STRATASCRATCH MEDIUM QUESTIONS

**Find the top 10 ranked songs in 2010**

select distinct year\_rank,

group\_name,

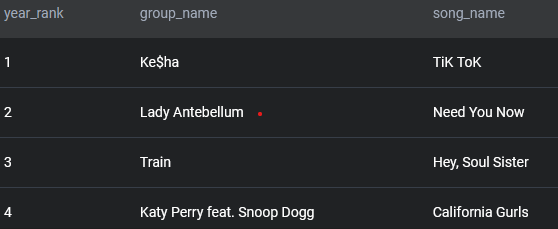
song\_name

from billboard\_top\_100\_year\_end

where year = 2010

group by 1

limit 10;



Find songs that have ranked in the top position. Output the track name and the number of times it ranked at the top. Sort your records by the number of times the song was in the top position in descending order.

select trackname,

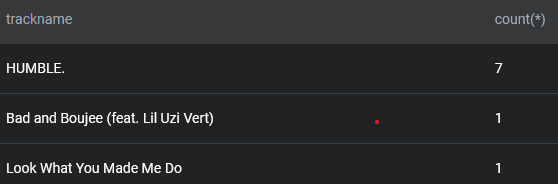
count(\*)

from spotify\_worldwide\_daily\_song\_ranking AS top\_ranking

where position = 1

group by 1

order by 2 DESC;



Output share of US users that are active. Active users are the ones with an "open" status in the table.

WITH CTE1 AS (

SELECT \*

FROM fb\_active\_users

WHERE country = 'USA' AND status = 'open'

),

CTE2 AS (

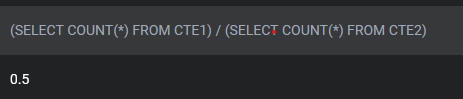
SELECT \*

FROM fb\_active\_users

WHERE country = 'USA'

)

SELECT (SELECT COUNT(\*) FROM CTE1) / (SELECT COUNT(\*) FROM CTE2);



Calculate each user's average session time. A session is defined as the time difference between a page\_load and page\_exit. For simplicity, assume a user has only 1 session per day and if there are multiple of the same events on that day, consider only the latest page\_load and earliest page\_exit, with an obvious restriction that load time event should happen before exit time event . Output the user\_id and their average session time.

SELECT user\_id,

AVG(TIMESTAMPDIFF(SECOND, p\_load\_time, p\_exit\_time)) AS avg1

FROM

(

SELECT user\_id,

DATE(timestamp),

MAX(CASE WHEN action = 'page\_load' THEN timestamp ELSE NULL END) AS p\_load\_time,

MIN(CASE WHEN action = 'page\_exit' THEN timestamp ELSE NULL END) AS p\_exit\_time

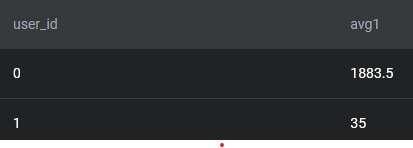
FROM facebook\_web\_log

GROUP BY user\_id, DATE(timestamp)

) AS t

GROUP BY user\_id

HAVING avg1 IS NOT NULL;



Identify projects that are at risk for going overbudget. A project is considered to be overbudget if the cost of all employees assigned to the project is greater than the budget of the project.

You'll need to prorate the cost of the employees to the duration of the project. For example, if the budget for a project that takes half a year to complete is $10K, then the total half-year salary of all employees assigned to the project should not exceed $10K. Salary is defined on a yearly basis, so be careful how to calculate salaries for the projects that last less or more than one year.

Output a list of projects that are overbudget with their project name, project budget, and prorated total employee expense (rounded to the next dollar amount).

HINT: to make it simpler, consider that all years have 365 days. You don't need to think about the leap years.

WITH cte1 AS (

SELECT p.title, p.budget, e.salary,

(e.salary / 365) AS 1\_day\_of\_sal,

DATEDIFF(p.end\_date, p.start\_date) AS total\_days

FROM linkedin\_projects AS p

JOIN linkedin\_emp\_projects AS ep ON p.id = ep.project\_id

JOIN linkedin\_employees AS e ON e.id = ep.emp\_id

)

SELECT title, budget, prorated\_employee\_expense

FROM (

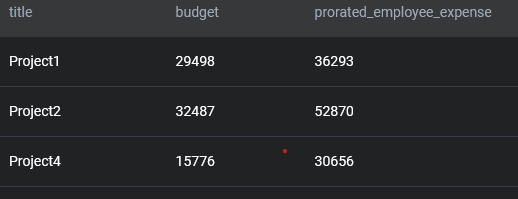
SELECT title, budget, CEIL(SUM(1\_day\_of\_sal \* total\_days)) AS prorated\_employee\_expense

FROM cte1

GROUP BY title, budget

) AS t

WHERE budget < prorated\_employee\_expense;



WITH cte1 AS (

SELECT p.title, p.budget, e.salary,

(e.salary / 365) AS 1\_day\_of\_sal,

DATEDIFF(p.end\_date, p.start\_date) AS total\_days

FROM linkedin\_projects AS p

JOIN linkedin\_emp\_projects AS ep ON p.id = ep.project\_id

JOIN linkedin\_employees AS e ON e.id = ep.emp\_id

)

SELECT title, budget, prorated\_employee\_expense

FROM (

SELECT title, budget, CEIL(SUM(1\_day\_of\_sal \* total\_days)) AS prorated\_employee\_expense

FROM cte1

GROUP BY title, budget

) AS t

WHERE budget > prorated\_employee\_expense;

