

Napredne baze podataka

Vježbe

Domaća zadaća br. 4 – rješenja i komentari

U ovom dokumentu...

- Zadaci koje je trebalo proći u 4. domaćoj zadaći i moguća rješenja
- Komentari na neka od vaših rješenja i usputne opaske
- Koristimo MongoDB
 - Docker container (prema uputama za vježbe)
 - Isti setup kao iza drugu domaću zadaću

Priprema baze podataka

- Prema uputama, trebalo je ponovno inicijalizirati isti kontejner koji se koristio za drugu domaću zadaću:
 - Ili uništiti cijeli kontejner, pa stvoriti novi i ponovno napuniti podatke iz pripremljene datoteke
 - Ili samo uništiti bazu nbp ili samo kolekciju orders i ponovno napuniti podatke iz pripremljene datoteke

Zadatak 1

- Korištenjem M/R frameworka izračunati prosječnu cijenu svakog prodanog proizvoda za sve narudžbe nakon 1.1.1998. koje su naručene iz Francuske. Pohraniti rezultate u kolekciju france98. Polje u koje se sprema prosječna cijena se treba zvati mr_avg. Ispisati prvih 10 zapisa iz kolekcije france98.

Zadatak je vrlo sličan onom s predavanja.

Uvjeti na datum i državu se zadaju u samom pozivu map/reduce funkcije.

S obzirom na to da u pojedinoj narudžbi može biti kupljena veća količina istog proizvoda po jednoj cijeni (npr. 10 komada po cijeni od 5), a u drugoj narudžbi druga količina tog proizvoda po drugoj cijeni (npr. 20 komada po cijeni od 6), da bismo izračunali ukupan prosjek (što je u ovom slučaju $(10 \cdot 5 + 20 \cdot 6) / 30$), trebamo posebno zbrajati ukupne cijene plaćene za proizvod u jednoj narudžbi i broj kupljenih proizvoda u toj narudžbi.

(popuste smo odlučili ignorirati zbog jednostavnosti, ali nije greška ako ste ih uzeli u obzir)

Zadatak 1. - mapper

```
var map1 = function() {  
  for (var i = 0; i < this.details.length; i++) {  
    var key = this.details[i].product.name;  
  
    var value = {  
      count: this.details[i].quantity,  
      price: this.details[i].unit_price * this.details[i].quantity  
    };  
    emit(key, value);  
  }  
};
```

Koristimo product_name kao ključ.

Izlazni value je dokument koji ima dva polja, *count* i *price*, što je važno u reduceru!

Emitiramo jedan ključ-vrijednost za svaki kupljeni proizvod u jednom dokumentu (narudžbi).

Zadatak 1. - reducer

```
var reduce1 = function(keyProdName, countValues) {  
    reducedVal = { count: 0, price: 0 };  
  
    for (var idx = 0; idx < countValues.length; idx++) {  
        reducedVal.count += countValues[idx].count;  
        reducedVal.price += countValues[idx].price;  
    }  
    return reducedVal;  
};
```

Ulazni dokument koji ima dva polja, *count* i *price*, na što valja računati.

U reduceru samo zbrajamo ukupne plaćene cijene za pojedini ključ (proizvod) i količine.

Zbog toga što se reducer može pozivati više puta, a jedna od vrijednosti u polju je izlazni dokument iz ranijeg poziva reducera, izlazni dokument mora odgovarati ulaznom, tj. mora imati ista polja i istu semantiku (vidjeti svojstva reducer funkcije u predavanjima).

Neki od vas su ovdje griješili i iz reducera vraćali drugačiji dokument ili vrijednost.

Zadatak 1. - finalize

```
var finalize1 = function (key, reducedVal) {  
    reducedVal.mr_avg = reducedVal.price/reducedVal.count;  
    return reducedVal;  
};
```

Finalize se poziva samo jednom za svaki ključ.

U njemu možemo u izlazni dokument dodati i prosječnu cijenu proizvoda, tako da podijelimo zbroj svih cijena i zbroj svih količina.

Zadatak 1. - upit

```
db.orders.mapReduce(  
  map1,  
  reduce1,  
  {  
    out: { replace: "france98" },  
    query: { "order_date" : {$gte: "1998-01-01"} ,  
            "customer.country":"France"},  
    finalize: finalize1  
  }  
);
```



U konačnici, vidimo da kolekcija france98 sadrži 39 dokumenata, odnosno, informacije za 39 proizvoda koji su naručivani iz Francuske u 1998. godini.

```
{  
  "result" : "france98",  
  "timeMillis" : 199,  
  "counts" : {  
    "input" : 23,  
    "emit" : 56,  
    "reduce" : 14,  
    "output" : 39  
  },  
  "ok" : 1  
}
```


Zadatak 1. – kako izgleda kolekcija france98

```
> db.france98.find().limit(10)
```

```
{ "_id" : "Alice Mutton", "value" : { "count" : 37, "price" : 1443, "mr_avg" : 39 } }  
{ "_id" : "Carnarvon Tigers", "value" : { "count" : 6, "price" : 375, "mr_avg" : 62.5 } }  
{ "_id" : "Chartreuse verte", "value" : { "count" : 21, "price" : 378, "mr_avg" : 18 } }  
{ "_id" : "Chocolade", "value" : { "count" : 8, "price" : 102, "mr_avg" : 12.75 } }  
{ "_id" : "Côte de Blaye", "value" : { "count" : 5, "price" : 1317.5, "mr_avg" : 263.5 } }  
{ "_id" : "Filo Mix", "value" : { "count" : 5, "price" : 35, "mr_avg" : 7 } }  
{ "_id" : "Geitost", "value" : { "count" : 11, "price" : 27.5, "mr_avg" : 2.5 } }  
{ "_id" : "Gorgonzola Telino", "value" : { "count" : 35, "price" : 437.5, "mr_avg" : 12.5 } }  
{ "_id" : "Grandma's Boysenberry Spread", "value" : { "count" : 70, "price" : 1750, "mr_avg" : 25 } }  
{ "_id" : "Guaraná Fantástica", "value" : { "count" : 10, "price" : 45, "mr_avg" : 4.5 } }
```

Zadatak 2.

- Korištenjem *aggregation pipeline frameworka* izračunati prosječnu cijenu svakog prodanog proizvoda za sve narudžbe nakon 1.1.1998. koje su naručene iz Francuske (isto kao u prvom zadatku). Rezultate spremiti u istu kolekciju i isti dokument (upariti prema `_id` polju), tako da se ovaj izračunati prosjek doda u novo polje `agg_avg`.
- Ispisati prvih 10 zapisa iz kolekcije `france98`. Napraviti upit za usporedbu prosjeka izračunatih putem M/R i AP frameworka.

Zadatak 2.

```
db.orders.aggregate( [
  { $match: { "order_date" : {$gte: "1998-01-01"} ,
    "customer.country":"France" } },

  { $unwind: "$details" },

  { $group: {_id: "$details.product.name",
    quantity: {$sum: "$details.quantity"},
    price: {$sum: { $multiply: ["$details.unit_price", "$details.quantity"] } } } },

  { $project: {agg_avg: {$divide: ["$price", "$quantity"] } } },

  { $merge: { into: "france98", on: "_id",
    whenMatched: "merge",  whenNotMatched: "insert" }}

])
```

1. Bitno je odmah na početku napraviti match i odrezati samo one dokumente koji nas zanimaju, radi brzine i manjeg opterećenja.

2. Razmotamo detalje

3. Grupiramo po nazivu proizvoda i njega koristimo kao `_id`. Računamo količinu i ukupnu plaćenu cijenu proizvoda kao i u M/R

4. Dalje uz ključ samo projiciramo prosjek kojih odmah izračunamo. Ostala polja nestaju iz dokumenata

5. Uklapamo dokumente u postojeću kolekciju. Ključno je *whenMatched: "merge"* kako bismo spojili dokumente s istim `_id`-jem.

Zadatak 2. – par napomena

- Umjesto

```
{ $project: {agg_avg: {$divide: ["$price", "$quantity"]} } } },
```

je bilo moguće napravili i nešto kao:

```
{ $set: {agg_avg: {$divide: ["$price", "$quantity"]} } } },  
{ $unset: ["$price", "$quantity"]} },
```

- U slučaju spajanja dokumenata ste imali razna rješenja, koja su također ok, npr.:

```
{ $merge: { into: "france98",  
            whenMatched:  
                [{ $addFields: { "value.agg_avg": "$$new.agg_avg" } } ],  
            whenNotMatched: "insert" } }
```

a ovo koje sam napisao je najjednostavnije.

Zadatak 2. – kako sada izgleda kolekcija france98

```
> db.france98.find().limit(10)
{ "_id" : "Alice Mutton", "value" : { "count" : 37, "price" : 1443, "mr_avg" : 39 }, "agg_avg" : 39 }
{ "_id" : "Carnarvon Tigers", "value" : { "count" : 6, "price" : 375, "mr_avg" : 62.5 }, "agg_avg" : 62.5 }
{ "_id" : "Chartreuse verte", "value" : { "count" : 21, "price" : 378, "mr_avg" : 18 }, "agg_avg" : 18 }
{ "_id" : "Chocolade", "value" : { "count" : 8, "price" : 102, "mr_avg" : 12.75 }, "agg_avg" : 12.75 }
{ "_id" : "Côte de Blaye", "value" : { "count" : 5, "price" : 1317.5, "mr_avg" : 263.5 }, "agg_avg" : 263.5 }
{ "_id" : "Filo Mix", "value" : { "count" : 5, "price" : 35, "mr_avg" : 7 }, "agg_avg" : 7 }
{ "_id" : "Geitost", "value" : { "count" : 11, "price" : 27.5, "mr_avg" : 2.5 }, "agg_avg" : 2.5 }
{ "_id" : "Gorgonzola Telino", "value" : { "count" : 35, "price" : 437.5, "mr_avg" : 12.5 }, "agg_avg" : 12.5 }
{ "_id" : "Grandma's Boysenberry Spread", "value" : { "count" : 70, "price" : 1750, "mr_avg" : 25 }, "agg_avg" : 25 }
{ "_id" : "Guaraná Fantástica", "value" : { "count" : 10, "price" : 45, "mr_avg" : 4.5 }, "agg_avg" : 4.5 }
```

Ovo je bilo najjednostavnije rješenje. Moglo se dodati `agg_avg` unutar `value`, moglo se napraviti `value_agg`, sva rješenja su u redu.

Usporedba rješenja

- Ovdje ste imali razna kreativna ili manje kreativna rješenja.
- Najjednostavnije je ispisati broj dokumenata koji se razlikuju:

```
➤ db.france98.find(  
    { $where: "this.value.mr_avg != this.agg_avg" }).count()
```

0

Nema dokumenata koji se razlikuju

Usporedba rješenja

- Malo složeniji upit za usporedbu rješenja:

```
> db.france98.aggregate([  
  {$project: {_id:1, equals: {$eq: ["$value.mr_avg", "$agg_avg"]}} }  
])
```

```
{ "_id" : "Alice Mutton", "equals" : true }  
{ "_id" : "Carnarvon Tigers", "equals" : true }  
{ "_id" : "Chartreuse verte", "equals" : true }  
{ "_id" : "Chocolade", "equals" : true }  
{ "_id" : "Côte de Blaye", "equals" : true }  
{ "_id" : "Filo Mix", "equals" : true }  
{ "_id" : "Geitost", "equals" : true }  
{ "_id" : "Gorgonzola Telino", "equals" : true }
```

Svaki upit kojim ste pokazali da su rješenja jednaka (ili nisu) je bio ok.

Hvala na pažnji!

Javite se na mail ako treba što
individualno komentirati ...