

Baze podataka - 2. jesenski rok - 6.9.2020.

1. (4 boda)

Uz pomoć jedne SQL naredbe izvedite sljedeće: za sve timove mlađe od dvije godine u trenutku izvođenja upita a čiji se naziv sastoji od barem tri riječi, ispisati naziv tima i broj igrača u timu. Ispis poredati po nazivu tima uzlazno.

Primjer redaka ispisa (podaci ne moraju odgovarati stvarnima):

| NazivTim | BrojIgraca |
|------------------------------|------------|
| Au Revior Arm Pits | 5 |
| Hello-darkness-my-old-friend | 7 |

NAPOMENA: Riječi u nazivima timova su odvojene crticama ili razmacima, pri čemu svaki naziv tima koristi isključivo jedan od tih razdvojnika.

Rješenje:

```
SELECT nazivTim, COUNT(igracTim.idIgrac) AS brojIgraca
FROM tim JOIN igracTim ON tim.idTim = igracTim.idTim
WHERE CURRENT_TIMESTAMP - datVrOsnutak < '2 YEARS'::INTERVAL
      AND (nazivTim LIKE '% % %' OR nazivTim LIKE '%-%-%')
GROUP BY tim.idtim, tim.nazivTim
ORDER BY nazivtim;
```

2. (4 boda)

Uz pomoć jedne SQL naredbe izvedite sljedeće: za sve igrače koji su sudjelovali u barem jednoj borbi ali nikada u onoj koja je trajala dulje od 222 minute (stvarno trajanje borbe) ispisati prezime ime i datum rođenja. Ispis poredati abecedno po prezimenu a potom po imenu.

Primjer retka ispisa (podaci ne moraju odgovarati stvarnima):

| Prezime | Ime | DatRodjenja |
|---------|------|-------------|
| Perić | Pero | 30.3.1977 |

Rješenje:

```
SELECT DISTINCT igrac.prezime, igrac.ime, igrac.datrodjenja
FROM
    igrac NATURAL JOIN igractim
    JOIN borba ON (idTim = idTim1 OR idTim = idTim2)
WHERE igrac.idIgrac NOT IN
    (SELECT igracTim.idigrac
     FROM igractim JOIN borba ON (idTim = idTim1 OR idTim = idTim2)
     WHERE
         zavrsetakBorba - pocetakTermin > '222 minutes'::INTERVAL)
ORDER BY prezime ASC, ime ASC;
```

3. (6 bodova)

Sljedeći zadatak riješite **jednim** SQL upitom.

Za arene u kojima se više borbi odigralo (završilo) prije 16h (uključivo) nego nakon 16h, ispišite identifikator i naziv te arene. Osim identifikatora i naziva arene ispišite i koliki je postotak borbi odigranih (završenih) prije 16h u toj areni u odnosu na ukupan broj borbi odigranih u toj areni.

Primjer ispisa:

| Idarena | Nazivarena | Udio |
|---------|-----------------|-------------------------|
| 8 | Higham Arena | 53.88601036269430051800 |
| 11 | Lake View Arena | 52.96167247386759581900 |
| ... | ... | ... |

Rješenje:

```
SELECT arena.idarena, nazivarena, (CAST(count(*) AS decimal)/
                                   (SELECT CAST(count(*) AS decimal)
                                    FROM borba AS bo NATURAL JOIN igra AS ig
                                    WHERE ig.idarena = arena.idarena))*100 AS udio
FROM borba NATURAL JOIN igra
      NATURAL JOIN arena
WHERE borba.zavrsetakBorba::TIME <= '16:00:00'::TIME AND zavrsetakBorba IS NOT NULL
GROUP BY arena.idArena, arena.nazivarena
HAVING COUNT(*) > (SELECT COUNT(*)
                   FROM borba AS b NATURAL JOIN igra AS i
                   WHERE i.idarena = arena.idArena
                   AND b.zavrsetakBorba::TIME > '16:00:00'::TIME AND
b.zavrsetakBorba IS NOT NULL)
```

4. (5 bodova)

Napisati niz SQL naredbi za kreiranje svih potrebnih objekata kojima će se pri **unosu** n-torke u relaciju **borba** spriječiti borbe timova koji imaju zajedničke igrače.

Pri narušavanju opisanog ograničenja korisniku javiti sljedeću grešku: **'Pogreška: timovi imaju zajedničke igrače! Broj zajedničkih igrača je: <broj>.'**

Umjesto <broj> potrebno je ispisati broj igrača koji se nalaze u oba tima.

U svim ostalim slučajevima n-torka mora biti uspješno upisana u relaciju **borba**.

Rješenje:

```
CREATE or replace FUNCTION igracObaTima() RETURNS TRIGGER AS
$$
DECLARE
    brZajednickih SMALLINT;
BEGIN
    SELECT COUNT(*) INTO brZajednickih
    FROM igracTim igracTim1, igracTim igracTim2
    WHERE igracTim1.idTim = NEW.idTim1
        AND igracTim2.idTim = NEW.idTim2
        AND igracTim1.idIgrac = igracTim2.idIgrac;

    IF (brZajednickih > 0) THEN
        RAISE EXCEPTION
        'Pogreška: timovi imaju zajedničke igrače! Broj zajedničkih igrača je: %.', brZajednickih;
    END IF;

RETURN NEW;

END;
$$ language plpgsql;

CREATE TRIGGER insertBorba
BEFORE INSERT ON borba
FOR EACH ROW EXECUTE FUNCTION igracObaTima();
```

5. (6 bodova) Pretpostavite da je relacijska shema relacije *igrac* izmijenjena dodavanjem atributa *korIme* čija je vrijednost jednaka korisničkom imenu korisnika koji uspostavlja sjednicu:

| idIgrac | ime | prezime | datRodjenja | korIme |
|---------|------|---------|-------------|---------|
| 1 | John | Morris | 19.11.1982 | jmorris |
| ... | ... | ... | ... | ... |

Administrator baze podataka je nakon kreiranja baze podataka i relacija u shemi *public* obavio sljedeće SQL naredbe:

```
REVOKE CONNECT ON DATABASE tagfight FROM public;  
REVOKE ALL ON SCHEMA public FROM public;
```

Napisati SQL naredbe kojima će administrator sustava:

- Kreirati ulogu *igrac* te korisniku (igraču) kojem se dodijeli ta uloga omogućiti pregled i izmjenu podataka o završenim borbama u kojima je taj igrač sudjelovao.
- Već kreiranom korisniku *jmorris* dodijeliti ovlasti za spajanje na bazu podataka te ovlasti za rad s ulogom *igrac*.
- Korisniku *jmorris* dodijeliti ovlasti za stvaranje vlastitih shema.

Rješenje:

```
a)  
CREATE VIEW borbeTimoviJaIgrac AS  
SELECT *  
FROM borba  
WHERE EXISTS (SELECT *  
FROM igrac NATURAL JOIN igracTim  
WHERE (borba.idTim1 = igracTim.idTim  
OR borba.idTim2 = igracTim.idTim)  
AND korIme = SESSION_USER  
AND zavrsetakBorba IS NOT NULL  
) -- ne može CURRENT_USER  
WITH CHECK OPTION;  
  
CREATE ROLE igrac;  
--GRANT CONNECT ON DATABASE tagfight TO igrac; ne treba jer  
nikada neće uspostavljati korisničku sjednicu niti doći u priliku  
da se spaja na bazu podataka  
GRANT USAGE ON SCHEMA public TO igrac;  
GRANT SELECT, UPDATE ON public.borbeTimovaJaIgrac TO igrac;  
  
b)  
GRANT CONNECT ON DATABASE tagfight TO jmorris;  
GRANT igrac TO jmorris;  
--GRANT USAGE ON SCHEMA public TO jmorris; ne treba mu jer će dobiti preko  
igrac  
  
c)  
GRANT CREATE ON DATABASE tagfight to jmorris;
```

6. (3 boda)

Koristeći SQL naredbe, kreirajte najmanji mogući broj indeksa za učinkovito obavljanje dolje navedenih upita. Osim indeksa koje PostgreSQL kreira automatski, nisu kreirani nikakvi dodatni indeksi. Indeksima treba dodijeliti imena i1, i2,...,in.

1. SELECT * FROM tim WHERE datVrOsnutak :: DATE BETWEEN '01.01.2017' AND '31.12.2017' ;
2. SELECT * FROM tim ORDER BY idTim ASC, nazivTim DESC;
3. SELECT * FROM tim WHERE nazivtim BETWEEN 'Au Revoir Arm Pits' and 'Deadher';
4. SELECT * FROM tim WHERE idOsnivac < 150;
5. SELECT * FROM tim WHERE nazivTim = 'Crane Crane Croo' OR idTim = 135;
6. SELECT * FROM tim ORDER by idOsnivac ASC;

Rješenje:

| | |
|---|-------|
| CREATE INDEX i1 ON tim (datVrOsnutak) | 1 |
| CREATE INDEX i2 ON tim (idTim, nazivTim DESC) | 2 i 5 |
| CREATE INDEX i3 on tim (nazivTim) | 3, 5 |
| CREATE INDEX i4 on tim (idOsnivac) | 4, 6 |

7. (5 bodova)

Zadana je relacijska shema SkolaNaTrecem gdje se evidentiraju predavanja za pojedine razrede, koja se emitiraju na Trećem programu HRT, predmeti na koje se odnose te njihovi izvođači. Predavanja se označavaju rednim brojem unutar predmeta i razreda (npr. 1. predavanje iz Matematike za drugi razred). Relacijska shema SkolaNaTrecem ima sljedeće atribute:

| | |
|----------------|---|
| oznRazred | oznaka razreda - od 1 do 4 - na koji se odnosi predavanje |
| sifPredmet | šifra predmeta - predmet Matematika ima istu šifru za sve razrede |
| nazPredmet | naziv predmeta |
| rbrPredavanje | redni broj predavanja iz nekog predmeta u nekom razredu |
| temaPredavanje | tema predavanja |
| datEmitiranja | datum emitiranja predavanja |
| sifNastavnik | šifra nastavnika koji izvodi predavanje |
| imeNastavnik | ime nastavnika |
| prezNastavnik | prezime nastavnika |
| sifSkola | šifra škole u kojoj je zaposlen nastavnik |
| nazSkola | naziv škole nastavnika |

Odrediti ključ relacijske sheme SkolaNaTrecem tako da ona bude u 1NF, a zatim postupno normalizirati relacijsku shemu na 2NF i 3NF.

U svakom koraku navesti funkcijske zavisnosti na temelju kojih se obavlja dekompozicija te označiti ključeve novonastalih relacijskih shema.

Navesti relacijske sheme od kojih se sastoji konačna shema baze podataka SKOLA_NA_TRECEM.

Rezultate upišite u prostor za odgovore ispod teksta zadatka

Rješenje:

1NF

SN3_1 = oznRazred, sifPredmet, rbrPredavanje, temaPredavanje, datEmitiranja, nazPredmet, sifNastavnik, imeNastavnik, prezNastavnik, sifSkola, nazSkola

2NF

PREDMET = sifPredmet, nazPredmet

SN3_2 = oznRazred, sifPredmet, rbrPredavanje, temaPredavanje, datEmitiranja, sifNastavnik, imeNastavnik, prezNastavnik, sifSkola, nazSkola

3NF

SN3_3 = sifNastavnik, imeNastavnik, prezNastavnik, sifSkola, nazSkola

PREDAVANJE = oznRazred, sifPredmet, rbrPredavanje, temaPredavanje, datEmitiranja, sifNastavnik

NASTAVNIK = sifNastavnik, imeNastavnik, prezNastavnik, sifSkola

SKOLA = sifSkola, nazSkola

SKOLA_NA_TRECEM = PREDMET, PREDAVANJE, NASTAVNIK, SKOLA

8. (5 bodova)

U PostgreSQL SUBP nad bazom podataka koja se koristi u ispit u aktivne su samo dvije sjednice: Sjednica 1 i Sjednica 2 u kojima se izvode dolje navedene naredbe, prema brojevima navedenim ispred svake naredbe.

Korisnik kojem je zbog zaključavanja dojavljena pogreška, privremeno obustavlja obavljanje daljnjih naredbi **ali ne prekida transakciju**, a drugi korisnik nastavlja s radom dok ne izvede sve svoje naredbe ili dok se i njemu ne dogodi pogreška zbog nemogućnosti postavljanja ključa. Ako se korisniku oslobodi ključ na kojeg čeka, on nastavlja s obavljanjem svojih naredbi.

Za naredbe {3} – {6}, napišite koja vrsta ključa na koji objekt se traži od SUBP, uspijeva li postavljanje ključa i koliko dugo će ključ biti postavljen. Ako se ključ ne traži ili ne uspije postaviti obrazložite zbog čega. Ako prilikom izvođenja naredbe dođe do pogreške, navedite u kojoj sjednici pri izvođenju koje naredbe će pogreška biti dojavljena te obrazložite uzrok pogreške.

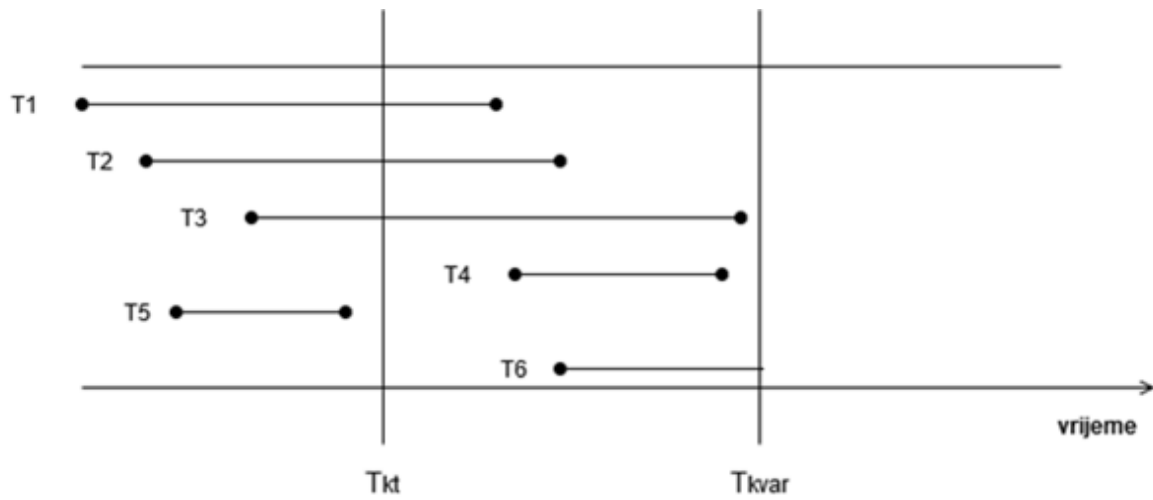
Objasnite što se s eventualno postavljenim ključevima događa u naredbama {7} i {8}.

NAPOMENA: svoje odgovore unesite u prostor za slobodni unos teksta ispod teksta zadatka.

| | Sjednica 1 | | Sjednica 2 |
|-----------------|--|-----|--|
| {1} | BEGIN TRANSACTION; | {2} | BEGIN TRANSACTION; |
| {3} | SELECT * FROM igrac WHERE idIgrac >= 20 FOR SHARE; | {4} | SELECT * FROM igrac WHERE idIgrac >= 30; |
| {5} | UPDATE igrac SET imeIgrac = 'Helena' WHERE idIgrac = 30; | {6} | UPDATE igra SET imeIgrac = 'Ava-May' WHERE idIgrac = 20; |
| {7} | COMMIT TRANSACTION; | {8} | COMMIT TRANSACTION; |
| Rješenje | | | |
| {3} | traži postavljanje ključeva za čitanje nad n-torkama sa idIgrac>=20 (zbog FOR SHARE). Ključevi se postavljaju i traju do kraja transakcije. | {4} | ne traži se postavljanje ključa. Ne postavljaju se ključevi nad ntorka ma s idIgrac >= 30 jer MVCC protokol pri obavljanju SELECT naredbe ne postavlja shared lock. On se postavlja samo ako je to izričito zatraženo (npr. pomoću FOR SHARE). |
| {5} | traži postavljanje ključa za pisanje nad n-torkom s idIgrac=30. Ključ se postavlja (u sjednici 1 nije na tu n-torku postavljen nikakav ključ) i traje do kraja transakcije. | {6} | traži postavljanje ključa za pisanje nad n-torkom s idIgrac=20. Ključ se NE postavlja jer je nad tom n-torkom u sjednici 1 naredbom {3} postavljen ključ za čitanje - transakcija čeka dok se taj ključ ne otpusti. |
| {7} | otpušta sve u sjednici 1 postavljene ključeve | | |
| | | {6} | Već zatraženi ključ za pisanje nad n-torkom s idIgrac=20 se postavlja i traje do kraja transakcije. |
| | | {8} | otpušta sve u sjednici S2 postavljene ključeve |

9. (3 boda)

U SUBP došlo je do ispada na razini računalnog sustava u trenutku T_{kvar} . Dnevnik izmjena je uspješno očuvan i njegov sadržaj prikazan je na slici niže:



- Prema dnevniku zapisa, navedite identifikatore transakcija
 - Koje je potrebno poništiti
 - Koje je potrebno ponovno obaviti
 - Za koje nije potrebno izvršiti nikakvu dodatnu akciju
- Koja je prednost korištenja kontrolne točke u odnosu na pristup kada se kontrolne točke ne koriste? Kada ne bi koristili kontrolnu točku, bi li postojala razlika u broju transakcija (s gornje slike) koje je potrebno poništiti?
- Koje se informacije iz kontrolne točke koja se dogodila u T_{kt} (na slici) koriste pri kreiranju liste za poništavanje?

Rješenje:

a) 1. T6; 2. T1, T2, T3, T4; 3. T5

b) Bez korištenja kontrolne točke trebalo bi pretražiti dnevnik izmjena od početka kako bi se detektirale transakcije koje je potrebno poništiti.

Broj transakcija koje se poništavaju ostaje isti jer se poništavaju samo transakcije za koje ne postoji završetak (COMMIT / ROLLBACK).

c) Lista svih aktivnih transakcija (T1, T2, T3) i adresa najnovijeg zapisa u datoteci dnevnika za svaku od aktivnih transakcija.

10. (8 bodova)

Oblikovati ER model baze podataka za evidenciju posvajanja životinja u zoološkom vrtu. Evidentiraju se podatci o životinjama: šifra, ime, vrsta, datum rođenja, datum dolaska u zoo. Za vrstu se evidentira šifra i naziv (lavovi, vukovi, ...) te porodica kojoj pripada. Za porodicu se evidentira šifra i naziv (sisavci, ptice, ribe, vodozemci, gmazovi, beskralježnjaci, ...).

Svaka životinja može imati jednog ili više posvojitelja. Posvojitelj može biti osoba ili organizacija. Za posvojenje se evidentira životinja koja se posvaja, posvojitelj te razina posvojenja. Za razinu posvojenja evidentira se šifra i naziv (brončani, srebrni, zlatni, VIP). Za posvojitelja se evidentira OIB, mjesto stanovanja ili sjedišta, ulica i kućni broj. Mjesto je opisano šifrom mjesta, poštanskim brojem i nazivom mjesta. Za osobu se evidentira ime, prezime i datum rođenja, a za organizaciju naziv organizacije i vrsta organizacije. Za vrstu organizacije evidentira se šifra i naziv.

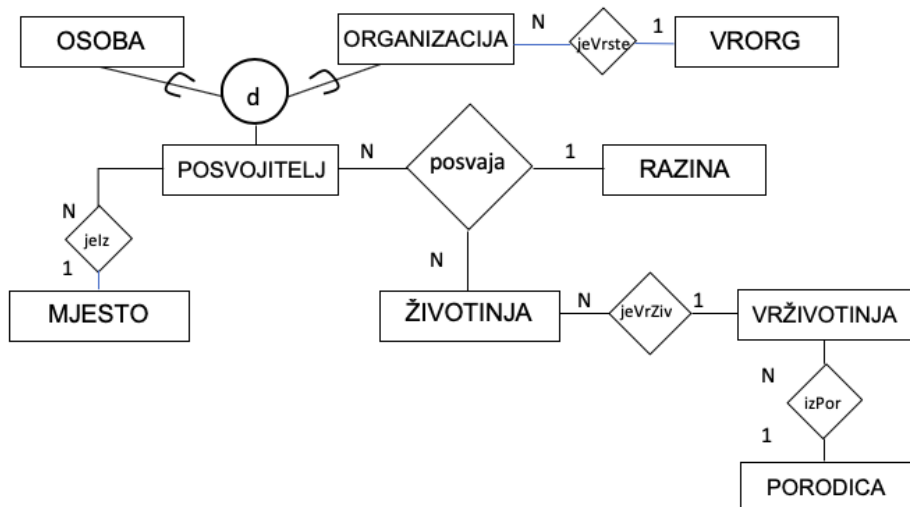
Posvojitelj za jednu životinju dogovara jednu razinu posvojenja. Životinju s istom razinom posvojenja može posvojiti više posvojitelja. Jedan posvojitelj može posvojiti više životinja s istom razinom posvojenja.

Potrebno je nacrtati ER dijagram (bez atributa) i označiti spojnosti entiteta u vezama. Sve entitete, osim slabih, opišite isključivo vlastitim atributima. Definirati relacijske sheme svih entiteta i svih veza, označiti ključeve i provjeriti zadovoljavaju li 3NF.

a) Nacrtajte ER model baze podataka na dobivenom papiru.

b) U prostor za slobodan unos teksta ispod teksta zadatka navedite sheme entiteta i sheme veza.

Rješenje:



| | |
|--|--|
| POSVOJITELJ = <u>oib</u> | VRŽIVOTINJA = <u>sifVrZivotinja</u> , nazVrZivotinja |
| MJESTO = <u>sifMjesto</u> , postBr, nazMjesto | PORODICA = <u>sifPorodica</u> , nazPorodica |
| OSOBA = <u>oib</u> , ime, prez, datRodOsoba | posvaja = <u>oib</u> , <u>sifZivotinja</u> , sifRazina |
| ORGANIZACIJA = <u>oib</u> , nazOrganizacija | jelz = <u>oib</u> , sifMjesto, ulica, kbr |
| RAZINA = <u>sifRazina</u> , nazRazina | jeVrZiv = <u>sifZivotinja</u> , sifVrZivotinja |
| VRORG = <u>sifVrOrg</u> , nazVrOrg | izPor = <u>sifVrZiv</u> , sifPorodica |
| ZIVOTINJA = <u>sifZivotinja</u> , imeZivotinja, datRod, datDolazak | jeVrste = <u>oib</u> , sifVrOrg |