Operation Analytics and Investigating Metric Spike

Project Description

Operation Analytics is the analysis done for the complete end to end operations of a company and investigating metric spike is also an important part of operation analytics. Deriving certain insights and answer the questions from different data sets, tables for a company to predict the overall growth of a company's fortune.

Approach

Taking the dataset into consideration and solving each query using SQL to get an insight for marketing and investor metrics.

Tech-Stack Used

Microsoft SQL Server Management Studio

- Version: 18.11.1
- I have already installed SQL server in my laptop and had good hands on experience with the SQL server

Online editors

- Hacker rank, lettcode, strata scratch, mode.com
- To get a good grip on different platforms, I have used these online editors as well.

Insights

To put it down in one sentence Gained knowledge on how to analyze each given problem and find relevant solutions to it using SQL. Also learned what questions to be asked for better solution.

Result

I have achieved strong knowledge on SQL while making the project and it helped me to improve my basic knowledge in SQL to advance level and also learned how to deal a problem to find better solution.

Queries

Case Study 1 (Job Data)

A.
```
select ds,count(job_id) as jobs, sum(time_spent)*0.000277778 as in_hours from
job_data  group by ds
```

B.
```
WITH niharika AS (SELECT ds, COUNT(job_id) AS num_jobs, SUM(time_spent) AS
total_time FROM job_data  GROUP BY ds )
SELECT ds, ROUND(1.0* SUM(num_jobs) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING
AND CURRENT ROW) / SUM(total_time) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND
CURRENT ROW),2) AS throughput_7d FROM niharika
```
C.
```
select language, (Count(language)* 100 / (Select Count(*) From job_data)) as
percnt from job_data group by language
```
D.
```
SELECT job_id,language, COUNT(*) as count_of FROM job_data GROUP BY
job_id,language HAVING COUNT(*) > 1
```

Case Study 2 (Investigating metric spike)

A.
```
SELECT DATE_TRUNC('week', e.occurred_at) AS "week", COUNT(DISTINCT e.user_id) AS
weekly_active_users FROM events e  WHERE e.event_type = 'engagement' AND
e.event_name = 'login' GROUP BY week
```
B.
```
SELECT DATE_TRUNC('month', created_at) AS day,COUNT(*) AS all_users,
COUNT(CASE WHEN activated_at IS NOT NULL THEN user_id ELSE NULL END) AS
activated_users FROM users GROUP BY day
```

C.
```
SELECT  DATE_TRUNC('week', e.occurred_at) AS "week", COUNT(e.user_id) as
"count_of_users" FROM users u , events e WHERE e.event_type='signup_flow' AND
u.user_id=e.user_id AND u.activated_at IS NOT NULL  GROUP BY week
```

D.
```
SELECT  device , DATE_TRUNC('week', occurred_at) AS "week", COUNT(DISTINCT
user_id) as "users" FROM events WHERE event_type='engagement'  GROUP BY device,
week HAVING COUNT(device) > 1
```

E.
```
SELECT action, DATE_TRUNC('week', occurred_at) AS "week", COUNT(DISTINCT user_id)
as "users" FROM email_events  GROUP BY action, week HAVING COUNT(action) > 1
```