Zadatak 1 (4 boda)

Zadatak riješite **jednom SQL naredbom**.

Za sve sezone serija čija je ocjena **veća od 50** te čiji naziv **sadrži dva uskličnika na bilo kojem mjestu u nazivu**, ispišite broj epizoda u sezoni, redni broj te sezone, naziv serije te ocjenu serije.

Zapise u rezultatu poredajte abecedno po nazivu serije te po rednom broju sezone uzlazno. Stupce nazvati prema danom predlošku (podaci u predlošku ne moraju odgovarati stvarnima).

numEp	seasonno	showtitle	showrating
11	1	Afronta! Facing It!	96.32
14	1	Comedy Bang! Bang!	82.46
…	…	…	…

Rješenje:

```
select max(episodeno) as numEp, seasonno, showtitle, showrating
from show
natural join showep
where showrating > 50 and showtitle LIKE '%!%!%'
group by seasonno, showtitle, showrating
order by showtitle, seasonno;
```

Zadatak 2 (4 boda)

Zadatak riješite **s dvije SQL naredbe**.

Izbrišite zapise o gledanju medijskog sadržaja za koje je pohranjeni interval gledanja kraći od jedne minute.

Zatim izbrišite sve zapise o ocjenjivanju sadržaja za medijske sadržaje za koje ne postoji pohranjeni interval gledanja sadržaja duži od jedne minute (uključivo). Vodite pritom računa o referencijskom integritetu tablica trackView i profileTrack i redoslijedu izvršavanja naredbi (potrebno je prvo obrisati zapise iz trackView).

Rješenje:

```
DELETE
FROM trackView
WHERE savedProgress < '1 minute'::interval;

DELETE
FROM profileTrack
WHERE trackid not in
    (SELECT trackid
     FROM trackView
     WHERE savedProgress >= '1 minute'::interval);
```

Zadatak 3 (6 bodova)

Uz pomoć jedne SQL naredbe za sve profile trenutno aktivnih `Premium` paketa koji su više medijskih sadržaja ocijenili negativno nego pozitivno, a također i više medijskih sadržaja ocijenili negativno nego što su ih ostavili neocijenjenima ispisati identifikator vlasnika, naziv profila, ukupan broj negativnih ocjena koje su dali medijskim sadržajima te vrijeme pokretanja zadnjeg medijskog sadržaja kojeg su negativno ocijenili.

Ispis poredati po broju loših ocjena silazno, a potom po vremenu zadnjeg pokretanja silazno. Stupce nazvati prema danom predlošku (podaci u predlošku ne moraju odgovarati stvarnima).

ownerid	profileName	negativeReviews	lastWatch
132	justinbieber	75	12.03.2019. 11:04:21
432	abewilliams	75	22.10.2017. 08:40:11
254	saulgoodman	72	05.10.2020. 09:32:09
...	...	...	...

Rješenje:

```
SELECT profile.ownerid, profile.profilename,
       COUNT(DISTINCT profiletrack.trackid) AS negativeReviews,
       MAX(viewStartDateTime) AS lastWatch
  FROM profile
  NATURAL JOIN profileTrack
  NATURAL JOIN owner
  NATURAL JOIN ownerPack
  NATURAL JOIN pack
  NATURAL JOIN trackView
 WHERE liked = -1
   AND packEndDateTime IS NULL
   AND packName LIKE '%Premium%'
GROUP BY profile.ownerid, profile.profilename
HAVING COUNT(DISTINCT profiletrack.trackid) > ALL ((SELECT COUNT(*)
                     FROM profiletrack pt
                    WHERE pt.ownerid = profile.ownerid
                      AND pt.profilename = profile.profilename
                      AND liked = 1)
          UNION
                     (SELECT COUNT(*)
                     FROM profiletrack pt
                    WHERE pt.ownerid = profile.ownerid
                      AND pt.profilename = profile.profilename
                      AND liked IS NULL))
ORDER BY negativeReviews DESC, lastWatch DESC;
```

Zadatak 4. (4 boda)

Za B-stablo stupnja 5 koje sadrži zapise s ključevima: 1,3,4,7,9,12,13,14,17,18,19,21 izračunati brojeve zapisa u korijenu, ostalim unutarnjim čvorovima i listovima te nacrtati B-stablo:

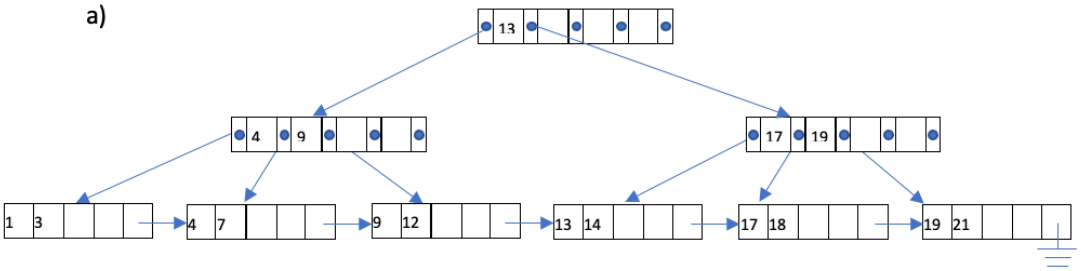
a) za minimalnu popunjenost svih čvorova

b) za maksimalnu popunjenost svih čvorova

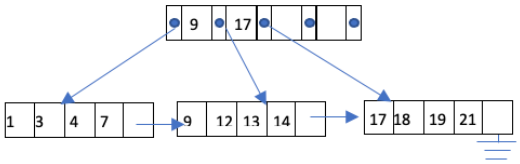
Rješenje:

	minimalna popunjenost	maksimalna popunjenost
korijen	2	5
ostali unutarnji čvorovi	3	5
listovi	2	4

a)



b)



Zadatak 5 (5 bodova)

Kad gledatelj počne gledati neki medijski sadržaj, u tablicu ***trackview*** unese se n-torka u kojoj se za atribut ***viewEndDateTime*** postavlja vrijednost NULL. Na primjer, uzmimo da se s profila ***derrickcurrie*** korisničkog računa gdje je vrijednost atributa ***ownerid*** jednaka 4 na datum ***1.9.2021.*** počeo gledati određeni medijski sadržaj:

```
INSERT INTO trackview

VALUES ('5BC96EE4C251FBB8AAC91BE8BEEF5EB5', '01.09.2021 11:07:16', 1314, 4,
'derrickcurrie', '00:00:00', NULL);
```

Prestankom gledanja tog medijskog sadržaja, odgovarajuća se n-torka ažurira:

```
UPDATE trackview

SET (savedProgress, viewEndDateTime) = ('00:32:02', '01.09.2021 11:39:18')

WHERE deviceid = '5BC96EE4C251FBB8AAC91BE8BEEF5EB5'

  AND viewStartDateTime = '07.07.2021 11:07:16';
```

Aktivna su samo ona gledanja medijskih sadržaja gdje u tablici **trackview** vrijednost atributa ***viewEndDateTime*** NULL.

Broj mogućih istovremenih gledanja medijskih sadržaja preko svih profilâ pojedinog korisničkog računa ograničen je paketom (***pack.streamNo***) vlasnika računa.

Napisati niz SQL naredbi za kreiranje svih potrebnih objekata kojima će se spriječiti da broj istovremenih gledanja medijskih sadržaja preko svih profilâ pojedinog korisničkog računa bude veći od vrijednosti atributa ***pack.streamNo*** u aktivnom paketu tog vlasnika računa.

Pri narušavanju opisanog ograničenja korisniku javiti sljedeću grešku:

Pogreška: Najveći broj sadržaja koji se mogu istovremeno gledati s korisnikovih profila je: <streamNo>

Umjesto ***<streamNo>*** potrebno je ispisati najveći mogući broj istovremenih gledanja sadržaja kako je definiran u aktivnom paketu tog vlasnika računa. Primjer:

Pogreška: Najveći broj sadržaja koji se mogu istovremeno gledati s korisnikovih profila je: 1.00

Rješenje:

```
CREATE FUNCTION chkViewNo() RETURNS TRIGGER AS
$$
   DECLARE
     p_streamNo              pack.streamNo%TYPE;

   BEGIN
      SELECT streamNo
      INTO p_streamNo
      FROM ownerpack natural join pack
      WHERE packEndDateTime IS NULL AND ownerid = NEW.ownerID;
```

```
      IF((SELECT COUNT(*) FROM trackview
      WHERE ownerid = NEW.ownerid
      AND viewEndDateTime IS NULL) = p_streamNo)
      THEN
     RAISE EXCEPTION 'Pogreška: Najveći broj sadržaja koji se mogu istovremeno
gledati s korisnikovih profila je:  %', p_streamNo;
      END IF;
    RETURN NEW;
  END;
 $$ language plpgsql;

CREATE TRIGGER insTrackView
BEFORE INSERT ON trackView
FOR EACH ROW EXECUTE FUNCTION chkViewNo();
```

Zadatak 6 (5 bodova)

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:

- Kreirati korisnika *bpeterson* s lozinkom 'pete12boris',
- Korisniku *bpeterson* dati ovlasti za:
    - pristup objektima sadržanima u shemi *public* s mogućnošću dodjeljivanja tog prava drugim korisnicima, te stvaranje novih objekata u shemi *public* bez mogućnosti dodjeljivanja tog prava drugim korisnicima,
    - pregled, unos, izmjenu i brisanje podataka o serijama (*show*) i epizodama serija (*showEp*),
    - pregled podataka o medijskim sadržajima (*track*), žanrovima (*genre)* i pripadnosti medijskog sadržaja žanru (*trackGenre*) s mogućnošću dodjele tog prava drugim korisnicima,

b) Nakon toga korisnik *bpeterson* želi delegirati dio svojeg posla za žanr drame (*Dramas*) korisniku *dqueen*. Napisati SQL naredbe kojima će korisnik *bpeterson* dati ovlasti korisniku *dqueen* za:

- pristup objektima u shemi *public*,
- pregled podataka o medijskim sadržajima (*track*), žanrovima (*genre*) i pripadnosti medijskog sadržaja žanru (*trackGenre*),
- pregled, upis i brisanje epizoda serije (*showEp*), te izmjenu podataka o **rednom broju sezone i epizode** (isto u tablici *showEp*) za serije u žanru drame (*Dramas*). Kreirajte sve objekte u shemi *public* potrebne za dodjelu opisanih ovlasti korisniku *dqueen*.

U odgovorima naznačite koji korisnik obavlja koje SQL naredbe.

Rješenje:

a)
admin:

```
create user bpeterson with password 'pete12boris';
grant usage on schema public to bpeterson with grant option;
grant create on schema public to bpeterson;
grant select on track,trackGenre,genre to bpeterson with grant option;
grant all on show to bpeterson;
grant all on showep to bpeterson;
```

b)
bpeterson:

```
grant usage on schema public to dqueen;
create view drama_shows as
select * from showep
where trackId in (select trackId
```

```
                  from track
                        natural join trackgenre
                        natural join genre
                  where genreName='Dramas')
with check option;

grant select on track,trackGenre,genre to dqueen;
grant select,insert,update(seasonNo,episodeNo),delete on drama_shows to
dqueen;
```

Zadatak 7 (4 boda)

Za bazu podataka *StreamFlix* vrijedi sljedeći skup funkcijskih zavisnosti:

F = {trackid, genrename → boxincome,

genreid → genrename

ownerid → firstname, lastname, dateofbirth}

Potrebno je ispitati vrijedi li funkcijska zavisnost genreid, trackid, ownerid → firstname, lastname, boxincome, genrename. Za svaki korak dokaza napisati pravilo koje se koristi.

Rješenje:

genreid → genrename i trackid, genrename → boxincome; genreid, trackid → boxincome (pseudotranzitivnost)

genreid, trackid → genrename (uvećanje)

genreid, trackid → boxincome i genreid, trackid → genrename; genreid, trackid → boxincome, genrename (unija)

genreid, trackid, ownerid → firstname, lastname, dateofbirth (uvećanje)

genreid, trackid, ownerid → firstname, lastname (dekompozicija)

genreid, trackid, ownerid → boxincome, genrename (uvećanje)

genreid, trackid, ownerid → firstname, lastname, boxincome, genrename (unija)

Funkcijska zavisnost vrijedi

Zadatak 8 (5 bodova)

Korištenjem Postgres-a i njegove implementacije upravljanja istodobnim pristupom navedite ***po jedan primjer*** za a) odnosno b) dio zadatka, koji koristi isključivo podatke iz tablice ***language,*** kojim ćete demonstrirati da **čitanje blokira pisanje** i pri tom:

a) dolazi do **potpunog zastoja** *(deadlock)*
b) **NE** dolazi do **potpunog zastoja** (*deadlock*)

Svaki od dva primjera treba sadržavati minimalan broj transakcija koje se paralelno odvijaju i minimalan broj SQL naredbi (ne nužno jednakih u oba primjera) potrebnih za demonstraciju traženog. Naznačite:

- 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
- u primjeru pod a) naznačite naredbu čijim je izvršavanjem došlo do potpunog zastoja. Objasnite zašto je do njega došlo.

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

Rješenje

a)

Obje transakcije koriste defaultnu razinu izolacije, ali razina izolacije nije važna jer FOR SHARE drži objekt zaključanim do kraja transakcije bez obzira na razinu izolacija.

	T1		T2
1	`BEGIN TRANSACTION;`	2	`BEGIN TRANSACTION;`

3	`SELECT *`
	`  FROM language`
	`WHERE langId = 1 FOR SHARE;`
		4	`SELECT *`
			`  FROM language`
			`WHERE langId = 1 FOR SHARE;`

5	`UPDATE language`
	`  SET langName = 'Albanski'`
	`WHERE langId = 1;`
		6	

		8	`UPDATE language`
			`  SET langName = 'Albanski'`
			`WHERE langId = 1;`

7	`COMMIT TRANSACTION;`		`COMMIT TRANSACTION;`

{3} T1 traži i **postavlja** ključ za **čitanje** (shared lock) nad n-torkom sa šifrom 1. Ključ traje do kraja transakcije.
{4} T2 traži i **postavlja** ključ za **čitanje** (shared lock) nad n-torkom sa šifrom 1. Ključ traje do kraja transakcije. To je moguće unatoč činjenici da i T1 ima S ključ nad istim objektom zato što su S ključevi kompatibilni.

{5} pokušava postaviti ključ za **pisanje** nad n-torkom sa šifrom 1, ali ne uspijeva jer T2 već ima ključ za čitanje (nekompatibilan s ključem za pisanje) nad tom istom n-torkom. Pisanje T1 je blokirano čitanjem T2. T1 čeka.
{6} pokušava postaviti ključ za **pisanje** nad n-torkom sa šifrom 1, 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. T2 čeka. Ovom naredbom je došlo da potpunog zastoja. Do njega je došlo jer je T1 blokirana od strane T2, a T2 je blokirana od strane T1 i nijedna transakcija ne može nastaviti s izvršavanjem dok se nešto ne dogodi s onom drugom.
b)

	T1		**T2**
1	`BEGIN TRANSACTION;`	2	`BEGIN TRANSACTION;`

3	`SELECT *`
	`  FROM language`
	`  WHERE langId = 1 FOR SHARE;`
		4	`UPDATE language`
			`  SET langName = 'Albanski'`

5	`COMMIT TRANSACTION;`	6	`WHERE langId = 1;`
			
			`COMMIT TRANSACTION;`

{3} T1 traži i **postavlja** ključ za **čitanje** (shared lock) nad n-torkom sa šifrom 1. Ključ traje do kraja transakcije.
{4} T2 pokušava postaviti ključ za **pisanje** nad n-torkom sa šifrom 1, 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. T2 čeka...
{5} T1 otpušta ključ za **čitanje** nad n-torkom sa šifrom 1. T2 obavlja naredbu 6 i 7.Nema potpunog zastoja.

Zadatak 9 (5 bodova)

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

Izvodi se upit:

```
SELECT *
FROM audioLang, track, trackView
WHERE track.trackId = audioLang.trackId
        AND track.trackId = trackView.trackId
        AND savedProgress < '1 hour 30 minutes'::INTERVAL
        AND releaseDate < '01.05.2017.'::DATE
        AND audioLangId = 13;
```

1. 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.
2. Provedite heurističku optimizaciju. Redoslijed spajanja tablica odredite temeljem procjene broja n-torki u međurezultatima.

   Navedite sve statističke izračune i izraze prema kojima je obavljena procjena broja n-torki u međurezultatima.

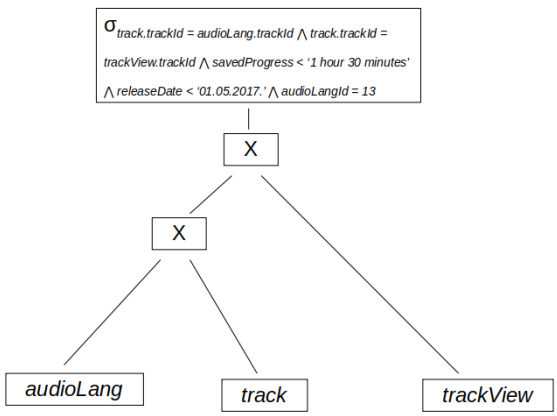
   Ako postoji više jednakovrijednih najboljih redoslijeda spajanja, odlučite se za jedan.

   Ako prilikom izračuna broja n-torki dobijete decimalnu vrijednost, zaokružite rezultat na prvi veći cijeli broj.

   Izračunate statističke podatke, postupak i izračun za procjenu broja n-torki **unesite u prostor za slobodni unos teksta ispod teksta zadatka**.
3. Nacrtajte (na papir) **konačno** stablo upita. U stablu upita naznačiti očekivani broj n-torki za svaku operaciju.

Rješenje

1.



2.

N(track) = 27176	N(trackView) = 63000 V(trackId, trackView) = 21429	N(audioLang) = 45000 V(audioLangId, audioLang) = 57 V(trackId, audioLang) = 26480

Selekciju *savedProgress < '1 hour 30 minutes'* treba potisnuti na **trackView**

- trackView.savedProgress < '1 hour 30 minutes' -> N(trackView') = N(trackView)/3 = 21000

Potiskivanje selekcije *releaseDate < '01.05.2017.'* na **track**

- track.releaseDate < '01.05.2017.' -> N(track') = N(track)/3 = 27176/3 ≈ 9059

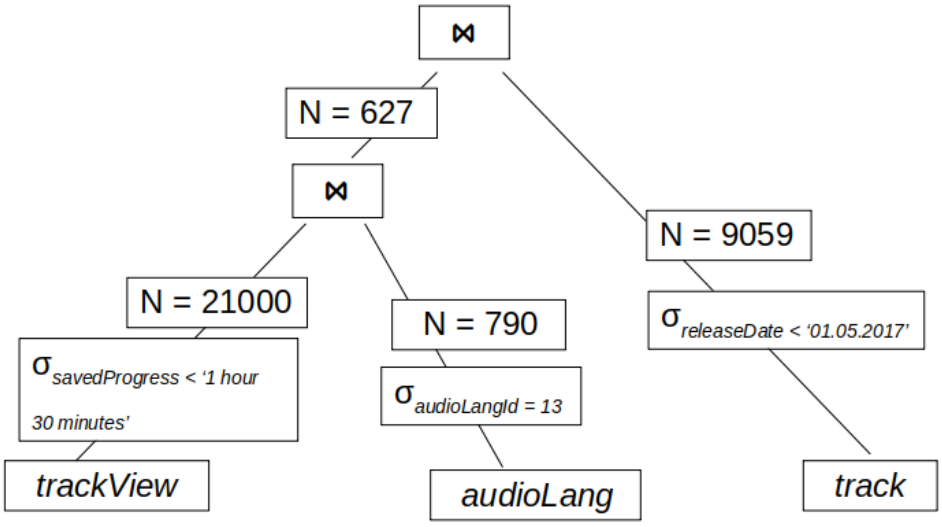
Potiskivanje selekcije *audioLangId = 13* na **audioLang**

- audioLang.audioLangId = 13 -> N(audioLang') = N(audioLang)/V(audioLangId, audioLang) = 45000/57 ≈ 790

Međurezultati spajanja:

- track' ⊳⊲ audioLang' – zajednički atribut trackId je primarni ključ u track' i strani ključ u audioLang' – svaka n-torka iz audiolang' može se spojiti s najviše jednom n-torkom iz track' -> N(track' ⊳⊲ audioLang') = 790
- **trackView' ⊳⊲ audioLang' - zajednički atribut trackId nije primarni ključ niti u jednoj od tablica -> N(trackView' ⊳⊲ audioLang') = (N(trackView') * N(audioLang'))/max(V(trackId, trackView), V(trackId, audioLang)) = (21000 * 790)/max(21429, 26480) ≈ 627**
- track' ⊳⊲ trackView' – zajednički atribut trackId je primarni ključ u track' i strani ključ u trackView' – svaka n-torka iz trackview' može se spojiti s najviše jednom n-torkom iz track' -> N(track' ⊳⊲ trackView') = 21000

3.



Zadatak 10 (8 bodova)

Potrebno je oblikovati ER model baze podataka za evidentiranje podataka o ocjenama pristupnika na prijemnom ispitu za sportsko visoko učilište.

Evidentiraju se podatci o osobama koje sudjeluju na prijemnom ispitu bilo kao pristupnici, bilo kao članovi povjerenstva.

Za osobe se bilježi jedinstveni identifikator, OIB, ime i prezime. Za osobu koja je pristupnik bilježi se dodatno i datum rođenja, spol te srednja škola koju je pristupnik završio. Pretpostavite da je svaki pristupnik završio točno jednu srednju školu. Za srednju školu se evidentira jedinstvena šifra i 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 osobu koja je član povjerenstva dodatno se bilježi uloga te osobe. Svaki član povjerenstva može imati više uloga, a svaku ulogu može obavljati više članova povjerenstva. Za ulogu se bilježi jedinstveni identifikator uloge i naziv uloge (kondicijski trener…).

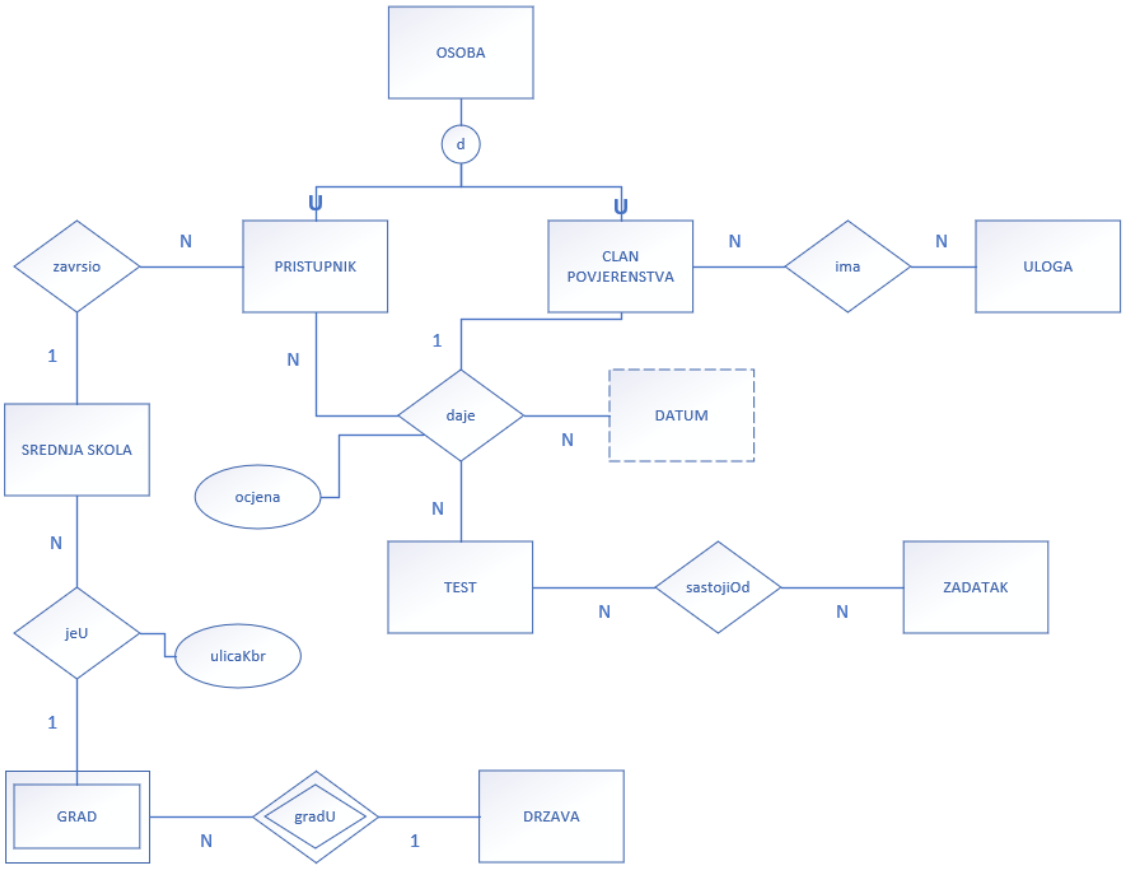
U bazi podataka se evidentiraju i testovi sposobnosti. Za test sposobnosti se evidentira jedinstveni identifikator i naziv testa te zadatci od kojih se test sastoji. Za zadatke se evidentira redni broj zadatka koji je jedinstven i težinski faktor zadatka. Svaki test sastoji se od više zadataka, a svaki zadatak može se naći u više različitih testova.

Na datume kada se na sportskom visokom učilištu održavaju prijemni ispiti, članovi povjerenstva pristupnicima daju testove sposobnosti. Jedan član povjerenstva isti test sposobnosti istom pristupniku može dati više puta – na više različitih datuma (npr. pristupnik je dvije godine za redom mogao doći na prijemni ispit, a oba puta je dobio isti test sposobnosti kod istog člana povjerenstva). Jednom pristupniku, jedan test na jedan datum daje samo jedan član povjerenstva. Jednom pristupniku na jedan datum isti član povjerenstva može dati više različitih testova sposobnosti. Jedan član povjerenstva isti test na isti datum može dati različitim pristupnicima. Uz sve navedeno bilježi se i ocjena koju je član povjerenstva dao pristupniku na **testu sposobnosti** tog datuma (napomena: član povjerenstva ocjenjuje test pristupnika, a ne pojedini zadatak).

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 primarne i alternativne ključeve). Svaki entitet (osim slabih entiteta) opisati **isključivo vlastitim atributima**. Nužno je da sve sheme zadovoljavaju 3NF.

Rješenje:



OSOBA – <u>idOsoba</u>, oib, ime, prezime $PK1_{osoba}$ = idOsoba, $PK2_{osoba}$= oib

PRISTUPNIK – <u>idOsoba</u>, datRod, spol

CLAN_POVJERENSTVA – <u>idOsoba</u>

ULOGA – <u>idUloga</u>, naziv

SREDNJA_SKOLA – <u>idSkola</u>, naziv

GRAD – <u>ISO, sifraGrad</u>, nazivGrad, pbr

DRZAVA – <u>ISO</u>, nazivDrzava $PK1_{drzava}$ = ISO, $PK2_{drzava}$ = nazivDrzava

TEST – <u>idTest</u>, nazivTest

ZADATAK – <u>rbr</u>, tezinskiFaktor

ima – <u>idOsoba, idUloga</u>

zavrsio – <u>idOsoba</u>, idSkola

jeU – idSkola, ISO, sifraGrad, ulicaKbr

gradU – ISO, sifraGrad

daje – idOsobaPristupnik, idTest, datum, idOsobaPovjerenstvo, ocjena

sastojiOd – idTest, rbr