MongoDB Project – Google Store Visitor Data

BUAN 6320.003

Group Members
Sai Pratheek Banda
Prithvi Gude
Sriharshith Reddy Nimmala
Nikhil Kumar Kasham

Group #: 10

Contents

Data Review

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

Notes: We as a team decided to use a single collection for our project, "*The Google Store"* Data has been cleaned and placed into a single Collection called "*Store*" in the database "*Google*". We have chosen to do so since MongoDB is "*Schema Less*" and data can be queried without actually defining a Schema.

Database: "Google"

```
MongoDB Enterprise > show databases
Google   0.206GB
admin    0.000GB
config   0.000GB
local    0.000GB
testemp  0.191GB
```

```
MongoDB Enterprise > use Google
switched to db Google
```

Collection: "Store"

```
MongoDB Enterprise > show collections
Store
```

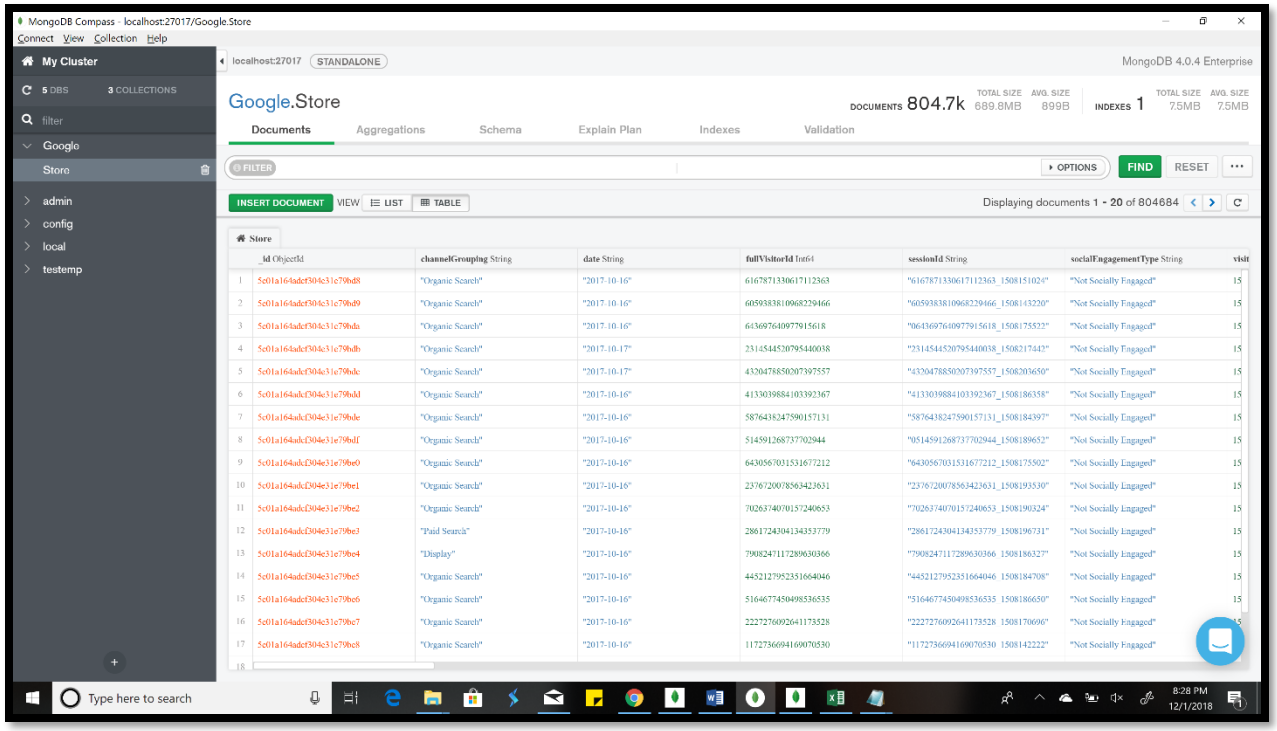
Physical Database

Assumptions/Notes About Data Set

Dataset has been cleaned such that each of key value pairs are placed into fields and documents respectively.

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

```
MongoDB Enterprise > db.Store.pretty()
2018-12-01T22:24:40.468-0600 E QUERY    [js] TypeError: db.Store.pretty is not a function :
@(shell):1:1
MongoDB Enterprise > db.Store.find().pretty()
{
        "_id" : ObjectId("5c01a164adcf304e31e79bd8"),
        "channelGrouping" : "Organic Search",
        "date" : "2017-10-16",
        "fullVisitorId" : NumberLong("6167871330617112363"),
        "sessionId" : "6167871330617112363_1508151024",
        "socialEngagementType" : "Not Socially Engaged",
        "visitId" : 1508151024,
        "visitNumber" : 2,
        "visitStartTime" : "2017-10-16T10:50:24",
        "browser" : "Chrome",
        "operatingSystem" : "Macintosh",
        "isMobile" : "FALSE",
        "deviceCategory" : "desktop",
        "continent" : "Asia",
        "subContinent" : "Southeast Asia",
        "country" : "Singapore",
        "region" : "(not set)",
        "metro" : "(not set)",
        "city" : "(not set)",
        "networkDomain" : "myrepublic.com.sg",
        "visits" : 1,
        "hits" : 4,
        "pageviews" : 4,
        "newVisits" : "NA",
        "bounces" : "NA",
        "campaign" : "(not set)",
        "source" : "google",
        "medium" : "organic",
        "keyword" : "(not provided)",
        "isTrueDirect" : "TRUE",
        "referralPath" : "NA",
        "adContent" : "NA",
        "adwordsClickInfo" : {
                "page" : "NA",
                "slot" : "NA",
                "gclId" : "NA",
                "adNetworkType" : "NA",
                "isVideoAd" : "NA"
        }
}
```



Data in the Database

Collection Name	Relationshps With Other Collections (if any)	# of Rows in Table
Store	N/A	804,684

Fields

channelGrouping	date	fullVisitorId	sessionId
socialEngagementType	visitId	visitNumber	visitStartTime
browser	operatingSystem	isMobile	deviceCategory
continent	subContinent	country	region
metro	city	networkDomain	visits
hits	pageviews	newVisits	bounces
campaign	source	medium	keyword
isTrueDirect	referralPath	adContent	adwordsClickInfo_page
adwordsClickInfo_slot	adwordsClickInfo_gclId		
adwordsClickInfo_adNetworkType	adwordsClickInfo_isVideoAd		

MongoDB Queries/Code

Query 1

Question

Which user had the maximum number of visits and when?

Translation

Grouping the data with *'FullVisitorID'* and then by *'Date'*, the visits are counted for each Visitor Id for each day. Results are then sorted in *'Descending'* order to get the number of maximum counts and the one with maximum count at the top is limited.

Cleanup

[{$group: {_id: {visitorID: "$fullVisitorId", day: "$date"},

count: {$sum: "$visits"}}},

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

{$limit: 1}],

MongoDB Query

db.runCommand({aggregate: "Store", pipeline: [{$group: {_id: {visitorID: "$fullVisitorId", day: "$date"}, count: {$sum: "$visits"}}}, {$sort: {count: -1}}, {$limit: 1}], allowDiskUse: true, cursor: {}});

Screen Shot of MongoDB Query/Code and Results

```
MongoDB Enterprise > db.runCommand({aggregate: "Store", pipeline: [{$group: {_id: {visitorID: "$fullVisitorId", day: "$date"}, count: {$sum: "$visits"}}}, {$sort: {count: -1}}, {$limit: 1}], allowDis
kUse: true, cursor: {}})
{
        "cursor" : {
                "firstBatch" : [
                        {
                                "_id" : {
                                        "visitorID" : NumberLong("3106093350313619815"),
                                        "day" : "2017-12-13"
                                },
                                "count" : 16
                        }
                ],
                "id" : NumberLong(0),
                "ns" : "Google.Store"
        },
        "ok" : 1
}
```

Results

Visitor ID *3106093350313619815* is the user with the maximum number of visits on the date *2017-12-13*.

Query 2

Question

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

Translation

Grouping the data with '*Operating System*', counting the number of documents under each grouped Operating System, followed by sorting the count in '*Descending'* order then limiting the result to 1 so that we get the maximum Operating Systems used by the Devices.

<u>Cleanup</u>

[{$group:{_id: {operatingSystem: "$operatingSystem"},

count:{$sum: 1}}},

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

{$limit:1}

MongoDB Query

db.runCommand({aggregate: "Store",pipeline: [{$group:{_id: {operatingSystem: "$operatingSystem"},count:{$sum: 1}}},{$sort: {count: -1}},{$limit:1}], allowDiskUse: true,cursor: {}});

Screen Shot of MongoDB Query/Code and Results

```
MongoDB Enterprise > db.runCommand({aggregate: "Store", pipeline:[{$group: {_id: {operatingSystem: "$operatingSystem"}, count: {$sum: 1}}}, {$sort: {count: -1}}, {$limit:1}], allowDiskUse: true, curs
or: {}})
{
        "cursor" : {
                "firstBatch" : [
                        {
                                "_id" : {
                                        "operatingSystem" : "Windows"
                                },
                                "count" : 269648
                        }
                ],
                "id" : NumberLong(0),
                "ns" : "Google.Store"
        },
        "ok" : 1
}
```

Results

Windows was the most popular operating system used to visit the store with a count of *269648*.

Query 3

Question

Which date had the least and most number of visits?

Translation

Grouping by '*date*', we count the no if '*Visits*' made per date and then sort the results in '*Ascending*' for the minimum visits in a day and '*Descending*' to get the maximum number of visits in a day.

Clean up

a)

{$group: {_id: {Date: "$date"},

count: {$sum:"$visits"}}},

{$sort: {count: +1}},

{$limit:1}

b)

{$group: {_id: {Date: "$date"},

count: {$sum:"$visits"}}},

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

{$limit:1}

MongoDB Query

a) Least Visits

db.runCommand({aggregate: "Store", pipeline:[{$group: {_id: {Date: "$date"}, count: {$sum:"$visits"}}}, {$sort: {count: +1}}, {$limit:1}], allowDiskUse: true, cursor: {}})

b) Most Visits

db.runCommand({aggregate: "Store", pipeline:[{$group: {_id: {Date: "$date"}, count: {$sum:"$visits"}}}, {$sort: {count: -1}}, {$limit:1}], allowDiskUse: true, cursor: {}})

Screen Shot of MongoDB Query/Code and Results

a)

```
MongoDB Enterprise > db.runCommand({aggregate: "Store", pipeline:[{$group: {_id: {Date: "$date"}, count: {$sum:"$visits"}}}, {$sort: {count: +1}}, {$limit:1}], allowDiskUse: true, cursor: {}})
{
        "cursor" : {
                "firstBatch" : [
                        {
                                "_id" : {
                                        "Date" : "2018-05-01"
                                },
                                "count" : 642
                        }
                ],
                "id" : NumberLong(0),
                "ns" : "Google.Store"
        },
        "ok" : 1
}
```

b)

```
MongoDB Enterprise > db.runCommand({aggregate: "Store", pipeline:[{$group: {_id: {Date: "$date"}, count: {$sum:"$visits"}}}, {$sort: {count: -1}}, {$limit:1}], allowDiskUse: true, cursor: {}})
{
        "cursor" : {
                "firstBatch" : [
                        {
                                "_id" : {
                                        "Date" : "2017-12-13"
                                },
                                "count" : 14710
                        }
                ],
                "id" : NumberLong(0),
                "ns" : "Google.Store"
        },
        "ok" : 1
}
```

Results

On *2018-05-01* the store had the least number of visits with a count of *642*.

On *2017-12-13* the store had the most number of visits with a count of *14710*.

Query 4

Question

Which country had the most number of iOS users who were socially engaged?

Translation

Filter the data where Operating System is "*iOS*" and Social Engagement type is "*Socially Engaged*", we then group the data by '*Country'* and count the same. Sorting them in '*Descending'* order to get the maximum number of users and its Country.

Cleanup

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

{socialEngagementType: "Socially Engaged"}]}},

{$group:{_id:"$country", count:{$sum:1}}},

{$sort:{"count":-1}}, {$limit:1}

MongoDB Query

db.Store.aggregate([{$match: {$and: [{operatingSystem:"iOS"},{socialEngagementType: "Socially Engaged"}]}}, {$group:{_id:"$country", count:{$sum:1}}},{$sort:{"count":-1}}, {$limit:1}])

Screen Shot of MongoDB Query/Code and Results

```
MongoDB Enterprise > db.Store.aggregate([{$match: {$and: [{operatingSystem:"iOS"},{socialEngagementType: "Socially Engaged"}]}}, {$group:{_id:"$country", count:{$sum:1}}},{$sort:{"count":-1}}, {$limi
t:1}])
```

Result

No Countries have the most number of iOS users who were socially engaged.

Query 5

Question

Provide a breakdown of unique visitors by operating system type

Translation

Grouping of '*Operating System'* followed by grouping of '*fullVisitorID'* and sending it into an array shows us a list of Operating Systems and all the unique Visitors under each operating System.

Clean up

```
{$group:{_id: {OS: "$operatingSystem", VID: "$fullVisitorId"}, count:{$sum:1}}}

toArray()
```

MongoDB Query

```
db.Store.aggregate([{$group:{_id: {OS: "$operatingSystem", VID: "$fullVisitorId"}, count:{$sum:1}}}],
{allowDiskUse: true, cursor: {}}).toArray()
```

Screen Shot of MongoDB Query/Code and Results

```
{
        "_id" : {
                "operatingSystem" : "iOS",
                "v" : "999942696836758000"
        },
        "count" : 3
},
{
        "_id" : {
                "operatingSystem" : "iOS",
                "v" : "99995103829892500"
        },
        "count" : 1
},
{
        "_id" : {
                "operatingSystem" : "iOS",
                "v" : "999962728776103000 0"
        },
        "count" : 1
},
{
        "_id" : {
                "operatingSystem" : "iOS",
                "v" : "9999789814107280000"
        },
        "count" : 1
},
{
        "_id" : {
                "operatingSystem" : "iOS",
                "v" : "9999803509476550000"
        },
        "count" : 1
},
{
        "_id" : {
                "operatingSystem" : "iOS",
                "v" : "9999997304197520000"
        },
        "count" : 1
}
```

Result

Result too huge to display.

Query 6

Question

How many users have used both mobile and nonmobile devices to visit the store?

Translation

Segregating the visitors, by filtering the device used to visit the store. If mobile device was used, count of '*visitorID'* is increased by 1 in "*Mobile*". Similarly, if non-mobile device was used, count of "visitorID" is increased by 1 in "*NotMobile*". Then, if both Mobile and NotMobile are greater than zero, the visitorID is picked as a user having used both mobile and non-mobile devices to visit the store.

Cleanup

{$group: {_id: {FullVisitorId: "$fullVisitorId"},
Mobile: {$sum: {"$switch": {"branches": [{"case": {"$eq": ["$isMobile", "TRUE"]},"then": 1}],"default": 0}}},

NotMobile: {$sum: {"$switch": {"branches": [{"case": {"$eq": ["$isMobile", "FALSE"]},"then": 1}],"default": 0}}} }},

{$match: {Mobile: {$gt: 0},NotMobile: {$gt: 0}}},

{$count: '1'}

MongoDB Query

db.runCommand({aggregate:"Store", pipeline:[{$group: {_id: {FullVisitorId: "$fullVisitorId"}, Mobile: {$sum: {"$switch": {"branches": [{"case": {"$eq": ["$isMobile", "TRUE"]},"then": 1}],"default": 0}}}, NotMobile: {$sum: {"$switch": {"branches": [{"case": {"$eq": ["$isMobile", "FALSE"]},"then": 1}],"default": 0}}} }}, {$match: {Mobile: {$gt: 0},NotMobile: {$gt: 0}}}, {$count: '1'}], allowDiskUse: true, cursor: {}})

Screen Shot of MongoDB Query/Code and Results

```
MongoDB Enterprise > db.runCommand({aggregate:"Store", pipeline:[{$group: {_id: {FullVisitorId: "$fullVisitorId"}, Mobile: {$sum: {"$switch": {"branches": [{"case": {"$eq": ["$isMobile", "TRUE"]},"th
en": 1}],"default": 0}}}, NotMobile: {$sum: {"$switch": {"branches": [{"case": {"$eq": ["$isMobile", "FALSE"]},"then": 1}],"default": 0}}} }}, {$match: {Mobile: {$gt: 0},NotMobile: {$gt: 0}}}, {$coun
t: '1'}], allowDiskUse: true, cursor: {}})
{
        "cursor" : {
                "firstBatch" : [
                        {
                                "1" : 225
                        }
                ],
                "id" : NumberLong(0),
                "ns" : "Google.Store"
        },
        "ok" : 1
}
```

Result

225 Users have used both mobile and nonmobile devices to visit the store.

Query 7

Question

Which country had the least number of hits higher than zero?

Translation

a) Filtering the data where the '*hits'* are greater than zero, then the data is grouped by '*Country'* and then the total number of hits by each country is counted, then the results are sorted in 'Ascending' order.

b) After running the previous query, we know that the least hits a country had is "1", so we filter the previous result to get all the countries with the minimum count.

c) Another way to show the country with *least hits* is to limit the results to one.

Cleanup

a)

```
{$match : {hits: {$gt: 0}}},

{$group: {_id: {Country: "$country"},

count: {$sum: "$hits"}}},

{$sort: {count: +1}}
```

b)

```
{$match : {hits: {$gt: 0}}},

{$group: {_id: {Country: "$country"},

count: {$sum: "$hits"}}},

{$sort: {count: +1}},

{$match : {count:1}}
```

c)

```
{$match : {hits: {$gt: 0}}},

{$group: {_id: {Country: "$country"},

count: {$sum: "$hits"}}},

{$sort: {count: +1}},

{$limit:1}
```

MongoDB Query

1) db.runCommand({aggregate:"Store", pipeline:[{$match : {hits: {$gt: 0}}}, {$group: {_id: {Country: "$country"}, count: {$sum: "$hits"}}}, {$sort: {count: +1}}], allowDiskUse: true, cursor: {}})

2) db.runCommand({aggregate:"Store", pipeline:[{$match : {hits: {$gt: 0}}}, {$group: {_id: {Country: "$country"}, count: {$sum: "$hits"}}}, {$sort: {count: +1}},{$match : {count:1}}], allowDiskUse: true, cursor: {}})

3) db.runCommand({aggregate:"Store", pipeline:[{$match : {hits: {$gt: 0}}}, {$group: {_id: {Country: "$country"}, count: {$sum: "$hits"}}}, {$sort: {count: +1}},{$limit:1}], allowDiskUse: true, cursor: {}})

Screen Shot of MongoDB Query/Code and Results

1)

```
MongoDB Enterprise > db.runCommand({aggregate:"Store", pipeline:[{$match : {hits: {$gt: 0}}}, {$group: {_id: {Country: "$country"}, count: {$sum: "$hits"}}}, {$sort: {count: +1}}], allowDiskUse: true
, cursor: {}})
{
        "cursor" : {
                "firstBatch" : [
                        {
                                "_id" : {
                                        "Country" : "American Samoa"
                                },
                                "count" : 1
                        },
                        {
                                "_id" : {
                                        "Country" : "Tonga"
                                },
                                "count" : 1
                        },
                        {
                                "_id" : {
                                        "Country" : "Seychelles"
                                },
                                "count" : 1
                        },
                        {
                                "_id" : {
                                        "Country" : "Solomon Islands"
                                },
                                "count" : 1
                        },
                        {
                                "_id" : {
                                        "Country" : "Equatorial Guinea"
                                },
                                "count" : 1
                        },
                        {
                                "_id" : {
                                        "Country" : "Dominica"
                                },
                                "count" : 1
                        },
                        {
                                "_id" : {
                                        "Country" : "St. Barth@lemy"
                                },
                                "count" : 2
                        },
```

2)

```
MongoDB Enterprise > db.runCommand({aggregate:"Store", pipeline:[{$match : {hits: {$gt: 0}}}, {$group: {_id: {Country: "$country"}, count: {$sum: "$hits"}}}, {$sort: {count: +1}},{$match : {count:1}}
], allowDiskUse: true, cursor: {}})
{
        "cursor" : {
                "firstBatch" : [
                        {
                                "_id" : {
                                        "Country" : "American Samoa"
                                },
                                "count" : 1
                        },
                        {
                                "_id" : {
                                        "Country" : "Tonga"
                                },
                                "count" : 1
                        },
                        {
                                "_id" : {
                                        "Country" : "Seychelles"
                                },
                                "count" : 1
                        },
                        {
                                "_id" : {
                                        "Country" : "Solomon Islands"
                                },
                                "count" : 1
                        },
                        {
                                "_id" : {
                                        "Country" : "Equatorial Guinea"
                                },
                                "count" : 1
                        },
                        {
                                "_id" : {
                                        "Country" : "Dominica"
                                },
                                "count" : 1
                        }
                ],
                "id" : NumberLong(0),
                "ns" : "Google.Store"
        },
        "ok" : 1
}
```

3)

```
MongoDB Enterprise > db.runCommand({aggregate:"Store",pipeline:[{$group: {_id: {country: "$country"},hitsCount: {$sum: "$hits"}}}, {$match: {hitsCount: {$gt: 0}}}, {$sort: {hitsCount: 1}}, {$limit: 1
}], allowDiskUse: true, cursor: {}})
{
        "cursor" : {
                "firstBatch" : [
                        {
                                "_id" : {
                                        "country" : "American Samoa"
                                },
                                "hitsCount" : 1
                        }
                ],
                "id" : NumberLong(0),
                "ns" : "Google.Store"
        },
        "ok" : 1
}
```

Result

American Samoa is the country with least number of hits higher than zero.

American Samoa, Tonga, *Seychellas*, *Solomon Islands*, *Equatorial*, *Guinea*, *Dominica* are countries with the least number of hits greater than zero.

Query 8

Question

Which region had more blackberry users than iOS users?

Translation

Grouping the '*Regions'*, by filtering the '*Operating system'* used to visit the store. If '*Blackberry'* was used, count of that region is increased by 1 in '*BlackBerryUsers'*. Similarly, if '*iOS'* was used, count of that region is increased by 1 in '*IOSUsers*'. Then, if value of BlackBerryUsers is greater than IOSUsers, the region is picked as a user with more Blackberry users than iOS users.

Cleanup

{$group: {_id: {region: "$region"},

BlackBerryUsers: {$sum:{"$switch": {"branches": [{"case": {"$eq": ["$operatingSystem", "BlackBerry"]},"then": 1}],"default": 0}}},

IOSUsers: {$sum:{"$switch": {"branches": [{"case": {"$eq": ["$operatingSystem", "iOS"]},"then": 1}],"default": 0}}}}},

{$match: {$expr: {$gt: ["$bbusers", "$iOSusers"]}}}

MongoDB Query

db.runCommand({aggregate:"Store", pipeline:[{$group: {_id: {region: "$region"},BlackBerryUsers: {$sum:{"$switch": {"branches": [{"case": {"$eq": ["$operatingSystem", "BlackBerry"]},"then": 1}],"default": 0}}},IOSUsers: {$sum:{"$switch": {"branches": [{"case": {"$eq": ["$operatingSystem", "iOS"]},"then": 1}],"default": 0}}}}}, {$match: {$expr: {$gt: ["$bbusers", "$iOSusers"]}}}], allowDiskUse: true, cursor: {}})

Screen Shot of MongoDB Query/Code and Results

```
MongoDB Enterprise > db.runCommand({aggregate:"Store", pipeline:[{$group: {_id: {region: "$region"},BlackBerryUsers: {$sum:{"$switch": {"branches": [{"case": {"$eq": ["$operatingSystem", "BlackBerry"
]},"then": 1}],"default": 0}}},IOSUsers: {$sum:{"$switch": {"branches": [{"case": {"$eq": ["$operatingSystem", "iOS"]},"then": 1}],"default": 0}}}}}, {$match: {$expr: {$gt: ["$bbusers", "$iOS
Users"]}}}], allowDiskUse: true, cursor: {}})
{
        "cursor" : {
                "firstBatch" : [ ],
                "id" : NumberLong(0),
                "ns" : "Google.Store"
        },
        "ok" : 1
}
```

Result

No region had more BlackBerry users than IOS users.