



Big Data Systems & Architectures

Redis-MongoDB Assignment

Christos Kallaras, p2822009

Stavros Kasiaris, p2822022

Athens, 14/03/2021

Contents

1. Task 1.....	3
1.1 How many users modified their listing on January?.....	3
1.2 How many users did NOT modify their listing on January?	3
1.3 How many users received at least one e-mail per month (at least one e-mail in January and at least one e-mail in February and at least one e-mail in March)?.....	3
1.4 How many users received an e-mail on January and March but NOT on February?	4
1.5 How many users received an e-mail on January that they did not open but they updated their listing anyway?	4
1.6 How many users received an e-mail on January that they did not open but they updated their listing anyway on January OR they received an e-mail on February that they did not open but they updated their listing anyway on February OR they received an e-mail on March that they did not open but they updated their listing anyway on March?.....	4
1.7 Does it make any sense to keep sending e-mails with recommendations to sellers? Does this strategy really work? How would you describe this in terms a business person would understand?.....	4
2. Task 2.....	5
2.1 Add your data to MongoDB.....	5
2.2 How many bikes are there for sale?	5
2.3 What is the average price of a motorcycle (give a number)? What is the number of listings that were used in order to calculate this average (give a number as well)? Is the number of listings used the same as the answer in 2.2? Why?.....	6
2.4 What is the maximum and minimum price of a motorcycle currently available in the market?	6
2.5 How many listings have a price that is identified as negotiable?.....	6

1. TASK 1

1.1 HOW MANY USERS MODIFIED THEIR LISTING ON JANUARY?

```
> r$BITCOUNT("ModificationsJanuary")  
[1] 9969
```

We created a bitmap called ModificationsJanuary that has 1 when the User modified his/her listing in January. We used the UserID as indicator in the bitmap in order to avoid Users that exists more than once in the dataset (we also use this in the other questions). The Users that modified their listing on January are 9969.

1.2 HOW MANY USERS DID NOT MODIFY THEIR LISTING ON JANUARY?

```
> r$BITCOUNT("NoModificationsJanuary")  
[1] 10031
```

Using BITOP NOT we found the users that have 0 in the bitmap ModificationsJanuary which are the users that did not modified their listing in January. There are 10031 such users. The sum of this users and the ones from 1.1 is 20000. Using excel and calculating the unique values of column UserID we found that the total number of users are 19999. That 1 difference can be explained by the fact that all BITOP operations happen at byte-level increments. Redis stores the length of the string at the byte level so for one extra bit Redis will store a full byte (which is 8 bits).

1.3 HOW MANY USERS RECEIVED AT LEAST ONE E-MAIL PER MONTH (AT LEAST ONE E-MAIL IN JANUARY AND AT LEAST ONE E-MAIL IN FEBRUARY AND AT LEAST ONE E-MAIL IN MARCH)?

```
> r$BITCOUNT("EmailsOnMonths")  
[1] 2668
```

First, we create 3 bitmaps to store which user received at least one e mail for each month. Then with BITOP AND we create a bitmap called EmailsOnMonths that contains the users that received at least one email per month. There are 2668 users.

1.4 HOW MANY USERS RECEIVED AN E-MAIL ON JANUARY AND MARCH BUT NOT ON FEBRUARY?

```
> r$BITCOUNT("answer")  
[1] 2417
```

Using the previous bitmaps we create a new one storing the users that received an email on January and March. Using BITOP NOT we create a bitmap that has the users that did not received an email in February. Finally, we use BITOP AND to combine those 2 bitmaps. The answer is 2417.

1.5 HOW MANY USERS RECEIVED AN E-MAIL ON JANUARY THAT THEY DID NOT OPEN BUT THEY UPDATED THEIR LISTING ANYWAY?

```
> r$BITCOUNT("EmailsNotOpenedJanuaryButModified")  
[1] 2807
```

We created a new bitmap that has the users that did not opened their e mail on January and then combine it with the bitmap with the users that modified their listing on January. The final users are 2807.

1.6 HOW MANY USERS RECEIVED AN E-MAIL ON JANUARY THAT THEY DID NOT OPEN BUT THEY UPDATED THEIR LISTING ANYWAY ON JANUARY OR THEY RECEIVED AN E-MAIL ON FEBRUARY THAT THEY DID NOT OPEN BUT THEY UPDATED THEIR LISTING ANYWAY ON FEBRUARY OR THEY RECEIVED AN E-MAIL ON MARCH THAT THEY DID NOT OPEN BUT THEY UPDATED THEIR LISTING ANYWAY ON MARCH?

```
> r$BITCOUNT("EmailsNotOpenedButModified")  
[1] 7221
```

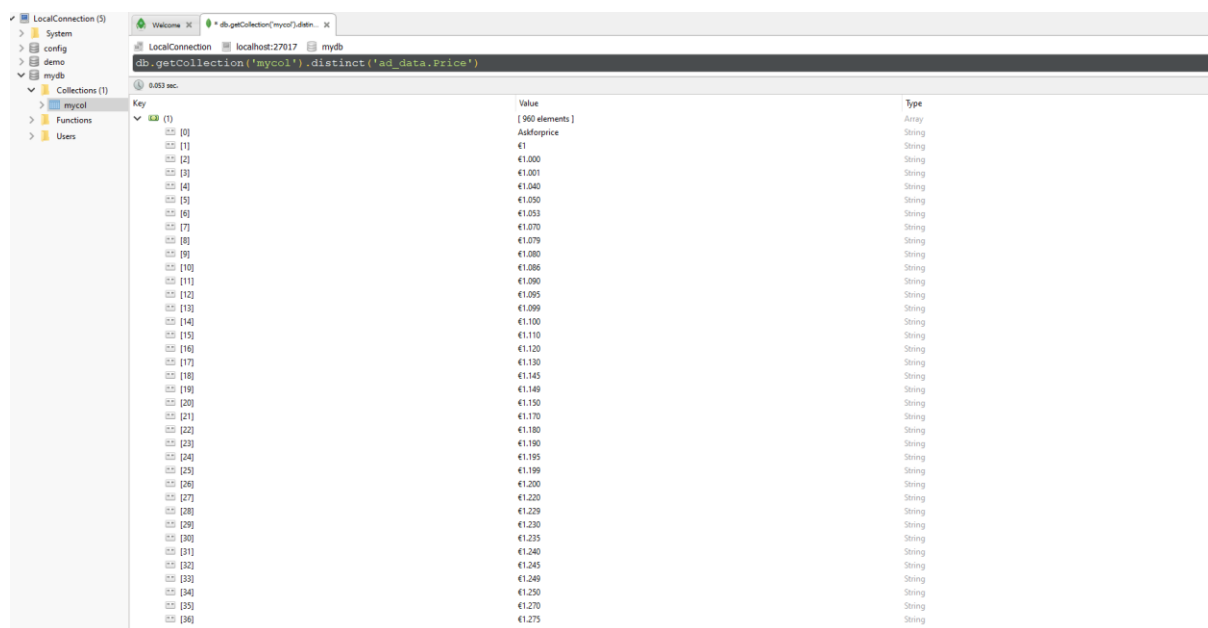
1.7 DOES IT MAKE ANY SENSE TO KEEP SENDING E-MAILS WITH RECOMMENDATIONS TO SELLERS? DOES THIS STRATEGY REALLY WORK? HOW WOULD YOU DESCRIBE THIS IN TERMS A BUSINESS PERSON WOULD UNDERSTAND?

We can see by analysing the data that **49,54% on January, 50,23% on February and 49,94% on March** opened their email and then also modified their listing. In general, **50% opened and then modified** which means that this strategy influences users, but it can be improved in order to further increase their impact. A percentage of 50% may be good now but it does not guarantee success in the future.

2. TASK 2

2.1 ADD YOUR DATA TO MONGODB.

Before going to the questions, we need to clean the data. First, we load them into MongoDB in a collection called mycol. In there we can see that the field “Price” has a value, among others, of “Askforprice” as we can see in Figure 1 . Also, there is the price of 1 euro. Most data come from car.gr so searching there we saw that prices with 1 euro are fake and the buyer must contact the dealer in such cases. There are also cases with 2 or 3 euros. Since it seems unlikely that a motorcycle cost less than 12 euros, we assigned null to all prices below 12 euros and for value “Askforprice”. We chose the limit of 12 because that is the smallest non logical value for a motorcycle. There are prices like 20 euros but we can assume that an old used motorcycle could be up for sale for 20 euros. For all the others we transformed them into numerics while also remove the “€” sign. We did that same transformation also for fields with numeric value plus a metric but defined as strings. That fields are “Mileage” (metric was km), “Cubic capacity” (metric was cc) and “Power” (metric was bhp). To load the data, we created a txt file containing the paths for every json file (using the command in the tip section of the assignment). We then load them into a new collection called mycol_2.



Key	Value	Type
[0]	Askforprice	String
[1]	€1	String
[2]	€1.000	String
[3]	€1.001	String
[4]	€1.040	String
[5]	€1.250	String
[6]	€1.253	String
[7]	€1.070	String
[8]	€1.079	String
[9]	€1.080	String
[10]	€1.086	String
[11]	€1.090	String
[12]	€1.095	String
[13]	€1.099	String
[14]	€1.100	String
[15]	€1.110	String
[16]	€1.120	String
[17]	€1.130	String
[18]	€1.145	String
[19]	€1.149	String
[20]	€1.150	String
[21]	€1.170	String
[22]	€1.180	String
[23]	€1.190	String
[24]	€1.195	String
[25]	€1.199	String
[26]	€1.200	String
[27]	€1.220	String
[28]	€1.229	String
[29]	€1.230	String
[30]	€1.235	String
[31]	€1.240	String
[32]	€1.245	String
[33]	€1.249	String
[34]	€1.250	String
[35]	€1.270	String
[36]	€1.275	String

Figure 1: Price's distinct values

2.2 HOW MANY BIKES ARE THERE FOR SALE?

```
> m2$count()  
[1] 29700
```

There are 29700 bikes for sale.

2.3 WHAT IS THE AVERAGE PRICE OF A MOTORCYCLE (GIVE A NUMBER)? WHAT IS THE NUMBER OF LISTINGS THAT WERE USED IN ORDER TO CALCULATE THIS AVERAGE (GIVE A NUMBER AS WELL)? IS THE NUMBER OF LISTINGS USED THE SAME AS THE ANSWER IN 2.2? WHY?

```
> # Question 3
> m2$aggregate('[{
+   "$match" : {"ad_data.Price" :
+     { "$exists" : true } }},
+   {"$group":{"_id": "ad_id",
+     "Average_Price": {"$avg":"$ad_data.Price"},
+     "Bikes": {"$sum" : 1 }}}]')
  _id Average_Price Bikes
1 ad_id      3020.992 28582
```

The average price of a motorcycle is 3020.992€ and the number of listings that were used to calculate this, is 28582. This number is smaller than the one of the previous question (29700) because for this calculation the motorcycles that have null in Price did not taken into account. Null in Price have the motorcycles that have "Askforprice" as value in price or value < 12.

2.4 WHAT IS THE MAXIMUM AND MINIMUM PRICE OF A MOTORCYCLE CURRENTLY AVAILABLE IN THE MARKET?

```
> # Question 4
> m2$aggregate('[{
+   "$match" :
+     { "ad_data.Price" :
+       { "$exists" : true } }},
+   {"$group":{"_id": null,
+     "Min_Price":{"$min":"$ad_data.Price"},
+     "Max_Price":{"$max":"$ad_data.Price"}
+   }}]')
  _id Min_Price Max_Price
1 NA      12      89000
```

The min price is 12 € which is the limit we defined when we cleaned the data, and the max price is 89000€. If we hadn't remove the "Askforprice" value then the max price would be NA and our calculations would be wrong.

2.5 HOW MANY LISTINGS HAVE A PRICE THAT IS IDENTIFIED AS NEGOTIABLE?

```
> m2$aggregate('[{
+   "$match" :
+     { "metadata.model":
+       { "$regex" : "Negotiable", "$options" : "i" } }},
+   {"$group":{"_id": null,
+     "Bikes": {"$sum" : 1 }
+   }}]')
  _id Bikes
1 NA  1348
```

1348 bikes have a price that is identified as negotiable.