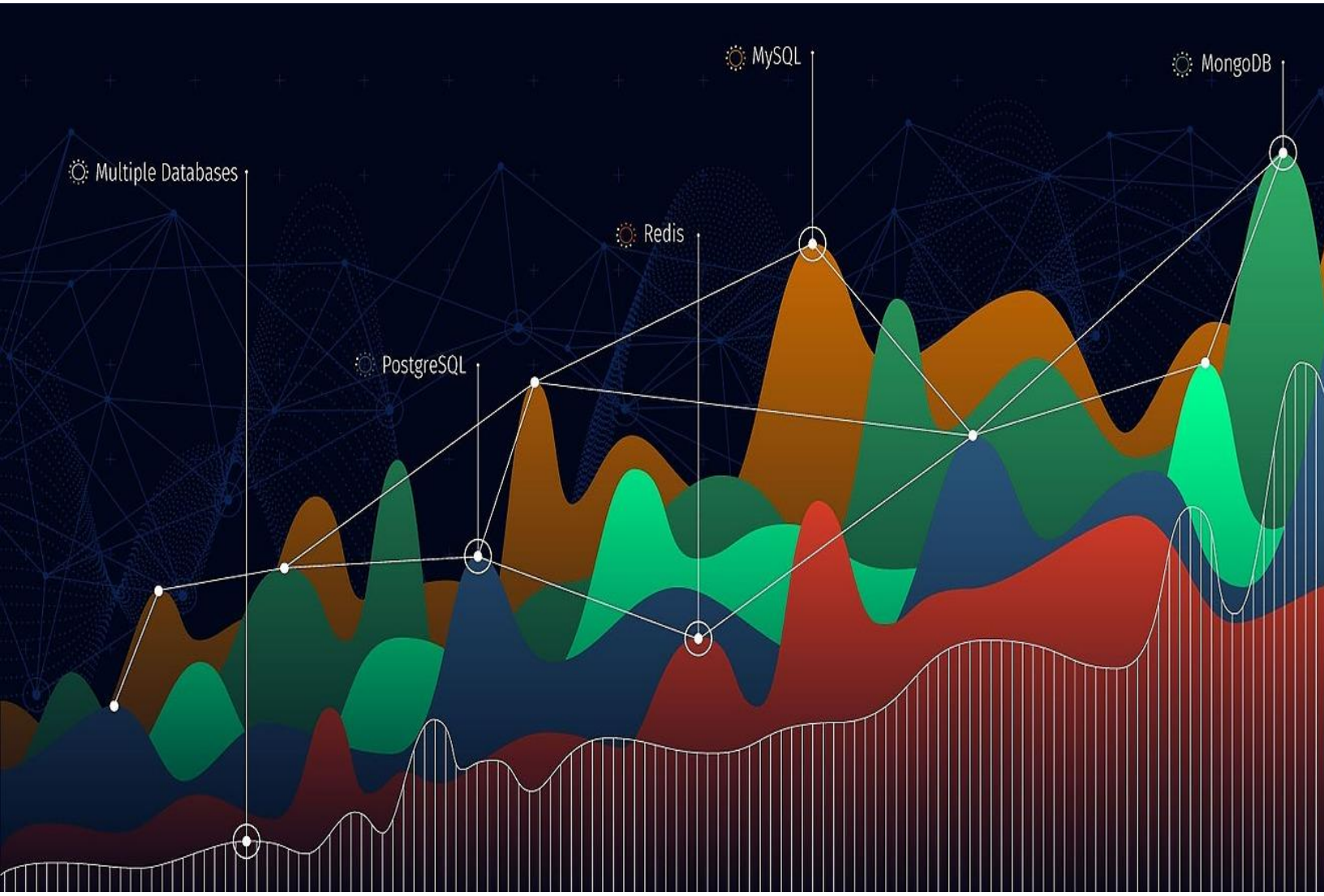


BIG DATA SYSTEMS AND ARCHITECTURES

MSc in Business Analytics
Athens University of Economics and Business

Alexandra Ntetsika 2822014
Georgia Vlassi 2822001

March 2021

Task 1 | REDIS

In order to answer the questions regarding Redis Task, commands like BITOP, SETBIT and BITCOUNT with arguments NOT, AND, OR were used. A brief summary of each result will be given for each question. The Task1.R file includes all relevant commented code for Redis part.

1. How many users modified their listing in January?

A BITMAP called "ModificationsJanuary" was created for each user in modified_listings data frame, who modified their listing in January. By the use of BITCOUNT all these users were calculated.

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

2. How many users did NOT modify their listing in January?

The inversion of the aforementioned BITMAP "ModificationsJanuary" was used with argument NOT at BITCOUNT, to calculate the users who did not modify their listing in January.

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

Adding the result of questions 1 and 2, the total amount of users with modifications or not in January are 20.000. Observing the data in data frame modified_listings for January, the total amount of users is 19.999. This difference is due to byte-level increments in BITMAP operations. The BITMAP operations should be an integer multiplied by 8, as each byte is equal to 8 bits.

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)?

Three BITMAPs called “EmailsJanuary”, “EmailsFebruary” and “EmailsMarch” were created for each user in emails_sent data frame, by setting the SETBIT to 1 if the number of month is 1, 2 or 3 respectively. By using BITCOUNT, in the final BITMAP, with argument AND all these users were calculated**.**

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

4. How many users received an e-mail on January and March but NOT in February?

Using the results of the previous query, we calculated the users of “EmailsJanuary” and “EmailsMarch”, by the use of BITCOUNT with argument AND. Also, the inversion of “EmailsFebruary” was calculated with BITCOUNT with argument NOT. In order to calculate the users received an e-mail in January and March but NOT in February, the previous results were combined, using BITCOUNT with argument AND.

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

5. How many users received an e-mail in January that they did not open but they updated their listing anyway?

In emails_sent data frame, a BITMAP called “EmailNotOpenedJanuary” was created for each user, who did not open his/her email in January. These users were combined with BITMAP "ModificationsJanuary" users from question 1. By the use of BITCOUNT with argument AND, the requested user were calculated.

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

6. How many users received an e-mail in January that they did not open but they updated their listing anyway in January OR they received an e-mail in February that they did not open but they updated their listing anyway in February OR they received an e-mail in March that they did not open but they updated their listing anyway in March? How many users received an e-mail in January that they did not open but they updated their listing anyway?

Using the same method as question 1, two new BITMAPs “ModificationsFebruary” and “ModificationsMarch” were created, including the users who modified their listing in February and March respectively. Similar to question 5, two BITMAPs “EmailNotOpenedFebruary” and “EmailNotOpenedMarch” were created. The above results of months February and March were calculated using BITCOUNT with argument AND. Finally, the requested users were calculated by combining the final three BITMAPS “EmailsOpenedJanuary”, “EmailsOpenedFebruary”, “EmailsOpenedMarch” to “EmailsOpened” using BITCOUNT with argument OR.

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

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?

In order to come in a conclusion whether, the method of sending e-mails with recommendations to sellers have an impact or not, a further investigation in our data was implemented.

In the above question, we calculate the following numbers:

Total number of users who opened their emails in January: 14.297

Total number of users who opened their emails in February: 14.290

Total number of users who opened their emails in March: 14.395

Total number of users who did modification in their listing in January: 9969

Total number of users who did modification in their listing in February: 10.007

Total number of users who did modification in their listing in March: 9.991

Total number of users who did not modification in their listing in January: 10.031

Total number of users who did not modification in their listing in February: 9.993

Total number of users who did not modification in their listing in March: 10.009

Total number of users who opened their email AND did a modification in their listing in January: 7.162

Total number of users who opened their email AND did a modification in their listing in February: 7.185

Total number of users who opened their email AND did a modification in their listing in March: 7.173

Total number of users who opened their email AND did not modify their listing in January: 7.135

Total number of users who opened their email AND did not modify their listing in February: 7.108

Total number of users who opened their email AND did not modify their listing in March: 7.173

From the above numbers we calculate the following percentages:

Percentage of users who opened their email AND modified their listing in January: (7.162/14.297)*100 ~ 50%

Percentage of users who opened their email AND modified their listing in February: (7.185/14.290)*100 ~ 50%

Percentage of users who opened their email AND modified their listing in March:
(7.173/14.395)*100 ~ 49%
Percentage of users who opened their email AND did not modify their listing in January:
(7.135/14.297)*100 ~ 50%
Percentage of users who opened their email AND did not modify their listing in February:
(7.185/14.293)*100 ~ 50.2%
Percentage of users who opened their email AND did not modify their listing in March:
(7.173/14.395)*100 ~ 49%
As a result, we observe that percentages of users who opened their email and modify their listing and percentages of users who opened their email but not modify their listing are very close. So, this specific strategy by sending emails has a little effect influencing on sellers to modify their listing.

8. Do the previous subtasks again by using any type of relational or non-relational database. Compare the complexity of the solutions. Then benchmark the query execution time for the dataset that you have. At last, boost the number of entries to 1 billion rows (create your own dummy entries). Perform the benchmark again.

For this question, we used Oracle PL/SQL (Procedural Language for SQL). More specifically, we created two tables in Oracle database, emails_sent and modified_listings. After that, we created a package specification and body with one procedure named generate_data. In this procedure, we inserted random values with specific range (e.g.: 1-3 in m MonthID column). We executed the procedure and then wrote the relevant queries, which can be found in the attachment “1.8.sql”. The code was implemented with 10.000 rows. Comparing these two solutions, we can suggest that Redis is better in such cases, because of better performance, speed and this dataset’s data type. Especially when the volume of data is near to 1 billion.

Task 2 | MONGO

1. Describe Data Cleaning

For the answers below you can find the relevant commented code in Task2.R file. Before implementing the second task regarding MongoDB, a list with all bikes was created in Unix OS and named "files_list.txt". This step is necessary, so as all json files could be read simultaneously in a for loop. All files were inserted to MongoDB and a further investigation in data was applied. Observing

the data and the below questions, some data modifications and manipulations were implemented. All these data updates were executed in a built-in function in R. More, specific the affected fields are:

- Milesage (ad_data$Mileage): The measurement unit “km” was removed and the type of field was updated into numeric.
- Cubic capacity (ad_data$'Cubic capacity'): The measurement unit “cc” was removed and the type of field was updated into numeric.
- Power (ad_data$Power): The measurement unit “bhp” was removed and the type of field was updated into numeric.
- Make/Model (ad_data$`Make/Model`): The part of year was removed (e.g. ‘85)
- Registration (ad_data$Registration): The part before year was removed.(e.g. 9/), as mentioned in question 2.8.
- New field Motoage (ad_data$Motoage): This field was created in the context of question 2.8. The Motoage field was calculated by abstracting the Registration filed of each motorcycle from the year 2021.
- Price (ad_data$Price): In order to manipulate questions with average price of a moto, it was important to handle numeric data. As a result, the currency symbol “€” and the “.” from the decimal part of the price were removed. Additionally, values with “Askforprice” were updated to null and a new field named ad_data$Askforprice was added to these files. Regarding question 2.4, where the minimum price of a motorcycle demanded, there are prices equal to 1. We assumed to exclude prices lower than 50, as they indicate sales or mechanical parts of the motorcycle.

2. How many bikes are there for sale?

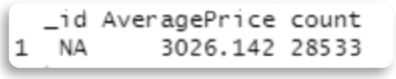
In order to get all the motorcycles that are for sale count() function was used. This result could also be verified from MongoDB by running the command: *db.bikescollection.count()*

```
> mongo$count()
[1] 29701
```

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?

As shown below the average price of a motorcycle is almost 3026 euro. This price occurred by calculating the prices of 28533 motorcycles. From question 2.2 was calculated that 29701 motorcycles inserted for sale in MongoDB. With the modifications in field ‘Price’, the null values and the

‘Askforprice’ values were omitted in the calculation of average price. This fact justifies the differences in count of motorcycles.



	_id	AveragePrice	count
1	NA	3026.142	28533

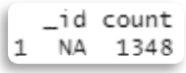
4. What is the maximum and minimum price of a motorcycle currently available in the market?

With the modifications mentioned in introduction, prices of motorcycles lower than 50 euro were omitted, as they were indicated to sales in mechanical parts of the motorcycle. We get these information from site Car.gr. Additionally, minimum and maximum prices could be verified from MongoDB by running the command: *db.bikescollection.distinct("ad_data.Price").*

	_id	MinimumPrice	MaximumPrice
1	NA	50	89000

5. How many listings have a price that is identified as negotiable?

In order to calculate the listings that have a price described as ‘Negotiable’, we used a regex in field metadata.model. To verify the results occurred, the following command could be used in MongoDB: *db.bikescollection.find({ "metadata.model" : { "$regex" : /Negotiable/ } })*



	_id	count
1	NA	1348

6. For each Brand, what percentage of its listings is listed as negotiable?

To find the percentage for each Brand with listings listed as negotiable, two data frame were created. The first data frame named ‘Negotiable_Grouping_Brand’ includes the models with tags negotiable, grouped by brand. The second data frame named ‘Total_Grouping_Brand’ includes all brands grouped. In order to calculate the percentage for each Brand, we calculated the ratio of Negotiable to Total Count multiplied by 100.

_id	Count_Negotiable	Count_Total	Negotiable_Percentage_Per_Brand
Jonway	3	5	60.00
Joyner	1	1	100.00
Kaisar	3	4	75.00
Kawasaki	65	1953	3.33
Keeway	9	103	8.74
Kinroad	1	4	25.00
Kreidler	17	83	20.48
KTM	21	966	2.17
Kuberg	1	1	100.00
Kxd	1	7	14.29
Kymco	21	1148	1.83
Lambretta	10	14	71.43
Lem	3	12	25.00
Lifan	25	104	24.04
Linhai	2	48	4.17
Lintex	1	3	33.33
LML	6	27	22.22
Loncin	3	25	12.00
Maico	1	4	25.00
Malaguti	2	55	3.64

7. What is the motorcycle brand with the highest average price?

To find the motorcycle brand with the highest average price, we grouped the motorcycles by brand and average price. Subsequently, we sort them in descending order in order to get the first brand.

```
    _id AveragePrice count
1 Semog        15600     1
```

8. What are the TOP 10 models with the highest average age?

To find the TOP 10 models with the highest average age, we grouped the motorcycles by Make/Model and Motoage fields. The creation and content of Motoage field was described in introduction. Then, the models were sorted by their Motoage in descending order. As output the first 10 Models are fetched. As there were a problem with encoding, the Greek word ‘Αλλο ’ was replaced with space.

	_id	AveragegAge
1	henderson indian replica	90.0
2	Bmw R 12	87.0
3	MATCHLESS G3 350	86.0
4	Norton H16	85.0
5	Matsoules	83.0
6	Matchless G3/L	82.0
7	NEW HUDSON	82.0
8	Bsa M20 ARMY MOTO!	82.0
9	Bsa M20	81.5
10	Solex 1700	81.0

9. How many bikes have “ABS” as an extra?

To find the bikes that have ‘ABS’ as an extra, we used a regex in field Extras and count these bikes.

```
  _id count
1  NA  4025
```

10. What is the average Mileage of bikes that have “ABS” AND “Led lights” as an extra?

Similar to the previous question, we used a regex to find the bikes with both extras ‘ABS’ and ‘Led Lights’. Then, we grouped these bikes by their average Milesage and count them.

```
  _id AverageMilage count
1  NA       30125.7  1135
```

11. What are the TOP 3 colors per bike category?

To find the TOP 3 colors per bike category, we grouped the bikes by brand and color and kept a count at each color. Subsequently, the colors were ordered in descending order and grouped the top colors per brand. Finally, we returned the slice of the top three colors of each category.

	_id	TopColors
1	CPI	White, Silver, Orange
2	Polaris	Black, Red, White
3	Gemini	Black, Black (Metallic)
4	Ariel	White
5	Dinli	Yellow, White, Grey
6	Lem	Red, Black, Orange
7	Loncin	Black, Red, Blue
8	Semog	White
9	G-force	Black
10	Buggy Motors	Black, Red, Black (Metallic)
11	New Force Motor	Black
12	Aprilia	Black, Black (Metallic), White
13	Boom-Trikes	Dark Red (Metallic)
14	Nomik	Other, White
15	Sachs	Red, Black, Grey
16	Mikilon	Black (Metallic)
17	Shandong Liangzi	Other
18	Aie	Red, Black (Metallic), Black
19	Vor	Black
20	Tgb	Silver (Metallic), White, Yellow
21	SMC	Bordeaux, White, Red (Metallic)
22	Kuberg	Dark Red
23	Quadro	White (Metallic), Red (Metallic), Black
24	Buell	Black (Metallic), Black, Blue
25	Polini	Red
26	Benelli	Red, White, White (Metallic)
27	Lifan	White, Black, Black (Metallic)
28	Moto Guzzi	Black, Red, Black (Metallic)
29	Italjet	Blue, Red (Metallic), Silver (Metallic)
30	TCB	Black, Black (Metallic), Blue (Metallic)
31	Laverda	Silver, Orange
32	Bashan	Black, Yellow, Orange
33	Husaberg	Blue, Yellow, Black
34	Ural	Beige, Black, Dark Green
35	Kinroad	Green, Grey, Yellow (Metallic)
36	Montesa	Yellow, White, Red
37	Piaggio	Black, Black (Metallic), White
38	Access	Black, White, Red
39	Baotian	Black (Metallic), Silver, Grey
40	Xinling	Yellow
41	Maico	Black, Blue, Red