Group 18

Sudharsana Rajasekaran

Varun Suresh

Zhining Chen

Mahesh M Iyer

**SQL Project – Google Store Visitor Data**

BUAN 6320.006

Contents

[Data Model 3](#_Toc528400248)

[Assumptions/Notes About Data Entities and Relationships 3](#_Toc528400249)

[Entity-Relationship Diagram 3](#_Toc528400250)

[Physical Database 4](#_Toc528400251)

[Assumptions/Notes About Data Set 4](#_Toc528400252)

[Screen shot of Physical Database objects 4](#_Toc528400253)

[Data in the Database 4](#_Toc528400254)

[SQL Queries 5](#_Toc528400255)

[Query 1 5](#_Toc528400256)

[Question 5](#_Toc528400257)

[Notes/Comments About SQL Query and Results (Include # of Rows in Result) 5](#_Toc528400258)

[Translation 5](#_Toc528400259)

[Screen Shot of SQL Query and Results 5](#_Toc528400260)

[Query 2 6](#_Toc528400261)

[Question 6](#_Toc528400262)

[Notes/Comments About SQL Query and Results (Include # of Rows in Result) 6](#_Toc528400263)

[Translation 6](#_Toc528400264)

[Screen Shot of SQL Query and Results 6](#_Toc528400265)

[Query 3 7](#_Toc528400266)

[Question 7](#_Toc528400267)

[Notes/Comments About SQL Query and Results (Include # of Rows in Result) 7](#_Toc528400268)

[Translation 7](#_Toc528400269)

[Screen Shot of SQL Query and Results 7](#_Toc528400270)

[Query 4 8](#_Toc528400271)

[Question 8](#_Toc528400272)

[Notes/Comments About SQL Query and Results (Include # of Rows in Result) 8](#_Toc528400273)

[Translation 8](#_Toc528400274)

[Screen Shot of SQL Query and Results 8](#_Toc528400275)

[Query 5 9](#_Toc528400276)

[Question 9](#_Toc528400277)

[Notes/Comments About SQL Query and Results (Include # of Rows in Result) 9](#_Toc528400278)

[Translation 9](#_Toc528400279)

[Screen Shot of SQL Query and Results 9](#_Toc528400280)

[Query 6 10](#_Toc528400281)

[Question 10](#_Toc528400282)

[Notes/Comments About SQL Query and Results (Include # of Rows in Result) 10](#_Toc528400283)

[Translation 10](#_Toc528400284)

[Screen Shot of SQL Query and Results 10](#_Toc528400285)

[Query 7 11](#_Toc528400286)

[Question 11](#_Toc528400287)

[Notes/Comments About SQL Query and Results (Include # of Rows in Result) 11](#_Toc528400288)

[Translation 11](#_Toc528400289)

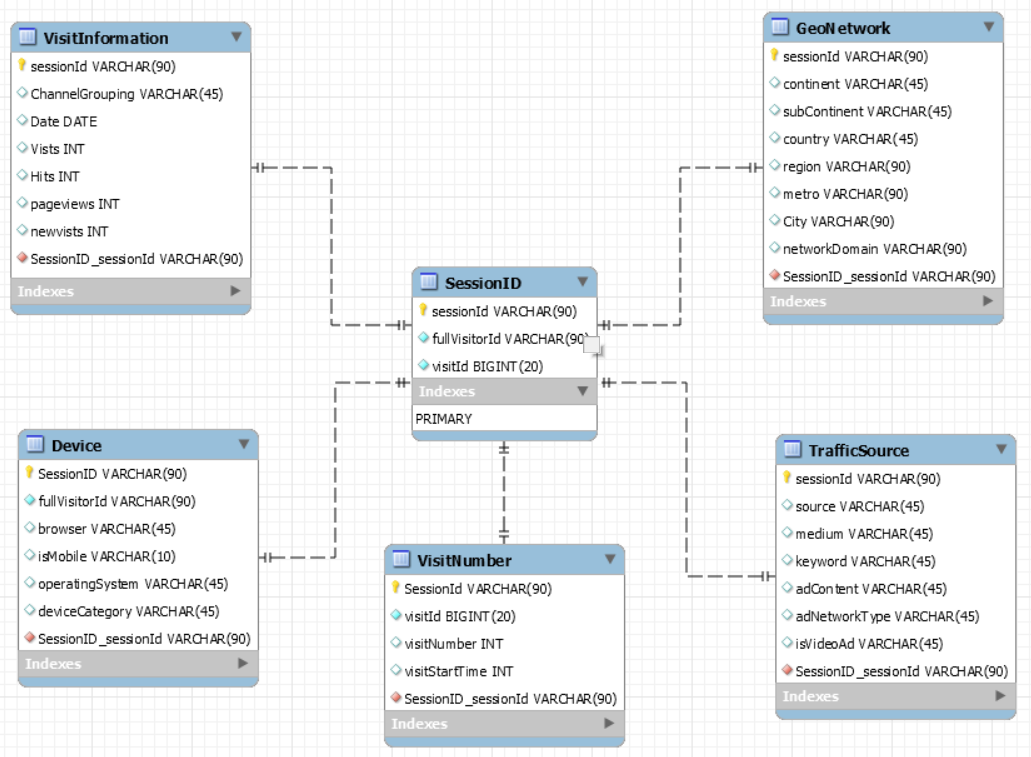
[Screen Shot of SQL Query and Results 11](#_Toc528400290)

# Data Model

## Assumptions/Notes About Data Entities and Relationships

One of our primary assumptions is that session\_id is a combination of full visitorID and visitID. We came to this conclusion because as per he definition “sessionID” is a unique identifier of a visit to a store and this is possible by combining visitorID and visitID

## Entity-Relationship Diagram



*Figure 1 (Entity-Relationship Diagram)*

# Physical Database

## Assumptions/Notes About Data Set

**VisitNumber table**

Ideally visit id should be primary key, but it does not give us unique set of rows for the table.

This is because we noticed that there are multiple records for the same visit id.

Therefore, we have considered session id as the primary key for this table

**Device :**

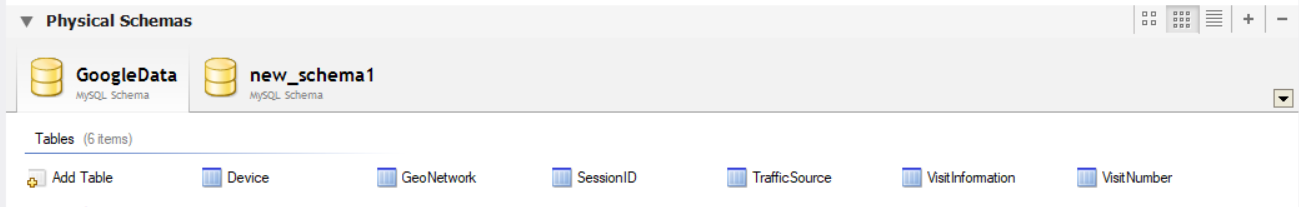
Though the table conveys the kind of device and respective sources through which the customers interacted with Google store, the data is however not unique at a visitor level. We noticed that on the same day the same visitor uses multiple devices/browsers to interact.

Hence, we have considered sessionId as the primary key

The GoogleData database consists of the following tables:

1. Device: Provides all related information regarding usage of devices, its operation system, the type of browser and device category for its customers
2. GeoNetwork: Provides geographical information on the stores
3. SessionId: Provides customer level information based on each visit to the store
4. TrafficSource: The table captures information based on the traffic source from which the session originated
5. VisitInformation: Captures all visit related information for each session
6. VisitNumber: Provides details such as when the visit happened

## Screen shot of Physical Database objects



*Figure 2 (Tables in the Database)*

## Data in the Database

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Primary Key** | **Foreign Key** | **# of Rows in Table** |
| Device | SessionID | Session\_sessionID | 803855 |
| GeoNetwork | SessionID | Session\_sessionID | 803863 |
| SessionID | SessionID | - | 803863 |
| TrafficSource | SessionID | Session\_sessionID | 802345 |
| VisitInformation | SessionID | Session\_sessionID | 803863 |
| VisitNumber | SessionID | Session\_sessionID | 803863 |

*Table 1 – Table and Data*

# SQL Queries

## Query 1

### Question

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

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

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

#Rows in result: 1

***Visitor with the ID =7210496296732356253 on 16th Sept 2017 has the maximum number of visits***

### Translation:

Pick fullvisitorID, date from visitInformation table and visits from sessionID table. Visits need to be aggregated at a visitor level for each day. The visitInformation and sessionID needs to be joined on sessionID. Since we require the aggregation at for each visitor for every day, we need to group the aggregation at the same level. Finally order the table by the sum of visits to identify the visitor who has maximum number of visits in a day

**SQL Query**

select s.fullvisitorid,Date,sum(vists) VisitMax

from sessionid s

inner join visitinformation v

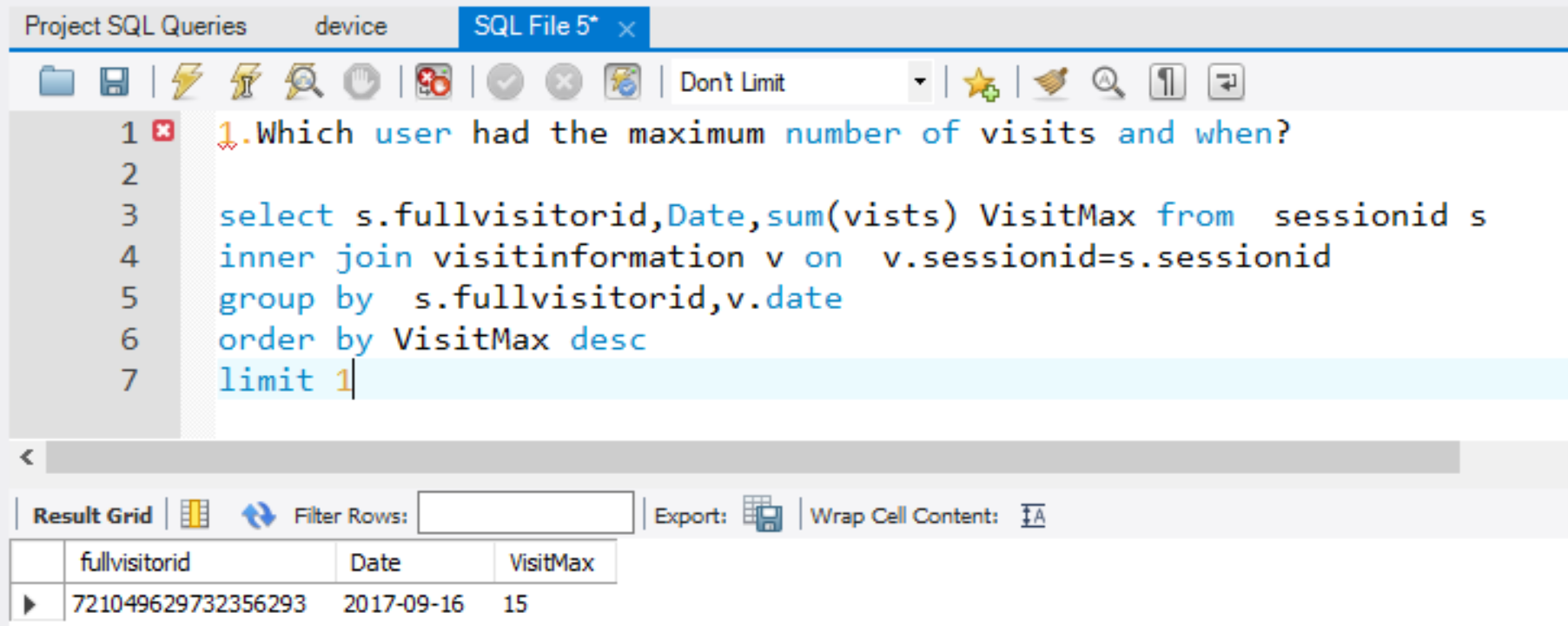
on v.sessionid=s.sessionid

group by s.fullvisitorid,v.date

order by VisitMax desc

limit 1

### Screen Shot of SQL Query and Results



*Figure 3 – Question 1 :Query and Results*

## Query 2

### Question

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

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

The objective of this question is to understand the no: of visits made by blackberry users in comparison to iOS users.

***Blackberry users are less likely to visit the store than iOS users. Blackberry users make only 0.12 % of the total visits.***

### Translation:

Select fullvisitorID and operatingsystem columns from sessionID and device tables. Perform a COUNT aggregation on operatingsystem to count the no: of records. The tables sessionID and device are joined on sessionID. The above set of rules need to be written as an inner query. In the outer query select operating system and SUM the previously aggregated columns to get the count of visits. Filter for OperatingSystem where OS in iOS and Blackberry. Due to the aggregation function, we need to group the aggregation by OS

**SQL Query**

select OS, sum(OScount) from

(select s.fullvisitorid,d.operatingsystem OS,count(d.operatingsystem) OSCount

from sessionid s

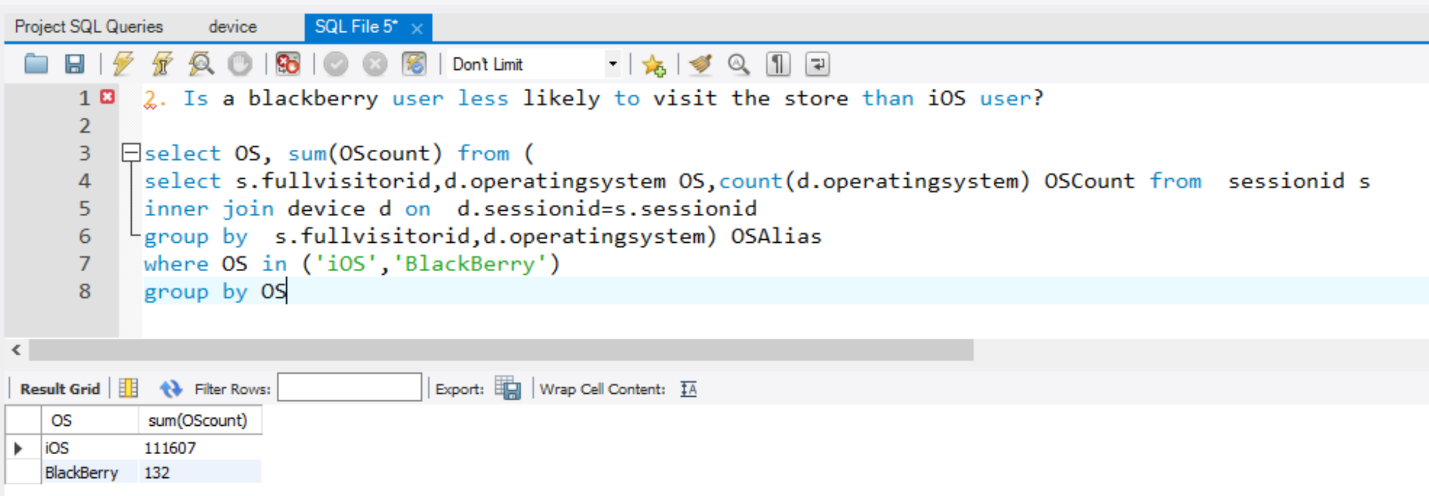
inner join device d

on d.sessionid=s.sessionid

group by s.fullvisitorid,d.operatingsystem) OSAlias

where OS in ('iOS','BlackBerry')

### Screen Shot of SQL Query and Results



*Figure 4 - Question 2 :Query and Results*

## Query 3

### Question

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

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

The objective here is to identify the maximum number of users filtered by operatingsystem = iOS and country = “Belgium”

#Rows in the result: 226 (however we choose only the 1st row)

***Most visits for iOS users from Belgium was on 14th August 2017***

### Translation

Select operating system , date as the parameters that we need to focus on. In order to get the above parameters along with country to filter on ,select from sessionid,device,geonetwork and visitinformation tables. Each of the tables need to be joined on sessionID. The joined tables then need to be filtered on operatingsystem = “iOS” and country like “Belgium”. Since we require maximum number of users , we need to COUNT the no: of records grouped at operatingsystem and date level.

To view the date with most number of iOS users order by the aggregated value

**SQL Query**

select Date,operatingsystem,count(operatingsystem) OSCount

from sessionid s

inner join device d

on d.sessionid=s.sessionid

inner join geonetwork g

on g.sessionid=s.sessionid

inner join visitinformation v

on v.sessionid=s.sessionid

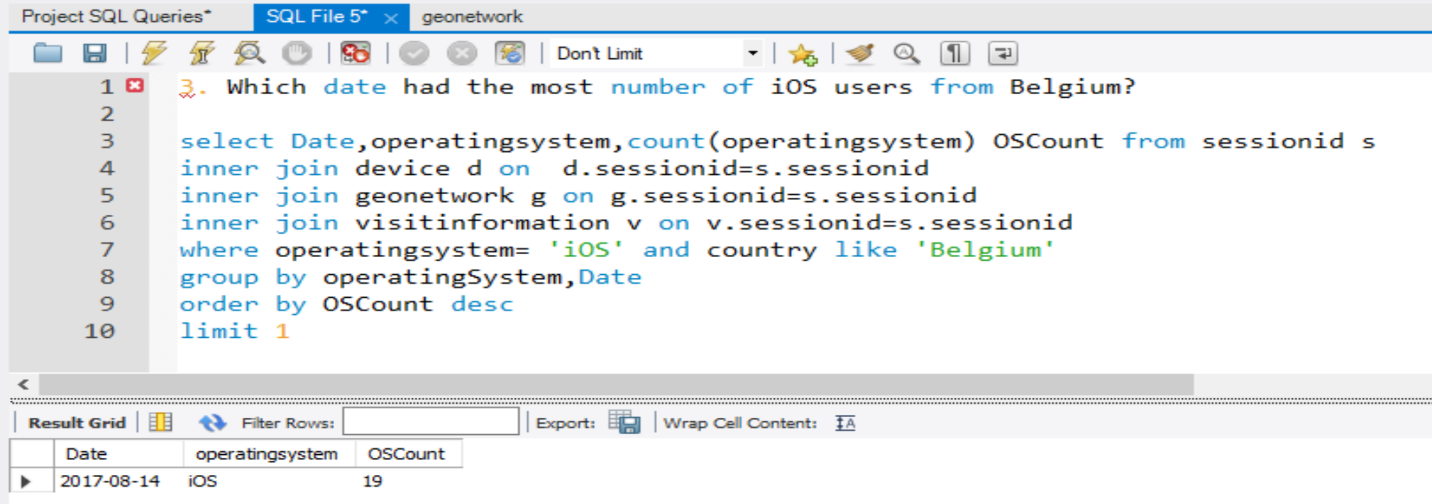
where operatingsystem= 'iOS' and country like 'Belgium'

group by operatingSystem,Date

order by OSCount desc

limit 1

### Screen Shot of SQL Query and Results



*Figure 5 - Question 3 :Query and Results*

## 

## Query 4

### Question

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

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

The objective is to group the data into mobile and non-mobile users and check which has more visits between the two.

***#Rows in the result = 2***

***Mobile phone users were tagged ‘1’ and non-mobile users ‘0’. From the data we notice that there are more non-mobile users (~63%) in comparison to mobile users visiting the store***

### Translation

Select the isMobile flag from device table. Since the records are in character data type, they need to be CASTED into Integer data type. Aggregate the records at the same level. Due to aggregation, the columns need to be grouped by at the isMobile level

**SQL Query**

select

cast(ismobile AS UNSIGNED) as Mobile,

count(cast(ismobile AS UNSIGNED)) CountMobile

from device

group by cast(ismobile AS UNSIGNED)

### Screen Shot of SQL Query and Results

Figure 6- *Question 4: Query and Results*

## Query 5

### Question

Provide a breakdown of store pageviews by city

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

The objective of the question is to see the total number of pageviews segregated by city

***#Rows in the result: 730***

### Translation:

Select the city and pageviews. Aggregate pageviews at the city level. Choose the above parameters from sessionID, visitinformation and geonetwork tables which needs to be joined on sessionID. Since the data has multiple “NULL” and “Not available in dataset” values, they need to be filtered for the same. Since we’re performing an aggregation operation the column city needs to be grouped by. Finally order the values in descending order of the calculated page view to see them in descending order.

**SQL Query**

select city,sum(pageviews) PageViewCount

from sessionid s

inner join geonetwork g

on g.sessionid=s.sessionid

inner join visitinformation v

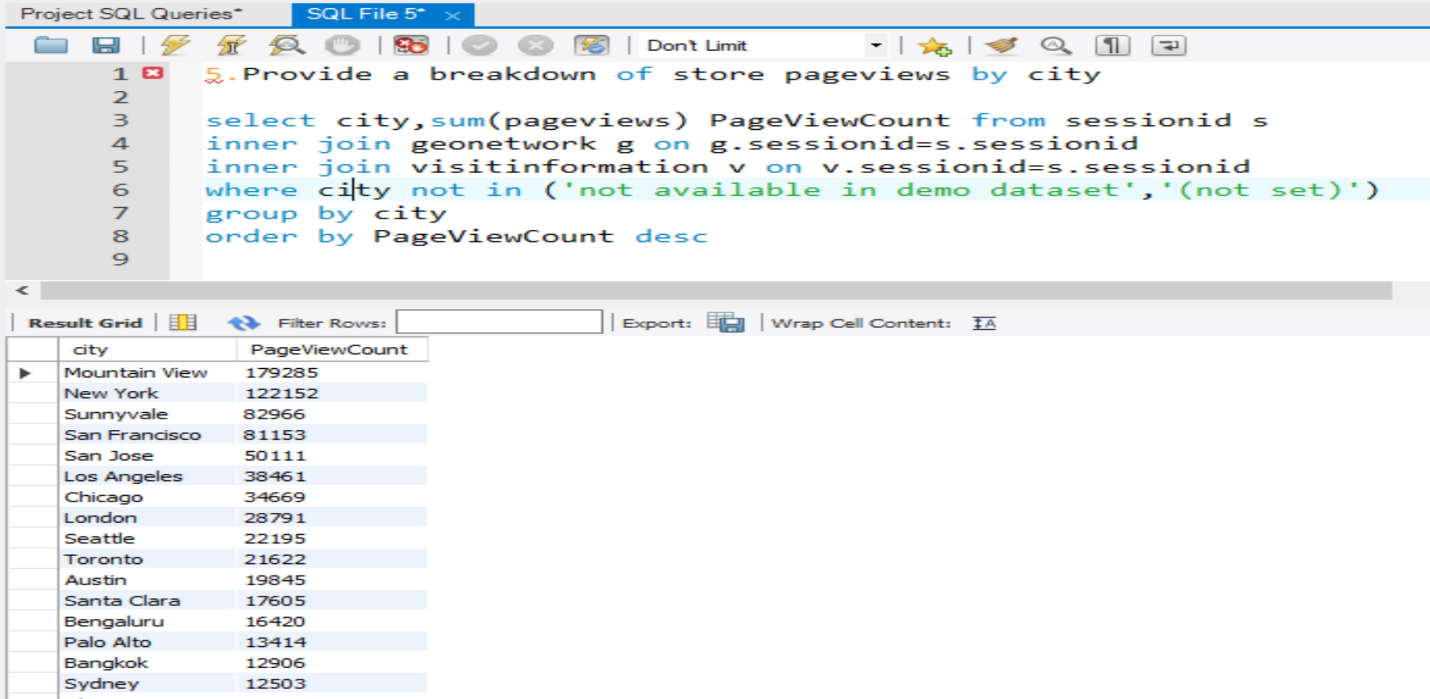
on v.sessionid=s.sessionid

where city not in ('not available in demo dataset','(not set)')

group by city

order by PageViewCount desc

### Screen Shot of SQL Query and Results



*Figure 7 - Question 5: Query and Results*

## Query 6

### Question

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

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

The objective of the question is to identify the number of users using Windows devices

#Rows in the result: 1

***Approximately 270k users visiting the stores use Windows devices***

### Translation

Choose fullvisitorID because that needs to be aggregated . Inorder to select fullvisitorID and Windows devices choose from sessionID and Device tables. Filter for users where operatingsystem =”Windows”

**SQL Query**

select count(s.fullvisitorId) WindowsUserCount

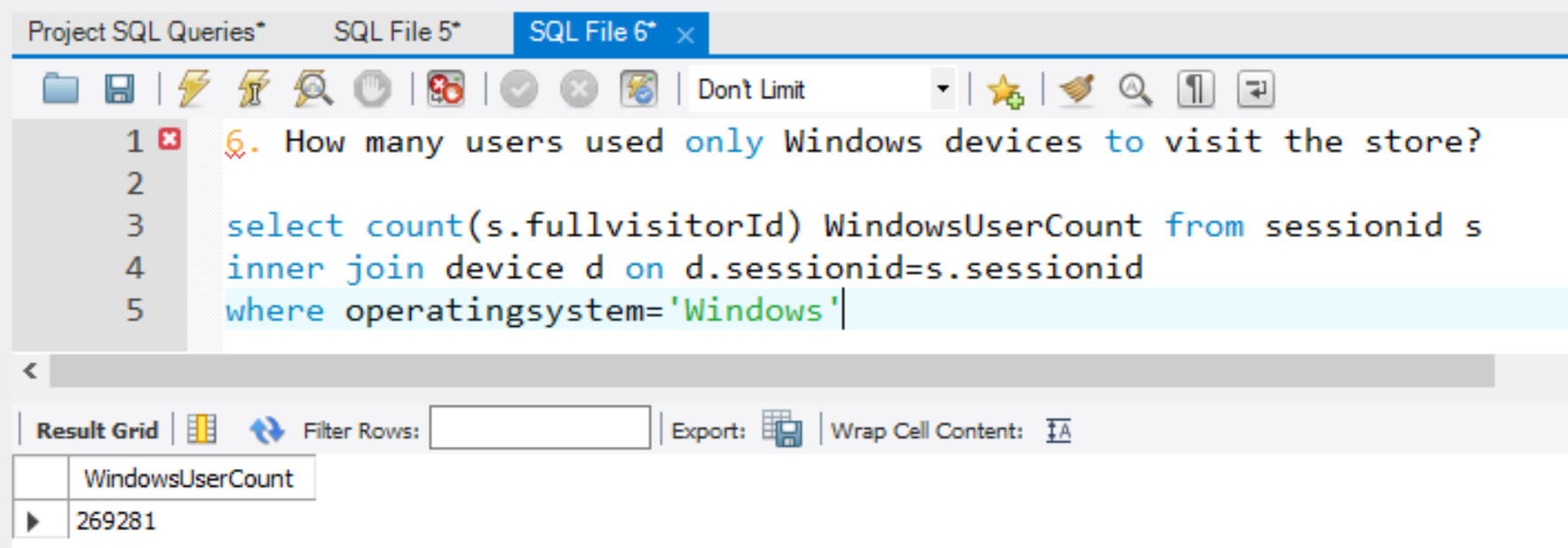
from sessionid s

inner join device d

on d.sessionid=s.sessionid

where operatingsystem='Windows'

### Screen Shot of SQL Query and Results



*Figure 8 -Question 6: Query and Results*

## Query 7

### Question

How many visitors had zero pageviews?

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

The objective of this filter for the visitors who had zero pageviews and aggregate them

#Rows in the result: 1

***There are approximately 100 visitors who had zero pageviews***

### Translation

Select visitID from sessionID and perform count aggregation. Join sessionID and visitInformation on session id to get the required columns. Filter for pageview where pageviews = ‘0’.

**SQL Query**

select count(visitid) ZeroPageviews

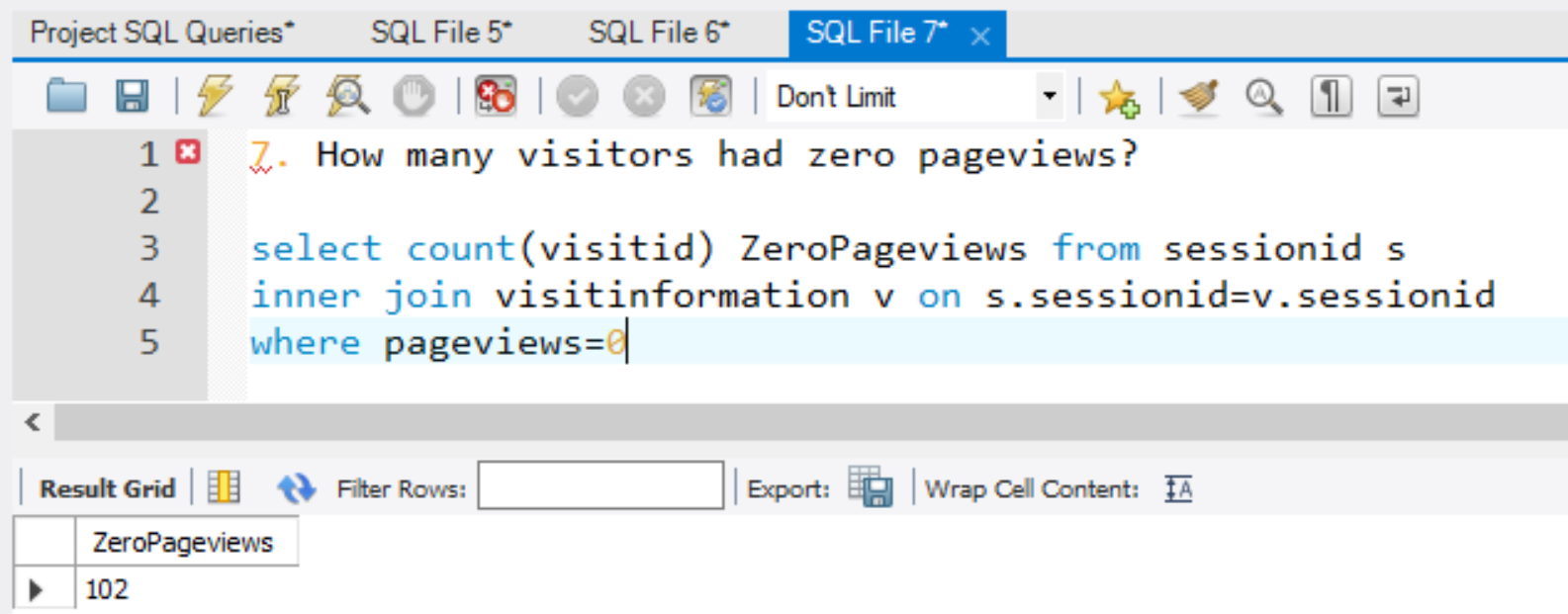
from sessionid s

inner join visitinformation v

on s.sessionid=v.sessionid

where pageviews=0

### Screen Shot of SQL Query and Results



*Figure 8 -Question 7: Query and Results*

## Query 8

### Question

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

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

The objective of the question is to aggregate the users who use desktops by each city and then select the city that has the maximum number of users

#Rows in result: 1

***Mountain View is the city that has the maximum number desktop users***

### Translation

Select the columns city and perform count aggregation on fullvisitorID. Inorder to select the above parameters and filter by Desktop users choose from sessionID, device and geonetwork tables which are joined on sessionID column. Apply the filters on devicecategory = “Desktop” and City “NOT IN” ‘not available in demo dataset’ and ‘not set’(to eliminate redundant data). Due to the aggregation group by the data at city level. Inorder to choose the city with Desktopuser count being maximum, order Desktopuser count by descending order

**SQL Query**

select city,count(s.fullvisitorid) DesktopUserCount

from sessionid s

inner join device d

on s.sessionid=d.sessionid

inner join geonetwork g

on g.sessionid=s.sessionid

where devicecategory = 'desktop'

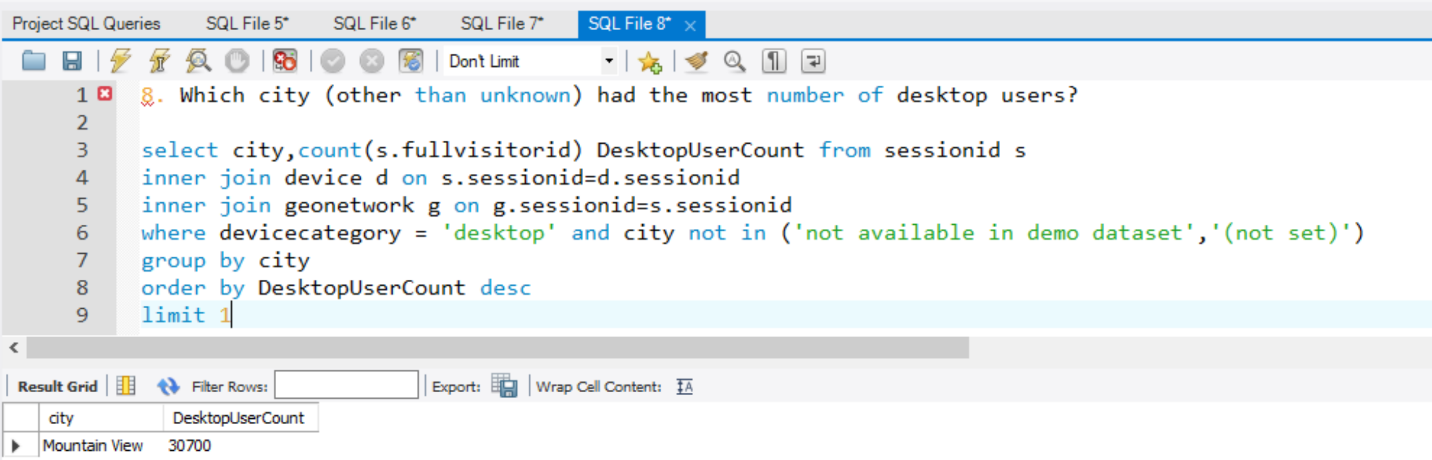
and city not in ('not available in demo dataset','(not set)')

group by city

order by DesktopUserCount desc

limit 1

### Screen Shot of SQL Query and Results



*Figure 8 -Question 8 :Query and Results*