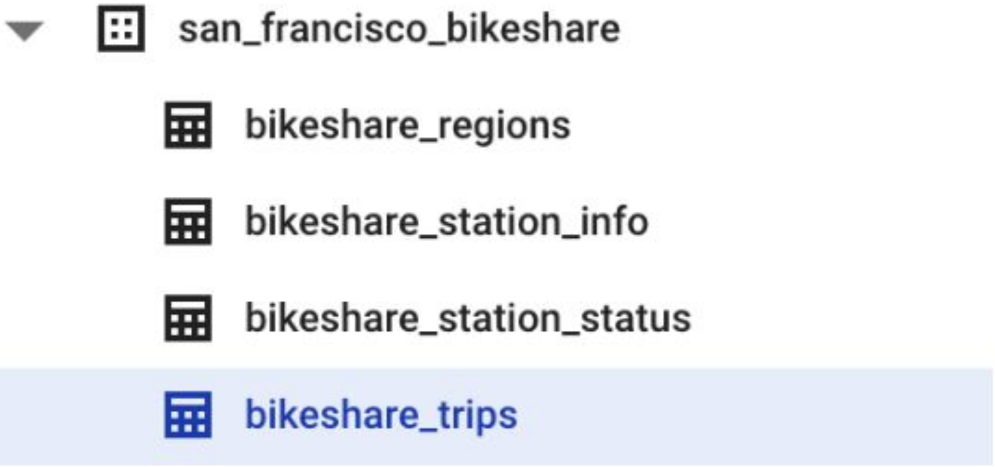
Intermediate & Advanced Assignment

Christopher Stephen

https://console.cloud.google.com/bigquery?sq=758675005227:d5f7c99a73cd433eb4d0cbe1771b4ee9

Overview of dataset

San Francisco Ford GoBike, managed by Motivate, provides the Bay Area’s bike share system. Bike share is a convenient, healthy, affordable, and fun form of transportation. It involves a fleet of specially designed bikes that are locked into a network of docking stations. Bikes can be unlocked from one station and returned to any other station in the system. People use bike share to commute to work or school, run errands, get to appointments, and more. The dataset contains trip data from 2013-2018, including start time, end time, start station, end station, and latitude/longitude for each station.



san_francisco_bikeshare

- bikeshare_regions
- bikeshare_station_info
- bikeshare_station_status
- bikeshare_trips

Then, try to answer the following:

Intermediate

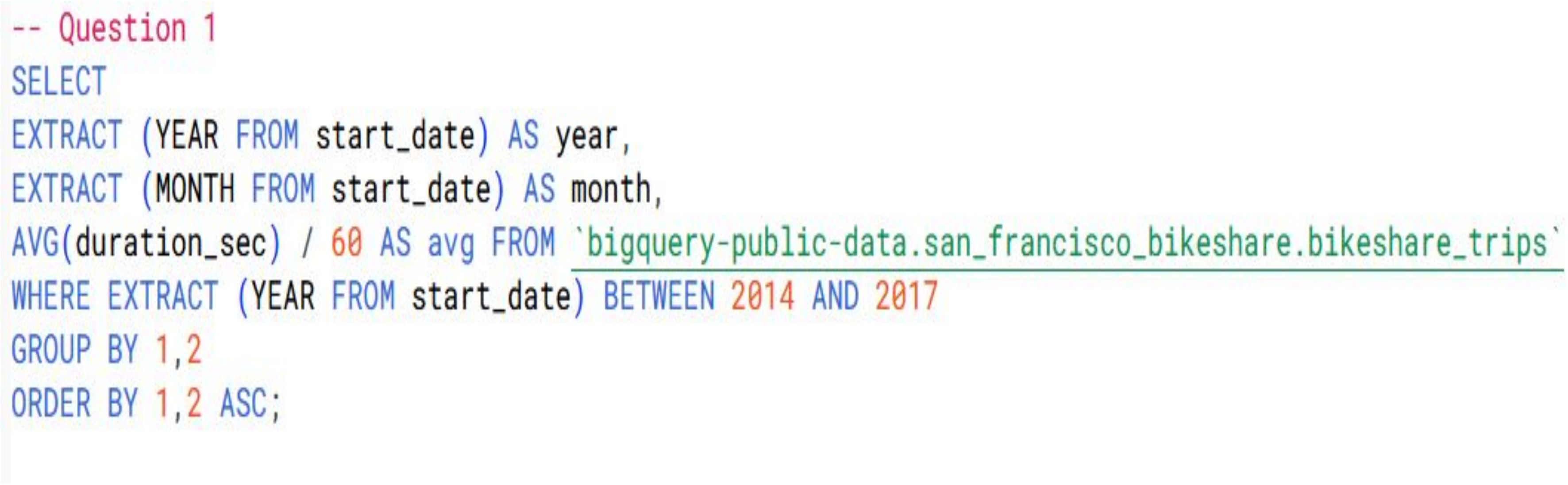
1. **Create a query to get the average amount of duration (in minutes) per month (Skillset: Basic SQL & Formatting and Cleaning in SQL)**

   please use the start date from 2014-2017

   Expected output:

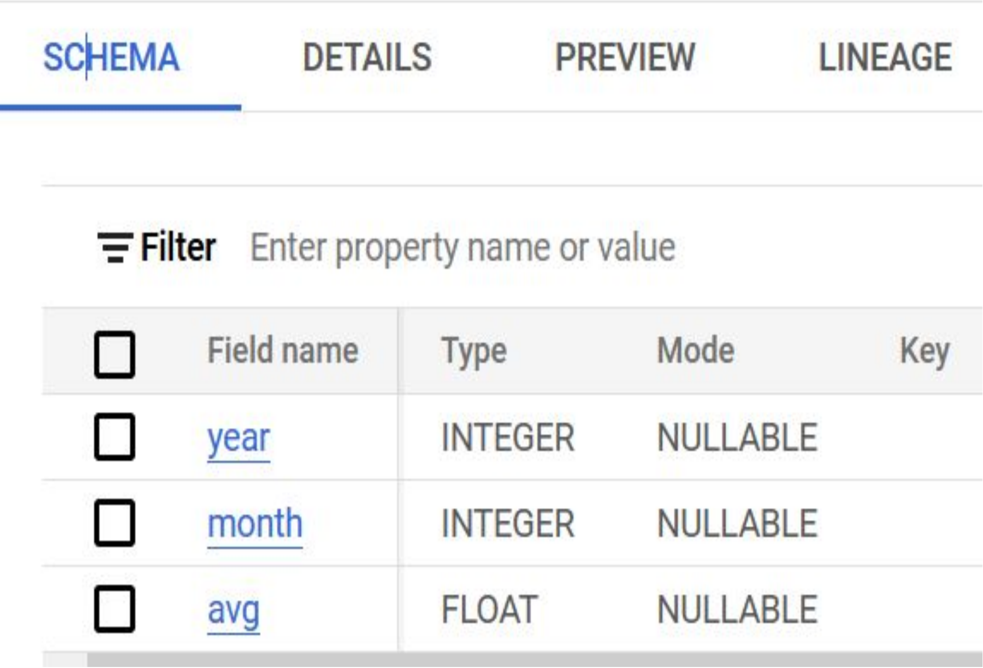
   - Month
   - Average (in minute)

Query

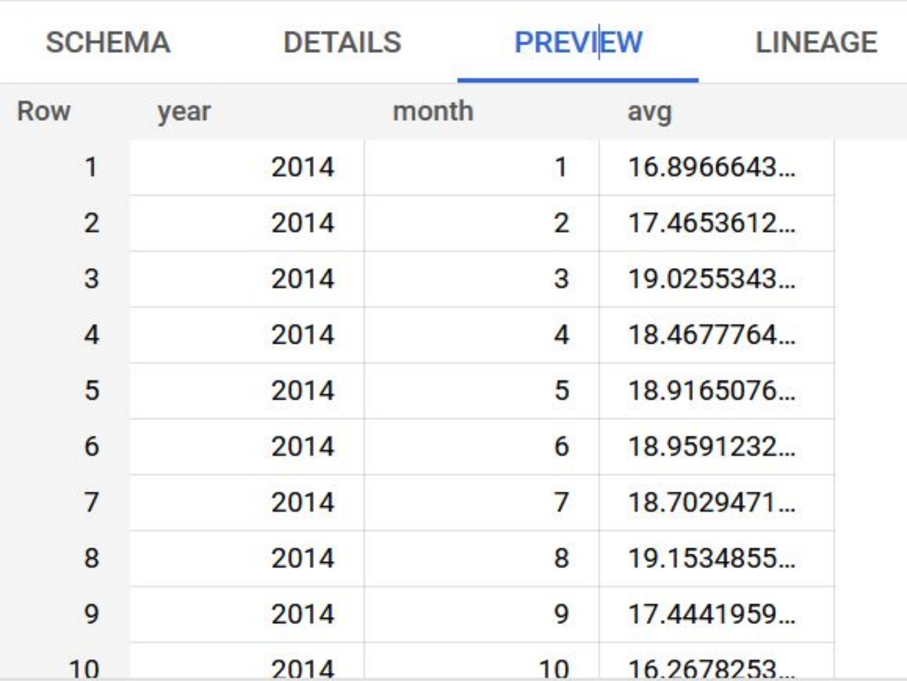


```
-- Question 1
SELECT
EXTRACT (YEAR FROM start_date) AS year,
EXTRACT (MONTH FROM start_date) AS month,
AVG(duration_sec) / 60 AS avg FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
WHERE EXTRACT (YEAR FROM start_date) BETWEEN 2014 AND 2017
GROUP BY 1,2
ORDER BY 1,2 ASC;
```

Table Schema & Preview



□	Field name	Type	Mode	Key
□	year	INTEGER	NULLABLE	
□	month	INTEGER	NULLABLE	
□	avg	FLOAT	NULLABLE	



Row	year	month	avg
1	2014	1	16.8966643...
2	2014	2	17.4653612...
3	2014	3	19.0255343...
4	2014	4	18.4677764...
5	2014	5	18.9165076...
6	2014	6	18.9591232...
7	2014	7	18.7029471...
8	2014	8	19.1534855...
9	2014	9	17.4441959...
10	2014	10	16.2678253...

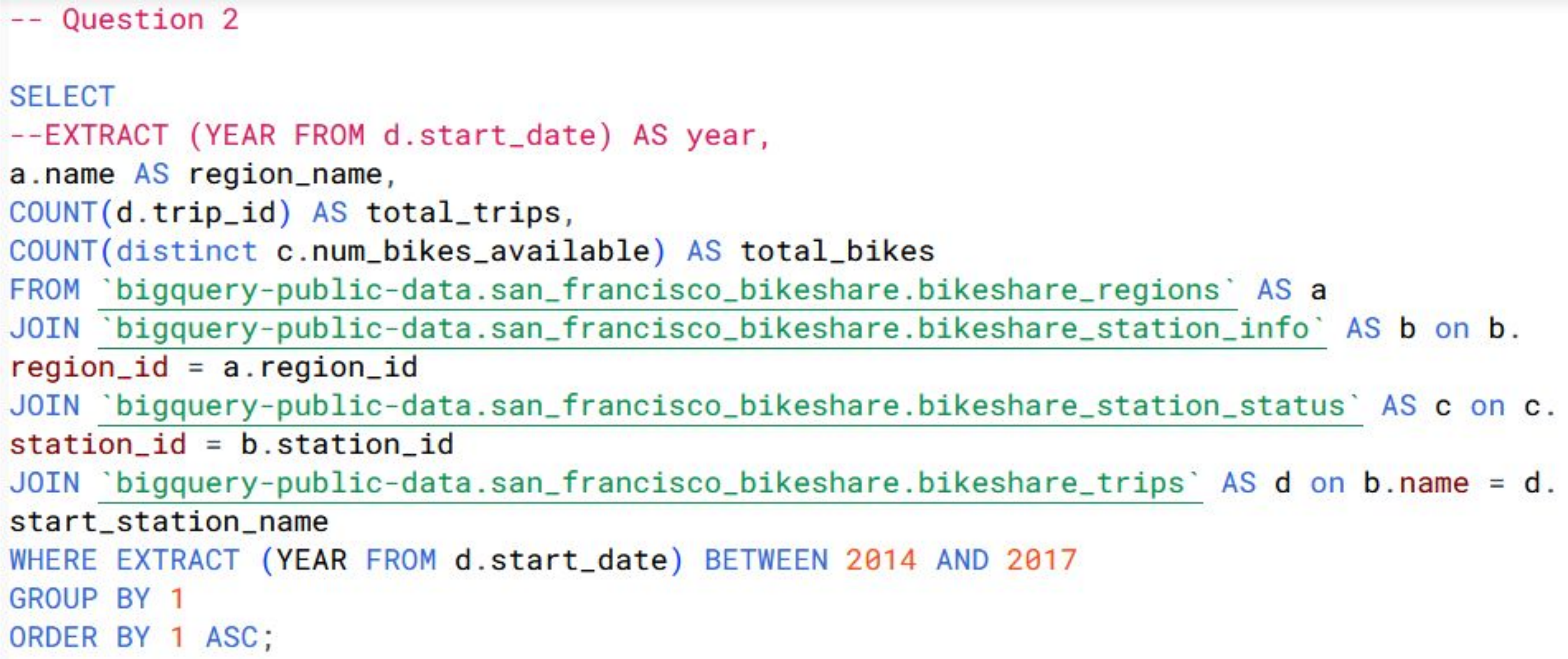
2. **Create a query to get total trips and total number of unique bikes grouped by region name (Skillset: Basic SQL & Joins)**

please use the start date from 2014-2017

Expected output:

- Region Name
- Total Trips
- Total Bikes

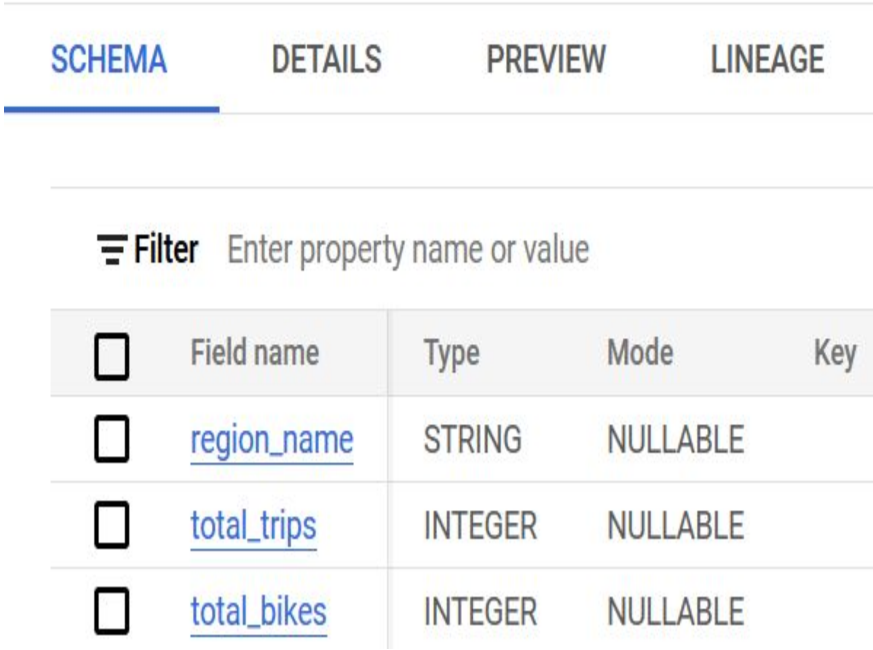
Query



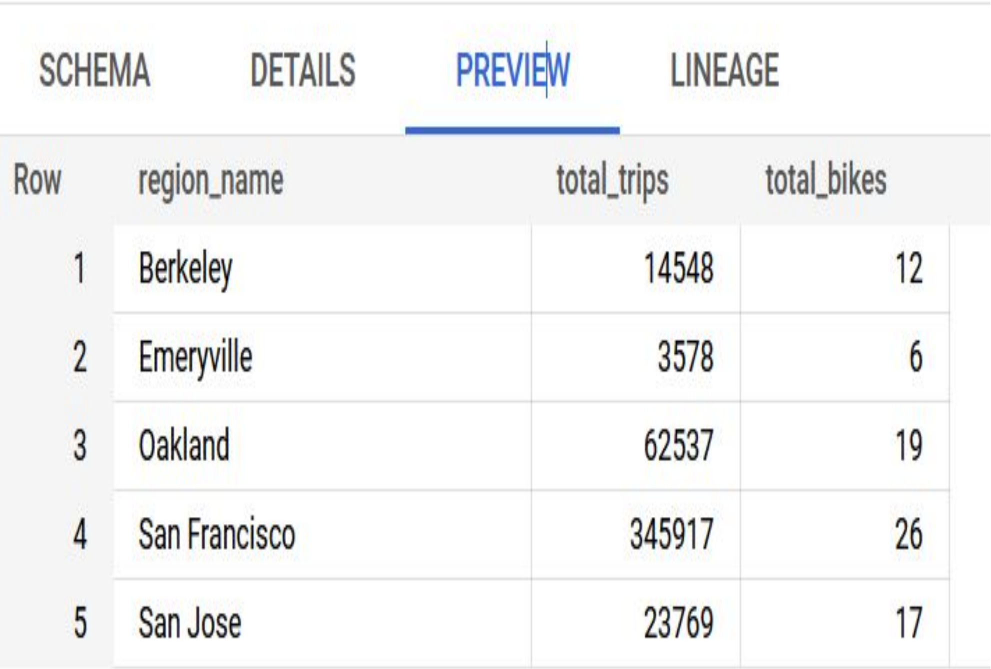
```
-- Question 2

SELECT
--EXTRACT (YEAR FROM d.start_date) AS year,
a.name AS region_name,
COUNT(d.trip_id) AS total_trips,
COUNT(distinct c.num_bikes_available) AS total_bikes
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_regions` AS a
JOIN `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_info` AS b on b.
region_id = a.region_id
JOIN `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_status` AS c on c.
station_id = b.station_id
JOIN `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips` AS d on b.name = d.
start_station_name
WHERE EXTRACT (YEAR FROM d.start_date) BETWEEN 2014 AND 2017
GROUP BY 1
ORDER BY 1 ASC;
```

Table Schema & Preview



□	Field name	Type	Mode	Key
□	region_name	STRING	NULLABLE	
□	total_trips	INTEGER	NULLABLE	
□	total_bikes	INTEGER	NULLABLE	



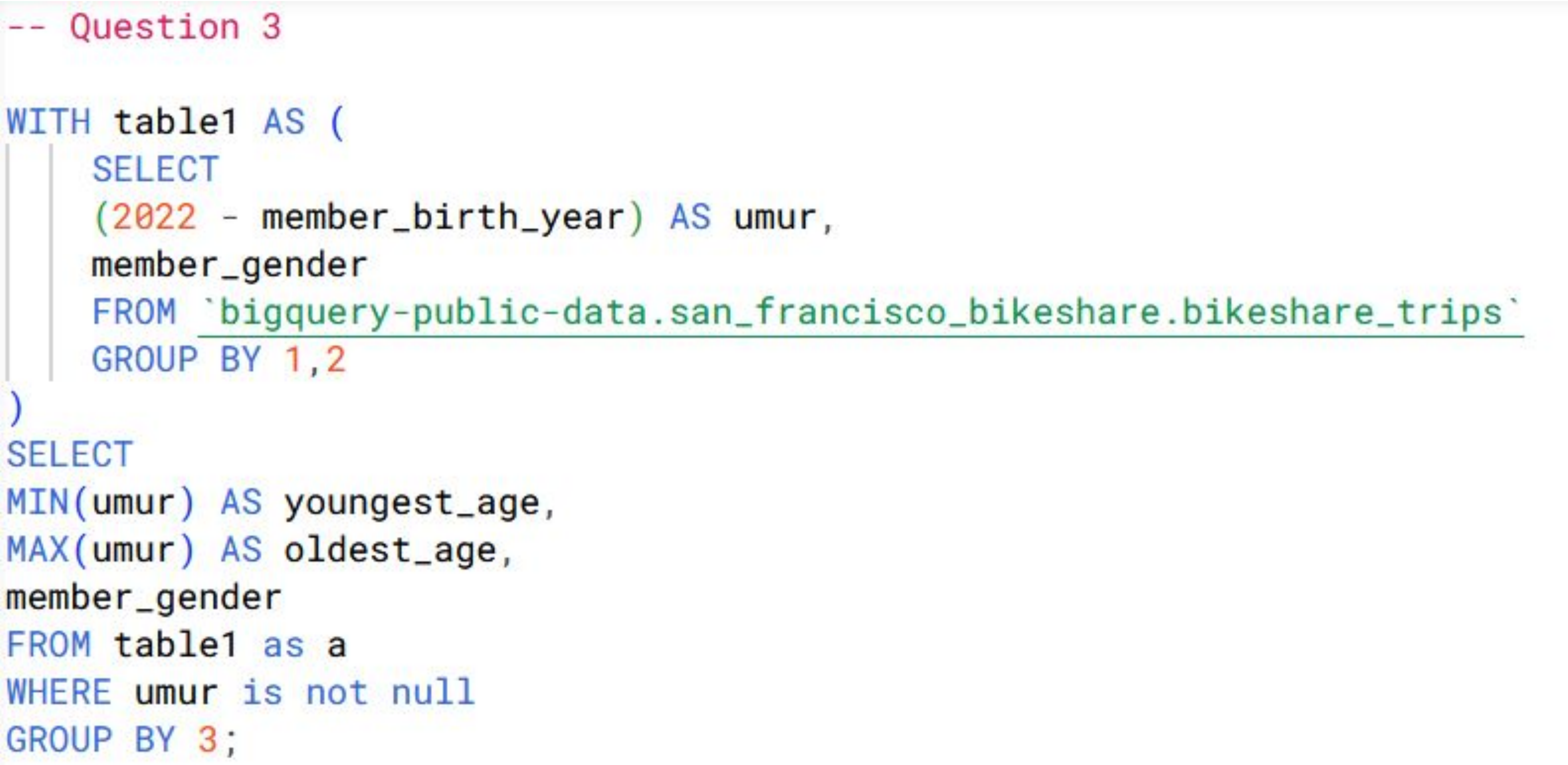
Row	region_name	total_trips	total_bikes
1	Berkeley	14548	12
2	Emeryville	3578	6
3	Oakland	62537	19
4	San Francisco	345917	26
5	San Jose	23769	17

3. **Find the youngest and oldest age of the members, for each gender. Assume this year is 2022. (Skillset: Basic SQL & SQL CTE)**

   Expected output:

   - Gender
   - Youngest Age
   - Oldest Age

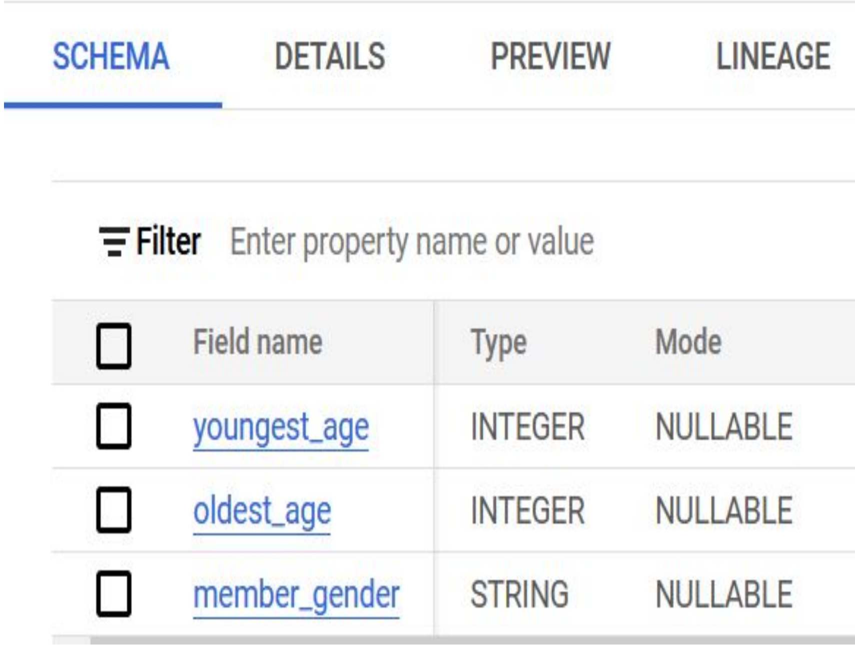
Query



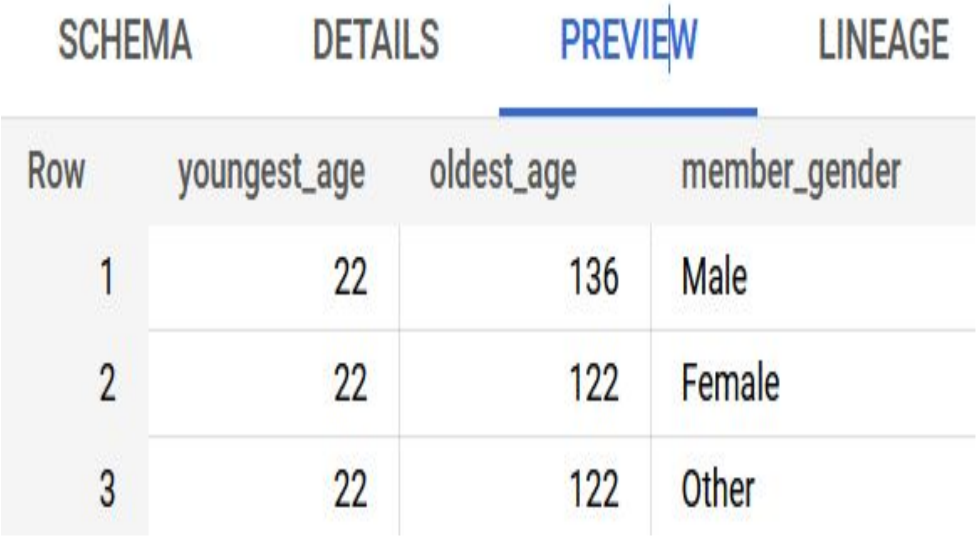
```
-- Question 3

WITH table1 AS (
    SELECT
    (2022 - member_birth_year) AS umur,
    member_gender
    FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
    GROUP BY 1,2
)
SELECT
MIN(umur) AS youngest_age,
MAX(umur) AS oldest_age,
member_gender
FROM table1 as a
WHERE umur is not null
GROUP BY 3;
```

Table Schema & Preview



□	Field name	Type	Mode
□	youngest_age	INTEGER	NULLABLE
□	oldest_age	INTEGER	NULLABLE
□	member_gender	STRING	NULLABLE



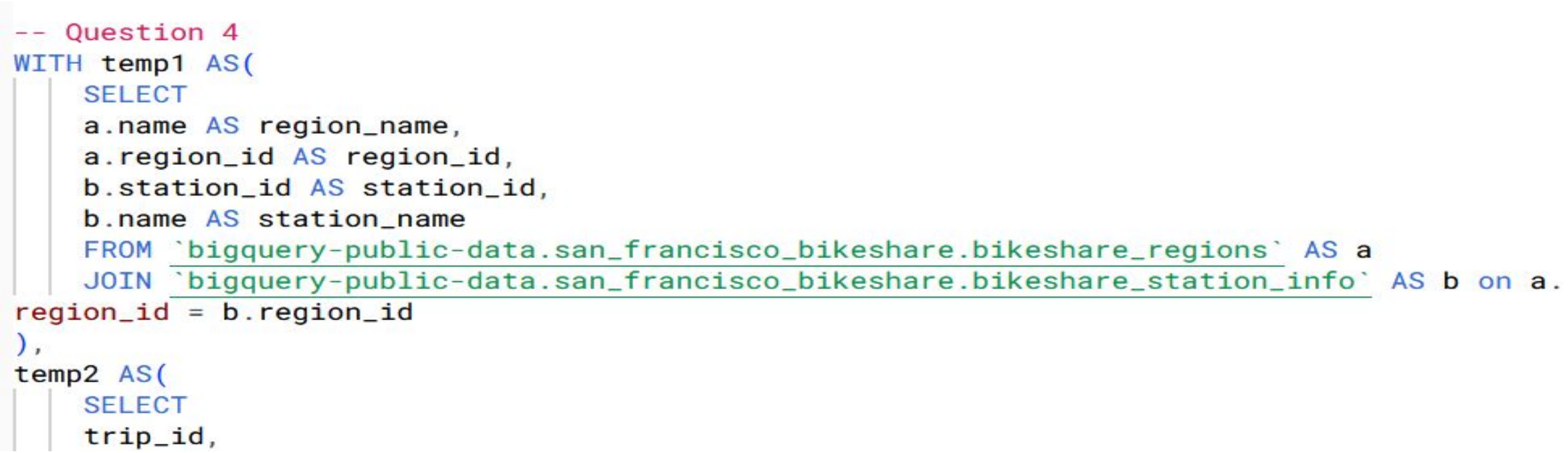
Row	youngest_age	oldest_age	member_gender
1	22	136	Male
2	22	122	Female
3	22	122	Other

4. **Get the latest departure trip in each region with detail below**

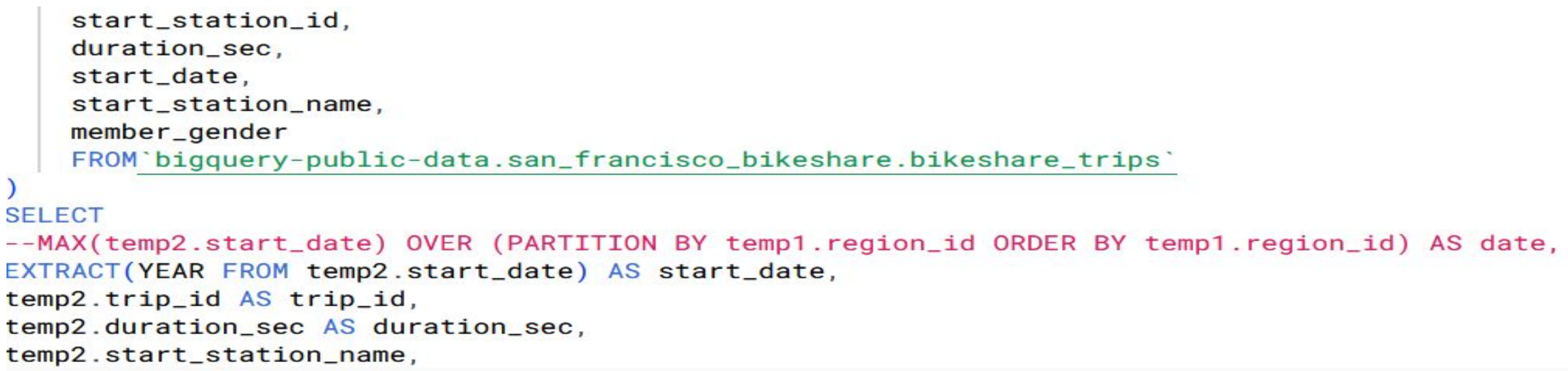
   (Skillset: Window functions, SQL CTE)

   a. trip_id
   b. duration_sec
   c. start_date
   d. start_station_name
   e. Member_gender

Query

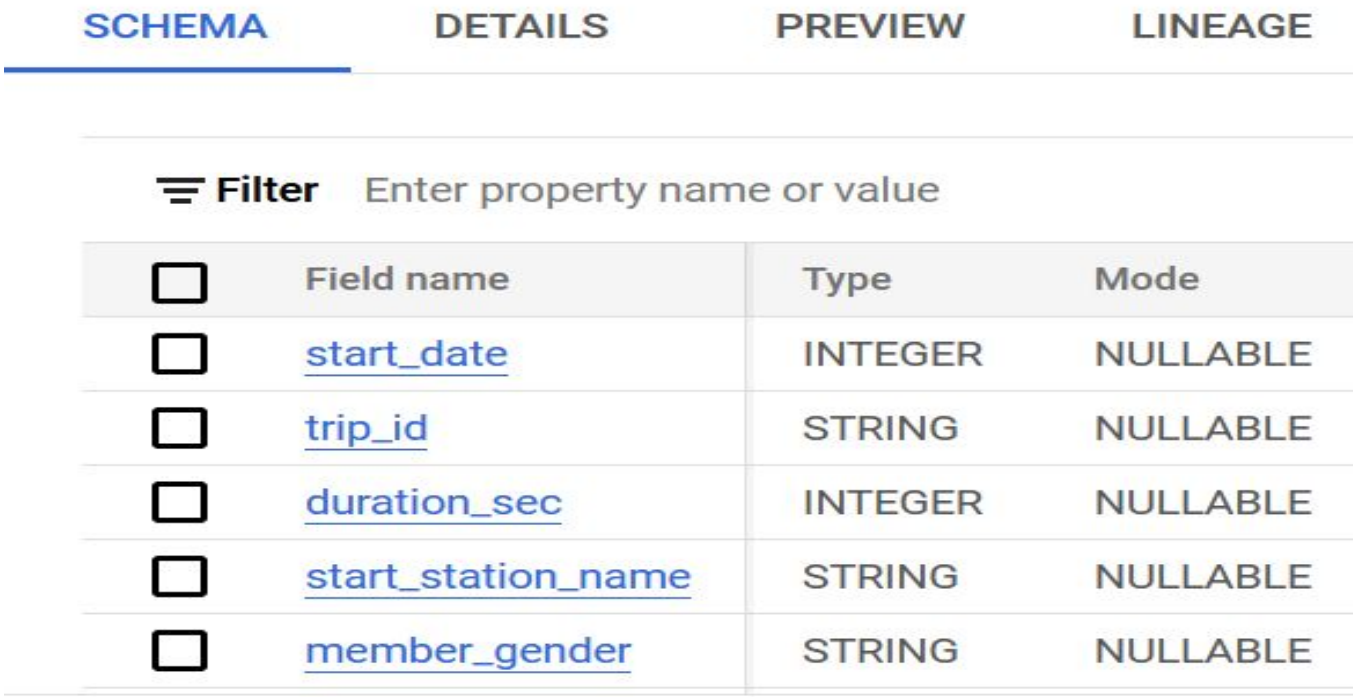


```
-- Question 4
WITH temp1 AS(
    SELECT
    a.name AS region_name,
    a.region_id AS region_id,
    b.station_id AS station_id,
    b.name AS station_name
    FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_regions` AS a
    JOIN `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_info` AS b on a.
region_id = b.region_id
),
temp2 AS(
    SELECT
    trip_id,
```



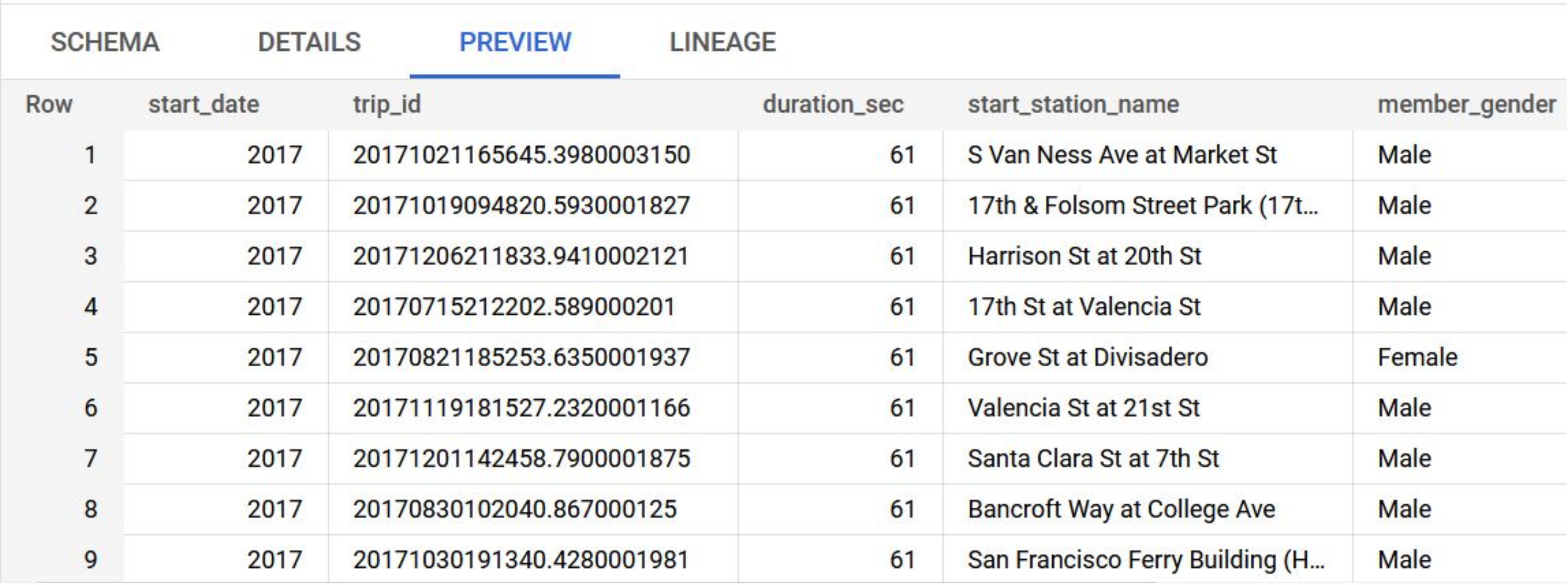
```
    start_station_id,
    duration_sec,
    start_date,
    start_station_name,
    member_gender
    FROM`bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
)
SELECT
--MAX(temp2.start_date) OVER (PARTITION BY temp1.region_id ORDER BY temp1.region_id) AS date,
EXTRACT(YEAR FROM temp2.start_date) AS start_date,
temp2.trip_id AS trip_id,
temp2.duration_sec AS duration_sec,
temp2.start_station_name,
```

Table Schema



□	Field name	Type	Mode
□	start_date	INTEGER	NULLABLE
□	trip_id	STRING	NULLABLE
□	duration_sec	INTEGER	NULLABLE
□	start_station_name	STRING	NULLABLE
□	member_gender	STRING	NULLABLE

Preview



Row	start_date	trip_id	duration_sec	start_station_name	member_gender
1	2017	20171021165645.3980003150	61	S Van Ness Ave at Market St	Male
2	2017	20171019094820.5930001827	61	17th & Folsom Street Park (17t...	Male
3	2017	20171206211833.9410002121	61	Harrison St at 20th St	Male
4	2017	20170715212202.589000201	61	17th St at Valencia St	Male
5	2017	20170821185253.6350001937	61	Grove St at Divisadero	Female
6	2017	20171119181527.2320001166	61	Valencia St at 21st St	Male
7	2017	20171201142458.7900001875	61	Santa Clara St at 7th St	Male
8	2017	20170830102040.867000125	61	Bancroft Way at College Ave	Male
9	2017	20171030191340.4280001981	61	San Francisco Ferry Building (H...	Male

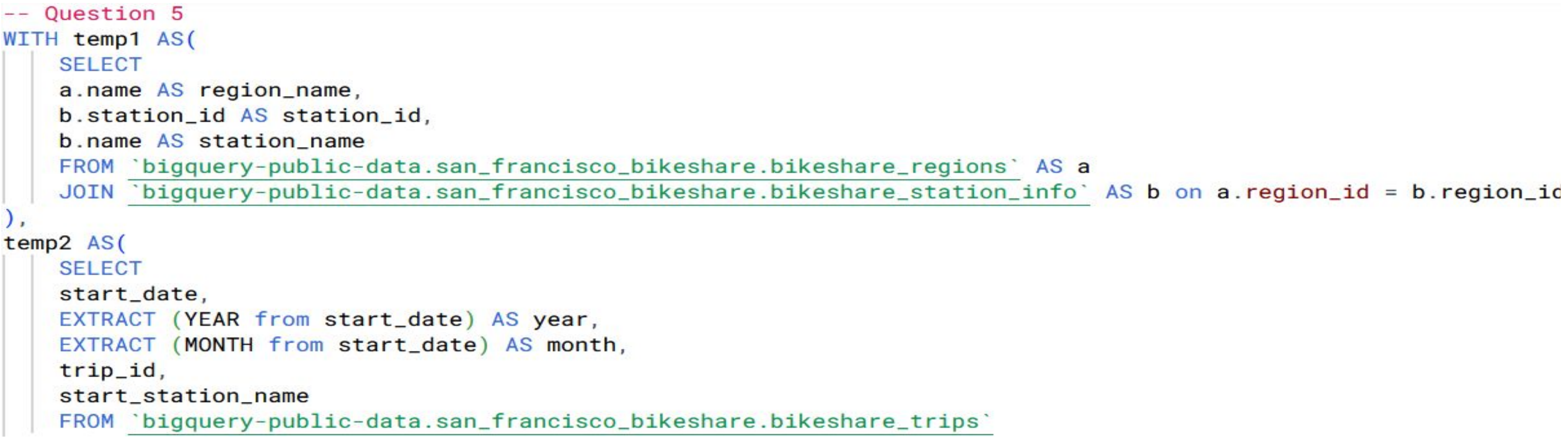
5. **Create a query to get Month to Date of total trips in each region, breakdown by date (Skillset: Basic SQL, Formatting and Cleaning in SQL, Window Function & SQL CTE)**

please use timeframe from November 2017 until December 2017

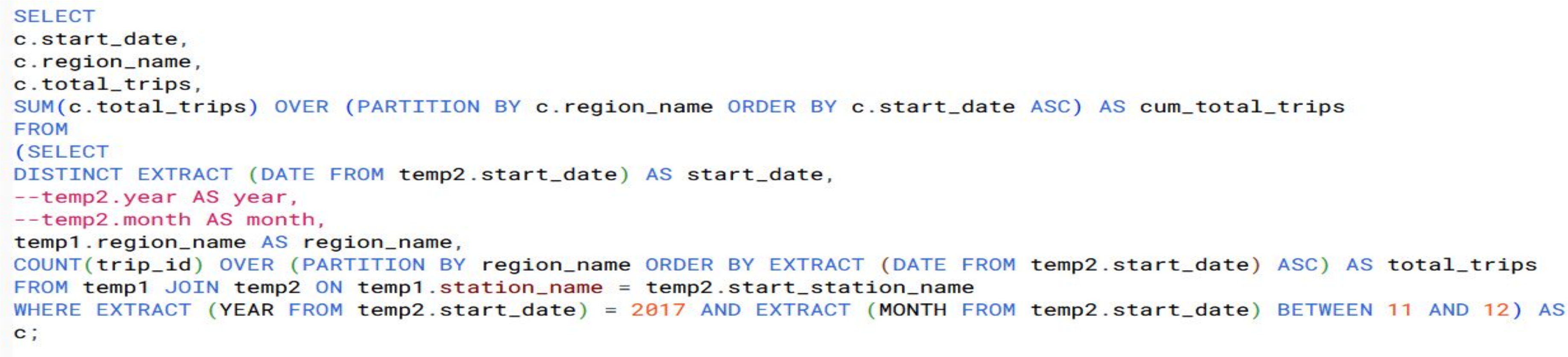
Expected Output:

- Start Date (in date format)
- Region Name
- Total Trips (in cumulative)

Query

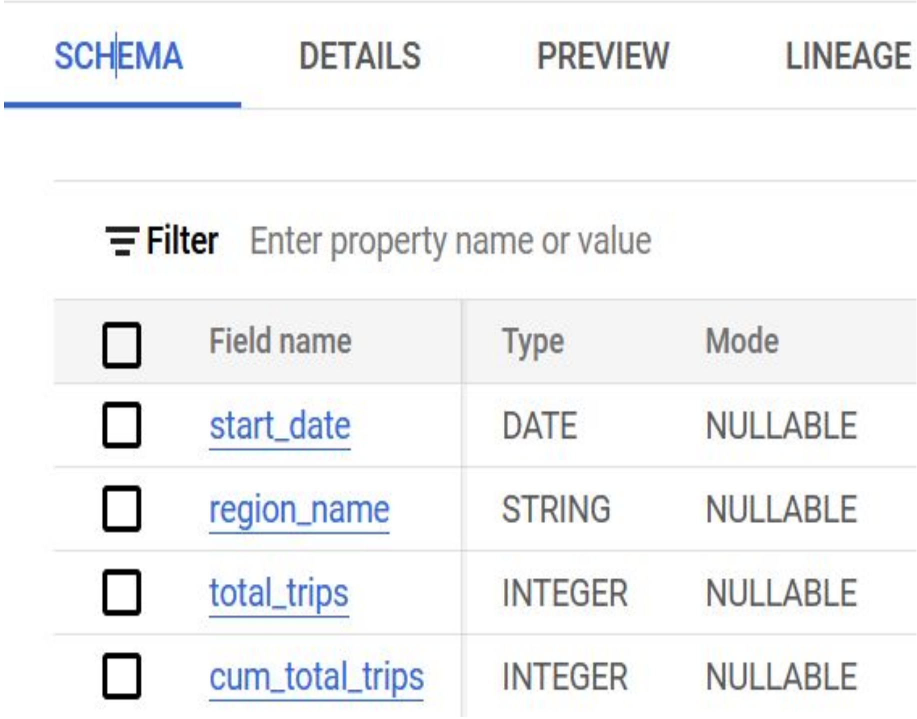


```
-- Question 5
WITH temp1 AS(
    SELECT
    a.name AS region_name,
    b.station_id AS station_id,
    b.name AS station_name
    FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_regions` AS a
    JOIN `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_info` AS b on a.region_id = b.region_ic
),
temp2 AS(
    SELECT
    start_date,
    EXTRACT (YEAR from start_date) AS year,
    EXTRACT (MONTH from start_date) AS month,
    trip_id,
    start_station_name
    FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
```

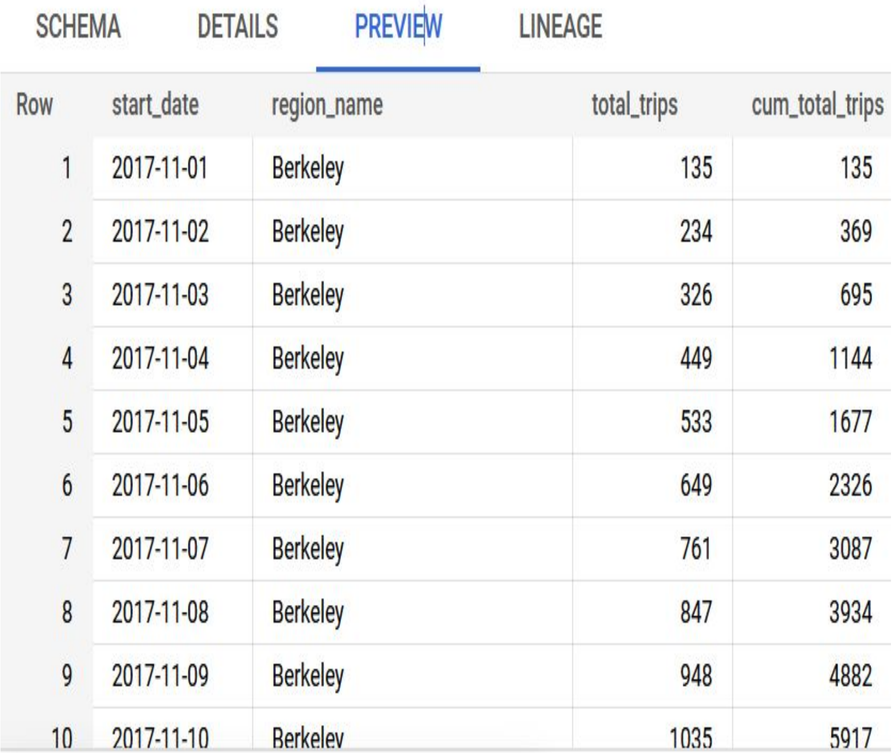


```
SELECT
c.start_date,
c.region_name,
c.total_trips,
SUM(c.total_trips) OVER (PARTITION BY c.region_name ORDER BY c.start_date ASC) AS cum_total_trips
FROM
(SELECT
DISTINCT EXTRACT (DATE FROM temp2.start_date) AS start_date,
--temp2.year AS year,
--temp2.month AS month,
temp1.region_name AS region_name,
COUNT(trip_id) OVER (PARTITION BY region_name ORDER BY EXTRACT (DATE FROM temp2.start_date) ASC) AS total_trips
FROM temp1 JOIN temp2 ON temp1.station_name = temp2.start_station_name
WHERE EXTRACT (YEAR FROM temp2.start_date) = 2017 AND EXTRACT (MONTH FROM temp2.start_date) BETWEEN 11 AND 12) AS
c;
```

Table Schema & Preview



□	Field name	Type	Mode
□	start_date	DATE	NULLABLE
□	region_name	STRING	NULLABLE
□	total_trips	INTEGER	NULLABLE
□	cum_total_trips	INTEGER	NULLABLE



Row	start_date	region_name	total_trips	cum_total_trips
1	2017-11-01	Berkeley	135	135
2	2017-11-02	Berkeley	234	369
3	2017-11-03	Berkeley	326	695
4	2017-11-04	Berkeley	449	1144
5	2017-11-05	Berkeley	533	1677
6	2017-11-06	Berkeley	649	2326
7	2017-11-07	Berkeley	761	3087
8	2017-11-08	Berkeley	847	3934
9	2017-11-09	Berkeley	948	4882
10	2017-11-10	Berkeley	1035	5917

Advanced

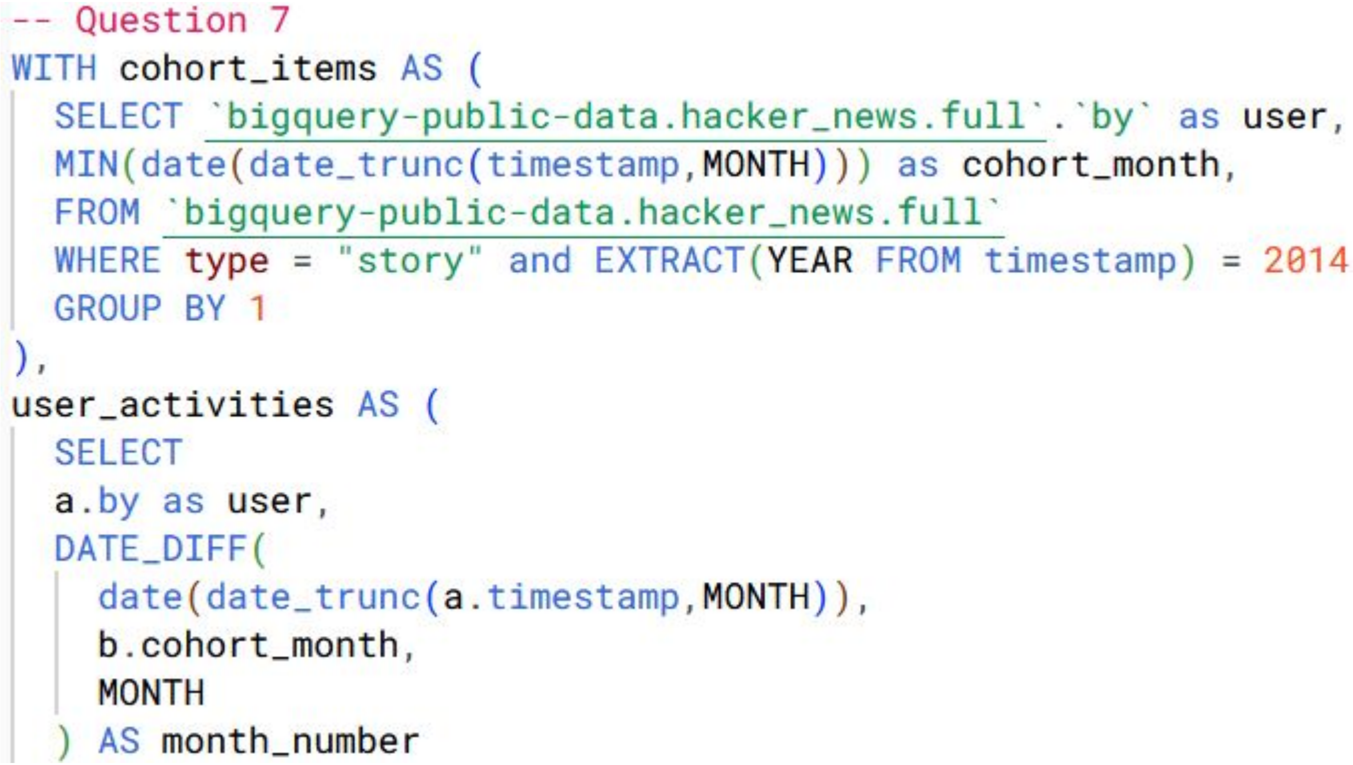
Given another dataset here: Hacker News please use table “stories” to create monthly retention cohorts (the groups, or cohorts, can be defined based upon the date that a user/ author started a story) and then how many of them (%) coming back for the following months in 2014. After analysing the retention cohort, is there any interesting insight that we can get? (Skillset: Basic SQL, Formatting and Cleaning in SQL, Window Function & SQL CTE)

Notes: initial start date can be defined using first story start date from each author in table stories

Expected output:

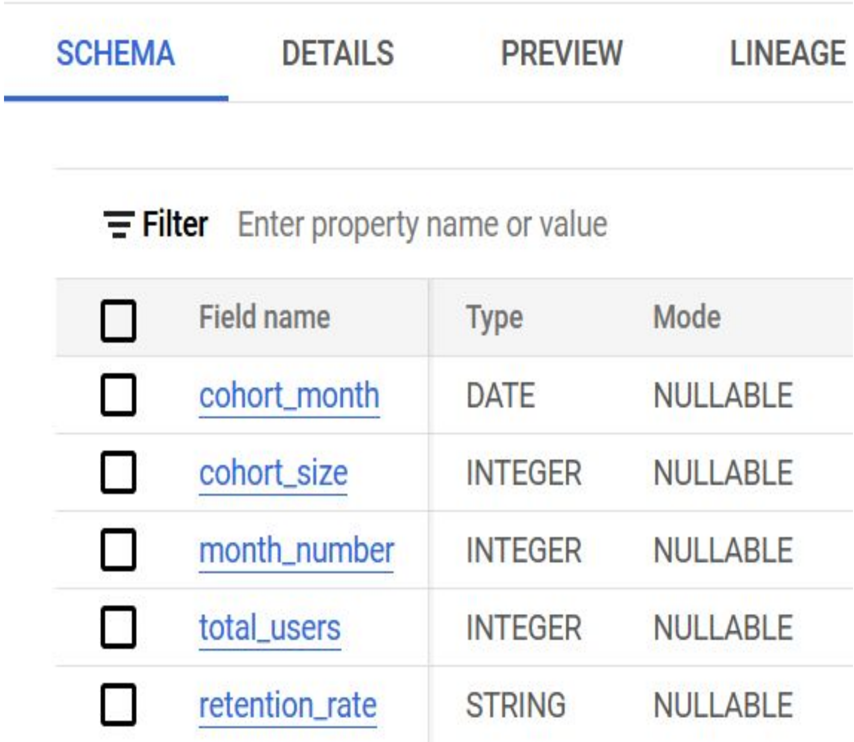
- First Post Month
- Active Post Month
- Number of Users

Query

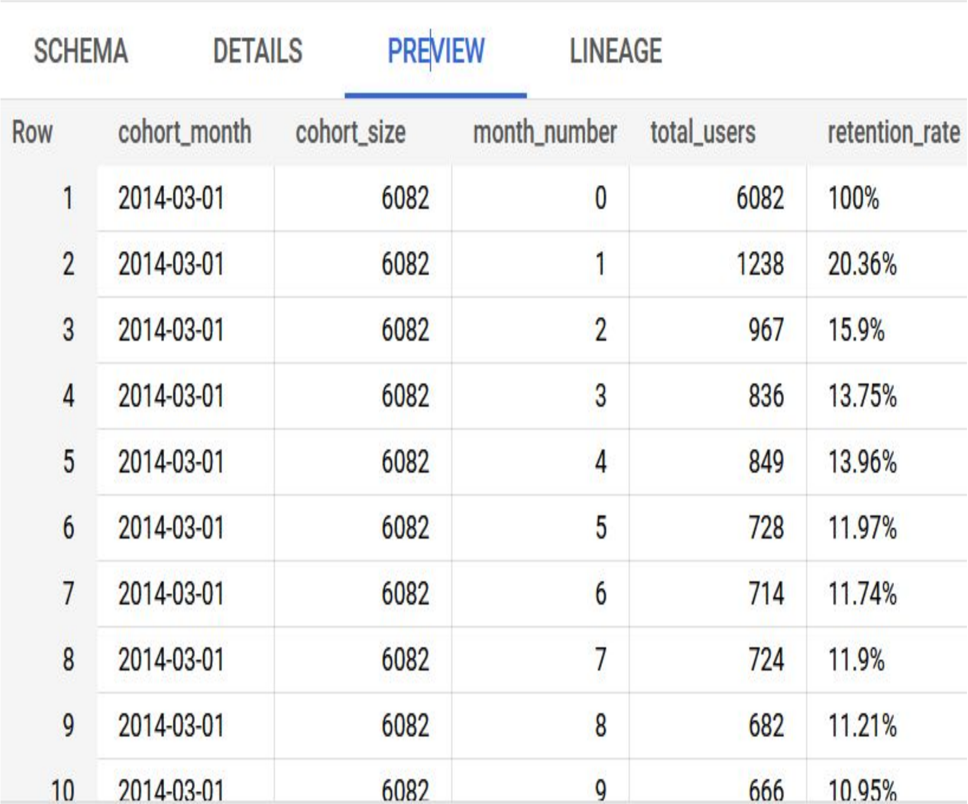


```
-- Question 7
WITH cohort_items AS (
  SELECT `bigquery-public-data.hacker_news.full`.`by` as user,
  MIN(date(date_trunc(timestamp,MONTH))) as cohort_month,
  FROM `bigquery-public-data.hacker_news.full`
  WHERE type = "story" and EXTRACT(YEAR FROM timestamp) = 2014
  GROUP BY 1
),
user_activities AS (
  SELECT
  a.by as user,
  DATE_DIFF(
    date(date_trunc(a.timestamp,MONTH)),
    b.cohort_month,
    MONTH
  ) AS month_number
```

Table Schema & Preview

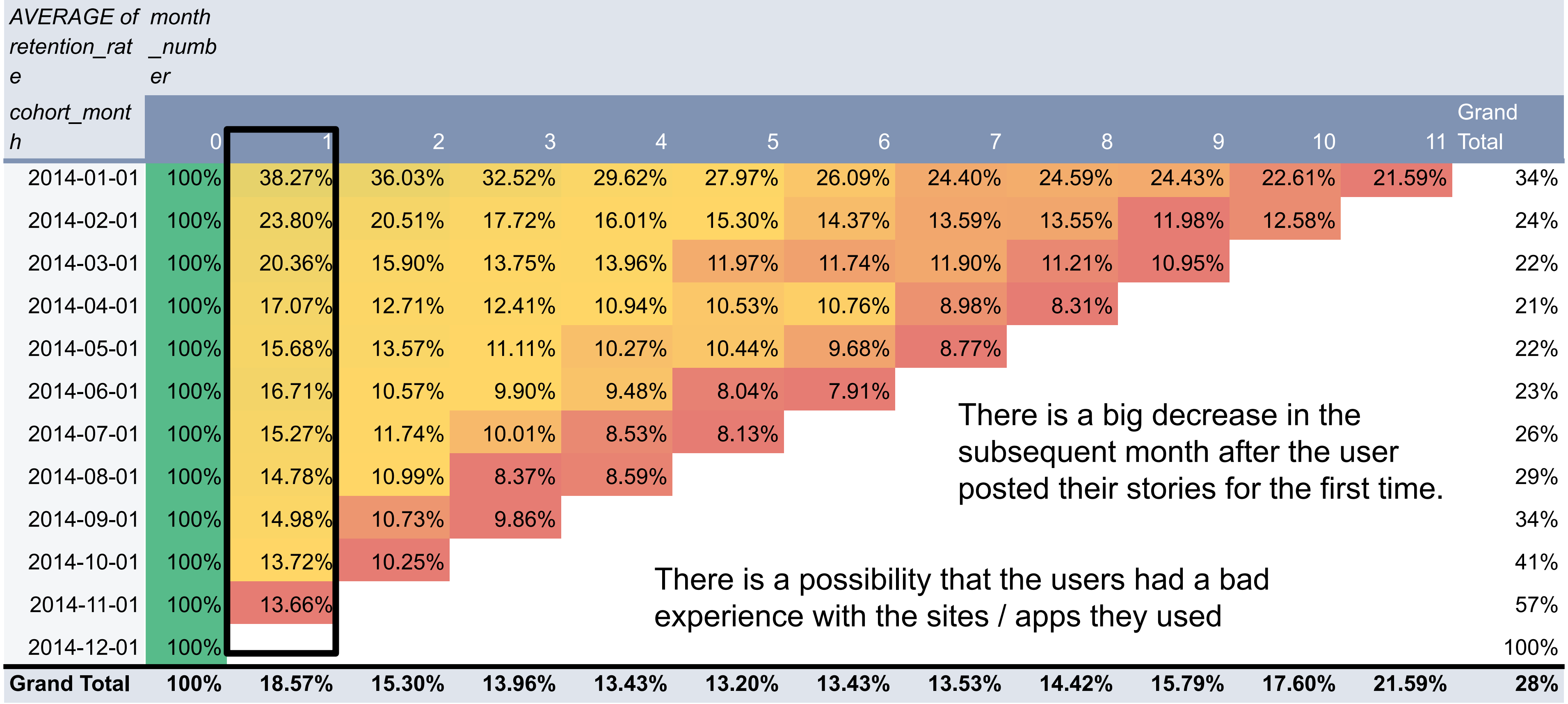


□	Field name	Type	Mode
□	cohort_month	DATE	NULLABLE
□	cohort_size	INTEGER	NULLABLE
□	month_number	INTEGER	NULLABLE
□	total_users	INTEGER	NULLABLE
□	retention_rate	STRING	NULLABLE



Row	cohort_month	cohort_size	month_number	total_users	retention_rate
1	2014-03-01	6082	0	6082	100%
2	2014-03-01	6082	1	1238	20.36%
3	2014-03-01	6082	2	967	15.9%
4	2014-03-01	6082	3	836	13.75%
5	2014-03-01	6082	4	849	13.96%
6	2014-03-01	6082	5	728	11.97%
7	2014-03-01	6082	6	714	11.74%
8	2014-03-01	6082	7	724	11.9%
9	2014-03-01	6082	8	682	11.21%
10	2014-03-01	6082	9	666	10.95%

Cohort Analysis



AVERAGE of retention_rate	*month_number*												
cohort_month	0	1	2	3	4	5	6	7	8	9	10	11	Grand Total
2014-01-01	100%	38.27%	36.03%	32.52%	29.62%	27.97%	26.09%	24.40%	24.59%	24.43%	22.61%	21.59%	34%
2014-02-01	100%	23.80%	20.51%	17.72%	16.01%	15.30%	14.37%	13.59%	13.55%	11.98%	12.58%		24%
2014-03-01	100%	20.36%	15.90%	13.75%	13.96%	11.97%	11.74%	11.90%	11.21%	10.95%			22%
2014-04-01	100%	17.07%	12.71%	12.41%	10.94%	10.53%	10.76%	8.98%	8.31%				21%
2014-05-01	100%	15.68%	13.57%	11.11%	10.27%	10.44%	9.68%	8.77%					22%
2014-06-01	100%	16.71%	10.57%	9.90%	9.48%	8.04%	7.91%						23%
2014-07-01	100%	15.27%	11.74%	10.01%	8.53%	8.13%							26%
2014-08-01	100%	14.78%	10.99%	8.37%	8.59%								29%
2014-09-01	100%	14.98%	10.73%	9.86%									34%
2014-10-01	100%	13.72%	10.25%										41%
2014-11-01	100%	13.66%											57%
2014-12-01	100%												100%
Grand Total	**100%**	**18.57%**	**15.30%**	**13.96%**	**13.43%**	**13.20%**	**13.43%**	**13.53%**	**14.42%**	**15.79%**	**17.60%**	**21.59%**	**28%**

There is a big decrease in the subsequent month after the user posted their stories for the first time.

There is a possibility that the users had a bad experience with the sites / apps they used