Group 03

Group Members-

Aishwarya Badlani

Apurva Joshi

Shruti Rajani

Swathi Sridharan

Sneha Ganjam

**MongoDB Project – Google Store Visitor Data**

BUAN 6320.003

Contents

[Data Review 3](#_Toc530664062)

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

[Physical Database 3](#_Toc530664064)

[Assumptions/Notes About Data Set 3](#_Toc530664065)

[Screen shot of Physical Database objects (Database, Collections and Attributes) 3](#_Toc530664066)

[Data in the Database 3](#_Toc530664067)

[MongoDB Queries/Code 4](#_Toc530664068)

[Query 1 4](#_Toc530664069)

[Question 4](#_Toc530664070)

[Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result) 4](#_Toc530664071)

[Translation 4](#_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 7](#_Toc530664078)

[Query 3 18](#_Toc530664079)

[Question 18](#_Toc530664080)

[Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result) 18](#_Toc530664081)

[Translation 18](#_Toc530664082)

[Screen Shot of MongoDB Query/Code and Results 18](#_Toc530664083)

[Query 4 8](#_Toc530664084)

[Question 8](#_Toc530664085)

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

[Translation 9](#_Toc530664087)

[Screen Shot of MongoDB Query/Code and Results 9](#_Toc530664088)

[Query 5 10](#_Toc530664089)

[Question 10](#_Toc530664090)

[Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result) 10](#_Toc530664091)

[Translation 10](#_Toc530664092)

[Screen Shot of MongoDB Query/Code and Results 11](#_Toc530664093)

[Query 6 12](#_Toc530664094)

[Question 12](#_Toc530664095)

[Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result) 12](#_Toc530664096)

[Translation 12](#_Toc530664097)

[Screen Shot of MongoDB Query/Code and Results 13](#_Toc530664098)

[Query 7 14](#_Toc530664099)

[Question 14](#_Toc530664100)

[Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result) 14](#_Toc530664101)

[Translation 14](#_Toc530664102)

[Screen Shot of MongoDB Query/Code and Results 15](#_Toc530664103)

Query 8………………………………………………………………………………………………………………………………………………16

Question………………………………………………………………………………………………………………………………………..16

Notes/Comments About MongoDB Query/Code and Results (Include # of Rows in Result)……………16

Translation…………………………………………………………………………………………………………………………………… 16

[Screen](#_Toc530664103) Shot of Mongo DB Query/ Code and Results………………………………………………………………………17

Data Review

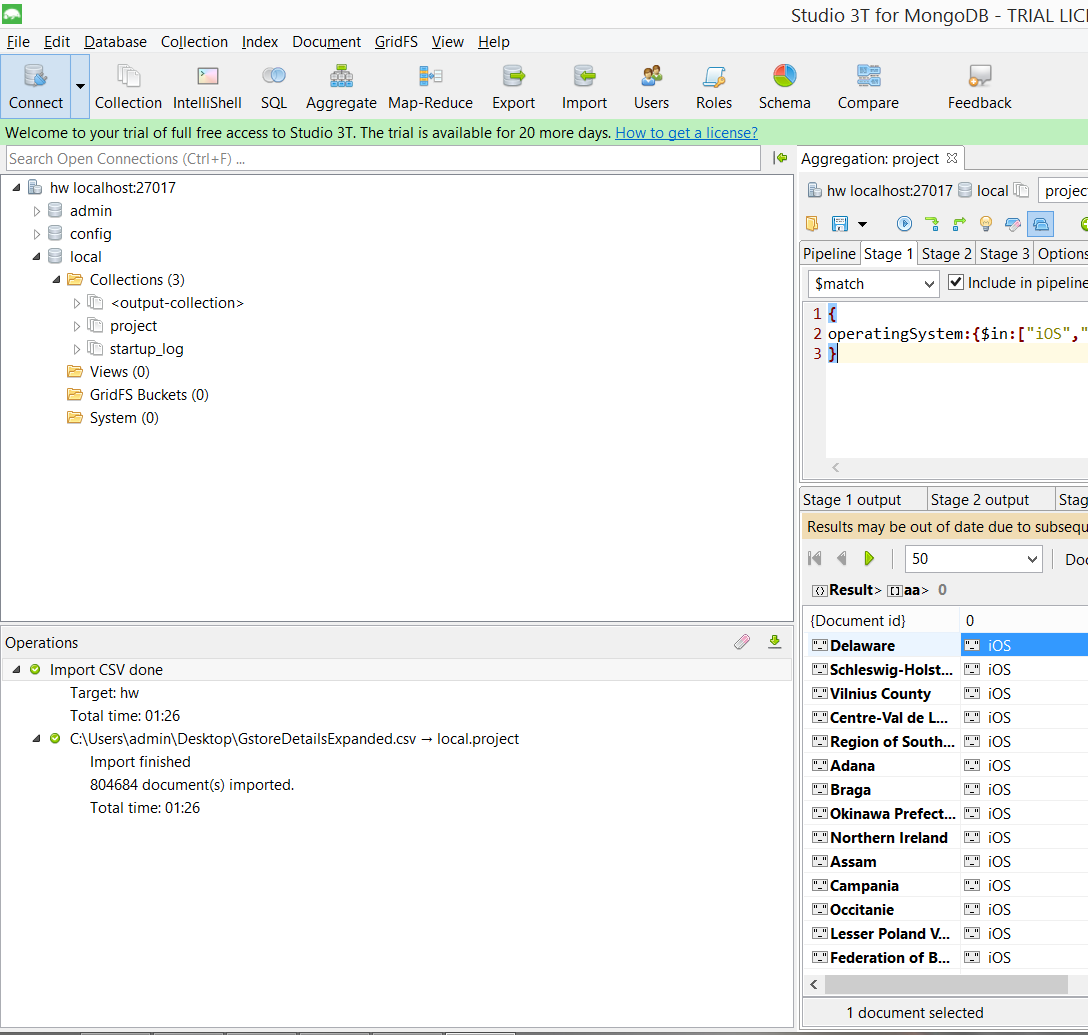
## Assumptions/Notes About Data Collections, Attributes and Relationships between Collections

## Physical Database

## Assumptions/Notes About Data Set

Collection name - "project" containing GStoreData is imported with 804684 documents

## 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** |
| project | NA | 804684 |

# MongoDB Queries/Code

## **Query 1**

### Question

Which user had the minimum number of visits and when?

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

Code:

db.getCollection**(**"project"**)**.aggregate**(**

**[**

**{**

"$group" **:** **{**

"\_id" **:** "$fullVisitorID"**,**

"count" **:** **{**

"$sum" **:** 1.0

**},**

"dateofvisit" **:** **{**

"$push" **:** "$date"

**}**

**}**

**},**

**{**

"$sort" **:** **{**

"count" **:** 1.0

**}**

**},**

**{**

"$match" **:** **{**

"count" **:** 1.0

**}**

**}**

**]**

**);**

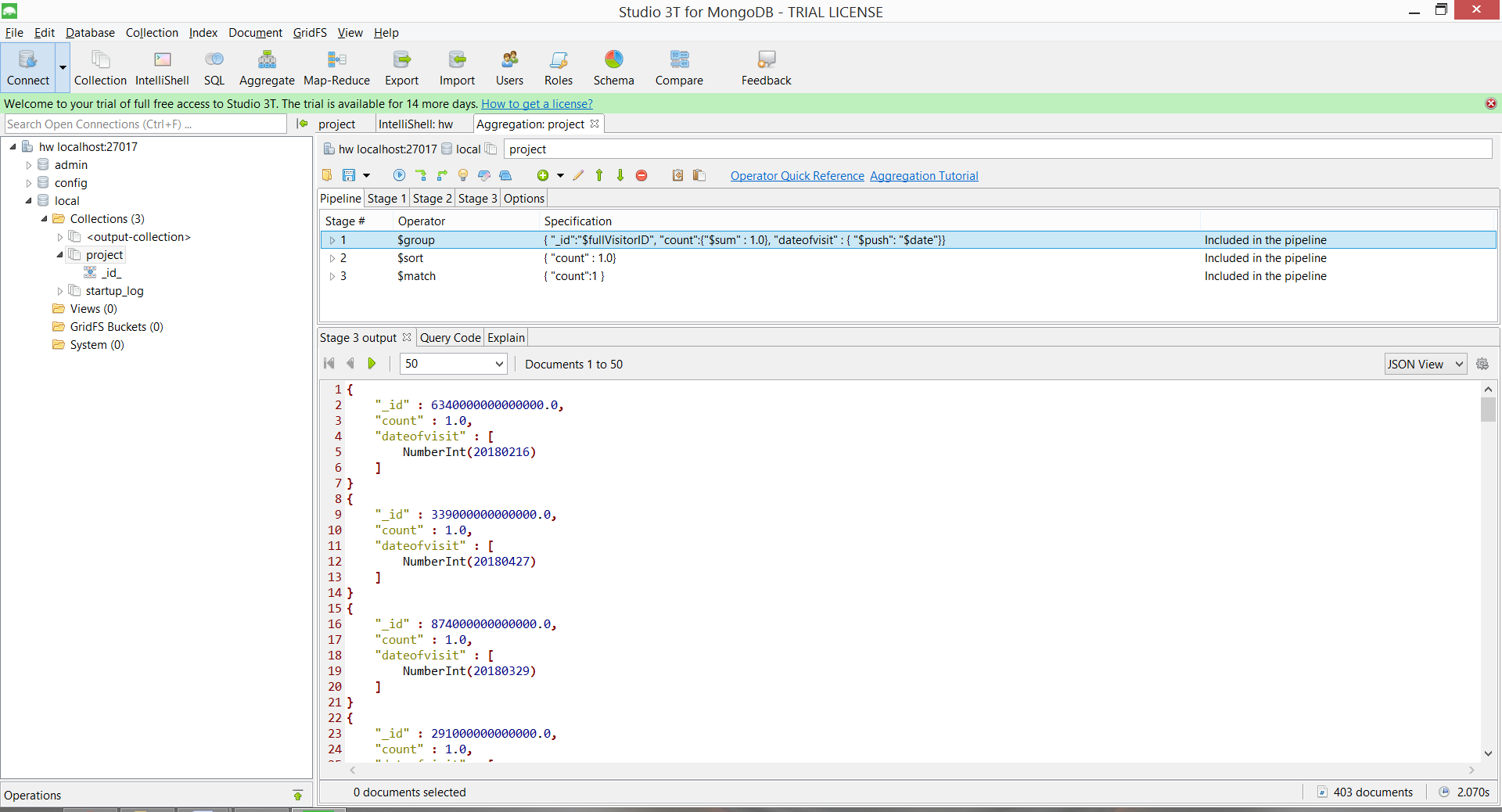
Result:

**#403 results with date of visit count=1**

### Translation

1. Group the data based on Visitor ID and date of visit and find the visitors with minimum number of visits and the when it was made.
2. Group the data based on fullVisitorId and dateofvisit columns. Count number of visit made by each visitor and date associated with visit.
3. $group : {“$fullVisitorID”, “dateofvisit”} and then “$sort” : {“count”:1.0} "$match" **:** {"count" **:** 1.0}

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



## **Query 2**

### Question

### Which operating system (devices) was the most popular amongst store visitors with mobile devices?

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

### Code:

db.getCollection**(**"project"**)**.aggregate**(**

**[**

**{**

"$match" **:** **{**

"isMobile" **:** **{**

"$in" **:** **[**

"true"

**]**

**}**

**}**

**},**

**{**

"$group" **:** **{**

"\_id" **:** "$operatingSystem"**,**

"count" **:** **{**

"$sum" **:** 1.0

**}**

**}**

**},**

**{**

"$sort" **:** **{**

"count" **:** -1.0

**}**

**},**

**{**

"$limit" **:** 1.0

**}**

**]**

**);**

### Result:

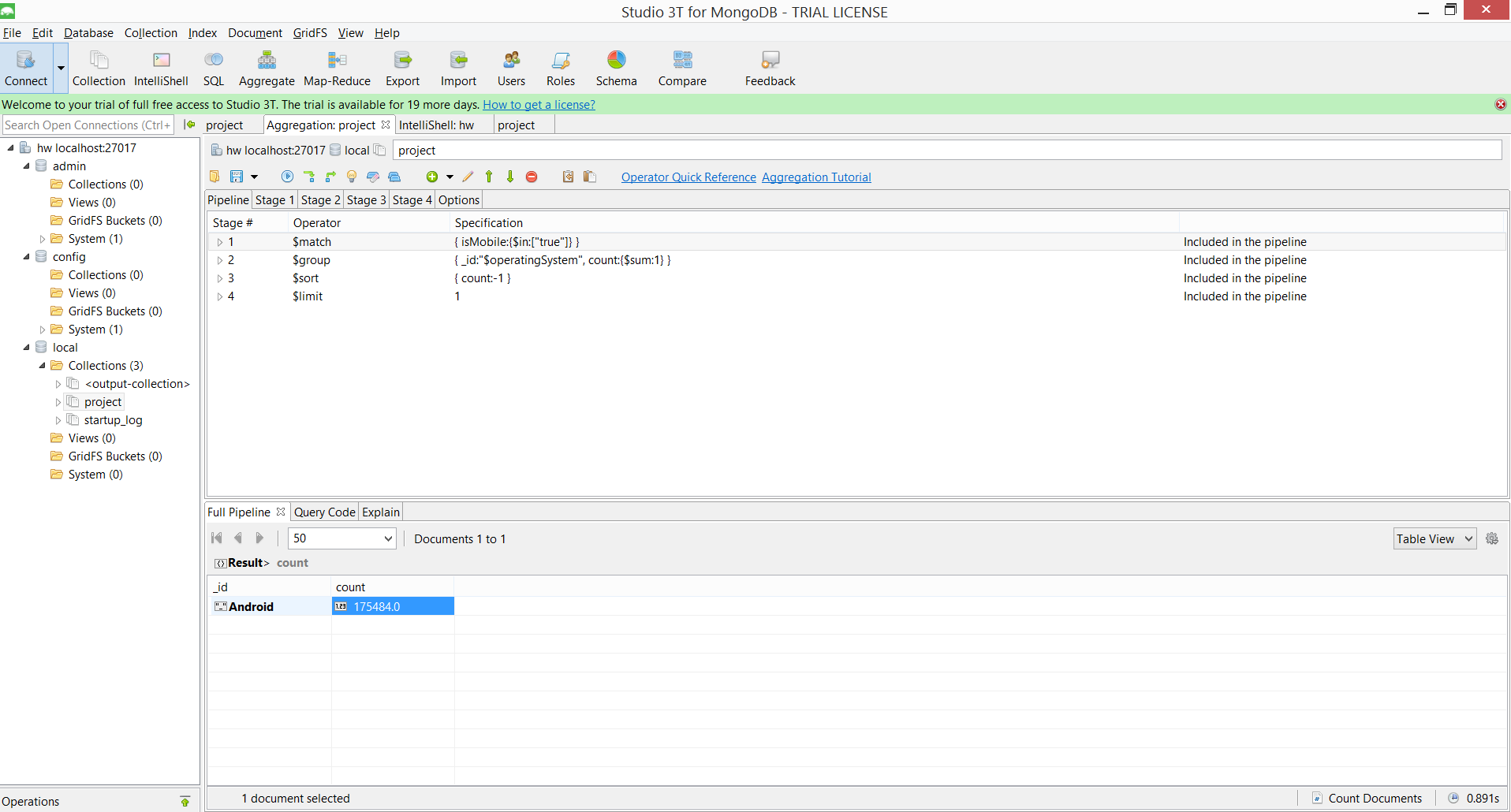
### **1 row returned with result. Android is most popular operating system.**

### Translation

1. Filter data for mobile device users. Group by operating system and count number of users for each operating system. Sort by number of users of each operating system. Obtain operating system with maximum users.
2. Filter data with isMobile is true. Then group the result by operating system and count the occurrence of each operating system. Finally sort the count of operating system in descending order and obtain the first row.
3. Match isMobile is true. Group with id operating system and count with sum :1. Sort with count: -1. Limit:1
4. “$match” : “isMobile” : {“$in” : [ “true”]}, “$group”: {“\_id” : “$operatingSystem”, “count”:{$sum:1}}. “$sort:{“count” : -1.0} “$limit” : 1.0

### 

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



## **Query 4**

### Question

### Users of which operating system were the least socially engaged?

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

### Code:

db.getCollection**(**"project"**)**.aggregate**(**

**[**

**{**

"$match" **:** **{**

"socialEngagementType" **:** "Socially Engaged"

**}**

**},**

**{**

"$group" **:** **{**

"\_id" **:** "$operatingSystem"**,**

"count" **:** **{**

"$sum" **:** 1.0

**}**

**}**

**},**

**{**

"$sort" **:** **{**

"count" **:** 1.0

**}**

**},**

**{**

"$limit" **:** 1.0

**}**

**],**

**{**

"allowDiskUse" **:** false

**}**

**);**

### Note:

### **#THE DATASET IMPORTED HAVE ALL VALUE IN SOCIALENGAGEMENTTYPE as "Not Socially Engaged"**

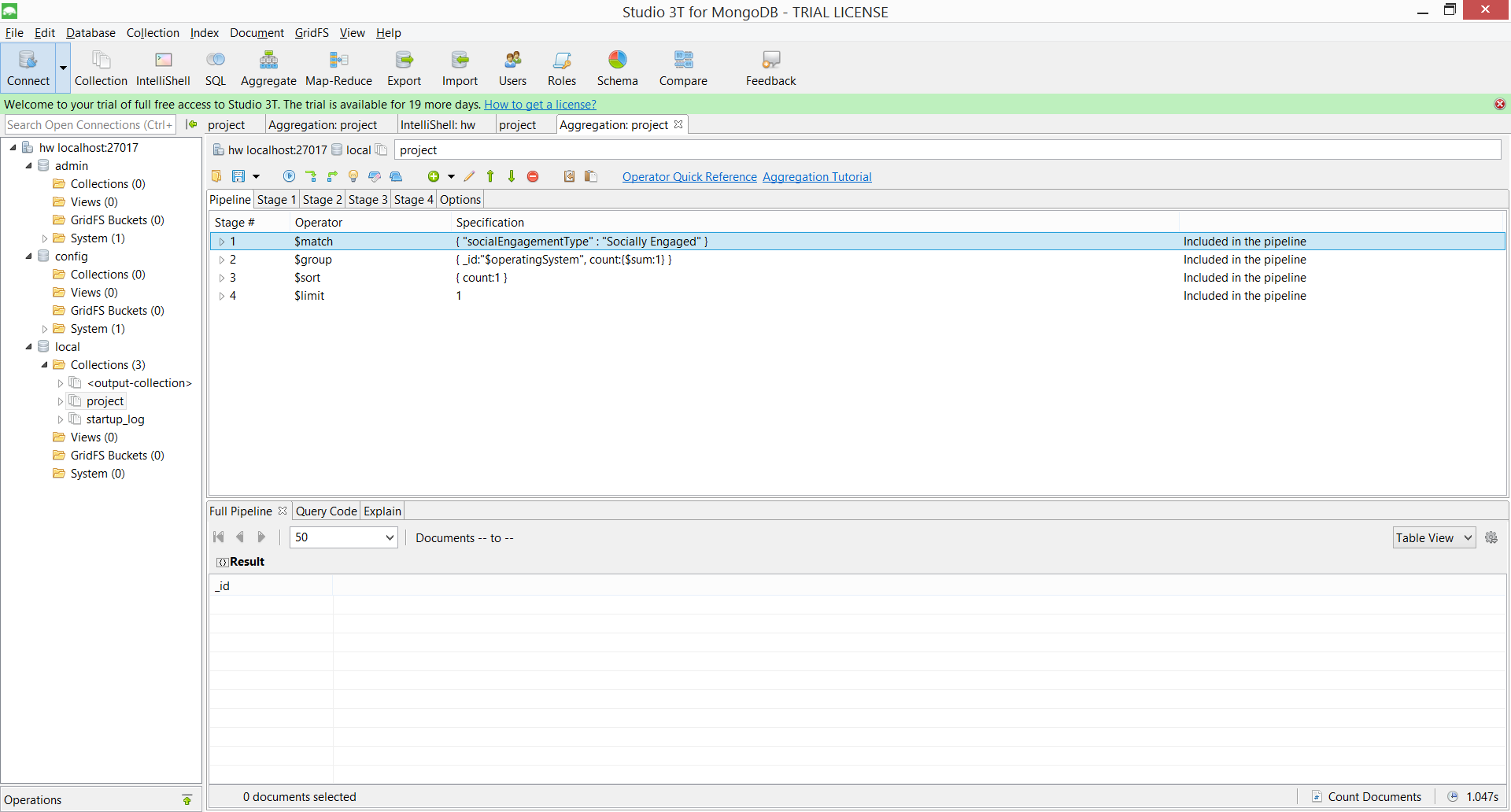
### Result:

**0 rows returned.**

### Translation

1. Filter users who are socially engaged. Group by their operating system and count the number of users for each operating system. Sort the data in ascending order and obtain first row.
2. Filter data with SociallEngagementType is Socially engaged. Then group the result by operating system and count the occurrence of each operating system. Finally sort the count of operating system in ascending order and obtain the first row.
3. Match SociallEngagementType is "Socially Engaged". Group with id operating system and count with sum :1. Sort with count: 1. Limit:1
4. "$match" : { "socialEngagementType" : "Socially Engaged"} , "$group" : {"\_id" : "$operatingSystem", “count”:{$sum:1}}, "$sort" : {"count" : 1.0} and "$limit" : 1.0

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



## **Query 5**

### Question

Provide a breakdown of unique visitors by mobile vs nonmobile users?

### 

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

Code:

db.getCollection**(**"project"**)**.aggregate**(**

**[**

**{**

"$match" **:** **{**

"isMobile" **:** **{**

"$in" **:** **[**

"true"**,**

"false"

**]**

**}**

**}**

**},**

**{**

"$group" **:** **{**

"\_id" **:** "$isMobile"**,**

"totalRows" **:** **{**

"$sum" **:** 1.0

**},**

"totalCustomer" **:** **{**

"$addToSet" **:** "$fullVisitorID"

**}**

**}**

**}**

**],**

**);**

Result:

**#2 rows obtained**

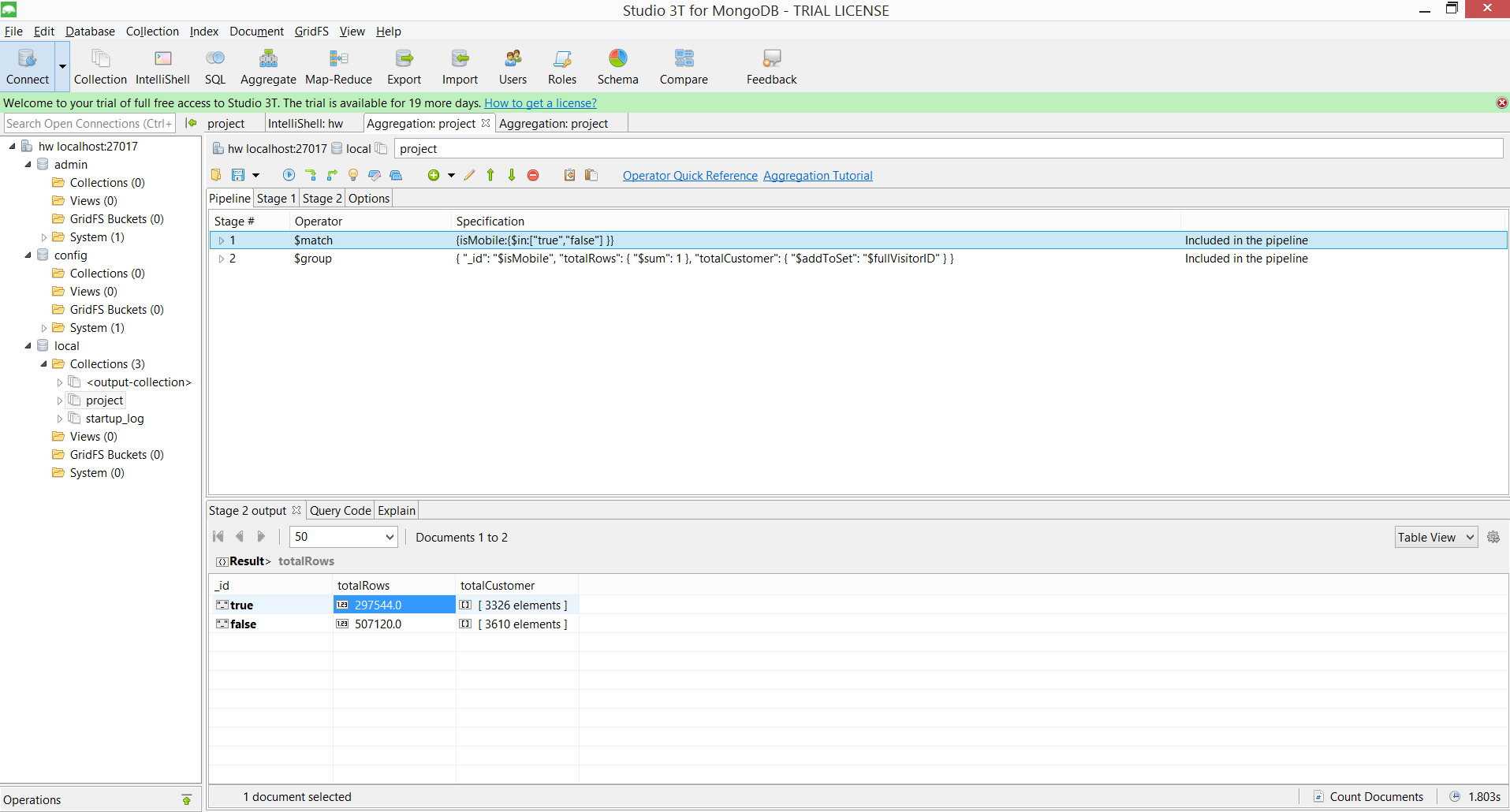
**Mobile users: 3326**

**Non-mobile users: 3610**

### Translation

1. Filter data for mobile device users and non-mobile users. Count number of users for mobile and non-mobile users. Count total number of unique users for mobile and non-mobile users.
2. Filter data with isMobile is true and false. Then group the result of true and false separately. Count the number of rows in each. Finally use full visitor id to get count total number of unique users for mobile and non-mobile users.
3. Match isMobile is true, false. Group with id isMobile and total rows with sum :1. Total customers with $addtoset: $fullvisitorID
4. “$match” : “isMobile” : {“$in” : [ “true”, “false”]}, “$group”: {“\_id” : “isMobile”, “total rows”:{$sum:1}}. Total customers: "$addToSet"**:** "$fullVisitorID"

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



## **Query 6**

### Question

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

### 

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

Code:

db.getCollection**(**"project"**)**.aggregate**(**

**[**

**{**

"$group" **:** **{**

"\_id" **:** "$fullVisitorID"**,**

"aa" **:** **{**

"$push" **:** "$operatingSystem"

**}**

**}**

**},**

**{**

"$match" **:** **{**

"aa" **:** **{**

"$nin" **:** **[**

"Windows"**,**

"Android"**,**

"Macintosh"**,**

"Linux"**,**

" Chrome OS "**,**

"(not set)"**,**

"Tizen"

**]**

**}**

**}**

**}**

**]**

**);**

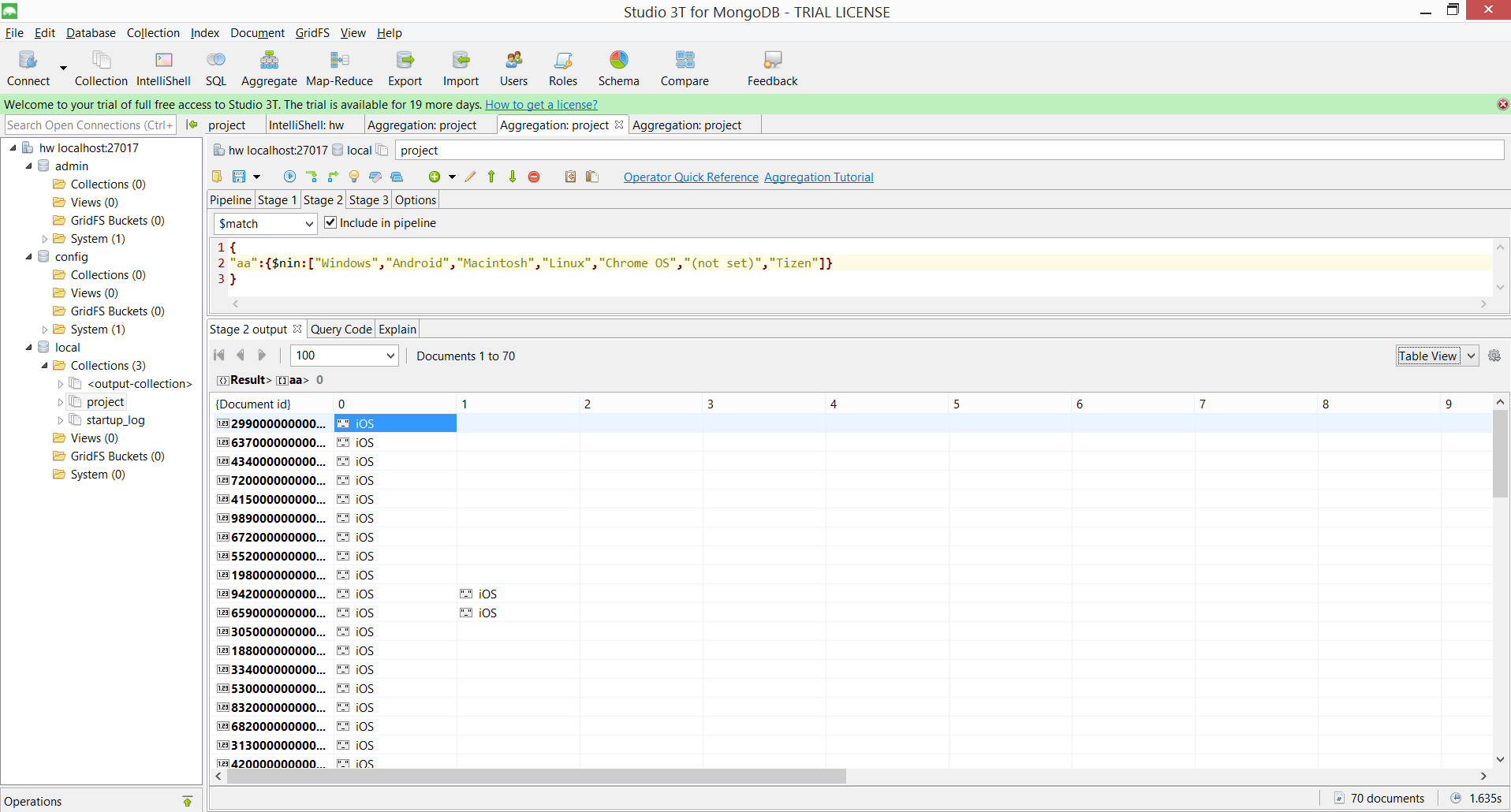
Result:

**#70 rows returned indicating 70 Visitors**

### Translation

1. Group the data by visitor ID and obtain operating system used by each user. Filter out visitors using operating systems such as windows, android, Macintosh, Linux, Chrome OS, tizen and not set.
2. Group by fullVisitorID and push operating system. Match operating system using nin function for windows, android, Macintosh, Linux, Chrome OS, tizen and not set.
3. "$group" : {"\_id" : "$fullVisitorID", "aa" : {"$push" : "$operatingSystem"}, "$match" : {"aa" : {"$nin" : ["Windows", "Android", "Macintosh", "Linux", " Chrome OS ","(not set)", "Tizen"}

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



## **Query 7**

### Question

Which user generated the least amount of hits and when?

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

Code:

db.getCollection**(**"project"**)**.aggregate**(**

**[**

**{**

"$group" **:** **{**

"\_id" **:** "$fullVisitorID"**,**

"countofhits" **:** **{**

"$sum" **:** "$hits"

**},**

"date" **:** **{**

"$push" **:** "$date"

**}**

**}**

**},**

**{**

"$sort" **:** **{**

"countofhits" **:** 1.0

**}**

**},**

**{**

"$match" **:** **{**

"countofhits" **:** 1.0

**}**

**}**

**]**

**);**

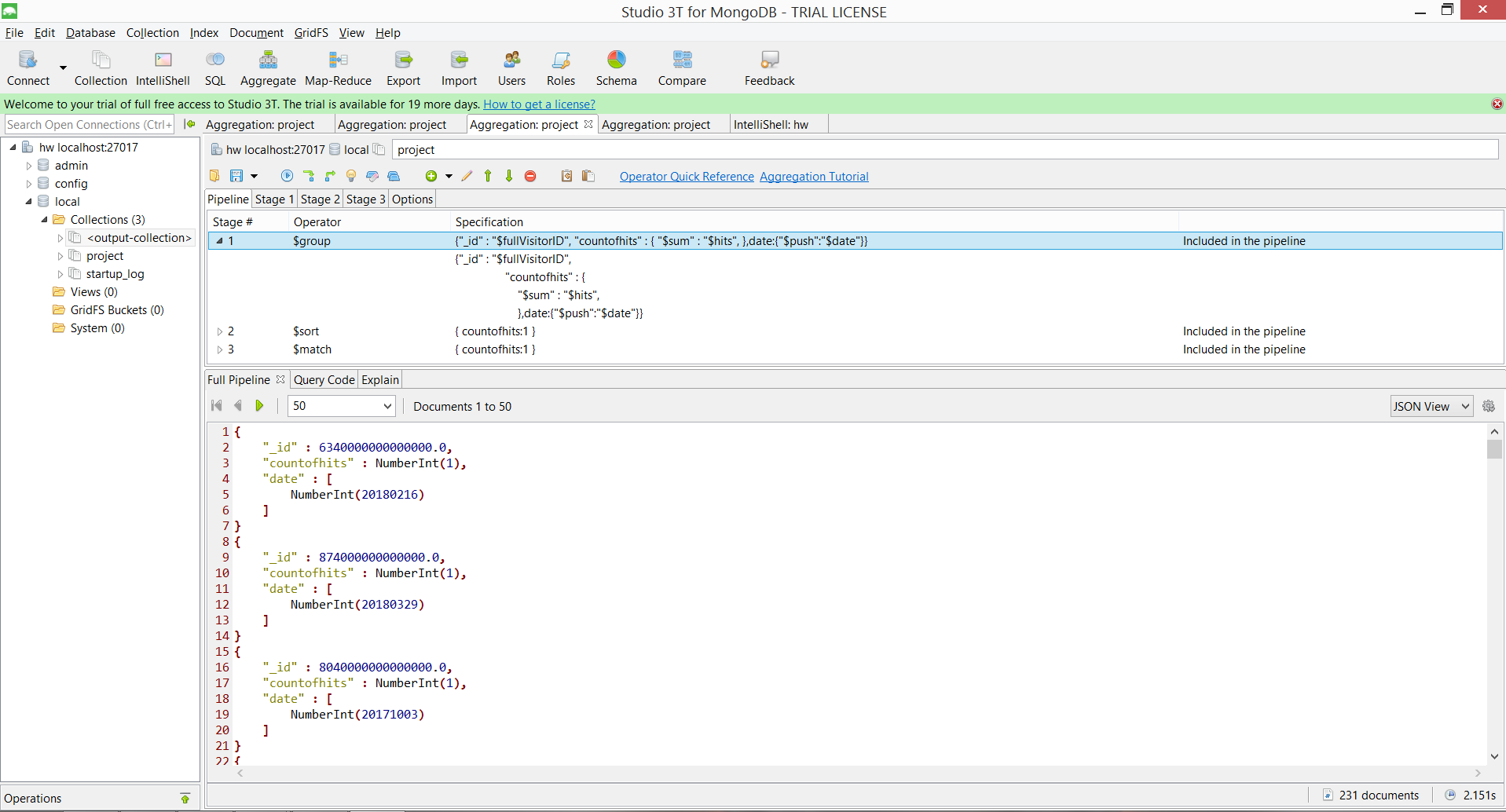
Result:

**231 rows obtained with fullVisitor ID, Countofhits and Date.**

### Translation

1. Group the visitors by visitor ID and count number of hits made by each visitor along with date information. Sort the number of hits in ascending order. Match number of hits to 1.
2. Group by fullVisitorID, count number of hits and push date. Sort number of hits in ascending order and match number of hits to 1.
3. "$group" : "\_id" : "$fullVisitorID", "countofhits" : {"$sum" : "$hits"}, "date" : {"$push" : "$date"},"$sort" : {"countofhits" : 1.0}, "$match" : {"countofhits" : 1.0}

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



## **Query 8**

### Question

Visitors from which country visited the store more than once?

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

Code:

db.getCollection**(**"project"**)**.aggregate**(**

**[**

**{**

"$match" **:** **{**

"visitNumber" **:** **{**

"$nin" **:** **[**

"visitNumber"**,**

1.0

**]**

**}**

**}**

**},**

**{**

"$group" **:** **{**

"\_id" **:** "$fullVisitorID"**,**

"country\_name" **:** **{**

"$push" **:** "$country"

**},**

"visit\_count" **:** **{**

"$push" **:** "$visitNumber"

**}**

**}**

**}**

**],**

**{**

"allowDiskUse" **:** false

**}**

**);**

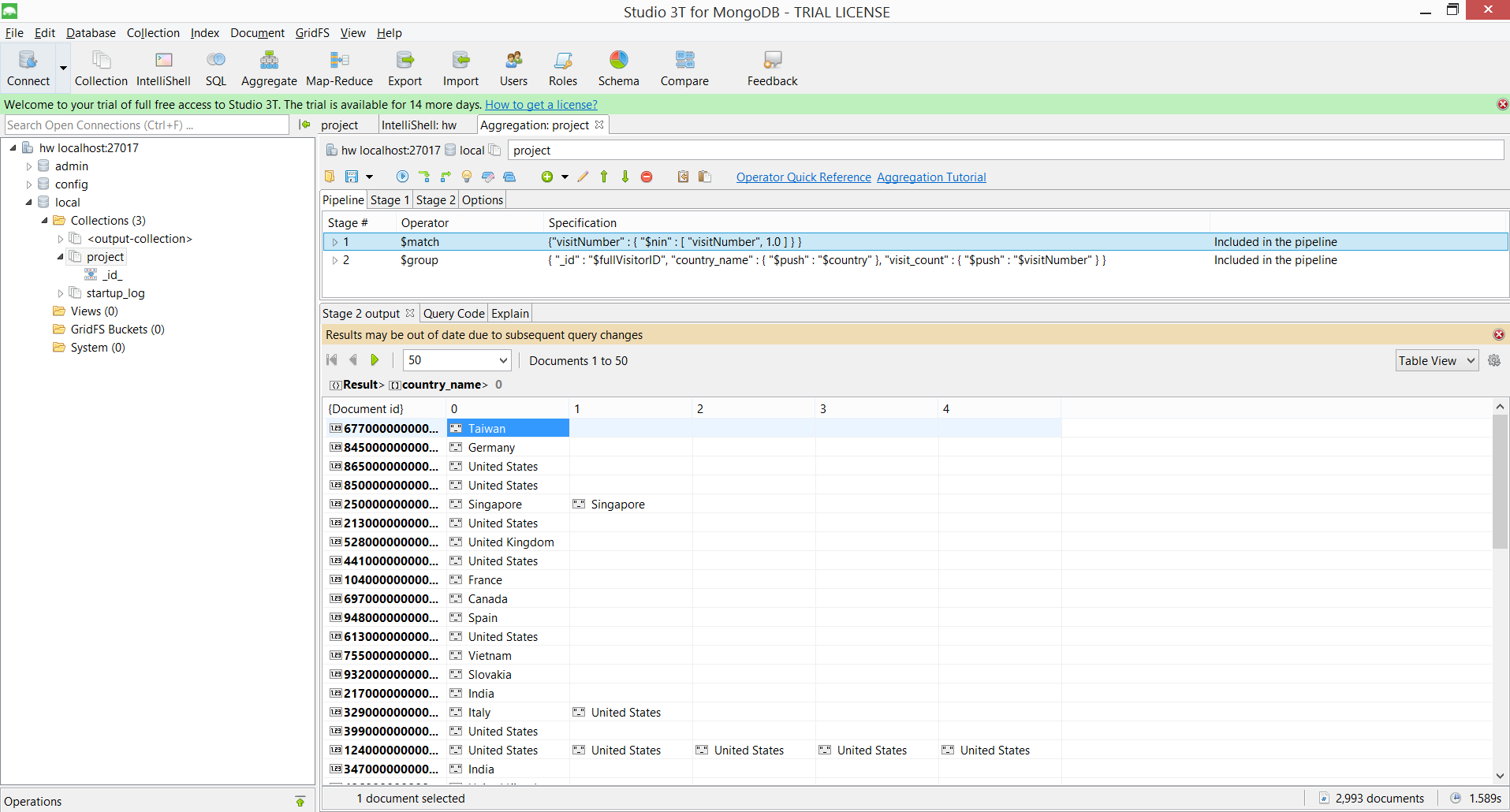
Result:

**#2993 rows obtained. Result shows list of visitors visiting the store more than once along with Country from which they visited.**

### Translation

1. Filter visitors visiting the store more than once. Group the filtered visitors using visitor id and include country information and number of visit.
2. Match visit number not equal to 1. Group by fullVisitorID and push country and VisitNumber.
3. "$match" : {"visitNumber" : {"$nin" : ["visitNumber", 1.0], "$group" : {"\_id" : "$fullVisitorID", "country\_name" : {"$push" : "$country"}, "visit\_count" : {"$push" : "$visitNumber"}

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



Assumption: We are listing countries per visitor.

## **Query 3**

### Question

Which date had the least and most number of visitors?

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

Code:

use local**;**

db.getCollection**(**"project"**)**.aggregate**(**

**[**

**{**

"$group" **:** **{**

"\_id" **:** "$date"**,**

"count" **:** **{**

"$sum" **:** 1.0

**}**

**}**

**},**

**{**

"$sort" **:** **{**

"count" **:** -1.0

**}**

**},**

**{**

"$limit" **:** 1.0

**}**

**],**

**{**

"allowDiskUse" **:** false

**});**

Result:

**# MOST NO OF VISITORS - Date:20171212 Number of Visitors: 9324**

**# LEAST NO OF VISITORS - Date:20171231 Number of visitors: 1495**

### Translation

1. Group the data by date of visit and count the number of visits on each day. Sort the data in ascending order to obtain least and descending order to obtain most number of visits.
2. Group with id as date and count with sum:1. Sort with count :-1 for least and count:1 for most. Limit:1 to obtain first row.
3. "$group" : { "\_id" : "$date"}, “count”: {$sum:1} "$sort" : {"count" : -1.0},"$sort" : {"count" : 1.0}, "$limit" : 1.0

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

