Group 11

Sudharsana Rajasekaran

Varun Suresh

Zhining Chen

Mahesh M Iyer

**MongoDB Project – Google Store Visitor Data**

BUAN 6320.006

Contents

[Data Extraction and loading 3](#_Toc531455161)

[Physical Database 4](#_Toc531455162)

[MongoDB Queries 5](#_Toc531455163)

[Query 1 5](#_Toc531455164)

[Question 5](#_Toc531455165)

[**Which user had the maximum number of visits and when?** 5](#_Toc531455166)

[Translation: 5](#_Toc531455167)

[Screen Shot of MongoDB Query and Results 5](#_Toc531455168)

[Query 2 6](#_Toc531455169)

[Question 6](#_Toc531455170)

[**Is a Windows user more likely to visit the store than Macintosh user?** 6](#_Toc531455171)

[Translation: 6](#_Toc531455172)

[Screen Shot of MongoDB Query and Results 6](#_Toc531455173)

[Query 3 7](#_Toc531455174)

[Question 7](#_Toc531455175)

[**Which date had the least and most number of visitors with non-mobile devices?** 7](#_Toc531455176)

[Translation: 7](#_Toc531455177)

[Screen Shot of MongoDB Query and Results 7](#_Toc531455178)

[Query 4 8](#_Toc531455179)

[Question 8](#_Toc531455180)

[**Were mobile devices users more socially engaged than non-mobile device users?** 8](#_Toc531455181)

[Translation: 8](#_Toc531455182)

[Screen Shot of MongoDB Query and Results 8](#_Toc531455183)

[Query 5 9](#_Toc531455184)

[Question 9](#_Toc531455185)

[**Provide a breakdown of store hits by region** 9](#_Toc531455186)

[Translation: 9](#_Toc531455187)

[Screen Shot of MongoDB Query and Results 9](#_Toc531455188)

[Query 6 11](#_Toc531455189)

[Question 11](#_Toc531455190)

[**How many users used only Macintosh devices to visit the store?** 11](#_Toc531455191)

[Translation: 11](#_Toc531455192)

[Screen Shot of MongoDB Query and Results 11](#_Toc531455193)

[Query 7 12](#_Toc531455194)

[Question 12](#_Toc531455195)

[**What was the average number of hits per unique visitor?** 12](#_Toc531455196)

[Translation: 12](#_Toc531455197)

[Screen Shot of MongoDB Query and Results 12](#_Toc531455198)

[Query 8 13](#_Toc531455199)

[Question 13](#_Toc531455200)

[**Visitors from which country visited the store the most?** 13](#_Toc531455201)

[Translation: 13](#_Toc531455202)

[Screen Shot of MongoDB Query and Results 13](#_Toc531455203)

# Data Extraction and loading

The entire Google Revenue data was extracted into a .csv using R codes. The dataset consisted of 52 columns. The code was able to pick previously inaccessible columns. This ensured that we were able to extract all the JSON columns and place them as new columns in the .csv file (instead of one clustered column)

***Code for extracting the entire dataset in R***



The dataset was then uploaded as a flat file into Mongo DB

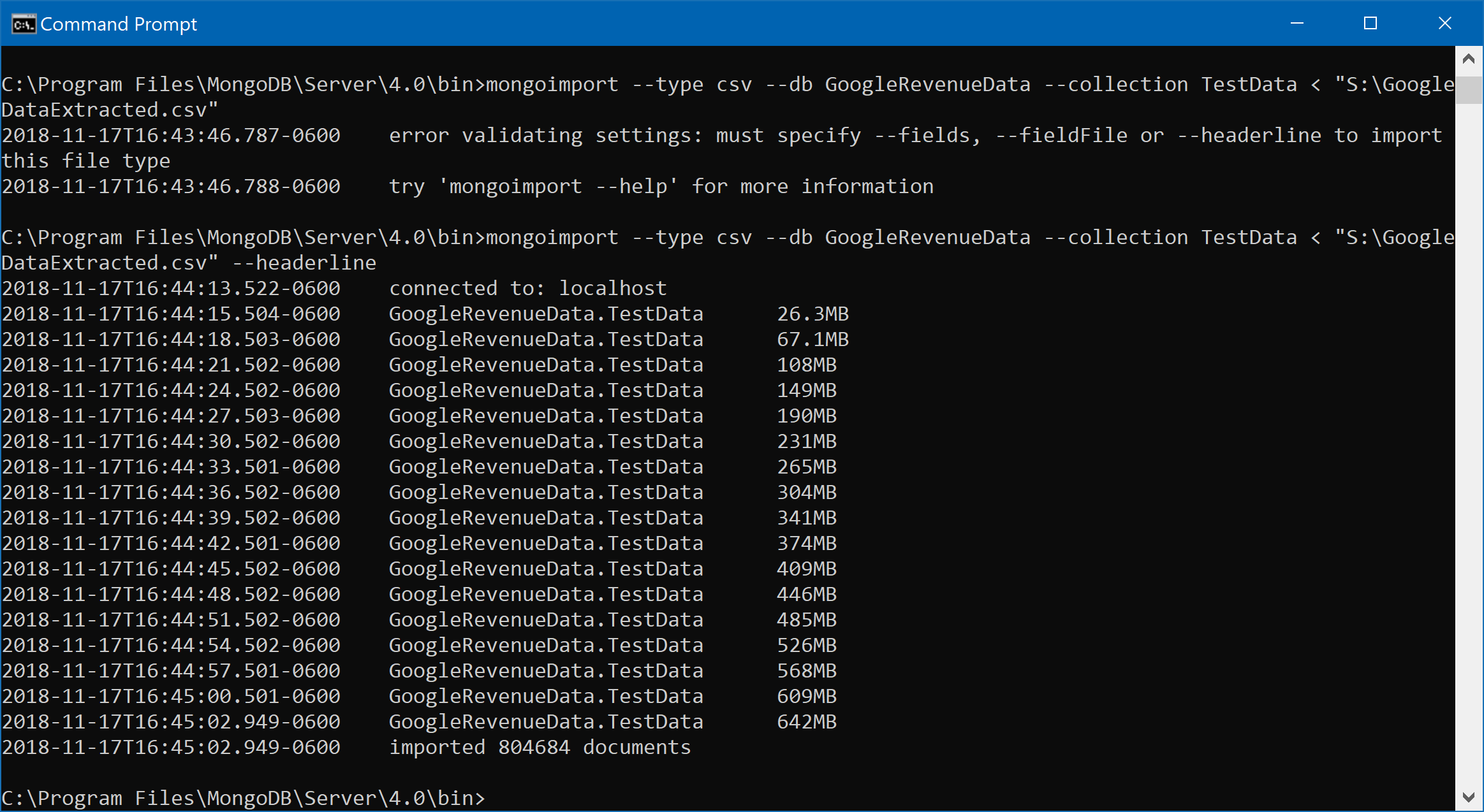


Figure 1: Data loading procedure to MongoDB

Once the data was loaded, we were able to access the same through the MongoDB Compass platform.

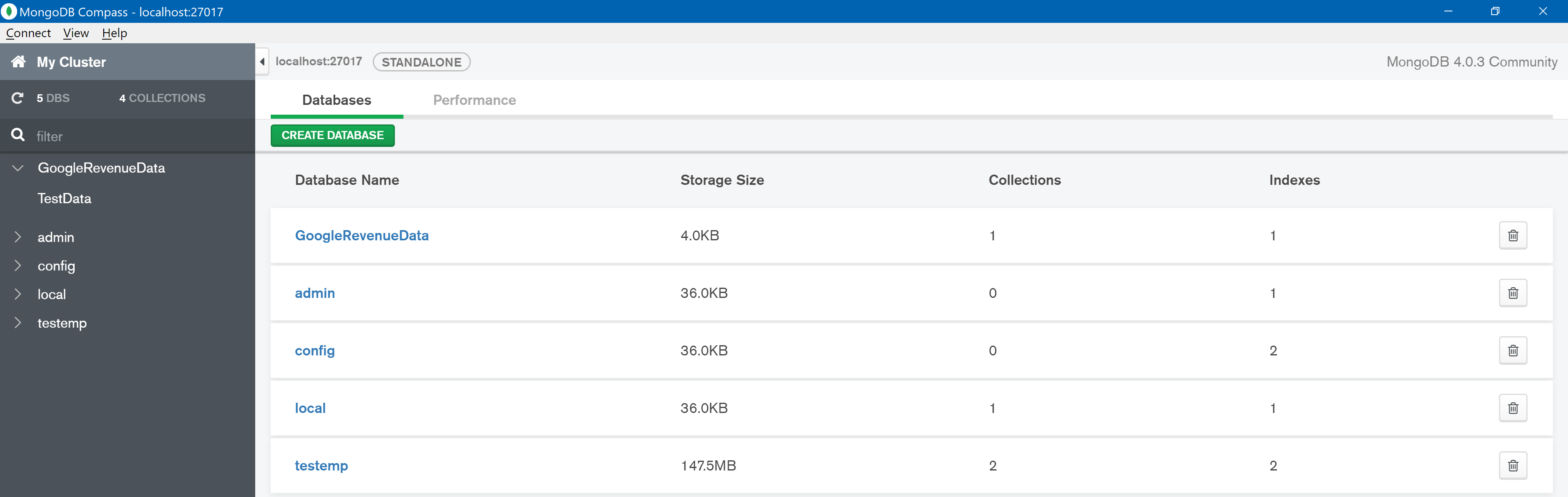


Figure 2: Dataset in MongoDB Compass platform

# Physical Database

Upon loading the dataset into the database, we were able to check its presence by executing the “show dbs” command. This provided the list of datasets that were present in our database currently. This step was a validation to check if the dataset was successfully loaded into the DB.

|  |  |  |
| --- | --- | --- |
| **Collection Name** | **Relationships with Other Collections (if any)** | **# of Rows in Table** |
| GoogleRevenueData | - | 804684 |

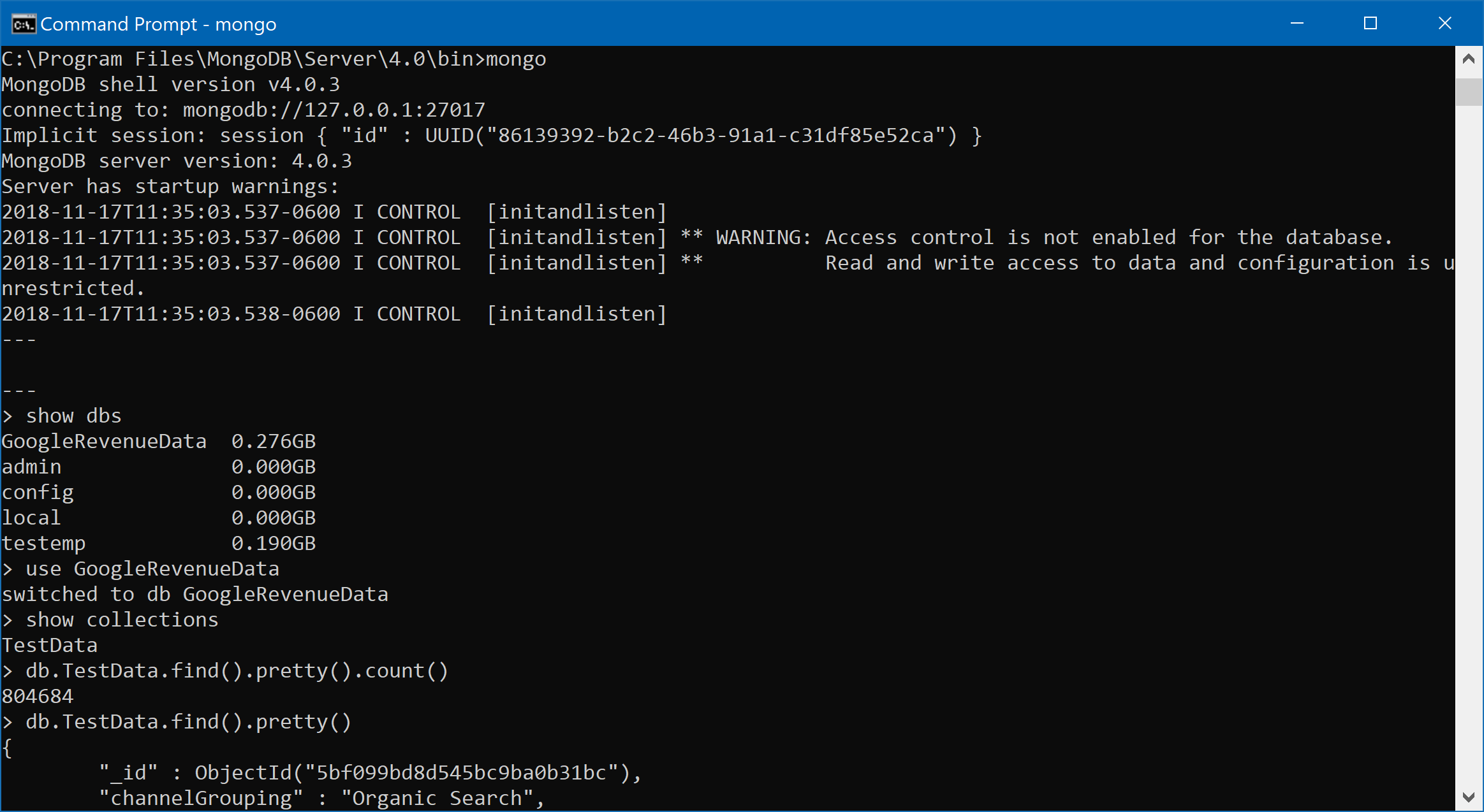
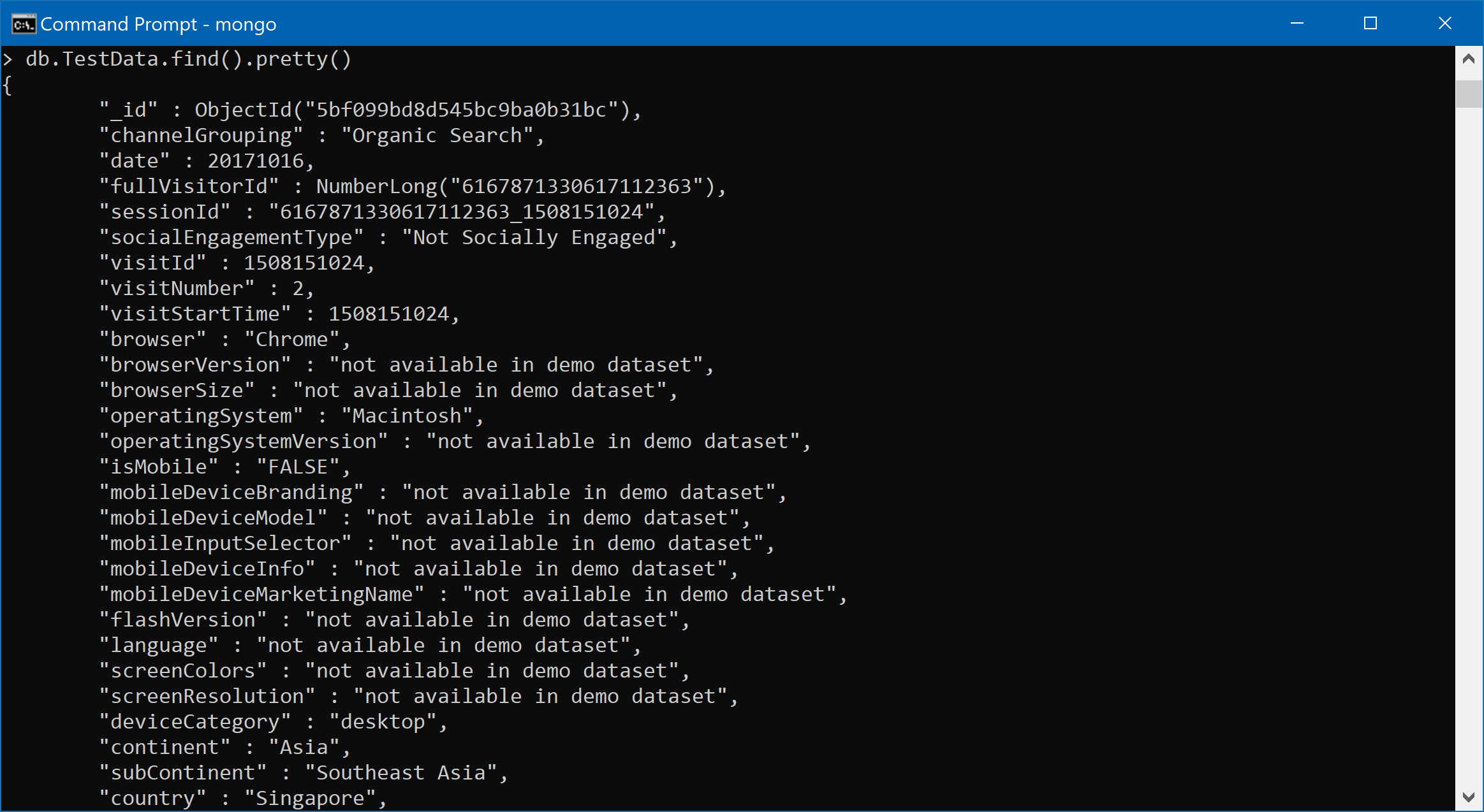


Figure 3: GoogleRevenueData after being loaded into the MongoDB

We were now able to access this data using “use GoogleRevenueData” command. Since the data is present in a JSON format, viewing the same was difficult.

This was solved by adding the “.pretty” command



# MongoDB Queries

## Query 1

### Question

### **Which user had the maximum number of visits and when?**

The Query is to identify the one visitor who has made maximum number of visits in a google store.

***The Visitor having fullVisitorId = 721049629732356293 made the maximum no: of visits = 15 on the 16th of September 2017***

### Translation:

The answer this question we need to aggregate the individuals at a date level to find the maximum number of times they visited in a day. Therefore, the “fullVisitorId” and “date” columns were chosen to be grouped under and aggregated by summing the visits at this level. In order to find the visitor with the maximum number of visits, they were sorted by the aggregated column – total in descending order. In order to isolate a single record, limit 1 was used

**MongoDB Query**:

db.TestData.aggregate([

{$group: {\_id: {fullVisitorId: "$fullVisitorId", date:"$date"}, total:{$sum:"$visits"}}},

{$sort: {total: -1}},

{$limit:1}

],

{allowDiskUse:true})

### Screen Shot of MongoDB Query and Results

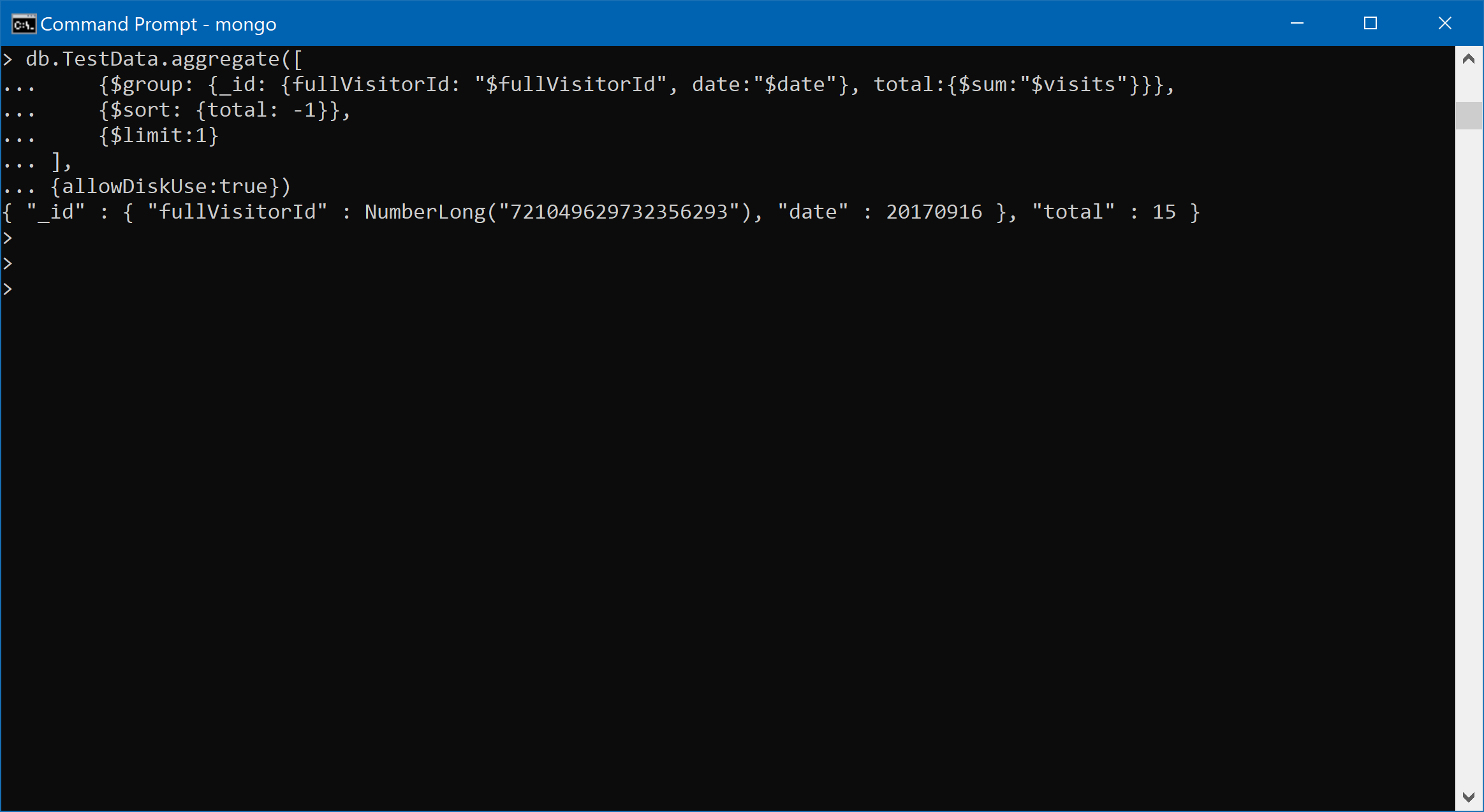


Figure 4: Question 1: MongoDB Query and Results

## Query 2

### Question

### **Is a Windows user more likely to visit the store than Macintosh user?**

The objective of this query is to compare the no: of visits made by a Windows and Macintosh user.

***Windows customer is more likely to make more visits to the store as historically between the two, Windows users made 70.7% (269,648 visits) of the visits compared to the 29.3%(111,669 visits) visits made by Macintosh users***

### Translation:

Between the different operatingSystems filter for Windows and iOS. Visits need to be aggregated at this level, therefore group them at operatingSystem level and count the number of visits.

**MongoDB Query**:

db.TestData.aggregate([

{$match:{operatingSystem: {$in: ["Windows","iOS"]}}},

{$group:{\_id: "$operatingSystem",CountOS:{$sum:1}}},

{$sort: {CountOS: -1}}

])

### Screen Shot of MongoDB Query and Results

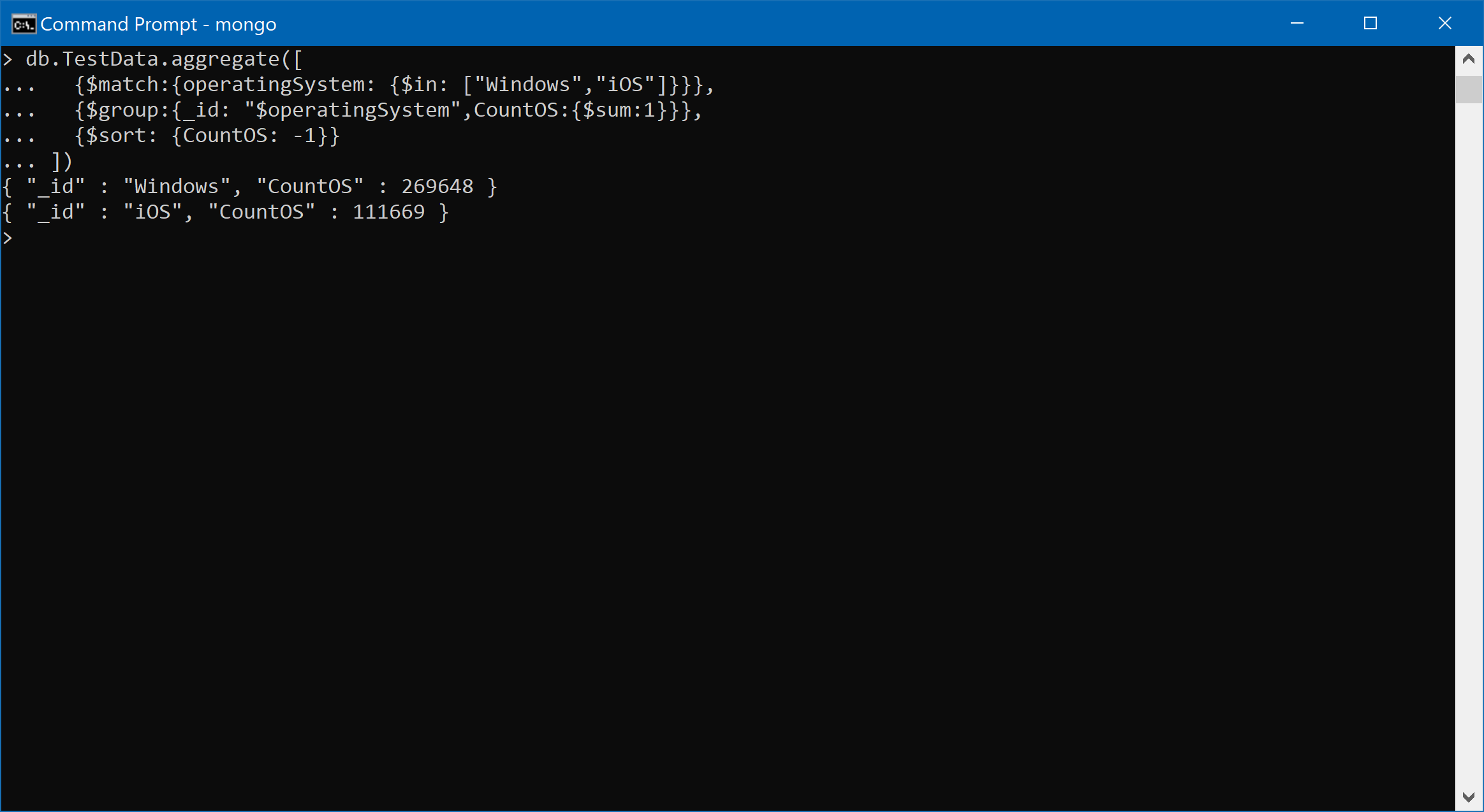


Figure 5: Question 2: MongoDB Query and Results

## Query 3

### Question

### **Which date had the least and most number of visitors with non-mobile devices?**

With this query we want to identify the dates at which the maximum and minimum number of visitors using non-mobile devices visited the store

***Maximum number of non-mobile visitors visited on 21st of March = 3305 visitors***

***Minimum number of non-mobile visitors visited on 31st Dec = 760 visitors***

### Translation:

The Query is divided into two parts – for maximum and minimum number of visitors. Initially the data needs to be filtered for non-mobile users by matching those customers where isMobile =False. The visits need to be aggregated at this level along with date. Sort by descending and limit to 1 to get the maximum number of visitors and the required date. Repeat the same but sort in ascending to get the minimum number of visitors

**MongoDB Query**:

db.TestData.aggregate([

{$match:{isMobile:"FALSE"}},

{$group:{\_id:{date:"$date",isMobile:"$isMobile"},Count:{$sum:1}}},

{$sort: {Count: 1}},

{$limit:1}

],

{allowDiskUse:true})

db.TestData.aggregate([

{$match:{isMobile:"FALSE"}},

{$group:{\_id:{date:"$date",isMobile:"$isMobile"},Count:{$sum:1}}},

{$sort: {Count: -1}},

{$limit:1}

],

{allowDiskUse:true})

### Screen Shot of MongoDB Query and Results

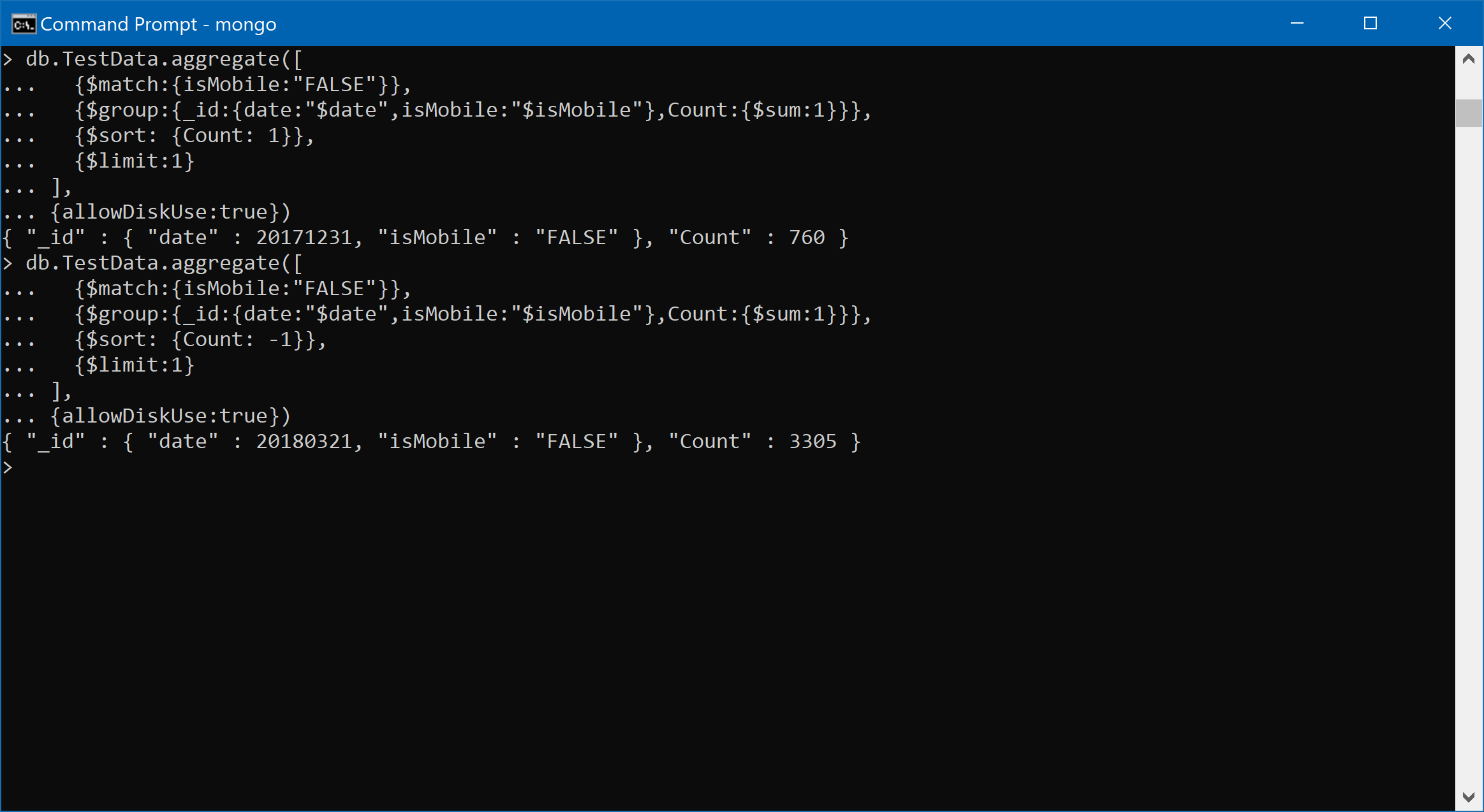


Figure 6: Question 3: MongoDB Query and Results

## Query 4

### Question

### **Were mobile devices users more socially engaged than non-mobile device users?**

The objective of this question is to identify among mobile and non-mobile users, who is more socially engaged

***We noticed that the collection contained only “not socially engaged” documents in it. We also noticed that non-mobile users were more not socially engaged – 63% in comparison to 37% of mobile users***

### Translation:

The visits need to be aggregated at isMobile and socialEngagementType level and therefore they need to be grouped at the respective levels and aggregated (count visits)

Note: “Not Socially Engaged” class was present as values within this column

**MongoDB Query**:

db.TestData.aggregate([

{$group:{\_id:{socialEngagementType:"$socialEngagementType"},Count:{$sum:1}}},

{$sort: {Count: -1}}

])

db.TestData.aggregate([

{$group:{\_id:{ismobile:"$isMobile",socialEngagementType:"$socialEngagementType"},Count:{$sum:1}}},

{$sort: {Count: -1}}

])

### Screen Shot of MongoDB Query and Results

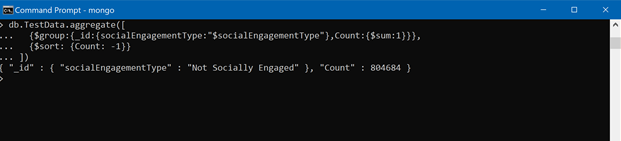


Figure 7: Question 4: MongoDB Query and Results

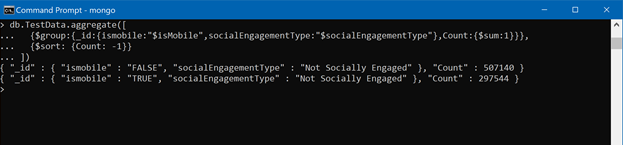


Figure 8: Question 4: MongoDB Query and Results

## Query 5

### Question

### **Provide a breakdown of store hits by region**

In this question we focus on listing all the regions by store hits

***The list of regions with descending order of store hits is shared***

***#No: of rows in the result: 374***

### Translation:

The data points having values “not available in demo dataset” and “not set” were removed from the analysis(equivalent to NULLs) using the $nin function. The remaining data points were grouped at region and hits level, aggregated at hits level. They are then sorted in descending order to view the stores regions that have maximum store hits

**MongoDB Query**:

db.TestData.aggregate([

{$match:{region:{$nin:["not available in demo dataset","(not set)"]}}},

{$group:{\_id:{region:"$region"},hits:{$sum:"$hits"}}},

{$sort: {hits: -1}}

])

### Screen Shot of MongoDB Query and Results

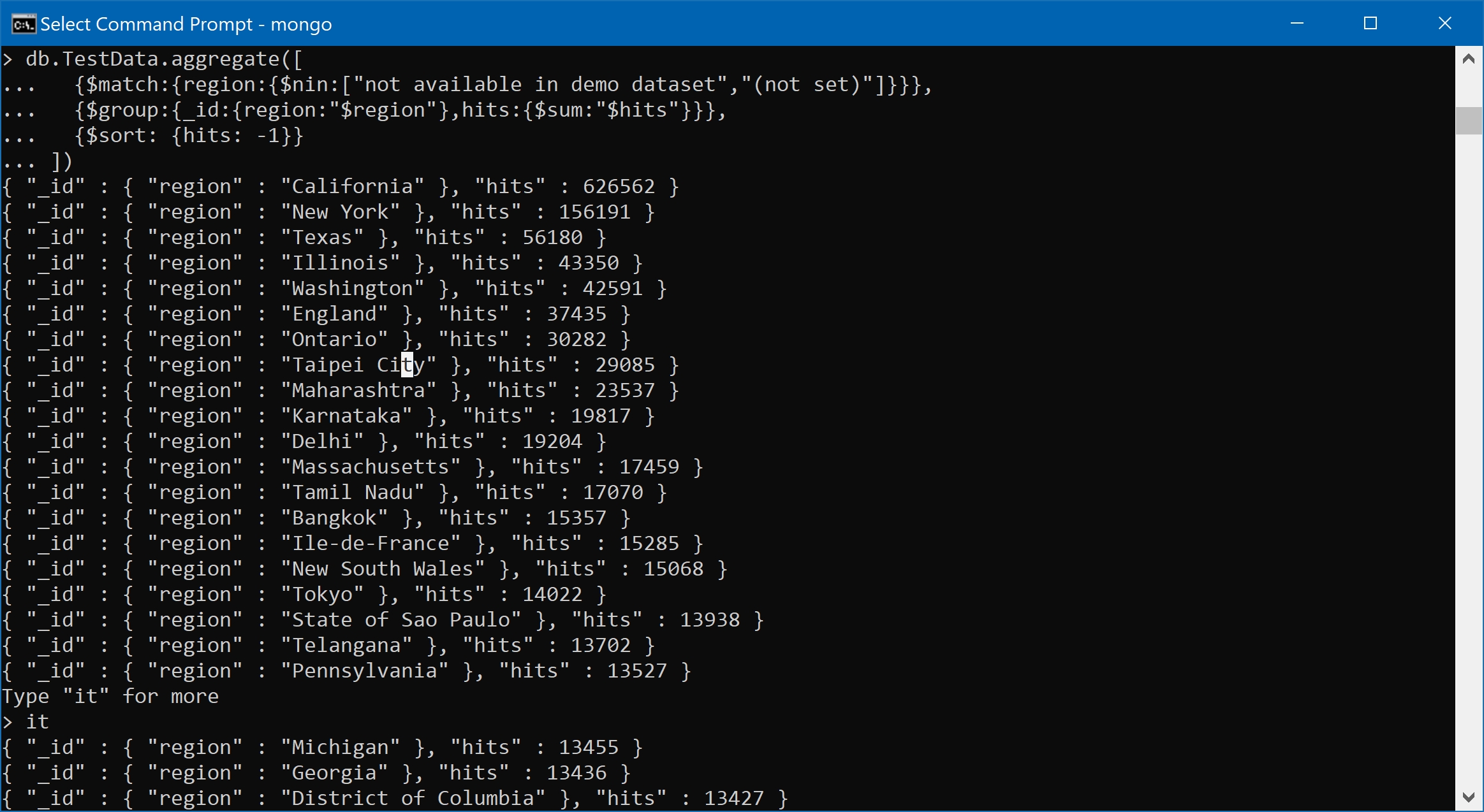


Figure 9: Question 5: MongoDB Query and Results

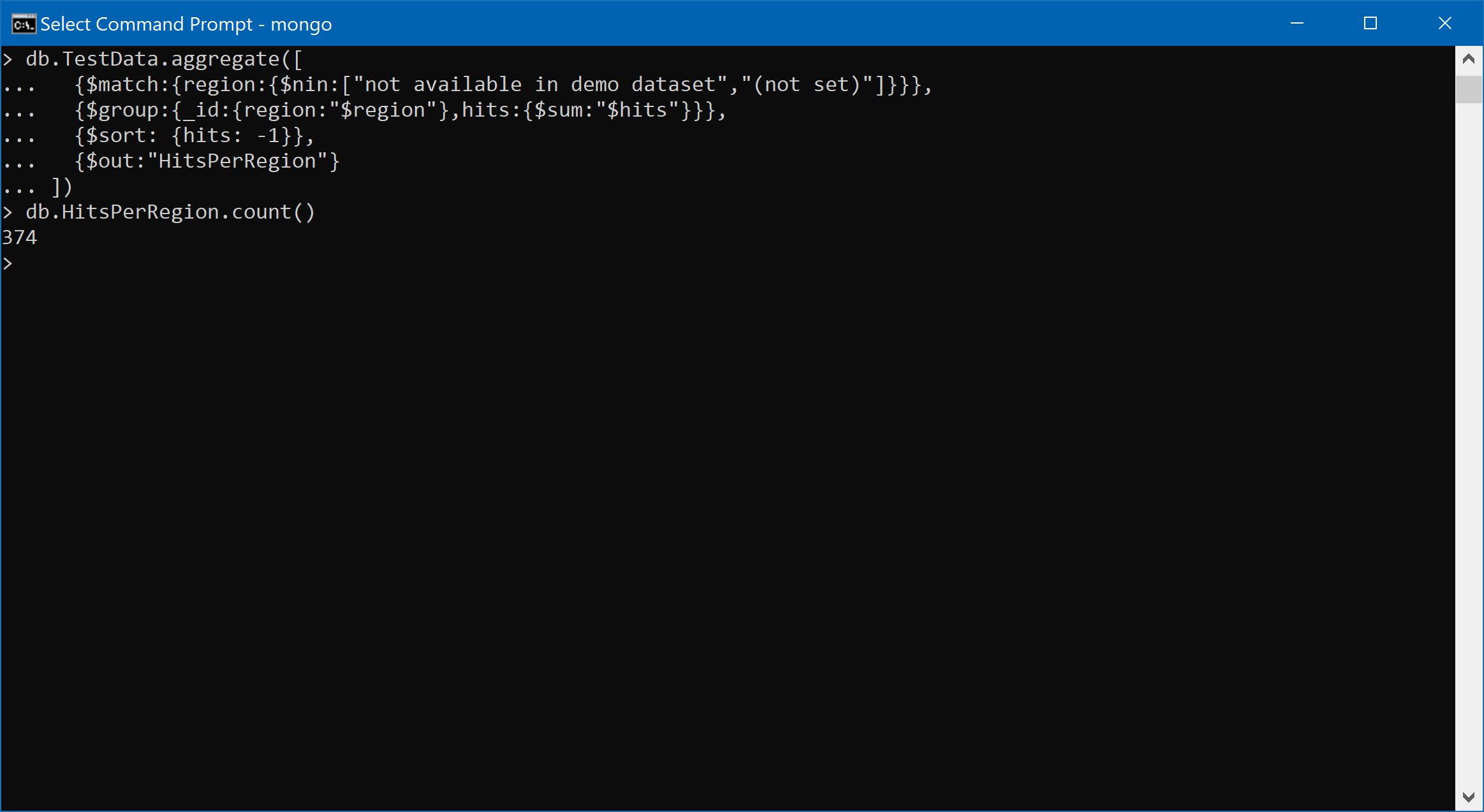


Figure 10: Question 5: MongoDB Query and Results

## Query 6

### Question

### **How many users used only Macintosh devices to visit the store?**

The objective of the question is to identify users with only Macintosh devices who visited the store

***Total of 111,669 users visited the store who had only Macintosh devices***

### Translation:

The datapoints are initially filtered for Macintosh users by matching operatingSystem for “iOS”. The remaining datapoints are then aggregated at the operatingSystem level where the no: of visitors are counted.

**MongoDB Query**:

db.TestData.aggregate([

{$match:{operatingSystem:"iOS"}},

{$group:{\_id:{operatingSystem:"$operatingSystem"},Count:{$sum:1}}}

])

### Screen Shot of MongoDB Query and Results

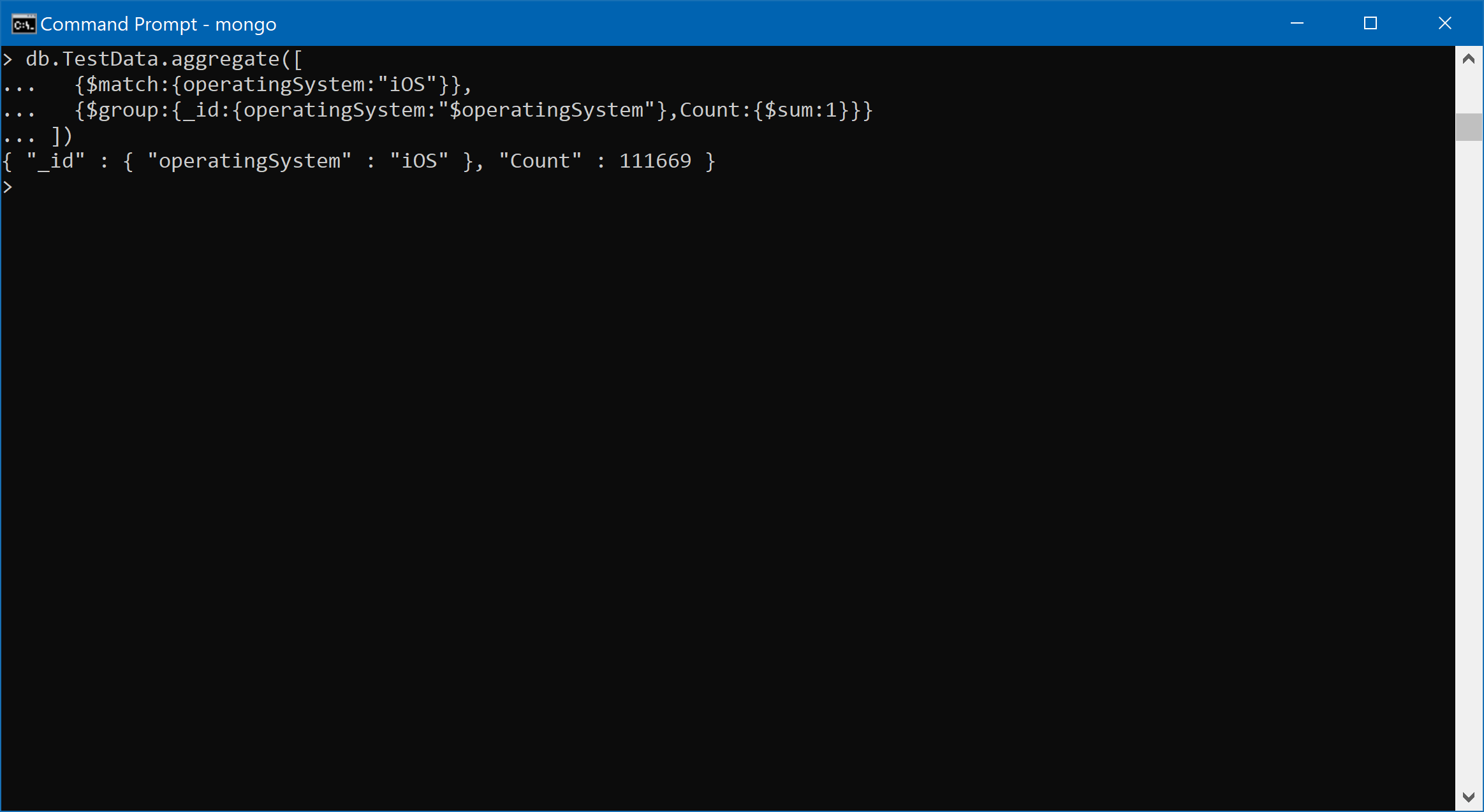


Figure 11: Question 6: Query and Results

## Query 7

### Question

### **What was the average number of hits per unique visitor?**

Here we calculate the average number of hits per every visitor (after eliminating the duplicates in the visitor’s columns)

***Average no: of hits per visitor = = = ~4 hits/customer***

### Translation:

The code is split into two parts. Firstly, we identify the number of unique customers visiting the stores

Secondly, we identify the total number of hits occurring at the store by aggregating the hits. Using these numbers, we can compute the hits/visitor which is the average value.

**No SQL Query**:

db.TestData.distinct("visitId").length

db.TestData.aggregate([

{$group:{\_id:null,Count:{$sum:"$hits"}}}

])

### Screen Shot of MongoDB Query and Results

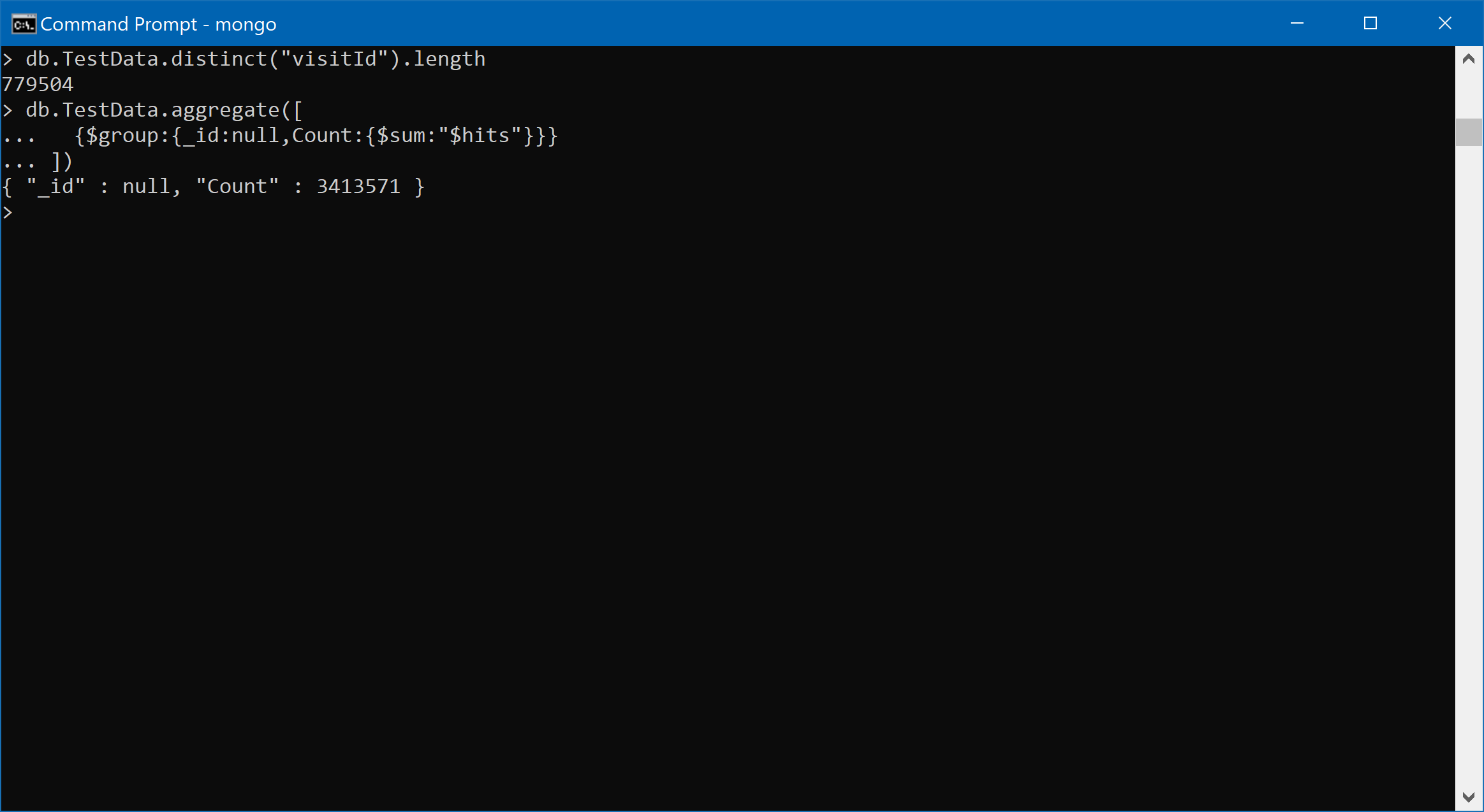


Figure 12: Question 7: Query and Results

## Query 8

### Question

### **Visitors from which country visited the store the most?**

The objective of this question is to identify the country which has the most number of visits in the google store

***United States had the highest number of visits – 352,473 visits***

### Translation:

The data needs to be aggregated for visits ( count of visits) at a country level and therefore it needs to be grouped at this level to start with. To obtain the country with the most visits , sort the data in descending order and limit to 1 to get only the highest result

**MongoDB Query**:

db.TestData.aggregate([

{$group:{\_id:"$country",Count:{$sum:1}}},

{$sort: {Count: -1}},

{$limit:1}

])

### Screen Shot of MongoDB Query and Results

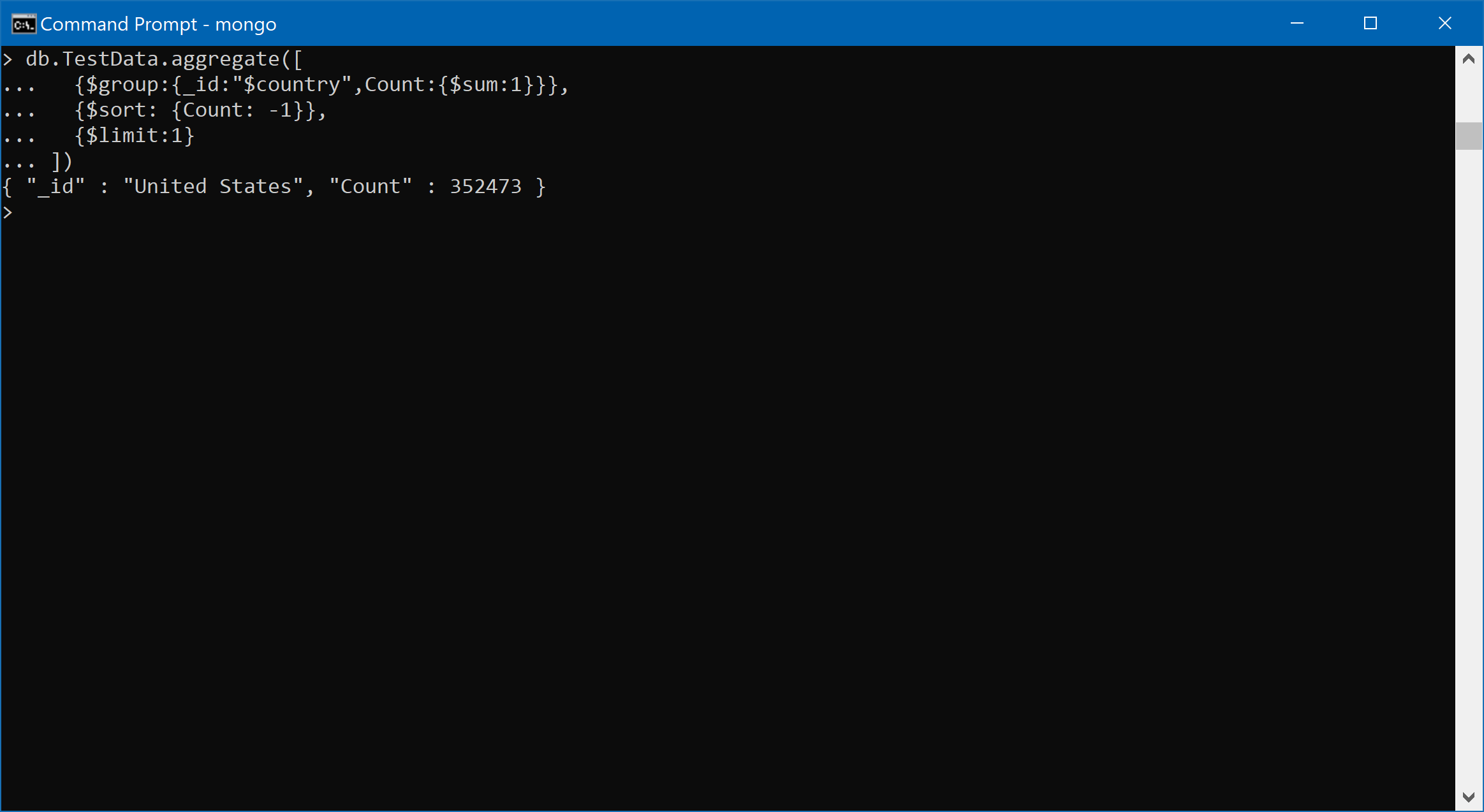


Figure 13: Question 8: MongoDB Query and Results