555004143	Flego	29
0555004276	Cupan	29
0555003926	Bobinac	29
0555005497	Otulić	29
0555005572	Runko	29
0555003898	Stojanović	29
0555003814	Štifter	29
0555003856	Brandić	29
0555003765	Rodić	29
0555003835	Polumirac	29
0555004073	Kovacic	29
0555004122	Perkovic	29
0555004229	Šaban	29
0555005679	Šegulja	29
<b>0555005754</b>	<b>Zekanović</b>	<b>29</b>
0555003947	Krmpotić	29

Napomena: slijedi još određen broj redaka

**Zadatak 2.3.**

Pokušamo li izvršiti naredbu

```
UPDATE odslusaliMalo  
SET prezime='UNKNOWN'  
WHERE ukupnobodova=24;
```

SUBP će javiti grešku (No UPDATE Permission). Zašto?

Virtualna relacija je neizmjenjiva. Sve virtualne relacije koje u glavnom SELECT dijelu svoje definicije koriste spoj više relacija su neizmjenjive. Također, neizmjenjive su i virtualne relacije koje se u glavnom dijelu SELECT naredbe odnose na samo jednu relaciju, no koriste izraz GROUP BY (naša relacija koristi GROUP BY, prema tome bila bi neizmjenjiva i kad bi vrijednosti svojih atributa uzimala iz samo jedne relacije).

Zapravo bi trebalo izvesti upit:

```
UPDATE student
SET prezime='UNKNOWN'
WHERE jmbag IN (
    SELECT upisanpredmet.jmbag
    FROM predmet, upisanpredmet, student
    WHERE predmet.sifpredmet=upisanpredmet.sifpredmet AND
        upisanpredmet.jmbag=student.jmbag
    GROUP BY upisanpredmet.jmbag
    HAVING SUM(predmet.ectsbod)=24
);
```

## Zadatak 2.4.

Virtualna relacija **punoBodova** stvara se naredbom

```
CREATE VIEW punoBodova (sifra
                        , naziv
                        , ects) AS
SELECT sifpredmet, naziv, ectsbod
FROM predmet
WHERE predmet.ectsbod>6;
```

Sadržaj virtualne relacije (14 redaka):

```
SELECT * FROM punoBodova;
```

sifra	naziv	ects
16	Osnove elektrotehnike	7
24	Matematika 2	7
18	Operacijski sustavi	7
36	Električni krugovi	7
11	Seminar	10
25	Matematika 1	7
13	Projektiranje digitalnih sustava	7
41	Automati, formalni jezici i jezični procesori II	7
10	Seminar	12
35	Elektronika 1	7
44	Arhitektura i organizacija računala	7
49	Električni strojevi II - konstrukcijske vježbe	7
54	Energetski sistemi i bilance - konstrukcijske vježbe	7
59	Električni strojevi III - konstrukcijske vježbe	7

Pokuša li se izvršiti tražena naredba unosa novog predmeta:

```
INSERT INTO punoBodova VALUES (111, 'Novi', 4);
```

SUBP javlja uspješan ishod izvedbe

Međutim, sadržaj virtualne relacije se ne mijenja! Logično, budući da virtualna relacija sadrži samo predmete koji nose 7 ili više ECTS bodova. Međutim, ukoliko pogledamo sadržaj temeljne relacije **predmet** (ispišemo sadržaj relacije **predmet** za sve šifre veće od 100) vidimo da je naredba INSERT preko virtualne relacije **punoBodova** indirektno stvorila novu n-torku u relaciji **predmet**.

```
SELECT *  
FROM predmet  
WHERE sifpredmet>100;
```

Rezultat naredbe je:

sifpredmet	naziv	ectsbod	ukbrsatitjedno
111	Novi	4	(null)

Primijetimo da je vrijednost atributa *ukbrsatitjedno* za danu n-torku 0. Dakako, virtualna relacija **punoBodova** nije se odnosila na atribut *ukbrsatitjedno* pa stoga niti nije omogućavala postavljanje vrijednosti atributa *ukbrsatitjedno*.

## Zadatak 2.5.

Virtualna relacija **novihPunoBodova** stvara se naredbom

```
CREATE VIEW novihPunoBodova (sifra  
                           , naziv  
                           , ects) AS  
  SELECT sifpredmet, naziv, ectsbod  
  FROM predmet  
  WHERE predmet.ectsbod>6  
WITH CHECK OPTION;
```

Izraz WITH CHECK OPTION onemogućuje da se u ovu virtualnu relaciju pokušaju dodati n-torke koje u nju ne pripadaju odnosno da se pokuša izmijeniti ili izbrisati takve n-torke.

Sadržaj virtualne relacije sastoji se od istih 14 redaka kao i na početku Zadatka 2.4:

Izraz

```
INSERT INTO novihPunoBodova VALUES (111, 'Novi', 4);
```

nije moguće izvršiti. Tom naredbom stvorile bi se n-torke u relaciji predmet te eventualno još nekim virtualnim relacijama, a ne bi se stvorila n-torka u samoj relaciji **novihPunoBodova**.

Izraz

```
INSERT INTO novihPunoBodova VALUES (112, 'Najnoviji', 8);
```

se uspješno izvršava, a u relaciji pojavljuje se nova n-torka.

```
SELECT * FROM novihPunoBodova;
```

sifra	naziv	ects
16	Osnove elektrotehnike	7
24	Matematika 2	7
18	Operacijski sustavi	7
36	Električni krugovi	7
11	Seminar	10
25	Matematika 1	7
13	Projektiranje digitalnih sustava	7
41	Automati, formalni jezici i jezični procesori II	7
10	Seminar	12
35	Elektronika 1	7
44	Arhitektura i organizacija računala	7
49	Električki strojevi II - konstrukcijske vježbe	7
54	Energetski sistemi i bilance - konstrukcijske vježbe	7
59	Elektricni strojevi III - konstrukcijske vježbe	7
111	<b>Najnoviji</b>	<b>8</b>

## Implementacija eksternih shema pomoću virtualnih relacija

Potrebno je kreirati pet eksternih shema koristeći virtualne relacije:

### Zadatak 3.1:

- Virtualna relacija *hrKnjige*:

Kreiramo tzv. pogled imena *hrKnjige* koji se temelji na SELECT naredbi u kojoj dohvaćamo tražene podatke. Provjeravamo vrijednosti atributa valuta i jezik koji moraju biti postavljeni na HRK i HR.

```
CREATE VIEW hrKnjige AS
SELECT autori.ime
      , autori.prezime
      , naslov.naslov
FROM autori, naslov
WHERE autori.sifautora = naslov.sifautora
AND naslov.valuta = 'HRK'
AND naslov.jezik = 'HR'
```

Nakon izvođenja naredbe:

```
SELECT * FROM hrKnjige
```

Rezultat bi trebao izgledati ovako:

hrKnjige

Ime	Prezime	Naslov
Zadie	Smith	O ljepoti
Orhan	Pamuk	Snijeg
Orhan	Pamuk	Zovem se crvena

### Zadatak 3.2:

- Virtualna relacija *strKnjige*:

Kreiramo tzv. pogled imena *strKnjige* koji se temelji na SELECT naredbi u kojoj dohvaćamo tražene podatke. Provjeravamo vrijednosti atributa valuta i jezik koji u ovom slučaju ne smiju imati vrijednost HRK i HR.

```
CREATE VIEW strKnjige AS
SELECT autori.ime
      , autori.prezime
      , naslov.naslov
FROM autori, naslov
WHERE autori.sifautora = naslov.sifautora
AND naslov.jezik NOT LIKE 'HR'
AND naslov.valuta NOT LIKE 'HRK'
```

Nakon izvođenja naredbe:

```
SELECT * FROM strKnjige
```

Rezultat bi trebao izgledati ovako:

strKnjige

Ime	Prezime	Naslov
Ray	Bradbury	Fahrenheit 451
Jack	Kerouac	Na Cesti
Zadie	Smith	Bijeli zub

### Zadatak 3.3:

- Virtualna relacija *zaradaPoValutama*:

Kreiramo tzv. pogled imena *zaradaPoValutama* koji se temelji na SELECT naredbi u kojoj dohvaćamo tražene podatke. Potrebno je izvršiti grupiranje atributa *valuta* da bi dobili sumu cijena po pojedinoj valuti.

```
CREATE VIEW zaradaPoValutama
( valuta
, prihod ) AS
SELECT valuta
, SUM(cijena)
FROM naslov
GROUP BY valuta
```

Nakon izvođenja naredbe:

```
SELECT * FROM zaradaPoValutama
```

Rezultat bi trebao izgledati ovako:

zaradaPoValutama

Valuta	Prihod
USD	47
HRK	795



### Zadatak 3.4:

Kreiramo tzv. pogled imena *zaradaPoJezicima* koji se temelji na SELECT naredbi u kojoj dohvaćamo tražene podatke. Potrebno je izvršiti grupiranje atributa *jezik* da bi dobili sumu cijena prodanih knjiga na jezicima koji su zastupljeni u bazi.

- Virtualna relacija *zaradaPoJezicima*:

```
CREATE VIEW zaradaPoJezicima
(jezik
, prihod) AS
SELECT jezik
      , SUM(cijena)
FROM naslov
GROUP BY jezik
```

Nakon izvođenja naredbe:

```
SELECT * FROM zaradaPoJezicima
```

Rezultat bi trebao izgledati ovako:

zaradaPoJezicima

Jezik	Prihod
ENG	47
HR	550
FR	245

### Zadatak 3.5:

- Virtualna relacija *zaradaUkupno*:

Kreiramo tzv. pogled imena *zaradaUkupno* koji se temelji na SELECT naredbi u kojoj dohvaćamo tražene podatke. Potrebno je povezati tablice *Autori* i *Naslov* preko zajedničkog ključa te provjeriti cijenu knjige. Ako je cijena knjige naznačena u dolarima potrebno je pomnožiti tu vrijednost sa brojem 5 da dobijemo približnu vrijednost knjige u kunama.

```

CREATE VIEW zaradaUkupno
(Ime
, Prezime
, Naslov
, Cijena_u_kunama) AS
SELECT autori.ime
      , autori.prezime
      , naslov.naslov
      , CASE
        WHEN valuta='USD' THEN cijena*5
        ELSE cijena*1
      END AS prihod
FROM autori, naslov
WHERE autori.sifautora = naslov.sifautora

```

Nakon izvođenja naredbe:

```
SELECT * FROM zaradaUkupno
```

Rezultat bi trebao izgledati ovako:

zaradaUkupno

Ime	Prezime	Naslov	Cijena_u_kunama
Ray	Bradbury	Fahrenheit 451	75
Jack	Kerouac	Na Cesti	60
Irene	Nemirovsky	Suite Francaise	95
Gustave	Flaubert	Madame Bovary	150
Zadie	Smith	O ljepoti	150
Orhan	Pamuk	Snijeg	220
Orhan	Pamuk	Zovem se crvena	180
Zadie	Smith	Bijeli zub	100