

## 1. (3 boda) B grupa

Zadatak riješite **jednom SQL naredbom**.

Ispišite naziv žanra, prosječnu ocjenu svih naslova u žanru (zaokruženu na dvije decimale) i broj naslova u žanru za sve žanrove čija je prosječna ocjena svih naslova u žanru veća ili jednaka od 50 i sadrže više od 100 naslova.

```
SELECT genrename, ROUND(AVG(trackrating),2) as average_trackrating,
COUNT(tracktitle)as tracktitle_count
FROM trackgenre
NATURAL JOIN genre
NATURAL JOIN track
GROUP BY genrename
HAVING COUNT(tracktitle)>100 and AVG(trackrating)>=50
```

## 2. (4 boda) B grupa

Zadatak riješite **jednom SQL naredbom**.

Za svaki film iz 2019. ili 2020. godine koji traje između 60 i 75 minuta (uključivo) ispišite jedinstveni identifikator, naziv filma, te datum izlaska filma. Ispišite također i naziv filma prethodnika te datum izlaska filma prethodnika, ali samo u slučaju ako je film prethodnik ostvario **veću** zaradu od prodaje kino ulaznica u odnosu na svog nasljednika. Ako film (iz 2019. ili 2020. godine koji traje između 60 i 75 minuta) **nema** prethodnika koji je ostvario veću zaradu od prodaje kino ulaznica, za sve podatke o prethodniku tog filma ispišite NULL vrijednosti. Dodatno ispišite omjer zarada filma prethodnika i filma nasljednika (prethodnik/nasljednik, NULL ako film nema prethodnika s većom zaradom) te jednu od tekstualnih poruka:

- \* prethodnik zaradio barem dvostruko više
- \* prethodnik nije zaradio dvostruko više
- \* nema podataka

ovisno o dobivenom omjeru zarada.

Stupce nazvati prema danom primjeru ispisa. Ispis poredajte po omjeru zarada uzlazno, a potom po nazivu filma abecedno.

Primjer ispisa (podaci ne moraju odgovarati stvarnima):

idFilma	nazivFilma	datumIzlaska	nazivPreth	datumIzlaskaPre th	omjerZarada	poruka
3201	Happy Autumn	05.09.2019.	Happy Summer	02.03.2017.	1.064423462365 27512	prethodnik nije zaradio dvostruko više
4013	Bumpty Bumpty 2	28.09.2020.	Bumpty Bumpty	11.07.2007.	2.543242337618 21624	prethodnik zaradio barem

						dvostruko više
5777	A Very Sad Christmas	01.05.2020.	null	null	null	nema podataka
...	...	...	...	...	...	...

```

SELECT m1.trackid as idFilma,
       t1.trackTitle as nazivFilma, t1.releasedate as datumIzlaska,
       t2.trackTitle as nazivPreth, t2.releasedate as datumIzlaskaPreth,
       1.0 * m2.boxincome / m1.boxincome as omjerZarada,
       CASE
         WHEN 1.0 * m2.boxincome / m1.boxincome >= 2
           THEN 'prethodnik zaradio barem dvostruko više'
         WHEN 1.0 * m2.boxincome / m1.boxincome < 2
           THEN 'prethodnik nije zaradio dvostruko više'
         ELSE  'nema podataka'
       END as poruka
FROM movie as m1
JOIN track as t1 ON m1.trackid = t1.trackid
LEFT JOIN movie AS m2 ON m1.prevmovieid = m2.trackid
      AND m1.boxincome < m2.boxincome
LEFT JOIN track AS t2 ON m2.trackid = t2.trackid
WHERE t1.duration between '1 HOUR' AND '75 MINUTES'
      AND EXTRACT (year from t1.releasedate) IN (2019,2020)
ORDER BY omjerZarada ASC, t1.tracktitle;

```

### 3 (5 bodova) B grupa

Za aktivan paket korisnika atribut datum deaktivacije (***ownerPack.packEndTime***) ima vrijednost NULL.

Za sve poznate (NOT NULL) vrijednosti atributa ***ownerPack.packEndTime*** ta vrijednost je manja (tj. starija) od trenutka "sada" (CURRENT\_TIMESTAMP) i to ne morate provjeravati.

Napisati niz SQL naredbi za kreiranje svih potrebnih objekata kojima će se pri unosu n-torke u relaciju ***ownerPack*** spriječiti upis (INSERT) n-torke:

- s poznatim datumom deaktivacije
- kojom bi bilo narušeno sljedeće ograničenje: Vlasnik (owner) može imati samo jedan aktivan paket.

Pri narušavanju ograničenja pod

- korisniku javiti sljedeću grešku:  
'Pogreška: Pri unosu paketa <packName> za korisnika <firstName>, <lastName> datum deaktivacije mora biti nepoznat '.
- korisniku javiti sljedeću grešku:  
'Pogreška: Paket <packName> je aktivan za korisnika <firstName> <lastName>.'.

Umjesto <firstName> i <lastName> potrebno je ispisati ime i prezime korisnika, a umjesto <packName> naziv paketa kojeg korisnik bezuspješno pokušava prilikom upisa deaktivirati (slučaj a)), odnosno naziv njegovog trenutno aktivnog paketa (slučaj b)).

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

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

```
--DROP TRIGGER insOwnerPack ON ownerPack;
--DROP FUNCTION chkInsOwnerPack();

CREATE FUNCTION chkInsOwnerPack() RETURNS TRIGGER AS $$
    DECLARE p_firstName owner.firstName%TYPE;
    DECLARE p_lastName owner.lastName%TYPE;
    DECLARE p_packName pack.packName%TYPE;

    BEGIN
        SELECT firstName, lastName
            INTO p_firstName, p_lastName
            FROM owner
            WHERE ownerId = NEW.ownerId;

        IF (NEW.packEndDateTime IS NOT NULL) THEN
            SELECT packName
                INTO p_packName
                FROM pack
                WHERE packId = NEW.packId;
            RAISE EXCEPTION 'Pogreška: Pri unosu paketa % za korisnika %, %
datum deaktivacije mora biti nepoznat.'
                , p_packName, p_firstName, p_lastName;
        END IF;

        SELECT packName
            INTO p_packName
            FROM ownerPack
                NATURAL JOIN pack
            WHERE ownerPack.ownerId = NEW.ownerId
            -- ovo nije potrebno zbog toga što je trigger BEFORE INSERT i tada još
nema NEW zapisa u tablici
            -- AND ownerPack.packStartDateTime != NEW.packStartDateTime
            AND ownerPack.packEndDateTime IS NULL;

        IF (p_packname IS NOT NULL) THEN
            RAISE EXCEPTION 'Pogreška: Paket % je aktivan za korisnika % %.'
                , p_packName, p_firstName, p_lastName;
        END IF;

        RETURN NEW;
    END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER insOwnerPack
BEFORE INSERT ON ownerPack
FOR EACH ROW
    EXECUTE FUNCTION chkInsOwnerPack();
```

Ovo dalje nije dio rješenja - služi za testiranje:

```
begin transaction;
-- ERROR greška #1
INSERT INTO ownerPack VALUES (1,3, CURRENT_TIMESTAMP, CURRENT_TIMESTAMP)
rollback transaction;
-- ERROR greška #2
begin transaction;
INSERT INTO ownerPack VALUES (2,1, CURRENT_TIMESTAMP, null);
--U poruci treba ispisati Premium jer je to njegov aktivan paket, iako mu
sada pokušavamo insertirati 2-Standard
rollback transaction;
-- SUCCESS
begin transaction;
INSERT INTO ownerPack VALUES (1,3, CURRENT_TIMESTAMP, null)
rollback transaction;
```

---

#### 4. (4 boda) B grupa

Administrator baze podataka je nakon kreiranja baze podataka *streamflix* i tablica u shemi *public* obavio sljedeću SQL naredbu:

```
REVOKE ALL ON SCHEMA public FROM public;
```

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

- pristup objektima sadržanima u shemi *public* s mogućnošću stvaranja novih objekata,
- pregled, unos, izmjenu i brisanje podataka iz tablice *track* samo za one medijske sadržaje koji imaju dostupne zvučne zapise na više od 4 jezika.

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

B) Potrebno je zatim već kreiranoj ulozi *gledatelj* dodijeliti ovlasti pregleda sadržaja tablice *genre* te dati korisniku *rossi* mogućnost korištenja te uloge.

C) Nakon što je korisnik *rossi* kreirao tablicu *device*, napisati SQL naredbe kojima će korisnik *rossi*:

- omogućiti korisniku *ferrari* unos, izmjenu i brisanje svih podataka u tablici *device*, uz mogućnost dodjeljivanja tih istih ovlasti ostalim korisnicima,
- ukinuti dozvolu unosa podataka u tablicu *device* korisniku *ferrari* i svim korisnicima koji su od korisnika *ferrari* tu dozvolu dobili.

A)

```
GRANT CREATE ON SCHEMA public TO rossi;

CREATE VIEW viseOd4Jezika AS
select * from track
where (select count(trackid)
      from audiolang
      where trackid=track.trackid) > 4
WITH CHECK OPTION;

GRANT SELECT, INSERT, UPDATE, DELETE ON viseOd4Jezika TO rossi;
--ili: GRANT ALL ON ...
```

B)

```
GRANT USAGE ON SCHEMA public TO gledatelj;
GRANT SELECT ON genre TO gledatelj;
GRANT gledatelj TO rossi;
```

C)

```
administrator:
GRANT USAGE ON SCHEMA public TO ferrari;

rossi:
GRANT INSERT, UPDATE, DELETE ON device TO ferrari WITH GRANT OPTION;
REVOKE INSERT ON device FROM ferrari CASCADE;
```

## 5. (4 boda) B grupa

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

```
SELECT trackId, savedProgress
FROM trackView
WHERE profilename = 'curiouswimp'
AND savedprogress > '40 minutes'::INTERVAL
ORDER BY trackId;
```

dobije se sljedeći rezultat:

```
trackid|savedprogress
1885|02:10:52
6505|00:45:00
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, 10 i 15 ali je radi preglednosti u tablici napisan samo početak i završetak naredbe.

	Ivo	Ana
1	BEGIN TRANSACTION;	BEGIN TRANSACTION; SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  DELETE FROM trackview WHERE profilename = 'curiouswimp' AND savedprogress > '1 HOUR'::INTERVAL;  UPDATE trackview SET savedProgress = '50 minutes'::INTERVAL WHERE profilename = 'curiouswimp' AND savedProgress = '45 MINUTES'::INTERVAL;  SELECT ... ORDER BY trackId;  DELETE FROM trackview WHERE profilename = 'curiouswimp' AND savedprogress = '44 MINUTES'::INTERVAL;  SELECT ... ORDER BY trackId; COMMIT;
2	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;	
3		
4		
5		
6	INSERT INTO TRACKVIEW (deviceid, viewstartdatetime, profilename, trackid, savedprogress) VALUES ('B2', CURRENT_TIMESTAMP, 'curiouswimp', 10890, '45 MINUTES'::INTERVAL);	
7	SELECT ... ORDER BY trackId;	
8	COMMIT;	
9		
10		
11	BEGIN TRANSACTION;	
12	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;	
13		
14	UPDATE trackview SET savedProgress = '1 HOUR'::INTERVAL WHERE profilename = 'curiouswimp' AND savedProgress = '44 MINUTES'::INTERVAL;	
15		
16		
17	COMMIT;	

- Navedite rezultate SELECT naredbi pod rednim brojem 7 i 10.
- Da bi korisnik Ivo mogao izvršiti naredbu pod rednim brojem 14 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 naredbe pod rednim brojevima 15 i 16.  
Što se dogodilo s Ivinom transakcijom nakon što je Anina potvrđena?
- Navedite rezultate SELECT naredbe pod rednim brojem 15.

7	1885 02:10:52 6505 00:45:00 10890 00:45:00 45284 00:44:00
10	6505 00:50:00 10890 00:50:00 45284 00:44:00
b)	Ana briše n-torku koju Ivo pokušava izmijeniti, zbog čega Ivo mora čekati dok ishod Anine transakcije ne bude poznat. Nakon što Ana izvrši COMMIT uvjet serijalizabilnosti se ne može održati tako da se Ivina transakcija poništava s greškom.
15	6505 00:50:00 10890 00:50:00

## 6. (4 boda) B 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 owner, ownerpack, pack
WHERE owner.ownerid = ownerpack.ownerid
      AND ownerpack.packId = pack.packId
      AND pack.packId = 1
      AND dateOfBirth < '01.01.2000';
```

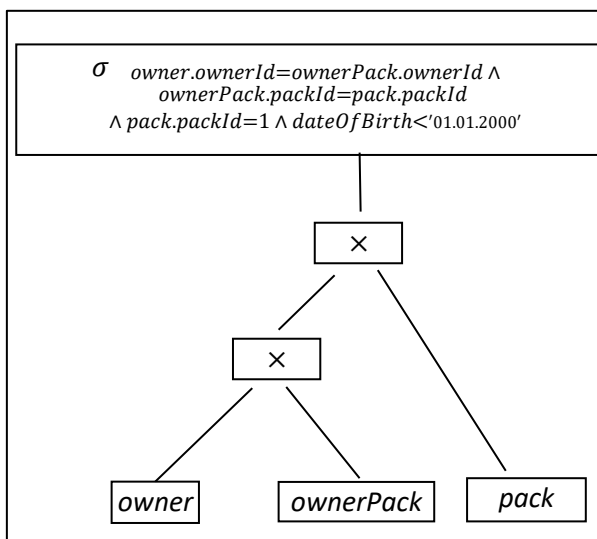
- Nacrtajte (na papir) stablo upita za **početni plan** izvršavanja upita pri čemu je redoslijed spajanja tablica određen redoslijedom kojim su tablice navedene u FROM dijelu SELECT naredbe.
- Nacrtajte (na papir) stablo upita **nakon** provedene heurističke optimizacije. Dovoljno je nacrtati samo konačno 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**.

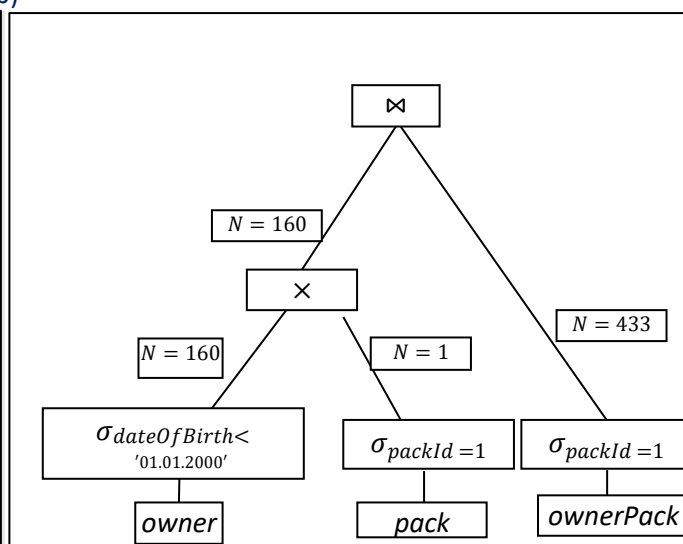
*Statistički podatci potrebni za izračun:*

owner	ownerPack	pack
N(owner) = 480	N(ownerpack) = 1299 V(ownerid, ownerpack) = 474 V(packId, ownerpack) = 3	N(pack) = 3

a)



b)



Prvo se spaja owner s pack.

Primijetiti:

```
WHERE ownerpack.packId = pack.packId
AND pack.packId = 1
```

potiskivanje selekcije (uvjet da je packId jednak 1) – na **obje** tablice: **pack** i **ownerPack**

potiskivanje selekcije  $\text{dateOfBirth} < '01.01.2000'$  – na **owner**

- **owner**.dateOfBirth < '01.01.2000'  $\rightarrow N(\text{owner}') = N(\text{owner}) / 3 = 480 / 3 = 160$
- **ownerPack**.packId = 1  $\rightarrow N(\text{ownerPack}') = N(\text{ownerPack}) / V(\text{packId}, \text{ownerPack}) = 1299 / 3 = 433$
- **pack**.packId = 1  $\rightarrow N(\text{pack}') = 1$  radi se o (PK)

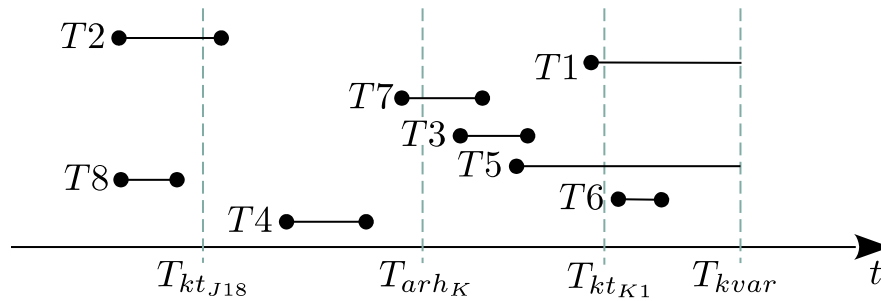
Međurezultati spajanja:

owner' i ownerPack'	(fk u ownerPack') $\Rightarrow N(\text{ownerPack}') = 433$
owner' i pack'	<b>Kartezijev</b> $\Rightarrow N(\text{owner}') \times N(\text{pack}') = 160 \times 1 = 160$
pack' i ownerPack'	(fk u ownerPack) $\Rightarrow N(\text{ownerPack}') = 433$



### 7. (3 boda) B grupa

U SUBP je došlo do pogreške računalnog sustava u trenutku  $T_{kvar}$ . Baza podataka je posljednji puta arhivirana (razina 0) u trenutku  $T_{arhK}$ . Odrađene su kontrolne točke  $T_{ktJ18}$  i  $T_{ktK1}$ . Dnevnici 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 sadržaj spremnika baze podataka u bazu podataka (fizički medij) ? Zašto je to tako?

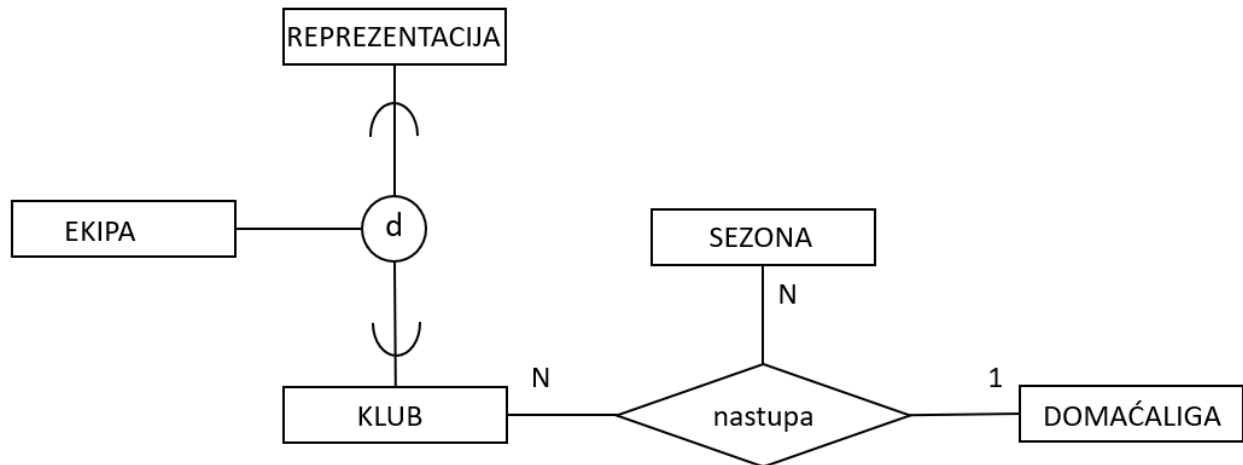
a) - lista za ponovno obavljanje = T6

- lista za poništavanje = T1, T5

b) TktK1 - zato jer se sadržaj spremnika baze podataka sprema u kontrolnim točkama

### 8. (3 boda) B 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.



EKIPIA (sifEkipa, nazivEkipa, vrstaEkipa)  
 KLUB (sifKlub, nazivKlub, godinaOsnutka)  
 DOMAĆALIGA (sifDomLiga, nazivDomLiga)  
 SEZONA (SifSezona, nazivSezona, godinaPocSezone)  
 REPREZENTACIJA (sifReprezentacija, nazivReprezentacija)

```

CREATE TABLE ekipa (
    sifEkipa ... PRIMARY KEY,
    nazivEkipa ...,
    vrstaEkipa ...);
  
```

```

CREATE TABLE reprezentacija (
    sifReprezentacija ... PRIMARY KEY,
    nazivReprezentacija ...,
    UNIQUE (sifEkipa)
    sifEkipa ... REFERENCES ekipa(sifEkipa));
  
```

```

CREATE TABLE klub (
    sifKlub ... PRIMARY KEY,
    nazivKlub ...,
    godinaOsnutka ...,
    UNIQUE (sifEkipa),
    sifEkipa ... REFERENCES ekipa(sifEkipa));
  
```

```

CREATE TABLE sezona (
    sifSezona ... PRIMARY KEY,
    nazivSezona ...,
    godinaPocSezona ...);
  
```

```

CREATE TABLE domaĆaLiga (
    sifDomLiga ... PRIMARY KEY,
    nazivDomLiga ...);
  
```

```

CREATE TABLE nastupa (
    sifKlub ... REFERENCES klub (sifKlub),,
    sifSezona ... REFERENCES sezona (sifSezona),,
    sifDomLiga ... REFERENCES domaĆaLiga (sifDomLiga),
    PRIMARY KEY (sifKlub, sifSezona));
  
```

## 9. (6 bodova) B grupa

Svake se godine u okviru Međunarodne vrtne izložbe FloraArt održava *Međunarodno natjecanje učenika srednjih škola* u aranžiranju cvijećem. Na natjecanje se mogu prijaviti škole diljem svijeta s time da svaka škola može prijaviti jednog učenika. Svaki učenik ima svojeg mentora.

Potrebno je oblikovati ER model baze podataka u kojoj će se evidentirati osnovni podatci o natjecanju i njihovim sudionicima. Za školu se evidentira jedinstvena šifra i jedinstveni naziv, država, grad (mjesto) i adresa (ulica i kućni broj u pripadnom mjestu).

Za državu se evidentira jedinstvena ISO oznaka i jedinstveni naziv. Za grad se evidentira država kojoj pripada, šifra koja je jedinstvena unutar pripadne države, naziv i poštanski broj (ne moraju biti jedinstveni).

Za osobe koje se prijavljuju na natjecanje evidentira se oznaka države i identifikacijski broj (jedinstven unutar pripadne države), ime, prezime i status (učenik/mentor).

Za učenika se dodatno evidentira šifra učenika (jedinstvena), datum rođenja, a za mentora šifra mentora (jedinstvena) i struka. Za struku se evidentira jedinstvena šifra i naziv. Jedan mentor može imati više struka.

Natjecanje ima svoju jedinstvenu šifru, naziv (ne mora biti jedinstven) i godinu u kojoj se odvija (jedinstvena).

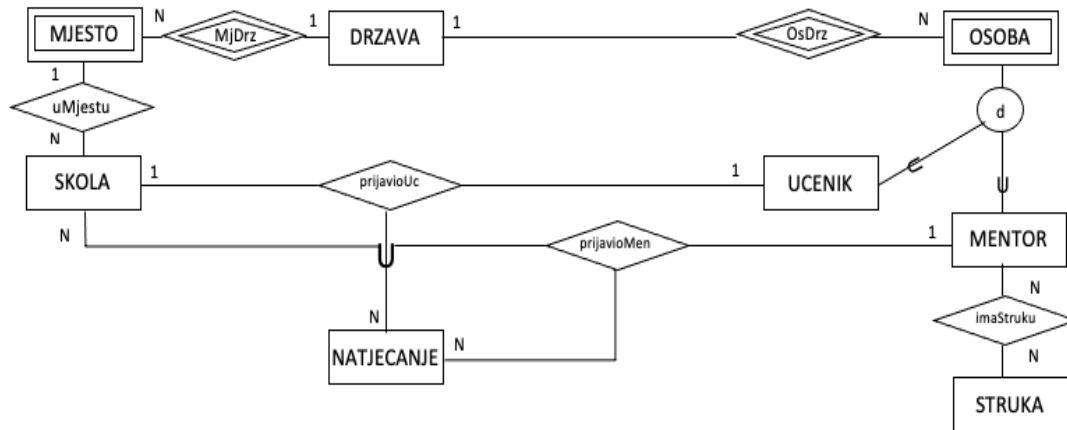
Jedan učenik može sudjelovati na natjecanjima više puta (u različitim godinama) i pri tome može zastupati razne škole (ako npr. učenik tijekom srednjoškolskog obrazovanja mijenja srednju školu).

Na natjecanje (koje se održava jedne godine) prijavljuje se učenik koji u toj godini pripada jednoj školi. Jednu školu jedan učenik može predstavljati na više natjecanja. Jednu školu na jednom natjecanju može predstavljati jedan učenik.

Također, na natjecanje se prijavljuje mentor koji u toj godini može voditi više učenika iz različitih škola. Jednu školu jedan mentor može predstavljati na više natjecanja (u različitim godinama). Jednu školu na jednom natjecanju (jedne godine) može predstavljati jedan mentor.

- a) Nacrtajte ER model baze podataka **na priloženom papiru** (nemojte koristiti alat ERDPlus jer jer nećete moći modelirati generalizaciju/specijalizaciju, niti ternarne veze), **nemojte crtati attribute**.
- 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.

Riešenie:



**Opisi entiteta** (primarni ključevi su podcrtani):

DRZAVA = ISOoznDrzava, nazDrzava

K1<sub>DRZAVA</sub> = ISOoznDrz; K2<sub>DRZAVA</sub> = nazDrzava

MJESTO = ISOozn, sifMjesto, postBroj, nazMjesto

K<sub>MJESTO</sub> = ISOoznDrz, sifMjesto

SKOLA = sifSkola, nazSkola

K1<sub>SKOLA</sub> = sifSkola; K2<sub>SKOLA</sub> = nazSkola

OSOBA = ISOozn, idOsoba, ime, prezime, status

K<sub>OSOBA</sub> = ISOoznDrz, idOsoba

UCENIK = sifUcenik, ISOozn, idOsoba

K1<sub>UCENIK</sub> = sifUcenik; K2<sub>UCENIK</sub> = ISOoznDrz, idOsoba;

MENTOR = sifMentor, ISOozn, idOsoba

K1<sub>MENTOR</sub> = sifMentor ; K2<sub>MENTOR</sub> = ISOoznDrz, idOsoba;

STRUKA = sifStruka, nazStruka

K<sub>STRUKA</sub> = sifStruka

NATJECANJE = sifNat, nazNat, godNat

K1<sub>NATJECANJE</sub> = sifNat; K2<sub>NATJECANJE</sub> = godNat

**Opisi veza** (primarni ključevi su podcrtani):

uDrzavi = ISOoznDrz, sifMjesto

K<sub>uDrzavi</sub> = ISOoznDrz, sifMjesto

uMjestu = sifSkola, ISOoznDrz, sifMjesto, ulicaKBr

K<sub>uMjestu</sub> = sifSkola

OsDrz = ISOoznDrz, osobald

K<sub>OsDrz</sub> = ISOoznDrz, osobald

imaStruku = sifMentor, sifStruka

K<sub>imaStruku</sub> = sifMentor, sifStruka

prijavioUc = sifNat, sifSkola, sifUcenik

K1<sub>prijavioUc</sub> = sifNat, sifSkola; K2<sub>prijavioUc</sub> = sifNat, sifUcenik

prijavioMen = sifNat, sifSkola, sifMentor

K<sub>prijavioMen</sub> = sifNat, sifSkola