

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

Baza podataka je namijenjena pohrani podataka o restoranima na području RH. Za restoran se evidentira naziv, lokacija i prosječna ocjena restorana. Osim restorana, baza sadrži i sva mjesta i sve ceste, odnosno dionice cesta u RH. Npr. A1 autocesta se vodi kao niz dionica (Zagreb-Donja Zdenčina, Donja Zdenčina – Jastrebarsko, Jastrebarsko-Karlovac, Karlovac-Bosiljevo, ...). Restaurant, road i place imaju geoprostorne attribute čiji tipovi se vide iz primjera.

restaurant			
restId	restName	avgGrade	geom
1	Cassandra	3,3	<point>
2	Batelina	4,8	<point>
...

road					
roadId	roadName	placeFrom	placeTo	idRType	Geom
1	A1 Zg-Ka	1	2	1	<polyline>
2	A1 Ka-Bo	2	3	1	<polyline>
...

place			
placeId	placeName	population	geom
1	Zagreb	790017	<point>
2	Karlovac	59395	<point>
3	Bosiljevo	1486	<point>
...

roadType	
rtypeId	rtypeName
1	Highway
2	Motorway
...	...

slika 1

1. (5 bodova) Korištenjem funkcionalnosti PostgreSQL SUBP napisati SQL naredbu kojom će se ispisati nazivi svih mjesta s više od 50 000 stanovnika u koja se može stići autoputom iz Zagreba u sljedećem obliku:

PlaceName	distance
Karlovac	43
Rijeka	148
Zadar	258
...	...

distance uzima u obzir samo duljinu autoputa u kilometrima od Zagreba do konkretnog mjesta.

2. (6 bodova)

- a) Na primjeru relacije **book** objasnite zbog čega standardni indeks (kao što je **contentIdx**) nije pogodan za pretraživanje teksta u RSUBP.

```
CREATE TABLE book (
  bookId INTEGER PRIMARY KEY,
  title VARCHAR (300) NOT NULL,
  content TEXT);
CREATE INDEX contentIdx on
  book (content);
```

bookId	title	content
88	Beauty and the Beast	Once upon a time, there was a girl named...
563	Snow White	Long ago, there lived a girl ...

- b) Uzevši u obzir funkcionalnosti PostgreSQL SUBP predložite izmijenjene sheme relacije **book** tako da nova shema omogući efikasno pretraživanje sadržaja knjiga (**content**) temeljem morfologije, sintakse i semantike engleskog jezika. Nije potrebno pisati SQL naredbe, dovoljno je riječima opisati promjene.

3. Vaš zadatak je napraviti turističku gastronomsku auto-kartu hrvatske.

Ovdje nećemo doista vizualizirati već analizirati kako organizirati i prirediti podatke odgovarajućeg tipa potrebne za vizualizaciju. Rješenja pišite u formi pseudokoda i SQL naredbi, te komentara, pri čemu **navedite GIS funkcije koje planirate koristiti**. Možete smatrati da su vam na raspolaganju svi mogući dodatni podatci i da možete raditi izmjene na bazi ako je potrebno.

(a) **(4 boda)** Potrebno je napraviti vizualizaciju auto-karte tako da se **motorne ceste i autoceste** po dionicama prikažu različitim bojama s obzirom na kvalitetu i brojnost ponude restorana koji pripadaju dionici. Možete smatrati da restoran pripada **najbližoj** motornoj cesti ili autocesti, a da kvalitetu restorana određuje prosječna ocjena.

Na primjer:

- dionica od 10 km koja ima jedan restoran ocjene 4 bi trebala biti bolje ocijenjena od dionice od 15 km kojoj također pripada jedan restoran ocjene 4.
- Isto tako, dionica od 10 km kojoj pripadaju dva restorana ocjene 4 bi trebala biti bolje ocijenjena od dionice od 10 km kojoj pripadaju dva restorana ocjene 2!

(b) **(4 boda)** Kako biste odredili područje s najvećom ponudom restorana po:

- „glavi stanovnika“
- m²

Područje definirajmo kao veće mjesto (npr. $N > 30k$) s pripadajućim manjim mjestima.

Potrebno je opisati potencijalne izmjene na bazi u pogledu dodatnih podataka, te SQL naredbe kojima se pripremaju podatci potrebni za vizualizaciju.

4. **(3 boda)** Koja od svojstava transakcije u relacijskim sustavima je teško ostvariti u distribuiranoj okolini i zašto?

5. **(3 boda)** Objasnite CAP teorem. Navedite primjer u kojem u slučaju mrežne particije nemamo samo binarni izbor.

6. **(3 boda)** Što je *combinable reducer* i koje su mu prednosti i ograničenja. Objasnite na primjeru.

7. **(4 boda)** Kako bi mrežu cesta i gradova modelirali koristeći grafovski model podataka? Na vašem modelu riješite prvi zadatak, tj. napišite Cypher upit.

8. **(3 boda)** Što je ontologija (u kontekstu Semantičkog weba) i čemu služi?
Od čega se sastoji (što je moguće opisati pomoću OWL-a)?

Rješenja:

1.

```
WITH RECURSIVE roadRoutes (placeFrom, placeTo, totalM)
AS
    ( (SELECT placeFrom,
              placeTo,
              st_length(road.geom)
        FROM road
        JOIN place ON road.placeFrom = place.placeId
        WHERE place.placeName = 'Zagreb'
          AND road.rdTypeId = 1
        )
      UNION ALL
      (SELECT roadChild.placeFrom,
              roadChild.placeTo,
              roadParent.totalM + ST_length (roadChild.geom)
        FROM roadRoutes roadParent
        JOIN road roadChild ON roadParent.placeTo = roadChild.placeFrom
        WHERE roadChild.rdTypeId = 1)
    )
SELECT place.placeName,
       roadRoutes.totalM/1000
FROM roadRoutes
JOIN place ON place.placeId = roadRoutes.placeTo
WHERE place.population >50000
```

ili

```
WITH RECURSIVE roadRoute (placeFrom, placeTo, geom)
AS
    ( (SELECT placeFrom,
              placeTo,
              geom,
        FROM road
        JOIN place ON road.placeFrom = place.placeId
        WHERE place.placeName = 'Zagreb'
          AND road.rdTypeId = 1
        )
      UNION ALL
      (SELECT roadChild.placeFrom,
              roadChild.placeTo,
              st_union(roadParent.geom, roadChild.geom)
              --ovdje zapravo treba obaviti CAST na jednak tip podatka kao što je road.geom
              --uz pretpostavku da je road.geom tipa GEOMETRY(MultiLineString,4326)
              --ispravno rješenje je
              --CAST(st_union(roadParent.geom, roadChild.geom) AS GEOMETRY(MultiLineString,4326))
        FROM roadRoute roadParent
        JOIN road roadChild ON roadParent.placeTo = roadChild.placeFrom
        WHERE roadChild.rdTypeId = 1)
    )
SELECT place.placeName,
       st_length(roadRoute.geom)
FROM roadRoute
JOIN place ON place.placeId = roadRoute.placeTo
WHERE place.population >50000
```

2.

- a) Zbog algoritma pretraživanja B-stabla, pomoću indeksa `bookContentIdx`, moguće je obavljati samo upite koji traže zapise s točno jednakim sadržajem atributa `content` ili zapise koji imaju na početku atributa traženi uzorak. Npr.:

```
SELECT * FROM book WHERE content LIKE 'Once%';
```

koristi indeks

```
SELECT * FROM book WHERE content LIKE '%Once %';
```

NE koristi indeks

Takvi upiti nisu realni pa ni indeks ne pomaže.

Pomoću ovakvog indeksa nije moguće obaviti pretraživanje koje bi vodilo računa o:

- stop riječima,
- različitim pojavnim oblicima riječi s istim korijenom/osnovom,
- različitim riječima jednakog značenja,

kao niti pretraživanje koje bi primjenjivalo neku od tehnika približne pretrage teksta kao što je npr. pretraga pomoću Q-gram algoritama.

b)

1. dodati u relaciju `book` novi atribut **`content_tsv`** tipa `TSVECTOR` koji bi čuvao **`content`** u normaliziranom obliku
2. kreirati trigger za `INSERT` i `UPDATE` on `book` koji bi održavao ažurnim vrijednost tog atributa koja se može se dobiti pomoću funkcije **`to_tsvector('english', content)`**
3. Kreirati invertirani indeks nad atributom **`content_tsv`**:
`CREATE INDEX contentidx2 ON book USING gin(bookContent_tsv)`

3.

(a)

Potrebno je restoranu pridijeliti id dionice kojoj pripada na temelju udaljenosti koju računamo s `st_distance()`, npr.:

```
ALTER TABLE restaurant ADD idRoad int ;
```

```
UPDATE restaurant
```

```
SET idRoad = (SELECT idRoad
              FROM Road r
              WHERE st_distance(restaurant.geom, r.geom) =
                (
                  SELECT min(st_distance(resautarant.geom, allRoads.geom))
                  FROM road allRoads
                  LIMIT 1
                )
            )
```

Može i preko `LIMIT`:

```
UPDATE restaurant
```

```
SET idRoad = (SELECT idRoad
              FROM Road r
              ORDER BY st_distance(restaurant.geom, r.geom) ASC
              LIMIT 1
            )
```

Zatim:

```
ALTER TABLE road ADD restDensity decimal(5,2);
```

```
UPDATE road
```

```
SET restDensity = 1./st_legth(geom) * (SELECT AVG(avgGrade)
                                       FROM restaurant
                                       where idRoad = road.idRoad)
```

I sve slične mjere bi ja uvažio, koje zadovoljavaju zahtjeve.

Npr.: `max(grade)` + ovo gore

(b)

Odabrao bih mjesta >30k, te napravio Voronoi dijagram kao na projektu.

(uglavnom se priznaju i rješenja s `st_distance`, pa `convexhull`, iako to nije skroz točno).

Potom bi to pohranio u bazu kao geometriju, jedan zapis po jednom >30k mjestu u tablicu područja.

Zatim bi u:

(b1) dijelu uzeo:

```
(COUNT(*) FROM restaurant where st_contains(podrucje.geom, rest.geom))  
/( SUM(population) FROM place where st_contains(podrucje.geom, place.geom))
```

a u (b2):

```
(COUNT(*) FROM restaurant where st_contains (podrucje.geom, rest.geom))  
/( SUM(st_area(podrucje.geom))
```

Teoretska rješenja nisu detaljno raspisana, pogledati više u predavanjima:

4. **(3 boda)** Koja od svojstava transakcije u relacijskim sustavima je teško ostvariti u distribuiranoj okolini i zašto?

Od ACID je teško je ostvariti A i I, zato jer se koriste distrib transakcije, odnosno u ostvarivanju tih svojstava moraju sudjelovati SVI čvorovi na način da međusobno ovise jedan od drugome.

Napomena većini: C iz ACID nije C iz CAP!!!

5. **(3 boda)** Objasnite CAP teorem. Navedite primjer u kojem u slučaju mrežne particije nemamo samo binarni izbor.

Vidjeti predavanja. Tamo je i primjer rezervacije hotelske sobe gdje sustav može biti u polu-funkcionirajućem stanju (1/2CA) na način da jedan server nastavi funkcionirati usred particije.

6. **(3 boda)** Što je *combinable reducer* i koje su mu prednosti i ograničenja. Objasnite na primjeru.

Vidjeti predavanja i projekte. CR se uvodi kako **ne bi svi čvorovi čekali najsporijeg prije ulaska u reduce fazu (dakle, prednost)**. Uslijed toga reduce f-je moraju imati isti ulaz i izlaz. To je ograničenje i nedostatak, koji onda uzrokuje i da se ne mogu riješiti neke klase problema, recimo SELECT DISTINCT ili SORT, ali to bi trebali znati (!) – to smo radili na projektu i komentirali kod predaje!

7. **(4 boda)** Kako bi mrežu cesta i gradova modelirali koristeći grafovski model podataka? Na vašem modelu riješite prvi zadatak, tj. napišite Cypher upit.

Mjesta su čvorovi sa svojstvima naziv, population i sl. Ceste se modeliraju kao relationships između mjesta, i mogu biti tipa Highway, motorway, ...

Cypher otprilike:

```
MATCH (zg: Place{name:"Zg"})-[r:Highway*]-(odrediste:Place{population>50000})  
RETURN odrediste, SUM(r.length)
```

8. **(3 boda)** Što je ontologija (u kontekstu Semantičkog weba) i čemu služi?

Od čega se sastoji (što je moguće opisati pomoću) OWL-a?

Vidjeti predavanja.

Ontologija je model podataka koji predstavlja skup pojmova unutar neke domene i veze između tih pojmova

Eksplisitna, formalna specifikacija zajedničke konceptualizacije. (Thomas R. Gruber, 1993)

Eksplisitna: nedvosmislene definicije, svi koncepti moraju biti definirani

Formalna specifikacija: "razumljiva računalima"

Zajednička: dijeljena, poznatog i općeprihvaćenog značenja

Konceptualizacija: apstraktni model (domena, koncepti, odnosi među konceptima)

Ovo se tražilo za OWL (što može napraviti):

Sastoji se od:

- a. klasa (i njihove hijerarhije)
- b. svojstva koja te klase imaju
- c. ograničenja i odnosa među svojstvima (tip, domena, kardinalnost, jednakost)
- d. instanca klasa