

1. (6 bodova) Prikazani segment modela podataka trgovački lanac koristi za praćenje prodaje proizvoda po prodavaonicama. Pretpostavka je da trgovački lanac trenutno ima 99 prodavaonica.

proizvod		prodaja						
sifProizv	nazivProizv	...	datum	sifProd	sifProizv	kolicina	iznos	...
15	Clarins maskara		15/01/2015	1	15	1	200	
25	Chanel smokey eyes sjenilo		15/01/2015	1	15	2	400	
35	Shiseido BB krema		16/01/2015	1	35	2	300	
...	...		16/01/2015	1	35	2	300	
			15/01/2015	2	35	2	300	
			15/01/2015	2	15	1	200	
			16/01/2015	2	25	2	200	
			16/01/2015	2	25	3	300	
			

prodavaonica		
sifProd	nazivProd	...
1	Store 1	
2	Store 2	
...	...	
854	Store 99	

Korištenjem mogućnosti PostgreSQL sustava baza podataka i ugrađenih funkcija za pivotiranje u tom sustavu, napišite SQL upit pomoću kojeg će se ispisati ukupan prihod po proizvodima i poslovnica u sljedećem obliku:

nazivProizv	Store1	Store2	...	Store99
Chanel smokey eyes sjenilo		500.00
Clarins maskara	600.00	200.00
Shiseido BB krema	600.00	300.00
...

Bi li napisani (Vaš) upit vraćao ispravne podatke u slučaju:

- da se pojavi jedan (ili više) novih proizvoda
- da se otvori još jedna (ili više) prodavaonica

Ako upit u nekom od dva gornja slučaja ne bi vraćao ispravne podatke, opišite što bi trebalo promijeniti da vraća ispravne podatke.

2. U zadatku se pretpostavlja korištenje PostgreSQL SUBP-a u kojem postoji intarray proširenje.

- a) (2 boda) Napišite SQL naredbu za kreiranje tablice **prodavaonicaOR** i svih ostalih objekata korištenih u definiciji te tablice. Tablica, osim šifre (primarni ključ) i naziva prodavaonice, sadrži i attribute:
- *poslovodja* - atribut tipa *poslovodjaT* čiji su atributi (elementi): *sifOsoba*, *datumOd*
 - *zaposlenici* – šifre zaposlenika u prodavaonici. Zaposlenici su poredani prema trenutku zapošljavanja u prodavaonici (kao prvi je naveden zaposlenik koji je prvi zaposlen).
- b) (1 bod) Napišite upit kojim će se ispisati šifra i naziv prodavaonica s više od jednog zaposlenika, u kojima je poslovođa zadnja zaposlena osoba u toj prodavaonici.
- c) (2 boda) Napišite upit kojim će se provjeriti postoje li osobe koje su zaposlene u više prodavaonica. Za takve osobe potrebno je ispisati šifru osobe i polje koje sadrži šifre prodavaonica u kojima je osoba zaposlena. Zadatak riješiti bez korištenja podupita.

3. (3 boda) Objasnite što podrazumijeva kvalitetna priprema tekstualnih sadržaja za kasnije efikasno pretraživanje cijelog teksta (Full Text Search). Na primjeru PostgreSQL SUBP, objasnite ulogu različitih rječnika pri pretraživanju cijelog teksta.

4. (2 boda) U kontekstu upravljanja vremenom, objasnite razliku između stanja i događaja. Navedite primjer relacije stanja i relacije događaja.

5. (3 boda) Što je Bloomov filter? Koja su njegova svojstva? Navedite primjer s barem tri člana skupa.

6. (8 bodova) Ulazna datoteka sadrži podatke o mjerenjima temperature i tlaka u 12:00 sati u svim mjestima u Hrvatskoj (za veći niz godina).

Napišite **M/R algoritam** koji će za sva mjesta vratiti prosječne mjesečne vrijednosti temperature i tlaka **po mjesecima 2014.** godine.

Nacrtajte čvorove, tijek podataka i **izlaze** iz map i reduce faza, **te konačan rezultat** za ove testne ulazne podatke:

```
(Zg, 31.12.2013, 1005, -3)
(Zg, 01.01.2014, 1010, -2)
(Zg, 02.01.2014, 1000, -4)
(Pu, 31.12.2013, 1000, +7)
(Pu, 01.01.2014, 1002, +5)
(Pu, 02.01.2014, 1004, +6)
```

i pritom pretpostavite da se izračunavanje obavlja na **tri čvora**. Pretpostaviti općenitu paradigmu s više reducera (tj. u ovom primjeru - tri reducera).

Kod „implementacije“ možete koristiti bilo koji programski jezik ili pseudo kod. Pritom, ako koristite pseudo kod, ne smijete prelaziti granice mogućnosti modernih programskih jezika, odnosno smijete koristiti različite strukture podataka (npr. hashmap) i funkcije (npr. sort, splitWords), ali ne smijete koristiti izmišljene napredne funkcije primjenjive upravo na ovaj problem (npr. rijesiZadatak3()).

7. Nadopunite sljedeće SPARQL upite nad semantičkim izvorom DBpedia:

Pretpostavite upotrebu sljedećih prefiksa

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>

PREFIX dbpprop: <http://dbpedia.org/property/>

a) (1 bod) Izračunajte ukupnu zaradu svih filmova u kojima je glumio George Clooney. Uzmite u obzir samo one filmove koji su pojedinačno zaradili preko 100 milijuna dolara.

```
WHERE {
  ?film a dbpedia-owl:Film .
  ?film dbpedia-owl:starring <http://dbpedia.org/resource/George_Clooney> .
  ?film dbpedia-owl:gross ?zarada .
}
```

b) (2 boda) Napišite upit koji će dati odgovor na pitanje:

Da li su George Clooney (<http://dbpedia.org/resource/George_Clooney>) i Brad Pitt

(<http://dbpedia.org/resource/Brad_Pitt>) glumili (dbpedia-owl:starring) u još nekom filmu osim u onima čiji naziv počinje sa „Ocean“? Rezultat upita mora biti true ili false.

```
WHERE {
  ?film a dbpedia-owl:Film .
  ?film dbpprop:name ?naziv .
}
```

Rješenja:

```
1) SELECT * FROM crosstab (
    'SELECT CAST(nazivProizv AS char(50)) AS nazivProizv
      , CAST(sifProd AS int) AS sifProd
      , CAST(SUM(iznos) AS DECIMAL(10,2)) AS prihod
    FROM prodaja, proizvod
    WHERE prodaja.sifProizv = proizvod.sifProizv
    GROUP BY prodaja.sifProd, nazivProizv
    ORDER BY nazivProizv, sifProd'
    , 'SELECT sifProd FROM prodavaonica ORDER BY sifProd'
)
AS pivotTable (nazivProizv CHAR(50),
               Store1 DECIMAL(10,2),
               Store2 DECIMAL(10,2),
               ...
               Store99 DECIMAL(10,2))
ORDER BY nazivProizv
```

Dodavanjem novih proizvoda upit ispravno radi i nije ga potrebno mijenjati.

Dodavanjem nove prodavaonice upit neće dobro raditi. Sada upit koji su argumenti crosstab funkciji vraćaju podatke za jednu dodatnu prodavaonicu:

upit koji je prvi argument funkcije crosstab vraća podatke za jednu prodavaonicu više

'SELECT sifProd FROM prodavaonica ORDER BY sifProd' vraća jednu n-torku više

i više nisu u skladu sa shemom pivotTable koja očekuje podatke za 99 prodavaonica. Potrebno je promijeniti shemu pivotTable na:

```
AS pivotTable (nazivProizv CHAR(50),
               CosmeticsHygieneNo1 DECIMAL(10,2),
               ...
               CosmeticsHygieneNo99 DECIMAL(10,2),
               CosmeticsHygieneNo100 DECIMAL(10,2))
```

Od studenata se ne očekuje objašnjenje koje slijedi u nastavku. Služi kao pomoć za bolje razumijevanje rješenja studentima koji ispite koriste pri uvježbavanju gradiva.

Dodavanjem nove prodavaonice pri izvođenju upita doći će do sljedeće pogreške:

ERROR: invalid return type

DETAIL: Query-specified return tuple has 100 columns but crosstab returns 101.

Query-specified return tuple has 100 columns – 100 je broj atributa specificiranih u shemi pivotTable: nazivProizv + CosmeticsHygieneNo1 – CosmeticsHygieneNo99

crosstab returns 101 – upiti koji su argumenti crosstab funkciji vraćaju podatke za 100 umjesto za 99 prodavaonica pa i sama crosstab funkcija vraća 101 atribut (naziv proizvoda + prihod od konkretnog proizvoda za 100 prodavaonicama).

```
2.a) CREATE TYPE poslovodjaT AS (sifOsoba INT, datumOd DATE);

CREATE TABLE prodavaonicaOR (sifProd INTEGER PRIMARY KEY,
                              nazProd VARCHAR(60),
                              poslovodja poslovodjaT,
                              zaposlenici INTEGER[]);

b) SELECT sifProd, nazProd FROM prodavaonica
   WHERE zaposlenici[icount (zaposlenici)] = (poslovodja).sifOsoba
   AND icount (zaposlenici) >1;

c) SELECT UNNEST(zaposlenici) AS sifOsoba, array_agg(sifProd)
   FROM prodavaonicaOR
   GROUP BY sifOsoba
   HAVING COUNT(*) > 1
```

3. (4 boda)

Efikasna pretraga cijelog teksta podrazumijeva prethodnu obradu teksta (dokumenta):

- parsiranje teksta i rastavljanje na tokene (riječi, brojevi, tagovi, razmak, url,...)
- konvertiranje tokena u lekseme – za to se koristi algoritam prilagođen konkretnom jeziku koji uvažava pravila jezika.
- pohrana obrađenog teksta optimiranog za pretragu
- postojanje indeksa prikladnih za pretragu cijelog teksta (npr. invertirani indeks)

Rječnici trebaju omogućiti:

- definiranje stop riječi, koje ne treba indeksirati
- definiranje sinonima
- stvaranje veza između fraza i pojedinih riječi

PostgreSQL ima nekoliko tipova rječnika (**naravno da studenti ne moraju navesti kako se koji rječnik zove – treba samo navesti što u globalu kolekcija rječnika radi**):

- Simple Dictionary Uklanja stop riječi i velika slova svodi na mala.
- Synonym Dictionary Različite riječi jednakog značenja zamjenjuje reprezentantnom riječju.
- Thesaurus Dictionary Omogućuje prepoznavanje fraza.
- iSpell Dictionary Svodi riječi na normalizirani oblik.
- Snowball Dictionary Algoritamski svodi riječi na korijenski oblik (stemming) i uklanja stop riječi

4. Stanja opisuju činjenice vezane uz neki objekt u bazi podataka koje su istinite u nekom vremenskom intervalu ili periodu. Te se činjenice ne smatraju točnima izvan pridruženog perioda.

Događaji opisuju činjenice vezane uz neki objekt u bazi podataka koje su se dogodile u određenom trenutku (chrononu) i nemaju trajanje.

stanje

placa

idZaposlenika	iznos	vrijediOd	vrijediDo
100	5000	1.1.2010.	NULL
101	8000	6.12.2001.	1.1.2005
101	10000	2.1.2005.	NULL
102	6000	1.7.2009.	1.7.2011.
105	6000	20.5.1998.	19.5.2008.
105	7000	20.5.2008.	NULL

događaj

nagrada

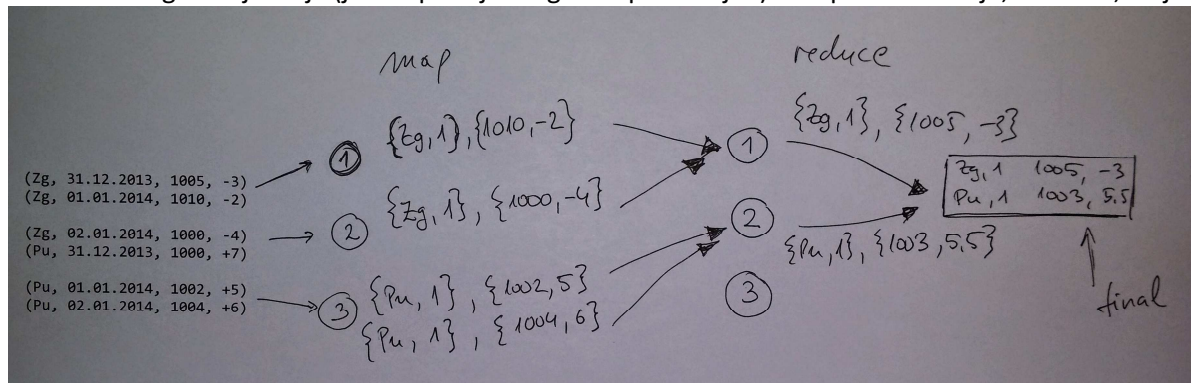
idZaposlenika	iznos	datum
100	5000	10.6.2011.
105	5000	24.12.2001.
101	10000	1.7.2011.
105	5000	20.5.2008.

6.

```
map (k, v) {
  foreach (item in v) {
    if (year(v[1]) == 2014) {
      key = {
        grad: v[0],
        mjesec: month(v[0])
      }
      val = {
        tlak: v[2],
        temp: v[3]
      }
      emit (key, val);
    }
  }
}

reduce (k, v) {
  foreach (pair in v) {
    sumP += pair.tlak;
    sumT += pair.temp;
  }
  val = {
    avgTlak: sumP/v.length,
    avgTemp: sumT/v.length,
  }
  return (k, val)
}
```

Jedno od mogućih rješenja (jer mapiranja mogu biti proizvoljna). Ukupan rezultat je, naravno, uvijek isti:



7. a) PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
 PREFIX dbpprop: <http://dbpedia.org/property/>
 SELECT SUM(xsd:integer(?zarada))
 WHERE {
 ?film a dbpedia-owl:Film .
 ?film dbpedia-owl:starring <http://dbpedia.org/resource/George_Clooney> .
 ?film dbpedia-owl:gross ?zarada
 FILTER (xsd:integer(?zarada) > 100000000)
 }

Cast operator (xsd:integer) je nužan za funkcioniranje upita na Dbpedia krajnjoj točki (potrebno je pretvoriti tip podataka US\$ u cijeli broj), no u ispitu se priznaje i rješenje bez cast operatora.

b) PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
 PREFIX dbpprop: <http://dbpedia.org/property/>
 ASK
 WHERE {
 ?film a dbpedia-owl:Film .
 ?film dbpedia-owl:starring <http://dbpedia.org/resource/George_Clooney> .
 ?film dbpedia-owl:starring <http://dbpedia.org/resource/Brad_Pitt> .
 ?film dbpprop:name ?naziv .
 FILTER (!REGEX(?naziv, "Ocean"))
 }

1. Zadatak

```
<http://dbpedia.org/resource/George_Clooney> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
.
<http://dbpedia.org/resource/George_Clooney> <http://dbpedia.org/ontology/starring> <http://dbpedia.org/resource/Oceans_Eleven> .
<http://dbpedia.org/resource/George_Clooney> <http://xmlns.com/foaf/0.1/knows> <http://dbpedia.org/resource/Brad_Pitt> .
<http://dbpedia.org/resource/Brad_Pitt> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://dbpedia.org/resource/Brad_Pitt> <http://dbpedia.org/ontology/starring> <http://dbpedia.org/resource/Oceans_Eleven> .
<http://dbpedia.org/resource/Brad_Pitt> <http://xmlns.com/foaf/0.1/knows> <http://dbpedia.org/resource/George_Clooney> .
<http://dbpedia.org/resource/Oceans_Eleven> <http://dbpedia.org/property/name> "Ocen's Eleven" .
```

Ili skraćeno:

```
<http://dbpedia.org/resource/Brad_Pitt> rdf:type foaf:Person ;  
  Dbpedia-owl:starring <http://dbpedia.org/resource/Oceans_Eleven> ;  
  foaf:knows <http://dbpedia.org/resource/George_Clooney> .  
  
<http://dbpedia.org/resource/George_Clooney> a foaf:Person ;  
  Dbpedia-owl:starring <http://dbpedia.org/resource/Oceans_Eleven> ;  
  foaf:knows <http://dbpedia.org/resource/Brad_Pitt> .  
  
<http://dbpedia.org/resource/Oceans_Eleven> dbpprop:name "Ocen's Eleven" .
```