Hi Moloco, this is my collection of answers for Q1 Analysis problems. I used SQLite3 for analysis.

Initially, I created a table then import data.

CREATE TABLE moloco2

(

ts TIMESTAMP,

user\_id VARCHAR,

country\_id VARCHAR,

site\_id VARCHAR

)

Q1.

Consider only the rows with country\_id = "BDV" (there are 844 such rows). For each site\_id, we can compute the number of unique user\_id's found in these 844 rows. Which site\_id has the largest number of unique users? And what's the number?

A1.

Site\_id: 5NPAU, 544 unique users

I used aggregate function COUNT and GROUP BY site\_id to see how many distinct user\_id per each website with country\_id ‘BDV’.

SELECT site\_id, COUNT(DISTINCT user\_id)

FROM moloco2

WHERE country\_id = 'BDV'

GROUP BY site\_id

Q2.

Between 2019-02-03 00:00:00 and 2019-02-04 23:59:59, there are four users who visited a certain site more than 10 times. Find these four users & which sites they (each) visited more than 10 times. (Simply provides four triples in the form (user\_id, site\_id, number of visits) in the box below.)

A2.

User\_id | site\_id | COUNT(\*)

LC06C3|N0OTG|25

LC3A59|N0OTG|26

LC3C7E|3POLC|15

LC3C9D|N0OTG|17

Timestamp object is comparable.

I used WHERE statement to limit the data for given time range, used aggregation function COUNT and GROUP BY user\_id and site\_id to display the user visit per each website. Additionally, I used HAVING statement to include the only rows where aggregation (COUNT) value is more than 10.

SELECT user\_id, site\_id, COUNT(\*)

FROM moloco2

WHERE ts > '2019-02-03 00:00:00'

AND ts < '2019-02-04 23:59:59'

GROUP BY user\_id, site\_id

HAVING COUNT(\*) > 10

Q3.

For each site, compute the unique number of users whose last visit (found in the original data set) was to that site. For instance, user "LC3561"'s last visit is to "N0OTG" based on timestamp data. Based on this measure, what are top three sites? (hint: site "3POLC" is ranked at 5th with 28 users whose last visit in the data set was to 3POLC; simply provide three pairs in the form (site\_id, number of users).)

A3.

site\_id | COUNT(m3.user\_id)

5NPAU | 992

N0OTG | 561

QGO3G | 289

I used subqueries to achieve the last visited website.

First subquery uses aggregation MAX function and correlated query (matching user\_id) to computes the latest time each user visit to certain website. Second subquery creates table of each user\_id and user’s latest visited website. Outer query uses aggregation COUNT function and GROUP BY site\_id with the descending order of count of user\_id that visited each site as last.

SELECT m3.site\_id, COUNT(m3.user\_id)

FROM

(SELECT m2.user\_id, m2.site\_id

FROM moloco2 m2

WHERE m2.ts =

(SELECT MAX(m1.ts)

FROM moloco2 m1

WHERE m1.user\_id = m2.user\_id)) m3

GROUP BY m3.site\_id

ORDER BY COUNT(m3.user\_id) DESC

Q4.

For each user, determine the first site he/she visited and the last site he/she visited based on the timestamp data. Compute the number of users whose first/last visits are to the same website. What is the number?

A4.

1671

Similar approach to previous question: using subqueries and aggregation function on timestamp.

I used MIN and MAX functions to compute the first and last timestamps of each user and use those timestamps with subquery to find the first and last website visited. Then, I made it as a table of unique user with their first and last visited site. Finally, I used the COUNT function with WHERE statement to count the user where first visited site and last visited site are same.

SELECT COUNT(\*)

FROM

(SELECT

DISTINCT m1.user\_id,

(SELECT m2.site\_id

FROM moloco2 m2

WHERE m1.user\_id = m2.user\_id

AND m2.ts =

(SELECT MIN(m3.ts)

FROM moloco2 m3

WHERE m3.user\_id = m1.user\_id)) AS first\_site,

(SELECT m4.site\_id

FROM moloco2 m4

WHERE m1.user\_id = m4.user\_id

AND m4.ts =

(SELECT MAX(m5.ts)

FROM moloco2 m5

WHERE m5.user\_id = m1.user\_id)) AS last\_site

FROM moloco2 m1

WHERE first\_site = last\_site)