

### Zadatak 1 (4 boda)

Zadatak riješite **jednom SQL naredbom**.

Internetski servis za gledanje medijskih sadržaja je odlučio nagraditi svoje korisnike s najvećim brojem gledanja različitih sadržaja.

Svim vlasnicima korisničkog računa, koji imaju trenutno aktivan paket, čiji naziv **nije** 'Premium', a čiji su profili započeli gledati isti ili različiti sadržaj ukupno **175 ili više** puta bez obzira jesu li sadržaj odgledali do kraja, treba ažurirati šifru paketa tako da taj mjesec koriste vrstu 'Premium'.

Napomena:

- Kod računanja ukupnog broja započetih gledanja različitih sadržaja, uzimaju se u obzir sva gledanja sadržaja. Nije potrebno provjeravati koji se paket plaćao u nekom trenutku.

### Rješenje:

```
update ownerpack
set packid = (select packid from pack where packname = 'Premium')
where packenddatetime is null
and packid in (select packid from pack where packname != 'Premium')
and ownerid in (
    select ownerid
    from trackview natural join profile natural join owner
    group by ownerid
    having count(viewstartdatetime) >= 175
    order by count(viewstartdatetime) desc
);
```

## Zadatak 2 (4 boda)

Zadatak riješite **jednom SQL naredbom**.

Uz pomoć jedne SQL naredbe za sve vlasnike računa koji tekuće godine navršavaju 18 godina ispisati ime i prezime (u istom stupcu - razdvojeni jednom prazninom), naziv korisničkog profila, naslove gledanih medijskih sadržaja za koje postoje prijevodi na njemačkom jeziku (naziv jezika je „German“) i ukupan broj gledanja svakog od tih sadržaja u protekle dvije kalendarske godine između siječnja i rujna (uključivo).

U rezultatima ne smije biti duplih redaka, a moraju se prikazati i korisnici i sadržaji za koje je broj gledanja sadržaja koji zadovoljavaju uvjet jednak 0.

Rezultate poredajte prema broju gledanja medijskog sadržaja silazno.

Niže je prikazan primjer ispisa (**obratiti pažnju na imena stupaca**) s podskupom stvarnih rezultata:

fullname	profilename	tracktitle	viewcount
...	...	...	...
Catherine Preece	catherinepreece	Oktoberfest: Beer & Blood S01E11	1
...	...	...	...
Rizwan Merrill	CarolBell	Take Me Home Tonight	0
...	...	...	...

### Rješenje:

```
SELECT DISTINCT firstName || ' ' || lastName as fullName, profileName, trackTitle,
               (SELECT COUNT(trackId)
                FROM trackview
                WHERE trackView.trackId = track.trackId
                  AND trackview.ownerId = profile.ownerId
                  AND trackview.profileName = profile.profileName
                  AND (EXTRACT(month FROM viewstartdatetime) BETWEEN 1 and 9 AND extract(year
FROM viewstartdatetime) >= 2020)) as viewCount
FROM profile NATURAL JOIN owner NATURAL JOIN track
      NATURAL JOIN trackView
      JOIN subtitle ON subtitle.trackid = track.trackid
      JOIN language ON subtitle.subtitlelangid = language.langid
WHERE EXTRACT(YEAR FROM owner.dateOfBirth) = EXTRACT(YEAR FROM CURRENT_DATE) - 18
      AND langname = 'German'
ORDER BY viewcount DESC
```

### Zadatak 3 (6 bodova)

Napišite **jednu SQL naredbu** koja će za svaku epizodu švedske serije (nativni jezik audiozapisa ima naziv 'Swedish') a koja ima barem jednu pozitivnu ocjenu ispisati naziv serije, broj sezone, broj epizode te nazive i e-mail adrese svih profila koji su joj u tekućoj godini barem jednom pristupili (gledano početak ili završetak gledanja). Za epizode kojima nijedan profil nije pristupao u tekućoj godini za naziv profila i e-mail adresu staviti *null*.

Ispis poredati po nazivu serije abecedno, potom po broju sezone uzlazno, broju epizode uzlazno i konačno imenu profila abecedno. Stupce nazvati prema danom predlošku (podaci u predlošku ne moraju odgovarati stvarnima). U rezultatu se ne smiju pojavljivati duplikati.

showtitle	seasonno	episodeno	profilename	pemail
Aaron the Suave Swede	1	3	jackblack	jackblack@aol.com
Aaron the Suave Swede	1	3	zanegrey	zingri@yahoo.com
Aaron the Suave Swede	1	4	null	null
Byron the Ugly Swede	1	1	null	null
...				

### Rješenje:

```
SELECT DISTINCT showTitle, seasonNo, episodeNo, profile.profilename, pemail
FROM show NATURAL JOIN showep NATURAL JOIN track
      NATURAL JOIN profiletrack
      NATURAL JOIN audiolang
      JOIN language ON language.langid = audiolang.audiolangid
      LEFT JOIN trackview ON trackview.trackid = track.trackid
      AND ((EXTRACT(YEAR FROM viewStartDateTime) = EXTRACT(YEAR FROM
CURRENT_DATE)
              OR (EXTRACT(YEAR FROM viewEndDateTime) = EXTRACT(YEAR FROM
CURRENT_DATE))))
      LEFT JOIN profile ON trackview.ownerid = profile.ownerid AND
trackview.profilename = profile.profilename
WHERE langname = 'Swedish' AND isNative = 1 AND liked = 1
ORDER BY showTitle, seasonNo, episodeNo, profilename;
```

#### 4. zadatak (4 boda)

U sustavu se obavlja inkrementalno arhiviranje. Uz arhivu pohranjuje se i pripadni dnevnik izmjena.

1.9.2021. u 00:15:00 napravljena je arhiva razine 0;

2.9.2021. u 00:15:00 napravljena je arhiva razine 2;

3.9.2021. u 00:15:00 napravljena je arhiva razine 2;

4.9.2021. u 00:15:00 napravljena je arhiva razine 2;

5.9.2021. u 00:15:00 napravljena je arhiva razine 1;

6.9.2021. u 00:15:00 napravljena je arhiva razine 2;

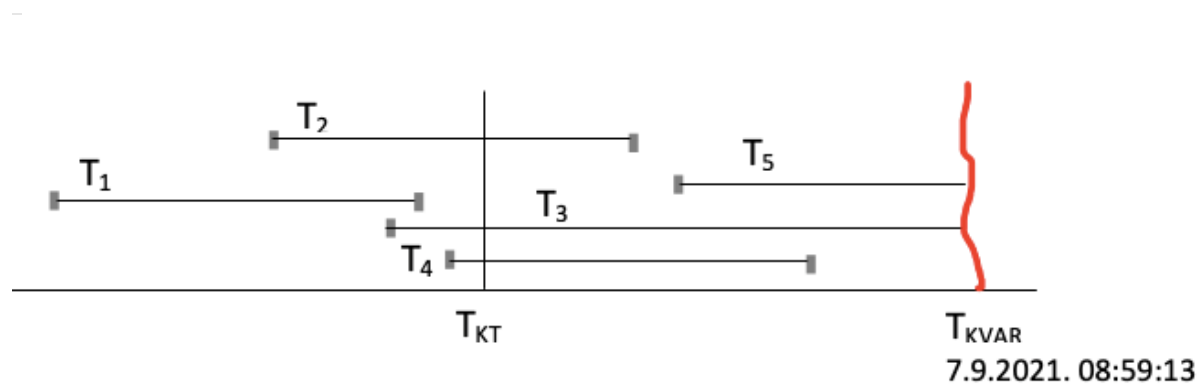
7.9.2021. u 00:15:00 napravljena je arhiva razine 2;

**Dana 7.9.2021. u 8:59:13 nastao je kvar medija.**

Tekući dnevnik izmjena pohranjen je na drugom mediju (disku) na kojem se ne nalazi baza podataka i ostao je sačuvan. Nakon ponovne uspostave sustava potrebno je obnoviti bazu podataka.

a) Opišite postupak obnove, navodeći arhive i dnevnike koji će sudjelovati u obnovi te njihov redoslijed.

b) Zadnji odsječak tekućeg dnevnika sadrži sljedeće transakcije:



$T_{KT}$  je vrijeme zadnje kontrolne točke.

$T_{KVAR}$  je vrijeme kvara.

Navedite transakcije čije izmjene NEĆE biti sadržane u obnovljenoj bazi podataka i objasnite zašto.

**Rješenje:**

a) Arhiva razine 0 od 1.9.;

arhiva razine 1 od 5.9.;

arhiva razine 2 od 7.9.;

dnevnik izmjena nastao nakon arhive 2 od 7.9.

b)  $T_3$  i  $T_5$  - nisu završile i njihove se izmjene poništavaju

### Zadatak 5 (5 bodova)

U internetskom servisu za gledanje medijskih sadržaja vrijedi pravilo da se određenom medijskom sadržaju mogu pridružiti najviše 3 žanra. Napisati niz SQL naredbi za kreiranje svih potrebnih objekata kojima će se pri **unosu** n-torke u relaciju **trackgenre** spriječiti pridjeljivanje novog žanra određenom medijskom sadržaju (**track**) u slučaju da taj medijski sadržaj ima već definirana 3 žanra.

Pri narušavanju opisanog ograničenja korisniku javiti sljedeću grešku:

***'Pogreška: Ne može se dodati još jedan žanr za medijski sadržaj: <trackid> <trackTitle>'.***

Umjesto <trackid> potrebno je ispisati šifru medijskog sadržaja, a umjesto <trackTitle> njegov naziv.

Primjer:

***Pogreška: Ne može se dodati još jedan žanr za medijski sadržaj: 30 Forbidden Planet***

U svim ostalim slučajevima n-torka mora biti uspješno upisana u relaciju **trackgenre**.

### Rješenje:

```
CREATE FUNCTION chkViseOdTriZanra() RETURNS TRIGGER AS
$$
    BEGIN
        IF(SELECT COUNT(genreid) as brojZanrova
           FROM trackgenre
           GROUP by trackid
           HAVING trackid = NEW.trackid)=3
        THEN RAISE EXCEPTION 'Pogreška: Ne može se dodati još jedan žanr za
medijski sadržaj: % %', NEW.trackid, (SELECT trackTitle FROM track
           WHERE trackid = NEW.trackid);
        END IF;
        RETURN NEW;
    END;
$$ language plpgsql;

CREATE TRIGGER insTrackGenre
BEFORE INSERT ON trackGenre
FOR EACH ROW EXECUTE FUNCTION chkviseOdTriZanra();

/*
INSERT INTO trackgenre VALUES (30, 7);
        Pogreška: Ne može se dodati još jedan žanr za medijski sadržaj: 30 Forbidden Planet
INSERT INTO trackgenre VALUES (29, 7);
*/
```

## 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;
```

Napisati SQL naredbe kojima će administrator sustava:

- a) Kreirati korisnika *statmaster* s lozinkom 'stats123',
- b) Korisniku *statmaster* dati ovlasti za:
  - a. pristup objektima sadržanima u shemi *public* s mogućnošću dodjeljivanja tog prava drugim korisnicima,
  - b. pregled podataka o medijskim sadržajima (*track*), žanrovima (*genre*) i pripadnosti medijskih sadržaja žanrovima (*trackGenre*) s mogućnošću dodjeljivanja tog prava drugim korisnicima,
  - c. ažuriranje ocjene medijskih sadržaja (*trackRating*),
  - d. pregled statistike ocjena medijskih sadržaja, koja treba sadržavati prosječnu ocjenu svih ocijenjenih medijskih sadržaja zaokruženu na dvije decimale, grupiranu po nazivima žanrova, te godinama i mjesecima izdavanja medijskog sadržaja. Statistika treba biti poredana silazno po godinama, a zatim silazno po mjesecima. Kreirajte sve objekte u shemi *public* potrebne za dodjelu opisanih ovlasti korisniku *statmaster*, s mogućnošću dodjeljivanja tog prava drugim korisnicima.

Primjer statistike:

year	month	genreName	avg_rating
2021	4	Dramas	87.45
2021	4	Horror	45.67
2021	3	Dramas	86.12
2021	3	Comedies	96.71
2021	3	Horror	49.77
...	...	...	...

### Rješenje:

```
create user statmaster with password 'stats123';
```

```
grant usage on schema public to statmaster with grant option;
```

```
grant select on track,genre,trackGenre to statmaster with grant option;
```

```
grant update(trackRating) on track to statmaster;
```

```
create view ratingStats as
```

```
select extract(year from releaseDate) as year,extract(month from releaseDate)
```

```
as month, genreName, round(avg(trackRating),2) as avg_rating
```

```
from track natural join trackGenre natural join genre
```

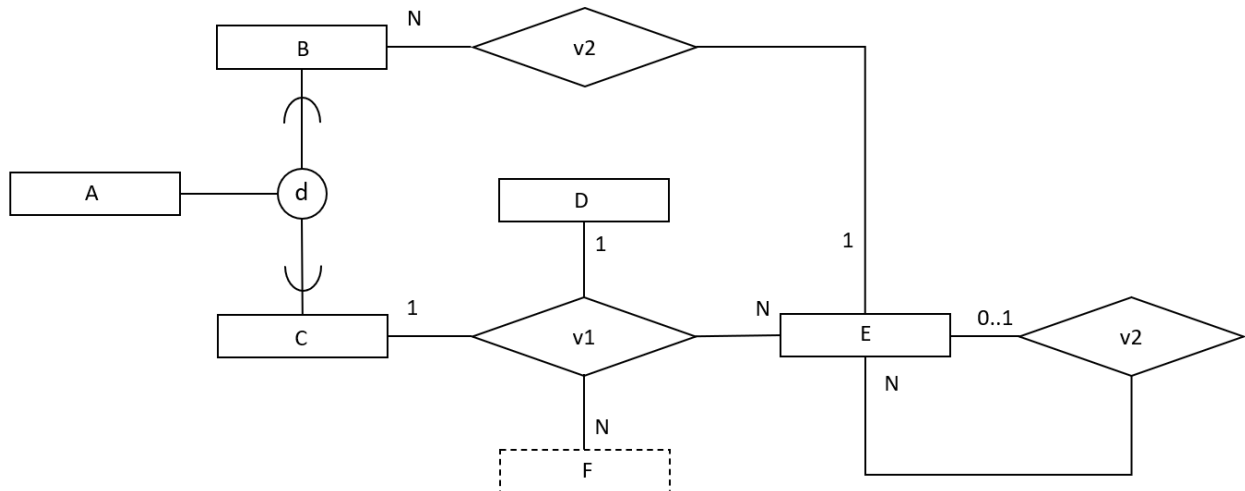
```
group by year,month,genreName
```

```
order by year desc,month desc;
```

```
grant select on ratingStats to statmaster with grant option;
```

### Zadatak 7. (4 boda)

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.



A = A1, A2, A3  
B = B1, B2  
C = C1, C2, C3  
D = D1, D2, D3  
E = E1, E2

### Rješenje:

```
CREATE TABLE A (A1 ... PRIMARY KEY,  
                A2 ...,  
                A3 ...);
```

```
CREATE TABLE B (B1 ... PRIMARY KEY,  
                B2 ...,  
                A1 ... UNIQUE REFERENCES A (A1),  
                E1 REFERENCES E (E1)  
                );
```

```
CREATE TABLE C (C1 ... PRIMARY KEY,  
                C2 ...,  
                C3 ...,  
                A1 ... UNIQUE REFERENCES A (A1)  
                );
```



```
CREATE TABLE D (D1 ...,  
                D2 ...,  
                D3 ...,  
                PRIMARY KEY (D1, D2) );
```

```
CREATE TABLE E (E1 ... PRIMARY KEY,  
                E2 ...,  
                E1refl ...REFERENCES E (E1)  
                );
```

```
CREATE TABLE v1 (A1 ... REFERENCES A(A1),  
                  C1 ... REFERENCES C(C1),  
                  D1 ...,  
                  D2 ...,  
                  E1   REFERENCES E (E1),  
                  F ,  
                  PRIMARY KEY (C1, E1, F),  
                  UNIQUE (D1, D2, E1, F),  
                  FOREIGN KEY (D1, D2) REFERENCES D (D1, D2));
```

### Zadatak 8 (5 bodova)

Navedite **po jedan primjer** za **a)** odnosno **b)** dio zadatka, koji koristi isključivo podatke iz tablice **show**, kojima ćete demonstrirati da se u transakciji pojavljuje problem:

- a) **prljavo čitanje**
- b) **neponovljivo čitanje**

Primjere izraditi uz pretpostavku korištenja relacijskog sustava za upravljanje bazama podataka koji implementira razine izolacije i mogućnost pojavljivanja karakterističnih problema istodobnog pristupa kao što je to predviđeno standardom SQL-92.

**Napomena:** PostgreSQL nije u potpunosti u skladu s SQL-92 standardom, ali neki drugi sustavi jesu. Zbog toga se u ovom zadatku u primjerima ne podrazumijeva korištenje PostgreSQL.

Svaki od dva primjera treba sadržavati minimalan broj transakcija koje se paralelno odvijaju i minimalan broj SQL naredbi potrebnih za demonstraciju traženog. Naznačite:

- razine izolacije koje u primjerima koristite
- redoslijed izvršavanja naredbi u transakcijama (globalni redoslijed za sve transakcije koje u primjeru koristite)
- u kojoj od transakcija se pojavljuje traženi problem, čime je uzrokovan i kako se to vidi.

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

### Rješenje:

Razina izolacije T2 u oba primjera nije važna tj. može biti bilo koja.

a)

Razina izolacije T1 mora biti READ UNCOMMITTED jer prljavo čitanje nije moguće kod drugih razina izolacije.

T1		T2	
1	BEGIN TRANSACTION;	2	BEGIN TRANSACTION;
3	SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;	4	UPDATE show SET showRating = 50 WHERE showId = 70;
5	SELECT * FROM show WHERE showId = 70;	7	COMMIT TRANSACTION;
6	COMMIT TRANSACTION;		

Problem prljavog čitanja se pojavljuje u transakciji T1 pri izvršavanju naredbe pod rbr 5, a sastoji se u tome da je pročitala nepotvrđeni podatak budući da je T2 za vrijeme izvršavanje naredbe pod 5 još uvijek u tijeku.

b)

Razina izolacije T1 može biti READ UNCOMMITTED ili READ COMMITTED.

T1		T2	
1	BEGIN TRANSACTION;	2	BEGIN TRANSACTION;
3	SET TRANSACTION ISOLATION LEVEL READ COMMITTED;		
4	SELECT * FROM show WHERE showId = 70;	5	UPDATE show . SET showRating = 50 WHERE showId = 70;
7	SELECT * FROM show WHERE showId = 70;	6	COMMIT TRANSACTION;
8	COMMIT TRANSACTION;		

Problem neponovljivog čitanja se pojavljuje u transakciji T1, a sastoji se u tome da ponovljenim čitanjem istog objekta (Select naredba pod rbr 4 i 7) T1 pročita istu n-torku ali je njen sadržaj drugačiji u ponovljenom čitanju. Neponovljivo čitanje je posljedica potvrđene izmjene koju je obavila transakcija T2 naredbom 5 i potvrdila naredbom 6 i to nakon izvršavanja naredbe 4 transakcije T1.

	showId	showName	showRating
{4}	70	7 (Seven)	18.04
{7}	70	7 (Seven)	50.00

### **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 showEp, track, trackGenre  
WHERE track.trackId = showEp.trackId  
      AND track.trackId = trackGenre.trackId  
      AND episodeNo < 6  
      AND genreId = 5  
      AND releaseDate < '01.01.2020.'::DATE;
```

- 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.
- 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.

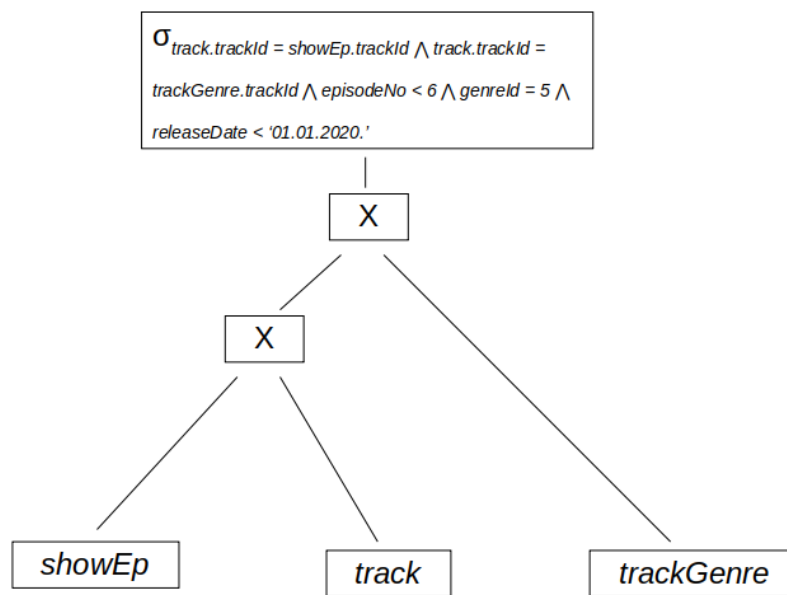
Ako prilikom izračuna broja n-torki dobijete decimalnu vrijednost, zaokružite rezultat na prvi veći cijeli broj.

Izračunate statističke podatke, postupak i izračun za procjenu broja n-torki **unesite u prostor za slobodni unos teksta ispod teksta zadatka**.

- Nacrtajte (na papir) **konačno** stablo upita. U stablu upita naznačiti očekivani broj n-torki za svaku operaciju.

## Rješenje:

1.



2.

N(track) = 27176	N(trackGenre) = 40668 V(genreId, trackGenre) = 27	N(showEp) = 22076
------------------	--	-------------------

Potiskivanje selekcije *genreId* = 5 na **trackGenre**

- trackGenre.genreId = 5 → N(trackGenre') = N(trackGenre)/V(genreId, trackGenre) = 40668/27 ≈ 1507

Potiskivanje selekcije *releaseDate* < '01.01.2020.' na **track**

- track.releaseDate < '01.01.2020.' → N(track') = N(track)/3 = 27176/3 ≈ 9059

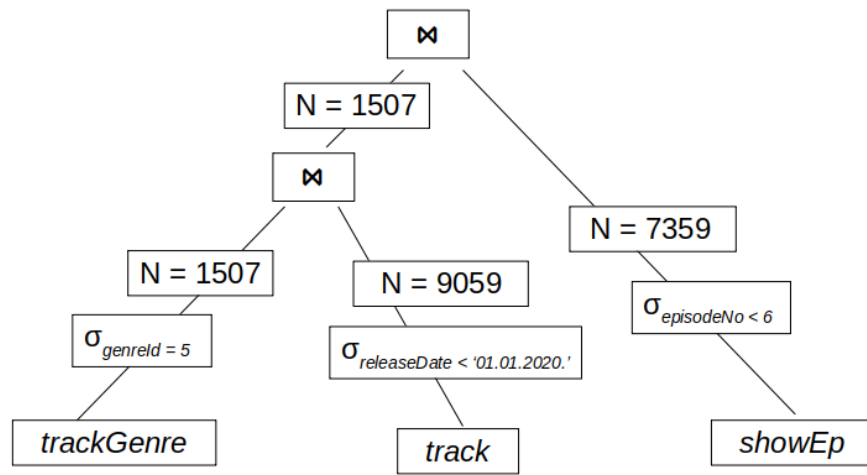
Potiskivanje selekcije *episodeNo* < 6 na **showEp**

- showEp.episodeNo < 6 → N(showEp') = N(showEp)/3 = 22076/3 ≈ 7359

Međurezultati spajanja:

- track' ▷ trackGenre'** - zajednički atribut *trackId* je primarni ključ u track' i strani ključ u trackGenre' - svaka n-torka iz trackGenre' može se spojiti s najviše jednom n-torkom iz track' → N(track' ▷ trackGenre') = 1507
- showEp' ▷ track'** - zajednički atribut *trackId* je primarni ključ u track' i strani ključ u showEp' - svaka n-torka iz showEp' može se spojiti s najviše jednom n-torkom iz track' → N(showEp' ▷ track') = 7359
- showEp' ▷ trackGenre'** - zajednički atribut *trackId* je primarni ključ u showEp' i strani ključ u trackGenre' - svaka n-torka iz trackGenre' može se spojiti s najviše jednom n-torkom iz showEp' → N(showEp' ▷ trackGenre') = 1507

3.



### Zadatak 10 (8 bodova)

Potrebno je oblikovati ER model baze podataka za internetsku kupovinu karata za zabavni park (nalik Gardalandu).

Zabavni park se sastoji od svjetova (npr. Zabavni svijet, Vodeni svijet ... ). Svaki svijet određen je jedinstvenim identifikatorom i nazivom, a dodatno se za svaki svijet bilježi i kratak opis.

Svaki svijet nudi više različitih sadržaja koji mogu biti ili uslužni ili zabavni ili edukativni. Budući da su lokacijski odijeljeni, pojedini sadržaj može pripadati samo jednom svijetu. Svaki sadržaj određen je identifikatorom koji je jedinstven unutar svijeta kojemu pripada.

Za uslužni sadržaj dodatno se bilježi naziv (npr. restoran *Dinosauri*) i vrsta usluge koju nudi. Neki uslužni sadržaj može nuditi više usluga, a za uslugu se bilježi jedinstveni identifikator i naziv (pizze, piće, slastice...)

Za zabavni sadržaj dodatno se bilježi naziv, opis i vrsta. Vrsta zabavnog sadržaja određena je jedinstvenim identifikatorom i nazivom (adrenalin, avantura i slično). Pojedini zabavni sadržaj može biti samo jedne vrste. Također, za svaki zabavni sadržaj bilježe se eventualna ograničenja na visinu i ograničenja na dob.

Za edukativni sadržaj dodatno se bilježi opis i jezici na kojima se edukacija nudi, a jezik je određen jedinstvenim identifikatorom i nazivom. Edukativni sadržaj se može nuditi na više jezika.

Cijena ulaznice za pojedini svijet ovisi o svijetu za koji se kupuje i o tipu ulaznice. Tip ulaznice određen je identifikatorom i nazivom (dječja, odrasla, obiteljska).

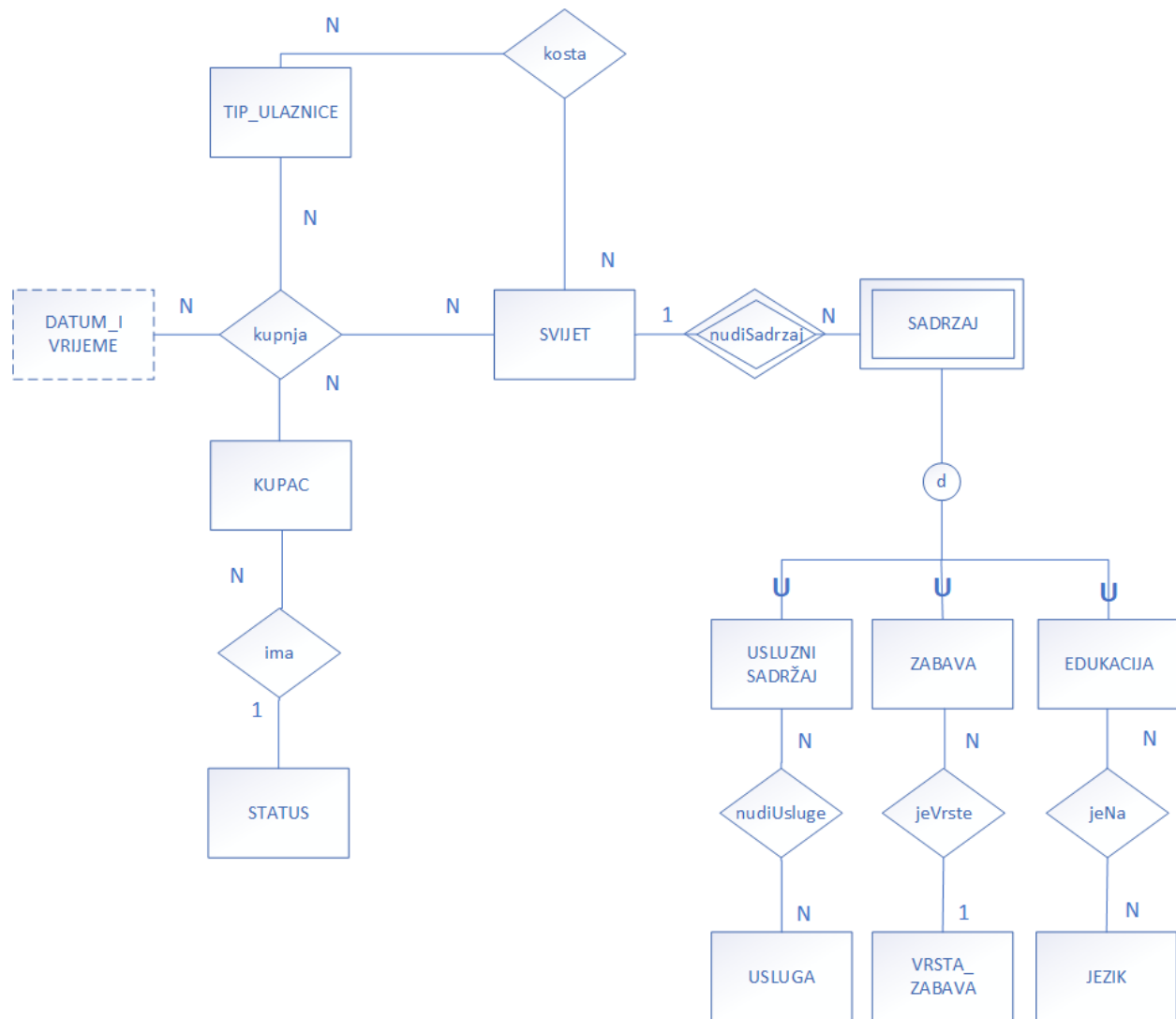
Kupac je određen jedinstvenim korisničkim imenom, a dodatno se bilježe lozinka i status koji odgovara vjernosti tog kupca. Status je određen jedinstvenim imenom (zlatni, srebrni, brončani, ostalo) te se bilježi iznos popusta za taj status.

Pojedini kupac prilikom internetske kupnje u nekom trenutku može za jednu vrstu svijeta kupiti više tipova ulaznica. Kupac može u jednom trenutku kupiti isti tip ulaznice za više svjetova. U jednom trenutku, jedan tip ulaznice za jedan svijet može kupiti više kupaca. Jedan kupac jedan tip ulaznice za isti svijet može kupiti u više vremenskih trenutaka.

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:



SVIJET – idSvijet, opisSvijet

SADRZAJ – idSadrzaj, idSvijet

USLUZNI\_SADRZAJ – idSadrzaj, naziv

ZABAVA - idSadrzaj, naziv, opis, ogranicenjeVisina, ogranicenjeGodine

EDUKACIJA – idSadrzaj, opis

USLUGA – idUsluga, nazivUsluga

VRSTA\_ZABAVA – idVrZabave, nazivZabave

JEZIK – idJezik, nazivJezik



TIP\_ULAZNICE – idTipUlaznica, nazivUlaznica

KUPAC – korisnickolme, lozinka

STATUS – nazivStatus, popust

kupnja – korisnickolme, idSvijet, idTipUlaznica, DATUM\_VRIJEME

kosta – idTipUlaznica, idSvijet, cijena

ima – korisnickolme, idStatus

nudiSadrzaj - idSadrzaj, idSvijet

nudiUsluge – idSadrzaj, idSvijet, idUsluga

jeVrste – idSadrzaj, idVrZabave

jeNa – idSadrzaj, idJezik