

SQL Project – Google Store Visitor Data

BUAN 6320.003

Contents

Data Model	3
Assumptions/Notes About Data Entities and Relationships.....	3
Entity-Relationship Diagram	3
Physical Database	4
Assumptions/Notes About Data Set	4
Screen shot of Physical Database objects.....	5
Data in the Database	7
SQL Queries	8
Query 1	8
Question	Error! Bookmark not defined.
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	8
Translation	8
Screen Shot of SQL Query and Results	8
Query 2	8
Question	8
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	8
Translation	9
Screen Shot of SQL Query and Results	9
Query 3	10
Question	10
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	10
Translation	10
Screen Shot of SQL Query and Results	10
Query 4	11
Question	11
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	11
Translation	11
Screen Shot of SQL Query and Results	11
Query 5	12
Question	12
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	12
Translation	12

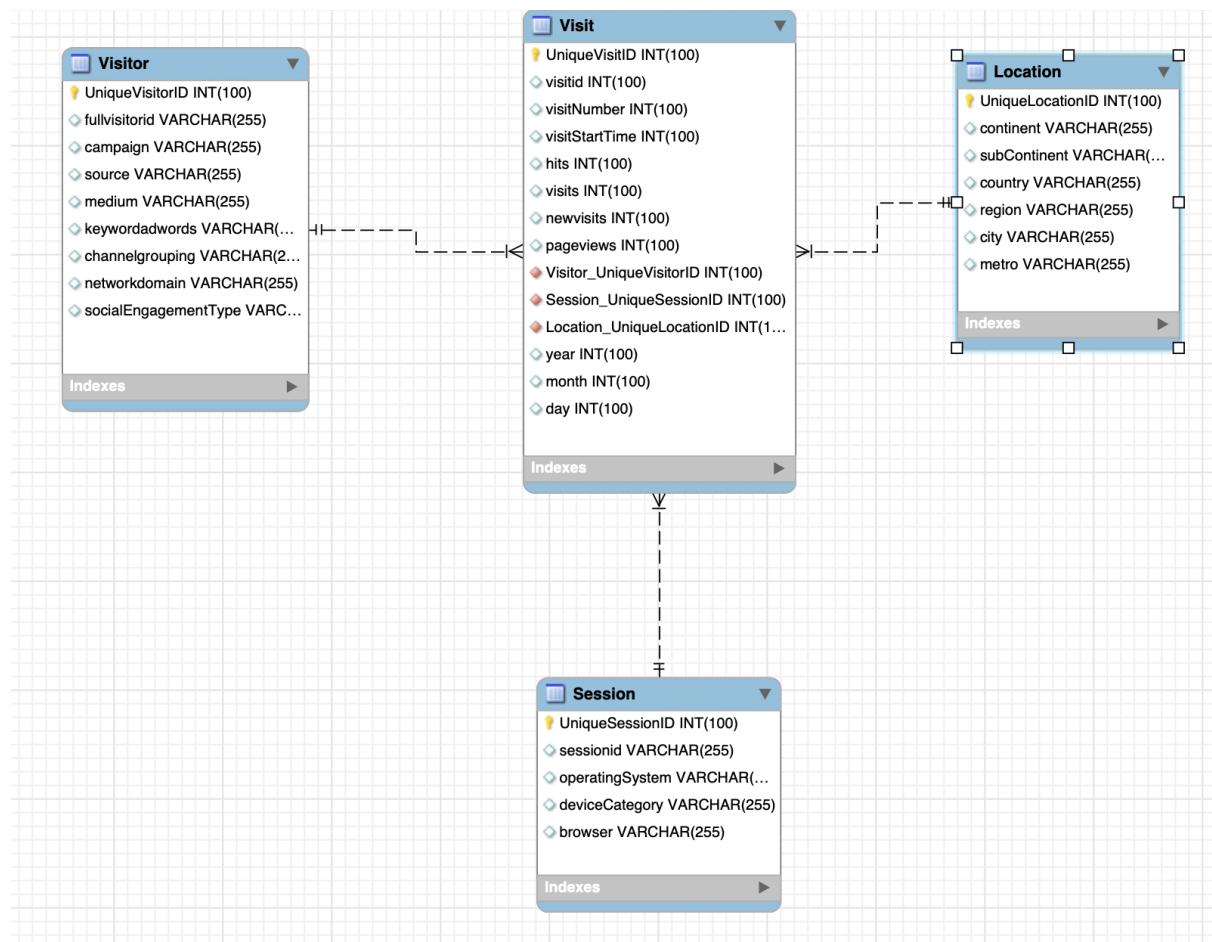
Screen Shot of SQL Query and Results	12
Query 6	13
Question	13
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	13
Translation	13
Screen Shot of SQL Query and Results	13
Query 7	14
Question	14
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	14
Translation	14
Screen Shot of SQL Query and Results	14

Data Model

Assumptions/Notes About Data Entities and Relationships

- Since the data didn't have any unique primary key, we have created unique primary keys in each table, as shown in the ER diagram below which is needed to query the data and get the required results.
 - The Data has been divided into 4 tables Visitor, Visit, Location and Session.
 - The tables Visitor, Location and Session have been linked to the visit table so the 3 foreign keys UniqueVisitorID, UniqueLocationID and UniqueSessionID are foreign keys in the Visit table.
 - The date column has been broken down into 3 columns – year, month and day in order to query the data efficiently.
- Relationship between the Visitor and Visit tables** - There can be **one or many** visits for each visitor, but each visit can be tagged to **one and only one** visitor.
 - Relationship between Visit and Location tables** – Each visit can be tagged to **one and only one** Location, but one location can have **one or many** visits.
 - Relationship between Visit and Session tables** – Each session can be tagged to **one or many** visits, but each visit can be from **one and only one** session.

Entity-Relationship Diagram



Physical Database

Assumptions/Notes About Data Set

We have deleted 17 columns from the existing dataset. These columns had data which had values like “not in data” from which we aren’t able to infer anything. Dropping these columns helped reduce the data size and increase the ease/efficiency in analyzing/querying the data.

We dropped the column isMobile as the column deviceCategory contains 3 values Desktop, Mobile and Tablet while isMobile has false for desktop and is true otherwise. This can be categorized in the query itself while getting the output if needed.

1. mobileDeviceBranding
2. mobileDeviceModel
3. mobileInputSelector
4. mobileDeviceInfo
5. mobileDeviceMarketingName
6. cityID
7. latitude
8. longitude
9. networkLocation
10. BrowserVersion
11. BrowserSize
12. OperatingSystemVersion
13. flashVersion
14. language
15. screencolors
16. screenResol
17. Ismobile

Screen shot of Physical Database objects

Screenshot 1: MySQL Workbench - mydb Schema

The screenshot shows the MySQL Workbench interface with the "mydb" schema selected. In the SQL Editor tab, the following SQL code is run:

```
## location table
desc location;
```

The Result Grid displays the structure of the "location" table:

Field	Type	Null	Key	Default	Extra
UniqueLocationID	int(100)	NO	PRI	NULL	
continent	varchar(255)	YES		NULL	
subContinent	varchar(255)	YES		NULL	
country	varchar(255)	YES		NULL	
region	varchar(255)	YES		NULL	
city	varchar(255)	YES		NULL	
metro	varchar(255)	YES		NULL	

The Session Output pane shows the execution history:

#	Time	Action	Message	Duration / Fetch
99	21:01:37	select count(*) from visit	LIMIT 0, 1000	0.172 sec / 0.000 sec
100	21:01:48	desc visitor		0.016 sec / 0.000 sec
101	21:01:53	select count(*) from visitor	LIMIT 0, 1000	0.406 sec / 0.000 sec
102	21:04:47	select max(hts).min(hts).year,month,day from Visit	LIMIT 0, 1000	0.422 sec / 0.000 sec
103	21:22:37	desc location		0.016 sec / 0.000 sec
104	21:23:06	desc location		0.000 sec / 0.000 sec

Screenshot 2: MySQL Workbench - mydb Schema

The screenshot shows the MySQL Workbench interface with the "mydb" schema selected. In the SQL Editor tab, the following SQL code is run:

```
## session table
desc session;
```

The Result Grid displays the structure of the "session" table:

Field	Type	Null	Key	Default	Extra
UniqueSessionID	int(100)	NO	PRI	NULL	
sessionid	varchar(255)	YES		NULL	
operatingSystem	varchar(255)	YES		NULL	
deviceCategory	varchar(255)	YES		NULL	
browser	varchar(255)	YES		NULL	

The Session Output pane shows the execution history:

#	Time	Action	Message	Duration / Fetch
100	21:01:48	desc visitor		0.016 sec / 0.000 sec
101	21:01:53	select count(*) from visitor	LIMIT 0, 1000	0.406 sec / 0.000 sec
102	21:04:47	select max(hts).min(hts).year,month,day from Visit	LIMIT 0, 1000	0.422 sec / 0.000 sec
103	21:22:37	desc location		0.016 sec / 0.000 sec
104	21:23:06	desc location		0.000 sec / 0.000 sec
105	21:25:13	desc session		0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL Router × MySQL Model (DB final.mwb) × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS SQL File 5* SQL File 6*

Tables: location session visit visitor Views Stored Procedures Functions

Schemas: mydb

SQL File 5*:

```
1 ## visit table;
2 • desc visit;
```

Result Grid | Filter Rows: Export: Wrap Cell Content: □

Field	Type	Null	Key	Default	Extra
UniqueVisitID	int(100)	NO	PRI	HULL	
visitID	int(100)	YES	HULL		
visitNumber	int(100)	YES	HULL		
visitStartTime	int(100)	YES	HULL		
hits	int(100)	YES	HULL		
visits	int(100)	YES	HULL		
newVisits	varchar(255)	YES	HULL		
pageViews	varchar(255)	YES	HULL		
Visitor_UniqueVisitorID	int(100)	NO	MUL	HULL	
Session_UniqueSessionID	int(100)	NO	MUL	HULL	
Location_UniqueLocationID	int(100)	NO	MUL	HULL	
year	int(100)	YES	HULL		
month	int(100)	YES	HULL		
day	int(100)	YES	HULL		

Result 3 ×

Action Output

#	Time	Action	Message	Duration / Fetch
101	21:01:53	select count(*) from visit LIMIT 0, 1000	1 row(s) returned	0.406 sec / 0.000 sec
102	21:04:47	select max(hits),min(hits),year,month,day from Visit LIMIT 0, 1000	1 row(s) returned	0.422 sec / 0.000 sec
103	21:22:37	desc location	7 row(s) returned	0.016 sec / 0.000 sec
104	21:23:06	desc location	7 row(s) returned	0.000 sec / 0.000 sec
105	21:25:13	desc session	5 row(s) returned	0.000 sec / 0.000 sec
106	21:25:55	desc visit	14 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

MySQL Workbench

Local instance MySQL Router × MySQL Model (DB final.mwb) × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS SQL File 5* SQL File 6*

Tables: location session visit visitor Views Stored Procedures Functions

Schemas: mydb

SQL File 6*:

```
1 ## visitor table;
2 • desc visitor;
```

Result Grid | Filter Rows: Export: Wrap Cell Content: □

Field	Type	Null	Key	Default	Extra
UniqueVisitorID	int(100)	NO	PRI	HULL	
fullvisitorid	varchar(255)	YES	HULL		
campaign	varchar(255)	YES	HULL		
source	varchar(255)	YES	HULL		
medium	varchar(255)	YES	HULL		
keywordwords	varchar(255)	YES	HULL		
channelgrouping	varchar(255)	YES	HULL		
networkdomain	varchar(255)	YES	HULL		
socialEngagementType	varchar(255)	YES	HULL		

Result 4 ×

Action Output

#	Time	Action	Message	Duration / Fetch
102	21:04:47	select max(hits),min(hits),year,month,day from Visit LIMIT 0, 1000	1 row(s) returned	0.422 sec / 0.000 sec
103	21:22:37	desc location	7 row(s) returned	0.016 sec / 0.000 sec
104	21:23:06	desc location	7 row(s) returned	0.000 sec / 0.000 sec
105	21:25:13	desc session	5 row(s) returned	0.000 sec / 0.000 sec
106	21:25:55	desc visit	14 row(s) returned	0.000 sec / 0.000 sec
107	21:27:26	desc visitor	9 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

Data in the Database

Table Name	Primary Key	Foreign Key	# of Rows in Table
Visit	UniqueVisitID	Visitor_UniqueVisitorID Session_UIQUESESSIONID Location_UniqueLocationID	14

Table Name	Primary Key	Foreign Key	# of Rows in Table
Visitor	UniqueVisitorID		9

Table Name	Primary Key	Foreign Key	# of Rows in Table
Location	UniqueLocationID		7

Table Name	Primary Key	Foreign Key	# of Rows in Table
Session	UniqueSessionID		5

Normalization:

- We have reduced the data to 3NF form
- 1NF was achieved by ensuring that each column is atomic valued with unique names
- 2NF was achieved by ensuring that the database does not contain any
- 3NF was achieved by ensuring that there are no transitive dependencies in the database

SQL Queries

Query 1

Which user had the minimum number of visits and when?

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

272 rows

Translation

Select the user who had the minimum number of visits and when these visits were.

Screen Shot of SQL Query and Results

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query is:

```
select distinct(UniqueVisitID), count(UniqueVisitID), year, month, day from visit group by year, month, day order by count(UniqueVisitID);
```

The results grid displays 272 rows of data:

UniqueVisitID	count(UniqueVisitID)	year	month	day
192	1495	2017	12	31
813	1768	2017	12	30
278	1796	2017	8	5
472	1810	2018	1	1
586	1843	2018	3	31
516	1900	2018	4	14
2006	1902	2017	12	29
332	1920	2018	4	21
22	1931	2018	1	6
160	1940	2017	12	24
1130	1966	2017	9	2
838	1972	2018	4	28

Query 2

Question

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

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

2

Translation

Select the operating system which has the maximum number of repetitions in the column for visitors using mobile devices.

Selecting the count of s.OperatingSystem, s.deviceCategory and creating joins from visitor V1 to visit v2, visit v2 to sessions. Grouping the data by s.deviceCategory will give the most popular operating systems.

Screen Shot of SQL Query and Results

The screenshot shows the MySQL Workbench interface with the following details:

- Top Bar:** MySQL Workbench, Local instance MySQL Router, MySQL Model (DB final.mwb), EER Diagram.
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (location, session, visit).
- SQL Editor:** SQL File 5*, SQL File 6*, SQL File 7*. The query is:7
8 • select count(*), s.OperatingSystem, s.deviceCategory as Countofoperating from
Visitor v1 join visit v2
ON v1.UniqueVisitorID = v2.Visitor_UniqueVisitorID
JOIN Session s ON
v2.Session_UniqueSessionID = s.UniqueSessionID
where s.devicecategory IN ('mobile', 'tablet')
group by s.deviceCategory;
17
-- select max(hits),min(hits),year,month,day from visit;
- Result Grid:** Shows the results of the query:| | count(*) | OperatingSystem | Countofoperating |
| --- | --- | --- | --- |
| ▶ | 262611 | iOS | mobile |
| ▶ | 34973 | iOS | tablet |
- Table Browser:** Table: location. Shows columns: UniqueLocationID, continent, subContinent, country, region, city, metro.
- Output Window:** Result 59 x. Shows the execution log:| # | Time | Action | Message | Duration / Fetch |
| --- | --- | --- | --- | --- |
| 197 | 23:27:20 | select count(*) as Countofoperating from Visitor v1 join visit v2 O... | 2 row(s) returned | 7.891 sec / 0.000 sec |
| 198 | 23:27:50 | select count(*), s.OperatingSystem as Countofoperating from Vis... | 2 row(s) returned | 7.938 sec / 0.000 sec |
| 199 | 23:28:23 | select count(session.operatingSystem),session.operatingSystem,... | Error Code: 1054. Unknown column 'visit.UniqueVisitorID' in field list' | 0.000 sec |
| 200 | 23:28:39 | select count(*), s.OperatingSystem, s.deviceCategory as Countof... | 2 row(s) returned | 7.906 sec / 0.000 sec |
- System Bar:** Type here to search, Task View, Start, File Explorer, Taskbar icons, System tray (23:28, ENG, 28-10-2018).

Query 3

Question

Visitors from which country visted the store more than once?

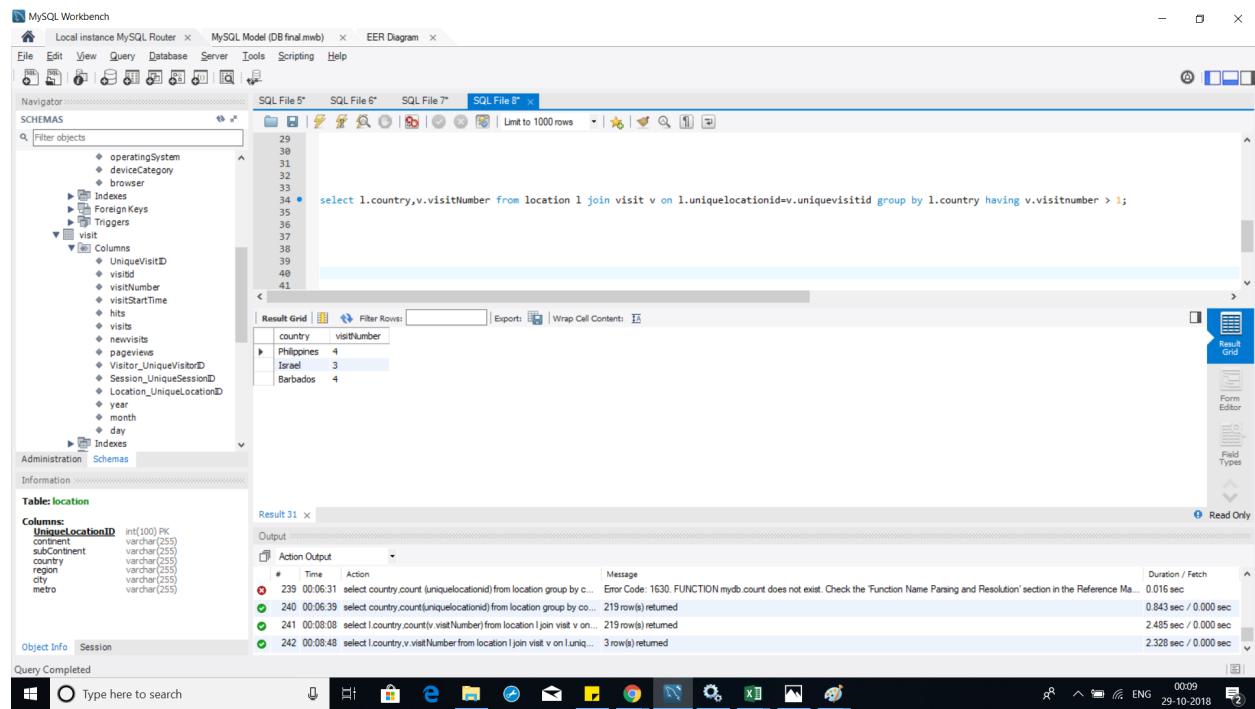
Notes/Comments About SQL Query and Results (Include # of Rows in Result)

3 row

Translation

Selecting 1.country, v.visitnumber from locations and creating a join between 1.uniquelocationid and uniquevisited and grouping by country

Screen Shot of SQL Query and Results



The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following query:

```
29
30
31
32
33
34 • select l.country,v.visitNumber from location l join visit v on l.uniquelocationid=v.uniquevisitid group by l.country having v.visitnumber > 1;
35
36
37
38
39
40
41
```

The results grid displays the following data:

country	visitNumber
Philippines	4
Israel	3
Barbados	4

The status bar at the bottom right shows the results were read-only.

Query 4

Question

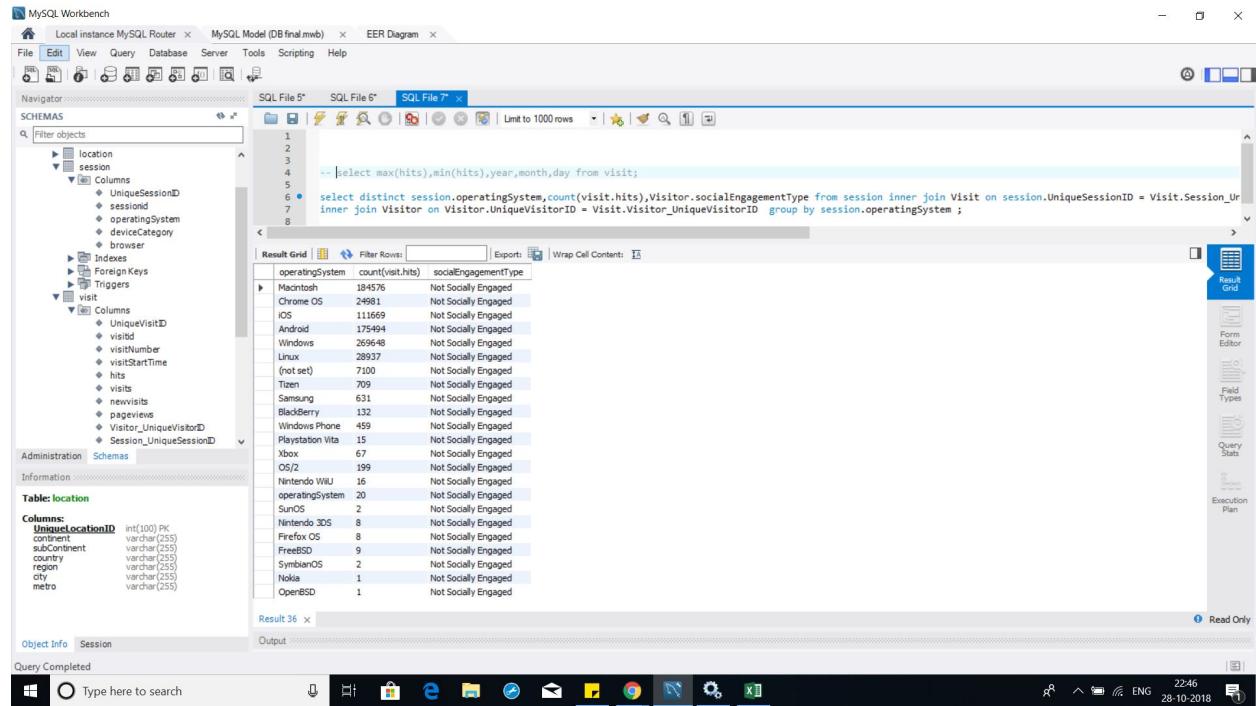
Users of which operating system were the least socially engaged?

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

Translation

Distinct values of operatingSystem, with counts of visitor, visit.hits and socialEngagementType and creating inner joins between , UniqueSessionID and Visit.Session_Under, Visitor.UniqueVisitorID and Visit.VisitorID and grouping the data by session.operatingSystem will give the results.

Screen Shot of SQL Query and Results



The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following query:

```
1
2
3
4 --| select max(hits),min(hits),year,month,day from visit;
5
6 • select distinct session.operatingSystem,count(visit.hits),Visitor.socialEngagementType from session inner join Visit on session.UniqueSessionID = Visit.Session_Under
inner join Visitor on Visitor.UniqueVisitorID = Visit.Visitor_UniqueVisitorID group by session.operatingSystem ;
```

The results grid displays the following data:

operatingSystem	count(visit.hits)	socialEngagementType
Macintosh	184576	Not Socially Engaged
Chrome OS	24981	Not Socially Engaged
iOS	11669	Not Socially Engaged
Android	175494	Not Socially Engaged
Windows	269648	Not Socially Engaged
Linux	28937	Not Socially Engaged
(not set)	7100	Not Socially Engaged
Tizen	709	Not Socially Engaged
Samsung	631	Not Socially Engaged
BlackBerry	132	Not Socially Engaged
Windows Phone	459	Not Socially Engaged
Playstation Vita	15	Not Socially Engaged
Xbox	67	Not Socially Engaged
OS/2	199	Not Socially Engaged
Nintendo WII	16	Not Socially Engaged
operatingSystem	20	Not Socially Engaged
SunOS	2	Not Socially Engaged
Nintendo 3DS	8	Not Socially Engaged
Firefox OS	8	Not Socially Engaged
FreeBSD	9	Not Socially Engaged
SymbianOS	2	Not Socially Engaged
Nokia	1	Not Socially Engaged
OpenBSD	1	Not Socially Engaged

Query 5

Question

Provide a breakdown of unique visitors by mobile vs non-mobile users

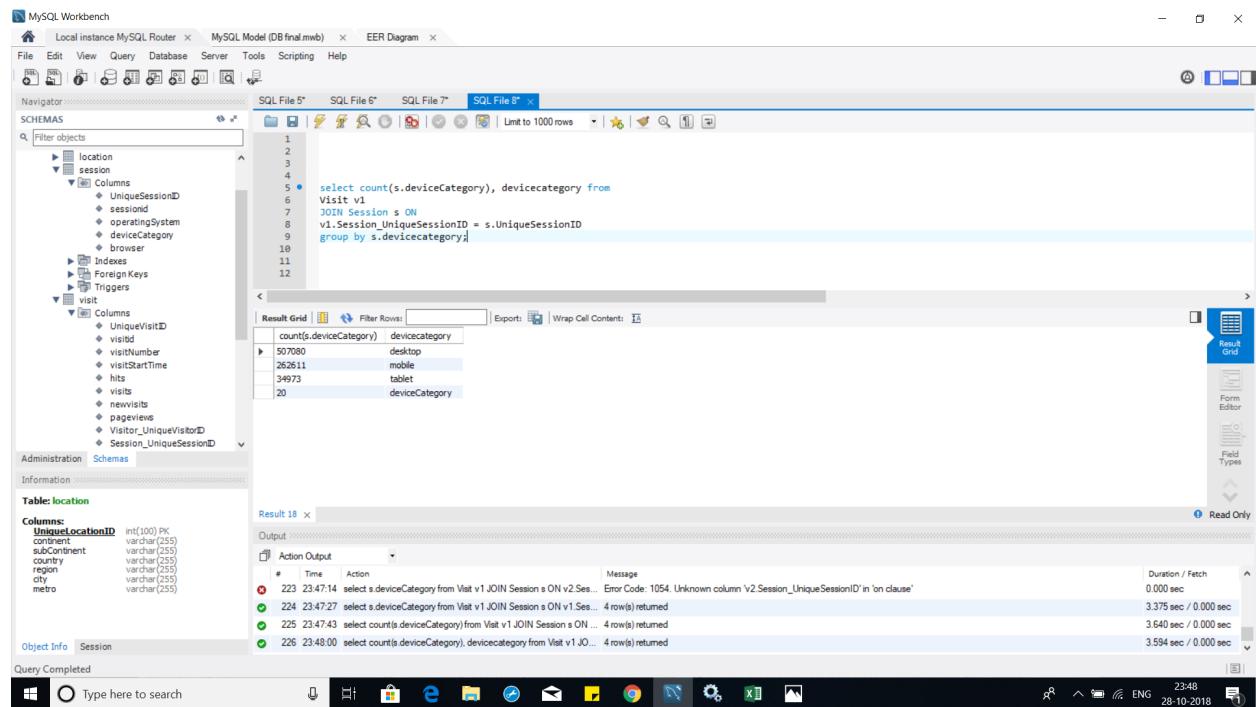
Notes/Comments About SQL Query and Results (Include # of Rows in Result)

4 rows

Translation

Using unique visitor ID to find the mobile and non-mobile users by using device category as not equal to 'desktop'

Screen Shot of SQL Query and Results



The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following query:

```
1
2
3
4
5  select count(s.deviceCategory), devicecategory from
6  Visit v1
7  JOIN Session s ON
8  v1.Session_UniqueSessionID = s.UniqueSessionID
9  group by s.devicecategory;
```

The results grid displays the following data:

count(s.deviceCategory)	devicecategory
507080	desktop
262611	mobile
34973	tablet
20	deviceCategory

The bottom pane shows the execution log for the query:

Action	Time	Message	Duration / Fetch
select a.deviceCategory from Visit v1 JOIN Session s ON v2.Session_UniqueSessionID = s.UniqueSessionID	223 23:47:14	Error Code: 1054. Unknown column 'v2.Session_UniqueSessionID' in 'on clause'	0.000 sec
select a.deviceCategory from Visit v1 JOIN Session s ON v1.Session_UniqueSessionID = s.UniqueSessionID	224 23:47:27	4 row(s) returned	3.375 sec / 0.000 sec
select count(a.deviceCategory) from Visit v1 JOIN Session s ON v1.Session_UniqueSessionID = s.UniqueSessionID	225 23:47:43	4 row(s) returned	3.640 sec / 0.000 sec
select count(a.deviceCategory), devicecategory from Visit v1 JOIN Session s ON v1.Session_UniqueSessionID = s.UniqueSessionID group by s.devicecategory	226 23:48:00	4 row(s) returned	3.594 sec / 0.000 sec

Query 6

Question

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

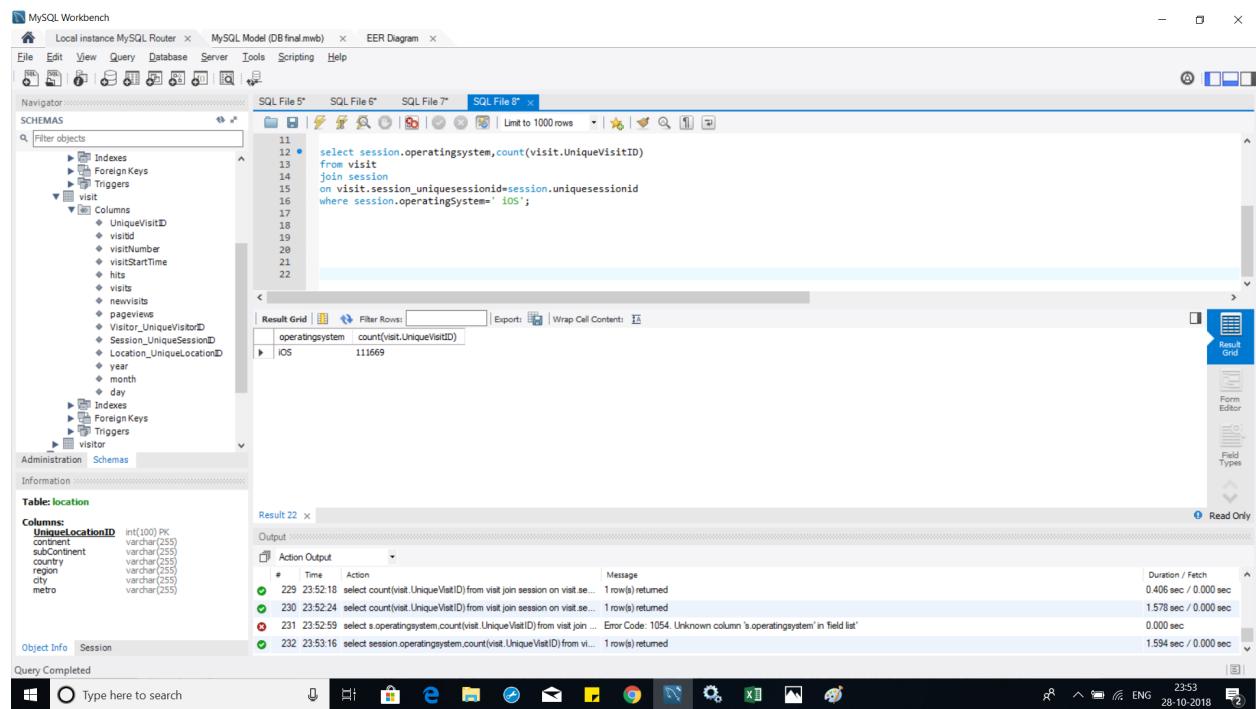
Notes/Comments About SQL Query and Results (Include # of Rows in Result)

1

Translation

Using OperatingSystem and UniquevisitID, users accessing the store can be found

Screen Shot of SQL Query and Results



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL Router, MySQL Model (DB final.mwb), EER Diagram.
- SQL Editor:** SQL File 8* contains the following query:

```
11 select session.operatingsystem, count(visit.UniqueVisitID)
12   from visit
13     join session
14       on visit.session_uniquesessionid=session.uniquesessionid
15      where session.operatingSystem='iOS';
16
17
18
19
20
21
22
```
- Result Grid:** Shows the output of the query:

operatingsystem	count(visit.UniqueVisitID)
iOS	111669
- Output Tab:** Action Output table showing the execution history of the query:

#	Time	Action	Message	Duration / Fetch
229	23:52:18	select count(visit.UniqueVisitID) from visit join session on visit.se...	1 row(s) returned	0.406 sec / 0.000 sec
230	23:52:24	select count(visit.UniqueVisitID) from visit join session on visit.se...	1 row(s) returned	1.578 sec / 0.000 sec
231	23:52:59	select s.operatingsystem,count(v.visit.UniqueVisitID) from visit join...	Error Code: 1054. Unknown column 's.operatingsystem' in 'field list'	0.000 sec
232	23:53:16	select session.operatingsystem,count(v.visit.UniqueVisitID) from vi...	1 row(s) returned	1.594 sec / 0.000 sec
- System Bar:** Shows the taskbar with various icons and the date/time: 28-10-2018 23:53.

Query 7

Question

Which user generated the least amount of hits and when?

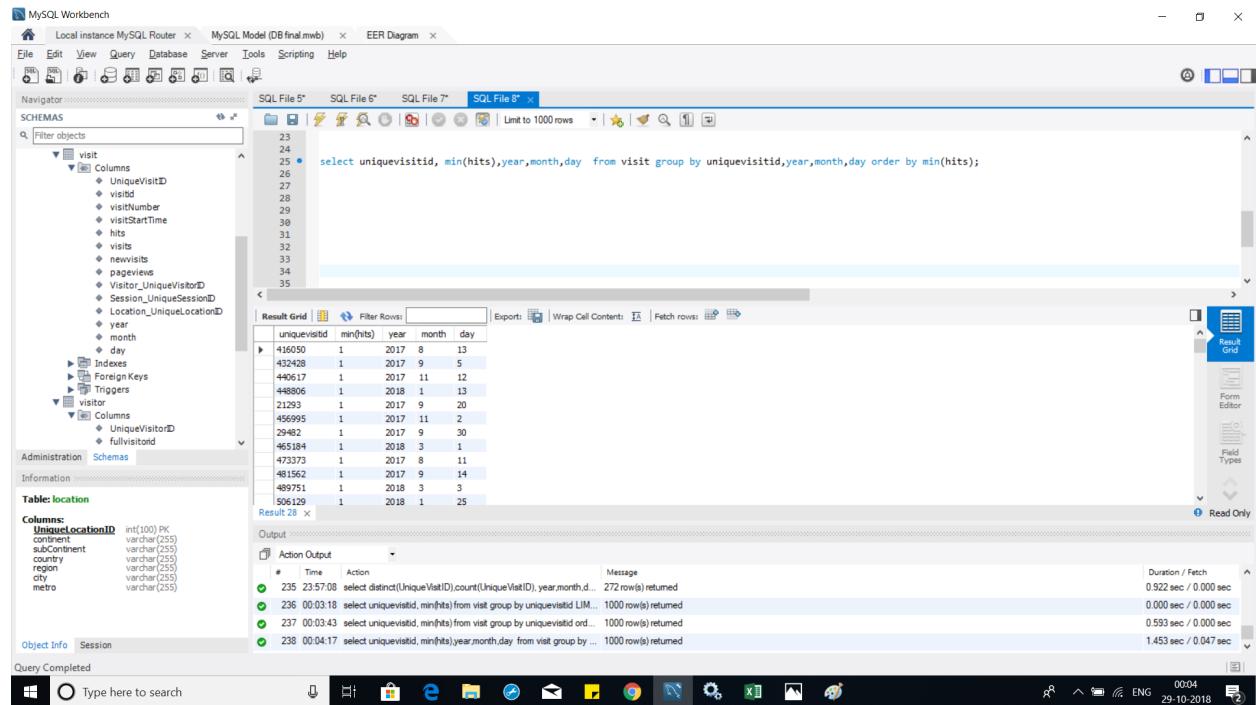
Notes/Comments About SQL Query and Results (Include # of Rows in Result)

1000

Translation

Using uniquevisitid and the function 'min', and group by by year, month, day and order by minimum hits

Screen Shot of SQL Query and Results



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL Router, MySQL Model (DB final.mwb), EER Diagram.
- Toolbar:** Standard MySQL Workbench toolbar.
- Navigator:** Shows the database schema with tables like visit, year, month, day, visitor, and location.
- SQL Editor:** Contains the following SQL query:

```
23
24
25 • select uniquevisitid, min(hits),year,month,day from visit group by uniquevisitid,year,month,day order by min(hits);
26
27
28
29
30
31
32
33
34
35
```
- Result Grid:** Displays the results of the query in a tabular format. The columns are uniquevisitid, min(hits), year, month, and day. The data shows 1000 rows of results, with the first few rows being:

uniquevisitid	min(hits)	year	month	day
416050	1	2017	8	13
492428	1	2017	9	5
440617	1	2017	11	12
448806	1	2018	1	13
21293	1	2018	9	20
295995	1	2017	11	2
29482	1	2017	1	30
455184	1	2018	3	1
473737	1	2017	8	11
- Output Window:** Shows the execution log with 4 entries, all successful, indicating 1000 rows returned each time.
- Taskbar:** Shows the Windows taskbar with various pinned icons.