`search_results` table

column	**type**
query	varchar
result_id	integer
position	integer
rating	integer

`search_events` table

column	**type**
search_id	integer
query	varchar
has_clicked	boolean

You're given a table that represents search results from searches on Facebook. The `query` column is the search term, `position` column represents each position the search result came in, and the `rating` column represents the human rating from 1 to 5 where 5 is high relevance and 1 is low relevance.

Each row in the `search_events` table represents a single search with the `has_clicked` column representing if a user clicked on a result or not.

We have a hypothesis that the CTR is dependent on the search result rating.
Write a query to return data to support or disprove this hypothesis

Solution

Analysis :
CTR → dependent on average rating ???
We need to find out

How to get average rating?

Search_results

Average(rating) group by query

What is click through rate CTR??

If CTR is high, then search results are high CTR is low then search results are low
Here search results in the sense search result ratings

If search result ratings are low and CTR is high then hypothesis is not proven

Increase in rating correlated to increase of CTR ,→ we can check this

```
WITH ratings AS (
    SELECT query
        , SUM(CASE WHEN
                rating <= 1 THEN 1 ELSE 0
            END) AS num_results_rating_one
        , SUM(CASE WHEN
                rating <= 2 THEN 1 ELSE 0
            END) AS num_results_rating_two
        , SUM(CASE WHEN
                rating <= 3 THEN 1 ELSE 0
            END) AS num_results_rating_three
        , COUNT(*) AS total_results
    FROM search_results
    GROUP BY 1
)

SELECT * FROM ratings
```

query	num_results_rating_one	num_results_rating_two	num_results_rating_three	total_results
you	1	1	4	6
year	4	6	6	9
within	0	4	4	7
war	2	5	6	10

```
WITH ratings AS (
    SELECT query
        , SUM(CASE WHEN
                rating <= 1 THEN 1 ELSE 0
            END) AS num_results_rating_one
        , SUM(CASE WHEN
                rating <= 2 THEN 1 ELSE 0
            END) AS num_results_rating_two
        , SUM(CASE WHEN
                rating <= 3 THEN 1 ELSE 0
            END) AS num_results_rating_three
        , COUNT(*) AS total_results
    FROM search_results
    GROUP BY 1
)

SELECT
    CASE
        WHEN total_results - num_results_rating_one = 0
            THEN 'results_one'
        WHEN total_results - num_results_rating_two = 0
            THEN 'results_two'
        WHEN total_results - num_results_rating_three = 0
            THEN 'results_three'
    END AS ratings_bucket
    , SUM(has_clicked)/COUNT(*) AS ctr
FROM search_events AS se
LEFT JOIN ratings AS r
    ON se.query = r.query
GROUP BY 1
```

ratings_bucket	ctr
null	0.4948
results_three	0.6667

a query with one result with a 1 rating and another with a 5 rating would equate to an average of 3. **Whereas in this system, a query with all of their results under 1 or all of their results under 2, we can determine to be actually bad ratings.**