

Završni ispit iz Baza podataka (35 bodova)

8. lipnja 2020.

U bazi podataka se evidentiraju podatci o borbama/dvoboju timova igrača (**igrac**, **tim**, **igracTim**) u nekoj od tematskih igara (**igra**). Igre se, zbog različitih zahtjeva za opremom, igraju u jednoj od arena (**arena**). Svaka igra može biti sadržana u više paketa (**paket**), a ovisno o predviđenom trajanju igre (**paket.trajanje**, tip INTERVAL), definirane su cijene paketa po igraču (**paket.cijena**). Osnivač tima (**tim.idOsnivac**) je uvijek član tima (**igracTim**) kojeg je osnovao, a igrač može biti i član i osnivač više timova. Relacija **borba** sadrži podatke o odigranim, planiranim i borbama koje su u tijeku: početak rezerviranog termina za borbu, vrijeme završetka borbe (NULL ako borba nije završila ili kraj nije poznat); igra i predviđeno trajanje (identificiraju paket), te timovi koji sudjeluju i pobjednik ako je poznat. Pobjednik je, naravno, jedan od timova koji su sudjelovali u dvoboju ili nepoznat (NULL) ako borba nije završila ili je iz nekog razloga neevidentiran.

U bazi podataka postoje mehanizmi koji osiguravaju konzistentnost podataka u tablici **borba** (igrač ne igra u isto vrijeme u dva tima, istovremeno u istoj areni se ne odvija više borbi i slično) i o tome ne treba brinuti.

Primarni ključevi relacija su podcrtani.

Slika 1 je ilustrativan prikaz mogućih n-torki u bazi podataka. Prikazane n-torke ne moraju odgovarati u potpunosti podatcima u bazi.

igrac

<u>idIgrac</u>	ime	prezime	datRodjenja
1	Christian	Reeves	30.03.1977
2	Lleyton	Pugh	28.06.1998
3	Tahmina	Preece	14.12.1965
...

tim

<u>idTim</u>	nazivTim	datVrOsnutak	idOsnivac
1	Titanmaya	3.12.2018 15:15:34	1
2	James Names	8.2.2018 12:23:00	3
3	Vileman	1.4.2017 16:00:00	3
4	A Dozendusti	2.4.2017 10:02:03	2
...

igracTim

<u>idIgrac</u>	<u>idTim</u>
1	1
2	1
3	2
4	2
...	...

arena

<u>idArena</u>	nazivArena
1	Primrose View Arena
2	Lakeside Arena
3	Tony's Arena
...	...

igra

<u>idIgra</u>	nazivIgra	idArena
1	Domination	1
2	Deathmatch	1
3	Battle Royale	2
4	Chernobyl Stalker	3
5	Zombie Biohazard	2
...

paket

<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>
1	00:30:00	100
1	01:00:00	150
1	02:00:00	250
2	01:00:00	170
2	02:00:00	230
3	02:00:00	200
4	01:00:00	160
5	01:30:00	180
...

borba

<u>idBorba</u>	pocetakTermin	zavrsetakBorba	idIgra	trajanje	idTim1	idTim2	idTimPobjeda
1	12.04.2019 14:00:00	12.04.2019 15:50:00	3	02:00:00	1	2	1
4	05.03.2020 18:00:00	05.03.2020 18:25:00	1	00:30:00	1	3	3
5	10.04.2020 14:00:00	10.04.2020 16:20:00	3	02:30:00	3	4	3
6	05.04.2020 12:00:00	05.04.2020 12:30:00	1	00:30:00	1	3	3
7	07.04.2020 12:00:00	07.04.2020 12:45:00	4	01:00:00	1	3	1
16	27.04.2020 17:00:00	NULL	3	02:00:00	2	4	NULL
...

Slika 1

1.1 (5 bodova)

Ispisati šifre i nazive timova koji su ukupno potrošili ponedjeljkom više nego što su ukupno potrošili bilo kojim drugim danom u tjednu. Pored šifre i naziva tima ispisati i ukupan iznos (zaokružen na dvije decimale) kojeg je pojedini tim potrošio na (završene) borbe odigrane ponedjeljkom.

Npr. tim naziva **Riot of the Purple Map** je ukupno do sada potrošio 19000.00 na igre nedjeljom, 43125.00 na igre ponedjeljkom, 23875.00 na igre srijedom i 25000.00 na igre četvrtkom pa je **Riot of the Purple Map** jedan od timova koji su ukupno potrošili ponedjeljkom više nego bilo kojim drugim danom u tjednu.

Kojeg dana u tjednu je tim potrošio neki iznos na borbu odredite prema datumu početka borbe.

Zapise poredati prema potrošenom iznosu silazno pa po nazivu tima uzlazno.

Primjer ispisa:

idTim	nazivTim	uklznos
121	Riot of the Purple Map	43125.00
105	Shoes for the Rappers	41000.00
118	Limp Map	35750.00

Rješenje:

```
SELECT tim.idTim, tim.nazivTim, ROUND(SUM(cijena), 2) ukIznos
FROM tim
NATURAL JOIN igracTim
JOIN borba ON (idTim1 = idTim OR idTim2 = idTim)
NATURAL JOIN paket
WHERE zavrsetakBorba IS NOT NULL
AND EXTRACT (DOW FROM pocetakTermin) = 1 --ponedjeljkom
GROUP BY tim.idTim, tim.nazivTim
HAVING SUM(cijena) >
    ALL (SELECT SUM(cijena)
          FROM tim tim2
          NATURAL JOIN igracTim
          JOIN borba ON (idTim1 = tim2.idTim OR idTim2 = tim2.idTim)
          NATURAL JOIN paket
          WHERE zavrsetakBorba IS NOT NULL
          AND tim2.idTim = tim.idTim
          AND EXTRACT(DOW FROM pocetakTermin) <> 1
          GROUP BY EXTRACT (DOW FROM pocetakTermin))
ORDER BY ROUND(SUM(cijena), 2) DESC, nazivTim;
```

1.2 (5 bodova)

Ispisati šifre i nazive igara na koje se ukupno potrošilo srijedom više nego se ukupno potrošilo ostalim danima u tjednu. Pored šifre i naziva igre ispisati i ukupan iznos (zaokružen na dvije decimale) koji je potrošen na (završene) borbe te igre odigrane srijedom.

Npr. na igru naziva **Action and Victory** se ukupno do sada potrošilo 103675.00 nedjeljom, 122200.00 ponedjeljkom, 154700.00 srijedom i 152100.00 četvrtkom pa je **Action and Victory** jedna od igara na koje se ukupno potrošilo srijedom više nego bilo kojim drugim danom u tjednu.

Kojeg dana u tjednu se na igru potrošio neki iznos na borbu odredite prema datumu početka borbe.

Zapise poredati prema potrošenom iznosu silazno pa po nazivu igre uzlazno.

Primjer ispisa:

idIgra	nazivIgra	ukIznos
16	Action and Victory	154700.00
17	Blockfight	84875.00
33	Raid of Campaigns	64500.00
...

Rješenje:

```
SELECT i1.idIgra, i1.nazivIgra, ROUND(SUM(cijena), 2) ukIznos
  FROM tim
  NATURAL JOIN igracTim
  JOIN borba ON (idTim1 = idTim OR idTim2 = idTim)
  NATURAL JOIN paket
  NATURAL JOIN igra AS i1
 WHERE zavrsetakBorba IS NOT NULL
  AND EXTRACT (DOW FROM pocetakTermin) = 3 --srijedom
 GROUP BY i1.idIgra, i1.nazivIgra
  HAVING SUM(cijena) >
        ALL (SELECT SUM(cijena)
              FROM tim
                NATURAL JOIN igracTim
                JOIN borba ON (idTim1 = idTim OR idTim2 = idTim)
                NATURAL JOIN paket
                NATURAL JOIN igra AS i2
              WHERE zavrsetakBorba IS NOT NULL
                AND i2.idIgra=i1.idIgra
                AND EXTRACT (DOW FROM pocetakTermin) <> 3 --osim srijedom
             GROUP BY EXTRACT (DOW FROM pocetakTermin))
 ORDER BY ROUND(SUM(cijena), 2) DESC, i1.nazivIgra;
```

1.3 (5 bodova)

Ispisati šifre i nazive arena na koje se ukupno potrošilo četvrtkom više nego se ukupno potrošilo ostalim danima u tjednu. Pored šifre i naziva arene ispisati i ukupan iznos (zaokružen na dvije decimale) koji je potrošen na (završene) borbe u toj areni odigrane četvrtkom.

Npr. na arenu naziva **Snowy Arena** se ukupno do sada potrošilo 358800.00 nedjeljom, 302325.00 ponedjeljkom, 342175.00 srijedom i 365600.00 četvrtkom pa je **Snowy Arena** jedna od igara na koje se ukupno potrošilo četvrtkom više nego bilo kojim drugim danom u tjednu.

Kojeg dana u tjednu se na arenu potrošio neki iznos na borbu odredite prema datumu početka borbe.

Zapise poredati prema potrošenom iznosu silazno pa po nazivu arene uzlazno.

Primjer ispisa:

idArena	nazivArena	ukIznos
4	Snowy Arena	365600.00
2	Lakeside Arena	321800.00
11	Lake View Arena	245925.00
...

Rješenje:

```
SELECT a1.idArena, a1.nazivArena, ROUND(SUM(cijena), 2) ukIznos
FROM tim
NATURAL JOIN igracTim
JOIN borba ON (idTim1 = idTim OR idTim2 = idTim)
NATURAL JOIN paket
NATURAL JOIN igra
NATURAL JOIN arena AS a1
WHERE zavrsetakBorba IS NOT NULL
AND EXTRACT (DOW FROM pocetakTermin) = 4 --četvrtkom
GROUP BY a1.idArena, a1.nazivArena
HAVING SUM(cijena) >
    ALL (SELECT SUM(cijena)
          FROM tim
          NATURAL JOIN igracTim
          JOIN borba ON (idTim1 = idTim OR idTim2 = idTim)
          NATURAL JOIN paket
          NATURAL JOIN igra
          NATURAL JOIN arena AS a2
          WHERE zavrsetakBorba IS NOT NULL
          AND a2.idArena=a1.idArena
          AND EXTRACT (DOW FROM pocetakTermin) <> 4 --osim četvrtkom
         GROUP BY EXTRACT (DOW FROM pocetakTermin))
ORDER BY ROUND(SUM(cijena), 2) DESC, a1.nazivArena;
```

1.4 (5 bodova)

Za svaki tim koji je sudjelovao u (završenim) borbama ispisati šifru, naziv i redni broj dana u tjednu (0-nedjelja, 1-ponedjeljak, itd.) u kojem je tim ukupno potrošio više nego što je ukupno potrošio bilo kojim drugim danom u tjednu. Pored navedenih podataka, ispisati i ukupan iznos (zaokružen na dvije decimale) koji je potrošen na (završene) borbe tog tima odigrane tog rednog broja dana u tjednu.

Npr. tim naziva **Saving Charlie** je ukupno do sada potrošio 31375.00 na igre nedjeljom, 17875.00 na igre ponedjeljkom, 46125.00 na igre srijedom i 26750.00 na igre četvrtkom. Tim **Saving Charlie** je potrošio više srijedom (redni broj dana u tjednu je 3) više nego bilo kojim drugim danom u tjednu.

Kojeg dana u tjednu je tim potrošio neki iznos na borbu odredite prema datumu početka borbe.

Zapise poredati prema potrošenom iznosu silazno pa po nazivu tima uzlazno.

Primjer ispisa:

idTim	nazivTim	danUTjednu	ukIznos
106	Saving Charlie	3	46125.00
115	Deaf Lions	0	45625.00
90	Les Bananas	0	44875.00
...

Rješenje:

```
SELECT tim.idTim, tim.nazivTim, EXTRACT(DOW FROM pocetakTermin) AS danUTjednu,
ROUND(SUM(cijena), 2) ukIznos
FROM tim
NATURAL JOIN igracTim
JOIN borba ON (idTim1 = idTim OR idTim2 = idTim)
NATURAL JOIN paket
WHERE zavrsetakBorba IS NOT NULL
GROUP BY tim.idTim, tim.nazivTim, EXTRACT(DOW FROM pocetakTermin)
HAVING SUM(cijena) >=
    ALL (SELECT SUM(cijena)
          FROM tim tim2
          NATURAL JOIN igracTim
          JOIN borba ON (idTim1 = tim2.idTim OR idTim2 = tim2.idTim)
          NATURAL JOIN paket
          WHERE zavrsetakBorba IS NOT NULL
          AND tim2.idTim = tim.idTim
          GROUP BY EXTRACT(DOW FROM pocetakTermin))
ORDER BY ROUND(SUM(cijena), 2) DESC, nazivTim;
```

2.1 (5 bodova)

U određenim arenama smiju igrati samo timovi koji su na dan početka borbe sastavljeni isključivo od punoljetnih igrača - starih 18 i više godina. Napisati niz SQL naredbi za kreiranje svih potrebnih objekata kojima će se pri unosu n-torke u relaciju **borba** spriječiti **upis** n-torke u areni s nazivom 'Apple Corner Arena' ako jedan od timova na dan početka borbe ima maloljetnih igrača.

Pri narušavanju opisanog ograničenja korisniku javiti sljedeću grešku: **'Pogreška: Jedan od timova (<tim1>, <tim2>) ima maloljetnih igrača'**.

Umjesto <tim1> i <tim2> potrebno je ispisati šifre timova koji sudjeluju u borbi.

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

Održavanje konzistentnosti navedenog pravila pri obavljanju ostalih operacija u relacijama **borba**, **igracTim** i **igrac** nije potrebno implementirati.

Rješenje:

```
CREATE FUNCTION chkStarost() RETURNS TRIGGER AS
$$
BEGIN
    IF((SELECT nazivArena
        FROM igra
        NATURAL JOIN arena
        WHERE idIgra = NEW.idIgra) = 'Apple Corner Arena') THEN
        IF((SELECT COUNT(*)
            FROM igrac
            NATURAL JOIN igracTim
            WHERE (idtim = NEW.idTim1 OR idtim = NEW.idTim2)
            AND (datRodjenja + '18 year'::INTERVAL
                > NEW.pocetakTermin::DATE)) >0 ) THEN
            RAISE EXCEPTION
                'Pogreška: Jedan od timova (% , %) ima maloljetnih
igrača',
                NEW.idtim1, NEW.idtim2;
        END IF;
    END IF;
    RETURN NEW;
END;
$$ language plpgsql;

CREATE TRIGGER insBorba
BEFORE INSERT ON borba
FOR EACH ROW EXECUTE FUNCTION chkStarost();
```

2.2 (5 bodova)

Arena 'Frosty Corner Arena' rezervirana je samo za najstarije/najvjernije timove, koji su osnovani barem 4 godine prije datuma početka borbe. Napisati niz SQL naredbi za kreiranje svih potrebnih objekata kojima će se pri unosu n-torke u relaciju **borba** spriječiti **upis** n-torke u areni s nazivom 'Frosty Corner Arena' ako jedan od timova u trenutku početka borbe nije osnovan prije barem 4 godine.

Pri narušavanju opisanog ograničenja korisniku javiti sljedeću grešku: **'Pogreška: Jedan od timova (<tim1>, <tim2>) nije osnovan prije barem 4 godine'**.

Umjesto <tim1> i <tim2> potrebno je ispisati šifre timova koji sudjeluju u borbi.

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

Održavanje konzistentnosti navedenog pravila pri obavljanju ostalih operacija u relacijama **borba** i **tim** nije potrebno implementirati.

Rješenje:

```
create function chkOsnutakTima() returns trigger as $$
begin
    if((select nazivarena
        from igra natural join arena
        where idigra=new.idigra)='Frosty Corner Arena') then
        if((select count(*)
            from tim
            where (idtim=new.idtim1 or idtim=new.idtim2)
            and datVrOsnutak + '4 year'::INTERVAL >
new.pocetakTermin::DATE) > 0) then
            raise exception 'Pogreška: Jedan od timova (% , %) nije
osnovan prije barem 4 godine',
                new.idtim1, new.idtim2;
        end if;
    end if;
    return new;
end;
$$ language plpgsql;

create trigger insBorba2
before insert on borba
for each row execute function chkOsnutakTima();
```

2.3 (5 bodova)

Arena 'Snowy Arena' ima opreme samo za 7 igrača. Napisati niz SQL naredbi za kreiranje svih potrebnih objekata kojima će se pri unosu n-torke u relaciju **borba** spriječiti **upis** n-torke u areni s nazivom 'Snowy Arena' ako oba tima ukupno imaju više od 7 igrača.

Pri narušavanju opisanog ograničenja korisniku javiti sljedeću grešku: **'Pogreška: Timovi (<tim1>, <tim2>) ukupno imaju više od 7 igrača'**.

Umjesto <tim1> i <tim2> potrebno je ispisati šifre timova koji sudjeluju u borbi.

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

Održavanje konzistentnosti navedenog pravila pri obavljanju ostalih operacija u relacijama **borba** i **igracTim** nije potrebno implementirati.

Rješenje:

```
create function chkUkIgraca() returns trigger as $$
begin
    if((select nazivarena
        from igra natural join arena
        where idigra=new.idigra)='Snowy Arena') then
        if((select count(*)
            from igractim
            where (idtim=new.idtim1 or idtim=new.idtim2)) > 7) then
            raise exception 'Pogreška: Timovi (%, %) ukupno imaju više
od 7 igrača', new.idtim1, new.idtim2;
        end if;
    end if;
    return new;
end;
$$ language plpgsql;

create trigger insBorba3
before insert on borba
for each row execute function chkUkIgraca();
```


2.4 (5 bodova)

Arena s nazivom 'Lakeside Arena' se koristi za igre koje zahtijevaju posebnu sigurnosnu opremu za maloljetne igrače. Međutim, u areni postoji sigurnosna oprema samo za 2 igrača. Napisati niz SQL naredbi za kreiranje svih potrebnih objekata kojima će se pri unosu n-torke u relaciju **borba** spriječiti **upis** n-torke za arenu s nazivom 'Lakeside Arena' ako među članovima oba tima na dan početka borbe ima više od 2 maloljetnika – člana mlađa od 18 godina na dan početka borbe.

Pri narušavanju opisanog ograničenja korisniku javiti sljedeću grešku: **'Pogreška: Timovi (<tim1>, <tim2>) zajedno imaju više od 2 maloljetna igrača.'**

Umjesto <tim1> i <tim2> potrebno je ispisati šifre timova koji sudjeluju u borbi.

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

Održavanje konzistentnosti navedenog pravila pri obavljanju ostalih operacija u relacijama **borba**, **igracTim** i **igrac** nije potrebno implementirati.

Rješenje:

```
CREATE FUNCTION chkUkBrMaloljetnika() RETURNS TRIGGER as $$
BEGIN
    IF((SELECT nazivArena
        FROM igra NATURAL JOIN arena
        WHERE idigra = NEW.idigra)='Lakeside Arena') THEN
        IF((SELECT COUNT(*)
            FROM igrac
            NATURAL JOIN igracTim
            WHERE (idtim = NEW.idTim1 OR idtim = NEW.idTim2)
            AND (datRodjenja + '18 year'::INTERVAL
                > NEW.pocetakTermin::DATE)) >2 ) THEN
            RAISE EXCEPTION 'Pogreška: Timovi (%, %) zajedno imaju više
od 2 maloljetna igrača.', new.idtim1, new.idtim2;
        END IF;
    END IF;
    RETURN NEW;
END;
$$ language plpgsql;

CREATE TRIGGER insBorba4
BEFORE INSERT ON borba
FOR EACH ROW EXECUTE FUNCTION chkUkBrMaloljetnika ();
```

3.1 (5 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	Jaydan	Atkins	28.11.2010	jatkins
...

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 korisniku *jatkins*, osnivaču nekih timova, dodijeliti ovlasti za:

- uspostavu korisničke sjednice (lozinku postaviti na *jatkinsPwd*),
- spajanje na bazu *tagfight*,
- pristup objektima sadržanima u shemi *public*,
- pregled podataka o svim timovima i svim igračima,
- pregled, dodavanje i brisanje članova tima čiji je on osnivač.

Kreirajte sve objekte potrebne za dodjelu opisanih ovlasti korisniku *jatkins*.

Rješenje:

```
CREATE USER jatkins WITH PASSWORD 'jatkinsPwd';  
GRANT CONNECT ON DATABASE tagfight TO jatkins;  
GRANT USAGE ON SCHEMA public TO jatkins;  
GRANT SELECT ON public.igrac TO jatkins;  
GRANT SELECT ON public.tim TO jatkins;
```

```
CREATE VIEW igracTimJaOsnivac AS  
SELECT *  
FROM igracTim  
WHERE idTim IN (SELECT idTim  
FROM tim  
JOIN igrac on igrac.idIgrac=idosnivac  
WHERE korIme = SESSION_USER)  
WITH CHECK OPTION;
```

```
GRANT SELECT, INSERT, DELETE ON public.igracTimJaOsnivac TO jatkins;
```

3.2 (5 bodova)

Pretpostavite da je relacijska shema relacije **igrac** izmijenjena dodavanjem atributa **korlme** čija je vrijednost jednaka korisničkom imenu korisnika koji uspostavlja sjednicu:

idIgrac	ime	prezime	datRodjenja	korlme
1	Jaydan	Atkins	28.11.2010	jatkins
...

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 korisniku **jatkins**, osnivaču nekih timova, dodijeliti ovlasti za:

- uspostavu korisničke sjednice (lozinku postaviti na **jatkinsPwd**),
- spajanje na bazu **tagfight**,
- pristup objektima sadržanima u shemi **public**,
- pregled podataka o svim timovima i svim igračima,
- pregled, dodavanje i brisanje vlastitog članstva u timovima.

Kreirajte sve objekte potrebne za dodjelu opisanih ovlasti korisniku **jatkins**.

Rješenje:

```
CREATE USER jatkins WITH PASSWORD 'jatkinsPwd';
```

```
GRANT CONNECT ON DATABASE tagfight TO jatkins;
```

```
GRANT USAGE ON SCHEMA public TO jatkins;
```

```
GRANT SELECT ON public.igrac TO jatkins;
```

```
GRANT SELECT ON public.tim TO jatkins;
```

```
CREATE VIEW MojiTimoviJaClan as
```

```
SELECT * FROM igracTim
```

```
WHERE EXISTS (SELECT idIgrac
```

```
FROM igrac
```

```
WHERE idIgrac=igracTim.idIgrac
```

```
AND korime = SESSION_USER)
```

```
WITH CHECK OPTION;
```

```
GRANT SELECT,INSERT,DELETE ON public.MojiTimoviJaClan to jatkins;
```

3.3 (5 bodova)

Pretpostavite da je relacijska shema relacije **igrac** izmijenjena dodavanjem atributa **korlme** čija je vrijednost jednaka korisničkom imenu korisnika koji uspostavlja sjednicu:

idIgrac	ime	prezime	datRodjenja	korlme
1	Jaydan	Atkins	28.11.2010	jatkins
...

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 korisniku **jatkins**, osnivaču nekih timova, dodijeliti ovlasti za:

- uspostavu korisničke sjednice (lozinku postaviti na **jatkinsPwd**),
- spajanje na bazu **tagfight**,
- pristup objektima sadržanima u shemi **public**,
- pregled podataka o svim timovima i svim igračima,
- pregled i izmjenu podataka o borbama koje nisu završile za timove kojima je on osnivač.

Kreirajte sve objekte potrebne za dodjelu opisanih ovlasti korisniku **jatkins**.

Rješenje:

```
CREATE USER jatkins WITH PASSWORD 'jatkinsPwd';
```

```
GRANT CONNECT ON DATABASE tagfight TO jatkins;
```

```
GRANT USAGE ON SCHEMA public TO jatkins;
```

```
GRANT SELECT ON public.igrac TO jatkins;
```

```
GRANT SELECT ON public.tim TO jatkins;
```

```
CREATE VIEW BorbeTimovaJaOsnivac as
```

```
SELECT * FROM borba
```

```
WHERE zavrsetakBorba IS NULL
```

```
AND EXISTS (SELECT *
```

```
FROM tim
```

```
JOIN igrac ON idigrac=idosnivac
```

```
WHERE (borba.idtim1=tim.idtim or borba.idtim2=tim.idtim)
```

```
AND korime = SESSION_USER)
```

```
WITH CHECK OPTION;
```

```
GRANT SELECT,UPDATE ON public.BorbeTimovaJaOsnivac to jatkins;
```

3.4 (5 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	Jaydan	Atkins	28.11.2010	jatkins
...

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 korisniku *jatkins*, osnivaču nekih timova, dodijeliti ovlasti za:

- uspostavu korisničke sjednice (lozinku postaviti na *jatkinsPwd*),
- spajanje na bazu *tagfight*,
- pristup objektima sadržanima u shemi *public*,
- pregled podataka o svim timovima i svim igračima,
- izmjenu i brisanje podataka o timovima kojima je on osnivač.

Kreirajte sve objekte potrebne za dodjelu opisanih ovlasti korisniku *jatkins*.

Rješenje:

```
CREATE USER jatkins WITH PASSWORD 'jatkinsPwd';
```

```
GRANT CONNECT ON DATABASE tagfight TO jatkins;
```

```
GRANT USAGE ON SCHEMA public TO jatkins;
```

```
GRANT SELECT ON public.igrac TO jatkins;
```

```
GRANT SELECT ON public.tim TO jatkins;
```

```
CREATE VIEW TimoviJaOsnivac as
```

```
SELECT *
```

```
FROM tim
```

```
WHERE EXISTS (SELECT *
```

```
FROM igrac
```

```
WHERE idigrac = tim.idosnivac
```

```
AND korIme = SESSION_USER)
```

```
WITH CHECK OPTION;
```

```
GRANT UPDATE,DELETE ON public.TimoviJaOsnivac to jatkins;
```

4.1 (4 boda)

Nad bazom podataka u Postgres SUBP-a aktivne su tri sjednice u kojima se odvijaju transakcije A, B i C. Tablica desno prikazuje sadržaj relacije **paket** neposredno prije započinjanja naredbe {1}.

paket		
<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>
1	01:00:00	150
2	01:00:00	100

Redni brojevi u vitičastim zagradama ukazuju na redoslijed izvršavanja naredbi.

Transakcija A		Transakcija B		Transakcija C	
{1}	BEGIN TRANSACTION;	{2}	BEGIN TRANSACTION;	{3}	BEGIN TRANSACTION; SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
{6}	INSERT INTO paket VALUES(1, '30 minutes', 100);	{4}	SELECT trajanje, cijena FROM paket WHERE idIgra = 1;	{5}	SELECT trajanje, cijena FROM paket WHERE idIgra = 1;
{7}	COMMIT TRANSACTION;	{8}	SELECT trajanje, cijena FROM paket WHERE idIgra = 1;	{9}	SELECT trajanje, cijena FROM paket WHERE idIgra = 1;
		{10}	INSERT INTO paket VALUES(1, '2 hours', 180);		
		{11}	SELECT trajanje, cijena FROM paket WHERE idIgra = 1;	{12}	SELECT trajanje, cijena FROM paket WHERE idIgra = 1;
		{14}	COMMIT TRANSACTION;	{13}	COMMIT TRANSACTION;

Pretpostavite da:

- u sustavu nema drugih aktivnih korisnika niti sjednica, osim navedenih triju
- su sve prethodne transakcije uspješno završile izvođenje

a) Navedite rezultate SELECT naredbi pod {8}, {9}, {11} i {12}

Uz svaki od rezultata komentirajte je li se u transakciji dogodio neki od tipičnih problema istodobnog pristupa? Ako da, navedite naziv problema, koja n-torka/n-torke su ga uzrokovale i objasnite kako je došlo do problema.

b) Ako postoji razlika u rezultatima naredbi {8} i {9} objasnite zbog čega je do razlike došlo.

Rješenje:

a)

{8}	<table><tr><th><u>idIgra</u></th><th><u>trajanje</u></th><th><u>cijena</u></th></tr><tr><td>1</td><td>00:30:00</td><td>100</td></tr><tr><td>1</td><td>01:00:00</td><td>150</td></tr></table>	<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>	1	00:30:00	100	1	01:00:00	150	U transakciji B se pojavila sablasna n-torka: (1, 00:30:00, 10) Uzrok pojave sablasne n-torke je taj što se snimka transakcije kod pretpostavljene razine izolacije (READ COMMITTED) određuje na početku svake naredbe u transakciji, a u trenutku obavljanja naredbe 8, transakcija A je potvrđena i njeni efekti su vidljivi transakciji B			
<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>												
1	00:30:00	100												
1	01:00:00	150												
{9}	<table><tr><th><u>idIgra</u></th><th><u>trajanje</u></th><th><u>cijena</u></th></tr><tr><td>1</td><td>01:00:00</td><td>150</td></tr></table>	<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>	1	01:00:00	150	U transakciji C se nije pojavio nijedan tipični problem istodobnog pristupa.						
<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>												
1	01:00:00	150												
{11}	<table><tr><th><u>idIgra</u></th><th><u>trajanje</u></th><th><u>cijena</u></th></tr><tr><td>1</td><td>00:30:00</td><td>100</td></tr><tr><td>1</td><td>01:00:00</td><td>150</td></tr><tr><td>1</td><td>02:00:00</td><td>180</td></tr></table>	<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>	1	00:30:00	100	1	01:00:00	150	1	02:00:00	180	U transakciji B je i dalje prisutna sablasna n-torka: (1, 00:30:00, 10) Objašnjenje jednako kao kod naredbe 8.
<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>												
1	00:30:00	100												
1	01:00:00	150												
1	02:00:00	180												
{12}	<table><tr><th><u>idIgra</u></th><th><u>trajanje</u></th><th><u>cijena</u></th></tr><tr><td>1</td><td>01:00:00</td><td>150</td></tr></table>	<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>	1	01:00:00	150	U transakciji C se nije pojavio nijedan tipični problem istodobnog pristupa.						
<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>												
1	01:00:00	150												

b)

{8} i {9}	<p>Razlika postoji i posljedica je razlike u trenutku određivanja snimke transakcije za razine izolacije READ COMMITTED transakcije B i SERIALIZABLE transakcije C.</p> <p>Za READ COMMITTED se snimka transakcije određuje na početku svake naredbe transakcije, a za SERIALIZABLE jednom na početku transakcije.</p> <p>Zbog toga efekti potvrđene transakcije A, tj. INSERTirana ntorka (1, 00:30:00, 100) će biti vidljivi transakciji B pri izvršavanju naredbe {8}, ali neće biti vidljivi transakciji C pri izvršavanju naredbe {9}.</p>
-----------	---

4.2 (4 boda)

Nad bazom podataka u Postgres SUBP-a aktivne su tri sjednice u kojima se odvijaju transakcije A, B i C. Tablica desno prikazuje sadržaj relacije **paket** neposredno prije započinjanja naredbe {1}.

Redni brojevi u vitičastim zagradama ukazuju na redoslijed izvršavanja naredbi.

paket		
idIgra	trajanje	cijena
1	01:00:00	150
2	01:00:00	100

Transakcija A		Transakcija B		Transakcija C	
{1}	BEGIN TRANSACTION;	{2}	BEGIN TRANSACTION;	{3}	BEGIN TRANSACTION;
{4}	SELECT trajanje, cijena FROM paket WHERE idIgra = 1;	{6}	INSERT INTO paket VALUES (1, '30 minutes', 100);	{5}	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; SELECT trajanje, cijena FROM paket WHERE idIgra = 1;
{8}	SELECT trajanje, cijena FROM paket WHERE idIgra = 1;	{7}	COMMIT TRANSACTION;	{9}	SELECT trajanje, cijena FROM paket WHERE idIgra = 1;
{11}	SELECT trajanje, cijena FROM paket WHERE idIgra = 1;			{10}	INSERT INTO paket VALUES (1, '2 hours', 180);
{14}	COMMIT TRANSACTION;			{12}	SELECT trajanje, cijena FROM paket WHERE idIgra = 1;
				{13}	COMMIT TRANSACTION;

Pretpostavite da:

- u sustavu nema drugih aktivnih korisnika niti sjednica, osim navedenih triju
- su sve prethodne transakcije uspješno završile izvođenje

a) Navedite rezultate SELECT naredbi pod {8}, {9}, {11} i {12}

Uz svaki od rezultata komentirajte je li se u transakciji dogodio neki od tipičnih problema istodobnog pristupa? Ako da, navedite naziv problema, koja n-torka/n-torke su ga uzrokovale i objasnite kako je došlo do problema.

b) Ako postoji razlika u rezultatima naredbi {8} i {9} objasnite zbog čega je do razlike došlo.

Rješenje:

a)

{8}	<table> <tr> <th>idIgra</th><th>trajanje</th><th>cijena</th></tr> <tr> <td>1</td><td>00:30:00</td><td>100</td></tr> <tr> <td>1</td><td>01:00:00</td><td>150</td></tr> </table>	idIgra	trajanje	cijena	1	00:30:00	100	1	01:00:00	150	U transakciji A se pojavila sablasna n-torka: (1, 00:30:00, 10) Uzrok pojave sablasne n-torke je taj što se snimka transakcije kod pretpostavljene razine izolacije (READ COMMITTED) određuje na početku svake naredbe u transakciji, a u trenutku obavljanja naredbe 8, transakcija B je potvrđena i njeni efekti su vidljivi transakciji A
idIgra	trajanje	cijena									
1	00:30:00	100									
1	01:00:00	150									
{9}	<table> <tr> <th>idIgra</th><th>trajanje</th><th>cijena</th></tr> <tr> <td>1</td><td>01:00:00</td><td>150</td></tr> </table>	idIgra	trajanje	cijena	1	01:00:00	150	U transakciji C se nije pojavio nijedan tipični problem istodobnog pristupa.			
idIgra	trajanje	cijena									
1	01:00:00	150									
{11}	<table> <tr> <th>idIgra</th><th>trajanje</th><th>cijena</th></tr> <tr> <td>1</td><td>00:30:00</td><td>100</td></tr> <tr> <td>1</td><td>01:00:00</td><td>150</td></tr> </table>	idIgra	trajanje	cijena	1	00:30:00	100	1	01:00:00	150	U transakciji A je i dalje prisutna sablasna n-torka: (1, 00:30:00, 10).
idIgra	trajanje	cijena									
1	00:30:00	100									
1	01:00:00	150									
{12}	<table> <tr> <th>idIgra</th><th>trajanje</th><th>cijena</th></tr> <tr> <td>1</td><td>01:00:00</td><td>150</td></tr> <tr> <td>1</td><td>02:00:00</td><td>180</td></tr> </table>	idIgra	trajanje	cijena	1	01:00:00	150	1	02:00:00	180	U transakciji C se nije pojavio nijedan tipični problem istodobnog pristupa.
idIgra	trajanje	cijena									
1	01:00:00	150									
1	02:00:00	180									

b)

{8} i {9}	<p>Razlika postoji i posljedica je razlike u trenutku određivanja snimke transakcije za razine izolacije READ COMMITTED transakcije A i SERIALIZABLE transakcije C.</p> <p>Za READ COMMITTED se snimka transakcije određuje na početku svake naredbe transakcije, a za SERIALIZABLE jednom na početku transakcije.</p> <p>Zbog toga efekti potvrđene transakcije B, tj. INSERTirana ntorka (1, 00:30:00, 100) će biti vidljivi transakciji A pri izvršavanju naredbe {8}, ali neće biti vidljivi transakciji C pri izvršavanju naredbe {9}.</p>
-----------	---

4.3 (4 boda)

Nad bazom podataka u Postgres SUBP-a aktivne su tri sjednice u kojima se odvijaju transakcije A, B i C. Tablica desno prikazuje sadržaj relacije **paket** neposredno prije započinjanja naredbe {1}.

paket		
idIgra	trajanje	cijena
1	01:00:00	150
2	01:00:00	100

Redni brojevi u vitičastim zagradama ukazuju na redoslijed izvršavanja naredbi.

Transakcija A		Transakcija B		Transakcija C	
{1}	BEGIN TRANSACTION;	{2}	BEGIN TRANSACTION;	{3}	BEGIN TRANSACTION;
{6}	INSERT INTO paket VALUES(2, '30 minutes', 75);	{4}	SELECT * FROM paket WHERE idIgra = 2;	{5}	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; SELECT * FROM paket WHERE idIgra = 2;
{7}	COMMIT TRANSACTION;	{8}	SELECT * FROM paket WHERE idIgra = 2;	{9}	SELECT * FROM paket WHERE idIgra = 2;
		{10}	INSERT INTO paket VALUES(2, '2 hours', 180);		
		{11}	SELECT * FROM paket WHERE idIgra = 2;	{12}	SELECT * FROM paket WHERE idIgra = 2;
		{14}	COMMIT TRANSACTION;	{13}	COMMIT TRANSACTION;

Pretpostavite da:

- u sustavu nema drugih aktivnih korisnika niti sjednica, osim navedenih triju
- su sve prethodne transakcije uspješno završile izvođenje

a) Navedite rezultate SELECT naredbi pod {8}, {9}, {11} i {12}

Uz svaki od rezultata komentirajte je li se u transakciji dogodio neki od tipičnih problema istodobnog pristupa? Ako da, navedite naziv problema, koja n-torka/n-torke su ga uzrokovale i objasnite kako je došlo do problema.

b) Ako postoji razlika u rezultatima naredbi {8} i {9} objasnite zbog čega je do razlike došlo.

Rješenje:

a)

{8}	<table> <tr> <th>idIgra</th><th>trajanje</th><th>cijena</th></tr> <tr> <td>2</td><td>01:00:00</td><td>100</td></tr> <tr> <td>2</td><td>00:30:00</td><td>75</td></tr> </table>	idIgra	trajanje	cijena	2	01:00:00	100	2	00:30:00	75	<p>U transakciji B se pojavila sablasna n-torka: (2, 00:30:00, 75)</p> <p>Uzrok pojave sablasne n-torke je taj što se snimka transakcije kod pretpostavljene razine izolacije (READ COMMITTED) određuje na početku svake naredbe u transakciji, a u trenutku obavljanja naredbe 8, transakcija A je potvrđena i njeni efekti su vidljivi transakciji B</p>			
idIgra	trajanje	cijena												
2	01:00:00	100												
2	00:30:00	75												
{9}	<table> <tr> <th>idIgra</th><th>trajanje</th><th>cijena</th></tr> <tr> <td>2</td><td>01:00:00</td><td>100</td></tr> </table>	idIgra	trajanje	cijena	2	01:00:00	100	U transakciji C se nije pojavio nijedan tipični problem istodobnog pristupa.						
idIgra	trajanje	cijena												
2	01:00:00	100												
{11}	<table> <tr> <th>idIgra</th><th>trajanje</th><th>cijena</th></tr> <tr> <td>2</td><td>01:00:00</td><td>100</td></tr> <tr> <td>2</td><td>00:30:00</td><td>75</td></tr> <tr> <td>2</td><td>02:00:00</td><td>180</td></tr> </table>	idIgra	trajanje	cijena	2	01:00:00	100	2	00:30:00	75	2	02:00:00	180	<p>U transakciji B je i dalje prisutna sablasna n-torka: (2, 00:30:00, 75)</p> <p>Objašnjenje jednako kao kod naredbe 8</p>
idIgra	trajanje	cijena												
2	01:00:00	100												
2	00:30:00	75												
2	02:00:00	180												
{12}	<table> <tr> <th>idIgra</th><th>trajanje</th><th>cijena</th></tr> <tr> <td>2</td><td>01:00:00</td><td>100</td></tr> </table>	idIgra	trajanje	cijena	2	01:00:00	100	U transakciji C se nije pojavio nijedan tipični problem istodobnog pristupa.						
idIgra	trajanje	cijena												
2	01:00:00	100												

b)

{8} i {9}	<p>Razlika postoji i posljedica je razlike u trenutku određivanja snimke transakcije za razine izolacije READ COMMITTED transakcije B i SERIALIZABLE transakcije C.</p> <p>Za READ COMMITTED se snimka transakcije određuje na početku svake naredbe transakcije, a za SERIALIZABLE jednom na početku transakcije.</p> <p>Zbog toga efekti potvrđene transakcije A, tj. INSERTirana ntorka (2, 00:30:00, 75) će biti vidljivi transakciji B pri izvršavanju naredbe {8}, ali neće biti vidljivi transakciji C pri izvršavanju naredbe {9}.</p>
-----------	--

4.4 (4 boda)

Nad bazom podataka u Postgres SUBP-a aktivne su tri sjednice u kojima se odvijaju transakcije A, B i C. Tablica desno prikazuje sadržaj relacije **paket** neposredno prije započinjanja naredbe {1}.

paket		
idIgra	trajanje	cijena
1	01:00:00	150
2	01:00:00	100

Redni brojevi u vitičastim zagradama ukazuju na redoslijed izvršavanja naredbi.

Transakcija A		Transakcija B		Transakcija C	
{1}	BEGIN TRANSACTION;	{2}	BEGIN TRANSACTION;	{3}	BEGIN TRANSACTION;
{4}	SELECT * FROM paket WHERE idIgra = 2;	{6}	INSERT INTO paket VALUES (2, '30 minutes', 80);	{5}	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; SELECT * FROM paket WHERE idIgra = 2;
{8}	SELECT * FROM paket WHERE idIgra = 2;	{7}	COMMIT TRANSACTION;	{9}	SELECT * FROM paket WHERE idIgra = 2;
{11}	SELECT * FROM paket WHERE idIgra = 2;			{10}	INSERT INTO paket VALUES (2, '2 hours', 190);
{14}	COMMIT TRANSACTION;			{12}	SELECT * FROM paket WHERE idIgra = 2;
				{13}	COMMIT TRANSACTION;

Pretpostavite da:

- u sustavu nema drugih aktivnih korisnika niti sjednica, osim navedenih triju
- su sve prethodne transakcije uspješno završile izvođenje

a) Navedite rezultate SELECT naredbi pod {8}, {9}, {11} i {12}

Uz svaki od rezultata komentirajte je li se u transakciji dogodio neki od tipičnih problema istodobnog pristupa? Ako da, navedite naziv problema, koja n-torka/n-torke su ga uzrokovale i objasnite kako je došlo do problema.

b) Ako postoji razlika u rezultatima naredbi {8} i {9} objasnite zbog čega je do razlike došlo.

Rješenje:

a)

{8}	<table><tr><th><u>idIgra</u></th><th><u>trajanje</u></th><th><u>cijena</u></th></tr><tr><td>2</td><td>01:00:00</td><td>100</td></tr><tr><td>2</td><td>00:30:00</td><td>80</td></tr></table>	<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>	2	01:00:00	100	2	00:30:00	80	U transakciji A se pojavila sablasna n-torka: (2, 00:30:00, 80) Uzrok pojave sablasne n-torke je taj što se snimka transakcije kod pretpostavljene razine izolacije (READ COMMITTED) određuje na početku svake naredbe u transakciji, a u trenutku obavljanja naredbe 8, transakcija B je potvrđena i njeni efekti su vidljivi transakciji A
<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>									
2	01:00:00	100									
2	00:30:00	80									
{9}	<table><tr><th><u>idIgra</u></th><th><u>trajanje</u></th><th><u>cijena</u></th></tr><tr><td>2</td><td>01:00:00</td><td>100</td></tr></table>	<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>	2	01:00:00	100	U transakciji C se nije pojavio nijedan tipični problem istodobnog pristupa.			
<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>									
2	01:00:00	100									
{11}	<table><tr><th><u>idIgra</u></th><th><u>trajanje</u></th><th><u>cijena</u></th></tr><tr><td>2</td><td>01:00:00</td><td>100</td></tr><tr><td>2</td><td>00:30:00</td><td>80</td></tr></table>	<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>	2	01:00:00	100	2	00:30:00	80	U transakciji A je i dalje prisutna sablasna n-torka: (2, 00:30:00, 80).
<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>									
2	01:00:00	100									
2	00:30:00	80									
{12}	<table><tr><th><u>idIgra</u></th><th><u>trajanje</u></th><th><u>cijena</u></th></tr><tr><td>2</td><td>01:00:00</td><td>100</td></tr><tr><td>2</td><td>02:00:00</td><td>190</td></tr></table>	<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>	2	01:00:00	100	2	02:00:00	190	U transakciji C se nije pojavio nijedan tipični problem istodobnog pristupa.
<u>idIgra</u>	<u>trajanje</u>	<u>cijena</u>									
2	01:00:00	100									
2	02:00:00	190									

b)

{8} i {9}	<p>Razlika postoji i posljedica je razlike u trenutku određivanja snimke transakcije za razine izolacije READ COMMITTED transakcije A i SERIALIZABLE transakcije C.</p> <p>Za READ COMMITTED se snimka transakcije određuje na početku svake naredbe transakcije, a za SERIALIZABLE jednom na početku transakcije.</p> <p>Zbog toga efekti potvrđene transakcije B, tj. INSERTirana ntorka (2, 00:30:00, 80) će biti vidljivi transakciji A pri izvršavanju naredbe {8}, ali neće biti vidljivi transakciji C pri izvršavanju naredbe {9}.</p>
-----------	--

5.1 (4 boda)

U bazi podataka koja se koristi u ovoj provjeri definirana su ograničenja primarnih i stranih ključeva. Optimizator upita raspolaže sljedećim statističkim podacima:

N(igrac) = 900	N(tim) = 1200	N(borba) = 6000
V(ime, igrac) = 100	V(idOsnivac, tim) = 600	V(idTim1, borba) = 100
V(datRodjenja, igrac) = 200	V(datVrOsnutak, tim) = 300	V(idTim2, borba) = 150

Napomena: navedeni statistički podatci ne odgovaraju stvarnim statističkim podacima u bazi na Edgaru.

Izvodi se upit:

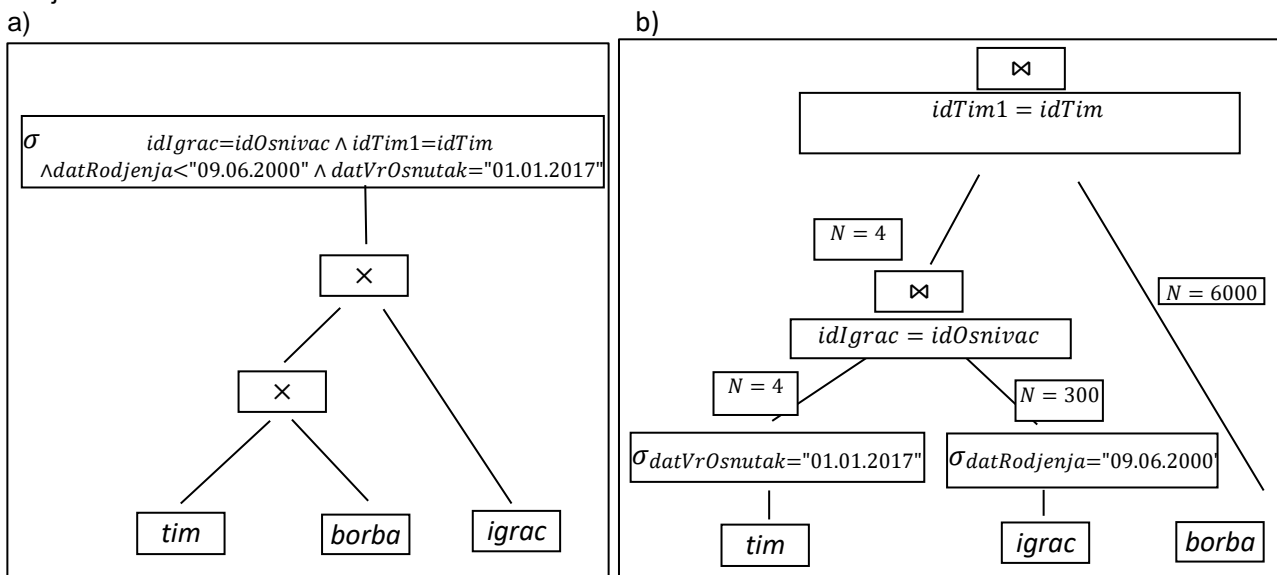
```
SELECT *
FROM tim, borba, igrac
WHERE idIgrac = idOsnivac
AND borba.idTim1 = tim.idTim
AND datVrOsnutak = '01.01.2017'
AND datRodjenja < '09.06.2000';
```

- Nacrtajte stablo upita za **početni plan** izvođenja upita pri čemu je redoslijed spajanja tablica određen redoslijedom kojim su tablice navedene u FROM dijelu SELECT naredbe. Crtež slikajte i **uploadajte ga**.
- Nacrtajte stablo upita **nakon** provedene heurističke optimizacije. Dovoljno je nacrtati samo konačno stablo upita. 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 i u konačnom rezultatu. U stablu upita naznačiti očekivani broj n-torki.

Napomena:

- Crtež stabla za početni plan izvođenja upita slikajte i **uploadajte**.
- Crtež stabla za plan izvođenja upita nakon provedene heurističke optimizacije slikajte i **uploadajte**.
- Postupak i izračun za procjenu broja n-torki **unesite u prostor za slobodni unos teksta ispod teksta zadatka**.

Rješenje:



Prvo se spajaju tim i igrac, zatim borba.

potiskivanje selekcije $\text{datVrOsnutak} = '01.01.2017'$ na **tim**:

$$N(\text{tim}') = N(\text{tim}) / V(\text{datVrOsnutak}, \text{tim}) = 1200/300 = 4$$

potiskivanje selekcije $\text{datRodjenja} < '01.01.2017'$ na **igrac**:

$$N(\text{igrac}') = N(\text{igrac})/3 = 900/3 = 300$$

Međurezultati spajanja:

tim' + borba (fk u borba) $\Rightarrow N(\text{borba}) = 6000$

tim' + igrac' (fk u tim) $\Rightarrow N(\text{tim}') = 4$

borba + igrac' (Kartezijev) $\Rightarrow 300 \cdot 6000 = 1\,800\,000$

5.2 (4 boda)

U bazi podataka koja se koristi u ovoj provjeri definirana su ograničenja primarnih i stranih ključeva. Optimizator upita raspolaže sljedećim statističkim podacima:

$N(\text{igrac}) = 900$ $V(\text{ime}, \text{igrac}) = 100$	$N(\text{tim}) = 1200$ $V(\text{idOsnivac}, \text{tim}) = 600$ $V(\text{datVrOsnutak}, \text{tim}) = 300$	$N(\text{borba}) = 6000$ $V(\text{idTim1}, \text{borba}) = 2000$ $V(\text{idTim2}, \text{borba}) = 3000$
--	---	--

Napomena: navedeni statistički podatci ne odgovaraju stvarnim statističkim podacima u bazi na Edgaru.

Izvodi se upit:

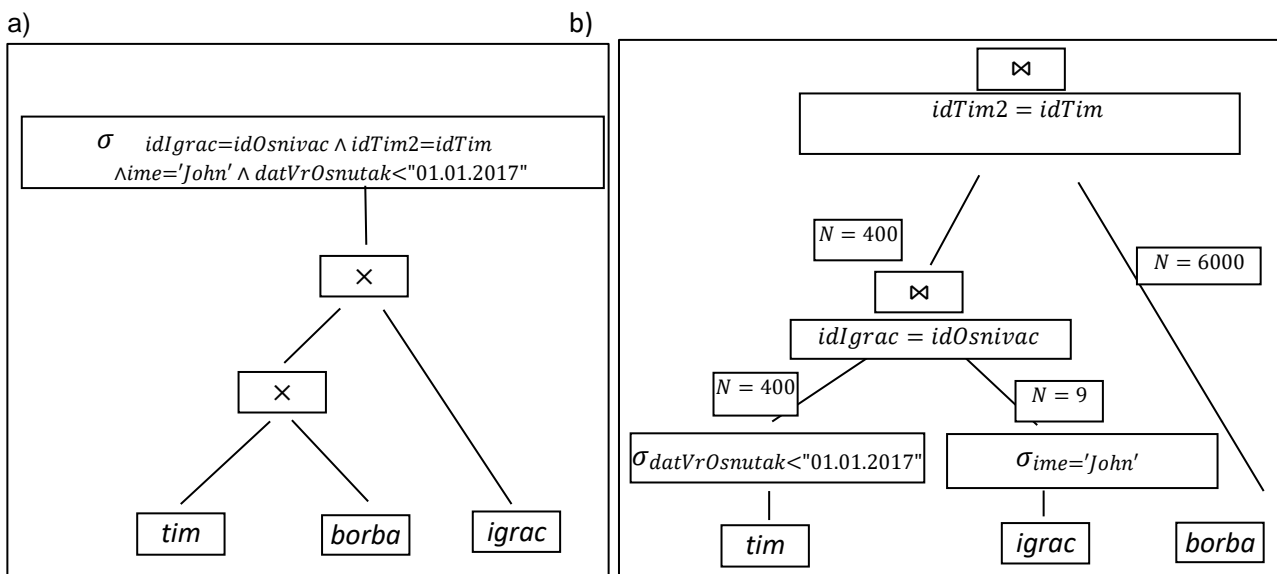
```
SELECT *
FROM tim, borba, igrac
WHERE idIgrac = idOsnivac
AND borba.idTim2 = tim.idTim
AND ime = 'John'
AND datVrOsnutak < '01.01.2017';
```

- Nacrtajte stablo upita za **početni plan** izvođenja upita pri čemu je redoslijed spajanja tablica određen redoslijedom kojim su tablice navedene u FROM dijelu SELECT naredbe. Crtež slikajte i **uploadajte ga**.
- Nacrtajte stablo upita **nakon** provedene heurističke optimizacije. Dovoljno je nacrtati samo konačno stablo upita. 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 i u konačnom rezultatu. U stablu upita naznačiti očekivani broj n-torki.

Napomena:

- Crtež stabla za početni plan izvođenja upita slikajte i **uploadajte**.
- Crtež stabla za plan izvođenja upita nakon provedene heurističke optimizacije slikajte i **uploadajte**.
- Postupak i izračun za procjenu broja n-torki **unesite u prostor za slobodni unos teksta ispod teksta zadatka**.

Rješenje:



Prvo se spajaju tim i igrac, zatim borba.

potiskivanje selekcije $\text{datVrOsnutak} < '01.01.2017'$ na **tim**:

$$N(\text{tim}') = N(\text{tim}) / 3 = 1200 / 3 = 400$$

potiskivanje selekcije $\text{ime} = 'John'$ na **igrac**:

$$N(\text{igrac}') = N(\text{igrac}) / V(\text{ime}, \text{igrac}) = 900 / 100 = 9$$

Međurezultati spajanja:

tim' + borba (fk u borba) $\Rightarrow N(\text{borba}) = 6000$

tim' + igrac' (fk u tim) $\Rightarrow N(\text{tim}') = 400$

borba + igrac' (Kartezijev) $\Rightarrow 9 * 6000 = 54\,000$

5.3 (4 boda)

U bazi podataka koja se koristi u ovoj provjeri definirana su ograničenja primarnih i stranih ključeva. Optimizator upita raspolaže sljedećim statističkim podacima:

N(igra) = 180 V(idArena, igra) = 20 V(nazivIgra, igra) = 150	N(arena) = 30 V(nazivArena, arena) = 30	N(paket)=720 V(cijena, paket) = 20 V(trajanje, paket) = 5
--	--	---

Napomena: navedeni statistički podatci ne odgovaraju stvarnim statističkim podacima u bazi na Edgaru.

Izvodi se upit:

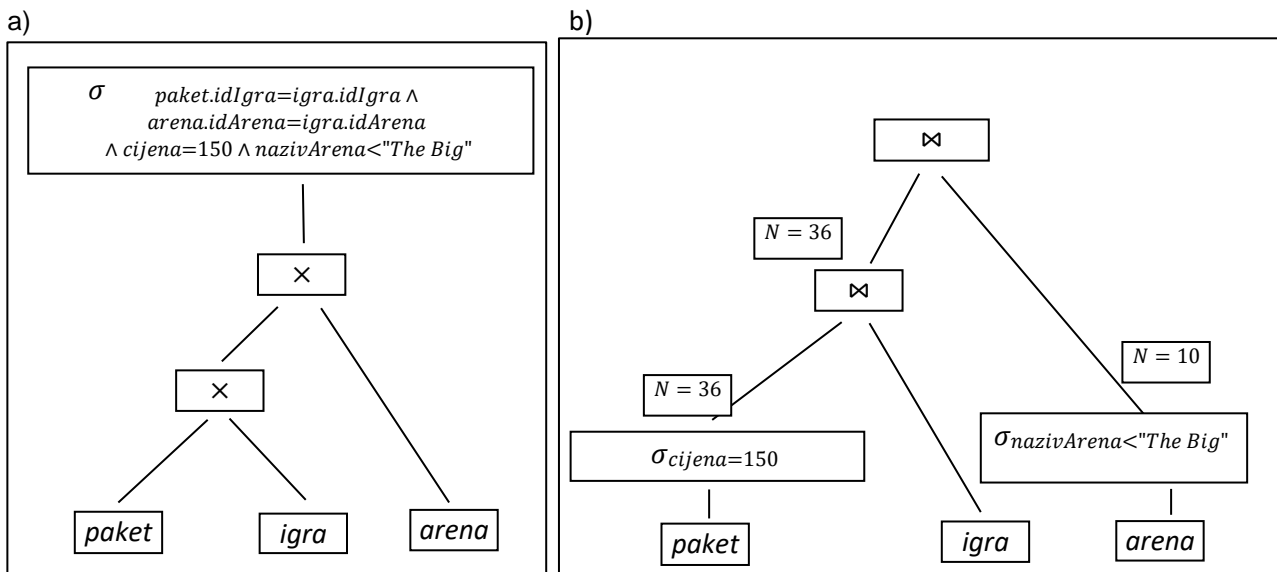
```
SELECT *
FROM paket, igra, arena
WHERE paket.idIgra = igra.idIgra
AND igra.idArena = arena.idArena
AND cijena = 150
AND nazivArena < 'The Big';
```

- Nacrtajte stablo upita za **početni plan** izvođenja upita pri čemu je redoslijed spajanja tablica određen redoslijedom kojim su tablice navedene u FROM dijelu SELECT naredbe. Crtež slikajte i **uploadajte ga**.
- Nacrtajte stablo upita **nakon** provedene heurističke optimizacije. Dovoljno je nacrtati samo konačno stablo upita. 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 i u konačnom rezultatu. U stablu upita naznačiti očekivani broj n-torki.

Napomena:

- Crtež stabla za početni plan izvođenja upita slikajte i **uploadajte**.
- Crtež stabla za plan izvođenja upita nakon provedene heurističke optimizacije slikajte i **uploadajte**.
- Postupak i izračun za procjenu broja n-torki **unesite u prostor za slobodni unos teksta ispod teksta zadatka**.

Rješenje:



Prvo se spajaju paket i igra, zatim arena.

b)

potiskivanje selekcije $cijena = 150$ na **paket**:

$$N(\text{paket}') = N(\text{paket}) / V(\text{cijena}, \text{paket}) = 720 / 20 = 36$$

potiskivanje selekcije $nazivArena < 'The Big'$ na **arena**:

$$N(\text{arena}') = N(\text{arena}) / 3 = 30 / 3 = 10$$

Međurezultati spajanja:

arena'+igra (fk u igra) $\Rightarrow N(\text{igra}) = 180$

arena'+paket (Kartezijev) $\Rightarrow 10 \cdot 36 = 360$

igra+paket (fk u paket) $\Rightarrow N(\text{paket}') = 36$

5.4 (4 boda)

U bazi podataka koja se koristi u ovoj provjeri definirana su ograničenja primarnih i stranih ključeva. Optimizator upita raspolaže sljedećim statističkim podacima:

N(igra) = 180 V(idArena, igra) = 20 V(nazivIgra, igra) = 150	N(arena) = 30 V(nazivArena, arena) = 30	N(paket)=720 V(cijena, paket) = 20 V(trajanje, paket) = 5
--	--	---

Napomena: navedeni statistički podatci ne odgovaraju stvarnim statističkim podacima u bazi na Edgaru.

Izvodi se upit:

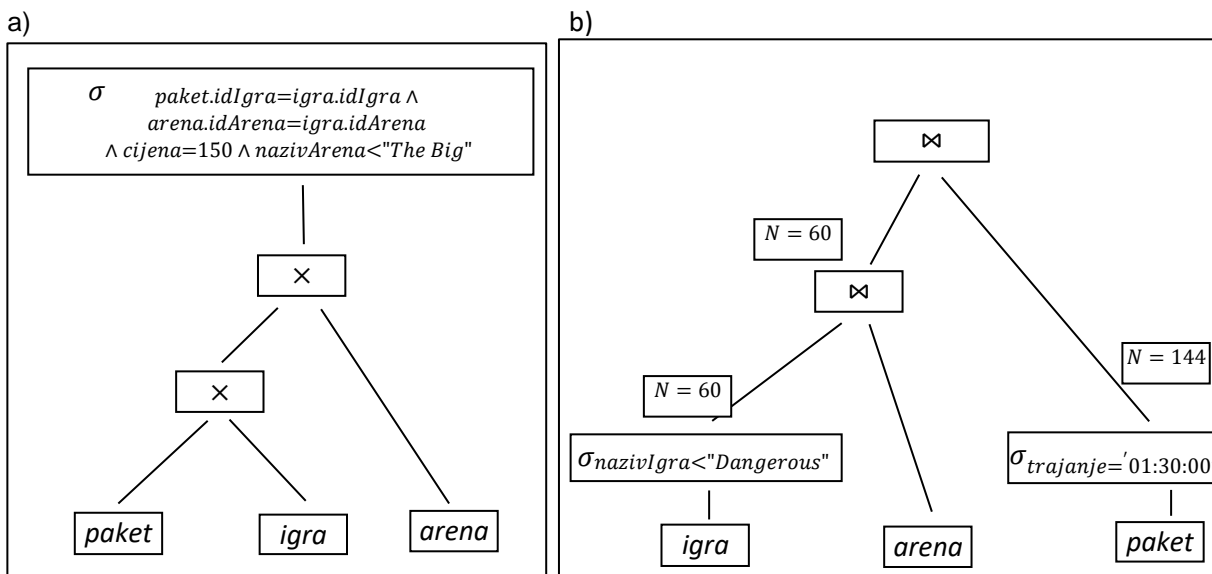
```
SELECT *
FROM paket, igra, arena
WHERE paket.idIgra = igra.idIgra
AND igra.idArena = arena.idArena
AND trajanje = '01:30:00'
AND nazivIgra < 'Dangerous';
```

- Nacrtajte stablo upita za **početni plan** izvođenja upita pri čemu je redoslijed spajanja tablica određen redoslijedom kojim su tablice navedene u FROM dijelu SELECT naredbe. Crtež slikajte i **uploadajte ga**.
- Nacrtajte stablo upita **nakon** provedene heurističke optimizacije. Dovoljno je nacrtati samo konačno stablo upita. 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 i u konačnom rezultatu. U stablu upita naznačiti očekivani broj n-torki.

Napomena:

- Crtež stabla za početni plan izvođenja upita slikajte i **uploadajte**.
- Crtež stabla za plan izvođenja upita nakon provedene heurističke optimizacije slikajte i **uploadajte**.
- Postupak i izračun za procjenu broja n-torki **unesite u prostor za slobodni unos teksta ispod teksta zadatka**.

Rješenje:



Prvo se spajaju arena i igra, zatim paket.

b)

potiskivanje selekcije trajanje = '01:30:00' na **paket**:

$$N(\text{paket}') = N(\text{paket})/V(\text{trajanje}, \text{paket}) = 720/5 = 144$$

potiskivanje selekcije nazivIgra < 'Dangerous' na **igra**:

$$N(\text{igra}') = N(\text{igra})/3 = 180/3 = 60$$

Međurezultati spajanja:

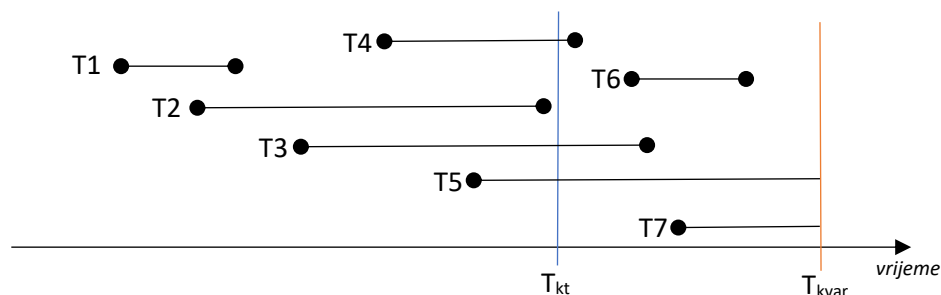
arena'+igra (fk u igra) => **N(igra') = 60**

arena'+paket' (Kartezijev) => 30*144 = 4320

igra+paket' (fk u paket) => N(paket') = 144

6.1 (4 boda)

U SUBP je došlo do pogreške na razini računalnog sustava u trenutku T_{kvar} . Posljednja kontrolna točka se dogodila u trenutku T_{kt} . Sadržaj dnevnika izmjena je prikazan na slici:



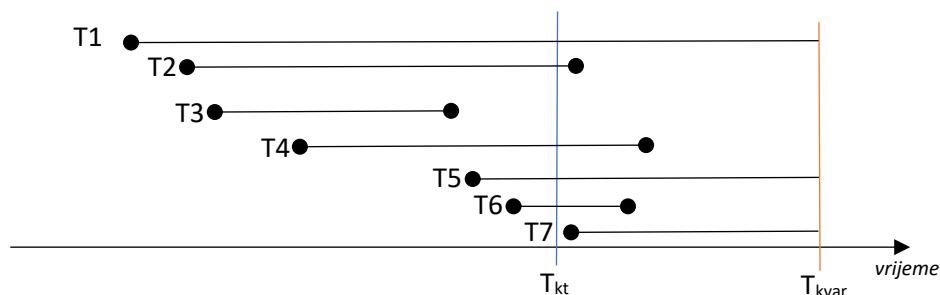
- Navedite identifikatore transakcija:
 - za koje ne treba izvršiti nikakvu akciju
 - koje treba poništiti
 - koje treba ponovno obaviti.
- Što je sadržano u zapisu kontrolne točke koja se dogodila u T_{kt} ?
- Koji je posljednji trenutak za kojeg pouzdano znamo da je sadržaj spremnika baze podataka pohranjen u bazu podataka?
- Koji je posljednji trenutak za kojeg pouzdano znamo da je sadržaj spremnika dnevnika pohranjen u datoteku dnevnika u sustavu koji poštuje svojstvo izdržljivosti transakcija?

Rješenje:

- ne treba izvršiti nikakvu akciju T1, T2
 - poništiti: T5, T7
 - ponovno obaviti T3, T4, T6
- Zapis kontrolne točke u T_{kt} sadrži:
 - listu svih aktivnih transakcija - T3, T4 i T5
 - za svaku od navedenih transakcija - adresu najnovijeg zapisa u datoteci dnevnika
- Spremnik baze podataka pohranjuje se u bazu podataka u kontrolnoj točki, dakle u T_{kt}
- Sadržaj spremnika dnevnika pohranjuje se u datoteku dnevnika na disku prije nego li završi procedura potvrđivanja transakcije T6 (Write-Ahead- Log Rule)

6.2 (4 boda)

U SUBP je došlo do pogreške na razini računalnog sustava u trenutku T_{kvar} . Posljednja kontrolna točka se dogodila u trenutku T_{kt} . Sadržaj dnevnika izmjena je prikazan na slici:



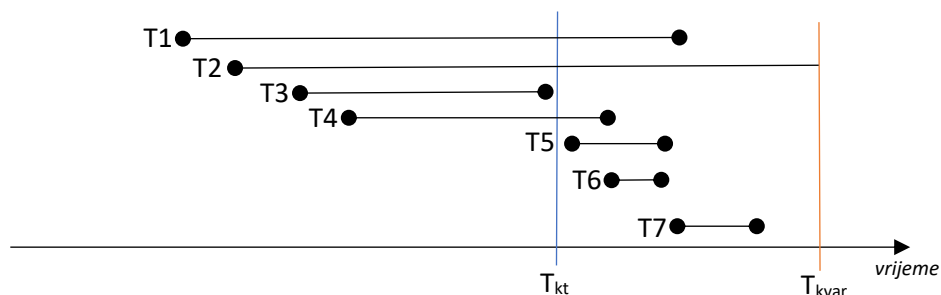
- Navedite identifikatore transakcija:
 - za koje ne treba izvršiti nikakvu akciju
 - koje treba poništiti
 - koje treba ponovno obaviti.
- Što je sadržano u zapisu kontrolne točke koja se dogodila u T_{kt} ?
- Koji je posljednji trenutak za kojeg pouzdano znamo da je sadržaj spremnika baze podataka pohranjen u bazu podataka?
- Koji je posljednji trenutak za kojeg pouzdano znamo da je sadržaj spremnika dnevnika pohranjen u datoteku dnevnika u sustavu koji poštuje svojstvo izdržljivosti transakcija?

Rješenje:

- ne treba izvršiti nikakvu akciju T3
 - poništiti: T1, T5, T7
 - ponovno obaviti T2, T4, T6
- Zapis kontrolne točke u T_{kt} sadrži:
 - listu svih aktivnih transakcija – T1, T2, T4, T5, T6
 - za svaku od navedenih transakcija - adresu najnovijeg zapisa u datoteci dnevnika
- Spremnik baze podataka pohranjuje se u bazu podataka u kontrolnoj točki, dakle u T_{kt}
- Sadržaj spremnika dnevnika pohranjuje se u datoteku dnevnika na disku prije nego li završi procedura potvrđivanja transakcije T4 (Write-Ahead- Log Rule)

6.3 (4 boda)

U SUBP je došlo do pogreške na razini računalnog sustava u trenutku T_{kvar} . Posljednja kontrolna točka se dogodila u trenutku T_{kt} . Sadržaj dnevnika izmjena je prikazan na slici:



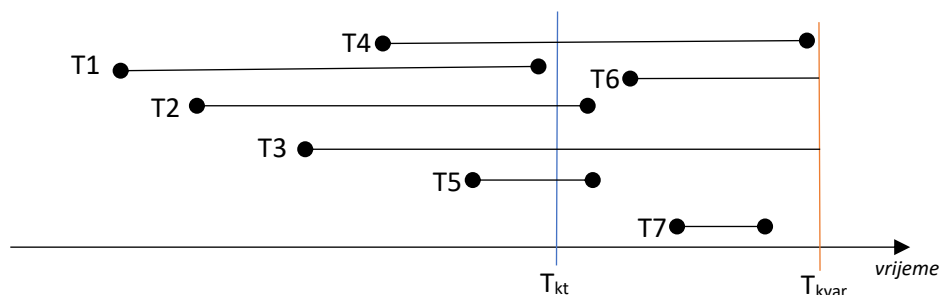
- Navedite identifikatore transakcija:
 - za koje ne treba izvršiti nikakvu akciju
 - koje treba poništiti
 - koje treba ponovno obaviti.
- Što je sadržano u zapisu kontrolne točke koja se dogodila u T_{kt} ?
- Koji je posljednji trenutak za kojeg pouzdano znamo da je sadržaj spremnika baze podataka pohranjen u bazu podataka?
- Koji je posljednji trenutak za kojeg pouzdano znamo da je sadržaj spremnika dnevnika pohranjen u datoteku dnevnika u sustavu koji poštuje svojstvo izdržljivosti transakcija?

Rješenje:

- ne treba izvršiti nikakvu akciju T3
 - poništiti: T2
 - ponovno obaviti T1, T4, T5, T6, T7
- Zapis kontrolne točke u T_{kt} sadrži:
 - listu svih aktivnih transakcija – T1, T2 i T4
 - za svaku od navedenih transakcija - adresu najnovijeg zapisa u datoteci dnevnika
- Spremnik baze podataka pohranjuje se u bazu podataka u kontrolnoj točki, dakle u T_{kt}
- Sadržaj spremnika dnevnika pohranjuje se u datoteku dnevnika na disku prije nego li završi procedura potvrđivanja transakcije T7 (Write-Ahead- Log Rule)

6.4 (4 boda)

U SUBP je došlo do pogreške na razini računalnog sustava u trenutku T_{kvar} . Posljednja kontrolna točka se dogodila u trenutku T_{kt} . Sadržaj dnevnika izmjena je prikazan na slici:



- a) Navedite identifikatore transakcija:
 - a1. za koje ne treba izvršiti nikakvu akciju
 - a2. koje treba poništiti
 - a3. koje treba ponovno obaviti.
- b) Što je sadržano u zapisu kontrolne točke koja se dogodila u T_{kt} ?
- c) Koji je posljednji trenutak za kojeg pouzdano znamo da je sadržaj spremnika baze podataka pohranjen u bazu podataka?
- d) Koji je posljednji trenutak za kojeg pouzdano znamo da je sadržaj spremnika dnevnika pohranjen u datoteku dnevnika u sustavu koji poštuje svojstvo izdržljivosti transakcija?

Rješenje:

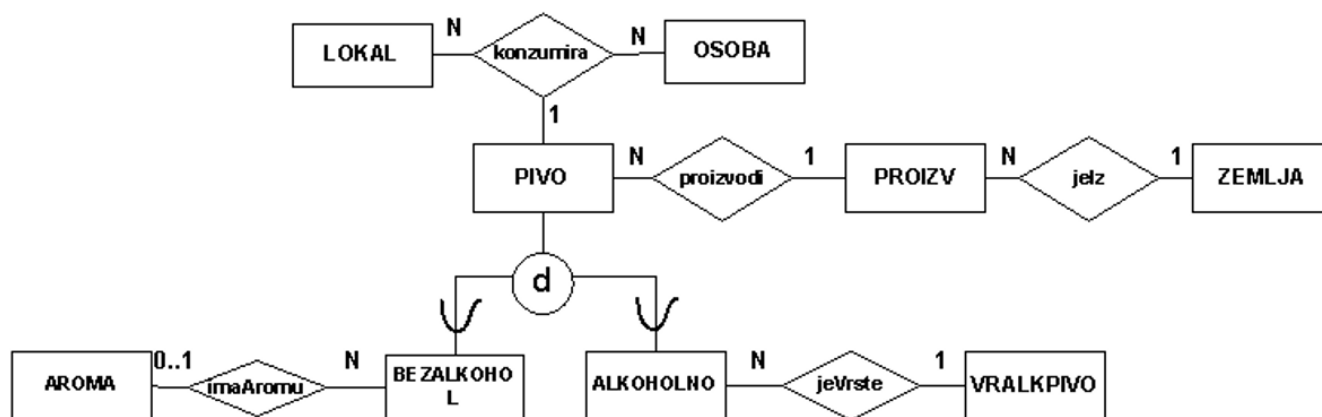
- a) a1. ne treba izvršiti nikakvu akciju T1
a2. poništiti: T3, T6
a3. ponovno obaviti T2, T4, T5, T7
- b) Zapis kontrolne točke u T_{kt} sadrži:
 - listu svih aktivnih transakcija – T2, T3, T4 i T5
 - za svaku od navedenih transakcija - adresu najnovijeg zapisa u datoteci dnevnika
- c) Spremnik baze podataka pohranjuje se u bazu podatak u kontrolnoj točki, dakle u T_{kt}
- d) Sadržaj spremnika dnevnika pohranjuje se u datoteku dnevnika na disku prije nego li završi procedura potvrđivanja transakcije T_4 (Write-Ahead- Log Rule)

7.1 (5 bodova)

Nakon popuštanja mjera okupljanja iduće subote se organizira pub crawl te je potrebno modelirati ER dijagram baze kojom će se bilježiti posjećenost lokala od strane sudionika te piva koja su pritom konzumirali po privremenoj promotivnoj cijeni. Za osobama koje sudjeluju evidentiraju se OIB, ime, prezime i datum rođenja. Za lokale se evidentira šifra i naziv. Osoba tu večer ide od lokala do lokala (redosljed obilaska nebitan) i u svakome konzumira **samo jedno** pivo po vlastitom izboru. Osoba se ne vraća u već posjećeni lokal. Isto pivo osoba može konzumirati u više lokala. Za pivo se bilježi šifra, naziv, regularna cijena, privremena promotivna cijena (pretpostavite da su regularna i promotivna cijena za isto pivo jednake među svim lokalima) i proizvođač. Za proizvođača se bilježe šifra, naziv i zemlja u kojoj je smješten. Za zemlju se bilježe šifra, jedinstvena oznaka (prema ISO standardu) i naziv. Pivo može biti alkoholno ili bezalkoholno. Bezalkoholno pivo može, ali ne mora biti obogaćeno jednom aromom (npr. grejp, limun, itd.). Za aromu se bilježe šifra i naziv. Za alkoholna piva se bilježi postotak alkohola te vrsta (*ale*, *stout*, itd.). Za vrstu piva se bilježi šifra i naziv.

- Nacrtajte ER model baze podataka (nemojte koristiti alat ERDPlus jer Vam neće biti dovoljan), slikajte nacrtani dijagram i **uploadajte ga**.
- U prostor za slobodni unos teksta ispod teksta zadatka navedite **sheme entiteta** i **sheme veza** (naznačite ključeve). Svaki entitet (osim slabih entiteta) opisati **isključivo vlastitim atributima**. Nužno je da sve sheme zadovoljavaju **3NF**.

Rješenje:



Entiteti

OSOBA = oib, ime, prezime, datumRod
 LOKAL = sifLokal, nazLokal
 PIVO = sifPivo, nazPivo, regCijena, promoCijena
 BEZALK_PIVO = sifPivo
 PROIZV = sifProizv, nazProizv
 ALK_PIVO = sifPivo, postotakAlk
 AROMA = sifAroma, nazAroma
 VRSTA = sifVrsta, nazVrsta
 ZEMLJA = sifZemlja, oznZemlja, nazZemlja
 (K1ZEMLJA = sifZemlja; K2ZEMLJA = oznZemlja)

Veze

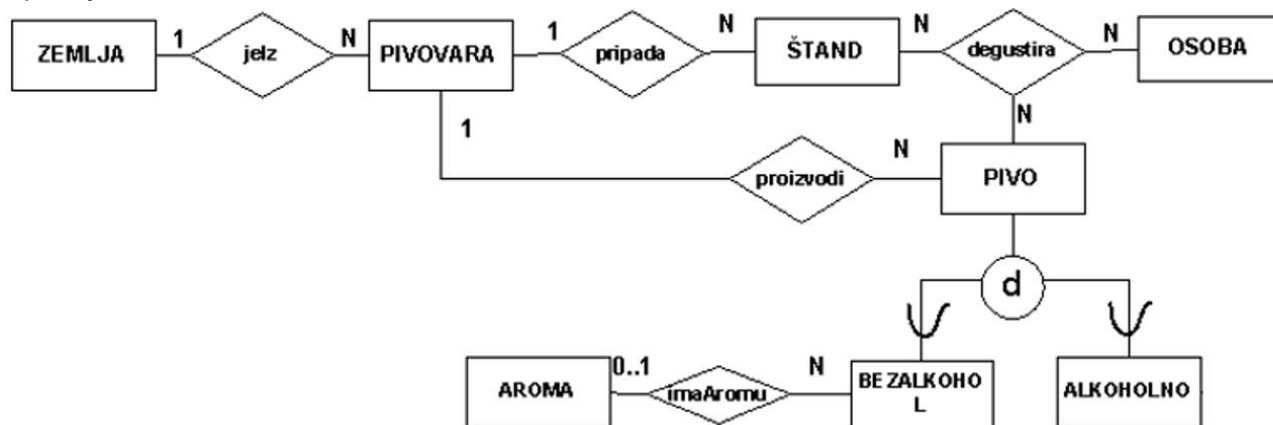
konzumira = sifLokal, oib, sifPivo
 proizvodi = sifPivo, sifProizv
 jelz = sifProizv, sifZemlja
 imaAromu = sifPivo, sifAroma
 jeVrste = sifPivo, sifVrsta

7.2 (5 bodova)

Nakon popuštanja mjera okupljanja iduće subote se organizira festival degustacije piva na kojem pivovare na svojim štandovima sudjeluju s isključivo vlastitim proizvodima. Potrebno je modelirati ER dijagram baze kojom će se bilježiti posjećenost štandova od strane sudionika te piva koja su pritom degustirali. Za osobe koje sudjeluju evidentiraju se OIB, ime, prezime i datum rođenja. Za štandove se evidentira šifra i opis lokacije (npr. „ulaz 1 lijevo“). Štand pripada jednoj pivovari. Jedna pivovara može izlagati proizvode na više štandova. Za pivovare se evidentiraju šifra, naziv, godina utemeljenja te zemlja porijekla. Za zemlju se bilježi šifra, jedinstvena oznaka (prema ISO standardu) i naziv. Osoba tijekom festivala obilazi štandove i na svakome može degustirati više različitih piva (moguće isto pivo iste pivovare na različitim njezinim štandovima). Pri tome se evidentira samo informacija koje je pivo osoba degustirala na kojem štandu – ponovljene degustacije se ****ne**** evidentiraju. Za pivo se bilježi šifra, naziv, regularna cijena te pivovara koja ga proizvodi. Pivo može biti alkoholno ili bezalkoholno. Za alkoholna piva se bilježi postotak alkohola i promotivna cijena. Bezalkoholno pivo može, ali ne mora biti obogaćeno jednom aromom (npr. grejp, limun, itd.). Za aromu se bilježi šifra i naziv.

- Nacrtajte ER model baze podataka (nemojte koristiti alat ERDPlus jer Vam neće biti dovoljan), slikajte nacrtani dijagram i **uploadajte ga**.
- U prostor za slobodni unos teksta ispod teksta zadatka navedite **scheme entiteta** i **scheme veza** (naznačite ključeve). Svaki entitet (osim slabih entiteta) opisati **isključivo vlastitim atributima**. Nužno je da sve sheme zadovoljavaju **3NF**.

Rješenje:



Entiteti

OSOBA = oib, ime, prezime, datumRod
 STAND = sifStand, opisLok
 PIVOVARA = sifPivov, nazPivov, godUtemeljenja
 PIVO = sifPivo, nazPivo, regCijena
 BEZALK_PIVO = sifPivo
 ALK_PIVO = sifPivo, postotakAlk, promoCijena
 AROMA = sifAroma, nazAroma
 ZEMLJA = sifZemlja, oznZemlja, nazZemlja
 (K1ZEMLJA = sifZemlja; K2ZEMLJA = oznZemlja)

Veze

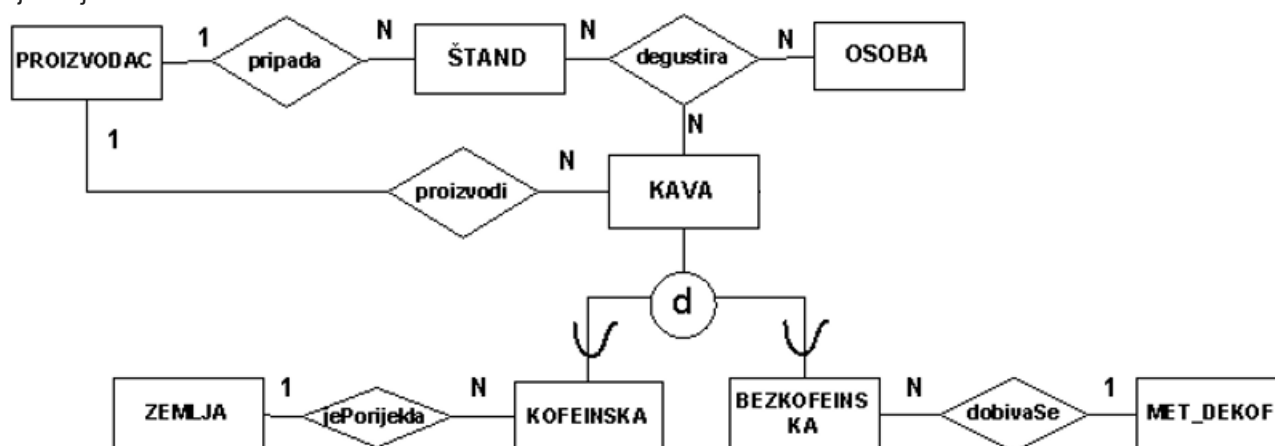
degustira = sifLokal, oib, sifPivo
 proizvodi = sifPivo, sifPivov
 pripada = sifStand, sifPivov
 jelz = sifPivov, sifZemlja
 imaAromu = sifPivo, sifAroma

7.3 (5 bodova)

Nakon popuštanja mjera okupljanja iduće subote se organizira festival degustacije kave na štandovima proizvođača kave. Potrebno je modelirati ER dijagram baze kojom će se bilježiti posjećenost štandova od strane sudionika te kave koju su pritom degustirali. Za osobe koje sudjeluju evidentiraju se OIB, ime, prezime i godina rođenja. Za štandove se evidentira šifra i opis lokacije (npr. „ulaz 1 lijevo“). Štand pripada jednom proizvođaču. Jedan proizvođač može izlagati proizvode na više štandova. Za proizvođače se evidentiraju šifra, naziv i godina utemeljenja. Osoba tijekom festivala obilazi štandove i na svakome može degustirati više različitih kava (moguće i istu kavu istog proizvođača na njegovim različitim štandovima). Pri tome se evidentira samo informacija koju je kavu osoba degustirala na kojem štandu – ponovljene degustacije se ****ne**** evidentiraju. Za kavu se bilježi šifra, naziv, cijena i proizvođač. Kava može biti kofeinska ili bezkofeinska. Za bezkofeinsku kavu bilježi se kojom metodom odstranjivanja kofeina iz kave je ona nastala (npr. „Zoselov proces“, „švicarski vodeni proces“, itd.). Za metodu odstranjivanja kofeina iz kave bilježe se šifra i naziv. Za kofeinske kave bilježi se ocjena jačine te zemlja porijekla kave. Za zemlju se bilježe šifra, jedinstvena oznaka (prema ISO standardu) i naziv.

- Nacrtajte ER model baze podataka (nemojte koristiti alat ERDPlus jer Vam neće biti dovoljan), slikajte nacrtani dijagram i **uploadajte ga**.
- U prostor za slobodni unos teksta ispod teksta zadatka navedite **sheme entiteta** i **sheme veza** (naznačite ključeve). Svaki entitet (osim slabih entiteta) opisati **isključivo vlastitim atributima**. Nužno je da sve sheme zadovoljavaju **3NF**.

Rješenje:



Entiteti

OSOBA = oib, ime, prezime, godinaRod
 STAND = sifStand, opisLok
 PROIZVODAC = sifProizv, nazProizv, godUtemeljenja
 KAVA = sifKava, nazKava, cijena
 BEZKOFEINSKA = sifKava
 KOFEINSKA = sifKava, jacina
 MET_DEKOF = sifMetoda, nazMetoda
 ZEMLJA = sifZemlja, oznZemlja, nazZemlja
 (K1ZEMLJA = sifZemlja; K2ZEMLJA = oznZemlja)

Veze

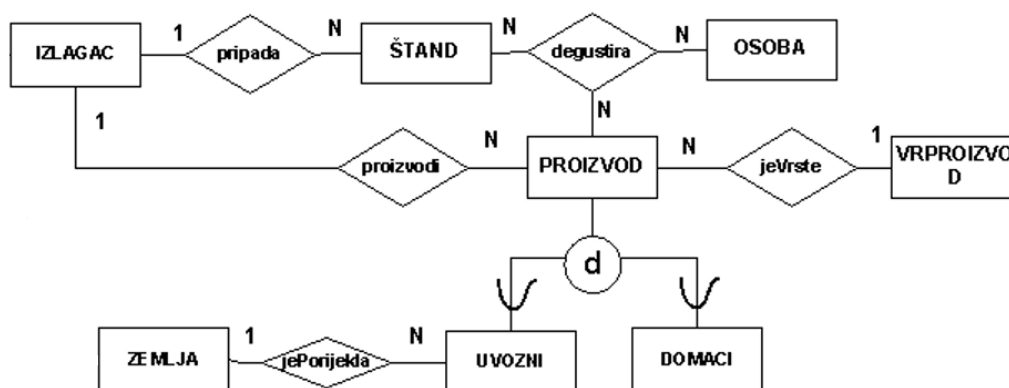
degustira = sifKava, oib, sifStand
 pripada = sifStand, sifProizv
 proizvodi = sifKava, sifProizv
 dobivaSe = sifKava, sifMetoda
 jePorijekla = sifKava, sifZemlja

7.4 (5 bodova)

Nakon popuštanja mjera okupljanja iduće subote se organizira degustacijski festival eko-proizvoda domaćih i stranih izlagača. Potrebno je modelirati ER dijagram baze kojom će se bilježiti posjećenost štandova od strane posjetitelja te proizvoda koje su pritom degustirali. Za osobe koje posjećuju festival evidentiraju se OIB, ime i prezime. Za štandove se evidentira šifra i opis lokacije (npr. „ulaz 1 lijevo“). Štand pripada jednom izlagaču (proizvođaču). Jedan izlagač može izlagati proizvode na više štandova. Za izlagače se evidentiraju šifra i naziv. Osoba tijekom festivala obilazi štandove i na svakome može degustirati više različitih proizvoda (moguće i istih proizvoda istog izlagača na njegovim različitim štandovima). Pri tome se evidentira samo informacija koji je proizvod osoba degustirala na kojem štandu – ponovljene degustacije se ****ne**** evidentiraju. Za proizvod se bilježi šifra, naziv, cijena, izlagač koji ga proizvodi i vrsta (npr. med, maslinovo ulje, itd.). Za vrstu proizvoda se evidentiraju šifra i naziv. Proizvod može biti domaći ili uvozni. Za uvozne proizvode evidentira se zemlja porijekla, dok se za domaće evidentira godina dobivanja eko-certifikata. Za zemlju se bilježe šifra, jedinstvena oznaka (prema ISO standardu) i naziv.

- Nacrtajte ER model baze podataka (nemojte koristiti alat ERDPlus jer Vam neće biti dovoljan), slikajte nacrtani dijagram i **uploadajte ga**.
- U prostor za slobodni unos teksta ispod teksta zadatka navedite **scheme entiteta** i **scheme veza** (naznačite ključeve). Svaki entitet (osim slabih entiteta) opisati **isključivo vlastitim atributima**. Nužno je da sve sheme zadovoljavaju **3NF**.

Rješenje:

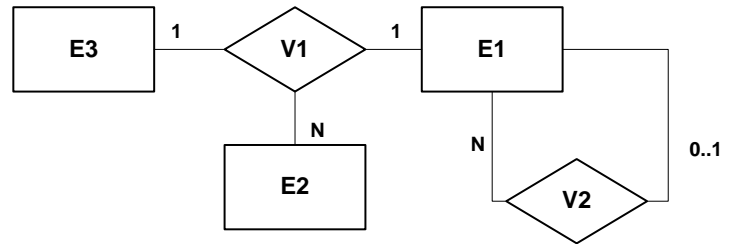


Entiteti	Veze
<p>OSOBA = <u>oib</u>, ime, prezime</p> <p>STAND = <u>sifStand</u>, opisLok</p> <p>IZLAGAC = <u>sifIzlagac</u>, nazIzlagac</p> <p>PROIZVOD = <u>sifProizvod</u>, nazProizvod, cijena</p> <p>VRPROIZVOD = <u>sifVrsta</u>, nazVrsta</p> <p>DOMACI = <u>sifProizvod</u>, godCertifikata</p> <p>UVOZNI = <u>sifProizvod</u></p> <p>ZEMLJA = <u>sifZemlja</u>, <u>oznZemlja</u>, nazZemlja (K1ZEMLJA = sifZemlja; K2ZEMLJA = oznZemlja)</p>	<p>degustira = <u>sifStand</u>, <u>oib</u>, <u>sifProizvod</u></p> <p>pripada = <u>sifStand</u>, <u>sifIzlagac</u></p> <p>proizvodi = <u>sifProizv</u>, <u>sifIzlagac</u></p> <p>jeVrste = <u>sifProizv</u>, <u>sifVrsta</u></p> <p>jePorijekla = <u>sifProizv</u>, <u>sifZemlja</u></p>

8.1 (3 boda)

Za zadani ER model napišite ekvivalentni relacijski model u obliku SQL naredbi za kreiranje relacija s ugrađenim opisima primarnih, alternativnih ključeva i općih pravila integriteta. Tipovi podataka su INT za šifre, VARCHAR(50) za nazive i DATE za datume, pri čemu za nazive i datume nisu dozvoljene NULL vrijednosti.

Entitet	atributi	ključevi
E1	= sifE1, nazE1, datE1	K _{E1} = sifE1
E2	= sifE2, datE2, nazE2	K _{E2} = sifE2 datE2
E3	= sifE3, nazE3, datE3	K _{E3} = sifE3



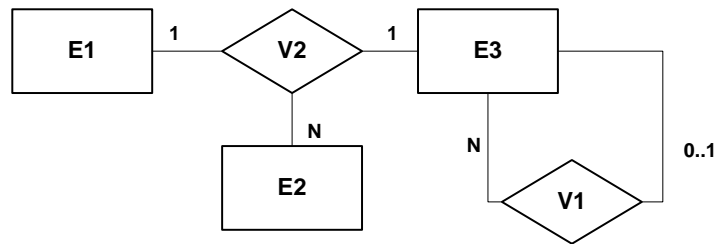
Rješenje:

```
CREATE TABLE E1(
    sifE1 INT PRIMARY KEY
    , nazE1 VARCHAR(50) NOT NULL
    , datE1 DATE NOT NULL
    , sifE1refl INT REFERENCES E1(sifE1));
CREATE TABLE E2(
    sifE2 INT
    , nazE2 VARCHAR(50) NOT NULL
    , datE2 DATE
    , PRIMARY KEY (sifE2, datE2));
CREATE TABLE E3(
    sifE3 INT PRIMARY KEY
    , nazE3 VARCHAR(50) NOT NULL
    , datE3 DATE NOT NULL);
CREATE TABLE V1(
    sifE1 INT REFERENCES E1(sifE1)
    , sifE2 INT
    , datE2 DATE
    , sifE3 INT NOT NULL REFERENCES E3(sifE3),
    PRIMARY KEY (sifE1, sifE2, datE2)
    , UNIQUE (sifE2, datE2, sifE3)
    , FOREIGN KEY (sifE2, datE2) REFERENCES E2(sifE2, datE2));
```

8.2 (3 boda)

Za zadani ER model napišite ekvivalentni relacijski model u obliku SQL naredbi za kreiranje relacija s ugrađenim opisima primarnih, alternativnih ključeva i općih pravila integriteta. Tipovi podataka su INT za šifre, VARCHAR(50) za nazive i DATE za datume, pri čemu za nazive i datume nisu dozvoljene NULL vrijednosti.

Entitet	atributi	ključevi
E1	= sifE1, nazE1, datE1	K _{E1} = sifE1
E2	= sifE2, datE2, nazE2	K _{E2} = sifE2 datE2
E3	= sifE3, nazE3, datE3	K _{E3} = sifE3

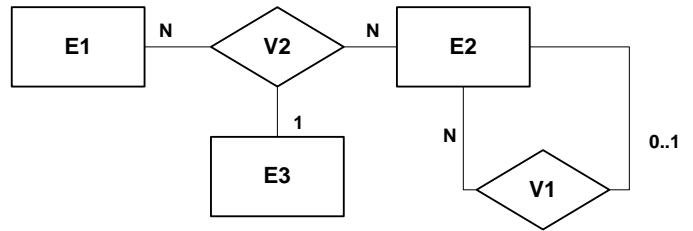


```
CREATE TABLE E1(
    sifE1 INT PRIMARY KEY
    , nazE1 VARCHAR(50) NOT NULL
    , datE1 DATE NOT NULL);
CREATE TABLE E2(
    sifE2 INT
    , nazE2 VARCHAR(50) NOT NULL
    , datE2 DATE NOT NULL
    , PRIMARY KEY (sifE2, datE2));
CREATE TABLE E3(
    sifE3 INT PRIMARY KEY
    , nazE3 VARCHAR(50) NOT NULL
    , datE3 DATE NOT NULL,
    , sifE3refl INT REFERENCES E3(sifE3));
);
CREATE TABLE V2(
    sifE1 INT REFERENCES E1(sifE1)
    , sifE2 INT
    , datE2 DATE
    , sifE3 INT NOT NULL REFERENCES E3(sifE3),
    , PRIMARY KEY (sifE1, sifE2, datE2)
    , UNIQUE (sifE2, datE2, sifE3)
    , FOREIGN KEY (sifE2, datE2) REFERENCES E2(sifE2, datE2));
```

8.3 (3 boda)

Za zadani ER model napišite ekvivalentni relacijski model u obliku SQL naredbi za kreiranje relacija s ugrađenim opisima primarnih, alternativnih ključeva i općih pravila integriteta. Tipovi podataka su INT za šifre, VARCHAR(50) za nazive i DATE za datume, pri čemu za nazive nisu dozvoljene NULL vrijednosti.

Entitet	atributi	ključevi
E1	= sifE1, nazE1, datE1	K _{E1} = sifE1
E2	= sifE2, datE2, nazE2	K _{E2} = sifE2 datE2
E3	= sifE3, nazE3, datE3	K _{E3} = sifE3

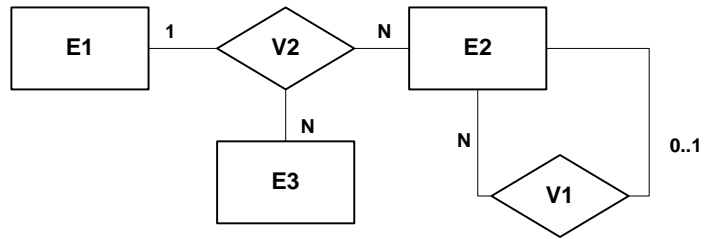


```
CREATE TABLE E1(
    sifE1 INT PRIMARY KEY
    , nazE1 VARCHAR(50) NOT NULL
    , datE1 DATE NOT NULL);
CREATE TABLE E2(
    sifE2 INT
    , nazE2 VARCHAR(50) NOT NULL
    , datE2 DATE
    , sifE2refl INT
    , datE2refl DATE NOT NULL
    , PRIMARY KEY (sifE2, datE2)
    , FOREIGN KEY (sifE2refl, datE2refl) REFERENCES E2(sifE2, datE2));
CREATE TABLE E3(
    sifE3 INT PRIMARY KEY
    , nazE3 VARCHAR(50) NOT NULL
    , datE3 DATE NOT NULL);
CREATE TABLE V2(
    sifE1 INT REFERENCES E1(sifE1)
    , sifE2 INT
    , datE2 DATE
    , sifE3 INT REFERENCES E3(sifE3),
    , PRIMARY KEY (sifE1, sifE2, datE2)
    , FOREIGN KEY (sifE2, datE2) REFERENCES E2(sifE2, datE2));
```


8.4 (3 boda)

Za zadani ER model napišite ekvivalentni relacijski model u obliku SQL naredbi za kreiranje relacija s ugrađenim opisima primarnih, alternativnih ključeva i općih pravila integriteta. Tipovi podataka su INT za šifre, VARCHAR(50) za nazive i DATE za datume, pri čemu za nazive nisu dozvoljene NULL vrijednosti.

Entitet	atributi	ključevi
E1	= sifE1, nazE1, datE1	K _{E1} = sifE1
E2	= sifE2, datE2, nazE2	K _{E2} = sifE2 datE2
E3	= sifE3, nazE3, datE3	K _{E3} = sifE3



```
CREATE TABLE E1(
    sifE1 INT PRIMARY KEY
    , nazE1 VARCHAR(50) NOT NULL
    , datE1 DATE NOT NULL);
CREATE TABLE E2(
    sifE2 INT
    , nazE2 VARCHAR(50) NOT NULL
    , datE2 DATE
    , sifE2refl INT
    , datE2refl DATE NOT NULL
    , PRIMARY KEY (sifE2, datE2)
    , FOREIGN KEY (sifE2refl, datE2refl) REFERENCES E2(sifE2, datE2));
CREATE TABLE E3(
    sifE3 INT PRIMARY KEY
    , nazE3 VARCHAR(50) NOT NULL
    , datE3 DATE NOT NULL);
CREATE TABLE V2(
    sifE1 INT REFERENCES E1(sifE1)
    , sifE2 INT
    , datE2 DATE
    , sifE3 INT REFERENCES E3(sifE3),
    , PRIMARY KEY (sifE2, datE2, sifE3)
    , FOREIGN KEY (sifE2, datE2) REFERENCES E2(sifE2, datE2));
```