

1. (3 boda) A grupa

Zadatak riješite **jednom SQL naredbom**.

Ispišite identifikator vlasnika te ime i prezime vlasnika korisničkih računa ženskog spola čije ime završava na „y“ ili „i“, i koji su bili preplaćeni na pakete čija je pretplata deaktivirana u prethodnih pet godina, gledano od trenutnog datuma u ponoć (granice uključene), a vrijedili su duže od jednog mjeseca i jednog dana.

ownerid	firstname	lastname
146	Dotty	Shannon
...

U rezultatu se ne smiju pojavljivati duplikati. Upit mora ispravno raditi bez obzira na vrijeme pokretanja.

```
SELECT DISTINCT ownerid, firstname, lastname
FROM owner
NATURAL JOIN ownerpack
WHERE (LOWER(firstname) LIKE '%y' or LOWER(firstname) LIKE '%i') AND GENDER = 'F'
AND (packenddatetime - packstartdatetime) > '1 month 1 day'::INTERVAL
AND packenddatetime::DATE >= CURRENT_DATE - '5 years'::INTERVAL
```

2. (4 boda) A grupa

Zadatak riješite **jednom SQL naredbom**.

Potrebno je napisati upit za potrebe analize popularnosti komedija kod korisnika koji preferiraju filmove kraćeg trajanja.

Za svakog vlasnika računa *Streamflix* internetskog servisa usluga ispišite identifikator, ime i prezime vlasnika te ukupan broj komedija (medijskih sadržaja naziva žanra `Comedies`) koje su s tog korisničkog računa (neovisno o profilu) **odgledane do kraja** (vrijednost atributa `savedProgress` gledanja jednak je trajanju sadržaja). Ispišite i tekstualnu poruku:

- * `Obožavatelji komedija` – ako je s korisničkog računa ukupno odgledano (do kraja) više od 10 komedija (uključivo)
- * `Nisu obožavatelji komedija` – ako je s korisničkog računa ukupno odgledano (do kraja) između 1 i 9 komedija

Podatke je potrebno ispisati samo za one vlasnike računa koji u **periodu trenutno aktivnog paketa** nisu **započeli gledati** niti jedan medijski sadržaj koji **traje barem 2 sata** (preko bilo kojeg profila).

Ako je unutar nekog korisničkog računa ista komedija gledana više puta do kraja, neovisno o profilu koji je gledao, treba prebrojati svako takvo gledanje. Nazive atributa potrebno je imenovati u skladu s tablicom iz primjera. Ispis poredajte po identifikatoru vlasnika uzlazno.

Primjer ispisa (podaci ne moraju odgovarati stvarnim):

idVlasnika	ime	prezime	brojKomedija	poruka
111	Matt	Berry	17	Obožavatelji komedija
112	Mare	Hawks	6	Nisu obožavatelji komedija
113	Gregory	House	2	Nisu obožavatelji komedija
...

```

SELECT owner.ownerid as idVlasnika, firstname as ime, lastname as prezime,
       COUNT(*) as brojKomedija,
       CASE
         WHEN count(*) >= 10 THEN 'Obožavatelji komedija'
         ELSE 'Nisu obožavatelji komedija'
       END as poruka
FROM owner
NATURAL JOIN trackview
NATURAL JOIN track
NATURAL JOIN trackgenre
NATURAL JOIN genre
WHERE genrename = 'Comedies'
AND savedProgress = duration
AND NOT EXISTS (SELECT *
                 FROM trackview as t
                 NATURAL JOIN track
                 NATURAL JOIN ownerpack
                 WHERE t.ownerid = owner.ownerid
                 AND viewstartdatetime >= packstartdatetime
                 AND packenddatetime is NULL
                 AND duration >= '2 HOURS'::INTERVAL)
GROUP BY owner.ownerid, firstname, lastName
ORDER BY owner.ownerid;

```

3. (5 bodova) A grupa

Gledatelj/gledatelji iza pojedinog korisničkog profila mogu ocijeniti sviđa li im se određeni sadržaj ili ne (**profileTrack.liked**, poprima vrijednost 1 ili -1).

Pravilo je da ocjenu može dodijeliti gledatelj/gledatelji iza korisničkog profila koji su odgledali barem 40% trajanja medijskog sadržaja.

Trajanje medijskog sadržaja pohranjeno je u atributu **track.duration** (u obliku intervala).

Koliko je određenog medijskog sadržaja korisnik pogledao pohranjeno je u atributu **trackView.savedProgress** (također u obliku intervala). Naravno, korisnik određeni medijski sadržaj (**track**) može gledati više puta i svako neprekinuto gledanje može imati drugačiji **savedProgress** (manji ili veći od prethodnih za isti medijski sadržaj)

Napisati niz SQL naredbi za kreiranje svih potrebnih objekata kojima će se pri izmjeni n-torke u relaciji **profileTrack** spriječiti narušavanje opisanog ograničenja o ocjenjivanju.

Pri narušavanju opisanog ograničenja korisniku javiti sljedeću grešku:

'Pogreška: Profilu <profileName> korisnika <firstName>, <lastName> nije dozvoljeno ocjenjivanje sadržaja <trackTitle>'.

Umjesto <firstName> i <lastName> potrebno je ispisati ime i prezime korisnika, a umjesto <trackTitle> naziv medijskog sadržaja, kojeg neuspješno pokušava ocijeniti.

U svim ostalim slučajevima izmjene n-torki u tablici **profileTrack** moraju biti uspješno izvedene.

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

```

--DROP TRIGGER updProfileTrack ON profileTrack;;
--DROP FUNCTION chkUpdProfileTrack() ;
CREATE FUNCTION chkUpdProfileTrack() RETURNS TRIGGER AS $$
    DECLARE p_profileName    trackView.profileName%TYPE;
    DECLARE p_firstName      owner.firstName%TYPE;
    DECLARE p_lastName       owner.lastName%TYPE;
    DECLARE p_trackTitle     track.trackTitle%TYPE;
    DECLARE p_savedProgress  trackView.savedProgress%TYPE;
    DECLARE p_duration       track.duration%TYPE;

    BEGIN
        SELECT trackView.profileName
            , firstName, lastName, trackTitle
            , trackView.savedProgress, track.duration
        INTO p_profileName
            , p_firstName, p_lastName, p_trackTitle
            , p_savedProgress, p_duration
        FROM trackView
            NATURAL JOIN owner
            NATURAL JOIN track
        WHERE trackView.ownerId = NEW.ownerId
            AND trackView.profileName = NEW.profileName
            AND trackView.trackId = NEW.trackId
        ORDER BY trackView.savedProgress DESC
        LIMIT 1;
        IF (p_savedProgress < 0.4*p_duration) THEN
            RAISE EXCEPTION
            'Pogreška: Profilu % korisnika %, % nije dozvoljeno ocjenjivanje sadržaja %.'
            , p_profileName, p_firstName, p_lastName, p_trackTitle;
        END IF;
        RETURN NEW;
    END;
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER updProfileTrack
BEFORE UPDATE ON profileTrack
FOR EACH ROW
    WHEN (NEW.liked IS NOT NULL)
EXECUTE FUNCTION chkUpdProfileTrack();

```

--ovo dalje nije dio rješenja - služi za testiranje okidača i procedure.
-- ERROR

```

Begin transaction;
UPDATE profileTrack SET liked = 1
    WHERE ownerId = 220
        AND profileName = 'alexisrodriguez'
        AND trackId = 2587
Rollback transaction;

```

--SUCCESS

```

Begin transaction;
UPDATE profileTrack SET liked = 1
    WHERE ownerId = 182
        AND profileName = 'emotionwarlike'
        AND trackId = 4994;
Rollback transaction;

```

S ovim upitom je moguće pronaći zapise za testiranje:

```

SELECT MAX(savedProgress), track.duration, MAX(savedProgress) > 0.4*duration,
trackView.ownerId, trackView.trackId
    , trackView.profileName
    , firstName, lastName, trackTitle
    --, trackView.savedProgress, track.duration
FROM trackView
    NATURAL JOIN owner
    NATURAL JOIN track
GROUP BY duration, trackView.ownerId, trackView.trackId

```

```

        , trackView.profileName
        , firstName, lastName, trackTitle
        , track.duration
HAVING MAX(savedProgress) >= 0.4*duration

```

4. (4 boda) A grupa

Administrator baze podataka je kreirao bazu podataka *streamflix* i tablice u shemi *public*.

A) Napisati SQL naredbe kojima će administrator sustava korisniku *kolar*, koji ima ovlasti za uspostavu korisničke sjednice, dodijeliti i ovlasti za:

- izmjenu naziva jezika u tablici *language* uz mogućnost dodjeljivanja tih istih ovlasti ostalim korisnicima,
- pregled identifikatora medijskog sadržaja (*trackId*), naziva medijskog sadržaja i broja žanrova kojima određeni medijski sadržaj pripada, za sadržaje za koje je specificirano da pripadaju barem jednom žanru,
- kreiranje novih shema (kako bi mogao kreirati shemu *media* i tablice u njoj)

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

B) Nakon što je korisnik *kolar* kreirao shemu *media* i u njoj tablicu *device*, napisati SQL naredbe kojima će korisnik *kolar* omogućiti korisniku *novak*:

- pristup objektima sadržanima u shemi *media* bez mogućnosti stvaranja novih objekata,
- pregled sadržaja tablice *device*.

A)

```
GRANT UPDATE(langname) ON language TO kolar WITH GRANT OPTION;
```

```

CREATE VIEW brojZanrovaPoSadrzaju AS
  SELECT trackId, trackTitle, count(genreId) AS numberOfGenres
  FROM track NATURAL JOIN trackgenre
  GROUP BY trackId, trackTitle;

```

Drugi način za view:

```

CREATE VIEW brojZanrovaPoSadrzaju AS
  SELECT trackId, trackTitle,
         (SELECT COUNT(*)
          FROM trackgenre where trackId = track.trackId) AS numberOfGenres
  FROM track
  WHERE (SELECT COUNT(*) from trackgenre where trackId = track.trackId) > 0;

```

```
--može COUNT(genreid) umjesto COUNT(*)
```

```
GRANT SELECT ON brojZanrovaPoSadrzaju TO kolar;
```

```
GRANT CREATE ON DATABASE streamflix TO kolar;
```

B)

```

GRANT USAGE ON SCHEMA media TO novak;
GRANT SELECT ON media.device TO novak;

```

5. (4 boda) A grupa

Izvršavanjem sljedeće SELECT naredbe jezika SQL nad bazom podataka *StreamFlix*

```
SELECT trackId, savedProgress
FROM trackView
WHERE profilename = 'curiouswimp'
AND savedprogress BETWEEN '35 min':INTERVAL AND '45 min':INTERVAL
ORDER BY trackId;
```

dobije se sljedeći rezultat:

```
trackid|savedprogress
6505|00:45:00
10261|00:39:00
45284|00:44:00
```

Pretpostavimo da su korisnici **Ivo** i **Ana** trenutno jedini korisnici nad bazom te da su uspješno uspostavili paralelne sjednice. Također pretpostavimo da oboje imaju prava izmjene podataka nad svim tablicama u bazi. **Ivo** i **Ana** izvode naredbe kao u tablici dolje naznačenim redoslijedom.

Gore navedena SELECT naredba izvršava se i pod rednim brojevima 7, 8 i 11 ali je radi preglednosti u tablici napisan samo početak i završetak naredbe.

	Ivo	Ana
1	BEGIN TRANSACTION;	BEGIN TRANSACTION; SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
2	SET TRANSACTION ISOLATION LEVEL READ COMMITTED;	
3		
4		
5	INSERT INTO TRACKVIEW (deviceid, viewstartdatetime, profilename, trackid, savedprogress) VALUES ('A0', CURRENT_TIMESTAMP, 'curiouswimp', 10890, '45 MINUTES':INTERVAL);	UPDATE trackview SET savedProgress = '41 minutes':INTERVAL WHERE profilename = 'curiouswimp' AND savedProgress = '39 MINUTES':INTERVAL;
6		
7	SELECT ... ORDER BY trackId;	
8		SELECT ... ORDER BY trackId;
9	DELETE FROM trackview WHERE profilename = 'curiouswimp' AND savedprogress = '39 MINUTES':INTERVAL;	COMMIT;
10		
11	SELECT ... ORDER BY trackId;	
12	COMMIT;	

- Navedite rezultate SELECT naredbi pod rednim brojevima 7 i 8.
- Da bi korisnik Ivo mogao izvršiti naredbu pod rednim brojem 9 mora čekati završetak Anine transakcije. Objasnite zbog čega.
Pretpostavite da je (bez obzira na to što Ivo čeka) Ana nastavila s izvršavanjem svoje transakcije i obavila naredbu pod rednim brojem 10.
Što se sada dogodilo s Ivinom transakcijom? Koliko je n-torki obrisala naredba 9?
- Navedite rezultat SELECT naredbe pod rednim brojem 11.

7	6505 "00:45:00" 10261 "00:39:00" 10890 "00:45:00" 45284 "00:44:00"
8	6505 "00:45:00" 10261 "00:41:00" 45284 "00:44:00"
b)	Radi se o tome da pisanje blokira pisanje. UPDATE naredba Anine transakcije je promijenila n-torku koja zadovoljava uvjet selekcije DELETE naredbe Ivine transakcije. Ivina transakcija mora čekati dok ishod Anine transakcije ne bude poznat. Nakon što Anina transakcija završi obavlja se naredba pod 9. Neće obrisati nijednu ntorku jer sada ova s trackid = 10261 ne zadovoljava uvjet selekcije.
11	6505 "00:45:00" 10890 "00:45:00" 45284 "00:44:00" 10261 "00:41:00"

6. (4 boda) A grupa

Za rješavanje sljedećeg zadatka koristite stvarne statističke podatke koji odgovaraju *Streamflix* bazi podataka koja se nalazi na Edgaru i koristi se u ovoj provjeri. U spomenutoj bazi podataka definirana su ograničenja primarnih i stranih ključeva.

Izvodi se upit:

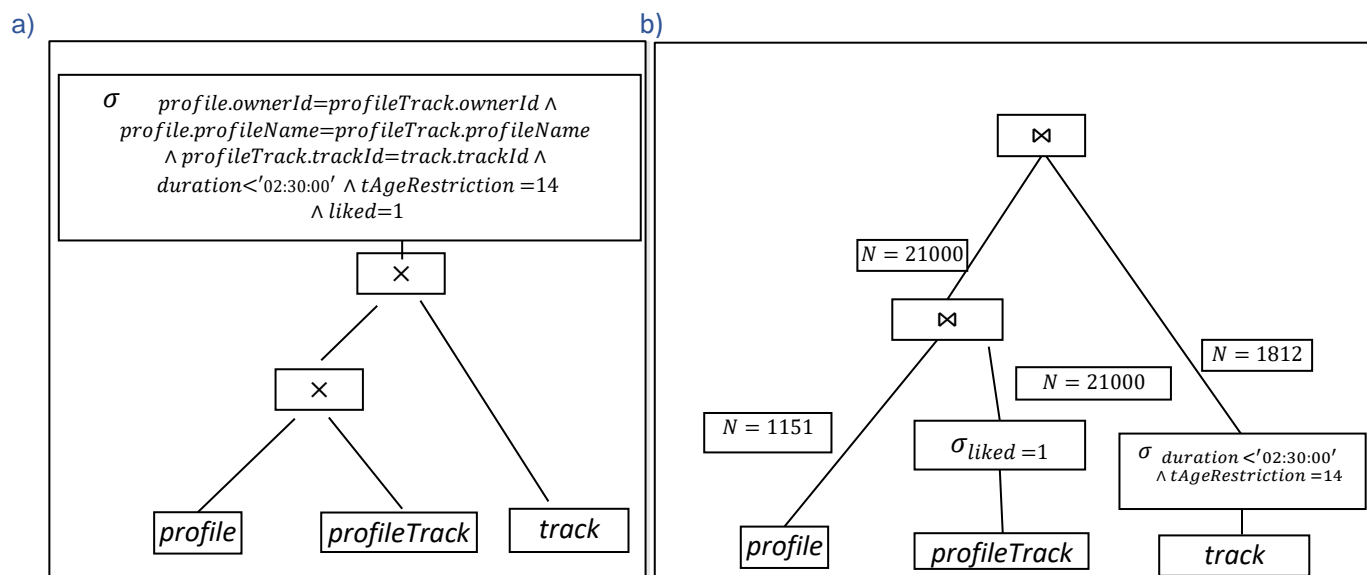
```
SELECT *
FROM profile, profiletrack, track
WHERE profile.ownerid = profiletrack.ownerid
AND profile.profilename = profiletrack.profilename
AND profiletrack.trackId = track.trackid
AND duration < '02:30:00'::interval
AND liked = 1
AND tAgeRestriction = 14;
```

- 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 statističke izračune i 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 jednakovrijednih najboljih redoslijeda spajanja, odlučite se za jedan i nacrtajte ga.

Napomena: Izračunate statističke podatke, kao i postupak i izračun za procjenu broja n-torki **unesite u prostor za slobodni unos teksta ispod teksta zadatka**. Ako prilikom izračuna broja n-torki dobijete decimalnu vrijednost, zaokružite rezultat na prvi veći cijeli broj.

Statistički podatki potrebni za izračun:

profile	profileTrack	track
$N(\text{profile}) = 1151$	$N(\text{profileTrack}) = 42000$ $V(\text{liked}, \text{profileTrack}) = 2$	$N(\text{track}) = 27176$ $V(\text{tAgeRestriction}, \text{track}) = 5$



Prvo se spaja profileTrack s profile ili s track.

Potiskivanje selekcije:

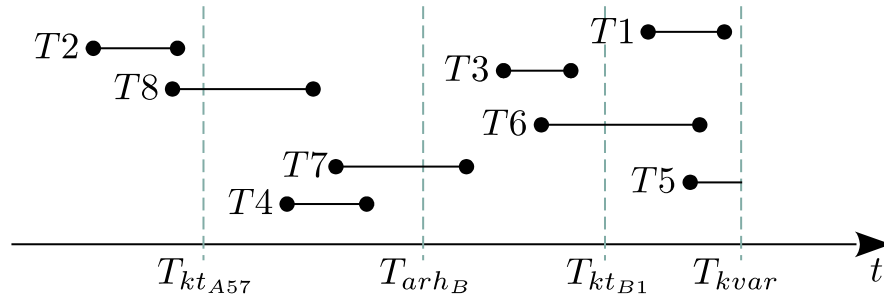
- `duration < '02:30:00'` and `tAgeRestriction = 14` na **track**
- `liked = 1` na **profileTrack**
- **track.duration < '02.30.00'** and **tAgeRestriction = 14**
 $\rightarrow N(\text{track}') = N(\text{track}) / (3 * V(\text{tAgeRestriction}, \text{track}))$
 $= 27176 / (3 * 5) \approx 1812$
- **profileTrack.liked = 1** $\rightarrow N(\text{profileTrack}')$
 $= N(\text{profileTrack}) / V(\text{liked}, \text{profileTrack})$
 $= 42000 / 2$
 $= 21000$

Međurezultati spajanja:

profile i profileTrack' (fk u profileTrack') $\Rightarrow N(\text{profileTrack}') = 21000$
profile i track' Kartezijev $\Rightarrow N(\text{profile}) \times N(\text{track}') = 1151 * 1812 = 2\,085\,612$
profileTrack' i track (fk u profileTrack') $\Rightarrow N(\text{profileTrack}') = 21000$

7. (3 boda) A grupa

U SUBP je došlo do pogreške računalnog sustava u trenutku T_{kvar} . Baza podataka je posljednji puta arhivirana (razina 0) u trenutku T_{arhB} . Odrađene su kontrolne točke T_{ktA57} i T_{ktB1} . Dnevnicu izmjena su sačuvani. Vremenski slijed događaja je sljedeći:



Pokrenut je postupak obnove baze podataka.

a) Navedite identifikatore transakcija koje su u:

- listi za ponovno obavljanje
- listi za poništavanje

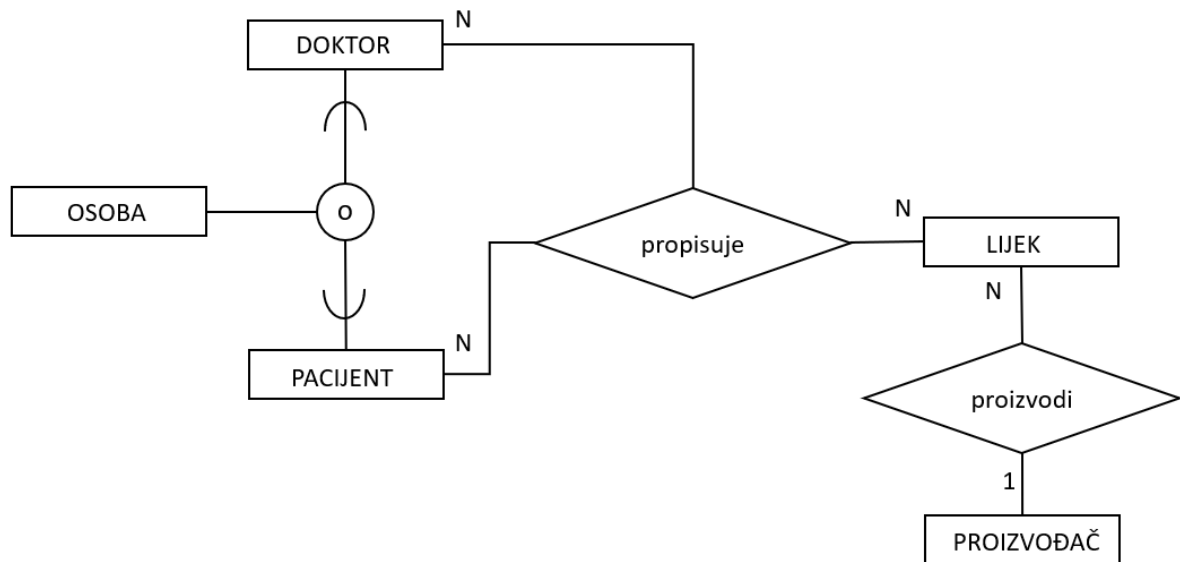
b) Koji je posljednji trenutak za koji pouzdano znamo da imamo zapisan dnevnik izmjena u datoteku dnevnika? Zašto je to tako?

- a)
- lista za ponovno obavljanje = T1,T6
 - lista za poništavanje = T5

b) COMMIT na T1 - *write-ahead log rule*

8. (3 boda) A grupa

Za zadani ER model napisati naredbe za kreiranje ekvivalentnog relacijskog modela u obliku SQL naredbi za kreiranje relacija s ugrađenim opisima primarnih, alternativnih ključeva i općih pravila integriteta. Tipove podataka u naredbama nije potrebno navoditi.



OSOBA (OIB, ime, prezime)

DOKTOR (OIB, email)

PACIJENT (OIB, brojOsiguranja, datumRodenja, telefon)

LIJEK (sifLijek, nazivLijek)

PROIZVOĐAČ (sifProizv, nazivProizv)

```
CREATE TABLE osoba (
    OIB ... PRIMARY KEY,
    ime ...,
    prezime ...);
```

```
CREATE TABLE doktor (
    OIB ... PRIMARY KEY REFERENCES osoba(OIB),
    email ... );
```

```
CREATE TABLE pacijent (
    OIB ... PRIMARY KEY REFERENCES osoba(OIB),
    brojOsiguranja ... UNIQUE,
    datumRodenja ...,
    telefon ... );
```

```
CREATE TABLE proizvođač (
    sifProizv ... PRIMARY KEY,
    nazivProizv ...);
```

```
CREATE TABLE lijek (
    sifLijek ... PRIMARY KEY,
    nazivLijek ...,
    sifProizv ...,
    sifProizvodac ... REFERENCES proizvođač(sifProizvodac));
```

```
CREATE TABLE propisuje (
    sifPacijent ... REFERENCES pacijent(OIB),
    sifDoktor ... REFERENCES doktor (OIB),
    sifLijek ... REFERENCES lijek(sifLijek),
    PRIMARY KEY (sifPacijent, sifDoktor, sifLijek);
```

9. (6 bodova) A grupa

U okviru Međunarodne vrtne izložbe FloraArt 2021. održava se *Hrvatski kup florista* na kojem natjecatelji izrađuju cvjetne aranžmane za svaku od zadanih tema.

Potrebno je oblikovati ER model baze podataka za potrebe organizacije i provedbe natjecanja.

Evidentiraju se podatci o osobama koje sudjeluju u natjecanju bilo kao natjecatelji, bilo kao članovi međunarodnog ocjenjivačkog povjerenstva.

Za osobu se evidentira država (državljanstvo) i identifikacijski broj koji je jedinstven na razini države (npr. za hrvatske državljane je to OIB), ime, prezime, spol i datum rođenja. Država je opisana ISO oznakom i nazivom države koji su jedinstveni. Za osobu koja je natjecatelj dodatno se evidentira jedinstvena šifra natjecatelja te datum prijave na natjecanje. Za osobu koja je ocjenjivač dodatno se evidentira šifra ocjenjivača (jedinstvena) i struka. Za struku se evidentira jedinstvena šifra i naziv. Jedan ocjenjivač može imati više struka.

U okviru natjecanja definirano je nekoliko tema. Za temu se evidentira jedinstvena šifra, naziv i vrijeme potrebno za izradu aranžmana na tu temu.

Za svaku temu organizator osigurava materijal kojeg će natjecatelji koristiti. Materijal je opisan jedinstvenom šifrom, nazivom i vrstom materijala (cvijeće, zelenilo, vrpce...). Za vrstu materijala evidentira se jedinstvena oznaka i naziv. Isti materijal može biti predviđen za više tema, pri čemu se za svaku temu i za svaki materijal definira količina materijala koji će biti na raspolaganju svakom natjecatelju (svima jednako).

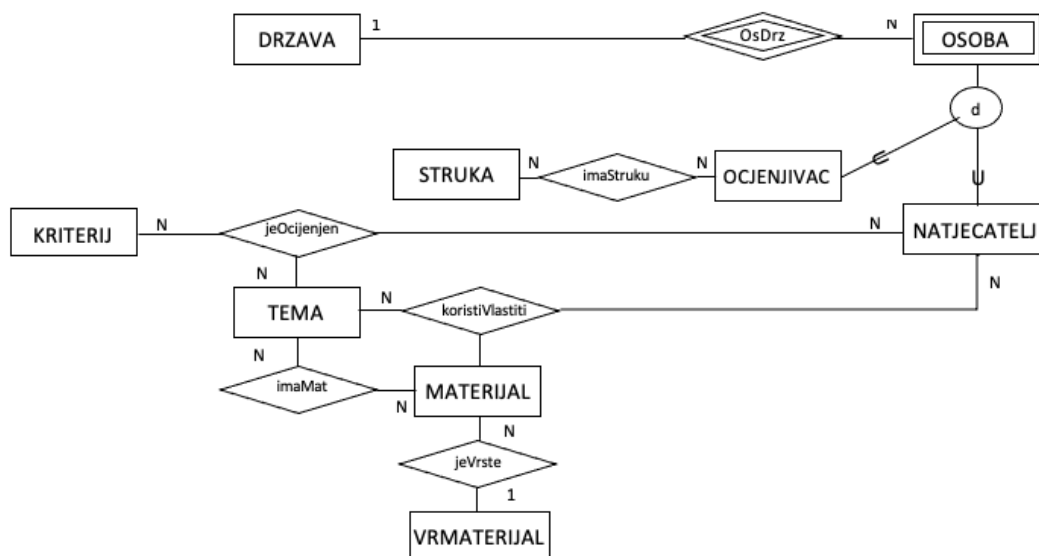
Natjecatelj za realizaciju svake teme, uz predefinirane materijale, može koristiti i vlastite, pri čemu se evidentira količina (vlastitog) materijala kojeg će natjecatelj koristiti za pojedinu temu. Za jednu temu natjecatelj može koristiti više materijala, jedan materijal natjecatelj može koristiti za više tema te jedan materijal za jednu temu može koristiti više natjecatelja.

Podatci o svim materijalima (organizatora i natjecatelja) evidentiraju se u zajedničkom katalogu (tablici) materijala.

Propisani su kriteriji za ocjenjivanje (ideja, boja, kompozicija i tehnika). Kriterij ima svoju oznaku i naziv koji su jedinstveni. Natjecatelj za svaku temu i svaki kriterij dobije jednu ocjenu (od 1 do 5).

a) Nacrtajte ER model baze podataka **na priloženom papiru**. Nemojte crtati atribute.

b) U prostor za slobodni unos teksta ispod teksta zadatka navedite **sheme entiteta** i **sheme veza** (označite ključeve). Svaki entitet (osim slabih entiteta) opisati **isključivo vlastitim atributima**. Nužno je da sve sheme zadovoljavaju **3NF**.



Opisi entiteta (primarni ključevi su podcrtani):

DRZAVA = ISOoznDrz, nazDrz

OSOBA = ISOoznDrz, idOsoba, ime, prezime, spol, datRod
 NATJECATELJ = sifNatjecatelj, ISOoznDrz, idOsoba, datumPrijava

OCJENJIVAC = sifOcjenjivac, ISOoznDrz, idOsoba

STRUKA = sifStruka, nazStruka
 TEMA = sifTema, nazTema, vrijemeIzrade
 MATERIJAL = sifMaterijal, nazMaterijal
 VRMATERIJAL = sifVrMaterijal, nazVrMaterijal
 KRITERIJ = sifKriterij, nazKriterij

K1_{DRZAVA} = ISOoznDrz;

K2_{DRZAVA} = nazDrz
 K_{OSOBA} = ISOoznDrz, idOsoba
 K1_{NATJECATELJ} = sifNatjecatelj;
 K2_{NATJECATELJ} = ISOoznDrz, idOsoba;
 K1_{OCJENJIVAC} = sifOcjenjivac
 K2_{OCJENJIVAC} = ISOoznDrz, idOsoba;
 K_{STRUKA} = sifStruka
 K_{TEMA} = sifTema
 K_{MATERIJAL} = sifMaterijal
 K_{VRMATERIJAL} = sifVrMaterijal
 K_{KRITERIJ} = sifKriterij

Opisi veza (primarni ključevi su podcrtani):

OsDrz = ISOoznDrz, osobald
 imaStruku = sifOcjenjivac, sifStruka
 imaMat = sifTema, sifMaterijal
 jeVrste = sifMaterijal, sifVrMaterijal
 koristiVlastiti = sifNatjecatelj, sifTema, sifMaterijal, kolicina

jeOcijenjen = sifNatjecatelj, sifTema, sifKriterij, ocjena

K_{OsDrz} = ISOoznDrz, osobald
 K_{imaStruku} = sifOcjenjivac, sifStruka
 K_{imaMaterijal} = sifTema, sifMaterijal
 K_{jeVrste} = sifMaterijal
 K_{koristiVlastiti} = sifNatjecatelj, sifTema, sifMaterijal
 K_{jeOcijenjen} = sifNatjecatelj, sifTema, sifKriterij