

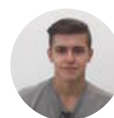


3. domaća zadaća iz NMBP-a (3rd homework Advanced Databases) (MR)

1(</>)

2(</>)

3(</>)



Luka Ilić
li51404@fer.hr
0036514047

3 days
02:08:49

[Submit](#)

Korištenjem kolekcije `dvdtrent` za zaposlenika **Larry Foster** odredite ukupan iznos **totAmount** kojeg je naplatio iznajmljivanjem filmova iz kategorije **Comedy** i listu tih filmova s iznosima **amount** ostvarenima iznajmljivanjem pojedinog filma navedene kategorije.

Film se treba pojavljivati samo jednom, ali iznos treba biti ukupan iznos za taj film.

Filmove **poredati uzlazno po abecedi**.

Iznose (**totAmount**, **amount**) zaokružiti na tri decimale.

Nazivi atributa i format se vide iz primjera rezultata:

```
[
  {
    "_id": {
      "staf": "Larry Foster"
    },
    "value": {
      "totAmount": 77.8,
      "films": [
        {
          "film_title": "Bringing Hysterical",
          "amount": 7.98
        },
        {
          "film_title": "Butterfly Chocolat",
          "amount": 6.98
        },
        ...
      ]
    }
  }
]
```

Using the `dvdtrent` collection for employee **Larry Foster** give the total amount `totAmount` that Larry charged by renting movies from the **Comedy** category and a list of these films and corresponding amounts made by renting films of this category.

The film should appear only once in the list but the amount should be the total amount for that film.

**** Sort films ascending alphabetically****. Round the amounts (**totAmount**, **amount**) to three decimal places.

Attribute names and the overall result format can be seen from the example above.

```
1 db.dvdtrent.mapReduce(
2   `function() {
3     var key = {staf: this.staff.first_name + " " + this.staff.last_name}
4     var value = {films: [ {film_title: this.film.title, amount: this.payment
```

[Run](#)[Save](#)

```

5      emit (key, value)
6    };`,
7    `function(key, values) {
8      var rv = {
9        films: []
10     }
11     values.forEach(function(value) {
12       value.films.forEach(function(film) {
13         var index = rv.films.findIndex(x => x.film_title==film.film_title);
14         if (index === -1){
15           rv.films.push(film)
16         } else {
17           rv.films[index].amount += film.amount
18         }
19       })
20     })
21     return rv
22   };`,
23   {
24     out: "videoteka",
25     query: {"staff.first_name": "Larry", "staff.last_name": "Foster", "film.categories.name": null}},
26     sort: {"film.title": 1},
27     finalize: `function(key, rv) {
28       rv.totAmount = 0
29       rv.films.forEach(function(film) {
30         rv.totAmount += film.amount
31         film.amount = parseFloat(film.amount.toFixed(2))
32       })
33       rv.totAmount = parseFloat(rv.totAmount.toFixed(2))
34       return rv
35     }`
36   }
37 )

```

😊 Correct! Well done!

Result:

```

[
  {
    "_id": {
      "staf": "Larry Foster"
    },
    "value": {
      "films": [
        {
          "film_title": "Bringing Hysterical",
          "amount": 7.98
        },
        {
          "film_title": "Butterfly Chocolat",
          "amount": 6.98
        },
        {
          "film_title": "Caper Motions",
          "amount": 2.97
        }
      ]
    }
  }
]

```

```
{,
{
  "film_title": "Element Freddy",
  "amount": 4.99
},
{
  "film_title": "Ferris Mother",
  "amount": 10.98
},
{
  "film_title": "Flintstones Happiness",
  "amount": 9.99
},
{
  "film_title": "Heaven Freedom",
  "amount": 2.99
},
{
  "film_title": "Hunchback Impossible",
  "amount": 7.99
},
{
  "film_title": "Hustler Party",
  "amount": 7.99
},
{
  "film_title": "Language Cowboy",
  "amount": 1.99
},
{
  "film_title": "Lion Uncut",
  "amount": 0.99
},
{
  "film_title": "Lonely Elephant",
  "amount": 3.99
},
{
  "film_title": "Paradise Sabrina",
  "amount": 3.99
},
{
  "film_title": "Perfect Groove",
  "amount": 2.99
},
{
  "film_title": "Tramp Others",
  "amount": 0.99
}
],
"totAmount": 77.8
}
]
```