Operation Analytics and Investigating Metric Spike

Project Description:

Operation Analytics is the analysis done for the complete end to end operations of a company. With the help of this, the company then finds the areas on which it must improve upon. Being one of the most important parts of a company, this kind of analysis is further used to predict the overall growth or decline of a company's fortune. In this project we have analyses job reviews, no of events happened in company, how many languages speak in company and analyze duplication record.

Investigating metric spike is also an important part of operation analytics as being a Data Analyst you must be able to understand or make other teams understand questions like- User Engagement, User Growth, Weekly Retention, Weekly Engagement, Email Engagement.

Approach:

First I carefully understood the project requirement and go through the dataset to better understand it then I loaded the data into SQL database and perform exploratory analysis by writing SQL query to understand the data and draw out useful insights which help answer the required question to marketing, support and development team.

Tech-Stack Used:

I used MySQL Workbench 8.0 CE for Case Study 1 & online platform Mode.com for Case Study 2 as loading data in MySQL Workbench through Table data import wizard option takes a lot of time whereas all those dataset were preloaded in mode.com

Insights:

Using this project I was able to apply the SQL knowledge and get useful insight from raw data, I also got to know how to import CSV data into MySQL workbench through command prompt load data infile method which was way faster than workbench import wizard tool, I also learned the usage of OVER function using this project.

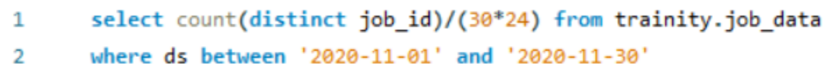
Result:

I derived the several data from the given data and the data I derived is mentioned below And this helped me to understand the real-time SQL commands.

Case Study 1 (Job Data):

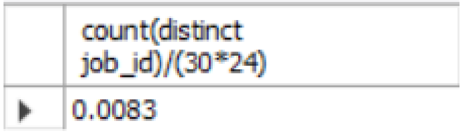
Task 1: Calculate the number of jobs reviewed per hour per day for November 2020.

Query:



```sql
select count(distinct job_id)/(30*24) from trainity.job_data
where ds between '2020-11-01' and '2020-11-30'
```

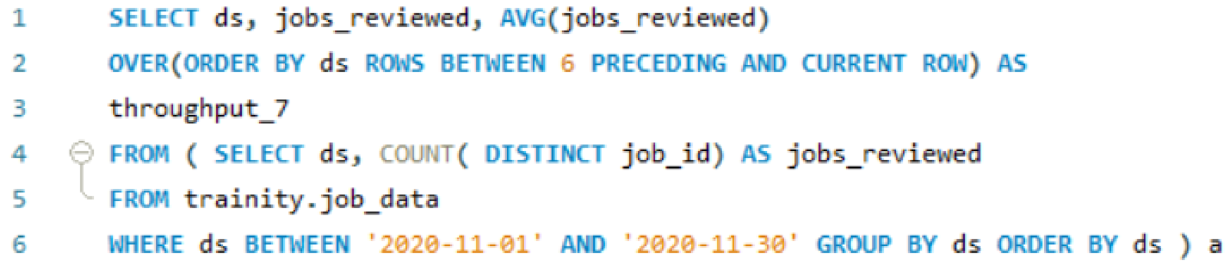
Result:



	count(distinct job_id)/(30*24)
▶	0.0083

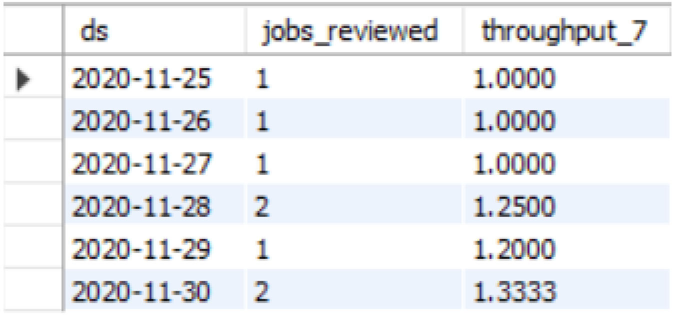
Task 2: Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

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 job_id) AS jobs_reviewed
FROM trainity.job_data
WHERE ds BETWEEN '2020-11-01' AND '2020-11-30' GROUP BY ds ORDER BY ds ) a
```

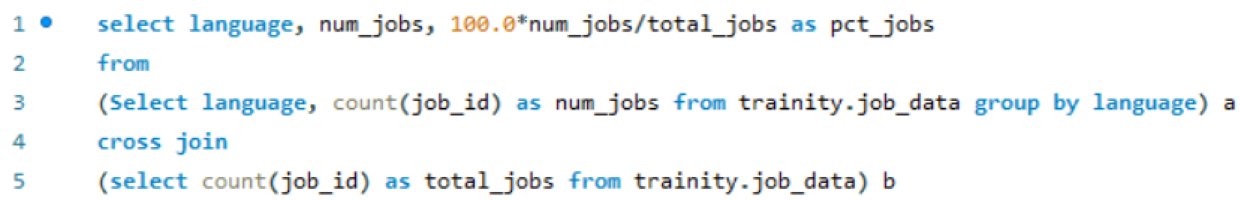
Result:



	ds	jobs_reviewed	throughput_7
▶	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
	2020-11-30	2	1.3333

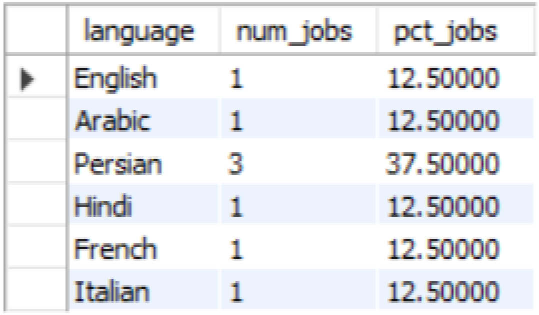
Task 3: Calculate the percentage share of each language in the last 30 days?

Query:



```
select language, num_jobs, 100.0*num_jobs/total_jobs as pct_jobs
from
(Select language, count(job_id) as num_jobs from trainity.job_data group by language) a
cross join
(select count(job_id) as total_jobs from trainity.job_data) b
```

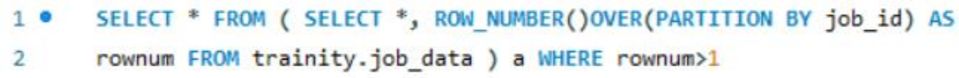
Result:



	language	num_jobs	pct_jobs
▶	English	1	12.50000
	Arabic	1	12.50000
	Persian	3	37.50000
	Hindi	1	12.50000
	French	1	12.50000
	Italian	1	12.50000

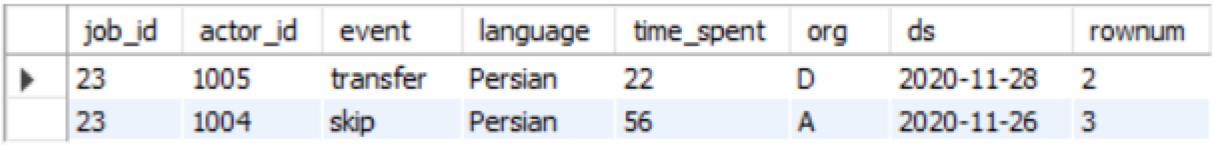
<u>Task 4</u>: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

Query:



```
SELECT * FROM ( SELECT *, ROW_NUMBER()OVER(PARTITION BY job_id) AS
rownum FROM trainity.job_data ) a WHERE rownum>1
```

Result:

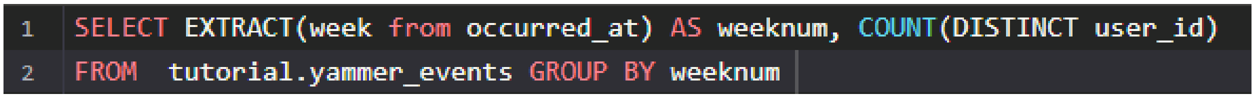


	job_id	actor_id	event	language	time_spent	org	ds	rownum
▶	23	1005	transfer	Persian	22	D	2020-11-28	2
	23	1004	skip	Persian	56	A	2020-11-26	3

<u>Case Study 2 (Investigating Metric Spike):</u>

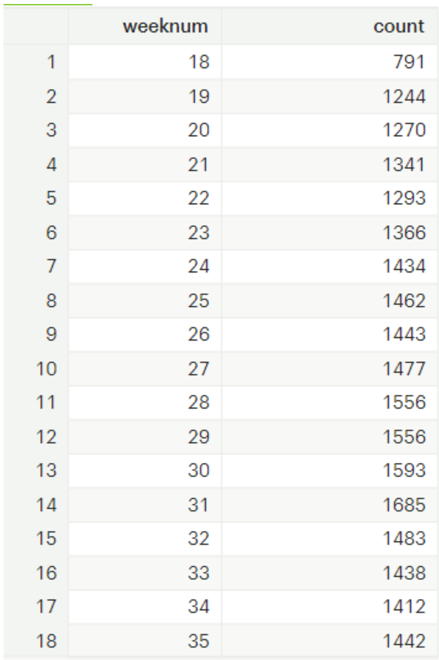
<u>Task 1</u>: Calculate the weekly user engagement?

Query:



```
SELECT EXTRACT(week from occurred_at) AS weeknum, COUNT(DISTINCT user_id)
FROM  tutorial.yammer_events GROUP BY weeknum
```

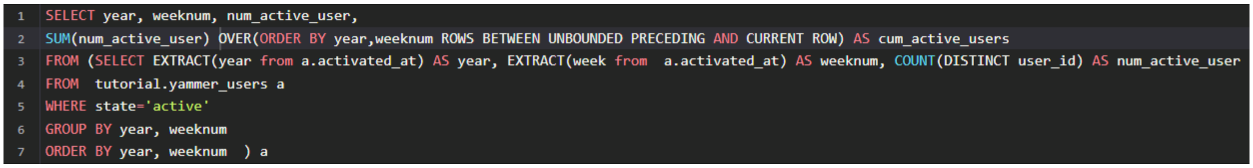
Result:



	weeknum	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
10	27	1477
11	28	1556
12	29	1556
13	30	1593
14	31	1685
15	32	1483
16	33	1438
17	34	1412
18	35	1442

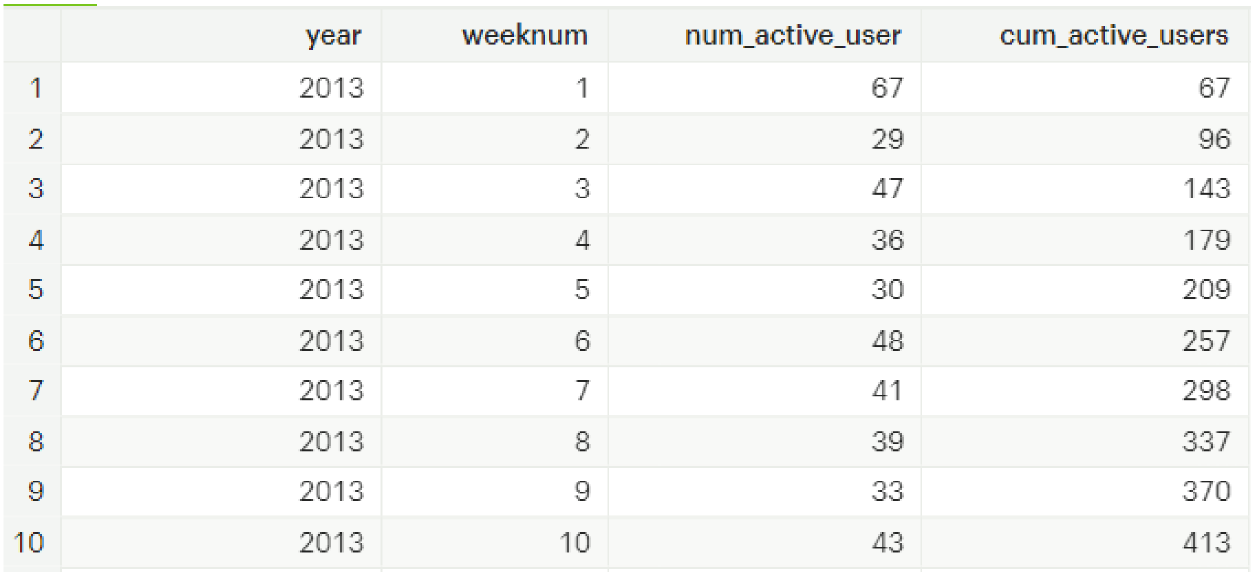
<u>Task 2</u>: Calculate the user growth for product?

Query:



```
SELECT year, weeknum, num_active_user,
SUM(num_active_user) OVER(ORDER BY year,weeknum ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cum_active_users
FROM (SELECT EXTRACT(year from a.activated_at) AS year, EXTRACT(week from  a.activated_at) AS weeknum, COUNT(DISTINCT user_id) AS num_active_user
FROM  tutorial.yammer_users a
WHERE state='active'
GROUP BY year, weeknum
ORDER BY year, weeknum  ) a
```

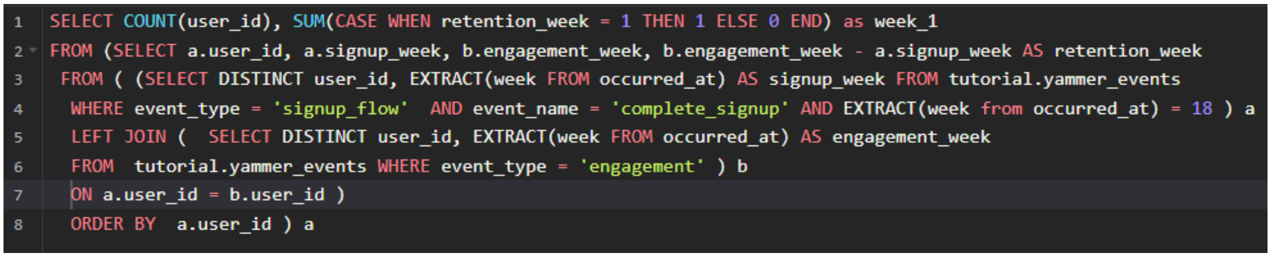
Result:



	year	weeknum	num_active_user	cum_active_users
1	2013	1	67	67
2	2013	2	29	96
3	2013	3	47	143
4	2013	4	36	179
5	2013	5	30	209
6	2013	6	48	257
7	2013	7	41	298
8	2013	8	39	337
9	2013	9	33	370
10	2013	10	43	413

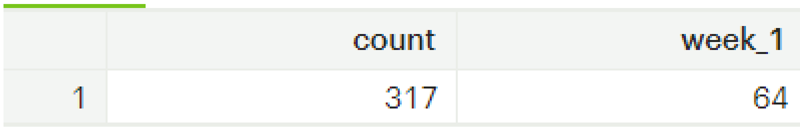
Task 3: Calculate the weekly retention of users-sign up cohort?

Query:



```sql
SELECT COUNT(user_id), SUM(CASE WHEN retention_week = 1 THEN 1 ELSE 0 END) as week_1
FROM (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 ) a
```

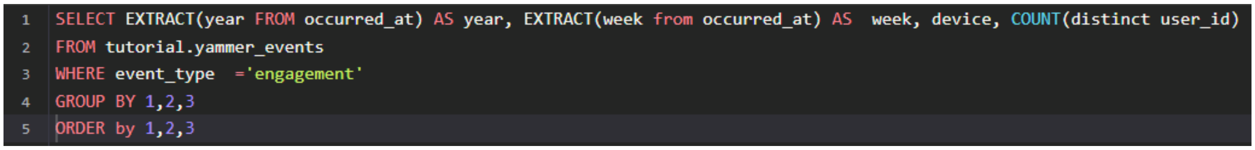
Result:



	count	week_1
1	317	64

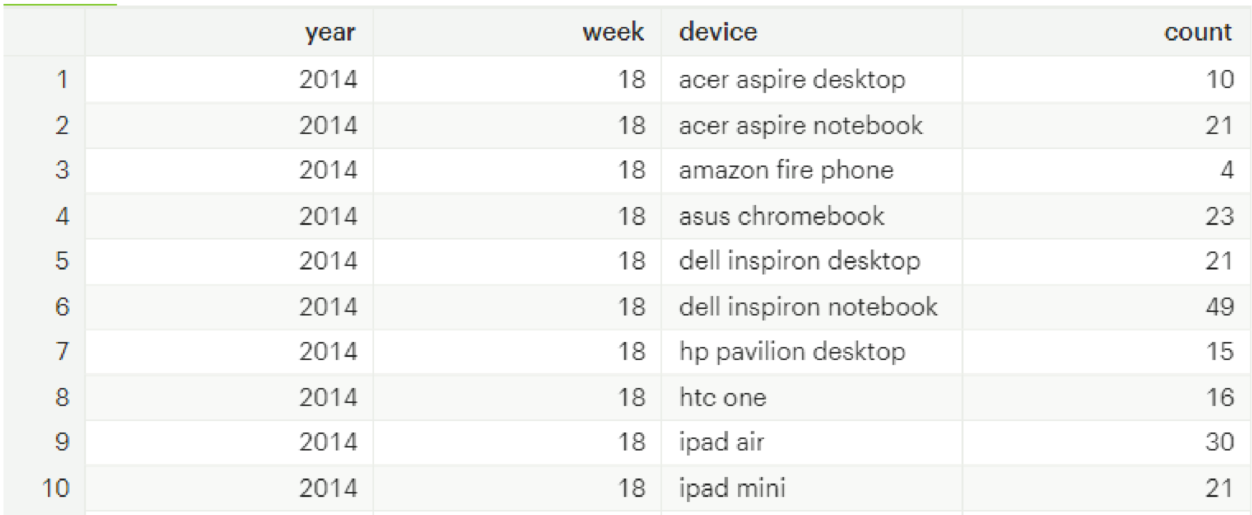
Task 4: Calculate the weekly engagement per device?

Query:



```sql
SELECT EXTRACT(year FROM occurred_at) AS year, EXTRACT(week from occurred_at) AS  week, device, COUNT(distinct user_id)
FROM tutorial.yammer_events
WHERE event_type  ='engagement'
GROUP BY 1,2,3
ORDER by 1,2,3
```

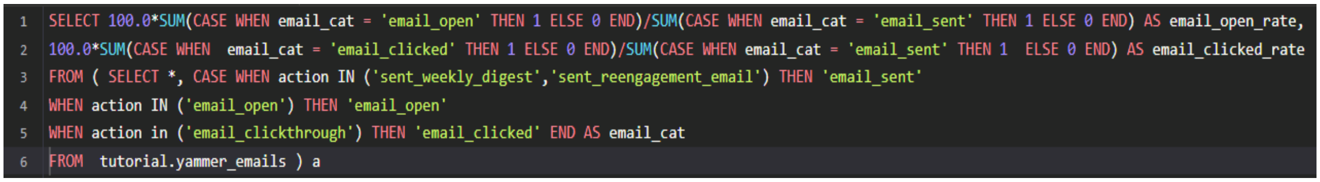
Result:



	year	week	device	count
1	2014	18	acer aspire desktop	10
2	2014	18	acer aspire notebook	21
3	2014	18	amazon fire phone	4
4	2014	18	asus chromebook	23
5	2014	18	dell inspiron desktop	21
6	2014	18	dell inspiron notebook	49
7	2014	18	hp pavilion desktop	15
8	2014	18	htc one	16
9	2014	18	ipad air	30
10	2014	18	ipad mini	21

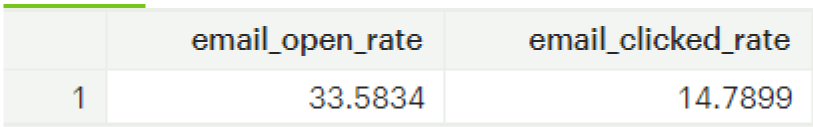
Task 5: Calculate the email engagement metrics?

Query:



```
SELECT 100.0*SUM(CASE WHEN email_cat = 'email_open' THEN 1 ELSE 0 END)/SUM(CASE WHEN email_cat = 'email_sent' THEN 1 ELSE 0 END) AS email_open_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_clicked_rate
FROM ( SELECT *, CASE WHEN action IN ('sent_weekly_digest','sent_reengagement_email') THEN 'email_sent'
WHEN action IN ('email_open') THEN 'email_open'
WHEN action in ('email_clickthrough') THEN 'email_clicked' END AS email_cat
FROM  tutorial.yammer_emails ) a
```

Result:



	email_open_rate	email_clicked_rate
1	33.5834	14.7899