

Prikazani segment modela podataka trgovački lanac koristi za praćenje prodaje proizvoda po prodavaonicama. Korištenjem PostgreSQL sintakse i modela s donje slike riješiti zadatke 1. i 2.

proizvod

sifProizv	nazivProizv
15	Clarins maskara
25	Chanel smokey eyes sjenilo
35	Shiseido BB krema
...	...

prodavaonica

sifProd	nazivProd	...
1	Cosmetics & Hygiene – No 1	
2	Cosmetics & Hygiene – No 2	
...	...	

prodaja

...	datum	sifProd	sifProizv	kolicina	iznos	...
	15/01/2015	1	15	1	200	
	15/01/2015	1	15	2	400	
	15/01/2015	1	25	1	100	
	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	
...

1. (6 bodova) Ispisati podatke o prodaji u sljedećem obliku:

nazivProd	nazivProizv	ukPrihod ProizvProd	prosPrihod Prod	Rang ProizvProd
Cosmetics & Hygiene – No 1	Shiseido BB krema	600.00	433.33	1
Cosmetics & Hygiene – No 1	Clarins maskara	600.00	433.33	1
Cosmetics & Hygiene – No 1	Chanel smokey eyes sjenilo	100.00	433.33	3
Cosmetics & Hygiene – No 2	Chanel smokey eyes sjenilo	500.00	333.33	1
Cosmetics & Hygiene – No 2	Shiseido BB krema	300.00	333.33	2
Cosmetics & Hygiene – No 2	Clarins maskara	200.00	333.33	3
...

ukPrihodProizvProd
prosPrihodProd

ukupan iznos zarađen prodajom nekog proizvoda u nekoj prodavaonici
prosječan prihod po prodavaonici - zarađen prodajom svih proizvoda u
konkretnoj prodavaonici (npr. za Cosmetics & Hygiene – No 1:
 $(600.00+600.00+100.00)/3= 433.33$)

rangProizvProd

proizvodi unutar prodavaonice rangiraju se temeljem ukupnog prihoda od
konkretnog proizvoda ostvarenog u toj prodavaonici

2. (3 boda) Ispisati podatke o kumulativnom prihodu po prodavaonicama i danima. Prihod sljedećeg dana obuhvaća i prihode svih prethodnih dana.

nazivProd	datum	kumPrihod DoDatum
Cosmetics & Hygiene – No 1	15/01/2015	700.00
Cosmetics & Hygiene – No 1	16/01/2015	1300.00
Cosmetics & Hygiene – No 2	15/01/2015	500.00
Cosmetics & Hygiene – No 2	16/01/2015	1000.00
...

3. (4 boda) Komentirajte potporu za transakcije u NoSQL sustavima.

Kakva je povezanost modela podataka s (ne)mogućnošću obavljanja transakcija u NoSQL sustavima?
Objasnite pomoću primjera.

4. (4 boda) Definirajte pojmove aditivna i semiaditivna mjera. Na primjeru komentirajte svojstvo semiaditivnosti mjere.

5. (10 bodova) Za svaki par članova društvene mreže potrebno je izračunati koliko imaju zajedničkih prijatelja, tako da je korisnika moguće obavijestiti npr.: „Vi i Pero imate 17 zajedničkih prijatelja“. Neka su podatci za svakog člana pohranjeni kao (član, lista_prijatelja) što ujedno predstavlja ulazne podatke u Map/Reduce algoritam kojim je potrebno riješiti ovaj zadatak. Napišite **M/R algoritam** kojim se za svaki par članova izračunava broj zajedničkih prijatelja. Napišite **izlaze** iz map i reduce faza, **te konačan rezultat** za ove testne ulazne podatke (članovi mreže su predstavljeni varijablama A, B, C i D):

- (A, BCD)
- (B, AD)
- (C, A)
- (D, AB)

i pritom pretpostavite da se izračunavanje obavlja na **tri čvora**.

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()).

6. (3 boda) Objasnite „Open World“ i „Unique Name“ pretpostavke. Vrijede li one u OWL-u?
7. (5 bodova) Na temelju zadanih RDF/XML podataka, nacrtajte RDF graf:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:p="http://hgu.hr/ontology/porin/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://porin.hr/2014/pjesma_godine">
    <dc:title>Suze nam stale na put</dc:title>
    <p:izvodjac>
      <rdf:Description rdf:about="http://porin.hr/izvodjac/Massimo">
        <foaf:givenName>Massimo</foaf:givenName>
        <foaf:surname>Savić</foaf:surname>
        <owl:sameAs rdf:resource="http://dbpedia.org/resource/Massimo_Savić"/>
      </rdf:Description>
    </p:izvodjac>
  </rdf:Description>
  <rdf:Description rdf:about="http://porin.hr/209/pjesma_godine">
    <dc:title>Zar više nema nas</dc:title>
    <p:izvodjac>
      <rdf:Description rdf:about="http://porin.hr/izvodjac/Massimo">
        <foaf:name>Massimo Savić</foaf:name>
        <foaf:knows rdf:resource="http://porin.hr/izvodjac/Neno_Belan"/>
      </rdf:Description>
    </p:izvodjac>
    <p:izvodjac>
      <rdf:Description rdf:about="http://porin.hr/izvodjac/Neno_Belan">
        <foaf:knows rdf:resource="http://porin.hr/izvodjac/Massimo"/>
      </rdf:Description>
    </p:izvodjac>
  </rdf:Description>
  <rdf:Description rdf:about="http://porin.hr/2011/hit_godine">
    <dc:title>Bižuterija</dc:title>
    <p:izvodjac>
      <rdf:Description rdf:about="http://porin.hr/izvodjac/Jelena_Rozga">
        <foaf:knows rdf:resource="http://porin.hr/izvodjac/Massimo"/>
        <foaf:knows rdf:resource="http://porin.hr/izvodjac/Neno_Belan"/>
      </rdf:Description>
    </p:izvodjac>
  </rdf:Description>
</rdf:RDF>
```

Rješenja:

1. (6 bodova)

```
select nazivProd, nazivProizv
      , SUM(iznos)
      , AVG(SUM(iznos)) OVER (PARTITION BY prodaja.sifProd) prosjPrihodProd
      , rank() OVER (PARTITION BY prodaja.sifProd ORDER BY SUM(iznos) DESC) rangProizvProd
FROM prodaja
JOIN proizvod
  ON prodaja.sifProizv = proizvod.sifProizv
JOIN prodavaonica
  ON prodaja.sifProd = prodavaonica.sifProd

GROUP BY prodaja.sifProizv, nazivProd, prodaja.sifProd, nazivProizv
ORDER BY nazivProd, rangProizvProd, nazivProizv desc
```

2. (4 bodova)

```
select DISTINCT nazivProd, datum
      , SUM(iznos) OVER (PARTITION BY prodaja.sifProd ORDER BY datum)
FROM prodaja
JOIN prodavaonica
  ON prodaja.sifProd = prodavaonica.sifProd
ORDER BY nazivProd, datum
```

5. (10 bodova)

Prvo komentirajmo najčešća **pogrešna** rješenja:

Mnogi su studenti u map funkciji samo prosljeđivali podatke dalje u reduce, ili još gore – razlomili ih na manje dijelove bez ikakve dodatne obrade, čime su samo povećali mrežni promet (npr. za A->BCD emitiraju A->B, A->C, A->D).

Dva tipična nastavka na ovo su:

- (a) Studenti su pretpostavili da postoji samo jedan reducer i sve, više ili manje „uspješno“, napravili u reduceru. Ovdje je važno podsjetiti da je M/R namijenjen **paralelnom** obavljanju, a ako samo raspršimo podatke kako bi ih kasnije skupili i sve obavili u jednom jedinom čvoru onda smo **potpuno promašili smisao M/R-a** i zapravo sve obavili **na neučinkovitiji način**.
- (b) Postoji više reduceru, ali budući da zapisi nisu dobro raspršeni, ne zna se koji će gdje završiti i ne može se točno odrediti presjek.

Ključno u rješavanju ovog zadatka je bilo dosjetiti se da map funkcija raspršiti (emitira) podatke po **abecedno poredanim ključevima parova**.

Npr. za A->BCD, treba emitirati **AB->CD**, AC->BD i AD->BC.

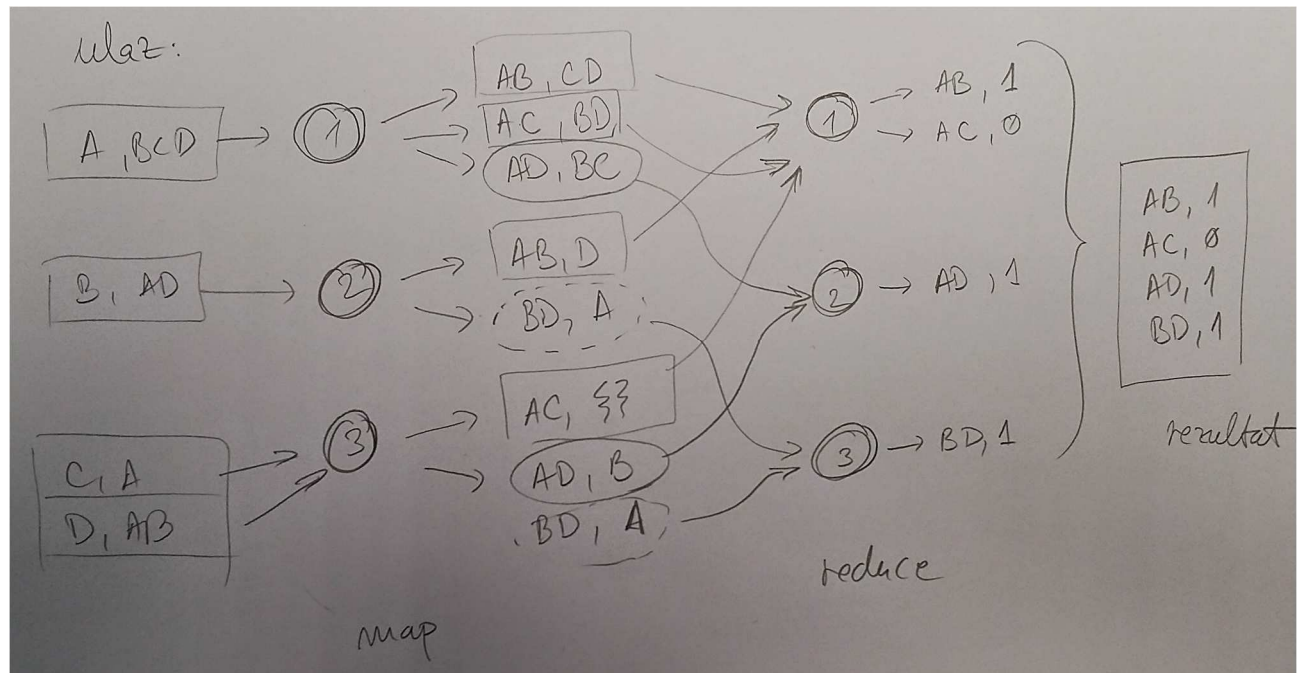
Na nekom drugom, ili istom čvoru, B će emitirati svoje parove i, ako je npr. A među njegovim prijateljima, svakako će emitirati **AB->ostali_B_prijatelji**.

M/R nam nakon toga **garantira** da će svi AB->* završiti **u istom reducer čvoru**, koji onda samo treba napraviti presjek i odrediti njegovu kardinalnosti.

```
map (k, v) {
    foreach (item in v) {
        list = k, item;
        list.sort();
        newkey = list.concat();
        newlist = v.minus(item);
        emit (newkey, newlist);
    }
}

reduce (k, v) {
    // v lista koja uvijek sadrži dvije liste
    mutual = v[0].intersect(v[1]);
    return (k, mutual.size());
}
```

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



7.

