

Ispit iz Baza podataka

(50 bodova)

20. rujna 2017.

Zadaci 1 - 5 odnose se na bazu podataka **eTrznica** koja je prikazana na **slici 1**. Na slici **nisu prikazane** sve n-torke sadržane u relacijama.

posiljka				posiljkaArtikl		
sifPosiljka	OIB	brojArtikala	vrPoslano	sifPosiljka	sifArtikl	kolicina
1002	08855334992	3	2016-09-01 10:52:48.1542	1002	1	15
323	86247384345	1	2017-08-28 08:53:25.719	1002	3	5
987	08855334992	1	<NULL>	1002	4	10
2359	04814996133	2	2017-09-13 10:53:25.719	323	2	1
				987	4	4
				2359	1	8
				2359	2	2

kupac			artikl	
OIB	ime	prezime	sifArtikl	opis
08855334992	Marko	Horvat	1	Rajčica
86247384345	Ivana	Puh	2	Crveni luk
04814996133	Igor	Matić	3	Kesten
			4	Mladi krumpir

Slika 1.

Baza podataka **eTrznica** sadrži podatke o kupljenima artiklima koje je potrebno dostaviti kupcima slanjem pošiljki. Relacija **posiljka** sadrži podatke o pošiljkama, gdje se evidentira kupac, broj (različitih) kupljenih artikala i vrijeme kada je pošiljka poslana (može biti NULL). U relaciji **posiljkaArtikl** za svaku pošiljku su evidentirani artikli koje pošiljka sadrži, a količina je izražena u kilogramima.

U zadacima 1 - 3 napisati **po jednu SQL naredbu** kojom će se obaviti sljedeće:

- (4 boda)** Obrisati pošiljke koje zadovoljavaju jedan od sljedeća dva uvjeta:
 - pošiljka je poslana prije više od 10 godina
 - ne sadrži niti jedan artikl
- (6 bodova)** Za sve kupcije čije prezime počinje slovom 'M' ispisati OIB, ime i prezime kupca te ukupnu količinu artikala koje je kupio. Zapiše poredati silazno prema ukupnoj količini. Potrebno je ispisati i podatke za kupce (čije prezime počinje slovom 'M') koji do sada nisu naručili nijedan artikl.
Zadatak riješiti bez upotrebe podupita.
- (6 bodova)** Ispisati artikle koji su prodavaniji u trenutnoj nego li u prethodnoj godini. Uz opis artikla potrebno je ispisati prodanu količinu artikla u trenutnoj godini.
- (7 bodova)** Pretpostavite da su u bazi podataka **eTrznica** kreirane sve relacije te da su definirani svi primarni i strani ključevi. Napisati naredbe kojima će se osigurati da je podatak **brojArtikala** u relaciji **posiljka** ažuriran prema promjenama podataka u relaciji **posiljkaArtikl**. Možete pretpostaviti da je nad tablicom **posiljkaArtikl** zabranjena operacija izmjene. Ne dopustiti da se poslanim pošiljkama mogu mijenjaju podaci o artiklima, niti dopustiti premašivanje ograničenja ukupne količine artikala po pošiljci od 1000 kg (dojaviti iznimke s odgovarajućim porukama).
- (5 bodova)** Korisnik **postgres** kreirao je bazu podataka **eTrznica** i relacije u njoj sljedećim SQL naredbama:

```
CREATE DATABASE eTrznica;  
CREATE TABLE artikl (...); CREATE TABLE kupac (...); ...  
REVOKE CONNECT ON DATABASE eTrznica FROM PUBLIC;  
REVOKE ALL ON SCHEMA public FROM PUBLIC;  
REVOKE ALL ON ALL TABLES IN SCHEMA public FROM PUBLIC;
```

- Kreirati ulogu *dostavljac* i shemu *dostavljac* čiji je vlasnik uloga *dostavljac*.
- Ulozi *dostavljac* omogućiti pregled relacija **kupac**, **artikl**, **posiljka** i **posiljkaArtikl**, te promjenu atributa *vrPoslano* u relaciji **posiljka**.
- Koristeći shemu *dostavljac* omogućiti ulozi *dostavljac* pregled podataka za dostavu s mogućnošću dodjeljivanja dozvole drugim korisnicima. Podaci o dostavi sastoje se od šifre pošiljke, svih podataka o kupcu te šifre, opisa i naručene količine artikla. Podaci se dohvaćaju jednim upitom.
Nije dozvoljeno stvarati objekte u shemi public.

6. (4 boda) U bazi su pohranjene tablice r, s i t koje odgovaraju relacijskim shemama R (A, B, C), S (C,D) i T(B, E). Optimizator upita ne raspolaže informacijama o ključevima tablica, ali raspolaže sljedećim statističkim informacijama vezanim uz navedene tablice:

r (A, B, C)
 N(r) = 3000
 V(A, r) = 500
 V(B, r) = 200
 V(C, r) = 100

s (C, D)
 N(s) = 1000
 V(C, s) = 50
 V(D, s) = 100

t (B, E)
 N(t) = 600
 V(B, t) = 100
 V(E, t) = 50

Obavlja se operacija $\sigma_{B < 50}(r \bowtie s \bowtie t)$.

Objasnite koji redoslijed spajanja će odabrati optimizator i nacrtajte stablo izvođenja upita.

7. (5 bodova)

Prikazan je odsječak sadržaja relacije **artikl** u trenutku započinjanja naredbe {1}. Uz pretpostavku da osim transakcija T1, T2 i T3 niti jedna druga transakcija ne obavlja nikakvu operaciju nad relacijom **artikl**:

artikl			
xmin	xmax	sifArtikl	opis
...
5	0	3	Kesten

- objasnite što se događa pri obavljanju naredbi {4}, {5}, {6}, {7}, {8} i {9}
- odredite rezultat naredbe {10}

	Transakcija T1 T _{id} =21		Transakcija T2 T _{id} =22		Transakcija T2 T _{id} =23
{1}	BEGIN TRANSACTION;	{2}	BEGIN TRANSACTION;	{3}	BEGIN TRANSACTION;
{4}	UPDATE artikl SET opis = 'Kesten pire' WHERE sifArtikl=3;	{5}	UPDATE artikl SET opis = 'Šumski kesten' WHERE sifArtikl=3;	{6}	UPDATE artikl SET opis = 'Pitomi kesten' WHERE sifArtikl=3;
{7}	COMMIT TRANSACTION;	{8}	COMMIT TRANSACTION;	{9}	COMMIT TRANSACTION;
{10}	SELECT xmin, xmax, * FROM artikl WHERE sifArtikl = 3;				

8. (8 bodova)

Potrebno je izraditi ER model baze podataka za praćenje podataka o znanstvenim konferencijama. Za konferenciju se evidentira jedinstvena kratica, naziv, datum početka i završetka konferencije. Konferencija se održava na točno jednoj lokaciji u nekom gradu. Za lokaciju se evidentira šifra, ulica i kućni broj, a za grad šifra, naziv i poštanski broj (ne mora biti jedinstven), te država kojoj pripada (oznaka i naziv države). Konferencija uključuje više sesija za koje se evidentira redni broj, naziv, vrijeme početka i vrijeme završetka te osoba koja ju vodi (točno jedna). Redni brojevi sesija kreću od 1 za svaku konferenciju. Jedna osoba može voditi više sesija.

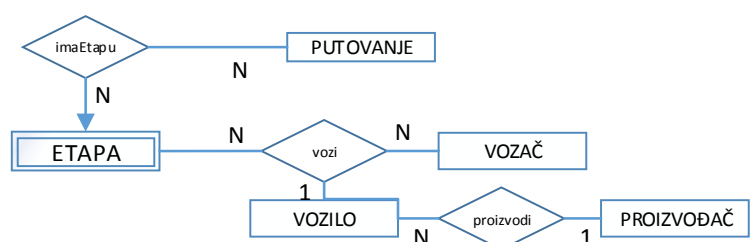
Za sesije se evidentira redoslijed izlaganja znanstvenih radova. Određeni rad na sesiji izlaže isključivo jedna osoba koja ne mora nužno biti jedan od autora rada, pri čemu osoba na jednoj sesiji može izlagati više radova. Jedan rad može imati više autora, pri čemu je potrebno evidentirati poredak autora. Za rad se bilježi šifra i naslov. Za osobe se evidentira e-mail (identificira osobu), ime i prezime, te država.

Nacrtati ER model baze podataka, navesti sheme entiteta i sheme veza (označiti ključeve). Svaki entitet (osim slabih entiteta) opisuje se isključivo vlastitim atributima. Nužno je da sve sheme zadovoljavaju 3NF.

9. (5 bodova) Za zadani ER model

- opisati sheme veza u jednakom obliku kao što su opisane sheme entiteta. Veze donjeg ER modela nemaju vlastitih atributa.
- napisati naredbe za kreiranje relacija ekvivalentnog relacijskog modela, koje će osigurati očuvanje integriteta ključa, entitetskog integriteta te referencijskog integriteta. Tipove podataka u naredbama nije potrebno navoditi.

PUTOVANJE (sifPut, nazPut)
 ETAPA (sifPut, rbrE, nazE)
 VOZAC (OIB, ime, prezime)
 VOZILO (sifV, regOznaka)
 PROIZVODAC (sifProizv, nazProizv)



RJEŠENJA

1. (4 boda)

```
DELETE FROM posiljka
WHERE CURRENT_DATE - vrPoslano > '10 year'::interval
OR NOT EXISTS (
    SELECT *
    FROM posiljkaArtikl
    WHERE posiljkaArtikl.sifPosiljka = posiljka.sifPosiljka
);
```

2. (6 bodova)

```
SELECT kupac.oib, ime, prezime, SUM(posiljkaArtikl.kolicina) ukKolicina
FROM kupac
LEFT JOIN posiljka
    ON posiljka.oib = kupac.oib AND
LEFT JOIN posiljkaArtikl
    ON posiljkaArtikl.sifPosiljka = posiljka.sifPosiljka
WHERE prezime LIKE 'M%'
GROUP BY kupac.oib, ime, prezime
ORDER BY 4 DESC;
```

3. (6 bodova)

```
SELECT artikl.opis, SUM(PA1.kolicina) prodKol
FROM artikl
    NATURAL JOIN posiljkaArtikl AS PA1
    NATURAL JOIN posiljka
WHERE EXTRACT(YEAR FROM vrPoslano) = EXTRACT(YEAR FROM CURRENT_DATE)
GROUP BY artikl.opis, PA1.sifArtikl
HAVING SUM(PA1.kolicina) >= (SELECT SUM(PA2.kolicina)
    FROM posiljkaArtikl AS PA2
    NATURAL JOIN posiljka AS P2
    WHERE PA2.sifArtikl = PA1.sifArtikl
    AND EXTRACT(YEAR FROM P2.vrPoslano) =
        EXTRACT(YEAR FROM CURRENT_DATE) - 1
    )
```

4. (7 bodova)

```
CREATE FUNCTION provPosiljkaArtikl()
RETURNS TRIGGER AS $$
DECLARE
    ukKolicina INTEGER;
    p_brArtikala INTEGER;
    p_sifPosiljka INTEGER;
BEGIN
    SELECT CASE WHEN (TG_OP = 'DELETE') THEN OLD.sifPosiljka
        ELSE NEW.sifPosiljka
    END
    INTO p_sifPosiljka;

    IF EXISTS (SELECT * FROM posiljka
        WHERE sifPosiljka = p_sifPosiljka AND vrPoslano IS NOT NULL) THEN
        RAISE EXCEPTION 'Pošiljka je već poslana';
    END IF;

    SELECT COUNT(sifArtikl), SUM(kolicina) INTO p_brArtikala, ukKolicina
    FROM posiljkaArtikl
    WHERE sifPosiljka = p_sifPosiljka;

    IF (ukKolicina > 1000) THEN
        RAISE EXCEPTION 'Prevelika količina pošiljke!';
    ELSE
        UPDATE posiljka
        SET brArtikala = p_brArtikala
        WHERE sifPosiljka = p_sifPosiljka;
```

```

END IF;

RETURN NEW;

END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER posiljkaArtiklInsDel
AFTER INSERT OR DELETE ON posiljkaArtikl
FOR EACH ROW
EXECUTE PROCEDURE provPosiljkaArtikl();

```

5. (5 bodova)

a)

```

CREATE ROLE dostavljac;
CREATE SCHEMA dostavljac AUTHORIZATION dostavljac;

```

b)

```

GRANT SELECT ON kupac, artikl, posiljka, posiljkaArtikl TO dostavljac;
GRANT UPDATE ON (vrPoslano) ON posiljka TO dostavljac;

```

c)

```

CREATE VIEW dostavljac.artikliZaDostavu AS
SELECT sifPosiljka, OIB, ime, prezime, sifArtikl, opis, kolicina
FROM posiljka
NATURAL JOIN posiljkaArtikl
NATURAL JOIN artikl
NATURAL JOIN kupac
WHERE vrPoslano IS NULL

```

```

GRANT SELECT ON dostavljac.artikliZaDostavu TO dostavljac WITH GRANT OPTION;

```

6. (4 boda)

Izračun za tablice r i t – potisnuta selekcija prema tablicama:

$$N(r_1) = N(r)/3 = 3000/3 = 1000$$

$$N(t_1) = N(t) / 3 = 600/3 = 200$$

1) prvo r i s

$$N(r \bowtie s) = N(r_1) * N(s) / \max(V(C,r), V(C,s)) = 1000 * 1000 / \max(100,50) = 1,000,000 / 100 = 10,000$$

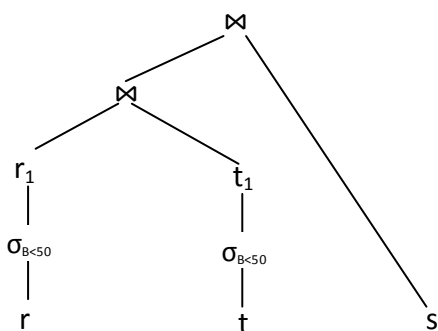
2) prvo r i t

$$N(r \bowtie t) = N(r_1) * N(t_1) / \max(V(B,r), V(B,t)) = 1000 * 200 / \max(200, 100) = 200,000 / 200 = 1,000$$

3) prvo s i t

$$N(s \bowtie t) = N(s) * N(t_1) = 1000 * 200 = 200,000$$

Najmanji međurezultat je u 2) slučaju – prvo se spajaju r i t, a zatim s. Selekcija je potisnuta do tablica r i t.



7. (5 boda)

a)

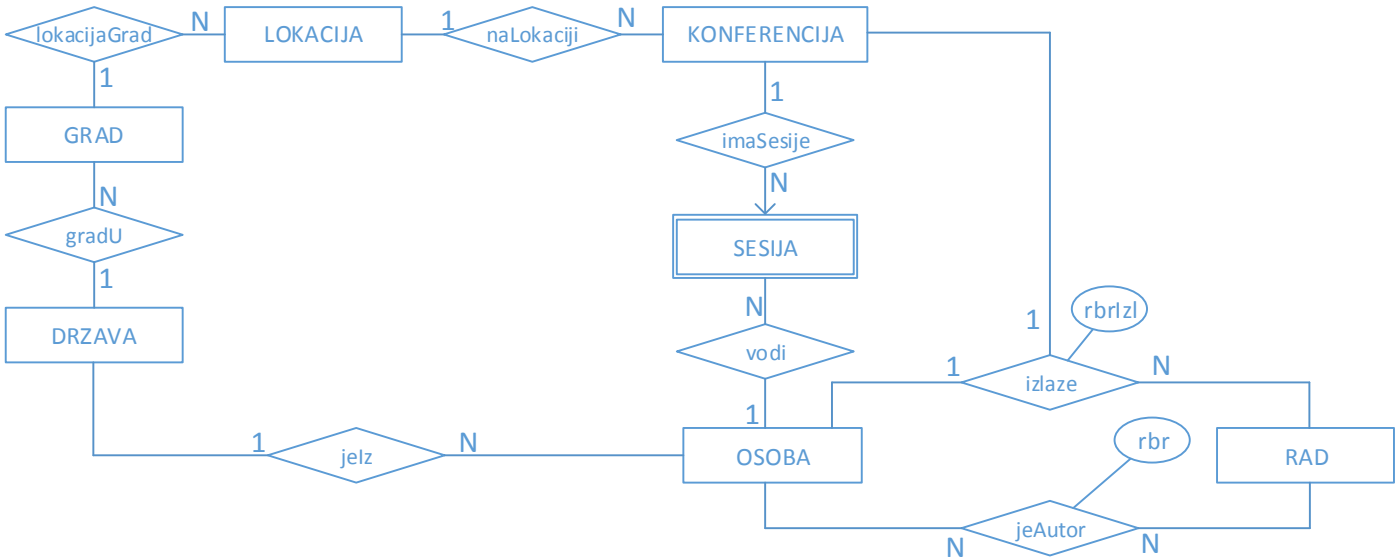
{4}	T1 postavlja ekskluzivni ključ na n-torku u artikl sa $sifArtikl = 3$.
{5}	Naredba se ne može obaviti jer je T1 postavila ekskluzivni ključ na n-torku. T2 zbog nekompatibilnosti ključeva ne može postići to isto. Čeka do završetka T1.
{6}	Naredba se ne može obaviti jer je T1 postavila ekskluzivni ključ na n-torku. T3 zbog nekompatibilnosti ključeva ne može postići to isto. Čeka do završetka T1.
{7}	Potvrđuje se transakcija T1, a odmah potom se uspješno obavlja naredba {5} i poništava (završava greškom) transakcija T3
{8}	Potvrđuje se transakcija T2
{9}	T3 nije u transakciji jer je poništena automatski neposredno nakon obavljanja naredbe {7} pa ova naredba završava greškom

b)

<i>xmin</i>	<i>xmax</i>	<i>sifArtikl</i>	<i>opis</i>
...
22	0	3	Šumski kesten

8. (8 boda)

priznaje se i:



KONFERENCIJA <u>kratica</u> naziv datPoc datZavr	LOKACIJA <u>sifLokacija</u> ulica kbr	GRAD <u>sifGrad</u> naziv pbr	DRZAVA <u>oznaka</u> naziv	SESIJA <u>kratica</u> <u>rbrSes</u> naziv vrPoc vrZavr	RAD <u>sifRad</u> naslov	OSOBA <u>e-mail</u> ime prezime
naLokaciji <u>kratica</u> sifLok	lokacijaGrad <u>sifLok</u> sifGrad	gradU <u>sifGrad</u> oznaka	jelz <u>e-mail</u> oznaka	imaSesije <u>kratica</u> <u>rbrSes</u>	vodi <u>kratica</u> <u>rbrSes</u> e-mail	
jeAutor <u>e-mail</u> <u>sifRad</u> rbr	izlaze <u>sifRad</u> <u>e-mail</u> kratica rbrSes rbrlzl alt. PK {sifRad, kratica, rbrSes}					

9. (5 bodova)

```
imaEtapu (sifPut,rbrE)
vozi (sifPut, rbrE, OIB, sifV)
proizvodi (sifProizv, sifV)
```

```
CREATE table putovanje (
    sifPut integer
, nazivPut char(30)
, PRIMARY KEY(sifPut));
```

```
CREATE table etapa (
    sifPut integer
, rbrE integer
, nazivE char(30)
, PRIMARY KEY(sifPut, rbrE)
, FOREIGN KEY (sifPut) REFERENCES putovanje(put));
```

```
CREATE table vozac (
    OIB char(11)
, ime char(30)
, prezime char (30)
, PRIMARY KEY(OIB) );
```

```
CREATE table proizvodac (
    sifProizv integer
, nazivProizv char(30)
, PRIMARY KEY(sifProizv));
```

```
CREATE table vozilo (
    sifV integer
, registracija char (10) UNIQUE
, sifProizv integer
, PRIMARY KEY(sifV)
, FOREIGN KEY(sifProizv) REFERENCES proizvodac(sifProizv) );
```

```
CREATE table imaEtapu (
    sifPut integer
, rbrE integer
, PRIMARY KEY(sifPut, rbrE)
, FOREIGN KEY(sifput, rbrE) REFERENCES etapa(sifput,rbrE)
, FOREIGN KEY(sifput) REFERENCES putovanje(sifput) );
```

```
CREATE table vozi (
    sifPut integer
, rbr integer
, OIB char(11)
, sifv integer
, PRIMARY KEY(sifPut, rbrE, OIB)
, FOREIGN KEY(sifput, rbrE) REFERENCES etapa(sifput, rbrE)
, FOREIGN KEY(OIB) REFERENCES vozac(OIB)
, FOREIGN KEY(sifV) REFERENCES vozilo(sifV) );
```