OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

ABHISHEK JAGTAP

PROJECT DESCRIPTION

The project is focused on Operation Analytics and Investigating Metric Spike using advanced SQL. The project aims to analyze the complete end-to-end operations of company, with the help of which the company can then find the areas that need improvement. The data collected is used to predict the overall growth or decline of a company's fortune, resulting in better automation, better understanding between cross-functional teams, and more effective workflows

APPROACH

To execute the project, I firstly collected and analyzed the relevant data sets and tables provided by the company. I then applied advanced SQL queries to derive insights and answer the questions asked by different departments.

TECH-STACK USED

- My SQL workbench
- Mode analytics (Mode is a collaborative data platform that combines SQL, R, Python, and visual analytics in one place. Connect, analyze, and share, faster)

INSIGHTS

In this project, I gained insights into the importance of operation analytics in predicting a company's growth or decline. I also learned how to use advanced SQL queries to extract insights from large data sets and how to use data visualization techniques to better understand the data.

RESULT

As a result of this project, I was able to provide a detailed report for the two operations mentioned in the project description. I was able to calculate the number of jobs reviewed per hour per day for November 2020, the 7-day rolling average of throughput, the percentage share of each language in the last 30 days, and how to display duplicate rows from the table. Additionally, I was able to calculate the weekly user engagement, user growth for a product, weekly retention of users-sign up cohort, and weekly engagement of users.

CASE STUDY 1

Job Data Analysis

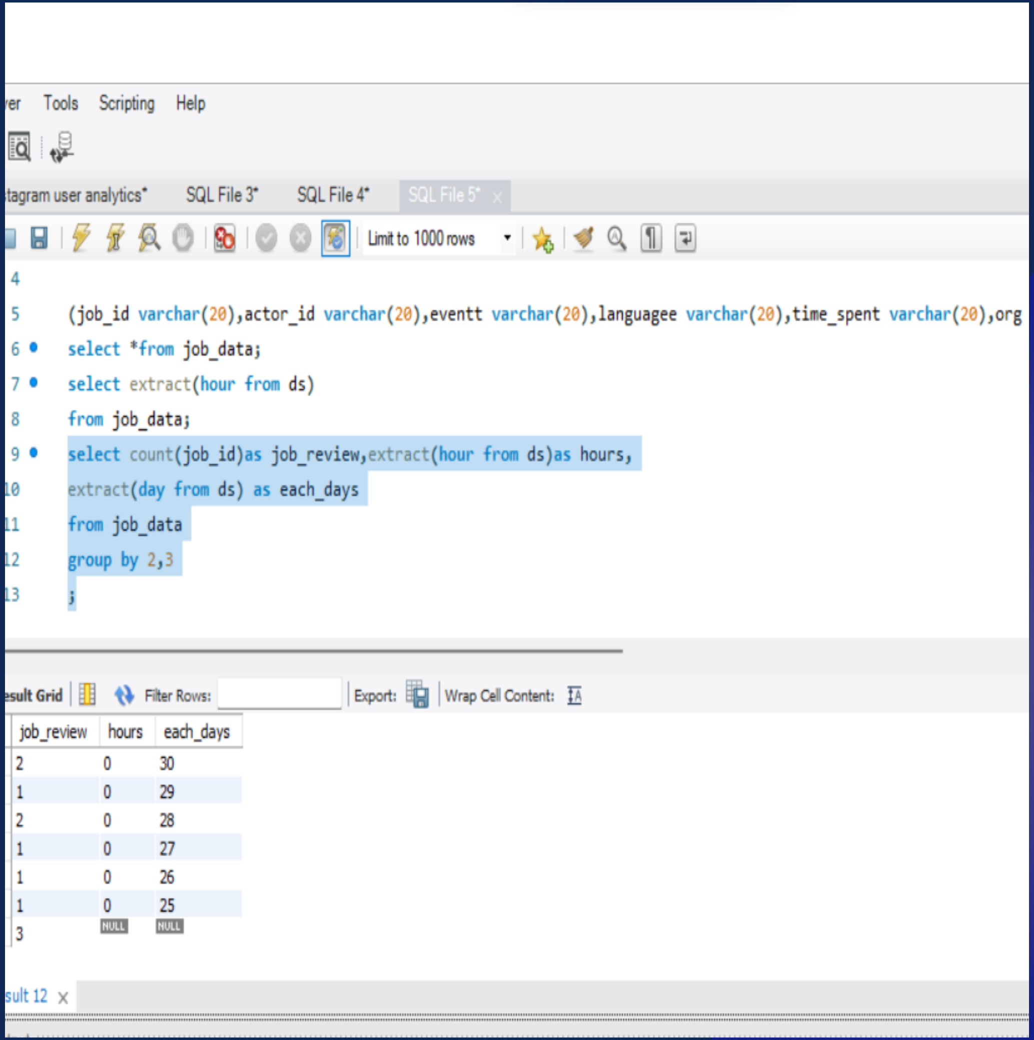
TASK 1

A. Jobs Reviewed Over Time:

- Objective: Calculate the number of jobs reviewed per hour for each day in November 2020.
- Your Task: Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.

Query

```
select
count(job_id)as job_review,
extract(hour from ds)as hours
,extract(day from ds) as each_days
from job_data
group by 2,3;
```



```
(job_id varchar(20),actor_id varchar(20),eventt varchar(20),languagee varchar(20),time_spent varchar(20),org
select *from job_data;
select extract(hour from ds)
from job_data;
select count(job_id)as job_review,extract(hour from ds)as hours,
extract(day from ds) as each_days
from job_data
group by 2,3
;
```

```
job_review  hours  each_days
2           0      30
1           0      29
2           0      28
1           0      27
1           0      26
1           0      25
3           NULL   NULL
```

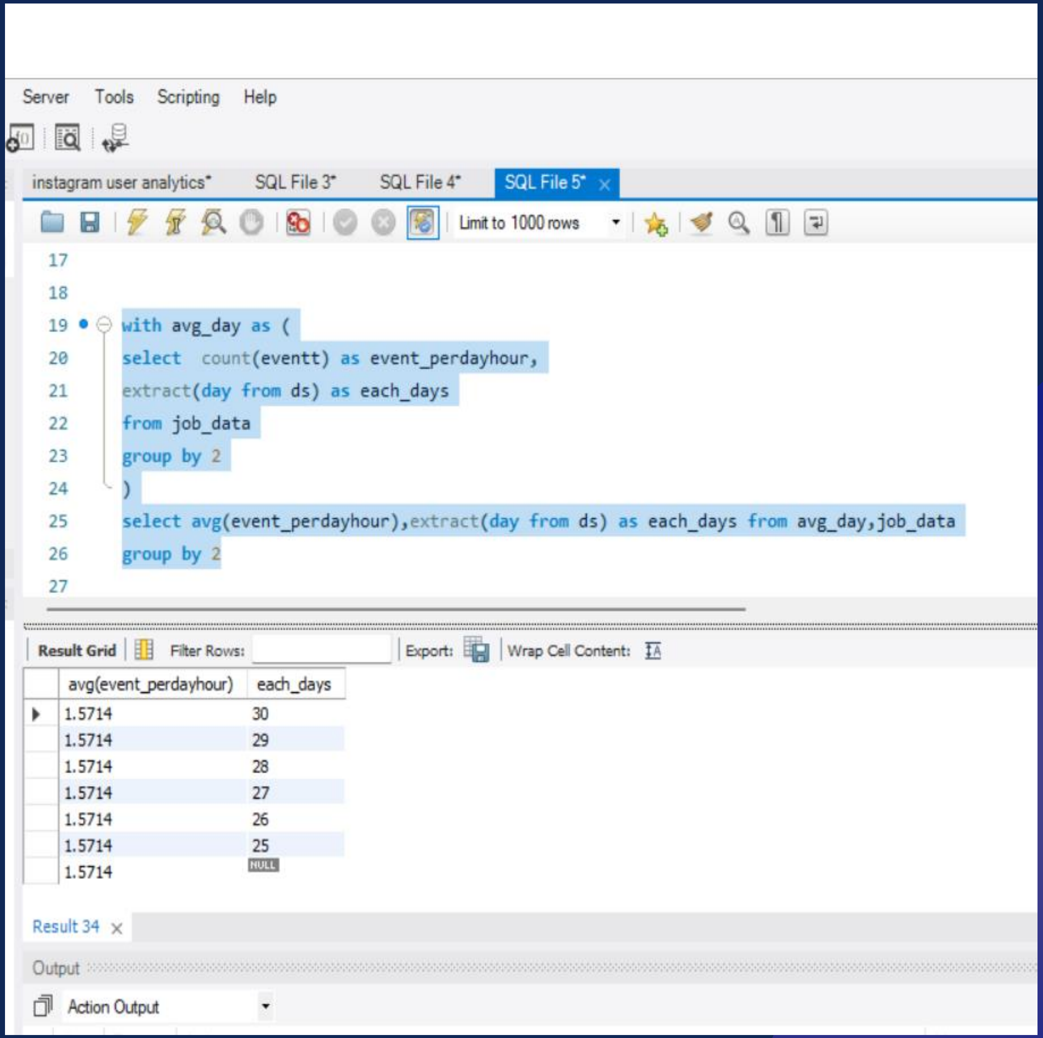
TASK 2

B. Throughput Analysis:

- Objective: Calculate the 7-day rolling average of throughput (number of events per second).
- Your Task: Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

Query

```
with avg_day as
 ( select
count(eventt) as event_perdayhour,
extract(day from ds) as each_days
from job_data
group by 2 )
select avg(event_perdayhour),
extract(day from ds) as each_days
from avg_day,job_data
group by 2;
```



```
with avg_day as (
select  count(eventt) as event_perdayhour,
extract(day from ds) as each_days
from job_data
group by 2
)
select avg(event_perdayhour),extract(day from ds) as each_days from avg_day,job_data
group by 2
```

```
avg(event_perdayhour)   each_days
1.5714                  30
1.5714                  29
1.5714                  28
1.5714                  27
1.5714                  26
1.5714                  25
1.5714                  NULL
```

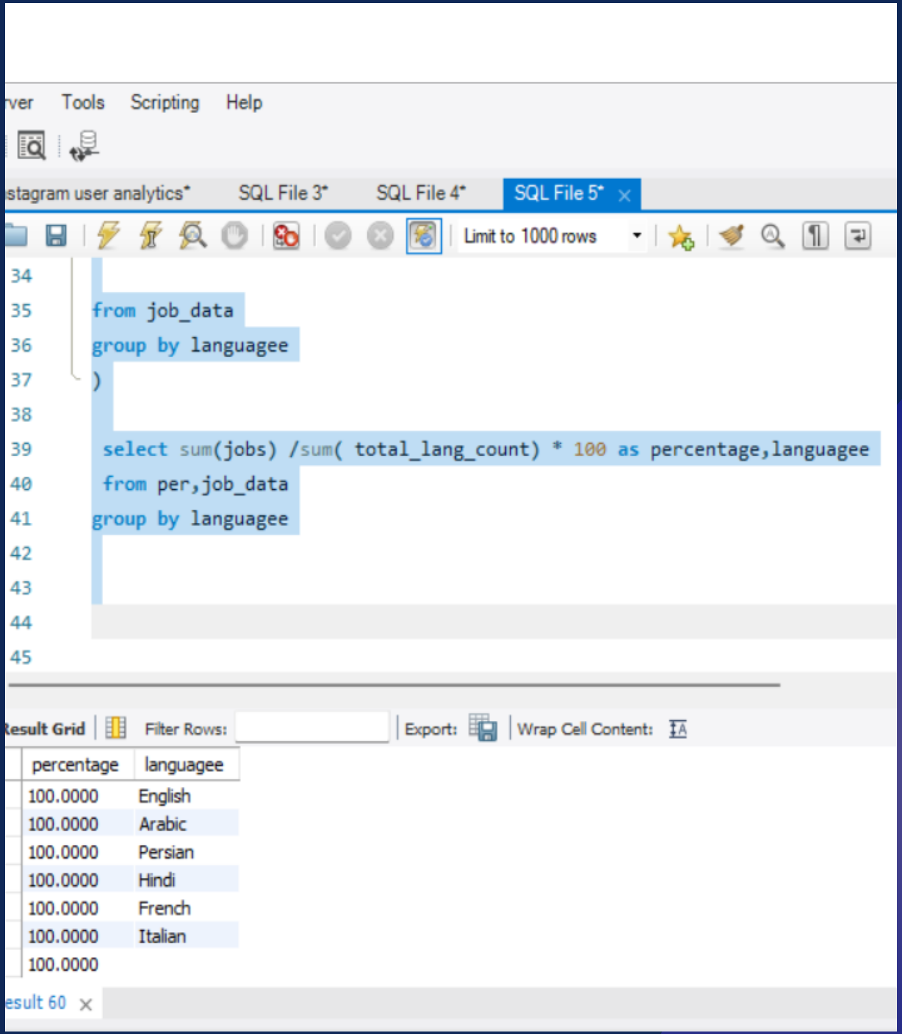
TASK3

C. Language Share Analysis:

- Objective: Calculate the percentage share of each language in the last 30 days.
- Your Task: Write an SQL query to calculate the percentage share of each language over the last 30 days.

Query

```
with per as
(select
count(languagee) as total_lang_count,
count( job_id)as jobs
from job_data
group by language )
 select
sum(jobs) /sum( total_lang_count) * 100 as percentage,language
 from per,job_data
group by languagee;
```



```
from job_data
group by languagee
)

 select sum(jobs) /sum( total_lang_count) * 100 as percentage,languagee
 from per,job_data
group by languagee
```

percentage	languagee
100.0000	English
100.0000	Arabic
100.0000	Persian
100.0000	Hindi
100.0000	French
100.0000	Italian
100.0000	

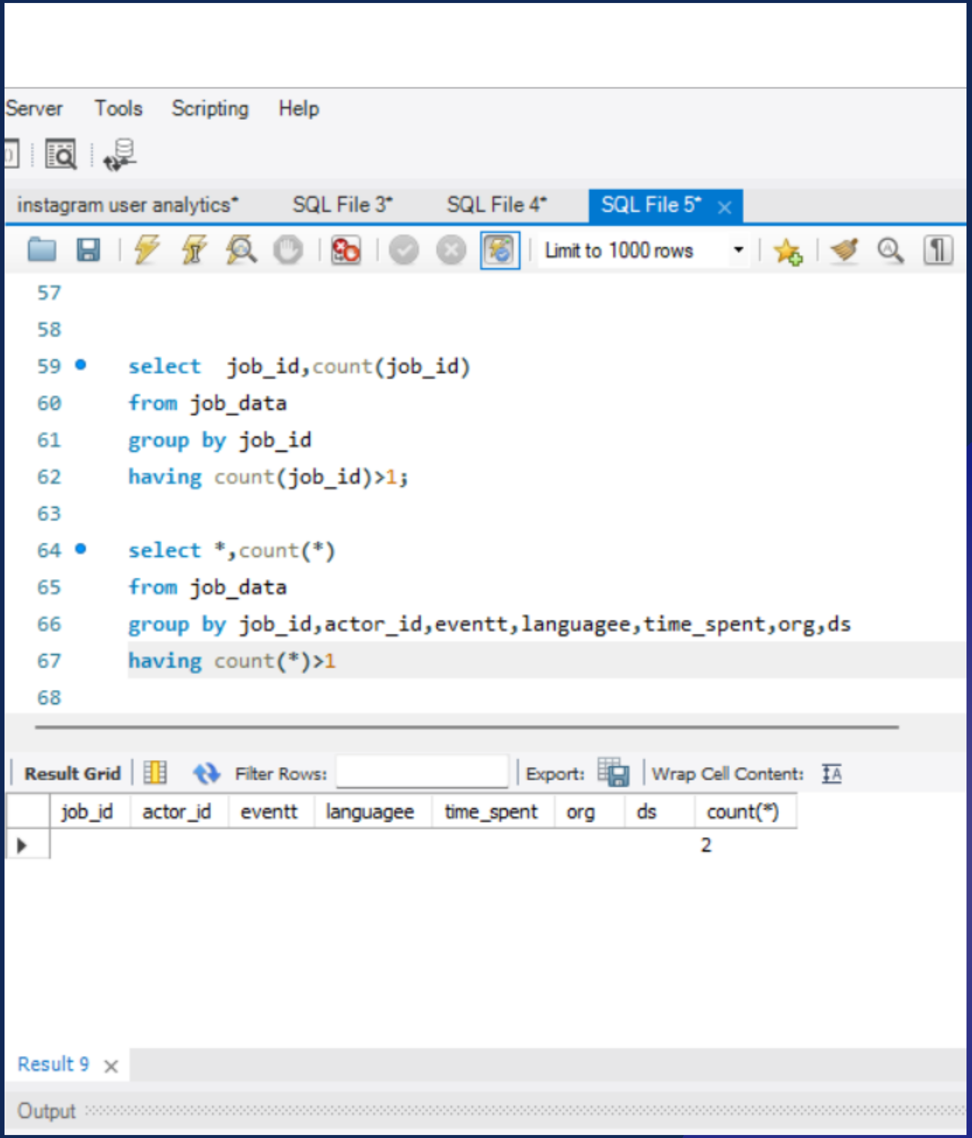
TASK4

D. Duplicate Rows Detection:

- Objective: Identify duplicate rows in the data.
- Your Task: Write an SQL query to display duplicate rows from the job_data table.

Query

```
select *,
count(*)
from job_data
 group by
job_id,actor_id,eventt,languagee,tim_spent,
org,ds
having
count(*)>1;
```



```
select  job_id,count(job_id)
from job_data
group by job_id
having count(job_id)>1;

select *,count(*)
from job_data
group by job_id,actor_id,eventt,languagee,time_spent,org,ds
having count(*)>1
```

```
job_id  actor_id  eventt  languagee  time_spent  org  ds  count(*)
                                                          2
```

CASE STUDY 2

Investigating Metric Spike

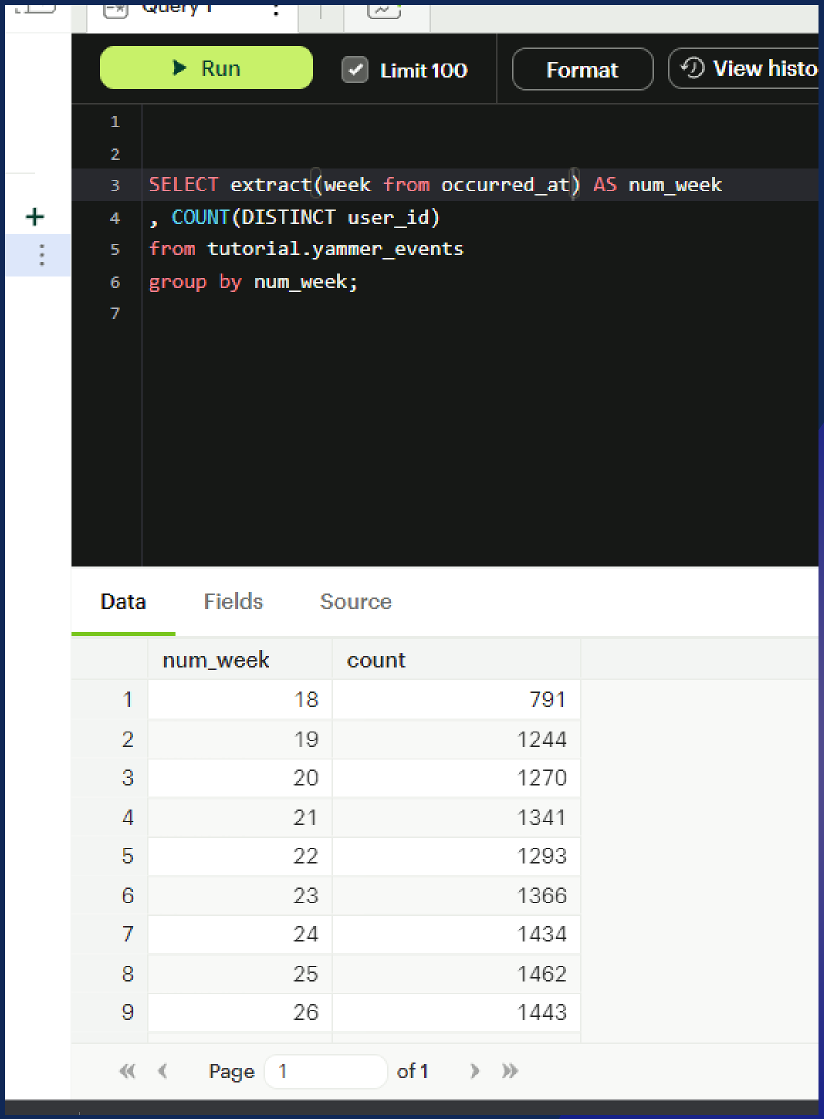
TASK 1

A. Weekly User Engagement:

- Objective: Measure the activeness of users on a weekly basis.
- Your Task: Write an SQL query to calculate the weekly user engagement.

Query

```
SELECT

 extract(week from occurred_at) AS num_week,
COUNT(DISTINCT user_id)

from tutorial.yammer_events

group by num_week;
```



```
SELECT extract(week from occurred_at) AS num_week
, COUNT(DISTINCT user_id)
from tutorial.yammer_events
group by num_week;
```

	num_week	count
1	18	791
2	19	1244
3	20	1270
4	21	1341
5	22	1293
6	23	1366
7	24	1434
8	25	1462
9	26	1443

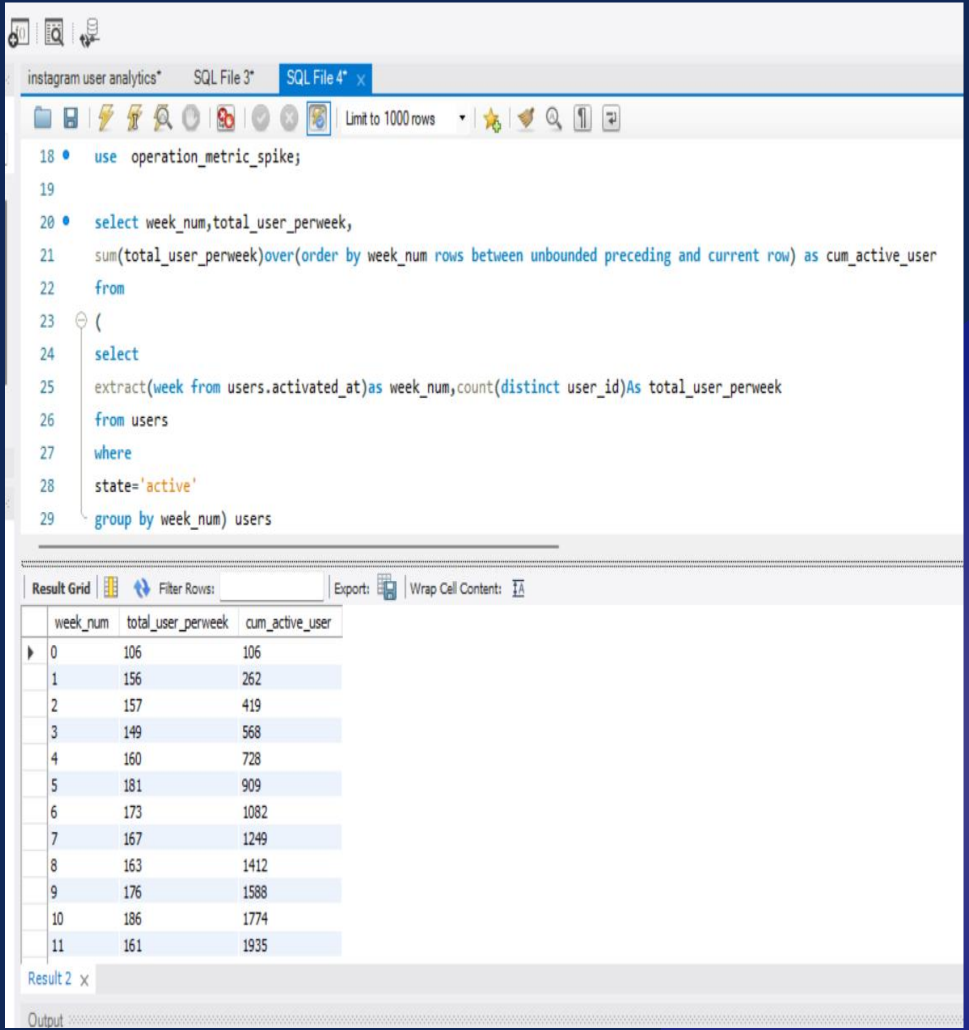
TASK 2

B. User Growth Analysis:

- Objective: Analyze the growth of users over time for a product.
- Your Task: Write an SQL query to calculate the user growth for the product.

Query

```
select
week_num,total_user_perweek,sum(total_user_perweek)

over(order by week_num rows between unbounded preceding and current row) as cum_active_user

From

(select

extract(week from users.activated_at)as week_num

,count(distinct user_id)As total_user_perweek

from users

where state='active’

group by week_num) users
```



```
use  operation_metric_spike;

select week_num,total_user_perweek,
sum(total_user_perweek)over(order by week_num rows between unbounded preceding and current row) as cum_active_user
from
(
select
extract(week from users.activated_at)as week_num,count(distinct user_id)As total_user_perweek
from users
where
state='active'
group by week_num) users
```

week_num	total_user_perweek	cum_active_user
0	106	106
1	156	262
2	157	419
3	149	568
4	160	728
5	181	909
6	173	1082
7	167	1249
8	163	1412
9	176	1588
10	186	1774
11	161	1935

TASK3

C. Weekly Retention Analysis:

- Objective: Analyze the retention of users on a weekly basis after signing up for a product.
- Your Task: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

Query

SELECT a.user_id, a.signup_week, b.engagement_week, b.engagement_week-a.signup_week as retention_week

FROM

((select distinct user_id,extract (week from occurred_at) as signup_week

from tutorial.yammer_events

WHERE event_type ='signup_flow' AND event_name = 'complete_signup' AND extract(week from occurred_at)=18)a

left join

(select distinct user_id, extract (week from occurred_at) as engagement_week

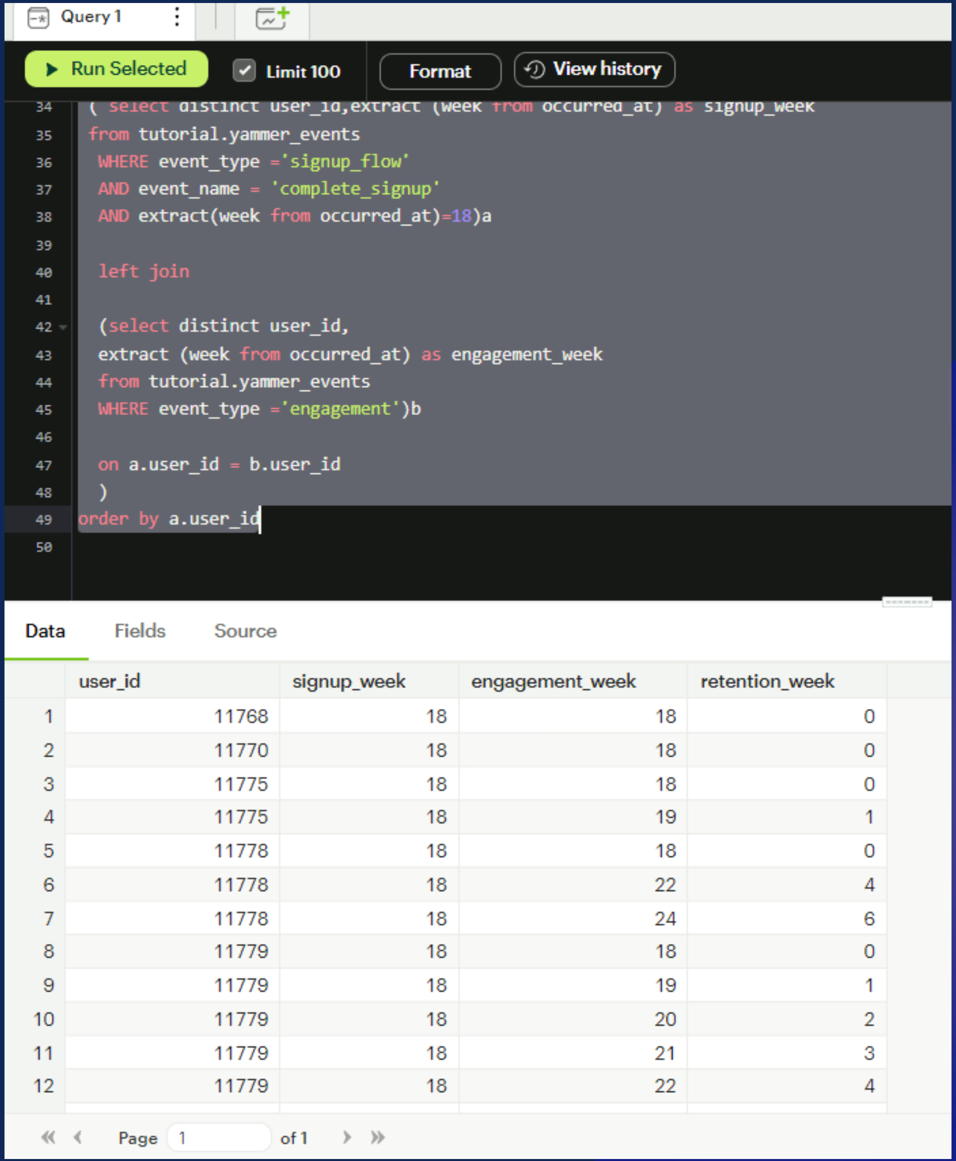
from tutorial.yammer_events

WHERE event_type ='engagement')

b

on a.user_id = b.user_id)

order by a.user_id



```
( select distinct user_id,extract (week from occurred_at) as signup_week
from tutorial.yammer_events
 WHERE event_type ='signup_flow'
 AND event_name = 'complete_signup'
 AND extract(week from occurred_at)=18)a

 left join

 (select distinct user_id,
 extract (week from occurred_at) as engagement_week
 from tutorial.yammer_events
 WHERE event_type ='engagement')b

 on a.user_id = b.user_id
 )
order by a.user_id
```

	user_id	signup_week	engagement_week	retention_week
1	11768	18	18	0
2	11770	18	18	0
3	11775	18	18	0
4	11775	18	19	1
5	11778	18	18	0
6	11778	18	22	4
7	11778	18	24	6
8	11779	18	18	0
9	11779	18	19	1
10	11779	18	20	2
11	11779	18	21	3
12	11779	18	22	4

TASK 4

D. Weekly Engagement Per Device:

- Objective: Measure the activeness of users on a weekly basis per device.
- Your Task: Write an SQL query to calculate the weekly engagement per device.

Query

Select

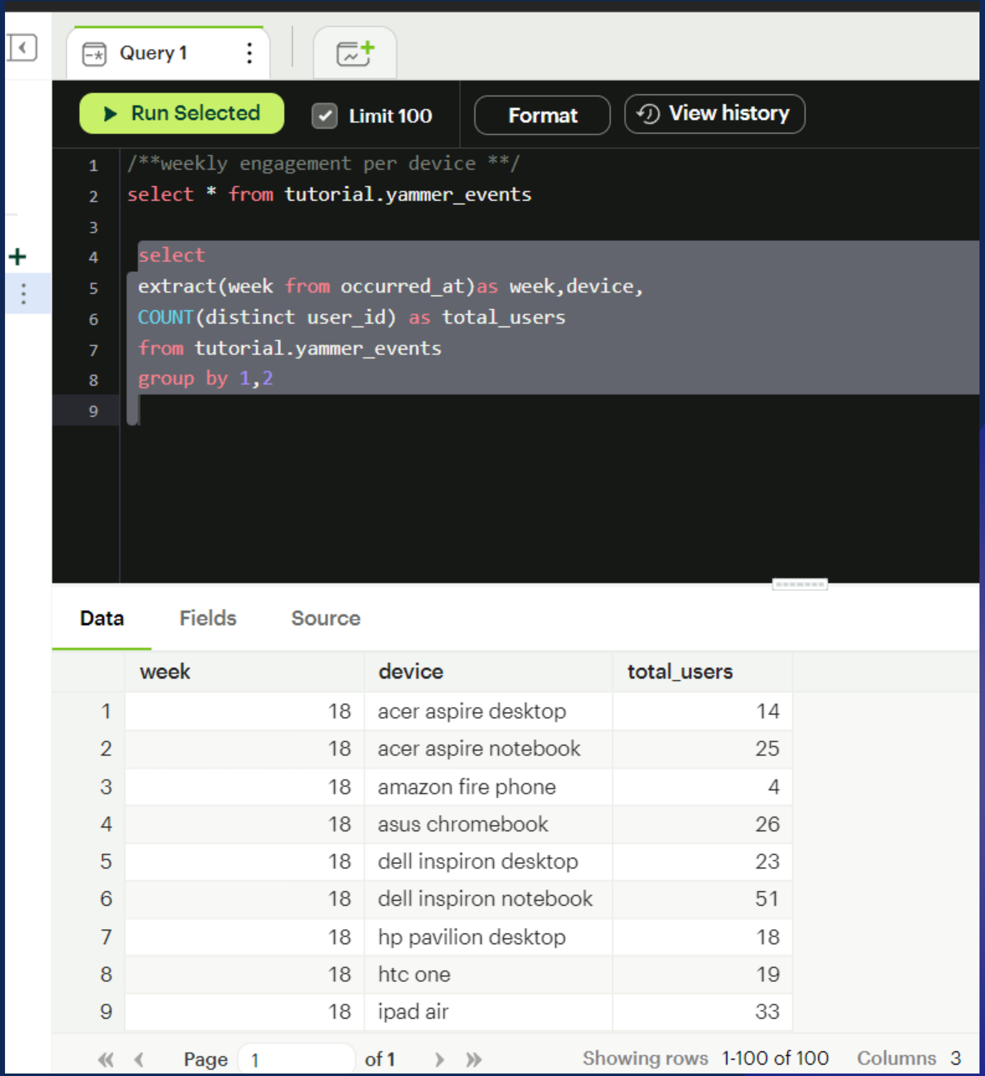
Extract(week from occurred_at) as week,

device,

count (distinct user_id) as total_users

From tutorial.yammer.events

Group 1,2;



```
/**weekly engagement per device **/
select * from tutorial.yammer_events

select
extract(week from occurred_at)as week,device,
COUNT(distinct user_id) as total_users
from tutorial.yammer_events
group by 1,2
```

	week	device	total_users
1	18	acer aspire desktop	14
2	18	acer aspire notebook	25
3	18	amazon fire phone	4
4	18	asus chromebook	26
5	18	dell inspiron desktop	23
6	18	dell inspiron notebook	51
7	18	hp pavilion desktop	18
8	18	htc one	19
9	18	ipad air	33

Page 1 of 1 Showing rows 1-100 of 100 Columns 3

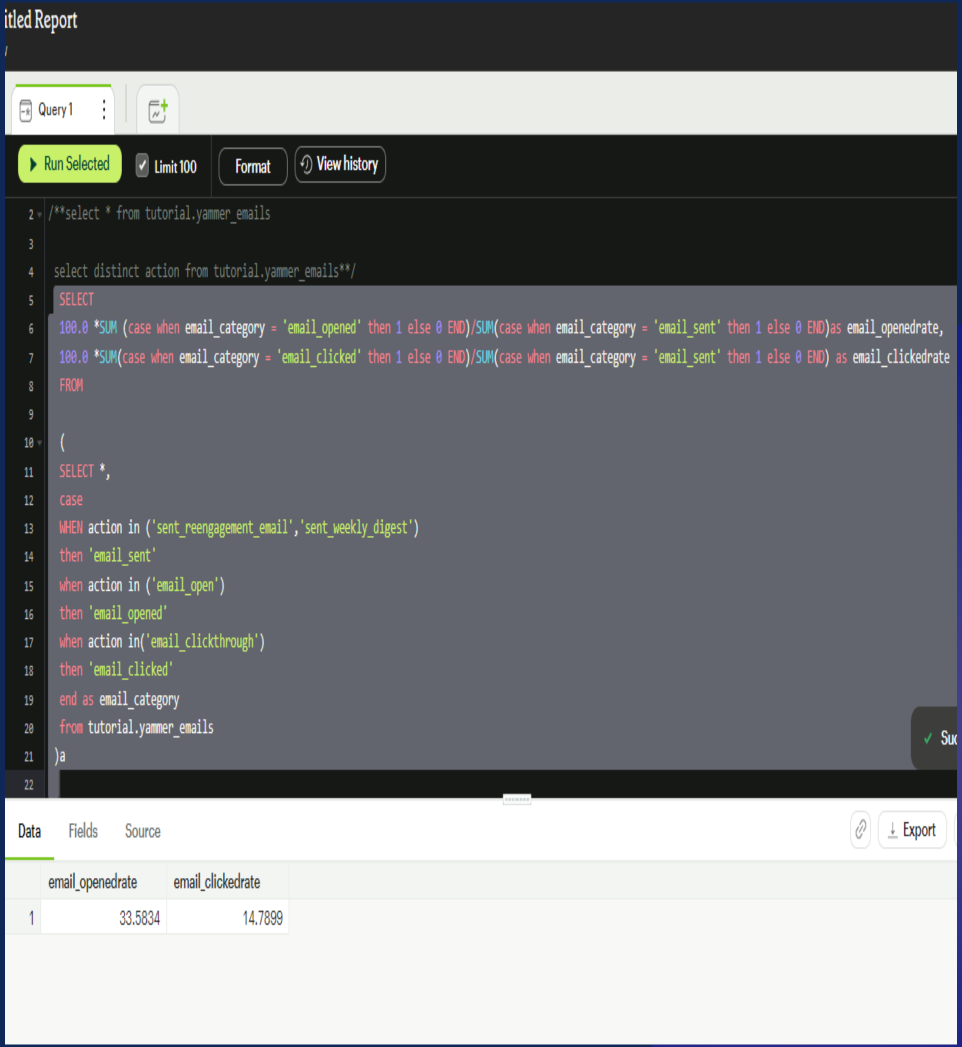
TASK 5

E. Email Engagement Analysis:

- ➢ Objective: Analyze how users are engaging with the email service.
- ➢ Your Task: Write an SQL query to calculate the email engagement metrics.

Query

```
SELECT

100.0 *SUM (case when email_category = 'email_opened' then 1 else 0 END)/SUM(case when email_category = 'email_sent' then 1 else 0 END)as email_openedrate,

100.0 *SUM(case when email_category = 'email_clicked' then 1 else 0 END)/SUM(case when email_category = 'email_sent' then 1 else 0 END) as email_clickedrate

FROM

(  SELECT *,

case   WHEN action in ('sent_reengagement_email','sent_weekly_digest')   then 'email_sent'  when action in ('email_open')  then 'email_opened'

 when action in('email_clickthrough')  then 'email_clicked' end as email_category

from tutorial.yammer_emails )a
```



```
/**select * from tutorial.yammer_emails

 select distinct action from tutorial.yammer_emails**/
 SELECT
 100.0 *SUM (case when email_category = 'email_opened' then 1 else 0 END)/SUM(case when email_category = 'email_sent' then 1 else 0 END)as email_openedrate,
 100.0 *SUM(case when email_category = 'email_clicked' then 1 else 0 END)/SUM(case when email_category = 'email_sent' then 1 else 0 END) as email_clickedrate
 FROM


 (
 SELECT *,
 case
 WHEN action in ('sent_reengagement_email','sent_weekly_digest')
 then 'email_sent'
 when action in ('email_open')
 then 'email_opened'
 when action in('email_clickthrough')
 then 'email_clicked'
 end as email_category
 from tutorial.yammer_emails
)a
```

	email_openedrate	email_clickedrate
1	33.5834	14.7899

THANK YOU