

U zadacima 1, 2, 3 i 4 se pretpostavlja korištenje PostgreSQL SUBP-a i baze podataka sa slike 1.

Segment baze podataka prikazan na **slici 1** služi za evidenciju podataka o putovanjima autobusom u ponudi turističke agencije. U relaciji **putLok** su opisane dionice konkretnog putovanja. Atribut **putLok.geom** predstavlja dionicu ceste između početne i završne lokacije te dionice. Atribut **putTermin.cijenaHRK** predstavlja cijenu putovanja za jednu osobu. Ključevi relacija su podcrtani.

lokacija				putovanje			drzava		
<u>sifLok</u>	<u>nazivLok</u>	<u>oznDrz</u>	<u>geom</u>	<u>sifPut</u>	<u>nazivPut</u>	...	<u>oznDrz</u>	<u>nazivDrz</u>	<u>geom</u>
11	Split	HR	<point>	20	Ljepote Jadrana		HR	Hrvatska	<polygon>
12	Plitvice	HR	<point>	21	Vikend u Dubrovniku		AT	Austrija	<polygon>
13	Dubrovnik	HR	<point>	23	Austrijska avantura		PL	Poljska	<polygon>
...	24	Finske bijele noći	

putLok					putTermin				
<u>sifPut</u>	<u>rbrDio</u>	<u>sifLokPoc</u>	<u>sifLokZav</u>	<u>geom</u>	<u>sifPut</u>	<u>datPolazak</u>	<u>cijenaHRK</u>	<u>ukMjesta</u>	<u>prodano Mjesta</u>
20	2	12	11	<polyline>	20	01.05.2016	1200	50	50
20	3	11	13	<polyline>	20	01.07.2016	1400	80	72
21	1	8	13	<polyline>	21	15.06.2016	400	50	46
21	4	11	15	<polyline>	22	04.06.2016	3200	50	50
21	6	16	13	<polyline>

Slika 1.

1. (4 boda) Po godinama u kojima su putovanja realizirana, ispisati godinu i naziv putovanja te ukupan i kumulativni broj kilometara koje autobusi pređu prevozeći putnike na tom putovanju. Npr. kumulativni broj kilometara za putovanje naziva 'Ljepote Jadrana' u 2016. godini pored broja kilometara pređenih 2016. uključuje i kilometre pređene (za to isto putovanje) u 2015. te kilometre u svim godinama prije 2015.

godina	nazivPut	kmGodina	kumulKm
...
2015	Ljepote Jadrana	5000	50000
2016	Ljepote Jadrana	6000	56000
...

Pomoć: godinu iz datuma možete izdvojiti pomoću funkcije EXTRACT (npr. EXTRACT (YEAR FROM imeAtributa)).

2. (4 boda) Za sva putovanja ispisati podatke o ukupnoj zaradi po mjesecima u sljedećem obliku:

putovanje	sijecanj	veljaca	ozujak	april							prosina
Jug Dalmacije	16864.75		25634.00	123265.50
Ljepote Jadrana

Pomoć: Upotrijebiti funkciju **crosstab**. Voditi računa da se podaci moraju ispravno ispisati i u slučaju da postoje mjeseci u kojima nije bilo polazaka za određeno putovanje.

3. (4 boda) Korištenjem PostgreSQL sintakse potrebno je napisati SQL naredbe za kreiranje relacije putovanjeOR čija je **shema i sadržaj** skiciran donjom tablicom a definiran shemom i sadržajem tablica sa slike 1.

<u>sifPut</u> <i>smallint</i>	<u>nazivPut</u> <i>ncharacter(250)</i>	<u>lokacije</u> <i>lokacija[]</i>
20	Ljepote Jadrana	"{" 8, \"Zagreb\", \"HR\",), \" 9, \"Plitvice\", \"HR\",), \"10, \"Zadar\", \"HR\",), ... }"
21	Vikend u Dubrovniku	...

U polju lokacije se pojavljuju sve lokacije koje putnici obilaze na određenom putovanju. Voditi računa da se lokacije ne ponavljaju.

4. **(5 bodova)** Napišite jedan ili više upita kojim ćete dobiti popis međunarodnih putovanja koja kreću iz Hrvatske, pri čemu za svako putovanje treba ispisati ukupno duljinu, te duljinu putovanja po cestama izvan Hrvatske.

5. **(5 bodova)** Napisati MapReduce upit koji će na temelju tablice putTermin za svako putovanje (šifru) ispisati kolika je promjena prihoda u postotcima u 2016. godini u odnosu na 2015. godinu, npr.

šifPut	Promjena
13	-10%
15	20%
...	...

Pretpostaviti istu M/R arhitekturu (s obzirom na (*combinable*) *reducer* i sl.) kao u MongoDB.

6. **(3 boda)** Objasnite mehanizam **kvoruma** kod ostvarivanja konzistencije. Navedite primjere i za čitanje i za pisanje gdje se objašnjava kako se (ne)postiže konzistencija.

7. **(5 bodova)** Napravite RDF model podataka (nacrtajte graf) za jednostavno obiteljsko stablo kojim ćete opisati odnose majka/otac, brat/sestra i kći/sin za neku izmišljenu obitelj.

Minimalan skup svojstava koje modelom treba moći opisati za osobu su ime, prezime i spol. U modelu možete koristiti proizvoljne (postojeće ili zamišljene) rječnike. Vodite računa o tome da svaku osobu morate moći jedinstveno identificirati.

Na temelju vlastitog modela napišite SPARQL upit koji će vratiti prezimena i imena osoba čije bake imaju jednako ime. Primjer: Marija Car i Marija Lončar su bake Ane Horvat pa se Ana Horvat treba nalaziti u rezultatu upita.

Rješenja:

1. (4 boda)

```
SELECT EXTRACT (YEAR FROM datPolazak), putovanje.nazivPut,
       SUM(ST_length(putLok.geom)),
       SUM(SUM(ST_length(putLok.geom))) OVER (PARTITION BY putovanje.sifPut
                                              ORDER BY EXTRACT (YEAR from datPolazak)
                                              ROWS BETWEEN UNBOUNDED PRECEDING
                                              AND CURRENT ROW)
--prethodna 2 retka se mogu izostaviti jer je to i defaultna definicija granica okvira
FROM putovanje, putLok, putTermin
WHERE putovanje.sifPut = putLok.sifPut
  AND putovanje.sifPut = putTermin.sifPut
GROUP BY EXTRACT (YEAR from datPolazak), putovanje.nazivPut, putovanje.sifPut
```

2. (4 boda)

```
CREATE EXTENSION tablefunc;
CREATE TEMP TABLE mjesec
(rbrMjesec int);
INSERT INTO mjesec VALUES ( 1);
INSERT INTO mjesec VALUES ( 2);
...
INSERT INTO mjesec VALUES ( 12);

SELECT *
FROM crosstab ('SELECT putovanje.nazivPut AS naziv
                  , CAST(EXTRACT(MONTH FROM datPolazak) AS int) AS mjesec
                  , CAST(SUM(putTermin.cijenaHrk*prodanoMjesta)
                        AS decimal (10,2)) AS zarada
                  FROM putovanje, putTermin
                  WHERE putovanje.sifPut = putTermin.sifPut
                  GROUP BY putovanje.nazivPut, EXTRACT(MONTH FROM datPolazak)
                  ORDER BY putovanje.nazivPut, EXTRACT(MONTH FROM datPolazak)'
, 'SELECT rbrMjesec FROM mjesec ORDER BY rbrMjesec')
AS pivotTable (nazivPutovanje CHAR(50)
, siječanj decimal (10,2), veljača decimal (10,2)
, ožujak decimal (10,2), travanj DECIMAL(10,2)
, svibanj DECIMAL(10,2), lipanj DECIMAL(10,2)
, srpanj DECIMAL(10,2), kolovoz DECIMAL(10,2)
, rujanj DECIMAL(10,2), listopad DECIMAL(10,2)
, studeni DECIMAL(10,2), prosinac DECIMAL(10,2))
```

3. (4 boda)

Budući da kreiranjem tablice u bazi podataka nastaje i istoimeni tip, tip lokacija možemo iskoristiti kao tip kojeg će biti elementi polja putovanjeOR.lokacije. Nije potrebno definirati ekstra tip.

```
CREATE TABLE putovanjeOR
(sifPut smallint CONSTRAINT pkPutovanjeOR PRIMARY KEY,
 nazivPut NCHAR(250),
 lokacije lokacija[]
);
INSERT INTO putovanjeOR
SELECT putovanje.*
, (SELECT array_agg(DISTINCT lokacija.*)
   FROM putLok, lokacija
   WHERE putovanje.sifPut = putLok.sifPut
     AND (lokacija.sifLok = putLok.sifLokPoc OR
          lokacija.sifLok = putLok.sifLokZav))
FROM putovanje
DISTINCT treba da se izbjegne ponavljanje lokacija.
CAST nije potreban u array_agg(CAST( ROW(DISTINCT lokacija.*) AS lokacija))
jer je lokacija.* istog tipa kao i element polja putovanjeOR.lokacije
```

4. (5 bodova)

// može i bez pomoćnog VIEW-a sve u jednom upitu:

```
CREATE VIEW putMedjSPolaskomIzHR AS
SELECT DISTINCT putovanje.sifPut,
                putovanje.nazivPut
FROM putovanje
JOIN putLok     ON putovanje.sifPut= putLok.sifPut
                AND rbrDio = 1
JOIN lokacija  ON putLok.sifLokPoc = lokacija.sifLok
WHERE lokacija.oznDrzava = 'HR'
AND EXISTS (SELECT *
            FROM putLok
            JOIN lokacija ON putLok.sifLokPoc = lokacija.sifLok
            WHERE sifPut = putovanje.sifPut
            AND oznDrz <> 'HR')

SELECT putMedjSPolaskomIzHR.*,
       SUM(st_length(putLok.geom)) as ukPut,
       SUM(st_length(st_intersection(putLok.geom,
                                     (SELECT st_union(geom) FROM drzava WHERE oznDrz <> 'HR')--sve ostale
                                     )))
FROM putMedjSPolaskomIzHR
JOIN putLok ON putMedjSPolaskomIzHR.sifPut = putLok.sifPut
```

--treba paziti, neke dionice mogu biti skroz vanjske, ili između dvije strane države, ---npr. Putovanje iz Hr u Mađ u Austr u Slo u Hr.

Može još elegantnije ako se napravi presjek s Hrvatskom, pa izračuna ukupno minus ceste po Hr:

```
putMedjSPolaskomIzHR.*,
SUM(st_length(putLok.geom)) as ukPut,
SUM(st_length(putLok.geom)) -
SUM(st_length(st_intersection(putLok.geom,
                              (SELECT st_union(geom) FROM drzava WHERE oznDrz = 'HR')
                              )))
)))
```

5. (5 bodova)

```
map () {
  god = YEAR(this.datPolazak)
  emit(this.sifPut, {
    g15: god == 2015 ? (this.cijenaHRK*this.prodanoMjesta) : 0,
    g16: god == 2016 ? (this.cijenaHRK*this.prodanoMjesta) : 0
  });
}
reduce (k, vlist) {
  var uk = {g15: 0, g16: 0};
  foreach (v in vlist) {
    uk.g15 += v.g15;
    uk.g16 += v.g16;
  }
  return uk;
}
finalize (k, v) {
  return { k, (v.g16/v.g15) * 100 - 100 };
}
```

6. (3 boda)

Kvorum je mehanizam kojim je moguće ostvariti konzistenciju u distribuiranim sustavima. Neka imamo sustav s 1000 čvorova (nije bitno), a faktor replikacije $N = 3$, što znači da je isti podatak zapisan na 3 mjesta od tih 1000 čvorova. Ako želimo osigurati konzistenciju kod pisanja, nužno je da novi zapis upišemo na barem:

$$W > N/2$$

mjesta (W je broj pisanja) zato jer onda kasnije, kod čitanja, možemo pročitati iste te, što čini većinu i pobjeđuje kod odlučivanja što je ispravna verzija. Npr. Neka je za studenta pisalo da ima 20 godina na 3 mjesta – kada upišemo da sad ima 21 godinu na barem 2 mjesta, sigurni smo da će naknadno kod glasanja biti barem dva zapisa s 21, što će odnijeti prevagu nad verzijom gdje se tvrdi da student ima 20 godina.

Konzistentno pisanje ne znači da kasnije ostvarujemo konzistentno čitanje!

Naime, moram pročitati ta dva zapisa da bismo bili sigurni da smo ostvarili konzistenciju. Ako pročitamo npr. samo jedan, možemo u našem primjeru krivo zaključiti da student ima 20 godina (ako pogodimo baš taj čvor u kojem piše 20).

Konzistencija kod čitanja je tako povezana s W, te je za ostvariti konzistenciju dovoljno:

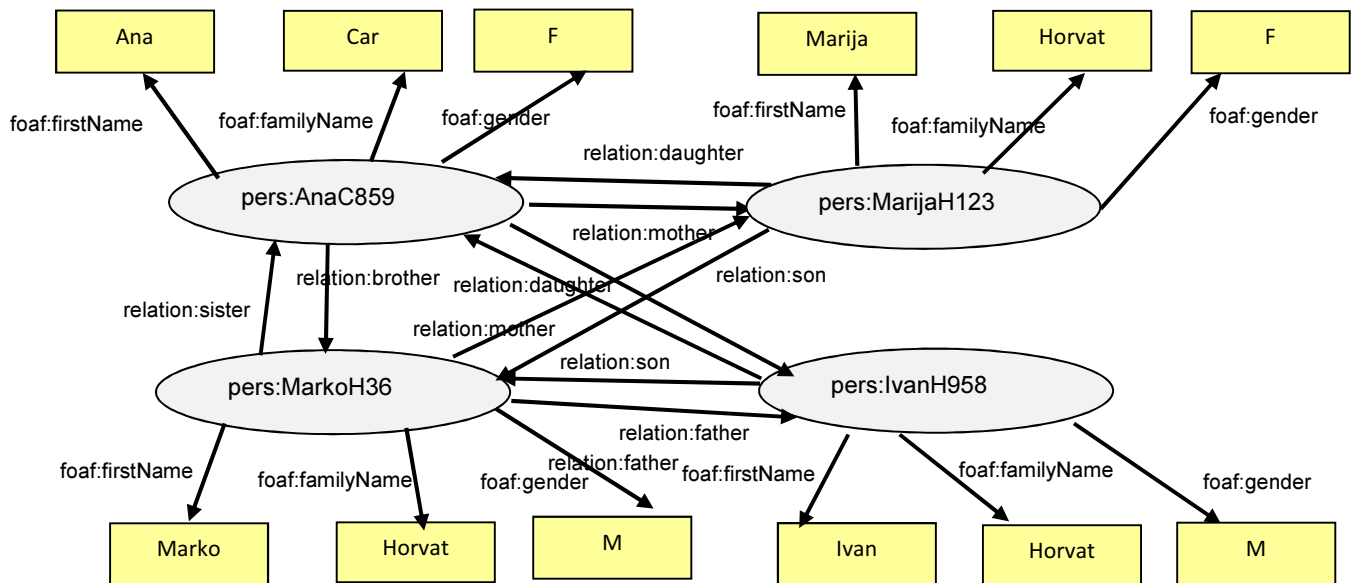
$$R + W > N$$

U našem primjeru, ako je $W = 2$ i $N = 3$, slijedi da $R > 1$.

Sve te parametre (R, W) možemo definirati na razini sjednice, ali i na razini requesta!

(nije nužno reći na ispitu): valja primijetiti i što se događa s ovom formulom kada imamo nekonzistentno pisanje, npr. $W = 1$ u tom slučaju je $R > 2$, odnosno moramo pročitati sve čvorove da bi imali „konzistentno“ čitanje – ali sada konzistentno samo u smislu da detektiramo konflikt.

7. (5 bodova)



```

PREFIX foaf:      <http://xmlns.com/foaf/0.1/
PREFIX pers:      < http://www.persons.com/allPersons/>
PREFIX relation:  <http://.../>
  
```

```

SELECT ?familyName ?firstName
WHERE {
  ?person relation:father ?father .
  ?person relation:mother ?mother .
  ?father relation:mother ?fatherMother .
  ?mother relation:mother ?motherMother .
  ?fatherMother foaf:firstName ?fatherMotherFName .
  ?motherMother foaf:firstName ?motherMotherFName .
  ?person foaf:firstName ?firstName .
  ?person foaf:familyName ?familyName
  FILTER(?fatherMotherFName = ?motherMotherFName)
}
  
```