

Jesenski ispitni rok iz Baza podataka (50 bodova)

6. rujna 2019.

Zadaci 1 - 7 odnose se na bazu podataka **festivali** koja se koristi za pohranu podataka o festivalima koji se organiziraju diljem Hrvatske. Različiti izvođači (**izvodac**) nastupaju na festivalima (**festival**) u različitim mjestima (**mjesto**). Za festivale se evidentiraju podaci o održavanju festivala (početak, mjesto, cijena, organizator i trajanje) (**raspored**). Organizatori (**organizator**) festivala raspoređuju nastupe izvođača (**nastup**) po festivalima. Festivali mogu trajati više dana (**brDana**), a za svaki se dan festivala evidentiraju izvođači i njihov raspored nastupa – redni broj nastupa (**rbrNastup**). Svakog dana (ovisno koliko dugo festival traje), nastupi započinju u isto vrijeme. Cijena festivala izražena je u eurima.

Na slici **nisu** prikazane sve n-torke sadržane u relacijama.

izvodac		mjesto		festival		raspored					
sifIzv	nazivIzv	pbr	nazivMj	sifFest	nazivFest	sifFest	datVrPoc	pbr	cijena	sifOrg	brDana
101	The Cure	52470	Umag	1	Sea star	1	2019-05-24 20:00:00	52470	70	1	3
102	Soffi Tukker	10000	Zagreb	2	In Music	1	2017-06-24 22:30:00	52470	70	1	3
103	Carl Cox	21000	Split	3	Ultra Europe	3	2019-07-12 20:30:00	21000	169	1	3
104	M.O.R.T	33515	Orahovica	4	Ferragosto	4	2019-08-02 21:00:00	33515	50	3	2
105	Vojko V			5	Zagreb Beer	5	2018-05-20 19:00:00	10000	0	2	5

nastup					organizator			
sifFest	datVrPoc	sifIzv	rbrDan	rbrNastup	sifOrg	ime	prezime	korIme
1	2019-05-24 20:00:00	101	1	1	1	Joe	Basic	JoBa
1	2019-05-24 20:00:00	102	1	2	2	Ante	Anic	AnAn
3	2019-07-12 20:30:00	103	1	1	3	Jozo	Josic	JoJo
3	2019-07-12 20:30:00	104	2	1				
3	2019-07-12 20:30:00	101	2	2				

- (5 bodova) Ispišite naziv, vrijeme početka te cijenu za sve festivale koji se održavaju u Zagrebu. Osim cijene izražene brojem, potrebno je ispisati i tekst koji ukazuje na cijenu festivala – 'skupo' ukoliko je cijena festivala skuplja od 150 eura, 'srednje' ukoliko je cijena festivala između 50 i 150 eura, te 'jeftino' ukoliko je cijena festivala ispod 50 eura. Ispišite i nazive izvođača čiji nastupi započinju drugog dana iza 22:00h, dok za ostale, naziv izvođača treba biti null.
Napomena: Pretpostavite da svaki nastup traje točno 1 sat i 30 minuta, te da između nastupa nema pauze.
- (6 bodova) Ispišite ime i prezime organizatora, naziv mjesta i broj besplatnih festivala koje je on organizirao u tom mjestu u posljednjih godinu dana. U obzir uzmite samo one organizatore koji su u pretposljednje dvije godine u odnosu na trenutak izvršavanja upita (ako se upit izvršava danas u obzir je potrebno uzeti vremenski period od 06.09.2016. do 06.09.2018.) organizirali 5 ili više besplatnih festivala (sveukupno - neovisno o mjestu). Rezultate poredajte uzlazno prema broju besplatnih festivala organiziranih u posljednjih godinu dana.
- (6 bodova) Napišite niz SQL naredbi za kreiranje svih potrebnih objekata kojima će se pri unosu n-torke u relaciju **nastup** osigurati sljedeće - potrebno je dozvoliti umetanje novog nastupa samo za festivale koji ove godine još nisu započeli. Ako vrijeme održavanja festivala još nije poznato dojavite pogrešku – „Podatci o festivalu nisu evidentirani“. Potrebno je paziti i na ispravnost unosa rednog broja dana za pojedini nastup – nastup se smije dodati samo na dane kada se festival i održava. Za pokušaj pogrešnog umetanja rednog broja dana festivala dojavite pogrešku „Festival ne traje zadani broj dana.“. Ako je izvođač već zabilježen za nastup koji se održava dan prije izvođenja upita ili dan poslije izvođenja upita na nekom drugom festivalu, dojavite grešku „Nastup izvođača već je evidentiran na drugom festivalu“. U suprotnom, nastup je potrebno dodati s rednim brojem za 1 većim od do tada najvećega. Ako se pokuša umetnuti netočan redni broj nastupa dojavite grešku: „Krivi redni broj nastupa.“.
- (7 bodova) Ako je vlasnik baze podataka izveo sljedeću naredbu `REVOKE ALL ON SCHEMA public FROM PUBLIC;`, napišite sve potrebne naredbe kojima će vlasnik baze podataka kreirati korisnika **bpadmin** s mogućnošću uspostavljanja SQL sjednice, kreiranja baza podataka i novih korisnika (koristiti autorizaciju pomoću lozinke). Zatim napišite sve potrebne naredbe kojima će korisnik **bpadmin** stvoriti bazu podatka **festivali** s dvije tablice (izvođač i festival sa slike 1) te dva korisnika **josipa** i **david**. Svim trenutnim i budućim korisnicima treba ukinuti dozvolu za spajanje na novokreiranu bazu podataka (osim korisnicima **david** i **josipa**). Korisniku **josipa** treba dozvoliti pregled svih festivala, te izmjenu naziva festivala. Korisniku **david** treba dozvoliti pregled, izmjenu, unos i brisanje podataka u tablici izvođač.

5. (3 boda) Za zadani izraz relacijske algebre:

$$\pi_{pbr, nazivMj} (mjesto) \setminus \pi_{pbr, nazivMj} (raspored \bowtie mjesto)$$

- opišite riječima što je rezultat izraza
- napišite SQL upit koji vraća ekvivalentan rezultat.

6. (4 boda) Nad relacijom **izvodac** baze podataka **festivali** često se obavljaju upiti:

```
SELECT naziv FROM izvodac WHERE naziv > 'The';
SELECT * FROM izvodac ORDER BY naziv DESC;
SELECT * FROM izvodac WHERE naziv = 'The Cure';
```

- Napišite **minimalan** broj naredbi kojima će se ubrzati izvođenje navedenih upita.
- Ako relacija sadrži 1000 izvođača i kreirano je B⁺ stablo reda 5, koliko je u **najboljem** slučaju potrebno U/I operacija da bi se dohvatio blok s podacima? Napišite detaljan postupak izračuna.

7. (6 bodova) a) Napišite SQL naredbe kojima će se odrediti vrijednosti N(izvodac) i V(ime, organizator).
b) U bazi podataka **festivali** obavlja se sljedeća SQL naredba:

```
SELECT * FROM organizator, mjesto, raspored
WHERE cijena < 90 AND brDana = 5 AND raspored.pbr = 45000
AND prezime = 'Anic'
AND raspored.sifOrganizator = organizator.sifOrganizator
AND raspored.pbr = mjesto.pbr;
```

Optimizator upita raspolaže sljedećim statističkim podacima:

N(raspored) = 1200	N(organizator) = 120	N(mjesto) = 6 000
V(sifFest, raspored) = 90	V(prezime, organizator) = 60	
V(cijena, raspored) = 60		
V(brDana, raspored) = 8		
V(pbr, raspored) = 10		

Nacrtajte **inicijalno** i **konačno** stablo upita nakon provedene heurističke optimizacije. Pretpostavite da je redoslijed spajanja relacija određen redoslijedom kojim su relacije navedene u FROM dijelu SELECT naredbe. Redoslijed spajanja relacija prilikom crtanja konačnog stabla upita odrediti temeljem procjene broja n-torki u rezultatima spajanja. Navesti izraze prema kojima je obavljena procjena broja n-torki u svim međurezultatima. U stablu upita naznačiti očekivani broj n-torki u svim međurezultatima.

8. (6 bodova) Kinološki savez evidentira podatke o psima, njihovim vlasnicima i rezultatima koje postižu na natjecanjima. Zadana je relacijska shema REZULTATI:

sifVlasnik, ime, prezime, ulicaKbr, pbr, imeGrad, sifZivotinja, imeZivotinja, nazNatjecanje, godina, certifikat, rezultat

Za vlasnika se evidentiraju ulica, kućni broj i mjesto stanovanja. Vlasnik može imati više životinja i s njima sudjelovati na više natjecanja. Životinja pripada isključivo jednom vlasniku. Istoimena natjecanja održavaju se isključivo u različitim godinama (ne postoje dva istoimena natjecanja u istoj godini). Svako natjecanje dobiva određeni certifikat prestiža, pri čemu se certifikat za pojedino natjecanje može mijenjati iz godine u godinu.

- Normalizirajte relaciju postepeno na 1NF, 2NF i 3NF.
- Navedite SQL upit kojim bi se provjerilo može li nad **početnom** relacijom REZULTATI vrijediti funkcijska zavisnost *ime, prezime* → *ulicaKbr*. Objasnite kako se donosi zaključak temeljem rezultata SQL upita.

9. (7 bodova) Glazbeni servis *firstFM* svojim korisnicima omogućava praćenje podataka o glazbi koju slušaju na pojedinim platformama. U bazi podataka evidentiraju se podatci o korisniku – jedinstveno korisničko ime i e-mail adresa. Za pjesme se evidentira šifra, naziv i trajanje. Pjesmu izvodi više izvođača, za koje se evidentiraju šifra i ime. U bazi podataka evidentira se i platforma preko koje je korisnik slušao pjesmu, pri čemu se razlikuju radio- i web-platforme. Za platformu se evidentiraju šifra i naziv, a dodatno se za radio-platforme evidentiraju maksimalni domet u km (najveća udaljenost na kojoj se čuju) i površina u km² koju pokrivaju, a za web-platformu URL početne stranice. Prilikom slušanja jedne pjesme od strane jednog korisnika, u bazi podataka evidentira se jedinstveni serijski broj slušanja, vremenski trenutak (datum i vrijeme) početka slušanja pjesme te preko koje je platforme je slušanje obavljeno.

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

Rješenja:

1.

```
SELECT festival.nazivFest, raspored.datVrpoc, cijena, izvodac.nazivIzv,
CASE   WHEN cijena > 150 THEN 'skupo'
        WHEN cijena BETWEEN 50 AND 150 THEN 'srednje'
        ELSE 'jeftino'
END
FROM festival JOIN raspored ON festival.sifFest = raspored.sifFest
        JOIN mjesto on raspored.pbr = mjesto.pbr
        LEFT JOIN nastup ON nastup.sifFest = raspored.sifFest
        AND nastup.datVrPoc = raspored.datVrPoc
        AND nastup.rbrDan = 2
        AND EXTRACT(HOUR FROM (nastup.datVrPoc
        + (rbrStart-1) * '1 hour 30 minute'::INTERVAL)) >= 22
        LEFT JOIN izvodac ON nastup.sifIzv = izvodac.sifIzv
WHERE mjesto.nazivMj = 'Zagreb';
```
2.

```
SELECT ime, prezime, nazivMj, COUNT(raspored.sifFest) AS brFest
FROM organizator NATURAL JOIN raspored NATURAL JOIN mjesto
WHERE raspored.cijena = 0
        AND raspored.datVrPoc > CURRENT_TIMESTAMP - '1 year'::INTERVAL
        AND organizator.sifOrg IN
        ( SELECT organizator.sifOrg
          FROM organizator NATURAL JOIN raspored
        WHERE cijena = 0
              AND datVrPoc > CURRENT_TIMESTAMP - '3 years'::INTERVAL
              AND datVrPoc < CURRENT_TIMESTAMP - '1 years'::INTERVAL
        GROUP BY organizator.sifOrg
        HAVING COUNT(*) >= 5 )
GROUP BY organizator.sifOrg, mjesto.pbr
ORDER BY brFest;
```

alternativno rješenje korištenjem koreliranog podupita:

```
SELECT ime, prezime, nazivMj, COUNT(raspored.sifFest) AS brFest
FROM organizator NATURAL JOIN raspored
        NATURAL JOIN mjesto
WHERE raspored.cijena = 0
        AND raspored.datVrPoc > CURRENT_TIMESTAMP - '1 year'::INTERVAL
        AND 5 <= ( SELECT COUNT(*) FROM raspored r
                  WHERE r.sifOrg = org.sifOrg          -- korelacija
                    AND cijena = 0
                    AND datVrPoc > CURRENT_TIMESTAMP - '3 years'::INTERVAL
                    AND datVrPoc < CURRENT_TIMESTAMP - '1 year'::INTERVAL
                  )
GROUP BY organizator.sifOrg, mjesto.pbr
ORDER BY brFest;
```

3.

```
CREATE OR REPLACE FUNCTION chkNastup() RETURNS trigger AS $$
BEGIN
    IF ( NEW.datVrPoc < CURRENT_TIMESTAMP )
    THEN RAISE EXCEPTION 'Festival je zapoceo';
    ELSIF ( ( SELECT brDana FROM raspored
              WHERE raspored.sifFest = NEW.sifFest
              AND raspored.datVrPoc = NEW.datVrPoc) IS NULL )
    THEN RAISE EXCEPTION 'Pogreška: Podatci o festivalu nisu evidentirani';
    ELSIF ( NEW.rbrDan < 1      or
            NEW.rbrDan > (SELECT brDan from raspored
                          where sifFest = new.sifFest
                          and datVrPoc = new.datVrPoc))
    THEN RAISE EXCEPTION 'Pogreška: Festival ne traje zadani broj dana';
    ELSIF ( NEW.sifIzv IN
            (SELECT sifIzv from raspored natural join nastup
             where ((datvrpoc+( rbrDan *'1 day'::INTERVAL
                               - '1 day'::INTERVAL))::DATE = CURRENT_DATE - 1)
                  OR((datvrpoc +(rbrDan *'1 day'::INTERVAL
                               - '1 day'::INTERVAL))::DATE = CURRENT_DATE+ 1)))
    THEN RAISE EXCEPTION 'Pogreška: Nastup izvodaca vec je evidentiran na
                          drugom festivalu';
    ELSIF ( new.rbrNastup!= (SELECT max(rbrNastup) FROM nastup
                             WHERE sifFest = new.sifFest
                             and datVrPoc = new.datVrPoc) + 1)
    THEN RAISE EXCEPTION 'Pogreška: Krivi startni broj nastupa';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER insNastup
BEFORE INSERT ON nastup
FOR EACH ROW
EXECUTE PROCEDURE chkNastup();
```

4.

```
-- ovo izvodi supeuser:
CREATE USER bpadmin WITH CREATEDB CREATEUSER PASSWORD 'bpadminPwd';

-- ovo izvodi bpadmin:
CREATE DATABASE festivali;

CREATE TABLE izvodac (
    sifIzv int NOT NULL,
    nazivIzv varchar(255) NOT NULL,
    PRIMARY KEY (sifIzv)
);

CREATE TABLE festival (
    sifFest int NOT NULL,
    nazivFest varchar(255) NOT NULL,
    PRIMARY KEY (sifFest)
);

CREATE USER david;
CREATE USER josipa;

REVOKE CONNECT ON DATABASE festivali FROM PUBLIC;
GRANT CONNECT ON DATABASE festivali TO david;
GRANT CONNECT ON DATABASE festivali TO josipa;

GRANT USAGE ON SCHEMA public TO david;
GRANT USAGE ON SCHEMA public TO josipa;

GRANT SELECT, INSERT , UPDATE, DELETE ON izvodac TO david;
GRANT SELECT on festival to josipa;
GRANT UPDATE (naziv) ON festival to josipa;
```

5. (3 boda)

a) rezultat izraza su poštanski brojevi i nazivi mjesta u kojima se nikad nije održao niti jedan festival

b)

```
SELECT pbr, nazivMj FROM mjesto
WHERE pbr NOT IN (SELECT pbr FROM raspored);
```

 -- može i EXCEPT

6. (4 boda)

a)

```
CREATE INDEX idx_izvodaci_naziv ON izvodaci (naziv);
```

b) $n = 5$, $m = 1000$; najbolji slučaj = maksimalna popunjenost

korijen: $n = 5$
čvor: $n = 5$
list: $n - 1 = 4$

1 razina	(K)	5	stane max 5 zapisa	→ premalo
2 razine	(KL)	$5 \cdot 4 = 20$	stane max 20 zapisa	→ - -
3 razine	(KČL)	$5 \cdot 5 \cdot 4 = 100$	stane max 100 zapisa	→ - -
4 razine	(KČČL)	$5 \cdot 5 \cdot 5 \cdot 4 = 500$	stane max 500 zapisa	→ - -
5 razina	(KČČČL)	$5 \cdot 5 \cdot 5 \cdot 5 \cdot 4 = 2500$	stane max 2500 zapisa	→ dovoljno!

Budući da stablo ima 5 razina, za dohvat podataka je potrebno 6 U/I operacija.

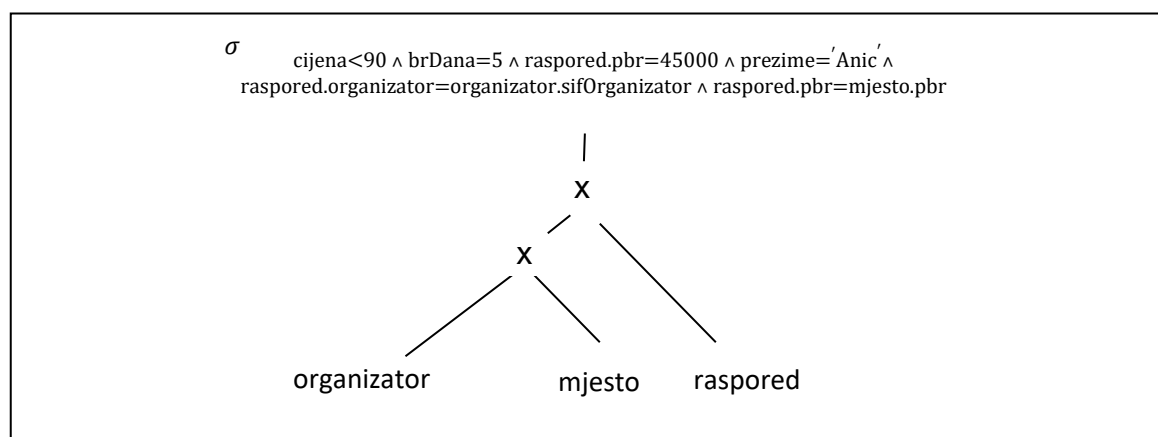
7. (6 bodova)

a)

```
SELECT COUNT(*) FROM izvodac;
SELECT COUNT(DISTINCT ime) FROM organizator;
```

b)

Inicijalno stablo:



Potiskivanje selekcije – procjena međurezultata:

$cijena < 90 \rightarrow raspored$
 $brDana = 5 \rightarrow raspored$
 $pbr = 45000 \rightarrow raspored, mjesto$
 $prezime = Anic \rightarrow organizator$

$$N(raspored') = (N(raspored)) / (3 * V(brDana, raspored) * V(pbr, raspored)) = N(raspored) / (3 * 8 * 10) = 1200 / 240 = 5$$

$$N(mjesto) = 1 \text{ (pbr je ključ)}$$

$$N(organizator') = N(organizator) / V(prezime, organizator) = 120 / 60 = 2$$

$$N(raspored') = 5$$

$$N(mjesto) = 1$$

$$N(organizator') = 2$$

Procjena međurezultata za različite redoslijede spajanja:

$organizator \bowtie mjesto \rightarrow$ nema zajedničkih atributa, izvodi se Kartezijev produkt

$$N(organizator \bowtie mjesto) = N(organizator') * N(mjesto) = 2 * 1 = 2$$

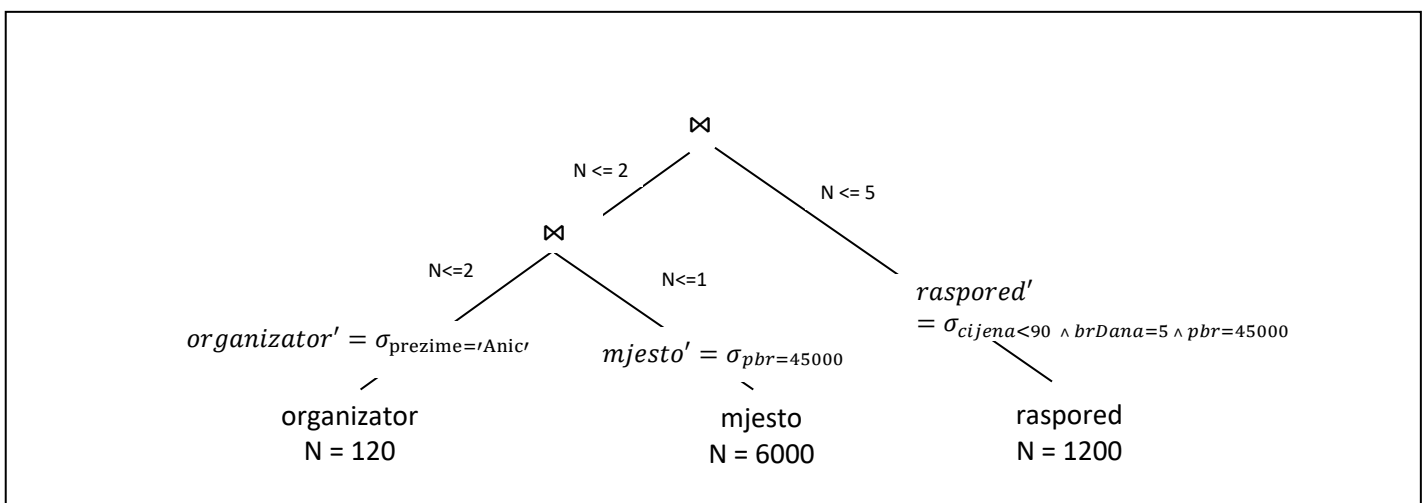
$mjesto \bowtie raspored \rightarrow$ pbr je zajednički atribut, a kao ključ je prisutan u relaciji mjesto

$$N(mjesto \bowtie raspored) = N(raspored') = 5$$

$organizator \bowtie raspored \rightarrow$ sifOrg je zajednički atribut, a kao ključ je prisutan u relaciji organizator

$$N(organizator \bowtie raspored) = N(raspored') = 5$$

Optimalni redoslijed je: $(organizator \bowtie mjesto) \bowtie raspored$



8. (6 bodova)

a)

1NF (ključ relacije):

REZULTATI:

*sifVlasnik, ime, prezime, ulicaKbr, pbr, imeGrad, sifZivotinja, imeZivotinja, nazNatjecanje, godina,
certifikat, rezultat*

2NF (zavisnost o dijelu ključa):

ZIVOTINJA:

sifZivotinja, imeZivotinja, sifVlasnik, ime, prezime, ulicaKbr, pbr, imeGrad

NATJECANJE:

nazNatjecanje, godina, certifikat

REZULTATI2:

sifZivotinja, nazNatjecanje, godina, rezultat

3NF (tranzitivne zavisnosti):

ZIVOTINJA2:

sifZivotinja, imeZivotinja, sifVlasnik,

VLASNIK:

sifVlasnik, ime, prezime, ulicaKbr, pbr, imeGrad

GRAD:

pbr, imeGrad

VLASNIK2:

sifVlasnik, ime, prezime, ulicaKbr, pbr

KONAČAN REZULTAT:

Normalizirana shema se sastoji od relacija:

{ VLASNIK2, GRAD, ZIVOTINJA2, NATJECANJE, REZULTATI2 }

b)

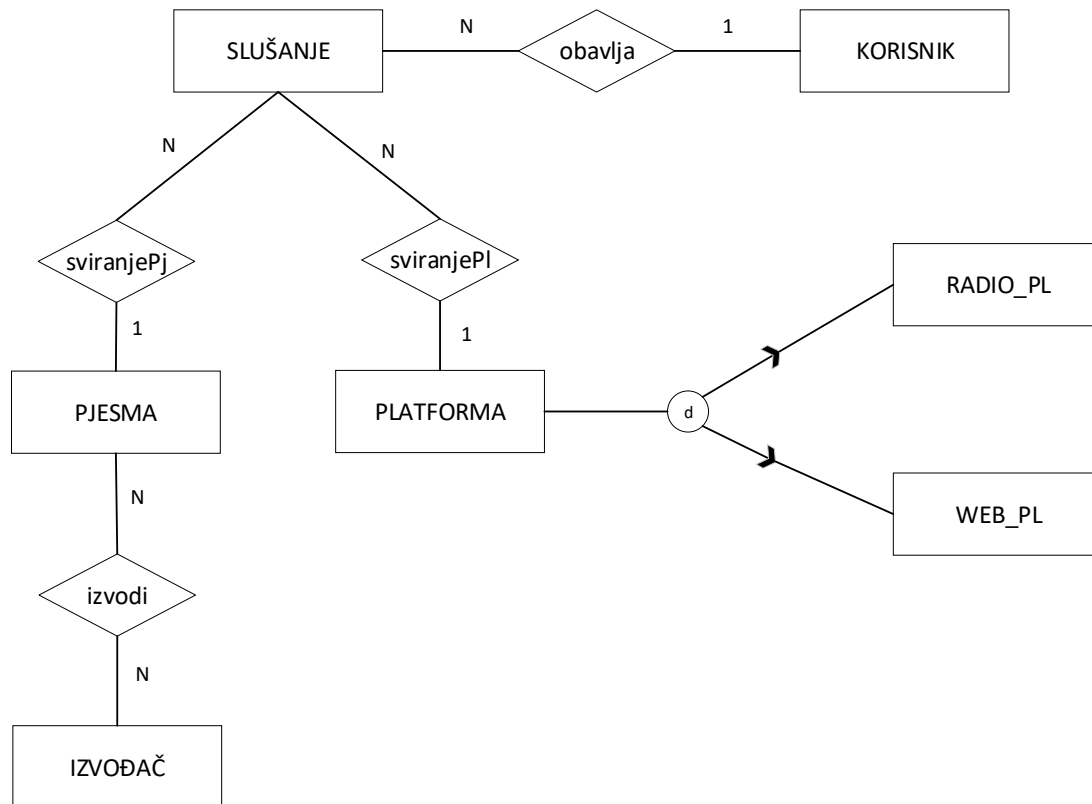
```
SELECT * FROM rezultati AS r1, rezultati AS r2
WHERE r1.ime = r2.ime AND r1.prezime = r2.prezime
AND r1.ulicaKbr <> r2. ulicaKbr;
```

Ako navedeni SQL upit vrati barem 1 n-torku, FZ **sigurno ne vrijedi**.

U suprotnom, FZ **bi mogla** vrijediti.

9. (7 bodova)

ER model:



Sheme:

entiteti	veze
KORISNIK = <u>korisnickolme</u> , email (K2 = <u>email</u>) PJEŠMA = <u>sifPjesma</u> , nazPjesma, trajanje IZVOĐAČ = <u>sifIzvodac</u> , nazIzvodac PLATFORMA = <u>sifPlatforma</u> , nazPlatforma RADIO_PL = <u>sifPlatforma</u> , maksDomet, povrsina WEB_PL = <u>sifPlatforma</u> , url (K2 = <u>url</u>) SLUŠANJE = <u>sifSlušanje</u> , datVrPocetak	izvodi = <u>sifPjesma</u> , <u>sifIzvodac</u> obavlja = <u>sifSlusanje</u> , sifKorisnik sviranjePj = <u>sifSlusanje</u> , sifPjesma sviranjePl = <u>sifSlusanje</u> , sifPlatforma