**<https://platform.stratascratch.com/coding/>**

**1. Distances Traveled**

Find the top 10 users that have traveled the greatest distance. Output their id, name and a total distance traveled.

select ll.id, lu.name, SUM(ll.distance) as total\_distance

from lyft\_rides\_log ll

join lyft\_users lu

on ll.user\_id = lu.id

GROUP BY ll.user\_id

ORDER BY total\_distance desc

Limit 10;

**2. Cities With The Most Expensive Homes**

Write a query that identifies cities with higher than average home prices when compared to the national average. Output the city names.

select city, avg(mkt\_price)

from zillow\_transactions a

GROUP BY city

HAVING AVG(a.mkt\_price) > (select AVG(mkt\_price)

from zillow\_transactions);

4. **Host Popularity Rental Prices**

You’re given a table of rental property searches by users. The table consists of search results and outputs host information for searchers. Find the minimum, average, maximum rental prices for each host’s popularity rating. The host’s popularity rating is defined as below: 0 reviews: New 1 to 5 reviews: Rising 6 to 15 reviews: Trending Up 16 to 40 reviews: Popular more than 40 reviews: Hot Tip: The `id` column in the table refers to the search ID. You'll need to create your own host\_id by concating price, room\_type, host\_since, zipcode, and number\_of\_reviews. Output host popularity rating and their minimum, average and maximum rental prices.

SELECT number\_of\_reviews, AVG(price) AS avg\_price, min(price) as min\_price, max(price) as max\_price,

CASE

when number\_of\_reviews = 0 THEN 'New'

when number\_of\_reviews BETWEEN 1 AND 5 THEN 'Rising'

when number\_of\_reviews BETWEEN 6 AND 15 THEN 'Trending Up'

when number\_of\_reviews BETWEEN 16 AND 40 THEN 'Popular'

when number\_of\_reviews > 40 THEN 'Hot'

END AS host\_rating

FROM airbnb\_host\_searches

GROUP BY host\_rating;

5. **Finding User Purchases**

Write a query that'll identify returning active users. A returning active user is a user that has made a second purchase within 7 days of any other of their purchases. Output a list of user\_ids of these returning active users.

with temp AS

(

select a.user\_id,

DATEDIFF(t.created\_at, a.created\_at) as date\_diff

from amazon\_transactions a

join amazon\_transactions t

on a.user\_id = t.user\_id

where a.id <> t.id

) select distinct user\_id from temp

where date\_diff between -7 AND 7

order by user\_id ;

**6. Monthly Percentage Difference**

Given a table of purchases by date, calculate the month-over-month percentage change in revenue. The output should include the year-month date (YYYY-MM) and percentage change, rounded to the 2nd decimal point, and sorted from the beginning of the year to the end of the year. The percentage change column will be populated from the 2nd month forward and can be calculated as ((this month's revenue - last month's revenue) / last month's revenue)\*100.

with temp AS

(

select DATE\_FORMAT(created\_at, "%Y-%m") AS year\_month1, SUM(value) as revenue

from sf\_transactions

Group by year\_month1

order by year\_month1

)

select year\_month1,

ROUND((((revenue - LAG(revenue) OVER(order by year\_month1))/LAG(revenue)OVER(order by year\_month1))\*100), 2)

as revenue\_diff\_pct

from temp;

7. **Class Performance**

You are given a table containing assignment scores of students in a class. Write a query that identifies the largest difference in total score of all assignments. Output just the difference in total score between the two students.

with cte as

(

select student, (assignment1 + assignment2 + assignment3) as total\_score

from box\_scores

order by total\_score

) select MAX(total\_score)-MIN(total\_score) as difference

from cte;

**8. Revenue Over Time**

Find the 3-month rolling average of total revenue from purchases given a table with users, their purchase amount, and date purchased. Do not include returns which are represented by negative purchase values. Output the year-month (YYYY-MM) and 3-month rolling average of revenue, sorted from earliest month to latest month. A 3-month rolling average is defined by calculating the average total revenue from all user purchases for the current month and previous two months. The first two months will not be a true 3-month rolling average since we are not given data from last year. Assume each month has at least one purchase.

with cte as

(

select DATE\_FORMAT(created\_at, '%Y-%m') as yearmonth, SUM(purchase\_amt) as revenue

from amazon\_purchases

group by yearmonth

) select yearmonth,

#LAG(revenue) OVER (order by yearmonth) as 3rd\_month,

#AVG((LAG(revenue, 2) OVER (order by yearmonth)) + LAG(revenue) OVER (order by yearmonth) + revenue) as total\_3months\_revenue

avg(revenue) OVER (order by yearmonth

rows between 2 preceding and current row) as rolling\_avg

from cte;

**9. Salaries Differences**

Write a query that calculates the difference between the highest salaries found in the marketing and engineering departments. Output just the absolute difference in salaries.

with cte as

(

select (select max(e.salary) from db\_employee e join db\_dept d

on e.department\_id = d.id

where d.department = 'marketing') as salary1,

(select distinct max(e.salary) from db\_employee e join db\_dept d

on e.department\_id = d.id

where d.department = 'engineering') as salary2

from db\_employee

) select max(salary1) - max(salary2 ) as difference from cte;