Homework | module 2 > week 8 > day 20

Topics covered: WINDOW FUNCTIONS

Standard Exercise:

1. You'll be using the data contained in the `bigquery-public-data.san_francisco.bikeshare_trips` table, but since it's quite a large table, you'll create a copy of it in your personal data set (called DataAnalytics in my case), making sure you only keep observations where the start_date was in 2015. Call the new table sf_bikeshare_2015.
2. What's the average number of trips per bike in 2015?

   ```
   SELECT count(distinct bike_number) as nr_bikes,
    count(distinct trip_id) as nr_trips,
    count(distinct trip_id)/count(distinct bike_number) as avg_trips_bike
   FROM `DataAnalytics.sf_bikeshare_2015`
   ```
3. Which month has the highest number of trips per bike?

   ```
   SELECT extract(month from start_date) as month_start,
    count(distinct bike_number) as nr_bikes,
    count(distinct trip_id) as nr_trips,
    count(distinct trip_id)/count(distinct bike_number) as avg_trips_bike
   FROM `DataAnalytics.sf_bikeshare_2015`
   group by month_start
   order by avg_trips_bike desc
   ```
4. What is the *monthly* average number of trips per bike? (*hint: here the result should be a single number*)

   ```
   WITH table1 as
   (SELECT extract(month from start_date) as month_start,
     count(distinct bike_number) as nr_bikes,
     count(distinct trip_id) as nr_trips,
     count(distinct trip_id)/count(distinct bike_number) as avg_trips_bike
   FROM `DataAnalytics.sf_bikeshare_2015`
   group by month_start
   order by avg_trips_bike desc)
   SELECT avg(avg_trips_bike) as mnth_avg_trips_bike,
    sum(avg_trips_bike*nr_trips)/(select sum(nr_trips) from table1) as
   mnth_avg_trips_bike_weighted
   FROM table1
   ```
5. Which bike has made the most trips?

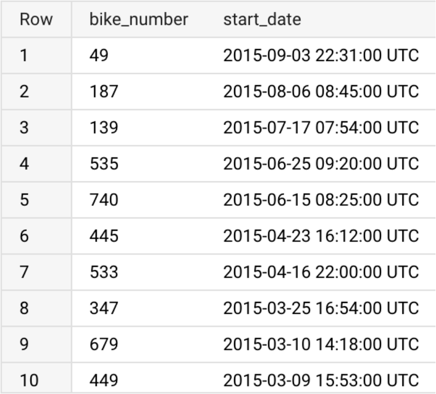
   ```
   SELECT bike_number, count(distinct trip_id) as nr_trips
   FROM `DataAnalytics.sf_bikeshare_2015`
   ```

```
GROUP BY bike_number
order by nr_trips desc
```

6. Write a query that, for each bike_number, keeps only the start_date of the first trip it took in 2015 (your output should be a table where each row has a distinct bike_number and that bike's first start_date for the year); which bike_number had the latest start_date? Which month was it?

```
SELECT bike_number, start_date FROM
 (SELECT *,
   rank() over (partition by bike_number order by start_date) as rank_bike_asc
 FROM `DataAnalytics.sf_bikeshare_2015`
 order by bike_number, start_date)
WHERE rank_bike_asc = 1
ORDER BY start_date desc
```

7. Thinking about the previous question, did you find the result odd? Remember that you have filtered the table keeping only data for the year 2015. You should have found that the first trip of 2015 for bike 49 took place in September (see screenshot below), which is quite strange. Think about what are the possible explanations for this fact and what would you do to investigate and possibly validate your hypothesis?



Row	bike_number	start_date
1	49	2015-09-03 22:31:00 UTC
2	187	2015-08-06 08:45:00 UTC
3	139	2015-07-17 07:54:00 UTC
4	535	2015-06-25 09:20:00 UTC
5	740	2015-06-15 08:25:00 UTC
6	445	2015-04-23 16:12:00 UTC
7	533	2015-04-16 22:00:00 UTC
8	347	2015-03-25 16:54:00 UTC
9	679	2015-03-10 14:18:00 UTC
10	449	2015-03-09 15:53:00 UTC

8. Write a query that shows, for each month, the number of bikes that started their first trip on that month:

```
SELECT extract(month from start_date) as month, count(bike_number) as nr_bikes
FROM
 (SELECT *,
   rank() over (partition by bike_number order by start_date) as rank_bike_asc
 FROM `DataAnalytics.sf_bikeshare_2015`
 order by bike_number, start_date)
WHERE rank_bike_asc = 1
GROUP by month
ORDER BY month
```

9. Does the distribution from the previous question make sense? What is it telling you?

10. You report the information from the previous two points to your boss; he finds it very interesting and wants to investigate the matter further, so he asks you to give him a list containing the (16) bike_number of every bike with a *first start_date* from March or later:

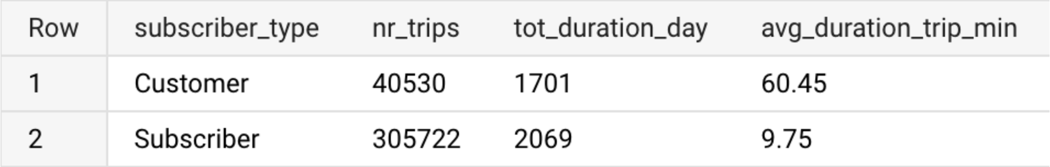
```
SELECT distinct bike_number
FROM
 (SELECT *,
   rank() over (partition by bike_number order by start_date) as rank_bike_asc
 FROM `DataAnalytics.sf_bikeshare_2015`
 order by bike_number, start_date)
WHERE rank_bike_asc = 1 and extract(month from start_date) > 2
```

Advanced Exercise (optional):

1. The DataAnalytics.sf_bikeshare_2015 table contains a column called subscriber_type which, as you can read in the field description of the schema tab, has two categories:
    a. Subscriber = annual or 30-day member
    b. Customer = 24-hour or 3-day member
2. You have been asked to perform an analysis on the different user's behaviour between those two types of customers. Explore the data and write one or more queries that show your results.

```
SELECT subscriber_type,
 count(trip_id) as nr_trips,
 cast(sum(duration_sec)/60/60/24 as INT) as tot_duration_day,
 round((sum(duration_sec)/60)/count(trip_id), 2) as avg_duration_trip_min
FROM `DataAnalytics.sf_bikeshare_2015`
GROUP BY subscriber_type
```

3. In the screenshot below is an example of such an analysis (but feel free to expand and go beyond this); what are these results telling you?



Row	subscriber_type	nr_trips	tot_duration_day	avg_duration_trip_min
1	Customer	40530	1701	60.45
2	Subscriber	305722	2069	9.75

It looks like Subscribers make much more trips (more than 7x as many) than regular Customers; however, although Subscribers use the bikes for a longer time, the gap between the two is narrow. Another interesting fact is the average trip duration: about an hour for regular Customers and only 10 minutes for Subscribers; this may suggest that Subscribers are locals that need the bike for short routine shifts, while the Customers type may very well be tourists or visitors renting the bike for longer routes and sightseeing. (*Hint: it would be interesting to also check if there are any differences in the trip's distance*).

4. As you can see on the San Francisco Municipal Transportation Agency website, there are different fares based on your type of subscription:
    a. $0.20/min for Subscribers
    b. $0.30/min for Customers

   After recreating the table in the screenshot above, add a new column called "revenue" that shows how much revenue was generated by each subscriber_type in 2015. Do you see anything interesting? Think about how would you report this insight to your boss.

```
SELECT subscriber_type,
 count(trip_id) as nr_trips,
 cast(sum(duration_sec)/60/60/24 as INT) as tot_duration_day,
 round((sum(duration_sec)/60)/count(trip_id), 2) as avg_duration_trip_min,
 CASE WHEN subscriber_type = 'Customer' then round(0.3 * sum(duration_sec)/60, 2)
     WHEN subscriber_type = 'Subscriber' then round(0.2 * sum(duration_sec)/60, 2)
 END AS revenue
FROM `DataAnalytics.sf_bikeshare_2015`
GROUP BY subscriber_type
```

5. Using Google Sheets or Data Studio, create a *what-if scenario tool* that allows you to understand *what would need to happen* in order for Subscribers to reach the same level of revenue as Customers.

   Answer here:
   https://datastudio.google.com/u/0/reporting/2758c775-ac27-4052-8096-001c22220854/page/ZZGtC/edit