

1.(3 boda)

Uz pomoć jedne SQL naredbe izvedite sljedeće: za sve osnivače timova kojima je već prošao 40. rođendan u trenutku osnivanja tima ispišite podatke o imenu, prezimenu i timu kojeg su osnovali u obliku:

"<inicijal imena>.<prezime>\_<ime tima>"

pri čemu simbol \_ označava razmak. Ispis poredati abecedno po prezimenu.

Primjer retka ispisa (podaci ne moraju odgovarati stvarnima):

Osnivac
P. Perić – Skull Crushers

```
SELECT SUBSTRING(ime FROM 1 FOR 1) || '. '
      || prezime || ' - ' || nazivTim AS Osnivac
FROM igrac join tim ON igrac.idIgrac = idOsnivac
WHERE datRodjenja + '40 YEARS'::INTERVAL < datVrOsnutak
ORDER BY prezime;
```

2. (5 bodova)

Uz pomoć jedne SQL naredbe izvedite sljedeće: za **svaku** arenu ispisati naziv i koliko je u listopadu 2019. bilo odigrano borbi u kojima je sudjelovalo više od 9 igrača (gledano oba tima ukupno). Ispis poredati abecedno po nazivu arene.

Primjer retka ispisa (podaci ne moraju odgovarati stvarnima):

NazivArena	VelikihBorbiUListopadu2019
Tony's Arena	27
Sandford Corner Arena	0

```
SELECT nazivarena, COUNT(pocetakTermin) AS VelikihBorbiUListopadu2019
FROM arena LEFT JOIN igra ON arena.idarena = igra.idarena
      LEFT JOIN borba ON
      (igra.idigra = borba.idigra AND
        EXTRACT(YEAR FROM borba.pocetakTermin) = 2019 AND
        EXTRACT(MONTH FROM borba.pocetakTermin) = 10 AND
        (SELECT COUNT(*)FROM igracTim
         WHERE igractim.idTim = borba.idTim1
          OR igractim.idTim = borba.idTim2) > 9)
GROUP BY arena.idarena, nazivarena
ORDER BY nazivarena;
```



#### 4. (5 bodova)

Zbog velikog interesa i želje da što većem krugu korisnika omoguće korištenje usluga, organizator ograničava igranje pojedine igre na jednom tjedno po timu i to bez obzira na paket igre. Tjedan planirane borbe određuje se prema planiranom početku borbe. Planirane borbe nemaju definirani završetak. Napisati niz SQL naredbi za kreiranje svih potrebnih objekata kojima će se pri unosu planirane borbe u relaciju **borba** spriječiti **upis** n-torke koji bi omogućio dvije borbe iste igre za bilo koji od timova u planiranoj borbi, u tjednu kada se borba planira.

Pri narušavanju opisanog ograničenja korisniku javiti sljedeću grešku: **'Pogreška: Jedan od timova (<tim1>,<tim2>) u <tjedan>. tjednu <godina>. već ima planiranu igru "<igra>"**.

Umjesto <tim1> i <tim2> potrebno je ispisati šifre timova koji sudjeluju u planiranoj borbi. Redni broj tjedna u godini treba se ispisati umjesto <tjedan>, a godina umjesto <godina>. Na mjestu <igra> potrebno je ispisati naziv igre.

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

Održavanje konzistentnosti navedenog pravila pri obavljanju ostalih operacija u relaciji **borba** nije potrebno implementirati.

Rješenje:

```
CREATE FUNCTION chkDuploPlanirano() RETURNS TRIGGER AS
$$
DECLARE
    borbe SMALLINT;
    nigre igra.nazivIgra%TYPE;
BEGIN
    IF (NEW.zavrsetakBorba IS NULL) THEN
        SELECT COUNT(*) INTO borbe
        FROM borba
        WHERE EXTRACT(WEEK FROM pocetakTermin)=EXTRACT(WEEK FROM
NEW.pocetakTermin)
        AND EXTRACT(YEAR FROM pocetakTermin)=EXTRACT(YEAR FROM
NEW.pocetakTermin)
        AND idIgra=NEW.idIgra
        AND (idTim1=NEW.idTim1 OR idTim2=NEW.idTim1 OR idTim1=NEW.idTim2
OR idTim2=NEW.idTim2);
        IF (borbe>0) THEN
            SELECT nazivIgra INTO nigre
            FROM igra
            WHERE idIgra=NEW.idIgra;
            RAISE EXCEPTION 'Pogreška: Jedan od timova (%,% ) u %. tjednu %.
već ima planiranu igru "%"', NEW.idTim1, NEW.idTim2, EXTRACT(WEEK FROM
NEW.pocetakTermin), EXTRACT(YEAR FROM NEW.pocetakTermin), nigre;
        END IF;
    END IF;
    RETURN NEW;
END;
$$ language plpgsql;

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

5. (6 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	David	Mendez	04.06.1994	dmendez
...	...	...	...	...

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;
```

a) Napisati SQL naredbe kojima će administrator sustava korisniku *dmendez*, osnivaču nekih timova, dodijeliti ovlasti za:

- uspostavu korisničke sjednice (lozinku postaviti na *dmendezPwd*),
- spajanje na bazu *tagfight*,
- pristup objektima sadržanima u shemi *public* bez mogućnosti kreiranja novih objekata,
- pregled svih podataka u tablici *tim* uz mogućnost dodjeljivanja tih istih ovlasti ostalim korisnicima,
- izmjenu naziva arene u tablici *arena*,
- pregled i izmjenu borbi u kojima je pobijedio tim kojem je on (odnosno bilo koji igrač koji je ostvario SQL-sjednicu) osnivač.

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

```
CREATE USER dmendez WITH PASSWORD 'dmendezPwd';
```

```
--ili CREATE ROLE dmendez WITH LOGIN PASSWORD 'dmendezPwd';
```

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

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

```
GRANT SELECT ON public.tim TO dmendez WITH GRANT OPTION;
```

```
GRANT UPDATE ON public.arena(nazivArena) TO dmendez;
```

```

CREATE VIEW borbePobjTimJaOsnivac AS
SELECT *
  FROM borba
 WHERE EXISTS (SELECT *
                FROM tim JOIN igrac on idIgrac = idOsnivac
                WHERE borba.idTimPobjeda = tim.idTim
 AND korIme = SESSION_USER)
-- može i CURRENT_USER ako nema CREATE ROLE
WITH CHECK OPTION;

GRANT SELECT, UPDATE ON public.borbePobjTimJaOsnivac TO dmendez;

```

- b) Neka je korisnik *dmendez* dodijelio ovlast pregleda podataka tablice *tim* korisnicima *jmorris* i *mwest*. Napišite **jednu** naredbu kojom će administrator ukinuti dozvole pregleda podataka tablice *tim* korisnicima *dmendez*, *jmorris* i *mwest*.

```

REVOKE SELECT ON tim FROM dmendez CASCADE;

```

#### 6. (3 boda)

Relacija **tim** (idTim, nazivTim, datVrOsnutak, idOsnivac) sadrži n-torke sa sljedećim vrijednostima atributa idTim: 1, 5, 8, 9, 11, 17, 52, 53, 65, 67, 70, 71, 75, 79, 81, 85, 90, 92, 95, 96, 97.

- Nacrtajte B<sup>+</sup> stablo reda 5 za atribut **idTim** tako da popunjenost stabla bude **maksimalna**.
- Za svaku od sljedećih dviju SQL naredbi, u slučaju prethodno nacrtanog stabla, napišite koja će se metoda koristiti za pristup podacima. Dodatno napišite koliko će biti potrebno obaviti U/I operacija kako bi se izvela svaka od navedenih naredbi.

1) SELECT COUNT(\*) FROM tim WHERE idTim BETWEEN 17 AND 79;

Pretpostavite da jedna U/I operacija dohvaća jedan čvor.

2) SELECT \* FROM tim WHERE idTim = 79;

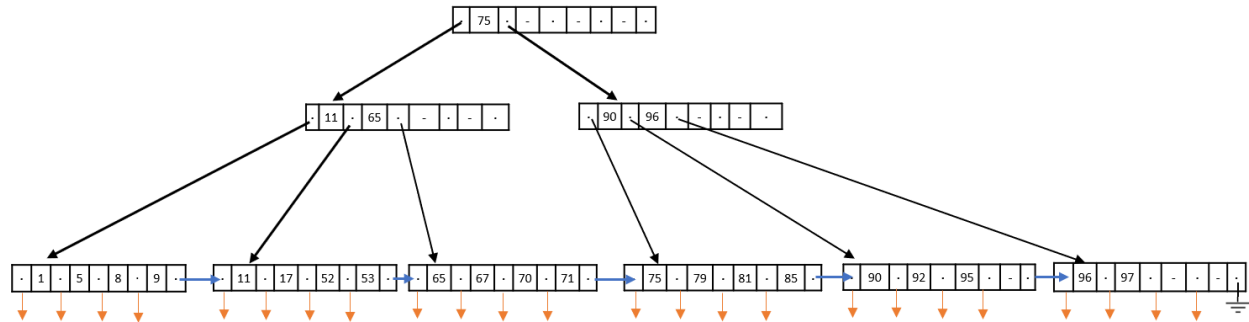
Kao rješenje a) dijela zadatka nacrtajte na papiru. **Na crtežu jasno naznačite mjesta predviđena za kazaljke i ključeve.**

Rješenje b) dijela zadatka unesite u prostor za slobodni unos teksta ispod teksta zadatka

### Rješenje:

a)

	MIN	MAX
Korijen	2	n=5
Interni čvor	$\lceil n/2 \rceil = 3$	n=5
List	$\lceil (n-1)/2 \rceil = 2$	n-1=4



b)

1) Korištenjem indeksa će se dohvatiti list u kojem se nalazi zapis s ključem 17 i potom će se pomoću kazaljke koja pokazuje na sljedeći list dohvaćati susjedni listovi dok se ne dohvati list u kojem se nalazi zapis s ključem 79.

3 U/I za dohvat zapisa s ključem 17 (*index-only scan*) + 2 U/I za dohvat sljedeća 2 lista = 5

2) Koristi se *index scan*. Potrebno: 3 U/I za dohvat lista s ključem 79 + 1 U/I za dohvat bloka s podacima - > ukupno 4 U/I operacije

7. (5 bodova)

Zadana je relacijska shema KvalitetaVode gdje se evidentiraju podatci o ispitivanju kvalitete vode na plažama. Atributi relacijske sheme KvalitetaVode su:

sifPlaza	šifra plaže - jedinstveni identifikator plaže
nazPlaza	naziv plaže
postBr	poštanski broj mjesta u kojem se nalazi plaža
nazMjesto	naziv mjesta
datUzorak	datum uzimanja uzorka vode
oznRazinaKvalitete	oznaka razine kvalitete vode
opisRazinaKvalitete	opis razine kvalitete vode
sifIspitivac	šifra ispitivača
imelspitivac	ime ispitivača
prezIspitivac	prezime ispitivača
sifLaboratorij	šifra laboratorija u kojem je obavljena analiza
nazLaboratorij	naziv laboratorija

Vrijede sljedeća pravila: na jednoj plaži jednoga dana uzima se samo jedan uzorak vode te jedan ispitivač radi u jednom laboratoriju.

Odrediti ključ relacijske sheme KvalitetaVode 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 KVALITETA\_VODE.

Rezultate upišite u prostor za odgovore ispod teksta zadatka.

Rješenje:

1NF

KV\_1 = sifPlaza, datUzorak, nazPlaza, postBr, nazMjesto, oznRazinaKvalitete, opisRazinaKvalitete, sifIspitivac, imelspitivac, prezIspitivac, sifLaboratorij, nazLaboratorij

2NF

KV\_1 = sifPlaza, nazPlaza, postBr, nazMjesto

KV\_2 = sifPlaza, datUzorak, oznRazinaKvalitete, opisRazinaKvalitete, sifIspitivac, imelspitivac, prezIspitivac, sifLaboratorij, nazLaboratorij

3NF

PLAZA = sifPlaza, nazPlaza, postBr

MJESTO = postBr, nazMjesto

RAZINAKVAL = oznRazinaKvalitete, opisRazinaKvalitete

KV\_3 = sifIspitivac, imelspitivac, prezIspitivac, sifLaboratorij, nazLaboratorij

ISPITIVAC = sifIspitivac, imelspitivac, prezIspitivac, sifLaboratorij

LABORATORIJ = sifLaboratorij, nazLaboratorij

UZORAK = sifPlaza, datUzorak, oznRazinaKvalitete, sifIspitivac

KVALITETA\_VODE = PLAZA, MJESTO, RAZINAKVAL, ISPITIVAC, LABORATORIJ, UZORAK



8. (5 bodova)

Navedite po jedan primjer za a) odnosno b) dio zadatka, koji koristi isključivo podatke iz tablice **arena**, kojima ćete pokazati da:

- a) **čitanje ne blokira pisanje** kada Postgres za upravljanje istodobnim pristupom koristi isključivo Multiverzijski protokol upravljanja istodobnim pristupom – MVCC
- b) **čitanje blokira pisanje** kada Postgres koristi eksplicitno postavljanje ključeva (elemente protokola zasnovanog na zaključavanju)

Svaki od dva primjera treba sadržavati minimalan broj transakcija koje se paralelno odvijaju i minimalan broj SQL naredbi potrebnih za demonstraciju ponašanja. Koristite razine izolacije koje smatrate prikladnima. Naznačite:

- početni sadržaj n-torke/n-torki koje u primjerima koristite
- redoslijed izvršavanja naredbi u transakcijama (globalni redoslijed za sve transakcije koje u primjeru koristite)
- objasnite ulogu svake SQL naredbe koju u primjerima koristite, obavezno navedite za svaku naredbu traži li ona i dodjeljuje li joj se ključ i koje vrste te koliko taj ključ traje
- objasnite zbog čega u primjeru pod a) čitanje ne blokira pisanje, odnosno zbog čega u primjeru pod b) čitanje blokira pisanje.

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

## Rješenje

Početno stanje n-torke koja se koristi u primjeru pod a) i pod b)

idArena	nazivArena
10	Rose Arena

Obje transakcije koriste defaultnu razinu izolacije – READ COMMITTED, premda u ovim primjerima razina izolacije nije važna jer bi ponašanje bilo jednako i kod bilo koje druge razine.

a)

T1		T2	
{1}	BEGIN TRANSACTION;	{2}	BEGIN TRANSACTION;
{3}	SELECT * FROM arena WHERE idArena = 10;	{4}	UPDATE arena SET nazivArena = 'Primrose Arena' WHERE idArena = 10;
{6}	COMMIT TRANSACTION;	{5}	COMMIT TRANSACTION;

{3} demonstrira čitanje n-torke. Pri tom ne traži i ne postavlja nikakav ključ i tu istu n-torku može čitati ili mijenjati bilo koja druga transakcija.

{4} demonstrira pisanje (mijenja naziv arene s idArena = 10) iste n-torke koju je T1 netom prije pročitala. Pri tom se postavlja ključ za pisanje koji traje do kraja transakcije, odnosno do obavljanja naredbe {5}.

b)

T1		T2	
{1}	BEGIN TRANSACTION;	{2}	BEGIN TRANSACTION;
{3}	SELECT * FROM arena WHERE idArena = 10 <b>FOR SHARE;</b>	{4}	UPDATE arena SET nazivArena = 'Primrose Arena' WHERE idArena = 10;
{6}	COMMIT TRANSACTION;	{5}	COMMIT TRANSACTION;

{3} traži i **postavlja** ključ za **čitanje** (shared lock) nad n-torkom sa šifrom 10. Ključ traje do kraja transakcije.

{4} **pokušava postaviti** ključ za **pisanje** nad n-torkom sa šifrom 10, ali ne uspijeva jer T1 već ima ključ za čitanje (nekompatibilan s ključem za pisanje) nad tom istom n-torkom. Pisanje T2 je blokirano čitanjem T1.

Zbog nekompatibilnosti ključeva – ključ za pisanje nije kompatibilan s ključem za čitanje, T2 ne uspijeva izmijeniti n-torku koju je T1 pročitala. Odnosno, postavljanjem ključeva koji traju do kraja transakcije postiže se da čitanje blokira pisanje.

Zajedničko u oba primjera je da se kod izmjene postavlja ključ za pisanje na objekt koji se mijenja i ključ traje do kraja transakcije.

Razlika između čistog MVCC protokola i implementacije koja koristi eksplicitno zaključavanje kod čitanja je što MVCC kod čitanja ne zaključava objekt koji pokušava pročitati dok u drugom slučaju se taj ključ eksplicitno traži i postavlja. Zbog toga u trenutku kad se pokuša obaviti pisanje i postaviti ekskluzivni ključ u primjeru pod a) je to moguće jer nad tim objektom nema nikakvog ključa, dok u primjeru pod b) to nije moguće jer postoji ključ za čitanje nad tim objektom.

9. (4 boda)

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

$N(\text{igracTim}) = 12\ 000$ $V(\text{idTim}, \text{igracTim}) = 1200$	$N(\text{tim}) = 1200$ $V(\text{datVrOsnutak}, \text{tim}) = 300$	$N(\text{borba}) = 60\ 000$ $V(\text{idIgra}, \text{borba}) = 100$ $V(\text{idTimPobjeda}, \text{borba}) = 150$
---	--	---

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

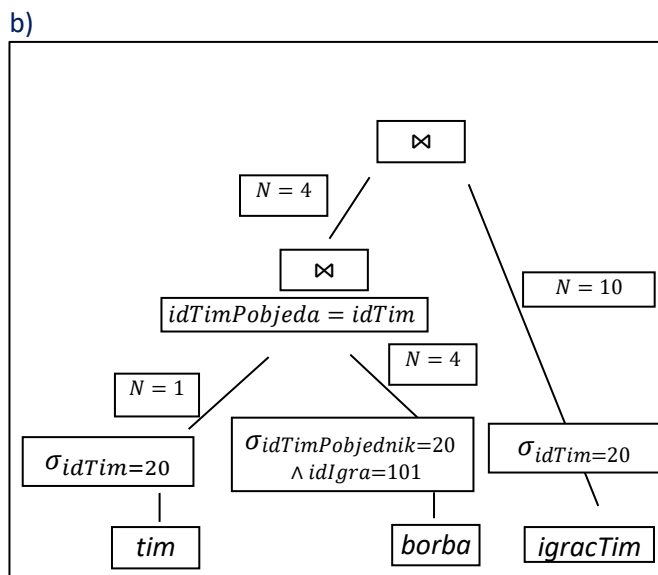
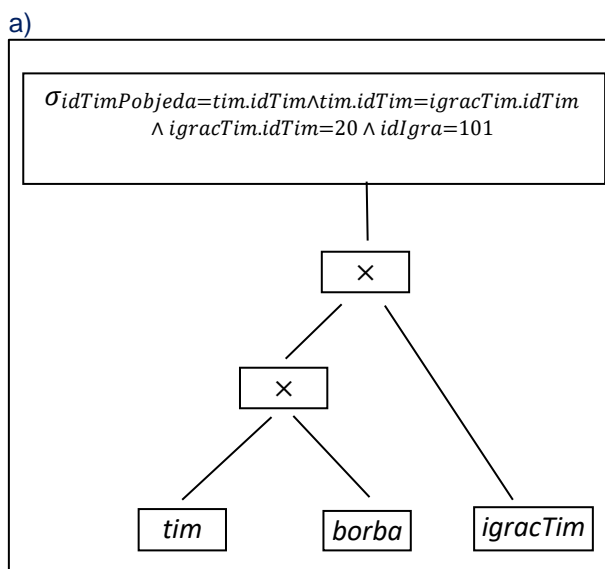
Izvodi se upit:

```
SELECT *
FROM tim, borba, igracTim
WHERE igracTim.idTim = tim.idTim
AND borba.idTimPobjeda = tim.idTim
AND igracTim.idTim = 20
AND idIgra = 101;
```

- Nacrtajte (na papir) stablo upita za **početni plan** izvođenja upita pri čemu je redoslijed spajanja tablica određen redoslijedom kojim su tablice navedene u FROM dijelu SELECT naredbe.
- Nacrtajte (na papir) 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. Ako postoji više „optimalnih“ redoslijeda spajanja, odlučite se za jedan i nacrtajte ga.

**Napomena:** Postupak i izračun za procjenu broja n-torki **unesite u prostor za slobodni unos teksta ispod teksta zadatka.**

Rj.



Prvo se spaja borba s tim ili s igracTim.

Primijetiti:

```
WHERE igracTim.idTim = tim.idTim  
AND borba.idTimPobjeda = tim.idTim  
AND igracTim.idTim = 20
```

potiskivanje selekcije (uvjet da je id tima jednak 20) – na **sve tri tablice**

potiskivanje selekcije idIgra = 101 – na **borba**

- **tim.idTim = 20:**  $N(\text{tim}') = 1$  (radi se o PK)
- **igracTim.idTim = 20:**  $N(\text{igracTim}') = N(\text{igracTim}) / V(\text{idTim}, \text{igracTim}) = 12\,000 / 1200 = 10$

- **borba.idTimPobjeda = 20 AND idIgra = 101:**

$$N(\text{borba}') = (N(\text{borba}) / V(\text{idTimPobjeda}, \text{borba})) / V(\text{idIgra}, \text{borba}) = 60\,000 / (150 * 100) = 4$$

Međurezultati spajanja:

<b>tim' + borba"</b>	<b>(fk u borba) =&gt; N(borba') = 4</b>
<b>tim' + igracTim</b>	<b>(fk u igracTim) =&gt; N(igracTim) = 10</b>
<b>borba" + igracTim</b>	<b>(fk u borba) =&gt; N(borba') = 4</b>

10. (8 bodova)

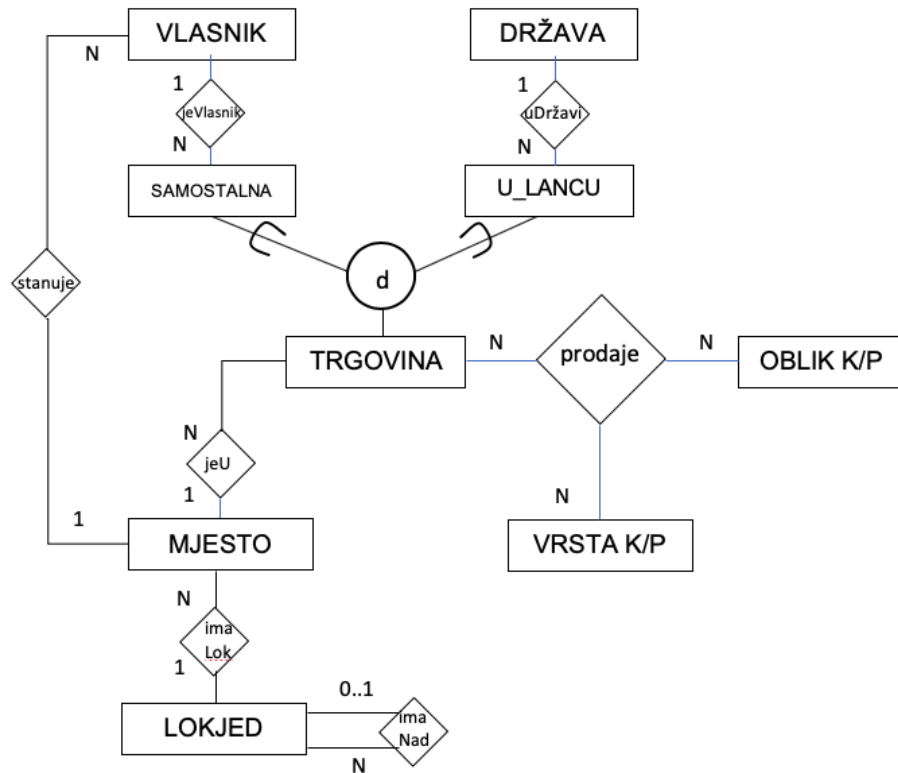
Oblikovati ER model baze podataka za analizu cijena kruha i peciva u različitim trgovinama. Evidentiraju se podatci o trgovinama, oblicima kruha/peciva (štruca, francuz, okrugli, žemlja, kajzerica, pletenica, ...) i vrstama kruha/peciva (bijeli, polubijeli, crni, raženi, graham, ...). Trgovina je opisana šifrom, nazivom i adresom (mjesto, ulica i kućni broj). Za mjesto se evidentira šifra mjesta, poštanski broj i naziv. Također se evidentira najuža lokalna jedinica kojoj mjesto pripada. Za lokalnu jedinicu se evidentira šifra i naziv. Lokalne jedinice su hijerarhijski organizirane (svaka lokalna jedinica može imati najviše jednu nadređenu i po volji mnogo podređenih jedinica). Trgovina može biti samostalna ili može pripadati trgovačkom lancu. Za samostalne trgovine treba evidentirati jednog vlasnika (oib, ime, prezime, mjesto, ulica i kućni broj) i oblik vlasništva (niz znakova, npr. obrt, d.o.o. i slično). Jedna osoba može biti vlasnikom više trgovina. Za trgovinu koja pripada trgovačkom lancu evidentira se oblik (niz znakova, npr. mala, maxi, super) i država sjedišta trgovačkog lanca. Za državu se evidentira šifra, jedinstvena ISO oznaka i jedinstveni naziv države. Za oblik i vrstu kruha/peciva evidentiraju se šifre i nazivi. U jednoj trgovini jedna vrsta kruha/peciva može postojati u različitim oblicima, jedan oblik kruha/peciva u istoj trgovini može biti različitih vrsta. Kruh/pecivo istih vrsta i istog oblika vrsta u različitim trgovinama mogu imati različite težine i cijene po kilogramu.

Potrebno je nacrtati ER dijagram (bez atributa) i označiti spojnosti entiteta u vezama. Sve entitete, osim slabih, opisati **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:



TRGOVINA = <u>sifTrgovina</u> , nazTrgovina	VRSTA_K/P = <u>sifVrsta</u> , nazVrsta
MJESTO = <u>sifMjesto</u> , postBr, nazMjesto	jeVlasnik = <u>sifTrgovina</u> , oib, oblik_vlasnistvo
SAMOSTALNA = <u>sifTrgovina</u>	stanuje = <u>sifVlasnik</u> , sifMjesto, ulica, kbr
U_LANCU = <u>sifTrgovina</u> , oblik	je = <u>sifTrgovina</u> , sifMjesto, ulica, kbr
VLASNIK = <u>oib</u> , ime, prezime	imaLok = <u>sifMjesto</u> , sifLokjed
DRZAVA = <u>sifDrzava</u> , ISOoznaka, nazDrzava	prodaje = <u>sifTrgovina</u> , <u>sifOblik</u> , <u>sifVrsta</u> , tezina, cijena_po_kg
K1 = sifDrzava; K2 = ISOoznaka; K3 = nazDrzava	
LOKJED = <u>sifLokjed</u> , nazLokjed	imaNad = <u>sifLokjed</u> , sif nadLokjed
OBLIK_K/P = <u>sifOblik</u> , nazOblik	uDrzavi = <u>sifTrgovina</u> , sifDrzava