Project - 3

Operation Analytics and Investigating Metric Spike

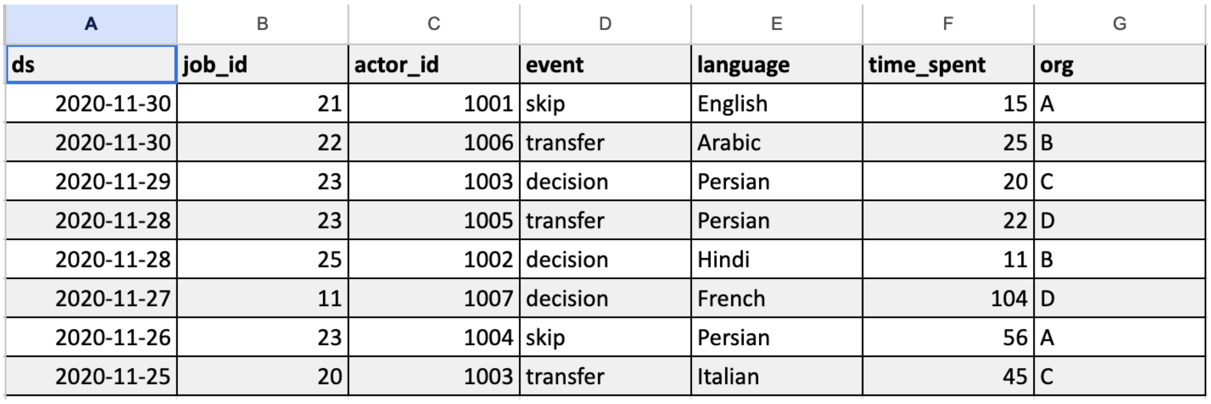
Project Description : Operation Analytics and Investigating Metric Spike

Approach : Microsoft designated as Data Analyst Lead and is provided with different data sets, tables from which you must derive certain insights out of it and answer the questions asked by different departments.

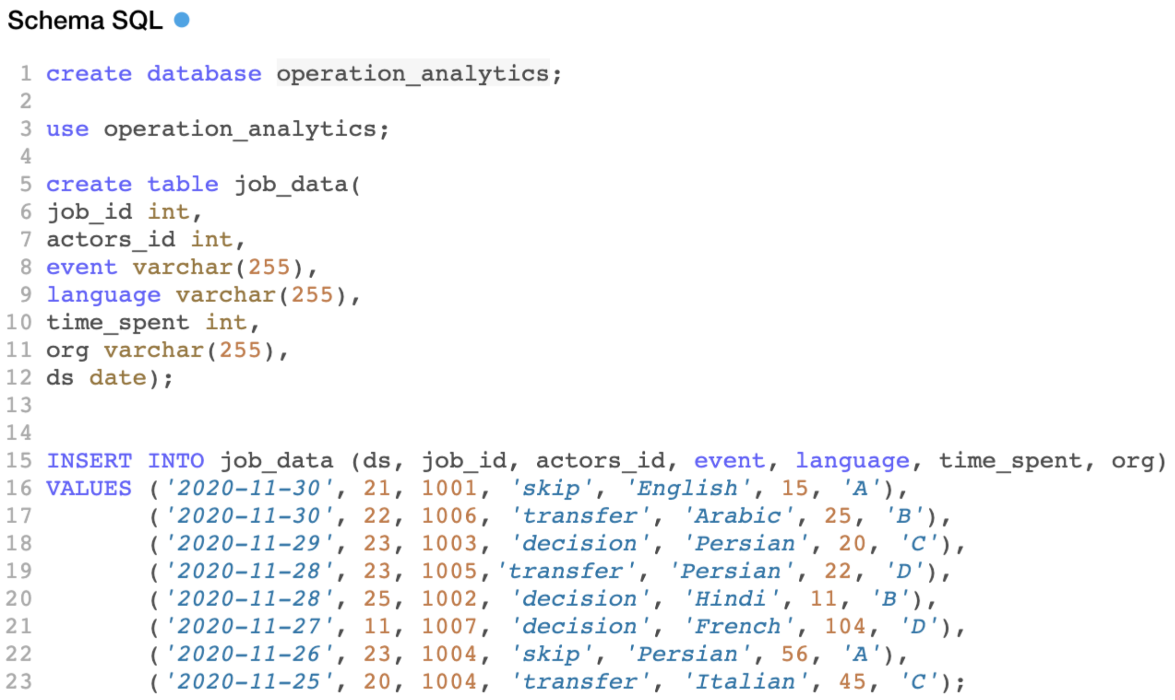
Tech-Stack Used : MySQL, Microsoft Excel, Google Drive, https://www.db-fiddle.com/

Insights :

Results : Case Study 1 (Job Data)



A	B	C	D	E	F	G
ds	job_id	actor_id	event	language	time_spent	org
2020-11-30	21	1001	skip	English	15	A
2020-11-30	22	1006	transfer	Arabic	25	B
2020-11-29	23	1003	decision	Persian	20	C
2020-11-28	23	1005	transfer	Persian	22	D
2020-11-28	25	1002	decision	Hindi	11	B
2020-11-27	11	1007	decision	French	104	D
2020-11-26	23	1004	skip	Persian	56	A
2020-11-25	20	1003	transfer	Italian	45	C



Schema SQL

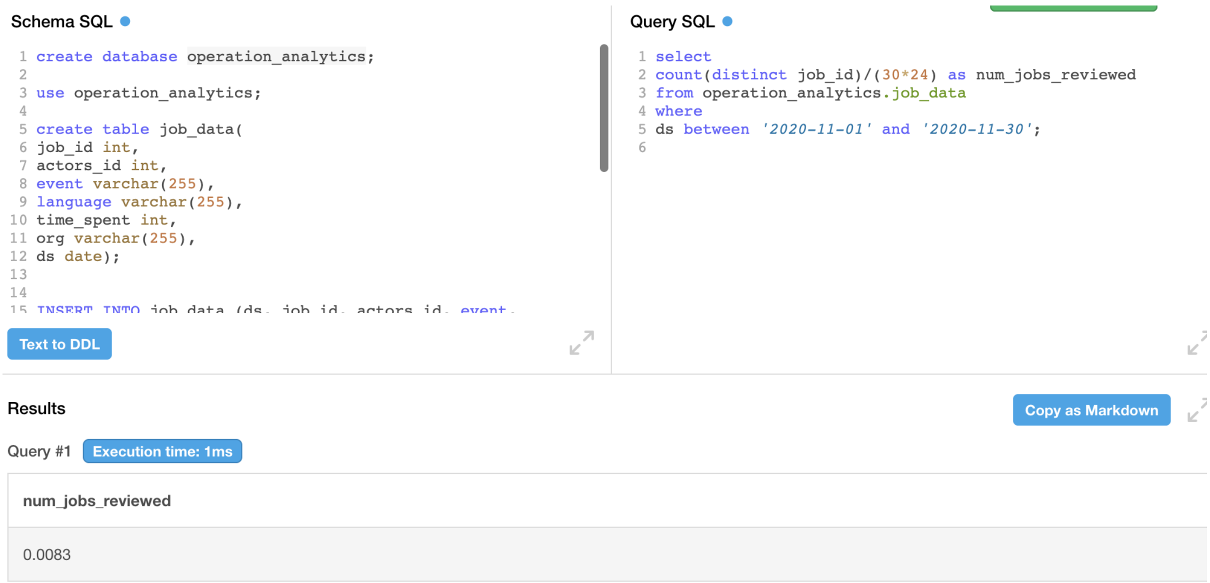
```
create database operation_analytics;

use operation_analytics;

create table job_data(
job_id int,
actors_id int,
event varchar(255),
language varchar(255),
time_spent int,
org varchar(255),
ds date);


INSERT INTO job_data (ds, job_id, actors_id, event, language, time_spent, org)
VALUES ('2020-11-30', 21, 1001, 'skip', 'English', 15, 'A'),
       ('2020-11-30', 22, 1006, 'transfer', 'Arabic', 25, 'B'),
       ('2020-11-29', 23, 1003, 'decision', 'Persian', 20, 'C'),
       ('2020-11-28', 23, 1005,'transfer', 'Persian', 22, 'D'),
       ('2020-11-28', 25, 1002, 'decision', 'Hindi', 11, 'B'),
       ('2020-11-27', 11, 1007, 'decision', 'French', 104, 'D'),
       ('2020-11-26', 23, 1004, 'skip', 'Persian', 56, 'A'),
       ('2020-11-25', 20, 1004, 'transfer', 'Italian', 45, 'C');
```

A. Calculate the number of jobs reviewed per hour per day for November 2020?



Schema SQL

```
create database operation_analytics;

use operation_analytics;

create table job_data(
job_id int,
actors_id int,
event varchar(255),
language varchar(255),
time_spent int,
org varchar(255),
ds date);


INSERT INTO job_data (ds, job_id, actors_id, event,
```

Text to DDL

Query SQL

```
select
count(distinct job_id)/(30*24) as num_jobs_reviewed
from operation_analytics.job_data
where
ds between '2020-11-01' and '2020-11-30';

```

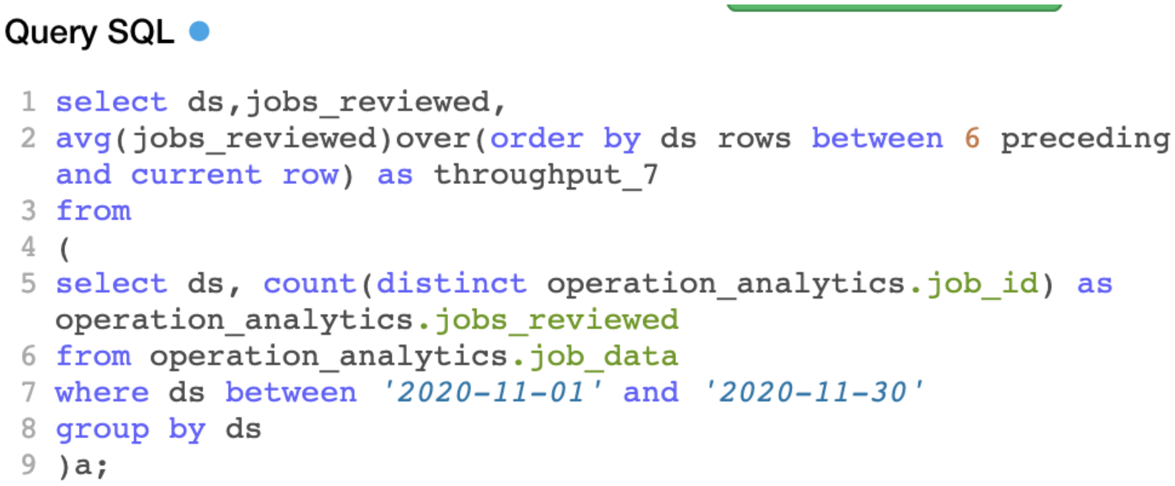
Results

Copy as Markdown

Query #1 Execution time: 1ms

num_jobs_reviewed
0.0083

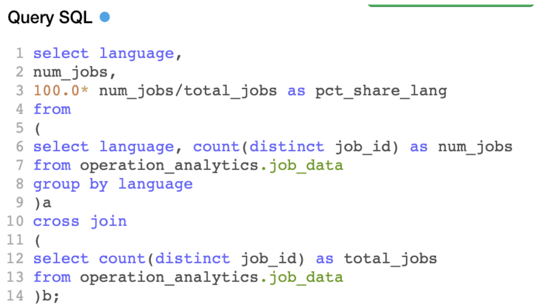
B. Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?



Query SQL

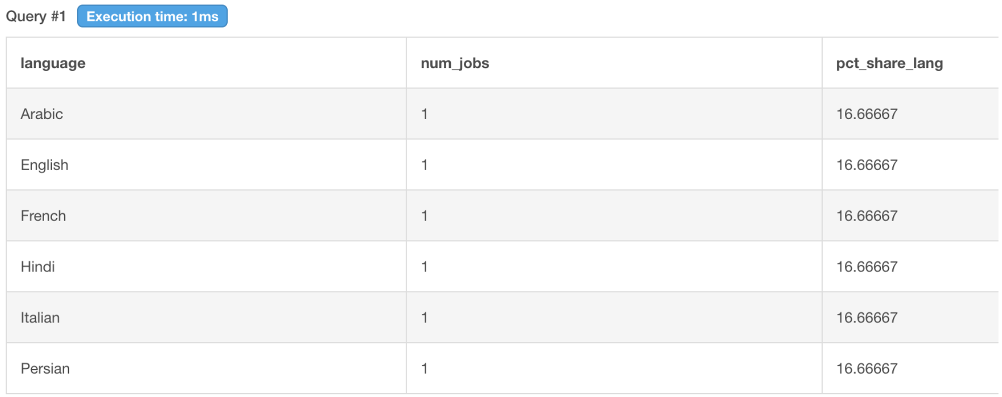
```
select ds,jobs_reviewed,
avg(jobs_reviewed)over(order by ds rows between 6 preceding and current row) as throughput_7
from
(
select ds, count(distinct operation_analytics.job_id) as operation_analytics.jobs_reviewed
from operation_analytics.job_data
where ds between '2020-11-01' and '2020-11-30'
group by ds
)a;
```

C. Calculate the percentage share of each language in the last 30 days?



Query SQL

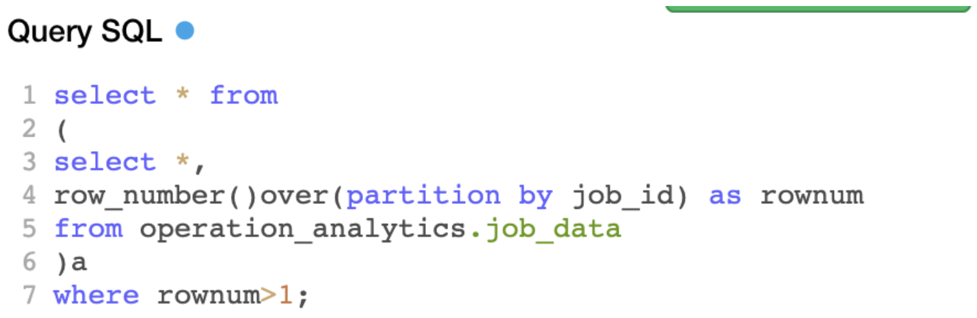
```
select language,
num_jobs,
100.0* num_jobs/total_jobs as pct_share_lang
from
(
select language, count(distinct job_id) as num_jobs
from operation_analytics.job_data
group by language
)a
cross join
(
select count(distinct job_id) as total_jobs
from operation_analytics.job_data
)b;
```



Query #1 Execution time: 1ms

language	num_jobs	pct_share_lang
Arabic	1	16.66667
English	1	16.66667
French	1	16.66667
Hindi	1	16.66667
Italian	1	16.66667
Persian	1	16.66667

D. Let’s say you see some duplicate rows in the data. How will you display duplicates from the table?

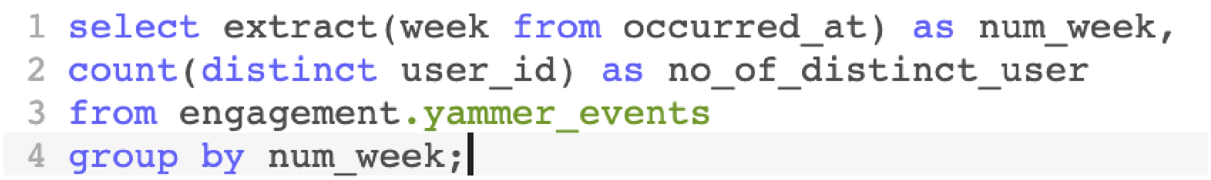


Query SQL ●

```sql
select * from
(
select *,
row_number()over(partition by job_id) as rownum
from operation_analytics.job_data
)a
where rownum>1;
```

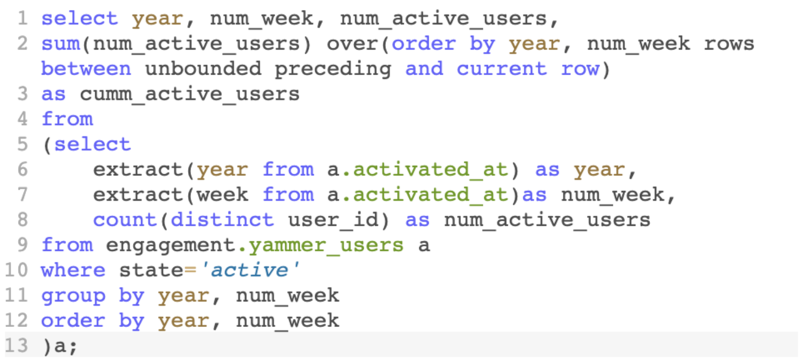
Case Study 2 (Investigating metric spike)

A. Calculate the weekly user engagement?



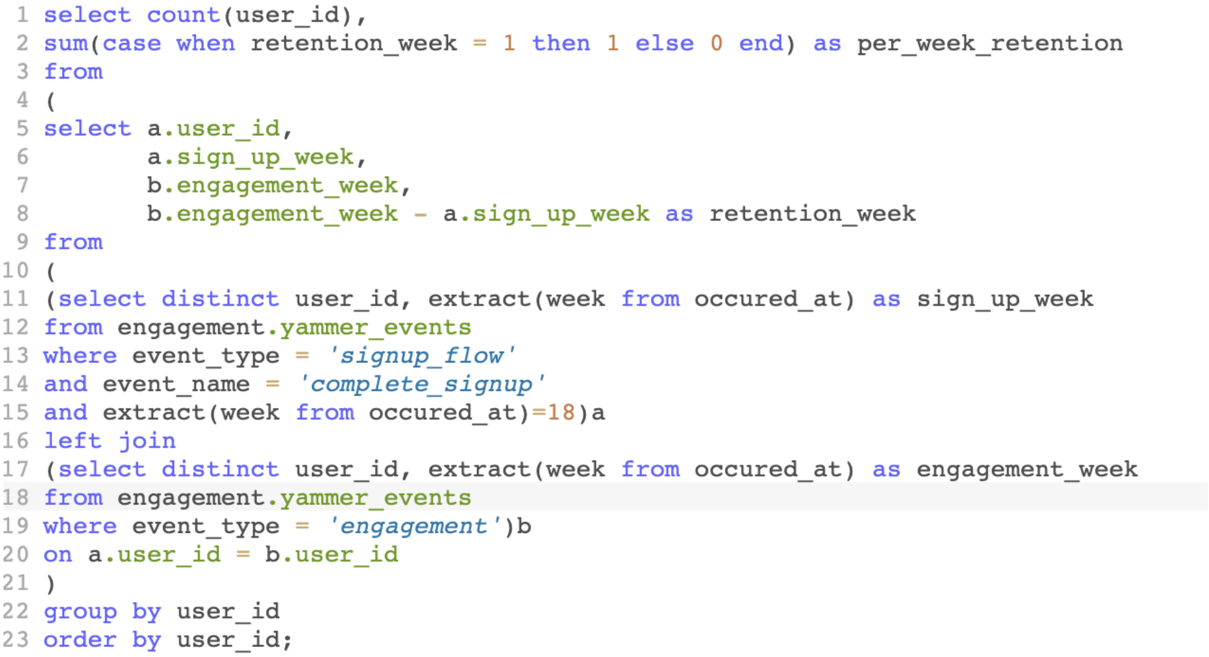
```sql
select extract(week from occurred_at) as num_week,
count(distinct user_id) as no_of_distinct_user
from engagement.yammer_events
group by num_week;
```

B. Calculate the user growth for product?



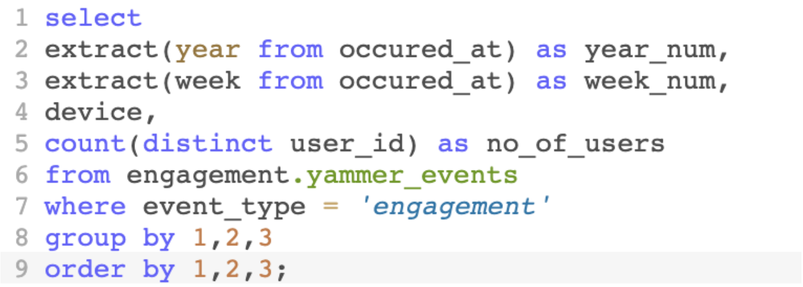
```sql
select year, num_week, num_active_users,
sum(num_active_users) over(order by year, num_week rows between unbounded preceding and current row)
as cumm_active_users
from
(select
    extract(year from a.activated_at) as year,
    extract(week from a.activated_at)as num_week,
    count(distinct user_id)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

C. Calculate the weekly retention of users-sign up cohort?



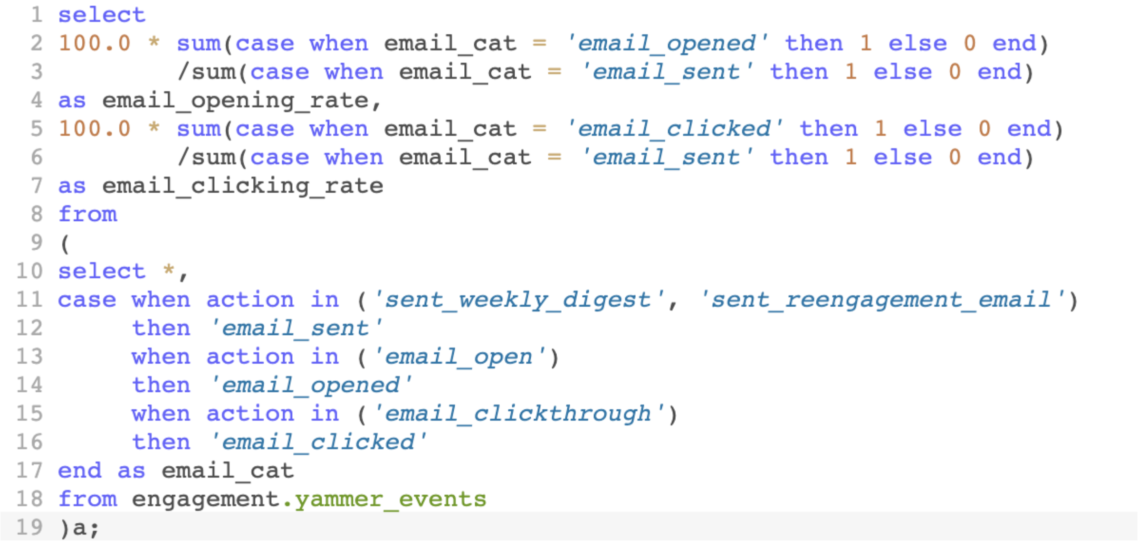
```sql
select count(user_id),
sum(case when retention_week = 1 then 1 else 0 end) as per_week_retention
from
(
select a.user_id,
       a.sign_up_week,
       b.engagement_week,
       b.engagement_week - a.sign_up_week as retention_week
from
(
(select distinct user_id, extract(week from occured_at) as sign_up_week
from engagement.yammer_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 engagement.yammer_events
where event_type = 'engagement')b
on a.user_id = b.user_id
)
group by user_id
order by user_id;
```

D. Calculate the weekly engagement per device?



```sql
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 engagement.yammer_events
where event_type = 'engagement'
group by 1,2,3
order by 1,2,3;
```

E. Calculate the email engagement metrics?



```sql
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
from engagement.yammer_events
)a;
```