

**Međuispit iz Baza podataka  
(30 bodova)**

21. travnja 2021.

**1. Zadatak (3 boda)**

a) Ispišite identifikator medijskog sadržaja, naslov i informaciju o trajanju za *film* koji su nastavak nekog ranijeg filma te kojima naslov:

- sadrži određeni član 'the', ali njime ne počinje,

ili

- započinje neodređenim članom 'a',

bez obzira na velika i mala slova.

Informaciju o trajanju je potrebno prikazati na sljedeći način:

- ako je trajanje filma manje od 1 sata, u rezultatu treba pisati 'kratko';
- ako film traje barem 1 sat, a manje od 2 sata, u rezultatu treba pisati 'srednje';
- ako film traje 2 sata ili više, u rezultatu treba pisati 'dugo'.

```
SELECT trackId, trackTitle,
       CASE
         WHEN duration < '1 hour'::interval THEN 'kratko'
         WHEN duration >= '1 hour'::interval AND
              duration < '2 hour'::interval THEN 'srednje'
         WHEN duration >= '2 hour'::interval THEN 'dugo'
       END
FROM track
   NATURAL JOIN movie
WHERE movie.prevMovieId IS NOT NULL
      AND (lower(trackTITLE) LIKE '% the %' AND lower(trackTITLE) NOT LIKE 'the %'
           OR lower(trackTITLE) LIKE 'a %')
```

b) Ispišite identifikator medijskog sadržaja, naslov i doba objave za **film** koji su nastavak nekog ranijeg filma te kojima naslov:

- započinje određenim članom 'the',

ili

- sadrži neodređeni član 'a', ali njime ne počinje,

bez obzira na velika i mala slova.

Doba objave je potrebno prikazati na sljedeći način:

- ako je film objavljen u periodu od 1. 1. 2000. do 31. 12. 2009. (uključene obje granice), u rezultatu treba pisati 'dvijetisućite';
- ako je film objavljen u periodu od 1. 1. 2010. do 31. 12. 2019. (uključene obje granice), u rezultatu treba pisati 'dvijeidesete';
- ako je film objavljen na datum 1. 1. 2020. (uključivo) ili kasnije, u rezultatu treba pisati 'dvijeidvadesete i nakon';
- ako datum objave filma ne odgovara ranijim kriterijima, u rezultatu treba pisati 'drugo'.

```
SELECT trackid, tracktitle,
       CASE
         WHEN releaseDate BETWEEN '01.01.2000' AND
           '31.12.2009' then 'dvijetisućite'
         WHEN releaseDate BETWEEN '01.01.2010' AND
           '31.12.2019' then 'dvijeidesete i nakon'
         WHEN releaseDate >='01.01.2020' then 'dvijeidvadesete'
         ELSE 'drugo'
       END
FROM track
   NATURAL JOIN movie
WHERE prevmovieid IS NOT NULL
   AND (LOWER(tracktitle) LIKE 'the %' OR (LOWER(tracktitle) LIKE '% a %'
     AND LOWER(tracktitle) NOT LIKE 'a %'));
```

c) Ispišite identifikator medijskog sadržaja, naslov i informaciju o ocjeni (*trackRating*) za **filmove** kojima je zarada od prodaje kino ulaznica bila veća od 100 000 000 dolara te im naslov sadrži riječ 'in', ali njome ne počinje, bez obzira na velika i mala slova.

Informaciju o ocjeni je potrebno prikazati na sljedeći način:

- ako je ocjena filma ispod 50%, u rezultatu treba pisati 'loš';
- ako je ocjena filma barem 50%, a manja od 80%, u rezultatu treba pisati 'dobar';
- ako je ocjena filma 80% i više, u rezultatu treba pisati 'odličan';
- ako ocjena filma nije poznata, u rezultatu treba pisati 'nepoznato'.

```
SELECT trackId, trackTitle,
       CASE
         WHEN trackRating < 50 THEN 'loš'
         WHEN trackRating >= 50 AND trackRating < 80 THEN 'dobar'
         WHEN trackRating >= 80 THEN 'odličan'
         ELSE 'nepoznato'
       END
FROM track
     NATURAL JOIN movie
WHERE movie.boxIncome > 100000000
      AND (lower(trackTITLE) LIKE '% in %' AND lower(trackTITLE) not LIKE 'in %')
```

d) Ispišite identifikator i naslov medijskog sadržaja (filma), te informaciju o zaradi od prodaje kino ulaznica za **filmove** koji su nastavak nekog ranijeg filma te kojima naslov započinje ili završava slovom 'r', bez obzira na velika i mala slova.

Informaciju o zaradi je potrebno prikazati na sljedeći način:

- ako je zarada filma manja od 100 000 000 dolara, u rezultatu treba pisati 'slaba';
- ako je zarada filma barem 100 000 000 dolara, a manja od 200 000 000 dolara, u rezultatu treba pisati 'dobra';
- ako je zarada filma 200 000 000 dolara ili više, u rezultatu treba pisati 'odlična';
- ako iznos zarade filma nije poznat, u rezultatu treba pisati 'nepoznato'.

```
SELECT trackId, trackTitle,
       CASE
         WHEN boxIncome < 100000000 THEN 'slaba'
         WHEN boxIncome >= 100000000 AND boxIncome < 200000000 THEN 'dobra'
         WHEN boxIncome >= 200000000 THEN 'odlična'
         ELSE 'nepoznato'
       END
FROM track
     NATURAL JOIN movie
WHERE movie.prevMovieId IS NOT NULL
      AND (lower(trackTITLE) LIKE '%r' OR lower(trackTITLE) LIKE 'r%')
```

## 2. Zadatak (3 boda)

a) Ispišite identifikator korisnika, naziv profila i naziv medijskog sadržaja gledanog na bilo koji **datum** (oslonite se na početak gledanja) 25 dana unazad od trenutnog datuma (granice uključene). Uzeti u obzir samo medijske sadržaje namijenjene osobama mlađim od 14 godina, i gledanja za koja vrijeme proteklo od početka do kraja iznosi kraće od 20 minuta i 3 sekunde.

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

```
SELECT DISTINCT ownerId, profileName, trackTitle
FROM trackView
    NATURAL JOIN track
WHERE (viewEndDateTime - viewStartDateTime) < '20 MINUTES 3 SECONDS'::INTERVAL
    AND tageRestriction < 14
    AND viewStartDateTime::DATE >= CURRENT_DATE - '25 DAYS'::INTERVAL;
```

b) Ispišite e-mail profila, identifikator korisnika i naziv medijskog gledanog na bilo koji **datum** (oslonite se na početak gledanja) 18 dana unazad od trenutnog datuma (granice uključene). Uzeti u obzir samo medijske sadržaje za koje je definirana dobna granica, i gledanja za koja vrijeme proteklo od početka do kraja iznosi kraće od 7 minute i 5 sekundi.

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

```
SELECT DISTINCT pEmail, ownerId, trackTitle
FROM profile
    NATURAL JOIN trackView
    NATURAL JOIN track
WHERE (viewEndDateTime - viewStartDateTime) < '7 MINUTES 5 SECONDS'::INTERVAL
    AND tageRestriction IS NOT NULL
    AND viewStartDateTime::DATE >= CURRENT_DATE - '18 DAYS'::INTERVAL;
```

c) Ispišite e-mail profila, naziv profila, identifikator korisnika i naziv medijskog sadržaja gledanog na bilo koji **datum** (oslonite se na početak gledanja) 18 dana unazad od trenutnog datuma (granice uključene). Uzeti u obzir samo medijske sadržaje za koje je definirana dobna granica, i gledanja za koja vrijeme proteklo od početka do kraja iznosi dulje od 1 sata i 17 minuta.

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

```
SELECT DISTINCT ownerId, profileName, pEmail, trackTitle
FROM profile
    NATURAL JOIN trackView
    NATURAL JOIN track
WHERE (viewEndDateTime - viewStartDateTime) > '1 HOUR 17 MINUTES'::INTERVAL
    AND tageRestriction IS NOT NULL
    AND viewStartDateTime::DATE >= CURRENT_DATE - '18 DAYS'::INTERVAL;
```

d) Ispišite naziv profila, identifikator korisnika te naziv i trajanje medijskog sadržaja gledanog na bilo koji **datum** (oslonite se na početak gledanja) 22 dana unazad od trenutnog datuma (granice uključene). Uzeti u obzir samo medijske sadržaje namijenjene osobama mlađim od 15 godina, i gledanja za koja vrijeme proteklo od početka do kraja iznosi dulje od 57 minuta.

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

```
SELECT DISTINCT ownerId, profileName, trackTitle, duration
FROM trackView
    NATURAL JOIN track
WHERE (viewEndDateTime - viewStartDateTime) > '57 MINUTES'::INTERVAL
    AND tageRestriction < 15
    AND viewStartDateTime::DATE >= CURRENT_DATE - '22 DAYS'::INTERVAL;
```

### 3. Zadatak (4 boda)

a) Za sve nizozemske filmove (tj. filmove koji imaju izvorni zvučni zapis naziva 'Dutch') ispišite naziv, trajanje te naziv i adresu elektroničke pošte profila čije je barem jedno neprekinuto trajanje gledanja filma bilo kraće od polovice trajanja samog filma.

Ako za neke filmove, s izvornim zvučnim zapisom naziva 'Dutch', ne postoje profili čije je barem jedno neprekinuto gledanja filma bilo kraće od polovice trajanja filma, za naziv profila i adresu elektroničke pošte ispišite NULL.

U rezultatu se ne smiju pojavljivati duplikati.

Zapise u rezultatu poredajte po nazivu filma uzlazno, a potom po imenu profila uzlazno.

Primjer rezultata (podaci ne moraju odgovarati stvarnima!):

tracktitle	duration	profilename	pemail
Angry Dutchman	43 minutes 1 hour	johnsnow	john.snow@gmail.we
Angry Dutchman	43 minutes 1 hour	mrbean	mr.bean@gmail.uk
Anika Goes to the Mall	5 minutes 2 hours	jamesbond	NULL
Erik's Breakfast	36 minutes 1 hour	NULL	NULL

```
SELECT DISTINCT track.tracktitle, duration, trackview.profilename, profile.pEmail
FROM track NATURAL JOIN movie
      NATURAL JOIN audiolang JOIN language ON langid = audioLangId
      LEFT JOIN trackview ON track.trackID = trackview.trackid
      AND duration > 2 * (viewEndDateTime - viewStartDateTime)
      LEFT JOIN profile ON trackview.ownerid = profile.ownerid
      AND trackview.profilename = profile.profilename
WHERE isNative = 1 AND langname = 'Dutch'
ORDER BY tracktitle, trackview.profilename;
```

b) Za sve epizode serija na španjolskom jeziku (tj. epizode serija koje imaju izvorni zvučni zapis jezika naziva 'Spanish') koje u nazivu imaju dvotočku ispišite naziv epizode, trajanje te imena i adrese elektroničke pošte profila čije je barem jedno neprekinuto trajanje gledanja te epizode bilo kraće od polovice trajanja same epizode.

Ako za neke epizode na španjolskom jeziku sa dvotočkom u nazivu ne postoje profili čije je barem jedno neprekinuto gledanja epizode bilo kraće od polovice trajanja epizode, za naziv profila i adresu elektroničke pošte ispišite NULL.

U rezultatu se ne smiju pojavljivati duplikati.

Zapise u rezultatu poredajte po nazivu epizode uzlazno, a potom po imenu profila silazno.

Primjer rezultata (podaci ne moraju odgovarati stvarnima!):

tracktitle	duration	profilename	pemail
Angry Spaniard: The Return S01E03	43 minutes	johnsnow	john.snow@gmail.we
Angry Spaniard: The Return S01E03	43 minutes	drbean	dr.bean@gmail.uk
Angry Spaniard: The Return S01E04	20 seconds 42 minutes	NULL	NULL
El Diablo: a True Story S01E01	5 minutes 1 hour	jamesbond	NULL
El Diablo: a True Story S01E02	3 minutes 1 hour	NULL	NULL
Madrid: A City in Spain S01E01	36 minutes	NULL	NULL

```
SELECT DISTINCT tracktitle, duration, trackview.profilename, profile.pEmail
FROM track NATURAL JOIN showep
      NATURAL JOIN audiolang JOIN language ON langid = audioLangId
      LEFT JOIN trackview ON track.trackID = trackview.trackid
      AND duration > 2 * (viewEndDateTime - viewStartDateTime)
      LEFT JOIN profile ON trackview.ownerid = profile.ownerid
      AND trackview.profilename = profile.profilename
WHERE isNative = 1 AND langname = 'Spanish' AND tracktitle LIKE ':%:'
ORDER BY tracktitle, trackview.profilename DESC;
```

c) Za sve filmove žanra 'Horror' koji se preporučuju publici od 17 godina naviše ispišite naziv, trajanje te imena i adrese elektroničke pošte profila čije je barem jedno neprekinuto trajanje gledanja filma bilo kraće od polovice trajanja samog filma.

Ukoliko za neke filmove ne postoje profili čije je barem jedno neprekinuto trajanje gledanja filma bilo kraće od polovice trajanja filma, za ime profila i adresu elektroničke pošte postavite NULL.

U rezultatu se ne smiju pojavljivati duplikati.

Ispis poredajte abecedno po nazivu filma uzlazno, a potom po imenu profila silazno

Primjer rezultata (podaci ne moraju odgovarati stvarnima!):

tracktitle	duration	profilename	pemail
Angry Frankenstein	43 minutes 1 hour	johnsnow	john.snow@gmail.we
Angry Frankenstein	43 minutes 1 hour	drbean	dr.bean@gmail.uk
Freddie Goes to the Mall	5 minutes 2 hours	jamesbond	NULL
Zombie's Breakfast	36 minutes 1 hour	NULL	NULL

```
SELECT DISTINCT track.tracktitle, duration, trackview.profilename, profile.pEmail
FROM track NATURAL JOIN movie
      NATURAL JOIN trackgenre NATURAL JOIN genre
      LEFT JOIN trackview ON track.trackID = trackview.trackid
      AND duration > 2 * (viewEndDateTime - viewStartDateTime)
      LEFT JOIN profile ON trackview.ownerid = profile.ownerid
      AND trackview.profilename = profile.profilename
WHERE genrename LIKE 'Horror' AND tAgeRestriction = 17
ORDER BY tracktitle, trackview.profilename DESC;
```



d) Za sve serije koje imaju ocjenu 90 i više ispišite naziv početne epizode (početna epizoda je prva epizoda prve sezone), trajanje te epizode, a potom imena i adrese elektroničke pošte profila čije je barem jedno neprekinuto trajanje gledanja te epizode bilo kraće od polovice trajanja same epizode.

Ukoliko za neke početne epizode ne postoje profili čije je barem jedno neprekinuto trajanje gledanja bilo kraće od polovice trajanja epizode, za ime profila i adresu elektroničke pošte postavite NULL.

U rezultatu se ne smiju pojavljivati duplikati.

Zapise u rezultatu poredajte po nazivu filma uzlazno, a potom po imenu profila uzlazno.

Primjer rezultata (podaci ne moraju odgovarati stvarnima!):

tracktitle	duration	profilename	pemail
Angry Birds S01E01	43 minutes	johnsnow	john.snow@gmail.we
Angry Birds S01E01	43 minutes	mrbean	mr.bean@gmail.uk
Bojack Horseman Goes to the Mall S01E01	5 minutes 1 hour	jamesbond	NULL
Queen's Breakfast S01E01	36 minutes	NULL	NULL

```
SELECT DISTINCT track.tracktitle, duration, trackview.profilename, profile.pEmail
FROM track NATURAL JOIN show NATURAL JOIN showep
    LEFT JOIN trackview ON track.trackID = trackview.trackid
        AND duration > 2 * (viewEndDateTime - viewStartDateTime)
    LEFT JOIN profile ON trackview.ownerid = profile.ownerid
        AND trackview.profilename = profile.profilename
WHERE seasonno = 1 AND episodeno = 1 AND showrating >= 90
ORDER BY tracktitle, trackview.profilename;
```

#### 4. Zadatak (4 boda)

a) Za svakog vlasnika računa ispišite njegov identifikator, ime i prezime, nazive njegovih profila te za svaki profil postotak medijskog sadržaja kojemu je dodijelio pozitivnu ocjenu u odnosu na ukupan broj medijskog sadržaja kojeg je ocijenio.

U obzir uzeti samo profile koji su ocijenili barem 80 posto medijskog sadržaja kojega su i pokrenuli.

**Napomena:** Pri prvom pokretanju nekog medijskog sadržaja od strane nekog profila nastaje zapis u **profileTrack** pa se broj medijskih sadržaja koje je profil pokrenuo može odrediti temeljem sadržaja te tablice.

Postotak ispišite kao decimalnu vrijednost (bez zaokruživanja) – uočite u zadnjem stupcu znak '%' odvojen jednim razmakom od brojčane vrijednosti.

Format ispisa mora odgovarati sljedećem primjeru ispisa:

ownerId	firstname	lastname	profilename	postotak
204	Tillie	Hood	tilliehood	48.7179487179487179 %
236	Anum	Williams	anumwilliams	55.5555555555555556 %
236	Anum	Williams	curiouswimp	33.3333333333333333 %
...	. . .	. . .	. . .	. . .

```
SELECT owner.ownerId, firstname, lastname, profileTrack.profilename,
       ((SELECT COUNT(*)::DECIMAL FROM profileTrack profileTrack2
          WHERE profileTrack2.profilename = profileTrack.profileName
          AND ownerId = owner.ownerId and liked = 1) * 100)
       / COUNT(*)::DECIMAL || ' %' AS postotak
FROM owner
NATURAL JOIN profileTrack
WHERE liked IS NOT NULL
GROUP BY owner.ownerId,firstname,lastname, profileTrack.profilename
HAVING (COUNT (liked))::DECIMAL >= (((SELECT count(*)::DECIMAL
                                         FROM profileTrack profileTrack2
                                         WHERE profileTrack2.profilename = profileTrack.profileName
                                         AND ownerId = owner.ownerId) * 80) / 100)
```

b) Za svakog vlasnika računa ispišite njegovo identifikator, ime i prezime, nazive njegovih profila te za svaki profil postotak medijskog sadržaja kojemu je dodijelio negativnu ocjenu u odnosu na ukupan broj medijskog sadržaja kojega je ocijenio.

U obzir uzeti samo profile koji su ocijenili barem 75 posto medijskog sadržaja kojega su i pokrenuli.

**Napomena:** Pri prvom pokretanju nekog medijskog sadržaja od strane nekog profila nastaje zapis u **profileTrack** pa se broj medijskih sadržaja koje je profil pokrenuo može odrediti temeljem sadržaja te tablice.

Ime i prezime ispišite u jednom stupcu odvojene razmakom, a postotak ispišite kao decimalnu vrijednost (bez zaokruživanja) - uočite u zadnjem stupcu znak '%' odvojen jednim razmakom od brojčane vrijednosti.

Format ispisa mora odgovarati sljedećem primjeru ispisa:

ownerId	fullname	profilename	postotak
145	Bertie Obrien	bertieobrien	31.5789473684210526 %
149	Yasir Dillard	ClarenceGonzales	41.1764705882352941 %
149	Yasir Dillard	Ogallagievous	18.1818181818181818 %
..	. . .	. . .	. . .

```
SELECT owner.ownerId, firstname || ' ' || lastname AS fullName, profilename,
      ((SELECT COUNT(*)::DECIMAL FROM profileTrack profileTrack2
        WHERE profileTrack2.profilename = profileTrack.profileName
        AND ownerId = owner.ownerId and liked = -1) * 100)
      / COUNT(*)::DECIMAL || ' %' AS postotak
FROM owner
NATURAL JOIN profileTrack
WHERE liked IS NOT NULL
GROUP BY owner.ownerId, owner.firstname,owner.lastname, profileTrack.profilename
HAVING (COUNT (liked))::DECIMAL >= (((SELECT count(*)::DECIMAL
      FROM profileTrack profileTrack2
      WHERE profileTrack2.profilename = profileTrack.profileName AND
      ownerId = owner.ownerId) * 75) / 100)
```

c) Za svakog vlasnika računa ispišite njegov identifikator, ime i prezime, nazive njegovih profila te za svaki profil postotak medijskog sadržaja kojemu je dodijelio negativnu ocjenu u odnosu na ukupan broj medijskog sadržaja kojeg je ocijenio.

U obzir uzeti samo profile koji su dodijelili barem 10 pozitivnih ocjena više od negativnih.

Postotak ispišite kao decimalnu vrijednost (bez zaokruživanja) - uočite u zadnjem stupcu znak '%' odvojen jednim razmakom od brojčane vrijednosti.

Format ispisa mora odgovarati sljedećem primjeru ispisa:

ownerId	firstname	lastname	profilename	postotak
1	Christian	Reeves	christianreeves	40.0000000000000000 %
5	Gage	Santiago	JeffreyMartinez	14.2857142857142857 %
21	Denny	Mcfarland	LoisRichardson	18.7500000000000000 %
...	. . .	. . .	. . .	. . .

```
SELECT owner.ownerId, firstname, lastname, profilename,
      ((SELECT COUNT(*)::DECIMAL FROM profileTrack profileTrack2
        WHERE profileTrack2.profilename = profileTrack.profileName
          AND ownerId = owner.ownerId and liked = -1) * 100)
      / COUNT (*)::DECIMAL || ' %' AS postotak

FROM owner
NATURAL JOIN profileTrack
WHERE liked IS NOT NULL
GROUP BY owner.ownerId, firstname, lastname, profilename
HAVING (COUNT (*)) >= (2 * (SELECT COUNT(*)
      FROM profileTrack profileTrack2
      WHERE profileTrack2.profilename = profileTrack.profileName
      AND ownerId = owner.ownerId and liked = -1) + 10)
```

--ili--

```
SELECT owner.ownerId, firstname, lastname, profilename,
      ((SELECT COUNT(*)::DECIMAL
        FROM profileTrack profileTrack2
        WHERE profileTrack2.profilename = profileTrack.profileName
          AND ownerId = owner.ownerId and liked = -1) * 100)
      / (SELECT COUNT(*)::DECIMAL
        FROM profileTrack profileTrack2
        WHERE profileTrack2.profilename = profileTrack.profileName
          AND ownerId = owner.ownerId
          AND liked IS NOT NULL) || ' %' AS postotak

FROM owner
NATURAL JOIN profileTrack
WHERE liked = 1
GROUP BY owner.ownerId, firstname, lastname, profilename
HAVING (COUNT(liked)) >= ((SELECT COUNT(*) FROM profileTrack profileTrack2
WHERE profileTrack2.profilename = profileTrack.profileName
      AND ownerId = owner.ownerId and liked = -1) + 10)
```

d) Za svakog vlasnika računa ispišite njegovo ime i prezime, naziv žanra te broj pozitivnih i negativnih ocjena upućenih s njegovog računa za taj žanr.

U obzir uzeti samo one žanrove za koje je vlasnik ocijenio barem 20 medijskih sadržaja.

Format ispisa mora odgovarati sljedećem primjeru ispisa:

ownerid	firstname	lastname	genreid	genreName	negLikes	pozLikes
112	Gene	Ruiz	7	Dramas	12	11
295	Sharna	Serrano	5	Comedies	8	12
295	Sharna	Serrano	7	Dramas	12	11
. . .		. . .	...	. . .	. . .	...

```
SELECT owner.ownerId, firstname, lastname, trackgenre.genreid, genreName,
      (SELECT COUNT(*) FROM profileTrack profileTrack2
        NATURAL JOIN trackgenre trackgenre2
      WHERE ownerId = owner.ownerId
        AND trackgenre2.genreId = trackgenre.genreId
        AND liked = -1) as negLikes,
      (SELECT COUNT(*) FROM profileTrack profileTrack2
        NATURAL JOIN trackgenre trackgenre2
      WHERE ownerId = owner.ownerId
        AND trackgenre2.genreId = trackgenre.genreId
        AND liked = 1 )as pozLikes
FROM owner NATURAL JOIN profileTrack
      NATURAL JOIN trackgenre
      NATURAL JOIN genre
WHERE liked IS NOT NULL
GROUP BY owner.ownerId, firstname, lastname, trackgenre.genreid, genreName
HAVING (COUNT(genreName)) >= 20
```

## 5. Zadatak (3 boda)

Postoji nekoliko desetaka inačica donje dvije verzije zadatka. U različitim zadacima koriste se različiti uvjeti selekcije, ali je svaki zadatak inačica jedne od donje dvije verzije.

a) Prvoj epizodi prve sezone serije koja ima ocjenu veću od 99.00 (uzeti u obzir ocjenu serije, a ne ocjenu epizode serije) potrebno je ažurirati naziv. Iz naziva je potrebno izbaciti dio koji se odnosi na podatke vezane o sezoni i epizodi serije.

**Napomena:** uočite da se posljednjih 7 znakova u nazivu epizode pojedine serije odnosi na podatak o sezoni i epizodi serije.

**Primjer:** naziv " Halt and Catch Fire S01E01" treba promijeniti na " Halt and Catch Fire ".

```
UPDATE track
SET tracktitle = substring(tracktitle, 1, (char_length(tracktitle)-7))
WHERE trackid IN (SELECT trackid
                  FROM showep
                  NATURAL JOIN show
                  WHERE seasonNo = 1 AND episodeNo = 1 AND showrating > 99.00)
```

b) Za 2% povećati ocjenu/rejting (*track.trackRating*) filmovima čiji je nasljednik zaradio između 10% i 20% više od njega. Uključiti obje granice: [10%, 20%].

**Primjer:** Filmu 'Man in black' s ocjenom/rejtingom 40.00 koji je zaradio od prodaje kino ulaznica 100€ nasljednik je film 'Man in black II' koji je zaradio 115€. Ocjenu za 'Man in black' treba povećati za 2% tj. na 40.80.

Brojevi u primjeru su, naravno, izmišljeni.

```
UPDATE track
SET trackRating = trackRating * 1.02
WHERE trackid IN (SELECT movie1.trackId
                  FROM movie movie1
                  JOIN movie movie2 ON movie1.trackId = movie2.prevMovieId
                  WHERE movie2.boxIncome BETWEEN 1.10*movie1.boxIncome
                  AND 1.20*movie1.boxIncome)
```

## 6. Zadatak (3 boda)

a) Za zadani izraz relacijske algebre napišite pripadni SQL upit:

$\rho_{\text{trackOwnerCount}}(\text{trackTitle}, \text{trackCount}) \left( \text{trackTitle} \mathrel{G} \text{COUNT}(\text{trackId}) \left( \sigma_{\text{dateOfBirth} \geq '01.01.2003.' \wedge \text{releaseDate} \geq '01.01.2021.'} (\text{track} \bowtie \text{profileTrack} \bowtie \text{profile} \bowtie \text{owner}) \right) \right)$

```
SELECT trackTitle, COUNT(trackId) as trackCount
FROM track
      NATURAL JOIN profileTrack
      NATURAL JOIN profile
      NATURAL JOIN owner
WHERE dateOfBirth >= '01.01.2003.' AND releaseDate >= '01.01.2021.'
GROUP BY trackTitle
```

b) Za zadani izraz relacijske algebre napišite pripadni SQL upit:

$\rho_{\text{genreTrackCount}}(\text{genreName}, \text{trackCount}) \left( \text{genreName} \mathrel{G} \text{COUNT}(\text{trackId}) \left( \sigma_{\text{trackRating} \geq 4} (\text{genre} \bowtie \text{trackGenre} \bowtie \text{track}) \right) \right)$

```
SELECT genreName, COUNT(trackId) as trackCount
FROM genre
      NATURAL JOIN trackGenre
      NATURAL JOIN track
WHERE trackRating >= 4
GROUP BY genreName
```

c) Za zadani izraz relacijske algebre napišite pripadni SQL upit:

$\rho_{\text{ownerAveragePrice}}(\text{ownerId}, \text{averagePrice}) (\text{ownerId } \mathcal{G} \text{ AVG}(\text{pricePerMonth}) (\sigma_{\text{dateOfBirth} \geq '01.01.2000.'} (\text{owner} \bowtie \text{ownerPack} \bowtie \text{pack})))$

```
SELECT ownerId, AVG(pricePerMonth) as averagePrice
FROM owner
      NATURAL JOIN ownerPack
      NATURAL JOIN pack
WHERE dateOfBirth >= '01.01.2000.'
GROUP BY ownerId
```

d) Za zadani izraz relacijske algebre napišite pripadni SQL upit:

$\rho_{\text{showTitleTotalEpisodes}}(\text{showTitle}, \text{numberOfEpisodes}) (\text{showTitle } \mathcal{G} \text{ COUNT}(\text{episodeNo}) (\sigma_{\text{releaseDate} \geq '01.01.2021.'} (\text{show} \bowtie \text{showEp} \bowtie \text{track})))$

```
SELECT showTitle, COUNT(episodeNo) as numberOfEpisodes
FROM show
      NATURAL JOIN showEp
      NATURAL JOIN track
WHERE releaseDate >= '01.01.2021.'
GROUP BY showTitle
```



## 7. Zadatak (2 boda)

a) Napisati SQL naredbe kojima će se kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B+ stabla) svih dolje navedenih upita nad tablicama **owner** i **profiletrack**. Osim indeksa koje PostgreSQL kreira automatski, nisu kreirani nikakvi dodatni indeksi. Indeksima treba dodijeliti imena i1, i2, ..., in.

Ako smatrate da za neku od navedenih SELECT naredbi ne treba kreirati indeks, navedite koji je razlog za to.

1. SELECT \* FROM owner ORDER BY lastName, firstName DESC;
2. SELECT \* FROM profiletrack WHERE liked = -1;
3. SELECT \* FROM owner WHERE lastName = 'Farley' AND firstName = 'Tobias';
4. SELECT \* FROM owner ORDER by dateofBirth DESC;
5. SELECT \* FROM owner ORDER BY firstName, lastName;
6. SELECT \* FROM owner WHERE dateofBirth BETWEEN '01.01.1992' AND '31.12.2001';

```
CREATE INDEX i1 ON owner(lastName, firstName DESC)          1
CREATE INDEX i2 ON owner(firstName, lastName)                3 i 5
CREATE INDEX i3 ON owner(dateofBirth) //ili (dateofBirth DESC) 4 i 6
```

Za 2 ne treba kreirati indeks:

Ne treba kreirati indeks ako vrijednosti atributa imaju relativno mali broj različitih vrijednosti. Atribut *liked* ima dozvoljene vrijednosti: -1 i 1 (ili je NULL)

b) Napisati SQL naredbe kojima će se kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B+ stabla) svih dolje navedenih upita nad tablicama **owner** i **profiletrack**. Osim indeksa koje PostgreSQL kreira automatski, nisu kreirani nikakvi dodatni indeksi. Indeksima treba dodijeliti imena i1, i2, ..., in.

Ako smatrate da za neku od navedenih SELECT naredbi ne treba kreirati indeks, navedite koji je razlog za to.

1. SELECT \* FROM owner ORDER BY lastName, firstName;
2. SELECT \* FROM owner WHERE dateofBirth < '01.01.1999';
3. SELECT \* FROM owner ORDER BY lastName DESC, dateofBirth;
4. SELECT \* FROM owner WHERE firstName = 'Tobias' AND lastName='Farley';
5. SELECT \* FROM profiletrack WHERE liked= 1;
6. SELECT \* FROM owner ORDER BY dateofBirth, oemail;

```
CREATE INDEX i1 ON owner(lastName, firstName)                1 i 4
CREATE INDEX i2 ON owner(lastName DESC, dateofBirth)          3
CREATE INDEX i3 ON owner(dateofBirth, oemail)                  2 i 6
```

Za 5 ne treba kreirati indeks:

Ne treba kreirati indeks ako vrijednosti atributa imaju relativno mali broj različitih vrijednosti. Atribut *liked* ima dozvoljene vrijednosti: -1 i 1 (ili je NULL).

c) Napisati SQL naredbe kojima će se kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B+ stabla) svih dolje navedenih upita nad tablicama **owner** i **profiletrack**. Osim indeksa koje PostgreSQL kreira automatski, nisu kreirani nikakvi dodatni indeksi. Indeksima treba dodijeliti imena i1, i2, ..., in. Ako smatrate da za neku od navedenih SELECT naredbi ne treba kreirati indeks, navedite koji je razlog za to.

1. SELECT \* FROM owner WHERE dateofBirth BETWEEN '01.01.1992' AND '31.12.2001';
2. SELECT \* FROM owner WHERE firstName BETWEEN 'Tanya' AND 'Tracy';
3. SELECT \* FROM profiletrack WHERE liked = -1;
4. SELECT \* FROM owner WHERE dateofBirth < '01.01.1999' AND lastName = 'Perry';
5. SELECT \* FROM owner ORDER BY dateofBirth, oemail DESC;
6. SELECT \* FROM owner ORDER BY lastName DESC, dateofBirth DESC;

```
CREATE INDEX i1 ON owner(dateofBirth, oemail DESC)      1 i 5
CREATE INDEX i2 ON owner(firstName)                    2
CREATE INDEX i3 ON owner(lastName, dateofBirth)         4 i 6
// ili (lastName DESC, dateofBirth DESC)
```

Za 3 ne treba kreirati indeks:

Ne treba kreirati indeks ako vrijednosti atributa imaju relativno mali broj različitih vrijednosti. Atribut *liked* ima dozvoljene vrijednosti: -1 i 1 (ili je NULL).

d) Napisati SQL naredbe kojima će se kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B+ stabla) svih dolje navedenih upita nad tablicama **owner** i **profiletrack**. Osim indeksa koje PostgreSQL kreira automatski, nisu kreirani nikakvi dodatni indeksi. Indeksima treba dodijeliti imena i1, i2, ..., in. Ako smatrate da za neku od navedenih SELECT naredbi ne treba kreirati indeks, navedite koji je razlog za to.

1. SELECT \* FROM owner WHERE firstName BETWEEN 'Tanya' and 'Vincenzo';
2. SELECT \* FROM owner ORDER BY lastName, oemail;
3. SELECT \* FROM owner WHERE dateofBirth < '01.01.1999' AND lastName = 'Perry';
4. SELECT \* FROM owner ORDER BY firstName, lastName, oemail;
5. SELECT \* FROM profiletrack WHERE liked = 1;
6. SELECT \* FROM owner ORDER BY lastName DESC, dateofBirth DESC;

```
CREATE INDEX i1 ON owner(firstName, lastName, oemail) 1 i 4
CREATE INDEX i2 ON owner(lastName, oemail)            2
CREATE INDEX i3 ON owner(lastName, dateofBirth)       3 i 6
// ili (lastName DESC, dateofBirth DESC)
```

Za 5 ne treba kreirati indeks:

Ne treba kreirati indeks ako vrijednosti atributa imaju relativno mali broj različitih vrijednosti. Atribut *liked* ima dozvoljene vrijednosti: -1 i 1 (ili je NULL).

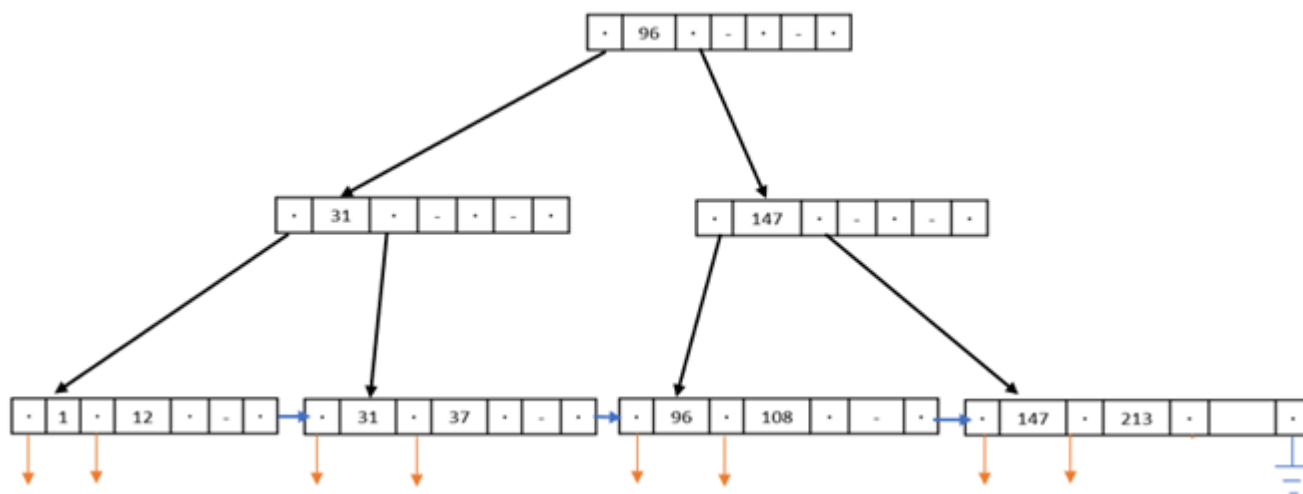
## 8. Zadatak (2 boda)

a) Relacija **owner** (*ownerid*, *oemail*, *firstname*, *lastname*, *dateofbirth*, *gender*) sadrži n-torke sa sljedećim vrijednostima atributa *ownerid*: 1, 12, 31, 37, 96, 108, 147, 213.

Nacrtajte B<sup>+</sup> stablo reda 4 za atribut *ownerid* tako da popunjenost stabla bude minimalna.

### Napomena:

- B<sup>+</sup> stablo nacrtati na papiru, uslikati i uploadati na Edgar.
- Potrebno je izračunati broj kazaljki za svaku vrstu čvora. Rješenje napišite u prostor za slobodan unos teksta (izračunati i za minimalnu i za maksimalnu popunjenost).



MIN:  $K = 2$ ,  $\check{C} = \lceil n/2 \rceil = 2$ ,  $L = \lceil (n-1)/2 \rceil = 2$

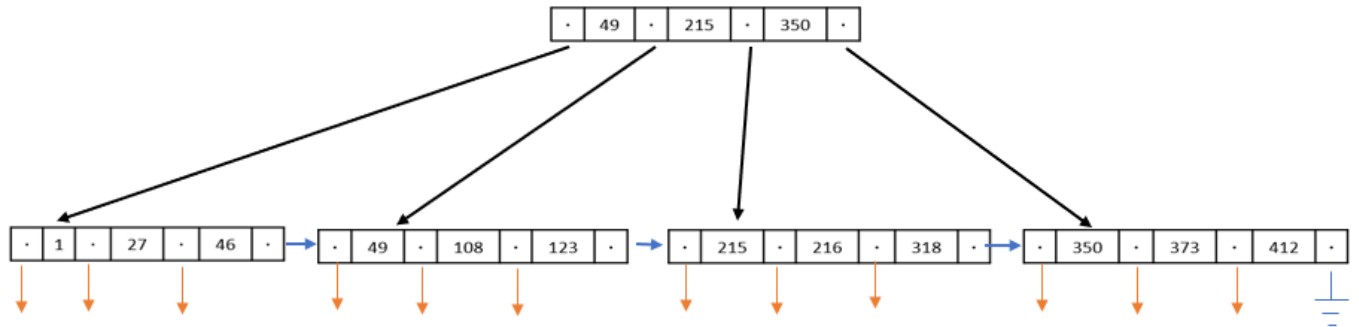
MAX:  $K = 4$ ,  $\check{C} = 4$ ,  $L = n-1 = 3$

b) Relacija **owner** (*ownerid*, *oemail*, *firstname*, *lastname*, *dateofbirth*, *gender*) sadrži n-torke sa sljedećim vrijednostima atributa *ownerid* 1, 27, 46, 49, 108, 123, 215, 216, 318, 350, 373, 412.

Nacrtajte B<sup>+</sup> stablo reda 4 za atribut *ownerid* tako da popunjenost stabla bude maksimalna.

**Napomena:**

- B<sup>+</sup> stablo nacrtati na papiru, uslikati i uploadati na Edgar.
- Potrebno je izračunati broj kazaljki za svaku vrstu čvora. Rješenje napišite u prostor za slobodan unos teksta (izračunati i za minimalnu i za maksimalnu popunjenost).



MIN:  $K = 2$ ,  $\check{C} = \lceil n/2 \rceil = 2$ ,  $L = \lceil (n-1)/2 \rceil = 2$

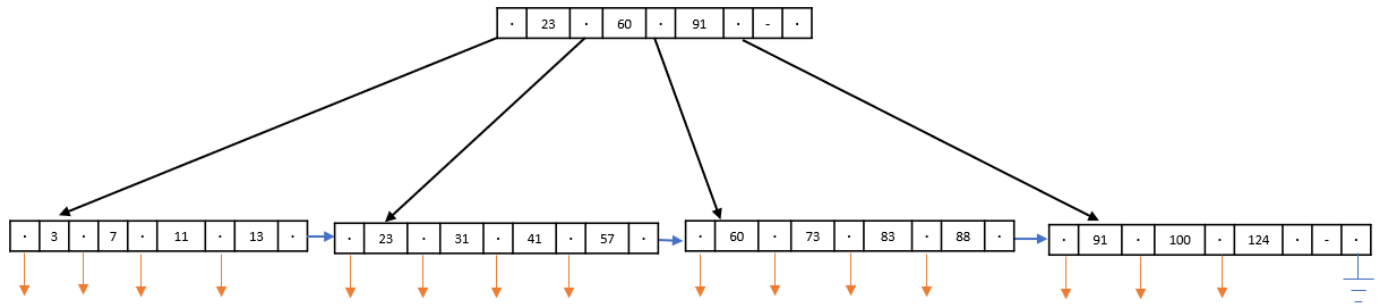
MAX:  $K = 4$ ,  $\check{C} = 4$ ,  $L = n-1 = 3$

c) Relacija **owner** (*ownerid*, *oemail*, *firstname*, *lastname*, *dateofbirth*, *gender*) sadrži n-torke sa sljedećim vrijednostima atributa *ownerid* 3, 7, 11, 13, 23, 31, 41, 57, 60, 73, 83, 88, 91, 100, 124.

Nacrtajte B<sup>+</sup> stablo reda 5 za atribut *ownerid* tako da popunjenost stabla bude maksimalna.

**Napomena:**

- B<sup>+</sup> stablo nacrtati na papiru, uslikati i uploadati na Edgar.
- Potrebno je izračunati broj kazaljki za svaku vrstu čvora. Rješenje napišite u prostor za slobodan unos teksta (izračunati i za minimalnu i za maksimalnu popunjenost).



MIN:  $K = 2$ ,  $\check{C} = \lceil n/2 \rceil = 3$ ,  $L = \lceil (n-1)/2 \rceil = 2$

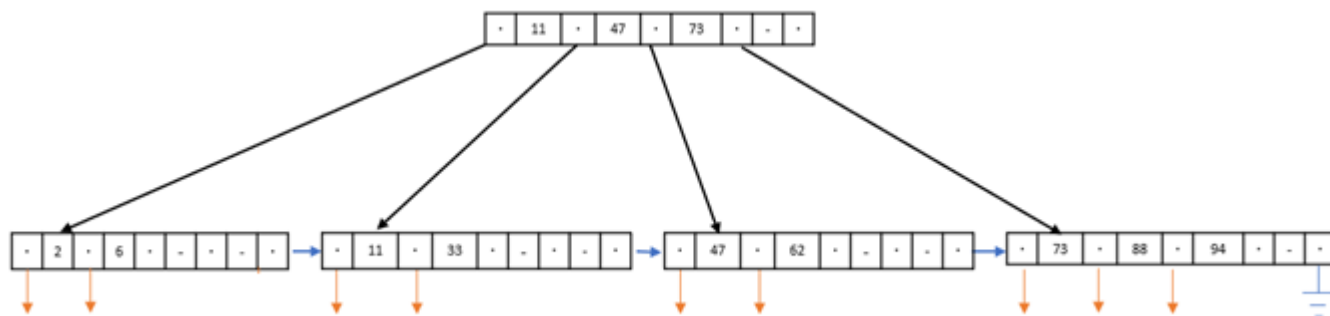
MAX:  $K = 5$ ,  $\check{C} = 5$ ,  $L = n-1 = 4$

d) Relacija **owner** (*ownerid*, *oemail*, *firstname*, *lastname*, *dateofbirth*, *gender*) sadži n-torke sa sljedećim vrijednostima atributa *ownerid* 2, 6, 11, 33, 47, 62, 73, 88, 94.

Nacrtajte B<sup>+</sup> stablo reda 5 za atribut *ownerid* tako da popunjenost stabla bude minimalna.

**Napomena:**

- B<sup>+</sup> stablo nacrtati na papiru, uslikati i uploadati na Edgar.
- Potrebno je izračunati broj kazaljki za svaku vrstu čvora. Rješenje napišite u prostor za slobodan unos teksta (izračunati i za minimalnu i za maksimalnu popunjenost).



MIN:  $K = 2$ ,  $\check{C} = \lceil n/2 \rceil = 3$ ,  $L = \lceil (n-1)/2 \rceil = 2$

MAX:  $K = 5$ ,  $\check{C} = 5$ ,  $L = n-1 = 4$

## 9. Zadatak (2 boda)

a) Uz pretpostavku da na relacijskoj shemi *StreamFlix* baze podataka vrijedi skup funkcijskih zavisnosti

$F = \{\text{trackId} \rightarrow \text{releaseDate duration},$   
 $\text{showId episodeNo seasonNo} \rightarrow \text{trackId},$   
 $\text{showId} \rightarrow \text{showRating}\}$

ispitati vrijedi li sljedeća funkcijska zavisnost:

$\text{showId episodeNo seasonNo} \rightarrow \text{duration showRating}.$

Za svaki korak dokaza navesti naziv pravila ili aksioma koji se koristi.

*A-1:*  $\text{showId episodeNo seasonNo} \rightarrow \text{showId episodeNo seasonNo}$

*Akumulacija:*  $(\text{showId} \rightarrow \text{showRating})$

$\text{showId episodeNo seasonNo} \rightarrow \text{showId episodeNo seasonNo showRating}$

*Akumulacija:*  $(\text{showId episodeNo seasonNo} \rightarrow \text{trackId})$

$\text{showId episodeNo seasonNo} \rightarrow \text{showId episodeNo seasonNo showRating trackId}$

*Akumulacija:*  $(\text{trackId} \rightarrow \text{releaseDate duration})$

$\text{showId episodeNo seasonNo} \rightarrow \text{showId episodeNo seasonNo showRating trackId releaseDate duration}$

*P-2:*  $\text{showId episodeNo seasonNo} \rightarrow \text{duration showRating}$

b) Uz pretpostavku da na relacijskoj shemi *StreamFlix* baze podataka vrijedi skup funkcijskih zavisnosti

$F = \{ \text{ownerId} \rightarrow \text{firstName lastName gender},$   
 $\text{ownerId profileName} \rightarrow \text{pEmail pAgeRestriction},$   
 $\text{ownerId profileName trackId} \rightarrow \text{liked} \},$

ispitati vrijedi li sljedeća funkcijska zavisnost:

$\text{ownerId profileName trackId} \rightarrow \text{firstName pAgeRestriction liked}.$

Za svaki korak dokaza navesti naziv pravila ili aksioma koji se koristi.

*A-1:*  $\text{ownerId profileName trackId} \rightarrow \text{ownerId profileName trackId}$

*Akumulacija:*  $(\text{ownerId} \rightarrow \text{firstName lastName gender})$

$\text{ownerId profileName trackId} \rightarrow \text{ownerId profileName trackId firstName lastName gender}$

*Akumulacija:*  $(\text{ownerId profileName} \rightarrow \text{pEmail pAgeRestriction})$

$\text{ownerId profileName trackId} \rightarrow \text{ownerId profileName trackId firstName lastName gender pEmail pAgeRestriction}$

*Akumulacija:*  $(\text{ownerId profileName trackId} \rightarrow \text{liked})$

$\text{ownerId profileName trackId} \rightarrow \text{ownerId profileName trackId firstName lastName gender pEmail pAgeRestriction liked}$

*P-2:*  $\text{ownerId profileName trackId} \rightarrow \text{firstName pAgeRestriction liked}$



c) Uz pretpostavku da na relacijskoj shemi *StreamFlix* baze podataka vrijedi skup funkcijskih zavisnosti

$F = \{\text{deviceId} \rightarrow \text{viewStartDateTime} \rightarrow \text{trackId} \rightarrow \text{viewEndDateTime},$   
 $\text{trackId} \rightarrow \text{trackTitle} \rightarrow \text{showId},$   
 $\text{showId} \rightarrow \text{showTitle} \rightarrow \text{showRating}\},$

ispitati vrijedi li sljedeća funkcijska zavisnost:

$\text{deviceId} \rightarrow \text{viewStartDateTime} \rightarrow \text{trackTitle} \rightarrow \text{showTitle}.$

Za svaki korak dokaza navesti naziv pravila ili aksioma koji se koristi.

*A-1:*     $\text{deviceId} \rightarrow \text{viewStartDateTime} \rightarrow \text{deviceId} \rightarrow \text{viewStartDateTime}$

*Akumulacija:*     $(\text{deviceId} \rightarrow \text{viewStartDateTime} \rightarrow \text{trackId} \rightarrow \text{viewEndDateTime})$

$\text{deviceId} \rightarrow \text{viewStartDateTime} \rightarrow \text{deviceId} \rightarrow \text{viewStartDateTime} \rightarrow \text{trackId} \rightarrow \text{viewEndDateTime}$

*Akumulacija:*     $(\text{trackId} \rightarrow \text{trackTitle} \rightarrow \text{showId})$

$\text{deviceId} \rightarrow \text{viewStartDateTime} \rightarrow \text{deviceId} \rightarrow \text{viewStartDateTime} \rightarrow \text{trackId} \rightarrow \text{viewEndDateTime} \rightarrow \text{trackTitle} \rightarrow \text{showId}$

*Akumulacija:*     $(\text{showId} \rightarrow \text{showTitle} \rightarrow \text{showRating})$

$\text{deviceId} \rightarrow \text{viewStartDateTime} \rightarrow \text{deviceId} \rightarrow \text{viewStartDateTime} \rightarrow \text{trackId} \rightarrow \text{viewEndDateTime} \rightarrow \text{trackTitle} \rightarrow \text{showId}$   
 $\text{showTitle} \rightarrow \text{showRating}$

*P-2:*     $\text{deviceId} \rightarrow \text{viewStartDateTime} \rightarrow \text{trackTitle} \rightarrow \text{showTitle}$

d) Uz pretpostavku da na relacijskoj shemi *StreamFlix* baze podataka vrijedi skup funkcijskih zavisnosti

$F = \{\text{deviceId viewStartDateTime} \rightarrow \text{trackId ownerId profileName savedProgress},$   
 $\text{trackId} \rightarrow \text{trackTitle},$   
 $\text{ownerId profileName} \rightarrow \text{pEmail pAgeRestriction}\},$

ispitati vrijedi li sljedeća funkcijska zavisnost:

$\text{deviceId viewStartDateTime} \rightarrow \text{pAgeRestriction trackTitle}.$

Za svaki korak dokaza navesti naziv pravila ili aksioma koji se koristi.

*A-1:*     $\text{deviceId viewStartDateTime} \rightarrow \text{deviceId viewStartDateTime}$

*Akumulacija:*     $(\text{deviceId viewStartDateTime} \rightarrow \text{trackId ownerId profileName savedProgress})$

$\text{deviceId viewStartDateTime} \rightarrow \text{deviceId viewStartDateTime trackId ownerId profileName savedProgress}$

*Akumulacija:*     $(\text{trackId} \rightarrow \text{trackTitle})$

$\text{deviceId viewStartDateTime} \rightarrow \text{deviceId viewStartDateTime trackId ownerId profileName savedProgress}$   
 $\text{trackTitle}$

*Akumulacija:*     $(\text{ownerId profileName} \rightarrow \text{pEmail pAgeRestriction})$

$\text{deviceId viewStartDateTime} \rightarrow \text{deviceId viewStartDateTime trackId ownerId profileName savedProgress}$   
 $\text{trackTitle pEmail pAgeRestriction}$

*P-2:*     $\text{deviceId viewStartDateTime} \rightarrow \text{pAgeRestriction trackTitle}$

## 10. Zadatak (4 boda)

a) Aplikacija za prijavu na cijepljenje protiv COVID-19 u svojoj bazi podataka evidentira podatke o osobama, njihovim adresama i mjestima stanovanja, odabranom liječniku obiteljske medicine, datumu i vremenu prijave, vrstama cjepiva te osobni odabir prioriteta za različita cjepiva.

Shema **PRIJAVLJIVANJE** se sastoji od sljedećih atributa:

<b>OIB</b>	– osobni identifikacijski broj osobe
<b>prezime</b>	– prezime osobe
<b>ime</b>	– ime osobe
<b>adresa</b>	– adresa osobe
<b>postBr</b>	– poštanski broj mjesta stanovanja osobe
<b>nazivMjesto</b>	– naziv mjesta stanovanja osobe
<b>sifraLiječnik</b>	– šifra odabranog liječnika obiteljske medicine
<b>prezimeLijecnik</b>	– prezime odabranog liječnika obiteljske medicine
<b>imeLijecnik</b>	– ime l odabranog liječnika obiteljske medicine
<b>datumVrijemePrijava</b>	– datum i vrijeme prijave
<b>sifraVrCjepivo</b>	– šifra vrste cjepiva
<b>nazivVrCjepivo</b>	– naziv vrste cjepiva
<b>prioritetVrCjepivo</b>	– prioritet odabira vrste cjepiva

Jedna osoba ima jednu adresu stanovanja, jednog odabranog liječnika obiteljske medicine i samo se jednom može prijaviti za cijepljenje. Osoba će odrediti prioritet za svaku vrstu cjepiva (ukoliko osoba ne želi primiti neko cjepivo, tada će za to cjepivo postaviti null vrijednost).

Odrediti ključ relacijske sheme **PRIJAVLJIVANJE** tako da ona bude u 1NF, a zatim postupno normalizirati relacijsku shemu na 2NF i 3NF.

Početna shema:

**PRIJAVLJIVANJE** = OIB, prezime, ime, adresa, postBr, nazivMjesto, sifraLiječnik, prezimeLijecnik, imeLijecnik, datumVrijemePrijava, sifraVrCjepivo, nazivVrCjepivo, prioritetVrCjepivo

### Normalizacija na 1NF

Vrijede sljedeće FZ:

- OIB → prezime, ime, adresa, postBr, nazivMjesto, sifraLiječnik, prezimeLijecnik, imeLijecnik, datumVrijemePrijava
- OIB, sifraVrCjepivo → nazivVrCjepivo, prioritetVrCjepivo

**PRIJAVLJIVANJE** = OIB, prezime, ime, adresa, postBr, nazivMjesto, sifraLiječnik, prezimeLijecnik, imeLijecnik, datumVrijemePrijava, sifraVrCjepivo, nazivVrCjepivo, prioritetVrCjepivo

K <sub>PRIJAVLJIVANJE</sub> = OIB, sifraVrCjepivo

### Normalizacija na 2NF

**OSOBA1** = OIB, prezime, ime, adresa, postBr, nazivMjesto, sifraLiječnik, prezimeLijecnik, imeLijecnik, datumVrijemePrijava

K <sub>OSOBA1</sub> = OIB

**VRCJEPIVO** = sifraVrCjepivo, nazivVrCjepivo

K <sub>VRCJEPIVO</sub> = sifraVrCjepivo

**PRIORITET** = OIB, sifraVrCjepivo, prioritetVrCjepivo

K<sub>PRIORITET</sub> = OIB, sifraVrCjepivo

***Normalizacija na 3NF***

**OSOBA** = OIB, prezime, ime, adresa, postBr, sifraLiječnik, datumVrijemePrijava

K<sub>OSOBA</sub> = OIB

**MJESTO** = postBr, nazivMjesto

K<sub>MJESTO</sub> = postBr

**LIJECNIK** = sifraLiječnik, prezimeLiječnik, imeLiječnik

K<sub>LIJECNIK</sub> = sifraLiječnik

***Konačna shema baze podataka sadrži sljedeće relacijske sheme:***

OSOBA, MJESTO, LIJECNIK, VRCJEPIVO, PRIORITET

b) Aplikacija za pozivanje na cijepljenje protiv COVID-19 u svojoj bazi podataka evidentira podatke o osobama, rednom broju cijepljenja (prvo, drugo), punktu na kojem će se obaviti cijepljenje, datumu i vremenu cijepljenja, da li je osoba potvrdila dolazak na cijepljenje, datumu i vremenu odgovora te razlogu eventualnog odbijanja poziva.

Schema **POZIVANJE** se sastoji od sljedećih atributa:

<b>OIB</b>	– osobni identifikacijski broj osobe
<b>prezime</b>	– prezime osobe
<b>ime</b>	– ime osobe
<b>rbrCijepljenje</b>	– redni broj cijepljenja (prvo, ili drugo)
<b>sifraPunkt</b>	– šifra punkta na kojem se obavlja cijepljenje
<b>nazivPunkt</b>	– naziv punkta na kojem se obavlja cijepljenje
<b>adresaPunkt</b>	– adresa punkta na kojem se obavlja cijepljenje
<b>postBr</b>	– poštanski broj punkta na kojem se obavlja cijepljenje
<b>nazivMjesto</b>	– naziv mjesta punkta na kojem se obavlja cijepljenje
<b>datumVrijemePoziv</b>	– datum i vrijeme cijepljenja na koje se poziv odnosi
<b>potvrdaPoziv</b>	– da li je osoba potvrdila dolazak
<b>razlogNedolazak</b>	– razlog nedolaska na cijepljenje
<b>datumVrijemeOdgovor</b>	– datum i vrijeme kad je osoba odgovorila na poziv

Jedna osoba može primiti jedan ili više poziva na cijepljenje (osoba može dobiti više poziva u slučaju kada se radi o cijepljenju koje se obavlja s više doza i/ili u slučaju opravdanog otkazivanja dolaska na prethodno upućeni poziv). Jedan poziv upućen jednoj osobi odnosi se na jedan punkt na kojem se obavlja cijepljenje i na jedan termin (datumvrijemePoziv). U isto vrijeme na isti punkt može biti pozvano više osoba. Različiti pozivi istoj osobi mogu se odnositi na iste ili različite punktove. Osoba na jedan poziv za jedan termin može uputiti samo jedan odgovor.

Odrediti ključ relacijske sheme **POZIVANJE** tako da ona bude u 1NF, a zatim postupno normalizirati relacijsku shemu na 2NF i 3NF.

Početna shema:

**POZIVANJE** = OIB, prezime, ime, rbrCijepljenje, sifraPunkt, nazivPunkt, adresaPunkt, postBr, nazivMjesto, datumVrijemePoziv, potvrdaPoziv, razlogNedolazak, datumVrijemeOdgovor

#### **Normalizacija na 1NF**

Vrijede sljedeće FZ:

- OIB → prezime, ime
- OIB, datumVrijemePoziv → rbrCijepljenje, sifraPunkt, nazivPunkt, adresaPunkt, postBr, nazivMjesto, potvrdaPoziv, razlogNedolazak, datumVrijemeOdgovor

**POZIVANJE** = OIB, prezime, ime, rbrCijepljenje, sifraPunkt, nazivPunkt, adresaPunkt, postBr, nazivMjesto, datumVrijemePoziv, potvrdaPoziv, razlogNedolazak, datumVrijemeOdgovor

K **POZIVANJE** = OIB, datumVrijemePoziv

#### **Normalizacija na 2NF**

**OSOBA** = OIB, prezime, ime

K **OSOBA** = OIB

**POZIV1** = OIB, datumVrijemePoziv, rbrCijepljenje, sifraPunkt, nazivPunkt, adresaPunkt, postBr, nazivMjesto, potvrdaPoziv, razlogNedolazak, datumVrijemeOdgovor

K **POZIV1** = OIB, datumVrijemePoziv

**Normalizacija na 3NF**

**POZIV** = OIB, datumVrijemePoziv, rbrCijepljenje, sifraPunkt, potvrdaPoziv, razlogNedolazak, datumVrijemeOdgovor  
K<sub>POZIV</sub> = OIB, datumVrijemePoziv

**PUNKT1** = sifraPunkt, nazivPunkt, adresaPunkt, postBrPunkt, nazivMjestoPunkt  
K<sub>PUNKT1</sub> = sifraPunkt

**MJESTO** = postBr, nazMjesto  
K<sub>MJESTO</sub> = postBr

**PUNKT** = sifraPunkt, nazivPunkt, adresaPunkt, postBr  
K<sub>PUNKT</sub> = sifraPunkt

**Konačna shema baze podataka sadrži sljedeće relacijske sheme:**  
OSOBA, POZIV, MJESTO, PUNKT

c) Aplikacija za evidenciju obavljenog cijepljenja protiv COVID-19 u svojoj bazi podataka evidentira podatke o osobama, rednom broju cijepljenja, punktu na kojem je obavljeno cijepljenje, datumu cijepljenja, liječniku koji je obavio cijepljenje, vrsti i seriji cjepiva.

Schema **CIJEPLJENJE** se sastoji od sljedećih atributa:

<b>OIB</b>	– osobni identifikacijski broj osobe
<b>prezime</b>	– prezime osobe
<b>ime</b>	– ime osobe
<b>rbrCijepljenje</b>	– redni broj cijepljenja (prvo ili drugo)
<b>sifraPunkt</b>	– šifra punkta na kojem je obavljeno cijepljenje
<b>nazivPunkt</b>	– naziv punkta na kojem je obavljeno cijepljenje
<b>sifraLijechnik</b>	– šifra liječnika koji je obavio cijepljenje
<b>prezimeLijechnik</b>	– prezime liječnika koji je obavio cijepljenje
<b>imeLijechnik</b>	– ime liječnika koji je obavio cijepljenje
<b>datumCijepljenje</b>	– datum cijepljenja
<b>sifraVrCjepivo</b>	– šifra vrste cjepiva
<b>nazivVrCjepivo</b>	– naziv vrste cjepiva
<b>serijaCjepivo</b>	– oznaka serije cjepiva kojim je osoba cijepljena

Jedna osoba može biti cijepljena jednom ili dva puta. Jedno cijepljenje jedne osobe obavlja se na jednom punktu, obavlja ga jedan liječnik, osoba dobije jednu vrstu cjepiva, pri čemu se evidentira oznaka serije tog cjepiva. Drugo cijepljenje iste osobe može se obaviti na istom ili različitom punktu, istom ili različitom vrstom cjepiva, u različitim terminima te s različitom serijom cjepiva.

Odrediti ključ relacijske sheme **CIJEPLJENJE** tako da ona bude u 1NF, a zatim postupno normalizirati relacijsku shemu na 2NF i 3NF.

Početna shema:

**CIJEPLJENJE** = OIB, prezime, ime, rbrCijepljenje, sifraPunkt, nazivPunkt, sifraLijechnik, prezimeLijechnik, imeLijechnik, datumCijepljenje, sifraVrCjepivo, nazivVrCjepivo, serijaCjepivo

### **Normalizacija na 1NF**

Vrijede sljedeće FZ:

- OIB → prezime, ime
- OIB, datumCijepljenje → rbrCijepljenje, sifraPunkt, nazivPunkt, sifraLijechnik, prezimeLijechnik, imeLijechnik, sifraVrCjepivo, nazVrCjepivo, serijaCjepivo
- OIB, rbrCijepljenje → datumCijepljenje, sifraPunkt, nazivPunkt, sifraLijechnik, prezimeLijechnik, imeLijechnik, sifraVrCjepivo, nazVrCjepivo, serijaCjepivo

**CIJEPLJENJE** = OIB, prezime, ime, rbrCijepljenje, sifraPunkt, nazivPunkt, datumCijepljenje, sifraLijechnik, prezimeLijechnik, imeLijechnik, sifraVrCjepivo, nazivVrCjepivo, serijaCjepivo

K1 **CIJEPLJENJE** = OIB, datumCijepljenje (*primarni ključ - PK*)

K2 **CIJEPLJENJE** = OIB, rbrCijepljenje (*alternativni ključ*)

### **Normalizacija na 2NF**

**OSOBA** = OIB, prezime, ime

K **OSOBA** = OIB

**CIJEPLJENJE1** = OIB, datumCijepljenje, rbrCijepljenje, sifraPunkt, nazivPunkt, sifraLijechnik, prezimeLijechnik, imeLijechnik, sifraVrCjepivo, nazivVrCjepivo, serijaCjepivo

K1 **CIJEPLJENJE1** = OIB, datumCijepljenje - PK

K2 **CIJEPLJENJE1** = OIB, rbrCijepljenje

**Normalizacija na 3NF**

**CIJEPLJENJE2** = OIB, datumCijepljenje, rbrCijepljenje, sifraPunkt, sifraLijecnik, sifraVrCjepivo, serijaCjepivo

K1 <sub>CIJEPLJENJE2</sub> = OIB, datumCijepljenje - PK

K2 <sub>CIJEPLJENJE2</sub> = OIB, rbrCijepljenje

**PUNKT** = sifraPunkt, nazivPunkt

K <sub>PUNKT</sub> = sifraPunkt

**LIJECNIK** = sifraLijecnik, prezimeLijecnik, imeLijecnik

K <sub>LIJECNIK</sub> = sifraLijecnik

**VRCJEPIVO** = sifraVrCjepivo, nazivVrCjepivo

K <sub>VRCJEPIVO</sub> = sifraVrCjepivo

**Konačna shema baze podataka sadrži sljedeće relacijske sheme:**

OSOBA, CIJEPLJENJE2, LIJECNIK, PUNKT, VRCJEPIVO



d) Aplikacija za planiranje cijepljenja protiv COVID-19 u svojoj bazi podataka evidentira podatke o punktovima na kojima će se obaviti cijepljenje, datumima kada će se obavljati cijepljenje, vrstama i količinama cjepiva za određeni punkt određenog dana.

Schema **PLAN\_CIEPLJENJE** se sastoji od sljedećih atributa:

<b>sifraPunkt</b>	– šifra punkta na kojem će se obavljati cijepljenje
<b>nazivPunkt</b>	– naziv punkta na kojem će se obavljati cijepljenje
<b>adresaPunkt</b>	– adresa punkta na kojem će se obavljati cijepljenje
<b>postBr</b>	– poštanski broj punkta na kojem će se obavljati cijepljenje
<b>nazivMjesto</b>	– naziv mjesta punkta na kojem će se obavljati cijepljenje
<b>datumCijepljenje</b>	– datum cijepljenja
<b>sifraVrCjepivo</b>	– šifra vrste cjepiva
<b>nazivVrCjepivo</b>	– naziv vrste cjepiva
<b>brojDoza</b>	– broj doza određenog cjepiva određenog dana za jedan punkt

Jednoga dana na jednom punktu može biti na raspolaganju više različitih vrsta cjepiva i za svaku se vrstu evidentira broj raspoloživih doza.

Odrediti ključ relacijske sheme **PLAN\_CIEPLJENJE** tako da ona bude u 1NF, a zatim **postupno** normalizirati relacijsku shemu na 2NF i 3NF.

Početna shema:

**PLAN\_CIEPLJENJE** = sifraPunkt, nazivPunkt, adresaPunkt, postBr, nazivMjesto, datumCijepljenje, sifraVrCjepivo, nazivVrCjepivo, brojDoza

#### **Normalizacija na 1NF**

Vrijede sljedeće FZ:

- sifraPunkt → nazivPunkt, adresaPunkt, postBr, nazivMjesto
- sifraCjepivo → nazivVrCjepivo
- sifraPunkt, datumCijepljenje, sifraVrCjepivo → sifraPunkt, nazivPunkt, adresaPunkt, postBr, nazivMjesto, nazivVrCjepivo, brojDoza

**PLAN\_CIEPLJENJE** = sifraPunkt, nazivPunkt, adresaPunkt, postBr, nazivMjesto, datumCijepljenje, sifraVrCjepivo, nazivVrCjepivo, brojDoza

K<sub>PLAN\_CIEPLJENJE</sub> = sifraPunkt, datumCijepljenje, sifraVrCjepivo

#### **Normalizacija na 2NF**

**PUNKT1** = sifraPunkt, nazivPunkt, adresaPunkt, postBr, nazivMjesto

K<sub>PUNKT1</sub> = sifraPunkt

**VRCJEPIVO** = sifraVrCjepivo, nazivVrCjepivo

K<sub>VRCJEPIVO</sub> = sifraVrCjepivo

**PLAN\_CIEPLJENJE1** = sifraPunkt, datumCijepljenje, sifraVrCjepivo, brojDoza

K<sub>PLAN\_CIEPLJENJE1</sub> = sifraPunkt, datumCijepljenje, sifraVrCjepivo

#### **Normalizacija na 3NF**

**PUNKT2** = sifraPunkt, nazivPunkt, adresaPunkt, postBr

K<sub>PUNKT2</sub> = sifraPunkt

**MJESTO** = postBr, nazMjesto

K<sub>MJESTO</sub> = postBr

***Konačna shema baze podataka sadrži sljedeće relacijske sheme:***

PLAN\_CIJEPLJENJE1, PUNKT2, VRCJEPIVO, MJEST