

Za filmove koji

- imaju prijevod (subtitle) na jeziku naziva "Arabic" i
- naslov im
 - **sadrži** određeni član "the" (neovisno sadrži li velika ili mala slova) ili
 - **počinje** neodređenim članom "An" (koji obavezno počinje velikim slovom)

ispišite naziv **filma**, prihod filma i informaciju o generaciji kojoj pripada.

Informaciju o generaciji odredite na sljedeći način:

- ako je film izdan između 1980. i 1990. (obje granice uključene), njegova oznaka generacije je "gen x"
- ako je film izdan između 1991. i 2018.(obje granice uključene), njegova oznaka generacije je "gen y"
- ako je film izdan nakon 2018., njegova oznaka generacije je "gen z"
- u svim ostalim slučajevima, oznaka generacije je "ostalo"

Primjer ispisa (ne mora odgovarati rezultatu):

tracktitle	boxincome	generation
The Other Side of the Wind	136490234	gen z
An Upper Egyptian	null	gen y
...
Disciples Of The 36th Chamber	62688139	gen y

Correct answer:

```
1 select tracktitle, boxincome,
2     case
3         when extract(year from releasedate) between 1980 and 1990 then 'gen x'
4         when extract(year from releasedate) between 1991 and 2018 then 'gen y'
5         when extract(year from releasedate) > 2018 then 'gen z'
6         else 'ostalo' end
7     as generation
8 FROM track
9 NATURAL JOIN movie
10 NATURAL JOIN subtitle
11 JOIN language ON subtitle.subtitlelangid = language.langid
12 where (lower(tracktitle) like '%the %' or tracktitle like 'An %')
13 AND langname = 'Arabic'
```

Ispišite inicijal imena (npr. ‘M.’ za ime ‘Mia’), prezime, datum rođenja i mjesečnu cijenu **aktivnog** paketa za vlasnike korisničkog računa koji su rođeni u **svibnju ili lipnju** te imaju **aktivan** paket koji je aktiviran u periodu **16 mjeseci i 16 dana unazad od trenutnog datuma** (granica uključena).

Pretpostavite da osoba ima samo jedno ime.

Rezultate poredajte silazno po mjesečnoj cijeni paketa te uzlazno po prezimenu.

Primjer ispisa (ne mora odgovarati rezultatu):

firstinitial	lastname	dateofbirth	pricepermonth
L.	Beil	10.06.1988	12
R.	Rodgers	08.06.1991	12
M.	Roth	20.05.2002	12
...

```
1 SELECT SUBSTRING(firstName, 1, 1) || ' ' AS firstInitial, lastName, dateOfBirth, pricePerMonth
2 FROM owner NATURAL JOIN ownerPack NATURAL JOIN pack
3 WHERE (EXTRACT(MONTH FROM dateOfBirth) = 5 OR EXTRACT(MONTH FROM dateOfBirth) = 6)
4     AND packStartDateTime::DATE >= CURRENT_DATE - '16 MONTH 16 DAY'::INTERVAL
5     AND (packEndDateTime::DATE IS NULL OR packEndDateTime::DATE >= CURRENT_DATE)
6 ORDER BY pricePerMonth DESC, lastName ASC;
```


Za svaki film čija je zarada između 50 milijuna i 228 milijuna dolara (obje granicu su uključene), bez obzira ima li prethodnika ili ne, ispisati naziv filma. Također ispisati naziv filma prethodnika, ali samo za one filmove koji na početku naslova sadrže naslov prethodnika.

Zapise poredati uzlazno prema nazivu filma prethodnika, a potom uzlazno prema nazivu filma (movieTitle).

Ispis ograničiti na 100 zapisa.

Primjer rezultata (podaci ne moraju odgovarati stvarnima!):

movieTitle	prevMovieTitle
#FriendButMarried 2	#FriendButMarried
A Christmas Prince: The Royal Wedding	A Christmas Prince
Final Destination 2	Final Destination
Chitty Chitty Bang Bang	NULL
Tinker Bell and the Legend of the NeverBeast	NULL

Correct answer:

```
1 SELECT track.trackTitle MovieTitle, trackPrev.trackTitle AS prevMovieTitle
2     FROM track
3         NATURAL JOIN movie
4         --ON track.trackId = movie.trackId    --bez NATURAL
5     LEFT JOIN track AS trackPrev
6         ON movie.prevMovieId = trackPrev.trackId
7     AND SUBSTRING (track.trackTitle FROM 1 FOR
8         CHAR_LENGTH(trackPrev.trackTitle)) = trackPrev.trackTitle
9         -- ili AND track.trackTitle LIKE trackPrev.trackTitle || '%'
10    WHERE boxincome BETWEEN 50000000 AND 228000000
11    ORDER BY prevMovieTitle, track.trackTitle
12    LIMIT 100;
```

Ispišite naziv korisničkog profila, identifikator vlasnika profila te identifikator serije od koje je osoba koja se nalazi iza korisničkog profila pogledala više od 30% **različitih** epizoda, u odnosu na ukupan broj objavljenih epizoda te serije, neovisno o sezoni.

Epizoda se smatra pogledanom čak i ako je osoba prekinula gledanje unutar 5 minuta od kraja epizode. Ako je osoba prekinula gledanje epizode prije više od 5 minuta od kraja epizode, epizoda se ne smatra pogledanom.

Uz identifikator vlasnika, naziv profila i identifikator serije potrebno je dodatno ispisati i koliko je pogledanih epizoda te serije ocijenila osoba koja se nalazi iza tog korisničkog profila.

Primjer rezultata (podaci ne moraju odgovarati stvarnima!):

ownerid	profilename	showid	count
276	ShirleyCook	652	2
269	ShirleyCook	523	5
302	disastercourse	825	0
...

Correct answer:

```
1 select profile.ownerid, profilename, showep.showid, (SELECT count(*)
2         from profiletrack natural join showep as s
3         where profiletrack.ownerid = profile.ownerid
4         and profiletrack.profilename =profile.profilename
5         and s.showid = showep.showid
6         and liked is not null)
7 from profile
8     natural join trackview
9     natural join track
10    natural join showep
11 where duration <= (savedProgress + '5 minute'::INTERVAL)
12 group by profile.ownerid, profilename, showep.showid
13 having count(distinct trackid) > 0.3 * (select count(*)
14         from showep as sep
15         where sep.showid = showep.showid)
```


Odgovor upisujete u tekstualnu kućicu ispod pitanja, na način prikazan u stupcu **Tekstualni oblik**.

Izraz relacijske algebre	Opis	Tekstualni oblik
$r1 \bowtie r2$	Prirodno spajanje	$r1 \bowtie r2$
$r1 \bowtie_{att1=att2 \wedge (att3>att4 \vee att5 \leq att6)} r2$	Spajanje uz uvjet	$r1 \bowtie_{att1=att2 \wedge (att3>att4 \vee att5 \leq att6)} r2$
$r1 * \bowtie_{att1=att2} r2, r1 \bowtie *_{att1=att2} r2$	Vanjska spajanja uz uvjet	$r1 *_{att1=att2} r2, r1 \bowtie *_{att1=att2} r2$
$\sigma_{att1 \neq att2 \wedge att2 = 'test'}(r1)$	Selekcija	$\sigma_{att1 \neq att2 \wedge att2 = 'test'}(r1)$
$\pi_{att1, att2}(r1)$	Projekcija	$\pi_{att1, att2}(r1)$
$\rho_{r2(att1, att2)}(r1)$	Preimenovanje	$\rho_{r2(att1, att2)}(r1)$
$att1, att2 \mathcal{G}_{SUM(att3)}(r1)$	Agregacija	$\{att1, att2\}G\{SUM(att3)\}(r1)$
$r1 \cup r2, r1 \cap r2, r1 \setminus r2, r1 \times r2$	Unija, Presjek, Razlika, Katerzijev produkt	$r1 \cup r2, r1 \cap r2, r1 \setminus r2, r1 \times r2$

Na primjer, izraz relacijske algebre

$$\rho_{student(mbr, ime, prezime)}(\pi_{jmbag, ime, prezime}(student \bowtie evidencijaboravka))$$

upisujete kao

$$\rho\{student(mbr, ime, prezime)\}(\pi\{jmbag, ime, prezime\}(student \bowtie evidencijaboravka))$$

U sljedećoj tablici možete naći simbole koje možete koristiti (copy-paste)

Naziv	Tekstualni oblik
Relacijski operatori	$\rho \ \pi \ \sigma \ G \cup \cap \setminus \bowtie \ *_{\bowtie} \ \bowtie \ * \times$
Logički operatori i usporedba	$<= \ >= \ \neq \ = \ < \ > \ \wedge \ \vee$

Zadatak 5. (3 boda)

Napišite izraz relacijske algebre koji predstavlja sljedeći upit: ispisati nazive serija (*showtitle* u *show*) za koje je **dana ocjena veća od 8** (*showrating*), te ukupno trajanje svih epizoda **prve sezone** tih serija. Ispis treba izgledati kao u sljedećoj tablici (podaci u ovoj tablici su simbolični):

serije

serija	trajanje
Dexter	04:12:00
Star Trek: Voyager	07:44:00
...	...

Correct answer:

$\rho\{\text{serije}(\text{serija}, \text{trajanje})\}(\{\text{showtitle}\}G\{\text{SUM}(\text{duration})\}(\text{track} \bowtie \sigma\{\text{seasonno}=1\}(\text{showep} \bowtie \sigma\{\text{showrating}>8\}(\text{show}))))$

$\pi\{\text{serija}, \text{trajanje}\}(\rho\{\text{serija}(\text{id}, \text{serija}, \text{trajanje})\}(\{\text{showid}, \text{showtitle}\}G\{\text{SUM}(\text{duration})\}(\sigma\{\text{showrating}>8 \wedge \text{seasonno}=1\}(\text{show} \bowtie \text{showep} \bowtie \text{track}))))$

ili neka kombinacija ovog dvojeg.... razlika može biti u spajanju i selekciji, a vanjski operatori bi trebali biti kao u gornjim odgovorima.

Za B+-stablo reda 4 koje sadrži zapise s ključevima: 5, 9, 13, 15, 25, 33, 50, 60, 63, 76, 86:

- izračunajte broj kazaljki za svaku vrstu čvora i to za minimalnu i za maksimalnu popunjenost stabla
- nacrtajte B+-stablo tako da popunjenost stabla bude minimalna.

Napomena:

- B+- stablo nacrtajte u alatu draw.io prema dolje navedenim uputama, spremite u formatu .drawio te priložite kao dio rješenja zadatka (File uploads).
- Izračune u vezi broja kazaljki u čvorovima upišite **u prostor za slobodni unos teksta ispod zadatka**.

Preuzmite priloženu datoteku *bstablo_cvorovi_shapes.drawio* (Attachment(s)).

U web pregledniku otvorite draw.io u novom prozoru.

- Odaberite opciju *Import -> From device* te označite preuzetu .drawio datoteku. U gornjem lijevom kutu će se pojaviti već kreirani čvorovi s različitim brojem kazaljki. Služe za lakše crtanje B+ stabla.
- Crtež **spremite lokalno u formatu .drawio**, zatim u Edgaru odaberite spremljenu datoteku (**Browse**) te konačno **potvrdite klikom na gumb desno od Browse**. Na ekranu ćete moći vidjeti koju ste datoteku uploadali.

Correct answer:

MIN:

$K = 2$

$\check{C} = \lceil n/2 \rceil = 2$

$L = \lceil (n-1)/2 \rceil = 2$

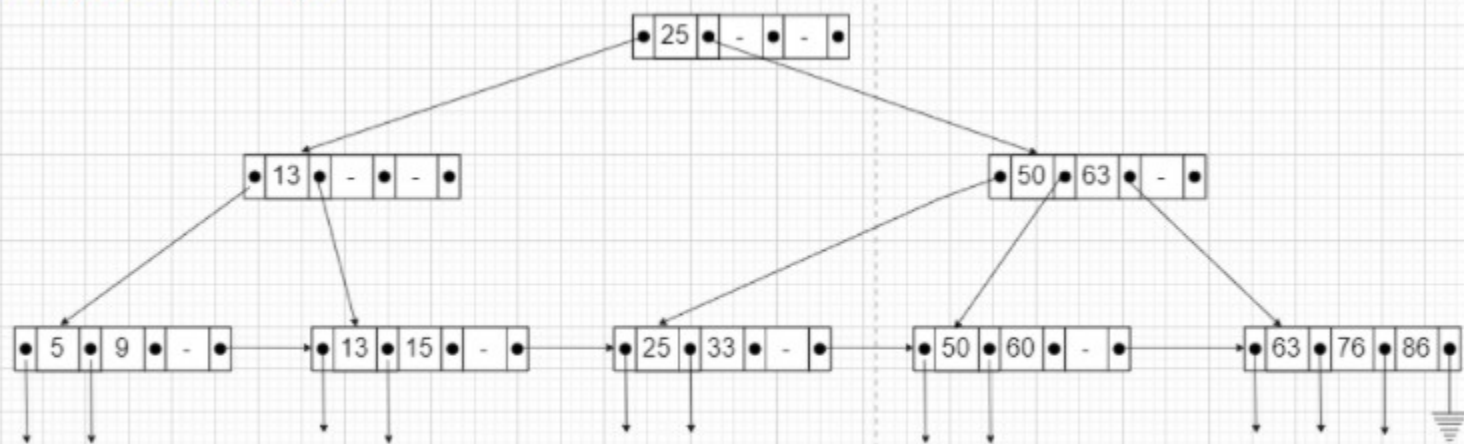
MAX:

$K = 4$

$\check{C} = 4$

$L = n-1 = 3$

Priznaje se sljedeće točno rješenje:



U *streamFlix* bazi kreirane su sljedeće tablice s definiranim pravilima integriteta:

Naredbe

```
CREATE TABLE owner (  
    --  
    , PRIMARY KEY (ownerId)  
);
```

```
CREATE TABLE profile (  
    --  
    , PRIMARY KEY (ownerId, profileName)  
    , FOREIGN KEY (ownerId) REFERENCES owner(ownerId)  
);
```

```
CREATE table profiletrack (  
    --  
    , PRIMARY KEY (ownerId, profileName, trackid)  
    , FOREIGN KEY (ownerId, profileName) REFERENCES  
profile(ownerId, profileName)  
);
```

```
CREATE TABLE trackview (  
    --  
    , FOREIGN KEY (ownerId, profileName) REFERENCES profile (ownerId,  
profileName)  
    , FOREIGN KEY (ownerId, profileName, trackid) REFERENCES  
profiletrack (ownerId, profileName, trackid)  
);
```

```
CREATE TABLE ownerpack (  
    --  
    , PRIMARY KEY (packId, ownerId, packStartDateTime)  
    , FOREIGN KEY (ownerId) REFERENCES owner(ownerId)  
);
```

Napišite niz naredbi čijim izvršavanjem će biti obrisani vlasnik *streamFlix* računa s identifikatorom (*ownerId*) 2 i podaci u svim ostalim tablicama koji su direktno ili indirektno povezani s njim. Naredbe moraju biti navedene redoslijedom kojim će se traženo brisanje izvršiti bez pogreške.

Ako postoji više mogućih redoslijeda, dovoljno je napisati samo jedan.

Napomena: Iznad su navedena samo ona pravila integriteta koja su nužna za rješavanje zadatka. Navedene CREATE TABLE naredbe ne smiju se mijenjati.

Correct answer:

Redoslijed	Naredba
1.	<code>DELETE FROM trackview WHERE ownerId = 2;</code> <code>DELETE FROM ownerpack WHERE ownerId = 2;</code>
2.	<code>DELETE FROM profiletrack WHERE ownerId = 2;</code>
3.	<code>DELETE FROM profile WHERE ownerId = 2;</code>
4.	<code>DELETE FROM owner WHERE ownerId = 2;</code>

ili

Redoslijed	Naredba
1.	<code>DELETE FROM trackview WHERE ownerId = 2;</code>
2.	<code>DELETE FROM profiletrack WHERE ownerId = 2;</code>
3.	<code>DELETE FROM profile WHERE ownerId = 2;</code> <code>DELETE FROM ownerpack WHERE ownerId = 2;</code>
4.	<code>DELETE FROM owner WHERE ownerId = 2;</code>

Za naredbe koje se nalaze u istom retku nije bitan redoslijed (npr. u tablici 1 nije bitno koja će se od prvih dviju naredbi izvesti prva).

Pretpostavimo da u bazi podataka streamFlix postoji (nenormalizirana) relacijska shema:

SHOW_UN = trackId, showId, seasonNo, episodeNo, showTitle, showRating

te je na njoj definiran skup funkcijskih zavisnosti:

$F = \{\text{trackId} \rightarrow \text{showId}, \text{seasonNo}, \text{episodeNo}; \text{showId} \rightarrow \text{showTitle}, \text{showRating}\}$

Pokažite da u relacijskoj shemi SHOW_UN vrijedi funkcijska zavisnost: **trackId \rightarrow showTitle, showRating**

Za svaki korak dokaza napisati aksiome ili pravila koje ste pritom koristili.

Svoj odgovor upišite u tekstualni okvir ispod pitanja!

Correct answer:

a) Primjena pravila o akumulaciji

a1: trackId -> trackId (polazno)

a2: trackId -> trackId &

trackId -> showId, seasonNo, episodeNo

=> trackId -> trackId, showId, seasonNo, episodeNo (akumulacija)

a3: trackId -> trackId, showId, seasonNo, episodeNo &

showId -> showTitle, showRating

=> trackId -> trackId, showId, seasonNo, episodeNo, showTitle, showRating (akumulacija)

a4: trackId -> showTitle, showRating (dekompozicija)

Zaključak: u relacijskoj shemi SHOW_UN postoji FZ : trackId -> showTitle, showRating

b) Primjena pravila o dekompoziciji i aksioma o tranzitivnosti

b1: trackId-> showId, seasonNo, episodeNo

=> trackId -> showId (dekompozicija)

b2: trackId -> showId &

showId -> showTitle, showRating

=> trackId -> showTitle, showRating (tranzitivnost)

Zaključak: u relacijskoj shemi SHOW_UN postoji FZ : trackId -> showTitle, showRating

Gradska komunalna služba vodi evidenciju o izdanim kaznama za parkiranje nakon isteka plaćenog vremena parkiranja.

Relacijska shema **PARKING** koja pohranjuje navedene informacije sadrži sljedeće attribute:

Naziv atributa	Opis atributa
sifKontrolor	šifra kontrolora
prezimeKontrolor	prezime kontrolora
imeKontrolor	ime kontrolora
registracija	registracija vozila
modelVoz	model vozila
bojaVoz	boja vozila
sifZona	šifra zone parkiranja
nazivZona	naziv zone parkiranja
brojMinuta	broj minuta nakon isteka plaćenog perioda parkiranja
datumKazna	datum izdavanja kazne
sifraUlica	šifra ulice gdje je izdana kazna
nazivUlica	naziv ulice gdje je izdana kazna

Grad je podijeljen na zone parkiranja koje imaju svoju šifru i naziv. Kontrolori koji uoče parkirano vozilo kojemu je isteklo plaćeno vrijeme parkiranja zapisuju lokaciju vozila u obliku šifre i naziva ulice u kojoj se vozilo nalazi, šifru i naziv parkirališne zone, uočeni broj minuta nakon isteka vremena te datum izdavanja kazne. Uz svaki zapis pohranjuju se i informacije o kontroloru koji je izdao kaznu. Kazna je u obliku dnevne karte za parkiranje, tako da isto vozilo na isti dan ne može dobiti više kazni unutar iste zone parkiranja. Ulica se može protezati kroz više parkirališnih zona.

Početna relacijska shema je:

PARKING = sifKontrolor, prezimeKontrolor, imeKontrolor, registracija, modelVoz, bojaVoz, sifZona, nazivZona, brojMinuta, datumKazna, sifUlica, nazivUlica

Ispišite sve prepoznate funkcijske zavisnosti.

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

Correct answer:

Početna shema:

PARKING = sifKontrolor, prezimeKontrolor, imeKontrolor, registracija, modelVoz, bojaVoz, sifZona, nazivZona, brojMinuta, datumKazna, sifUlica, nazivUlica

Vrijede sljedeće FZ:

- sifKontrolor* → prezimeKontrolor, imeKontrolor
- registracija* → modelVoz, bojaVoz
- sifZona* → nazivZona
- sifUlica* → nazivUlica
- registracija, sifZona, datumKazna* → sifKontrolor, prezimeKontrolor, imeKontrolor, modelVoz, bojaVoz, nazivZona, brojMinuta, sifUlica, nazivUlica

Normalizacija na 1NF

PARKING = sifKontrolor, prezimeKontrolor, imeKontrolor, registracija, modelVoz, bojaVoz, sifZona, nazivZona, brojMinuta, datumKazna, sifUlica, nazivUlica
K_PARKING = registracija, sifZona, datumKazna

Normalizacija na 2NF

VOZILO = registracija, modelVoz, bojaVoz
K_VOZILO = registracija

ZONA = sifZona, nazivZona
K_ZONA = sifZona

PARKING2 = sifKontrolor, prezimeKontrolor, imeKontrolor, registracija, sifZona, brojMinuta, datumKazna, sifUlica, nazivUlica
K_PARKING2 = registracija, sifZona, datumKazna

Normalizacija na 3NF

VOZILO = registracija, modelVoz, bojaVoz
K_VOZILO = registracija

ZONA = sifZona, nazivZona
K_ZONA = sifZona

ULICA = sifUlica, nazivUlica
K_ULICA = sifUlica

KONTROLOR = sifKontrolor, prezimeKontrolor, imeKontrolor
K_KONTROLOR = sifKontrolor

PARKING3 = sifKontrolor, registracija, sifZona, brojMinuta, datumKazna, sifUlica
K_PARKING3 = registracija, sifZona, datumKazna

Konačna shema baze podataka
VOZILO, ZONA, ULICA, KONTROLOR, PARKING3