

## Završni ispit iz Baza podataka (40 bodova)

26. lipnja 2018.

Zadaci 1 - 6 se odnose na bazu podataka **telekom** mobilnog telekomunikacijskog operatora - MTO prikazanu na **slici 1**. Evidentiraju se podaci o korisnicima MTO-a (**korisnik**), paketima na koje se korisnici pretplaćuju (**paket**) te cjenicima za ponuđene pakete (**cjenik**). Ovisno o ugovorenom paketu korisnik plaća fiksnu mjesečnu naknadu (**paket.mjNaknada**) pa i kada korisnik ne inicira nijedan poziv u nekom mjesecu.

Bilježe se pojedinačni pozivi (**poziv**) u kojima korisnici sudjeluju. Jedan od dva broja (**poziv.broj1**, **poziv.broj2**) koji sudjeluju u pozivu je uvijek MTO korisnik, a ponekad i oba. Poziv je uspješno uspostavljen ako traje (**poziv.trajePoziv**) više od 0 sekundi. Pored fiksne mjesečne naknade korisniku MTO se naplaćuje svaki **uspješan** poziv kojeg je inicirao (**poziv.broj1**), a cijena poziva (**cjenik.cijenaMin**) se iskazuje za minutu razgovora ovisno o paketu i tipu poziva. Radi jednostavnosti smatrati da korisnik ne mijenja ugovoreni paket. Atributi **poziv.trajePoziv** i **poziv.trajeZvano** su tipa INTERVAL.

Relacija **finTran** sadrži dva tipa financijskih transakcija korisnika (**finTran.tipTran**): zapise koji se odnose na zaduženja korisnika ili na njihove uplate.

Na slici **nisu** prikazane sve ntorke sadržane u relacijama.

**Slika 1.**

korisnik						paket			cjenik		
sifKor	...	rizican	broj	oznPaket	login	oznPaket	nazPaket	mjNaknada	OznPaket	tipPoziv	cijenaMin
474	...	ne	385916761802	S01	johnm	S01	Standardni	40,00	S01	fiksni	2,50
877	...	da	385987348771	S01	anas	Sr01	Srebrni	80,00	S01	mobilni	1,34
...	...	...	...	...	...	ZL01	Zlatni	100,00	S01	premium	57,00
...	...	...	...	...	...	...	...	...	...	...	...

poziv						finTran				
broj1	vrijeme	broj2	tip Poziv	traje Poziv	traje Zvano	sifTran	sifKor	datum	tipTran	iznos
385916761802	12.05.2018 10:35:12	4465772831	mobilni	00:03:21	00:00:35	23411	218	15.04.2018	zaduženje	100,00
44800288931	05.05.2018 09:11:55	385987348771	premium	00:00:00	00:00:02	23412	474	15.04.2018	zaduženje	52,66
385987348771	05.05.2018 09:15:02	44800288931	premium	00:00:07	00:00:03	23413	877	25.04.2018	uplata	58,00
...	...	...	...	...	...	...	...	...	...	...

U zadacima 1-6 i 9 se podrazumijeva korištenje SUBP-a PostgreSQL.

- (4 boda)** Napisati jednu SQL naredbu kojom će se ispisati podatci o tzv. Wangiri prevari u kojoj su oštećeni korisnici MTO. Wangiri prevaru čine dva poziva. U prvom pozivu broj-prevarant poziva MTO korisnika (oštećeni), s tim da se poziv ne uspostavlja uspješno (odrediti pomoću atributa **poziv.trajePoziv**) i zvonjava traje kraće od 5 sekundi. **Nakon toga** (bilo kada ali nužno nakon prvog poziva) korisnik koji je bio pozvan (oštećeni) uzvraća poziv broju-prevarantu. Taj drugi poziv se uspješno uspostavlja uz naplatu po cijeni većoj od 30 kn po minuti poziva. Potrebno je ispisati broj MTO korisnika (oštećenog), broj prevaranta te **vrijeme** početka prvog i drugog poziva. Pretpostaviti da poziv od strane broja-prevaranta nikad nije upućen prema istom MTO broju više od jedan put. Zapise poredati prema vremenu početka poziva broja-prevaranta uzlazno.
- (5 bodova)** Napisati jednu SQL naredbu kojom se izračunavaju mjesečni računi korisnika i upisuju u relaciju **finTran** (**tipTran** = 'zaduženje') za sve korisnike MTO za prošli kalendarski mjesec. Naredba mora, bez obzira na datum izvođenja, uvijek računati i upisivati zaduženja za prošli mjesec. Kao datum generiranja zaduženja staviti datum izvođenja upita. Iznos mjesečnog zaduženja korisnika računa se kao zbroj fiksne mjesečne naknade za korištenje ugovorenog paketa i troškova **uspješno** realiziranih odlaznih poziva korisnika. Trošak jednog poziva računa se kao umnožak trajanja poziva i pripadne cijene (trajanje poziva u minutama: (minute + (sekunde / 60))). Budući da se fiksna mjesečna naknada za korištenje paketa naplaćuje uvijek, zaduženja treba generirati svim korisnicima pa i onima koji u prošlom mjesecu nisu uspješno realizirali niti jedan poziv. Atribut **finTran.sifTran** je tipa SERIAL.
- (3 boda)** Korisnici MTO pregledavaju svoje podatke na *MojTelekom* portalu nakon registracije korisničkog imena (**korisnik.login**). Administrator sustava registriranim korisnicima treba omogućiti pregledavanje podataka o vlastitim odlaznim i dolaznim pozivima. U bazi podataka **telekom** administrator je obavio sljedeće dvije SQL naredbe:  

```
REVOKE CONNECT ON DATABASE telekom FROM PUBLIC;
```

```
REVOKE ALL ON SCHEMA public FROM PUBLIC;
```

Napisati SQL naredbe kojima će administrator
  - kreirati ulogu *korPortal* te omogućiti ulozi *korPortal* pregledavanje podataka o odlaznim i dolaznim pozivima korisnika koji trenutno koristi ulogu *korPortal*. Jedina schema koja postoji u bazi podataka je PUBLIC i sve potrebne objekte treba kreirati u schemi PUBLIC.
  - korisniku *cpratt* dodijeliti ovlasti za uspostavu korisničke sjednice, spajanje na bazu podataka **telekom** te ovlasti za rad s ulogom *korPortal*.
- (5 bodova)** Napisati niz SQL naredbi za kreiranje svih potrebnih objekata kojima će se pri unosu ntorke u relaciju **finTran** održavati ispravnom vrijednost atributa **rizican** u relaciji **korisnik**. Korisnik je rizičan (**korisnik.rizican**='da') ako prilikom unosa novog zaduženja još uvijek nije podmirio ono prethodno, to jest, zadnja financijska transakcija prije unosa novog zaduženja nije uplata, nego zaduženje. Korisnik prestaje biti rizičan (**korisnik.rizican**='ne') kod unosa bilo kakve uplate.  
Dodatno, pri pokušaju unosa zapisa s iznosom  $\leq 0$ , korisniku treba dojaviti pogrešku „Pogreška: iznos transakcije nije ispravan“.

5. (4 boda) Prikazan je sadržaj relacije **paket** u trenutku započinjanja naredbe {1}. Izuzev transakcija A i B niti jedna druga transakcija ne mijenja sadržaj relacije **paket**.  
Odredite rezultate obavljanja naredbi {4}, {6}, {7} i {9}.

**paket**

xmin	xmax	ozn Paket	naz paket	mj naknada
50	0	S01	Standardni	40.00
60	0	ZL01	Zlatni	100.00

**Napomena:** Za skraćeni zapis gornje dvije ntorkke možete (ali ne morate) koristiti odgovarajuću oznaku (\* ili \*\*), ako se u rezultatu neke od SELECT naredbi pojavljuje ntorka s identičnim vrijednostima atributa. U protivnom je potrebno napisati vrijednosti svih atributa ntorkke.

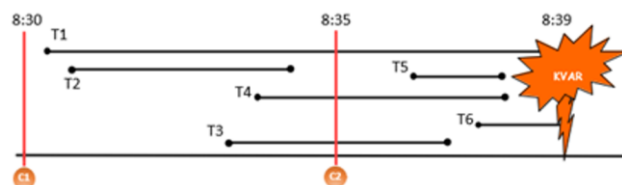
	Transakcija A $T_{id}=100$		Transakcija B $T_{id}=101$
{1}	BEGIN TRANSACTION;	{2}	BEGIN TRANSACTION; SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
{3}	INSERT INTO paket VALUES ( 'Sr01', 'Srebrni', 80.00);	{4}	SELECT xmin, xmax, * FROM paket;
		{5}	UPDATE paket SET mjNaknada = 110.00 WHERE oznPaket = 'ZL01';
{7}	SELECT xmin, xmax, * FROM paket;	{6}	SELECT xmin, xmax, * FROM paket;
{9}	SELECT xmin, xmax, * FROM paket;	{8}	COMMIT TRANSACTION;
{10}	COMMIT TRANSACTION;		

6. (5 bodova) Sve relacije sa slike 1 su kreirane u bazi podataka. Pri tom su definirani primarni i strani ključevi.

Izvodi se upit:	Optimizator raspolaže sljedećim podacima
SELECT * FROM poziv, korisnik, paket WHERE poziv.broj1 = korisnik.broj AND korisnik.oznPaket = paket.oznPaket AND poziv.trajePoziv = '5 minutes':INTERVAL AND mjNaknada < 100.00	N(poziv) = 6 000 000    V(broj, korisnik) = 250000 N(korisnik) = 250000    V(oznPaket, korisnik) = 40 N(paket) = 300    V(broj1, poziv) = 500 000 V(broj2, poziv) = 1 400 000 V(trajePoziv, poziv) = 1500

Nacrtati **konačno** stablo upita nakon provedene heurističke optimizacije. Redoslijed spajanja relacija odrediti temeljem procjene broja n-torki u rezultatima spajanja. Navesti sve izraze prema kojima je obavljena procjena broja n-torki u međurezultatima. U stablu upita naznačiti očekivani broj n-torki.

7. (3 boda) U SUBP-u je došlo do kvara na razini sustava 26.6.2018. u 8:39 sati. Dnevnik izmjena je uspješno očuvan. Slika prikazuje aktivne transakcije i kontrolne točke toga dana između 8:30 i 8:39.

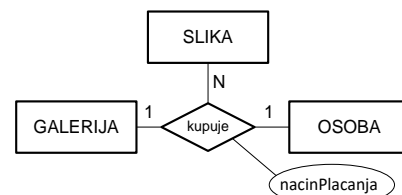


- a) Sukladno stanju na slici desno navedite sadržaj liste za poništavanje, odnosno ponovno obavljanje transakcija.
- b) Objasnite kako kontrolne točke pomažu pri kreiranju liste za poništavanje u odnosu na (hipotetsku) situaciju u kojoj se kontrolne točke ne koriste. Koje informacije iz zapisa kontrolne točke se pri tom koriste.

8. (7 bodova) Oblikujte ER-model segmenta baze podataka za praćenje podataka o filmovima, osobama uključenima u produkciju filma i projekcijama filmova u kinima. Za filmove se evidentiraju šifra, naslov, trajanje te žanrovi kojima pripadaju. Jedan film može pripadati više žanrova. Za žanrove se evidentiraju šifra i naziv. Osoba u filmu može imati više uloga (npr. istovremeno je i glavni glumac i producent). U jednom filmu istu ulogu može imati više osoba, jedna osoba istu ulogu može imati u različitim filmovima. Za osobe se evidentiraju šifra, ime, prezime, e-mail adresa te URL njihove osobne stranice, dok se za ulogu evidentiraju šifra i naziv. Za kino se evidentira šifra i naziv te dvorane u kojima se projiciraju filmovi. Dvorane u kinima su označene brojevima (1, 2, 3...) i za svaku se evidentira kapacitet. U različitim kinima mogu postojati dvorane s istim brojem. Za svaku projekciju u nekoj dvorani evidentira se jedinstvena šifra i vremenska oznaka početka projekcije.

Nacrtajte ER model baze podataka. Navedite sheme entiteta i sheme veza (označiti ključeve). Svaki entitet (osim slabih entiteta) opisati **isključivo vlastitim atributima**. Nužno je da sve sheme zadovoljavaju **3NF**.

9. (4 boda) Zadani ER-model treba predstaviti relacijskim modelom podataka. Uz pretpostavku da su relacije **osoba**, **slika** i **galerija** već kreirane, te da su definirana sva potrebna integritetska ograničenja napisati SQL naredbu(e) za kreiranje relacije kojom će u relacijskom modelu podataka biti predstavljena veza **kupuje** s ugrađenim definicijama primarnog, alternativnih i stranih ključeva. Osigurati pri tom da atribut **nacinPlacanja** može biti jedna od vrijednosti: 'gotovina', 'kartica'. Smatrati da su ključevi u kreiranim relacijama **osoba**, **slika** i **galerija** tipa INTEGER.



OSOBA = sifOsoba, ime, prezime, datRod  
SLIKA = sifSlika, naslovSlika  
GALERIJA = sifGalerija, nazGalerija

## Rješenja:

### 1.

```
SELECT Poziv1.broj2 osteceni, Poziv1.broj1 prevarant, Poziv1.vrijeme vrijeme1P, Poziv2.vrijeme
vrijeme2P
FROM poziv Poziv1
JOIN poziv Poziv2
  ON Poziv1.broj2 = Poziv2.broj1
  AND Poziv1.broj1 = Poziv2.broj2
  AND Poziv2.vrijeme > Poziv1.vrijeme
JOIN korisnik ON korisnik.broj = Poziv1.broj2
JOIN cjenik ON cjenik.oznPaket = korisnik.oznPaket
  AND cjenik.tipPoziv = Poziv2.tipPoziv
WHERE Poziv1.trajePoziv = '0 seconds'::INTERVAL
  AND Poziv1.trajeZvono < '5 seconds'::INTERVAL
  AND Poziv2.trajePoziv > '0 seconds'::INTERVAL
  AND cjenik.cijenaMin > 30
ORDER BY Poziv1.vrijeme;
```

### 2.

```
INSERT INTO finTran (sifKor, datum, tipTran, iznos)
SELECT sifKor, CURRENT_DATE, 'zaduženje',
  mjNaknada +
  CASE WHEN COUNT(DISTINCT poziv.broj2)>0 THEN
    SUM((EXTRACT(MINUTE FROM poziv.trajePoziv) +
      EXTRACT(SECOND FROM poziv.trajePoziv)/60) * cjenik.cijenaMin)
  ELSE 0
  END
FROM korisnik
  NATURAL JOIN paket
  LEFT JOIN poziv
    ON poziv.broj1 = korisnik.broj
    AND poziv.trajePoziv > '0 seconds'::INTERVAL
    AND EXTRACT(MONTH FROM poziv.vrijeme) =
      EXTRACT(MONTH FROM (CURRENT_DATE - '1 MONTH'::INTERVAL))
    AND EXTRACT(YEAR FROM poziv.vrijeme) =
      EXTRACT(YEAR FROM (CURRENT_DATE - '1 MONTH'::INTERVAL))
  LEFT JOIN cjenik
    ON cjenik.oznPaket = korisnik.oznPaket
    AND cjenik.tipPoziv = poziv.tipPoziv
GROUP BY sifKor, mjNaknada;
```

### 3.

```
a)
CREATE VIEW vPoziviKorisnika AS
  SELECT poziv.*
  FROM poziv
  JOIN korisnik
    ON korisnik.broj=poziv.broj1 OR korisnik.broj=poziv.broj2
  WHERE korisnik.login=SESSION_USER;
CREATE ROLE korPortal;
--GRANT CONNECT ON DATABASE telekom TO korPortal; ne treba jer nikada neće uspostavljati
korisničku sjednicu niti doći u priliku da se spaja na bazu podataka
GRANT USAGE ON SCHEMA public TO korPortal;
GRANT SELECT ON vPoziviKorisnika TO korPortal;

b)
CREATE LOGIN cpratt WITH PASSWORD 'JurassicWorld';
GRANT CONNECT ON DATABASE telekom TO cpratt;
GRANT korPortal TO cpratt;
--GRANT USAGE ON SCHEMA public TO cpratt; ne treba mu jer će dobiti preko korPortal
```

4.

```
CREATE FUNCTION azurirajRizicnost() RETURNS TRIGGER AS $$
DECLARE
    p_i INTEGER;
    p_riz CHAR(2);
BEGIN
    IF NEW.iznos<=0 THEN
        RAISE EXCEPTION 'Pogreška: iznos transakcije nije ispravan';
    END IF;
    p_riz='ne';
    IF NEW.tipTran='zaduženje' THEN
        SELECT COUNT(*) INTO p_i
        FROM finTran
        WHERE finTran.sifKor = NEW.sifKor
            AND finTran.tipTran='zaduženje'
            AND finTran.datum = (SELECT MAX(datum)
                                FROM finTran
                                WHERE sifKor = NEW.sifKor);

        IF p_i>0 THEN
            p_riz='da';
        END IF;
    END IF;
    UPDATE korisnik
    SET rizican = p_riz
    WHERE sifKor = NEW.sifKor;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER insFinTran
BEFORE INSERT ON finTran
FOR EACH ROW
EXECUTE PROCEDURE azurirajRizicnost();
```

5.

A					B				
{7}					{4}				
<u>xmin</u>	xmax	ozn	naz	mj	<u>xmin</u>	xmax	ozn	naz	mj
		Paket	paket	naknada			Paket	paket	naknada
*					*				
60	101	ZL01	Zlatni	100.00					
100	0	Sr01	Srebrni	80.00					
{9}					{6}				
<u>xmin</u>	xmax	ozn	naz	mj	<u>xmin</u>	xmax	ozn	naz	mj
		Paket	paket	naknada			Paket	paket	naknada
*					*				
101	0	ZL01	Zlatni	110.00	101	0	ZL01	Zlatni	110.00
100	0	Sr01	Srebrni	80.00					

## 6.

Procjena veličine međurezultata nakon potiskivanja selekcije

$\text{paket1} = \sigma_{\text{mjNaknada} < 2.0} (\text{paket})$

$N(\text{cjenik1}) = N(\text{paket}) / 3 = 100$

$\text{poziv1} = \sigma_{\text{trajePoziv}='5 \text{ minutes}'} (\text{poziv1})$

$N(\text{poziv1}) = N(\text{poziv}) / V(\text{trajePoziv}, \text{poziv}) = 6\,000\,000 / 1500 = 4\,000$

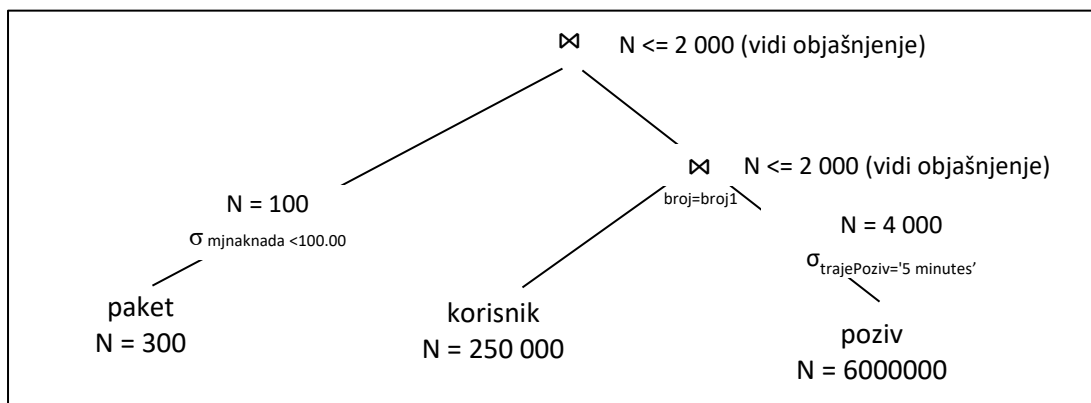
Procjena veličine međurezultata nakon spajanja

i)  $N(\text{paket1} \bowtie \text{korisnik}) \leq N(\text{korisnik}) = 250\,000$

ii)  $N(\text{poziv1} \bowtie \text{korisnik}) = N(\text{poziv1}) * N(\text{korisnik}) / (\text{MAX}(V(\text{broj1}, \text{poziv1}), V(\text{broj}, \text{korisnik})))$   
 $= 4000 * 250\,000 / (\text{MAX}(500\,000, 250\,000)) = 4000 * 250\,000 / 500\,000 = 2\,000$

iii)  $N(\text{poziv1} \bowtie \text{paket1}) \leq N(\text{poziv1}) * N(\text{paket1}) = 4000 * 100 = 400\,000$

Redoslijed spajanja: 1. korisnik  $\bowtie$  poziv1 pa potom međurezultat s paket1



## 7.

a)

lista za ponovno obavljanje	T3, T4, T5	završile su nakon kontrolne točke C2, ali prije trenutka kvara - njihovi efekti moraju postojati u bazi. Izvjesno je da su sve promjene nad bazom podataka pohranjene u posljednjoj kontrolnoj točki (C2). Za izmjene koje su nastale nakon kontrolne točke (C2) nismo sigurni jesu li pohranjeni u bazu podataka.
lista za poništavanje	T1, T6	ponišavaju se jer u trenutku kvara još nisu završile

b)

Transakcije koje, u trenutku kvara, nisu bile završene moraju biti poništene.

Potvrđene transakcije čije izmjene možda nisu zapisane u bazu podataka moraju biti ponovno obavljene.

Kad ne bi postojao koncept kontrolne točke, da bi se doznalo koje transakcije treba poništiti (imaju početak (BEGIN), ali ne i završetak (COMMIT ili ROLLBACK)) trebalo bi pretraživati dnevnik izmjena od početka.

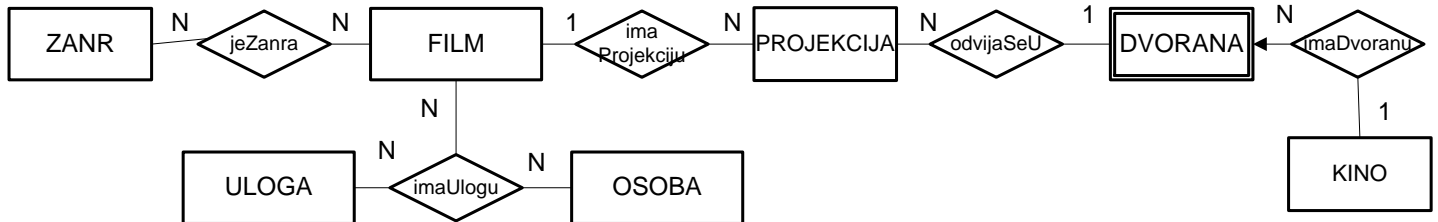
To može dugo trajati i zbog toga se koriste kontrolne točke, kako **bi se proces pretraživanja dnevnika izmjena skratio**.

Budući da zapis kontrolne točke sadrži:

- listu svih aktivnih transakcija
- za svaku transakciju - adresu najnovijeg zapisa u datoteci dnevnika

jednostavno se može napraviti početna lista transakcija koje treba poništiti, dok je početna lista za ponovno obavljanje prazna. Liste mijenjaju svoj sadržaj na temelju podataka iz dnevnika, koji se zapisani nakon kontrolne točke.

8.



ER model – sheme entiteta

ZANR	FILM	OSOBA	ULOGA	KINO	DVORANA	PROJEKCIJA
<u>sifZanr</u>	<u>sifFilm</u>	<u>sifOsoba</u>	<u>sifUloga</u>	<u>sifKino</u>	<u>sifKino</u>	<u>sifProjekcija</u>
nazZanr	naslov	e-mail	nazUloga	nazKino	rbr	pocetak
	trajanje	URL			kapacitet	
		ime				
		prezime				

ER model – sheme veza

jeZanra	imaUlogu	imaProjekciju	odvijaSeU	imaDvoranu
<u>sifZanr</u>	<u>sifOsoba</u>	<u>sifProjekcija</u>	<u>sifProjekcija</u>	<u>sifKino</u>
<u>sifFilm</u>	<u>sifUloga</u>	sifFilm	sifKino	<u>rbr</u>
	<u>sifFilm</u>		rbr	

9.

```

CREATE TABLE kupuje (
    sifSlika INT REFERENCES slika (sifSlika),
    sifOsoba INT REFERENCES osoba (sifOsoba),
    sifGalerija INT NOT NULL REFERENCES galerija(sifGalerija),
    nacinPlacanja VARCHAR(15) NOT NULL CHECK (nacinPlacanja IN 'gotovina', 'kartica'),
    PRIMARY KEY(sifSlika, sifOsoba),
    UNIQUE (sifSlika, sifGalerija));
    
```