Group #: 14

Group Members

Vikrant Dev Rathore

Krishna Darakh

Rohit Gurjar

Ritish Gupta

**MongoDB Project – Google Store Visitor Data**

BUAN 6320.00#

Contents

[Data Review 3](#_Toc530664062)

[Assumptions/Notes About Data Collections, Attributes and Relationships between Collections 3](#_Toc530664063)

[Physical Database 4](#_Toc530664064)

[Assumptions/Notes About Data Set 4](#_Toc530664065)

[Screen shot of Physical Database objects (Database, Collections and Attributes) 4](#_Toc530664066)

[Data in the Database 4](#_Toc530664067)

[MongoDB Queries/Code 5](#_Toc530664068)

[Query 1 5](#_Toc530664069)

[Question 5](#_Toc530664070)

[Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result) 5](#_Toc530664071)

[Translation 5](#_Toc530664072)

[Screen Shot of MongoDB Query/Code and Results 5](#_Toc530664073)

[Query 2 6](#_Toc530664074)

[Question 6](#_Toc530664075)

[Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result) 6](#_Toc530664076)

[Translation 6](#_Toc530664077)

[Screen Shot of MongoDB Query/Code and Results 6](#_Toc530664078)

[Query 3 7](#_Toc530664079)

[Question 7](#_Toc530664080)

[Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result) 7](#_Toc530664081)

[Translation 7](#_Toc530664082)

[Screen Shot of MongoDB Query/Code and Results 7](#_Toc530664083)

[Query 4 8](#_Toc530664084)

[Question 8](#_Toc530664085)

[Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result) 8](#_Toc530664086)

[Translation 8](#_Toc530664087)

[Screen Shot of MongoDB Query/Code and Results 8](#_Toc530664088)

[Query 5 9](#_Toc530664089)

[Question 9](#_Toc530664090)

[Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result) 9](#_Toc530664091)

[Translation 9](#_Toc530664092)

[Screen Shot of MongoDB Query/Code and Results 9](#_Toc530664093)

[Query 6 10](#_Toc530664094)

[Question 10](#_Toc530664095)

[Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result) 10](#_Toc530664096)

[Translation 10](#_Toc530664097)

[Screen Shot of MongoDB Query/Code and Results 10](#_Toc530664098)

[Query 7 11](#_Toc530664099)

[Question 11](#_Toc530664100)

[Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result) 11](#_Toc530664101)

[Translation 11](#_Toc530664102)

[Screen Shot of MongoDB Query/Code and Results 11](#_Toc530664103)

# Data Review

## Assumptions/Notes About Data Collections, Attributes and Relationships between Collections

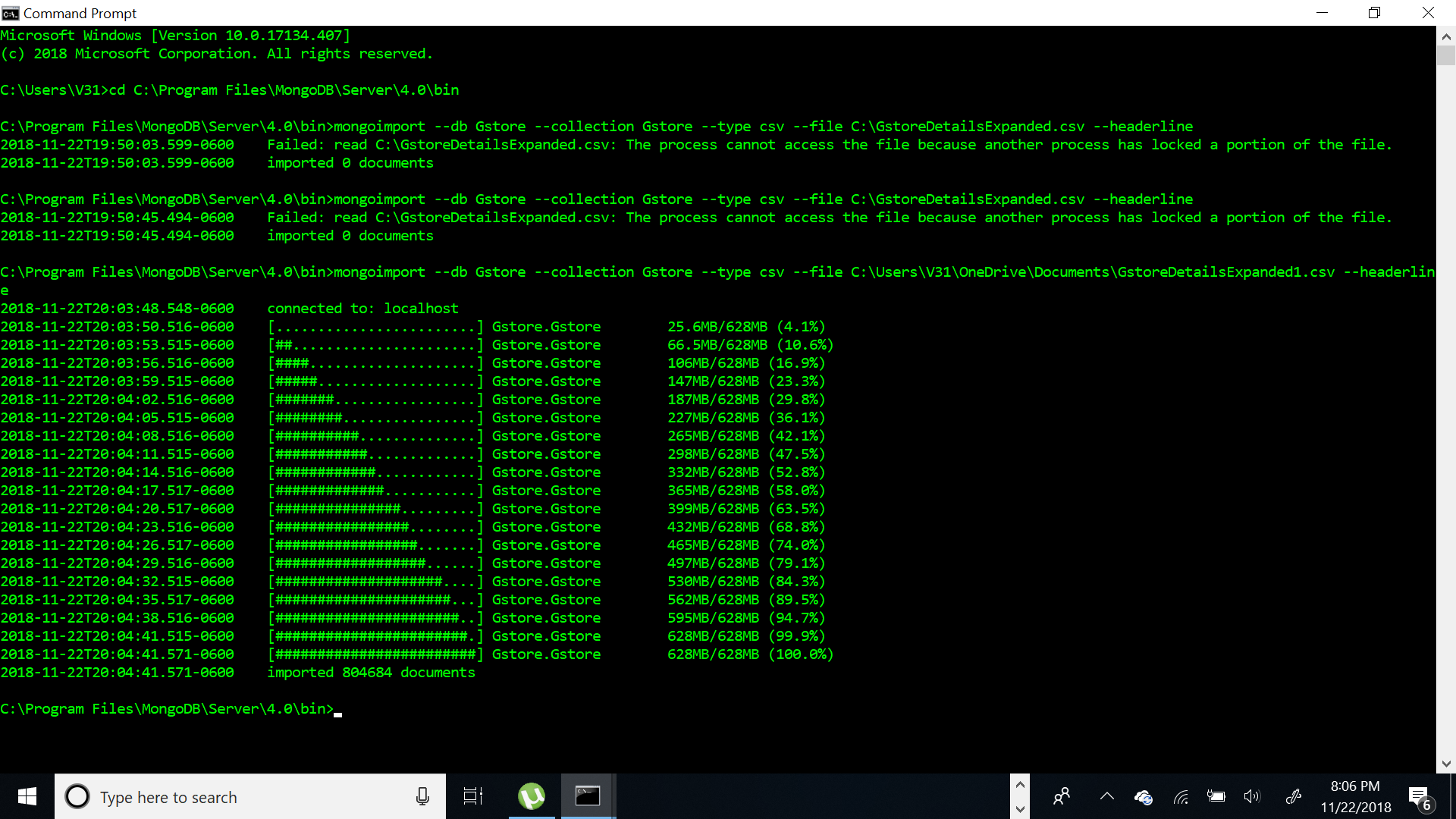
Since We are using MongoDB which can function as a no relational database, there is no need to form relationships between multiple collections and hence we uploaded the dataset into a one single collection called Gstore which contains 804684 rows. The number of distinct fullvisitorId in this collection is 617242.

For making the FullVisitorID, we were able to remove the ‘E’ values by using the LEFT command in Excel which was LEFT(A2,FIND("\_",A2)-1) where A2 is SessionID because sessionID was formed by a combination of FullVisitorID and VisitID.

The collection contains 43 fields.

The data was loaded into MongoDB using the mongoimport command:

Mongoimport --db Gstore --collection Gstore --type csv --file C:\GstoreDetailsExpanded.csv --headerline



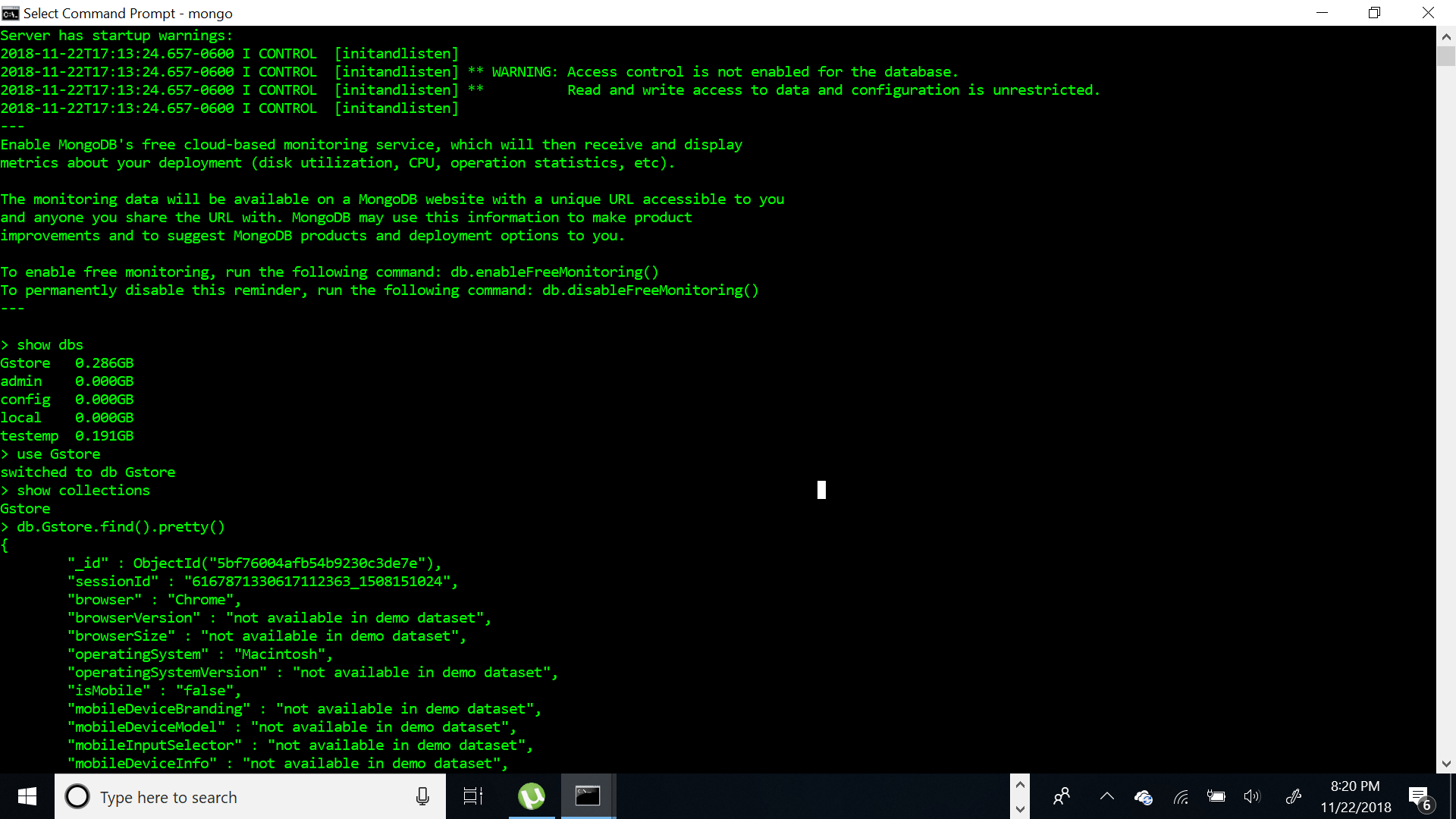
# Physical Database

## Assumptions/Notes About Data Set

For making the FullVisitorID, we were able to remove the ‘E’ values by using the LEFT command in Excel which was LEFT(A2,FIND("\_",A2)-1) where A2 is SessionID because sessionID was formed by a combination of FullVisitorID and VisitID.

## Screen shot of Physical Database objects (Database, Collections and Attributes)

## 



## Data in the Database

|  |  |  |
| --- | --- | --- |
| **Collection Name** | **Relationshps With Other Collections (if any)** | **# of Rows in Table** |
| Gstore | None | 804684 |

# MongoDB Queries/Code

## Query 1

### Question

Is a blackberry user less likely to visit the store than iOS user?

### Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result)

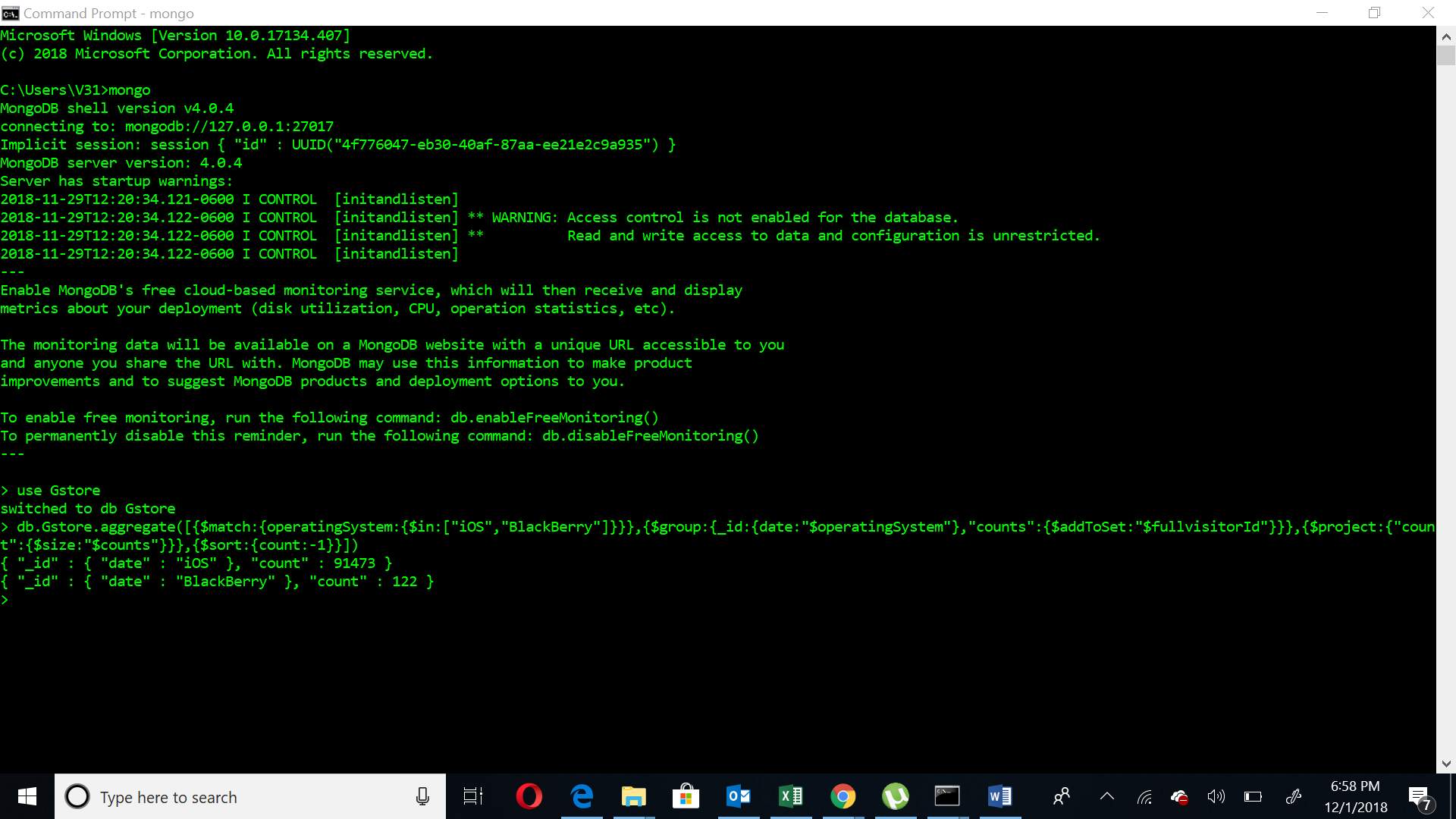
The query shows that Blackberry users(122) are less likely to visit the store in comparison to iOS users(91473). There are 2 rows in the result.

### Translation

Give the count of unique users for blackberry and iOS operating systems that visit the store.

#$match- matches the condition, $group- groups according to the field specified, $addtoset- creates a distinct array,$project-displays the fields, $size- gives size of the array,$sort- sorts in ASC(1) or DESC(-1) order, $in- Used like OR here

### Screen Shot of MongoDB Query/Code and Results

db.Gstore.aggregate([{$match:{operatingSystem:{$in:["iOS","BlackBerry"]}}},{$group:{\_id:{date:"$operatingSystem"},"counts":{$addToSet:"$fullvisitorId"}}},{$project:{"count":{$size:"$counts"}}},{$sort:{count:-1}}])

## Query 2

### Question

Which date had the most number of iOS users from Belgium?

### Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result)

There are two dates which have the most number of visitors which are 2017-08-14 and 2017-10-25(19 each). There are total 226 rows in the result.

### Translation

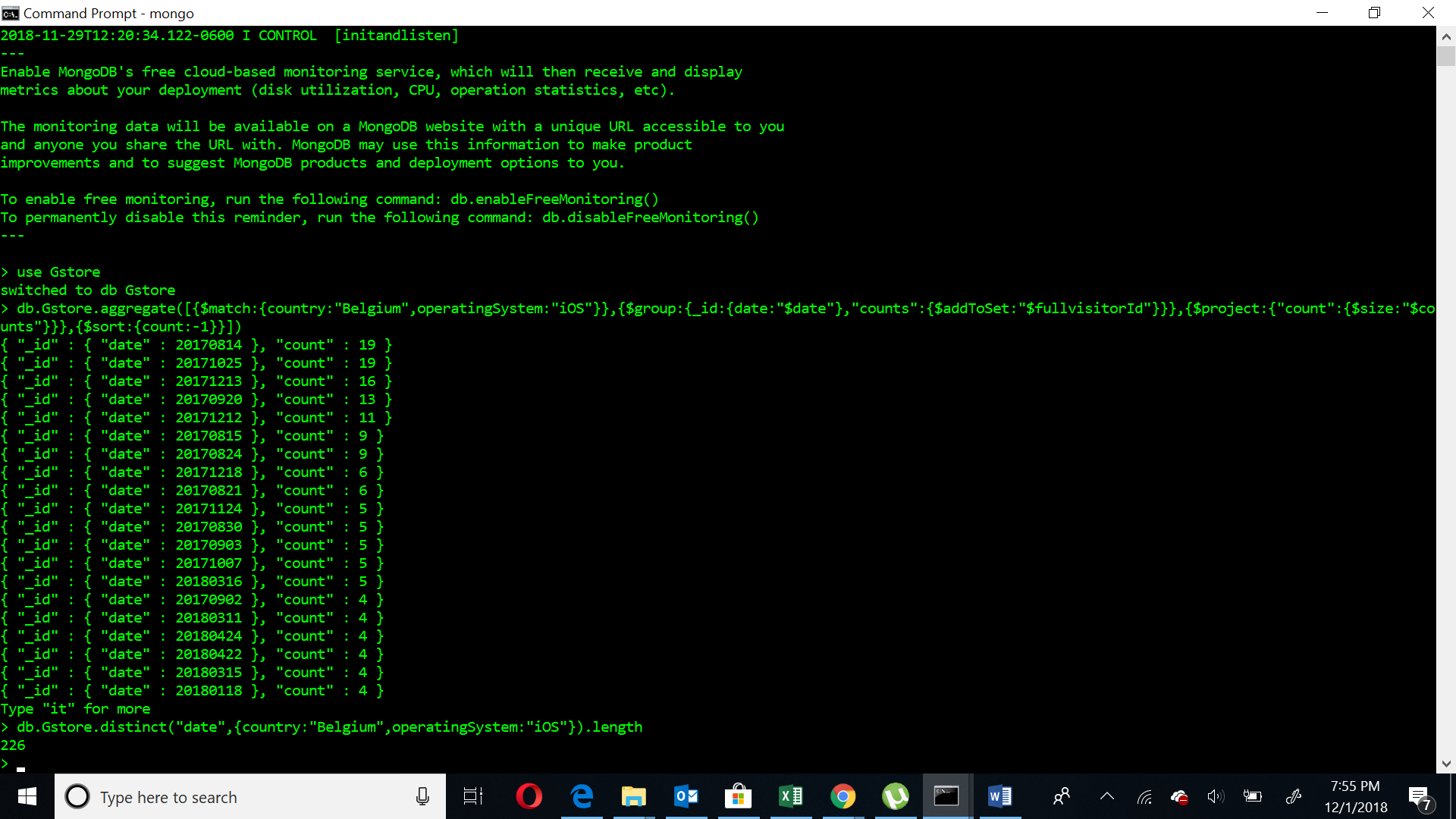
Give the count of unique users according to date which are in Belgium and use iOS operating System to visit the store.

#$match- matches the condition, $group- groups according to the field specified, $addtoset- creates a distinct array,$project-displays the fields, $size- gives size of the array,$sort- sorts in ASC(1) or DESC(-1) order.

### Screen Shot of MongoDB Query/Code and Results

db.Gstore.aggregate([{$match:{country:"Belgium",operatingSystem:"iOS"}},{$group:{\_id:{date:"$date"},"counts":{$addToSet:"$fullvisitorId"}}},{$project:{"count":{$size:"$counts"}}},{$sort:{count:-1}}])

#calculating number of result rows

db.Gstore.distinct("date",{country:"Belgium",operatingSystem:"iOS"}).length

## Query 3

### Question

Were more mobile devices (than non-mobile devices) used to visit the store?

### Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result)

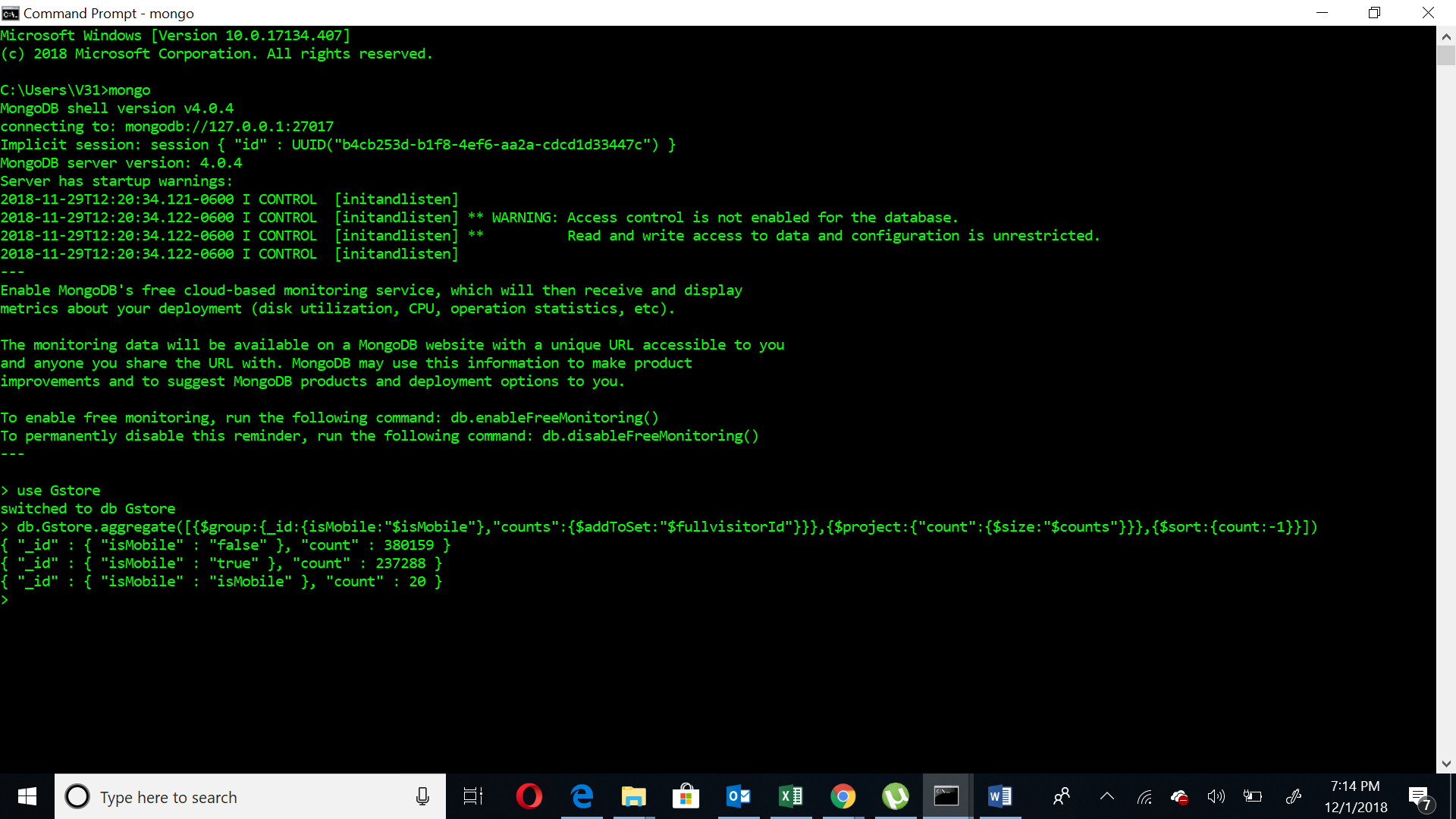
Non-mobile devices (380159) visited the store more than mobile devices (237288). A third type with Ismobile is also present (20) and it is not clear whether this represents mobile devices or non mobile devices. The result contains 3 rows.

### Translation

Show the count of mobile and non-mobile devices that access the store.

$group- groups according to the field specified, $addtoset- creates a distinct array,$project-displays the fields, $size- gives size of the array,$sort- sorts in ASC(1) or DESC(-1) order.

### Screen Shot of MongoDB Query/Code and Results

db.Gstore.aggregate([{$group:{\_id:{isMobile:"$isMobile"},"counts":{$addToSet:"$fullvisitorId"}}},{$project:{"count":{$size:"$counts"}}},{$sort:{count:-1}}])

## 

## Query 4

### Question

Provide a breakdown of store pageviews by city

### Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result)

The breakdown of pageviews has been given by city in the result(732 rows in total ). It is not clear whether “not available in demo dataset” and “(not set)” are to be excluded.

### Translation

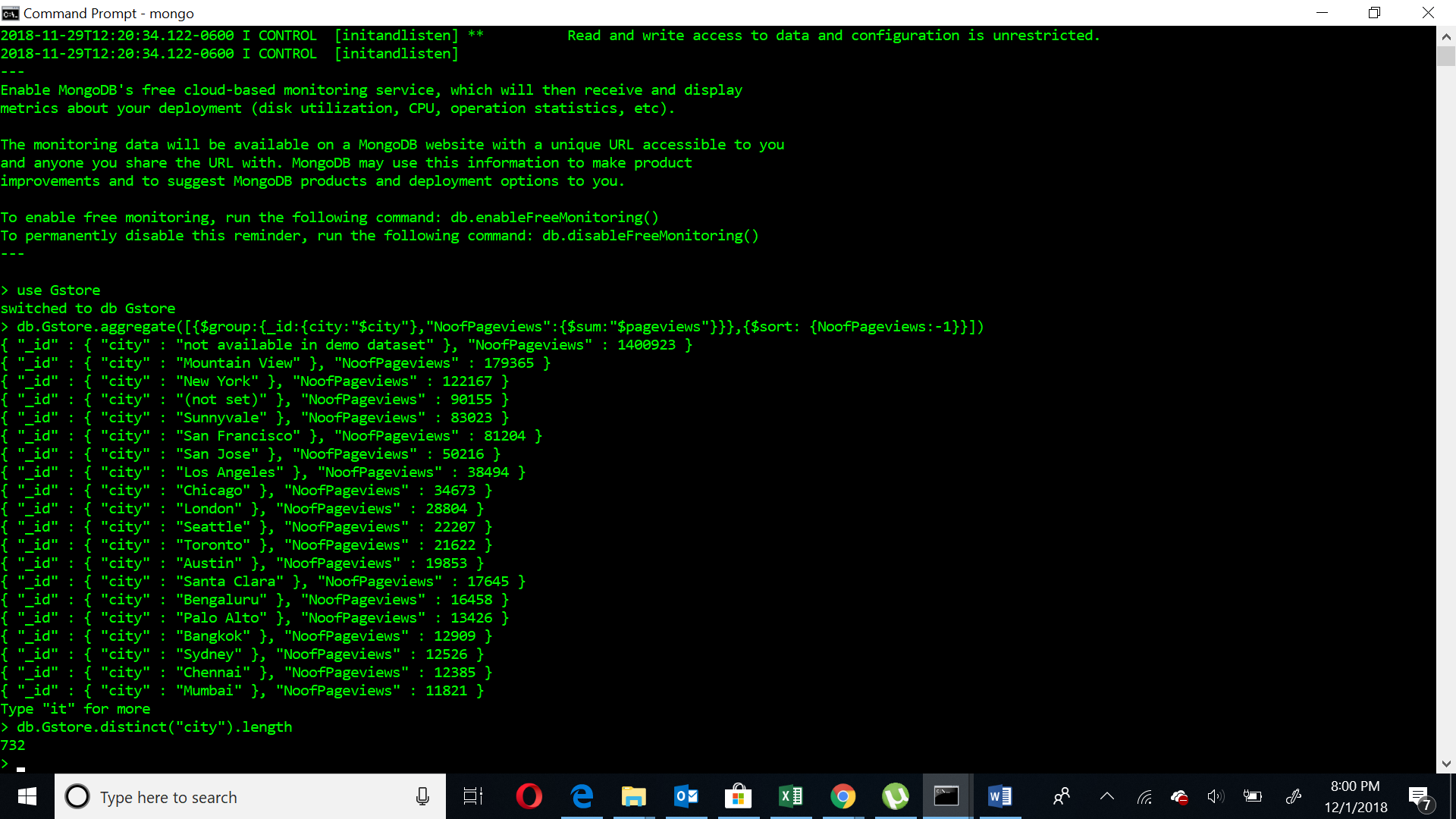
Give the count of pageviews by each city.

$group- groups according to the field specified, $sum- Gives sum,$sort- sorts in ASC(1) or DESC(-1) order.

### Screen Shot of MongoDB Query/Code and Results

db.Gstore.aggregate([{$group:{\_id:{city:"$city"},"NoofPageviews":{$sum:"$pageviews"}}},{$sort: {NoofPageviews:-1}}])

#calculating number of result rows

db.Gstore.distinct("city").length

## 

## Query 5

### Question

How many users used only Windows devices to visit the store?

### Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result)

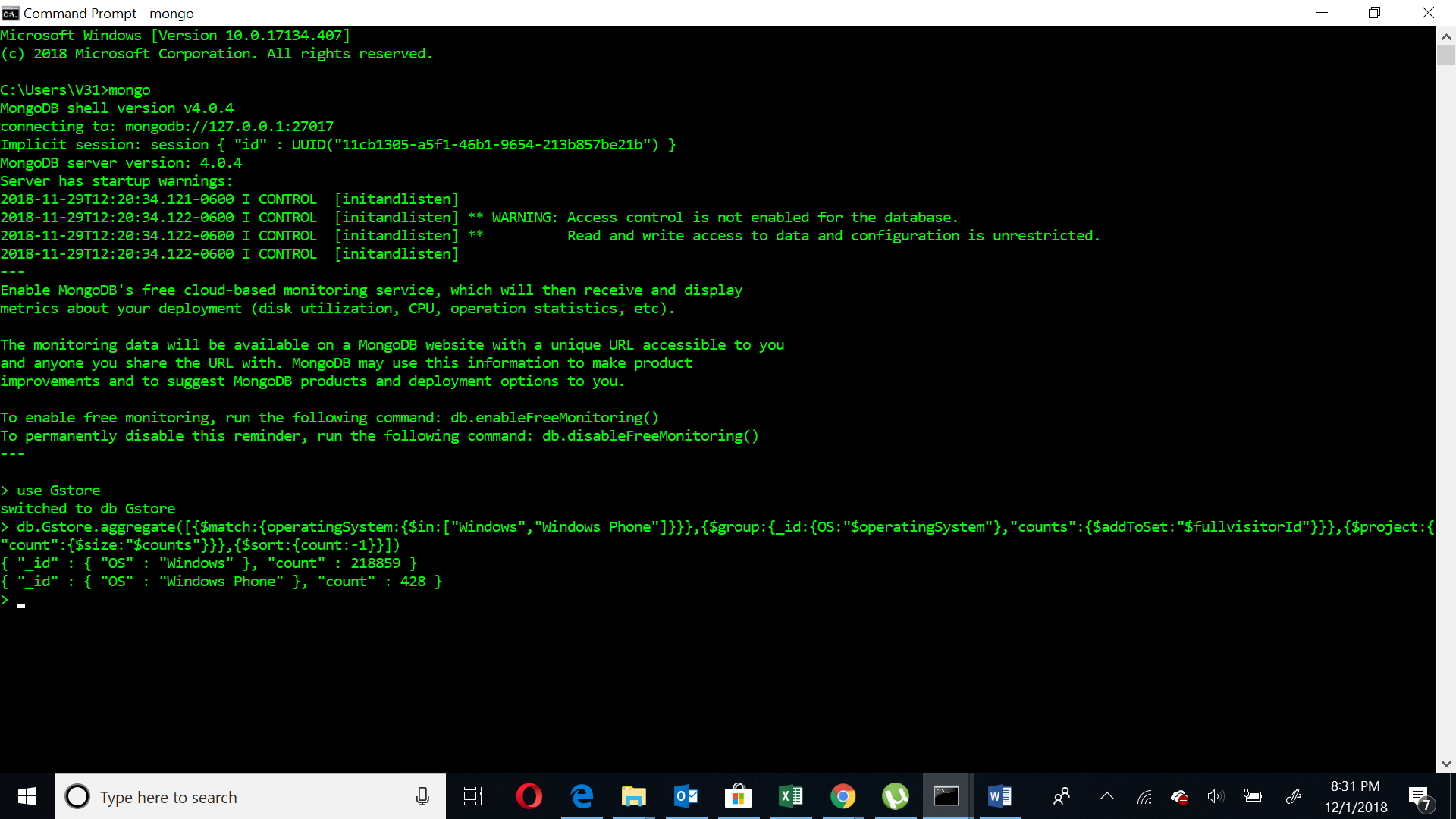
Since the question says devices, therefore all Windows OS Should be included which Windows OS and Windows Phone OS are. We see that 218859 users used windows OS to access the store and 428 users used Windows Phone OS to access the store (Total 219287 users). The result contains 2 rows.

### Translation

Show the count of distinct users that used windows OS and Windows Phone OS.

#$match- matches the condition, $group- groups according to the field specified, $addtoset- creates a distinct array,$project-displays the fields, $size- gives size of the array,$sort- sorts in ASC(1) or DESC(-1) order, $in- Used like OR here

### Screen Shot of MongoDB Query/Code and Results

db.Gstore.aggregate([{$match:{operatingSystem:{$in:["Windows","Windows Phone"]}}},{$group:{\_id:{OS:"$operatingSystem"},"counts":{$addToSet:"$fullvisitorId"}}},{$project:{"count":{$size:"$counts"}}},{$sort:{count:-1}}])

## Query 6

### Question

### How many visitors had zero pageviews?

### Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result)

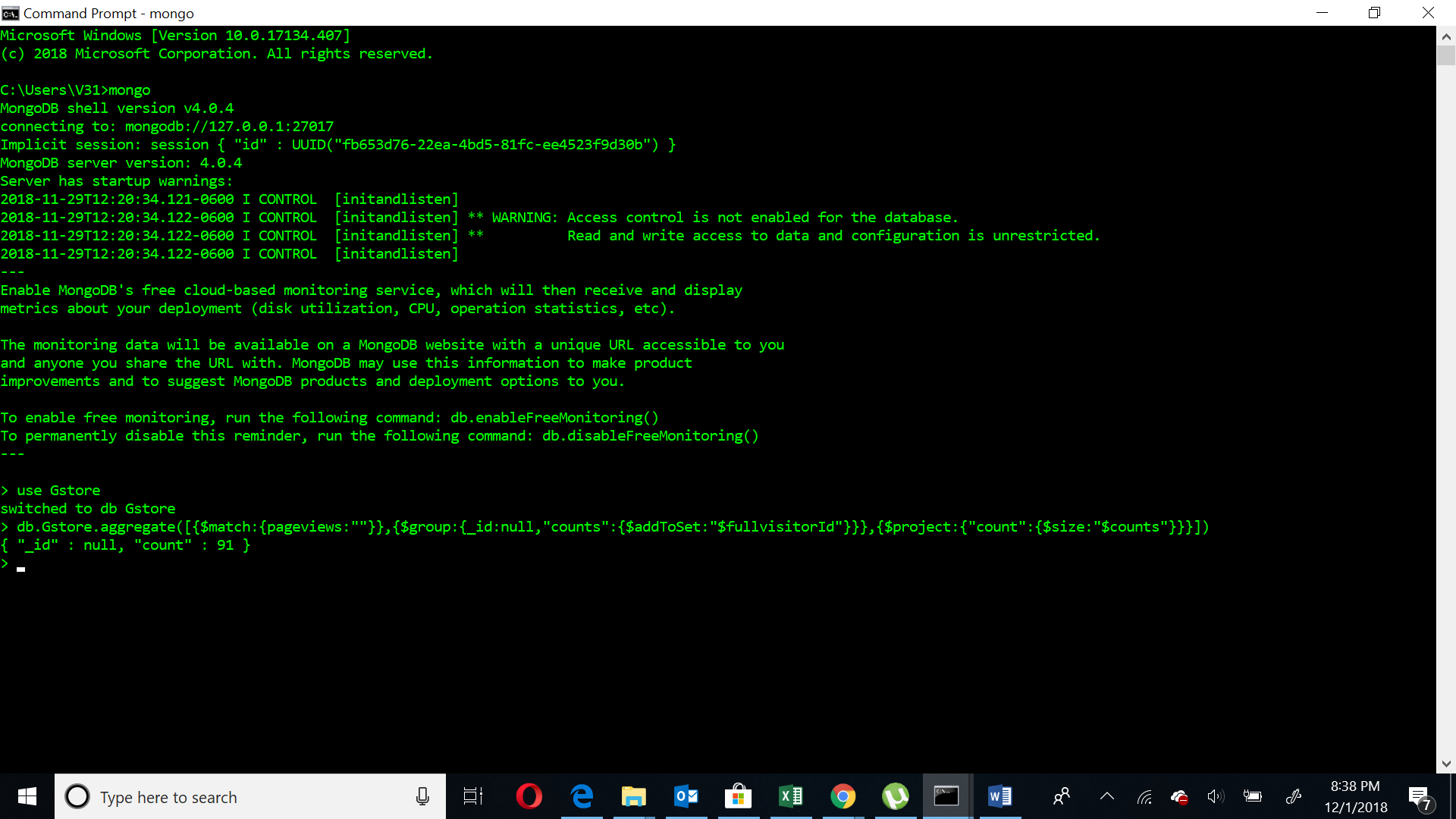
If we consider the records that are blank to be zero and calculating then 91 users had zero pageviews. The result contains 1 row.

### Translation

Find the distinct users that zero pageviews.

#$match- matches the condition, $group- groups according to the field specified, $addtoset- creates a distinct array, $size- gives size of the array.

### Screen Shot of MongoDB Query/Code and Results

db.Gstore.aggregate([{$match:{pageviews:""}},{$group:{\_id:null,"counts":{$addToSet:"$fullvisitorId"}}},{$project:{"count":{$size:"$counts"}}}])

## Query 7

### Question

Which city (other than unknown) had the most number of desktop users?

### Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result)

We can see Mountain View had the maximum desktop users(16756). The result contains 730 rows. In this case we consider both “not available in dataset” and “(not set)” are considered as unknown.

### Translation

Show the count of distinct desktop users for the city which has maximum desktop users excluding unknown.

#$match- matches the condition, $group- groups according to the field specified, $addtoset- creates a distinct array, $project-displays the fields, $size- gives size of the array,$sort- sorts in ASC(1) or DESC(-1) order,$nin- Not In

### Screen Shot of MongoDB Query/Code and Results

db.Gstore.aggregate([{$match:{city:{$nin:["not available in demo dataset","(not set)"]},deviceCategory:"desktop"}},{$group:{\_id:{city:"$city"},"counts":{$addToSet:"$fullvisitorId"}}},{$project:{"count":{$size:"$counts"}}},{$sort:{count:-1}}])

# calculating number of result rows

db.Gstore.distinct("city",{city:{$nin:["not available in demo dataset","(not set)"]}}).length