Project 3

Operation analytics and investigating metric spikes

Project description:-

To interpret the given into tables and the analyse the given questions/tasks.The datasets had data of the jobs based on which I have to derive certain insights.

Approach:-

I have spent some time on understanding the data/table given.

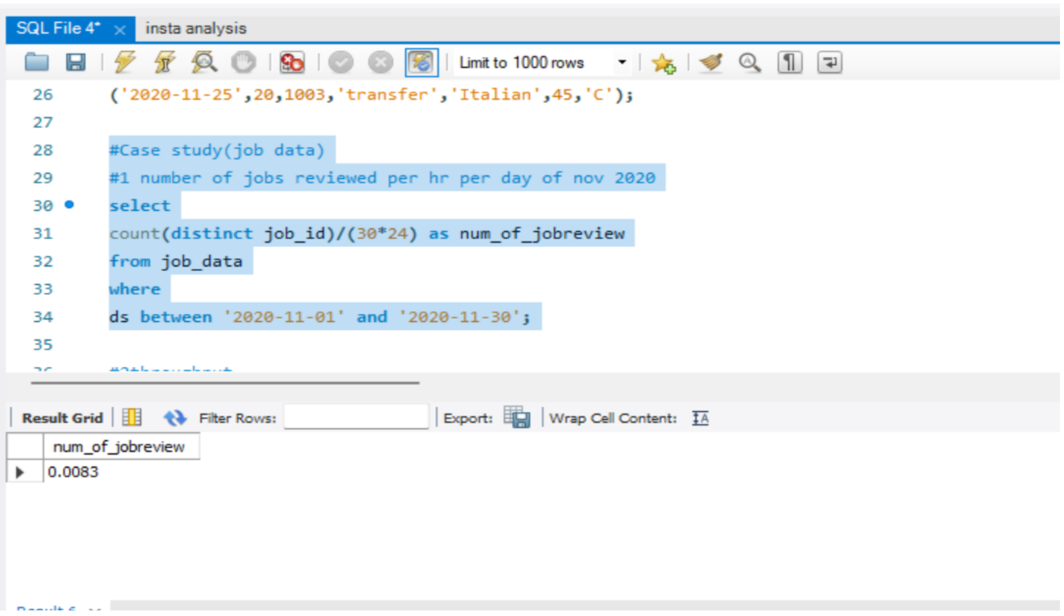
I just feed the data from the data sets into the tables in sql and then started deriving the useful insights from them. Starting with project 3 as my new database and tables using structures and tables provided.

Execution:-

Case study 1-Job data analysis

Tasks

A) jobs reviewed over time

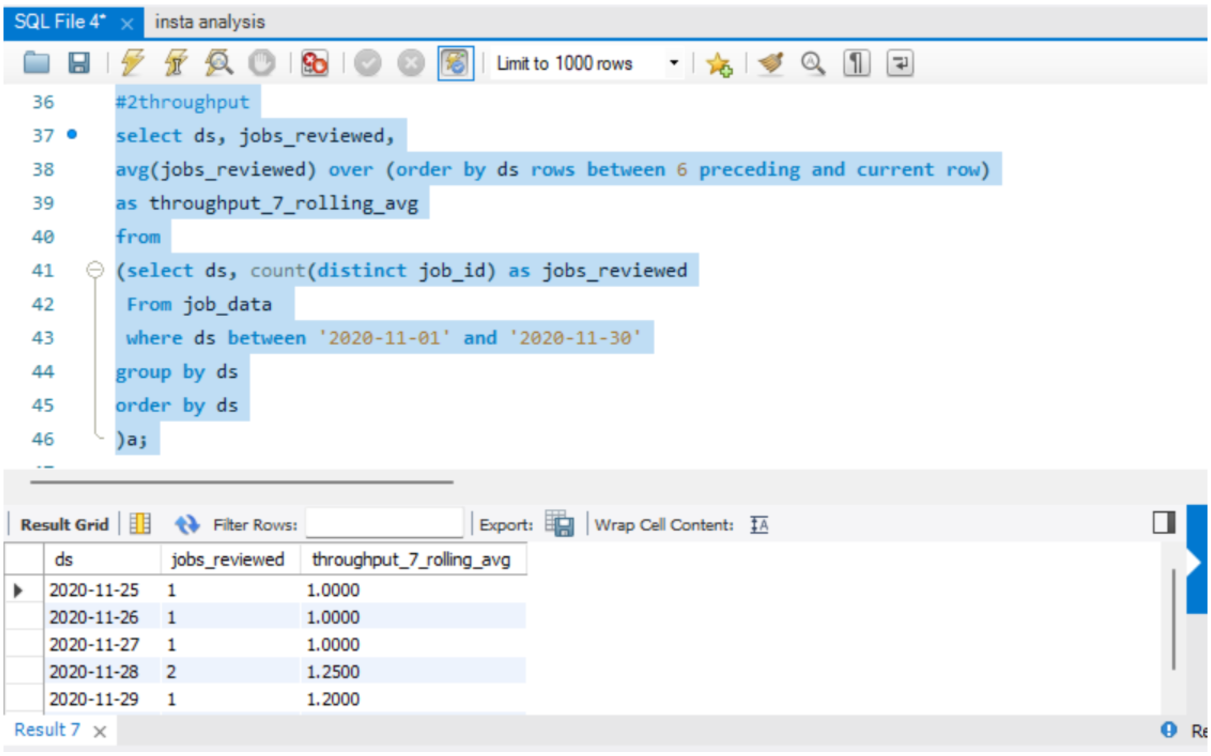


```
('2020-11-25',20,1003,'transfer','Italian',45,'C');

#Case study(job data)
#1 number of jobs reviewed per hr per day of nov 2020
select
count(distinct job_id)/(30*24) as num_of_jobreview
from job_data
where
ds between '2020-11-01' and '2020-11-30';
```

```
num_of_jobreview
0.0083
```

B)Throughput analysis:-

It’s the number of events happening per sec
Here we are deriving 7 day rolling average of throughput.

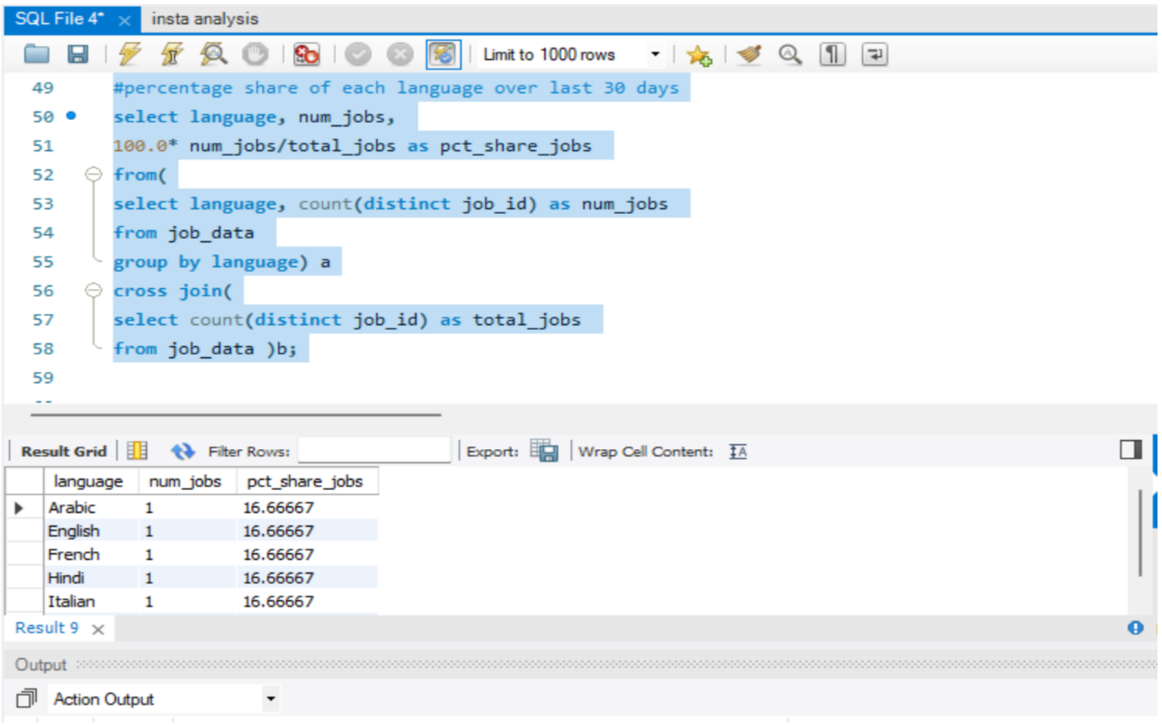


```sql
#2throughput
select ds, jobs_reviewed,
avg(jobs_reviewed) over (order by ds rows between 6 preceding and current row)
as throughput_7_rolling_avg
from
(select ds, count(distinct job_id) as jobs_reviewed
 From job_data
 where ds between '2020-11-01' and '2020-11-30'
group by ds
order by ds
)a;
```

ds	jobs_reviewed	throughput_7_rolling_avg
2020-11-25	1	1.0000
2020-11-26	1	1.0000
2020-11-27	1	1.0000
2020-11-28	2	1.2500
2020-11-29	1	1.2000

C)Language share analysis

Percentage share of each language in last 30 days

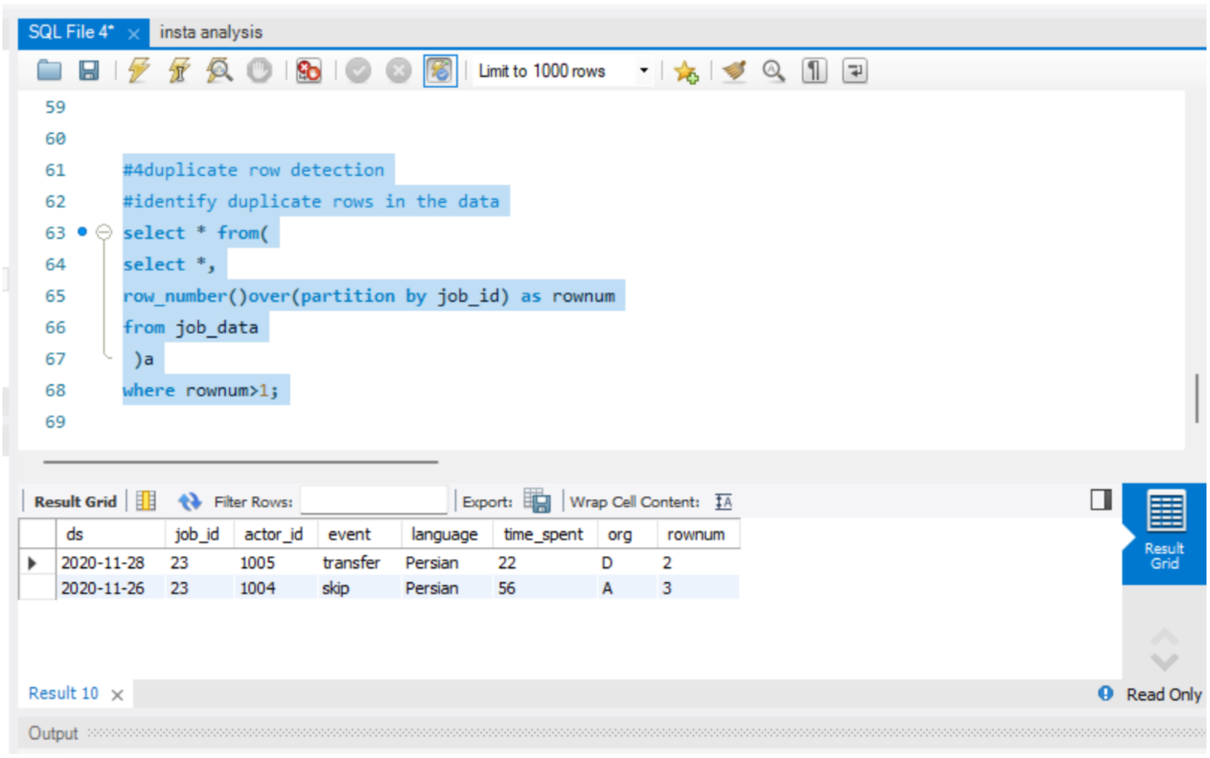


```sql
#percentage share of each language over last 30 days
select language, num_jobs,
100.0* num_jobs/total_jobs as pct_share_jobs
from(
select language, count(distinct job_id) as num_jobs
from job_data
group by language) a
cross join(
select count(distinct job_id) as total_jobs
from job_data )b;
```

language	num_jobs	pct_share_jobs
Arabic	1	16.66667
English	1	16.66667
French	1	16.66667
Hindi	1	16.66667
Italian	1	16.66667

D)Duplicate row detection

To identify duplicate rows in the data



SQL File 4* | insta analysis

Limit to 1000 rows

```
#4duplicate row detection
#identify duplicate rows in the data
select * from(
select *,
row_number()over(partition by job_id) as rownum
from job_data
 )a
where rownum>1;
```

```
ds          job_id  actor_id  event     language  time_spent  org  rownum
2020-11-28  23      1005      transfer  Persian   22          D    2
2020-11-26  23      1004      skip      Persian   56          A    3
```

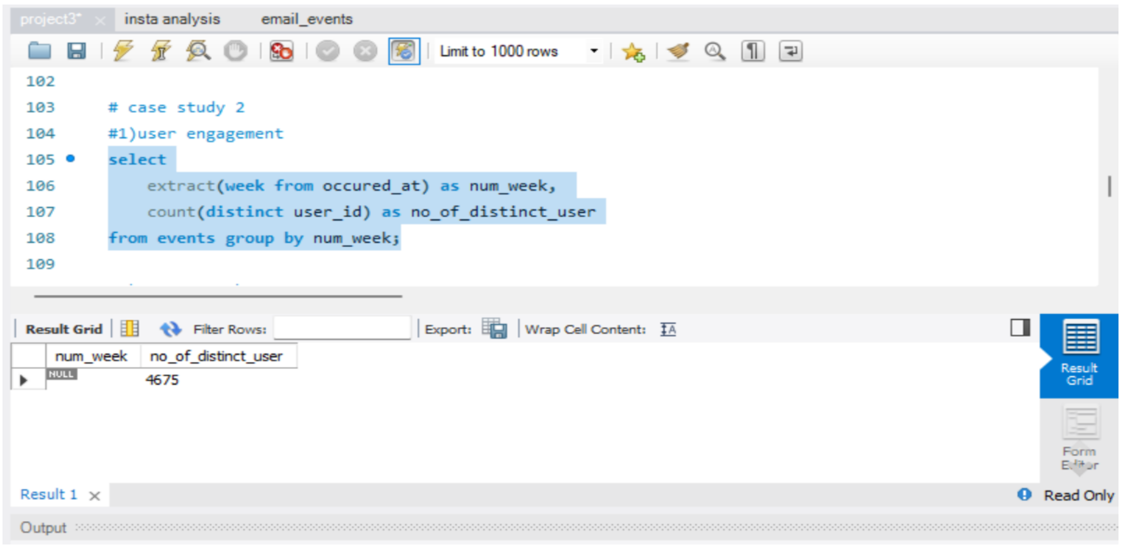
Result 10 | Read Only

Case study 2

(Investigating metric spike):

A) USER Engagement:

To measure the activeness of a user:



project3* | insta analysis | email_events

Limit to 1000 rows

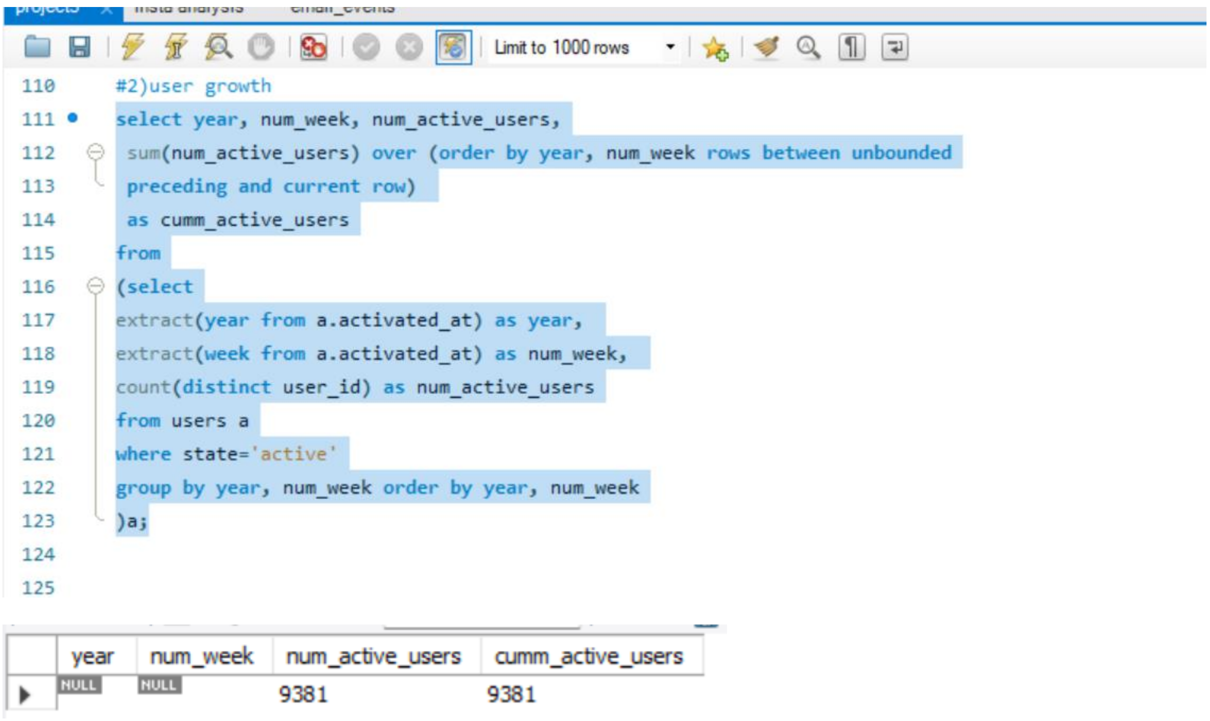
```
# case study 2
#1)user engagement
select
    extract(week from occured_at) as num_week,
    count(distinct user_id) as no_of_distinct_user
from events group by num_week;
```

```
num_week  no_of_distinct_user
NULL      4675
```

Result 1 | Read Only

B) User growth:-

Amount of users growing over time for a product.

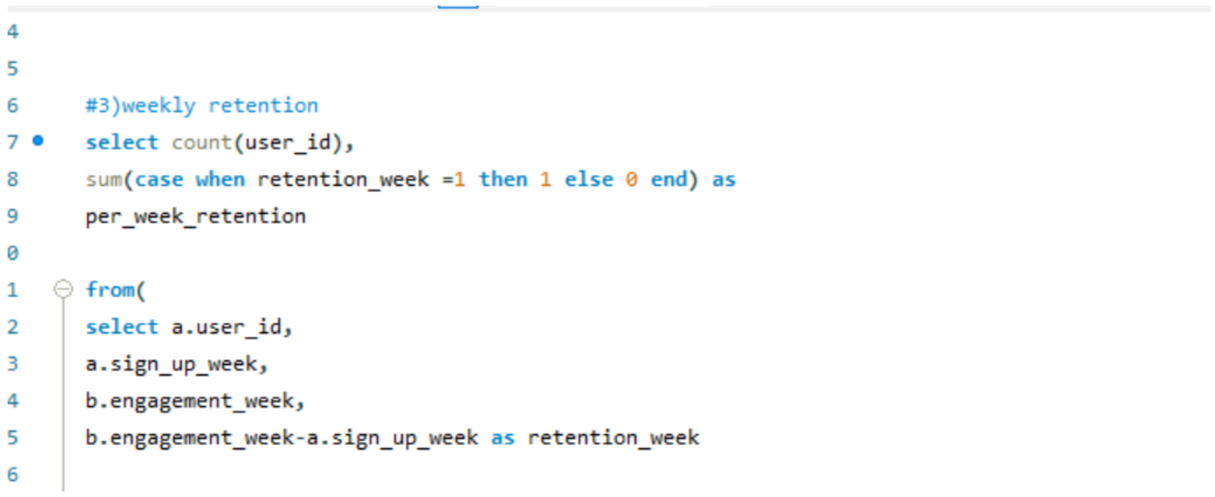


```
#2)user growth
select year, num_week, num_active_users,
 sum(num_active_users) over (order by year, num_week rows between unbounded
 preceding and current row)
 as cumm_active_users
from
(select
extract(year from a.activated_at) as year,
extract(week from a.activated_at) as num_week,
count(distinct user_id) as num_active_users
from users a
where state='active'
group by year, num_week order by year, num_week
)a;
```

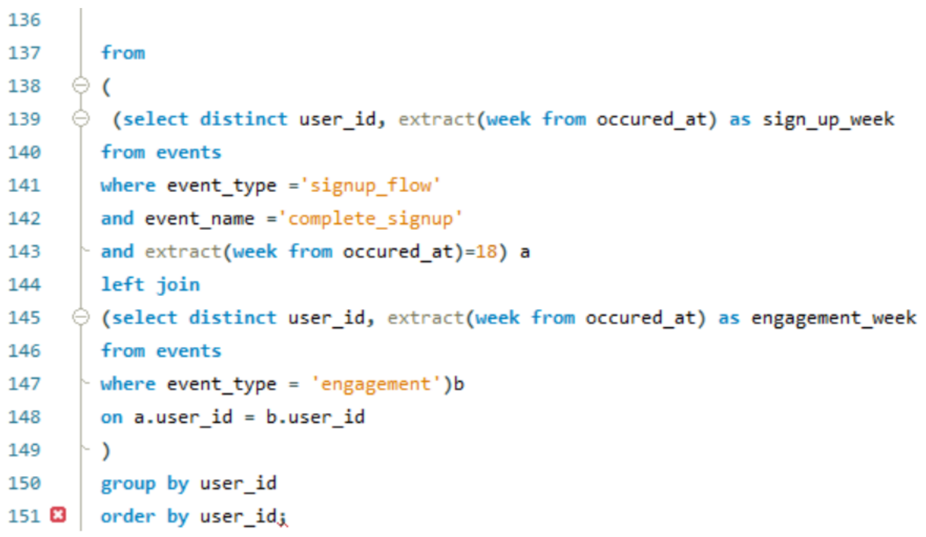
```
year   num_week   num_active_users   cumm_active_users
NULL   NULL       9381               9381
```

C) Weekly retention:-

Users getting retained weekly after signing-up for a product



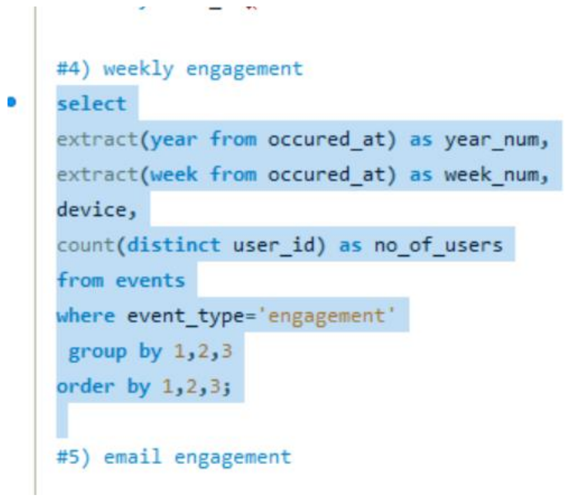
```
#3)weekly retention
select count(user_id),
sum(case when retention_week =1 then 1 else 0 end) as
per_week_retention

from(
select a.user_id,
a.sign_up_week,
b.engagement_week,
b.engagement_week-a.sign_up_week as retention_week
```



```
from
(
 (select distinct user_id, extract(week from occured_at) as sign_up_week
from events
where event_type ='signup_flow'
and event_name ='complete_signup'
and extract(week from occured_at)=18) a
left join
(select distinct user_id, extract(week from occured_at) as engagement_week
from events
where event_type = 'engagement')b
on a.user_id = b.user_id
)
group by user_id
order by user_id;
```

D) Weekly engagement:-

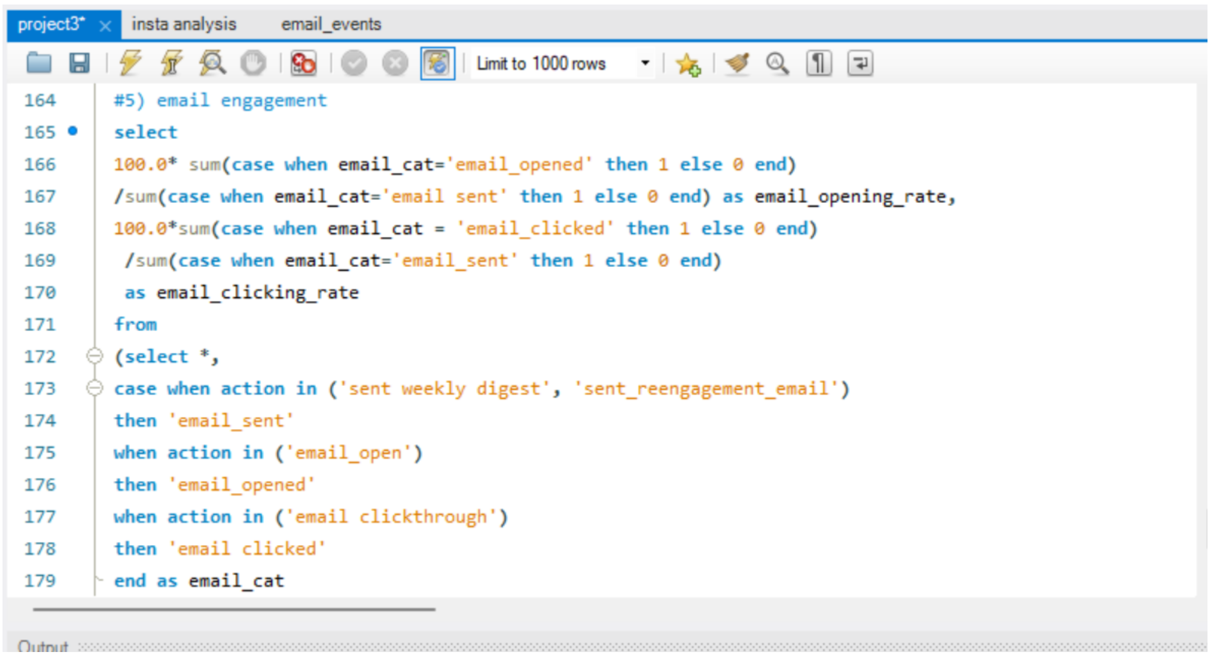
To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.



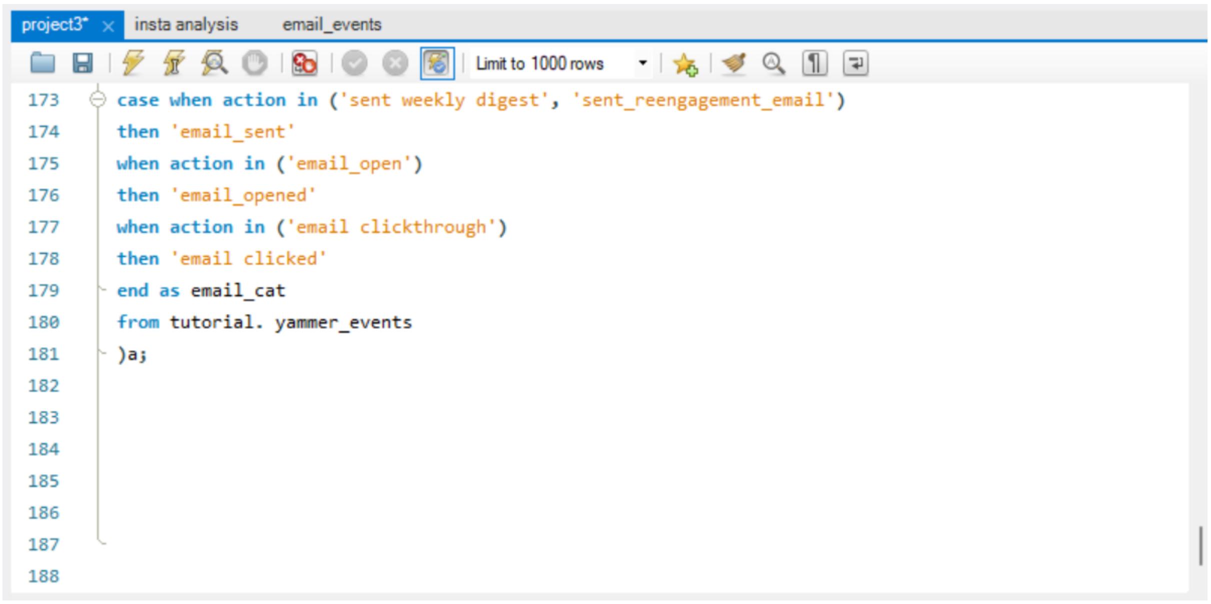
```
#4) weekly engagement
select
extract(year from occured_at) as year_num,
extract(week from occured_at) as week_num,
device,
count(distinct user_id) as no_of_users
from events
where event_type='engagement'
 group by 1,2,3
order by 1,2,3;

#5) email engagement
```

E) Email engagement:-

User engaging with the email service



```
#5) email engagement
select
100.0* sum(case when email_cat='email_opened' then 1 else 0 end)
/sum(case when email_cat='email sent' then 1 else 0 end) as email_opening_rate,
100.0*sum(case when email_cat = 'email_clicked' then 1 else 0 end)
 /sum(case when email_cat='email_sent' then 1 else 0 end)
 as email_clicking_rate
from
(select *,
case when action in ('sent weekly digest', 'sent_reengagement_email')
then 'email_sent'
when action in ('email_open')
then 'email_opened'
when action in ('email clickthrough')
then 'email clicked'
end as email_cat
```



```
case when action in ('sent weekly digest', 'sent_reengagement_email')
then 'email_sent'
when action in ('email_open')
then 'email_opened'
when action in ('email clickthrough')
then 'email clicked'
end as email_cat
from tutorial. yammer_events
)a;
```

Tech stack used

MySQL version 8.0

Office excel

Insights

1) The number of distinct jobs reviewed per hr per day for nov 2020 is 83%

2) We used the 7-day rolling average of throughput as it gives the average for all the days right
3) from day 1 to day 7 whereas, daily metric gives the average for only that particular day itself. The percentage share of Persian language is the most (37.5%).
4) There are two duplicate rows if we partition the data by job_id. But if we look the overall columns, all the rows are unique.

Case Study 2 (Investigating metric spike):

5) The weekly user engagement increased from week 18th to week 31st and then started declining from then onwards. This means that some of the users do not find much quality in the product/service in the last of the weeks.

6) There are in total 9381 active users from 1 week of 2013 to the 35th week of 2014.

7) The overall count of weekly engagement per device used is the most for MacBook users and iPhone users.

8) The email opening rate is around 34% and email clicking rate is around 15%. The users are engaging with the email service which is good for the company to expand.