Aggregation Homework Tasks

author: Hubert Łyczek
version: *1.0.0*

Introduction

This document contains 8 tasks for you, to master your skills.

Here are details of the collection that will be used in all the tasks:

database: *aggregation*
collection: *people*

This collection contains 200 000 documents.

```
> use aggregation
switched to db aggregation
>
> db.people.count()
200000
```

Expected results

For every task you will find an expected result. Don’t focus on its structure too much. If you create a solution that returns the right values, but the result has a slightly different structure, that’s fine. The right values are the most important thing.

Task 1 Easy

Task

Count the number of people, divided into countries.

Expected result

The result should be simil~~i~~ar to this example:

```
{
    "_id" : "Country1",
    "total" : 200
}
{
    "_id" : "Country2",
    "total" : 100
}
```

Task 2 Easy

Task

What is the most popular address, and how many people live there?

Expected result

The result should be similar to this example:

```
{
    "_id" : {
        "country" : "Country",
        "city" : "City",
        "postalCode" : "11111",
        "street" : "Street"
    },
    "total" : 100
}
```

Task 3

Easy

Task

How many people from each country have visited a restaurant?

Hint

People who have visited a restaurant have one or more payments like this one in the payments array:

```
{
    "category" : "food",
    "name" : "restaurant",
    "amount" : 17.53
}
```

Expected result

The result should be similar to this example:

```
{ "_id" : "Country1", "visits" : 500 }
{ "_id" : "Country2", "visits" : 500 }
```

Task 4

Medium

Task

Find the 3 people who have the most in their bank accounts.

Hint

You can find bank accounts in the *wealth.bankAccounts* field. This field contains an array, as every person can have many bank accounts. Add them all for each person.

Expected result

The expected result should contain 3 documents similar to this example:

```
{
    "firstName" : "firstName",
    "lastName" : "lastName",
    "totalBalance" : 11111.11
}
```

Task 5

Hard

Task

Count the number of all restaurant visits, the total amount spent, and the average amount per visit. All of this should be divided into countries.

Hint

People who have visited a restaurant have one or more payments like this one in the payments array:

```
{
    "category" : "food",
    "name" : "restaurant",
    "amount" : 17.53
}
```

One person could have been to a restaurant more than once.

Expected result

The result should be similar to this example:

```
{
    "_id" : "Country1",
    "totalVisits" : 97,
    "totalAmount" : 1132,
    "avgAmount" : 11.670103092783505
}
{
    "_id" : "Country2",
    "totalVisits" : 453,
    "totalAmount" : 9851,
    "avgAmount" : 21.746136865342163
}
```

Task 6 Hard

Task

There is one country in which the average payment in a restaurant is the highest, and one in which the average payment in a restaurant is the lowest. How many times do people from the first country spend more than people from the second country?

Hint

The same as in the previous task.

Expected result

The result could be similar to this example:

```
{
    "_id" : null,
    "diff" : 3.2434252345234
}
```

Which means that people from one country pay 3.24 times more on average.

Task 7 Hard

Task

Count the number of people in each country according to age groups:

- **18-29**
- **30-39**
- **40-49**

Hint

1. First of all, the number of people in each group depends on the current date. If we execute our query today, or in one week, we could get different results. This is because people are getting older every day. Some people who are 29 today, could be 30 tomorrow, the day after tomorrow and so on.
   So it is very important to set an arbitrary date that will be used to calculate a person's age.
   Let's suppose that the current date is: **2016-06-22**

2. To calculate a person’s age, let’s assume that only years, months and days matter. It isn’t important whether a person was born at 01:23:01 or 23:59:59. They both have the same age.

Examples:

Current date: 2016-06-22

Date of birth	Age
1990-01-25	26
1990-06-14	26
1990-06-22	26
1990-06-23	25
1990-08-01	25

So to calculate age you could use operators like $year, $month, $dayOfMonth, but remember that there are people born before 1970-01-01 in our collection:

```
> db.people.find({birthDate: {$lt: new ISODate('1970-01-01T00:00:00.000')}}).count()
23784
```

Some operating systems do not support dates before this date. This means that you must deal with this.

Fortunately, you can take advantage of the fact that there are no people older than 49 in our database.

Expected result

The result set should contain six documents similar to this example:

```
{
    "_id" : {
        "country" : "Country1",
        "ageRange" : "18-29"
    },
    "count" : 100
}
```

Task 8 Hard

Task

Calculate what percentage of the population of the country are the age groups:

- **18-29**
- **30-39**
- **40-49**

The result should be rounded to two decimal places.

Hint

Use the results from the previous task. Just save the results from the previous query to a new collection. Use this collection as the input.

Expected result

The result set should contain six documents similar to this example:

```
{
    "_id" : {
        "country" : "Country1",
        "ageRange" : "18-29"
    },
    "percent" : 33.33
}
```