Operation Analytics and Investigating Metric Spike

NAME: J.SUSRITHA
GMAIL: SUSRITHAJ2102@GMAIL.COM

Agenda

- Project Description
- Approach
- Tech-Stack Used
- Insights
- Result

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. You work closely with the ops team, support team, marketing team, etc. and help them derive insights out of the data they collect.

- 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. It means better automation, better understanding between cross-functional teams, and more effective workflows.

- 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- Why is there a dip in daily engagement? Why have sales taken a dip? Etc. Questions like these must be answered daily and for that its very important to investigate metric spike.

- You are working for a company like 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.

Approach

- For the first case study, the number of jobs reviewed can be calculated by counting the number of rows in the job_data table. The number of jobs reviewed per hour per day for November 2020 can be calculated by filtering the job_data table for November 2020 and then aggregating the number of jobs by hour and day. The throughput can be calculated by counting the number of events per second and then computing the 7-day rolling average of the throughput. For the throughput metric, the 7-day rolling average is preferred because it helps to smooth out fluctuations in the data and provides a clearer picture of the underlying trend. The percentage share of each language can be calculated by aggregating the number of jobs by language and then dividing each count by the total number of jobs. To display duplicates from the table, one can group the data by all columns and then filter for groups with more than one row.

- For the second case study, the weekly user engagement can be calculated by aggregating the number of events in the events table by week and user. The user growth for the product can be calculated by counting the number of unique users in the users table over time. The weekly Operation Analytics and Investigating Metric Spike retention of users-sign up cohort can be calculated by dividing the number of users who return after signing up by the number of users who signed up in a given week. The weekly engagement per device can be calculated by aggregating the number of events in the events table by week, device, and user. The email engagement metrics can be calculated by aggregating the number of email events in the email_events table by week and user.

Tech-Stack Used

I have used MYSQL Workbench 8.0 CE.

Insights

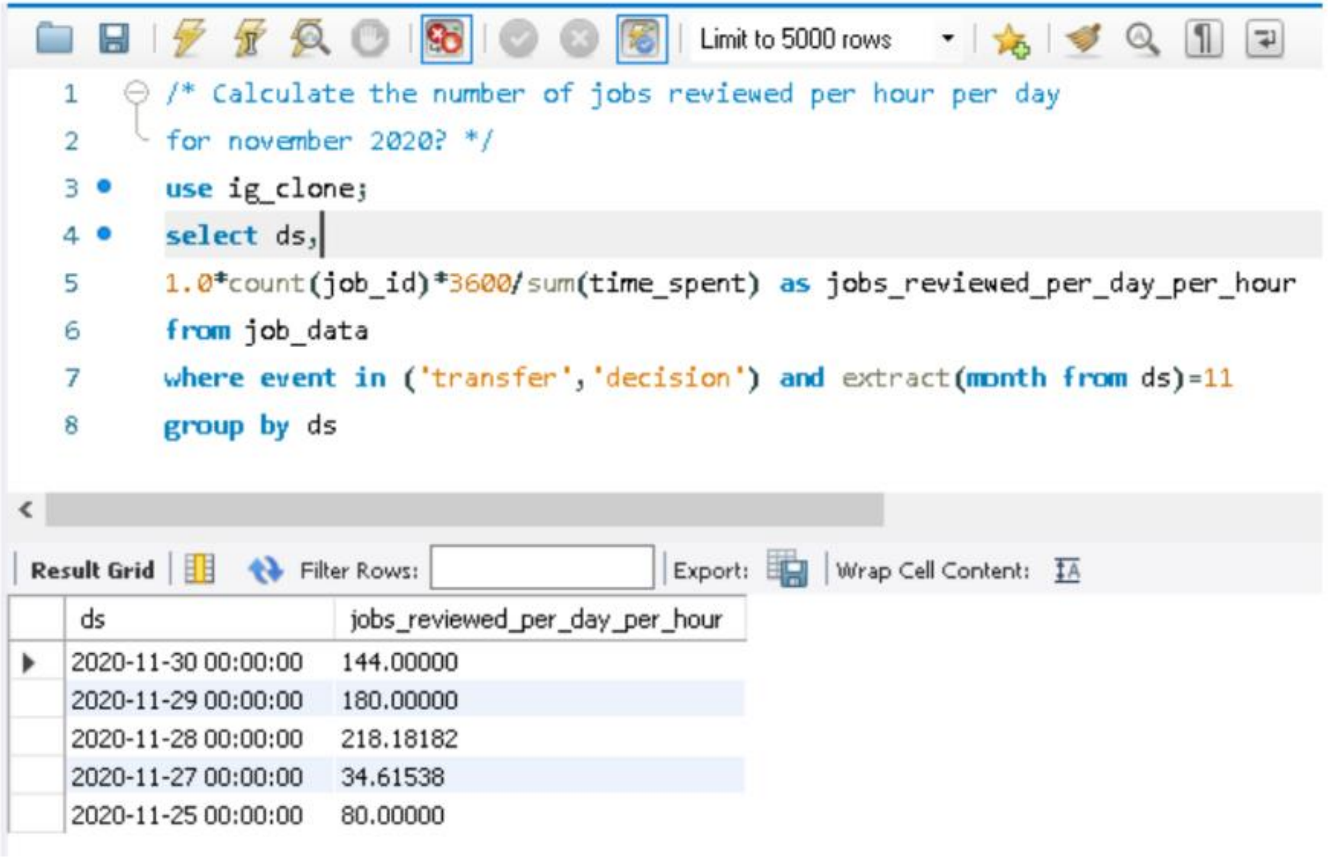
The project is extremely helpful to understand the basics of MySQL. It helped me to learn the structure. It also helped me to learn new keywords like week, day, etc. I have also learned the concept of BETWEEN, GROUP BY, ORDER BY, CASE, Window function, partition by, over, rows between, etc. This project gave me the confidence to work in SQL.

Also, I have learned to import CSV files in the MYSQL workbench. But the files consist of a high number of rows which took a lot of time.

Results

Case Study 1 (Job Data):

A. Number of jobs reviewed: Amount of jobs reviewed over time.

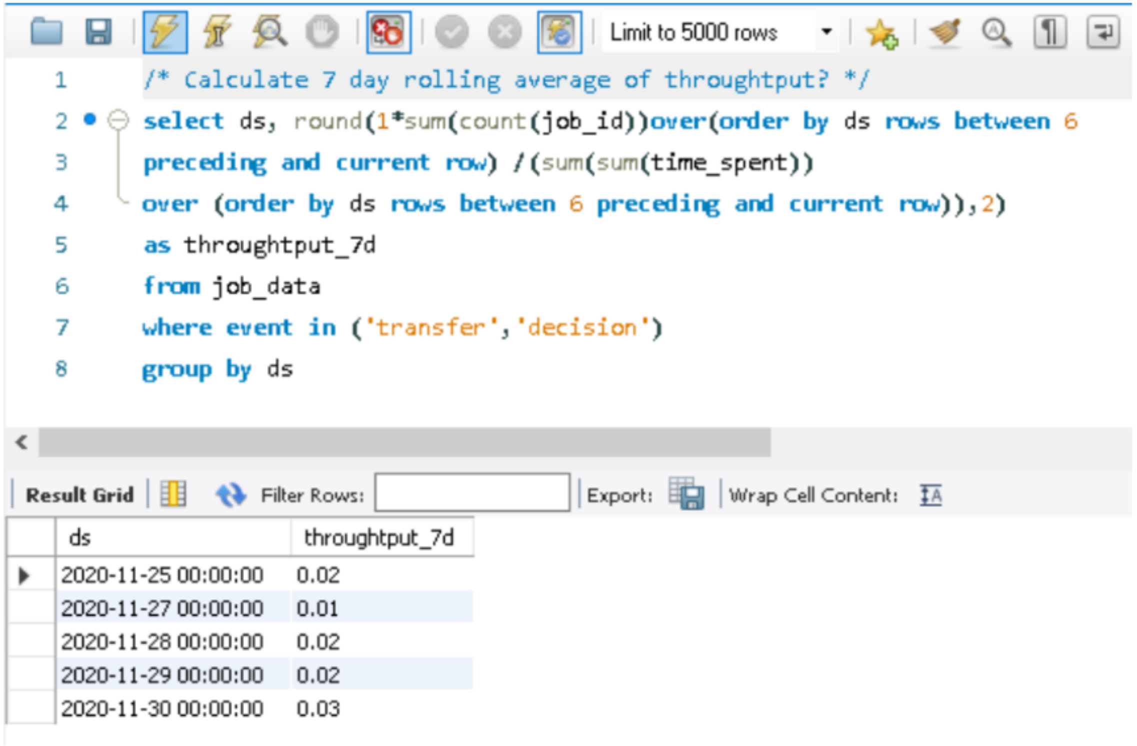


```sql
/* Calculate the number of jobs reviewed per hour per day
for november 2020? */
use ig_clone;
select ds,
1.0*count(job_id)*3600/sum(time_spent) as jobs_reviewed_per_day_per_hour
from job_data
where event in ('transfer','decision') and extract(month from ds)=11
group by ds
```

ds	jobs_reviewed_per_day_per_hour
2020-11-30 00:00:00	144.00000
2020-11-29 00:00:00	180.00000
2020-11-28 00:00:00	218.18182
2020-11-27 00:00:00	34.61538
2020-11-25 00:00:00	80.00000

B. Throughput Analysis:

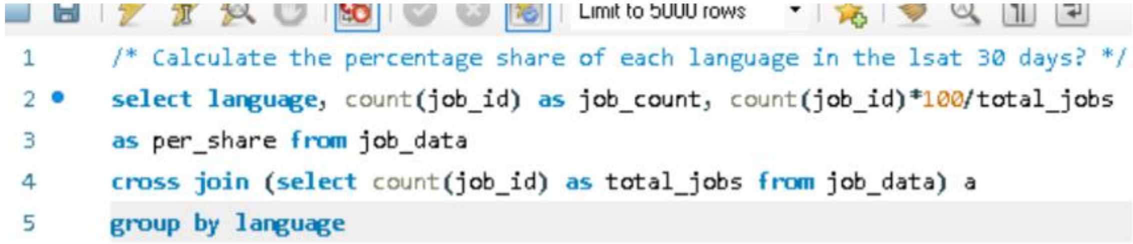
The choice between a daily metric and a 7-day rolling metric for measuring throughput depends on the specific context and the nature of the data being analyzed. If the density of the data is larger, we use the daily metric, and if the density is low the 7-day rolling works well. It also depends upon the anomaly in the data set. Because in 7 days metric the view is broader similar to the daily metric view becomes narrower.



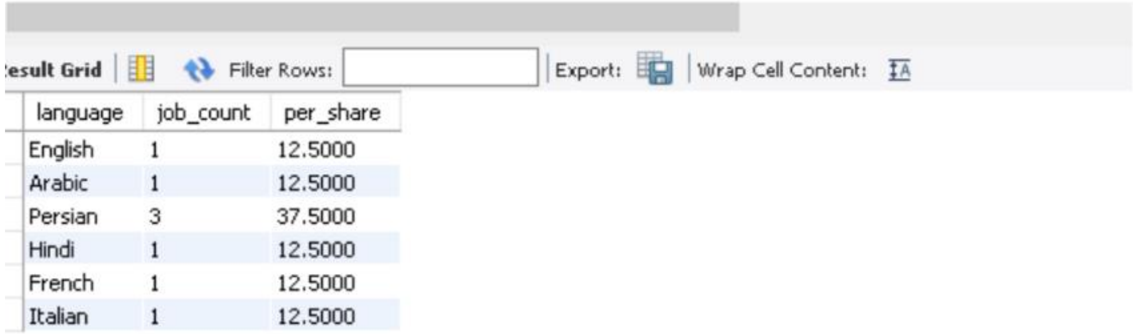
```
/* Calculate 7 day rolling average of throughput? */
select ds, round(1*sum(count(job_id))over(order by ds rows between 6
preceding and current row) /(sum(sum(time_spent))
over (order by ds rows between 6 preceding and current row)),2)
as throughtput_7d
from job_data
where event in ('transfer','decision')
group by ds
```

ds	throughtput_7d
2020-11-25 00:00:00	0.02
2020-11-27 00:00:00	0.01
2020-11-28 00:00:00	0.02
2020-11-29 00:00:00	0.02
2020-11-30 00:00:00	0.03

C. Language Share Analysis

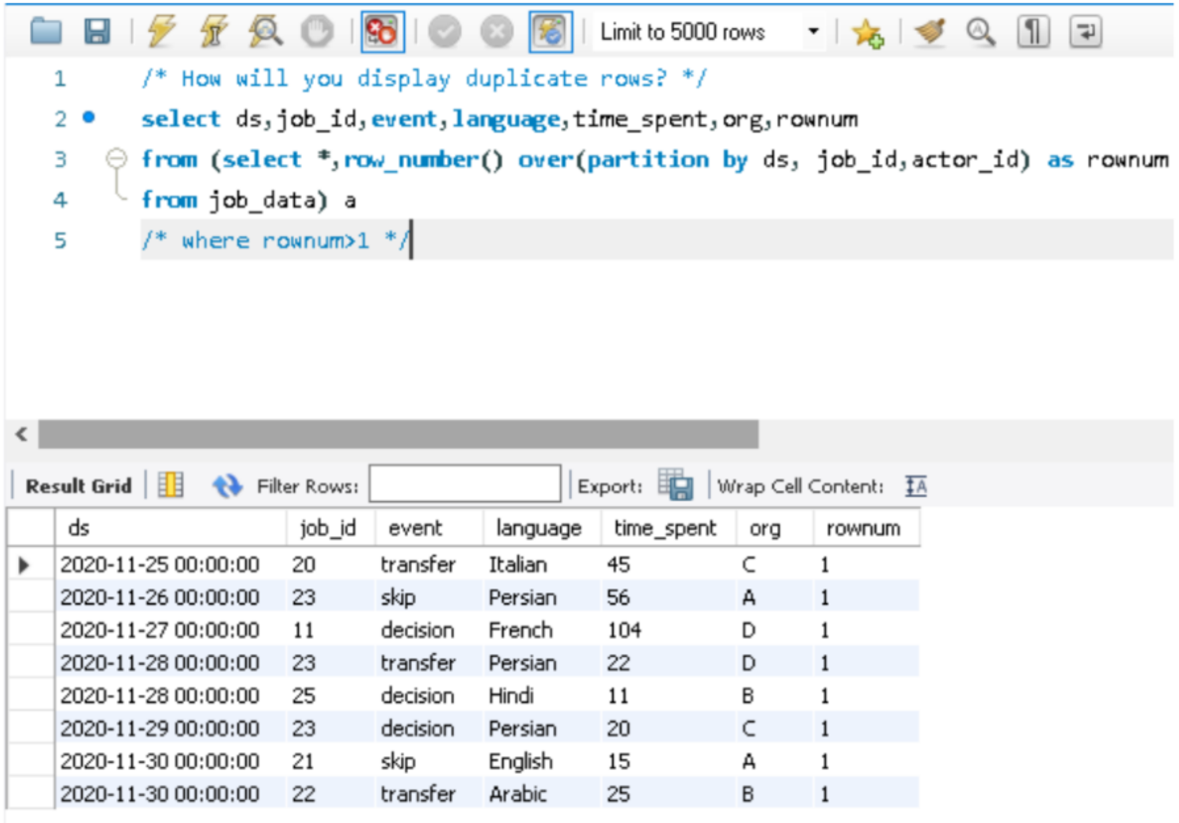


```sql
/* Calculate the percentage share of each language in the lsat 30 days? */
select language, count(job_id) as job_count, count(job_id)*100/total_jobs
as per_share from job_data
cross join (select count(job_id) as total_jobs from job_data) a
group by language
```



language	job_count	per_share
English	1	12.5000
Arabic	1	12.5000
Persian	3	37.5000
Hindi	1	12.5000
French	1	12.5000
Italian	1	12.5000

D. Duplicate Rows Detection:

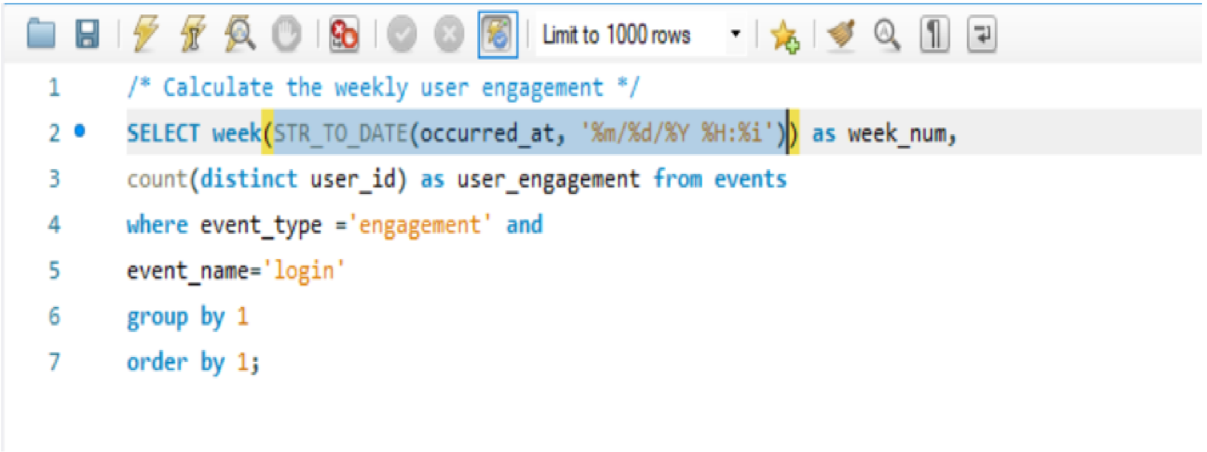


```
/* How will you display duplicate rows? */
select ds,job_id,event,language,time_spent,org,rownum
from (select *,row_number() over(partition by ds, job_id,actor_id) as rownum
from job_data) a
/* where rownum>1 */
```

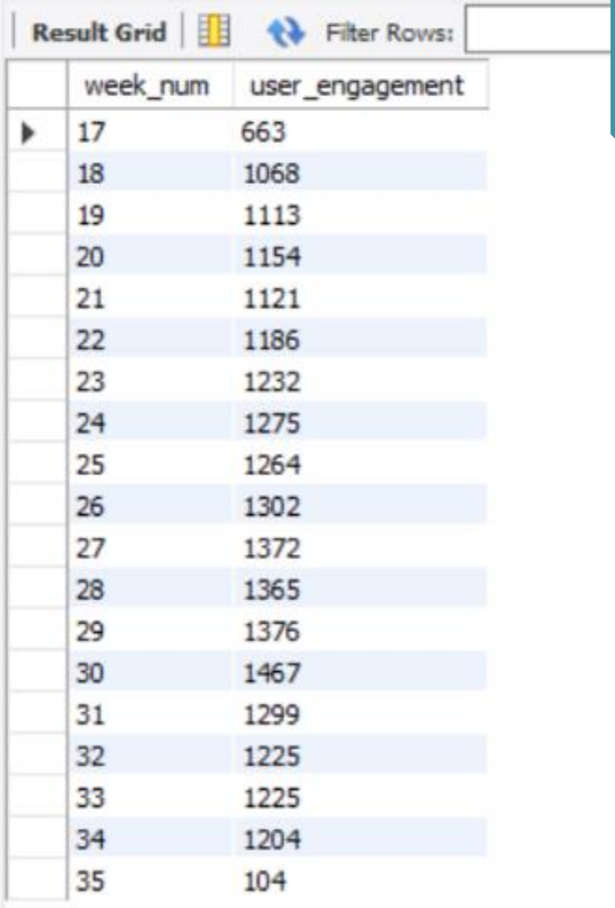
ds	job_id	event	language	time_spent	org	rownum
2020-11-25 00:00:00	20	transfer	Italian	45	C	1
2020-11-26 00:00:00	23	skip	Persian	56	A	1
2020-11-27 00:00:00	11	decision	French	104	D	1
2020-11-28 00:00:00	23	transfer	Persian	22	D	1
2020-11-28 00:00:00	25	decision	Hindi	11	B	1
2020-11-29 00:00:00	23	decision	Persian	20	C	1
2020-11-30 00:00:00	21	skip	English	15	A	1
2020-11-30 00:00:00	22	transfer	Arabic	25	B	1

Case Study 2: Investigating Metric Spike

A. Weekly User Engagement

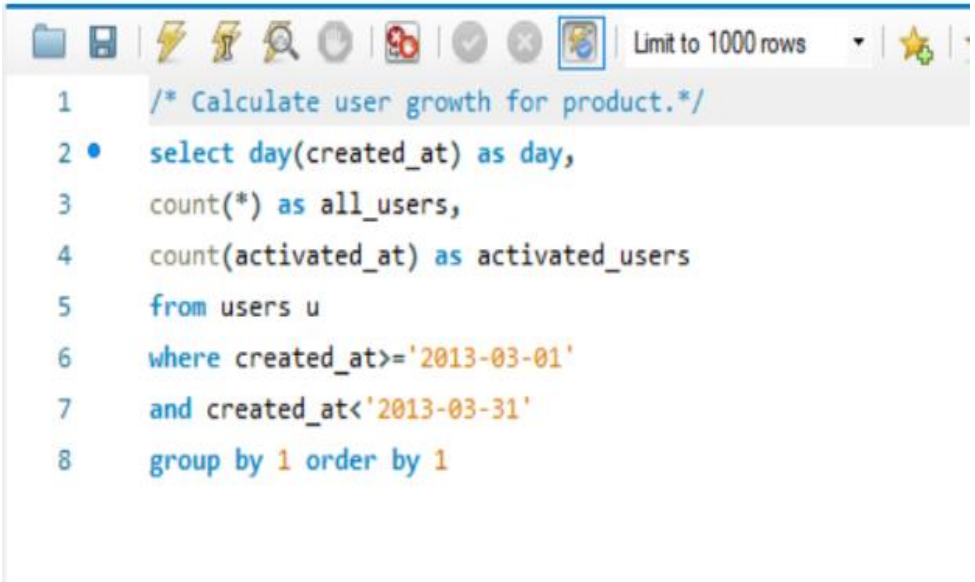


```
/* Calculate the weekly user engagement */
SELECT week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i')) as week_num,
count(distinct user_id) as user_engagement from events
where event_type ='engagement' and
event_name='login'
group by 1
order by 1;
```

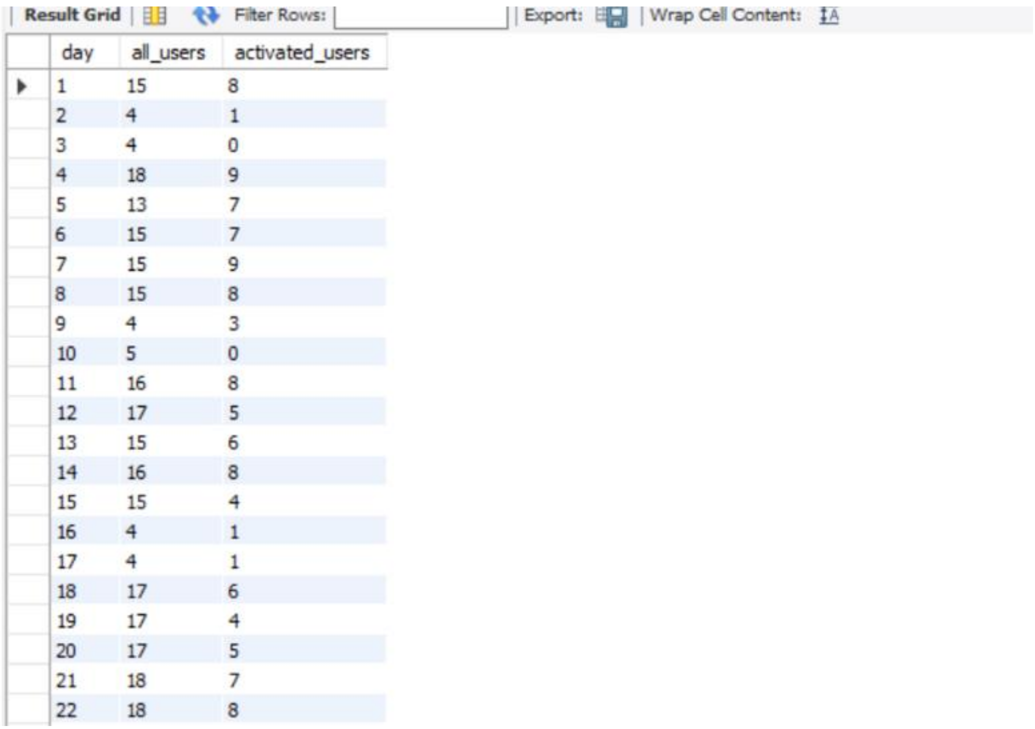


week_num	user_engagement
17	663
18	1068
19	1113
20	1154
21	1121
22	1186
23	1232
24	1275
25	1264
26	1302
27	1372
28	1365
29	1376
30	1467
31	1299
32	1225
33	1225
34	1204
35	104

B. User Growth Analysis:

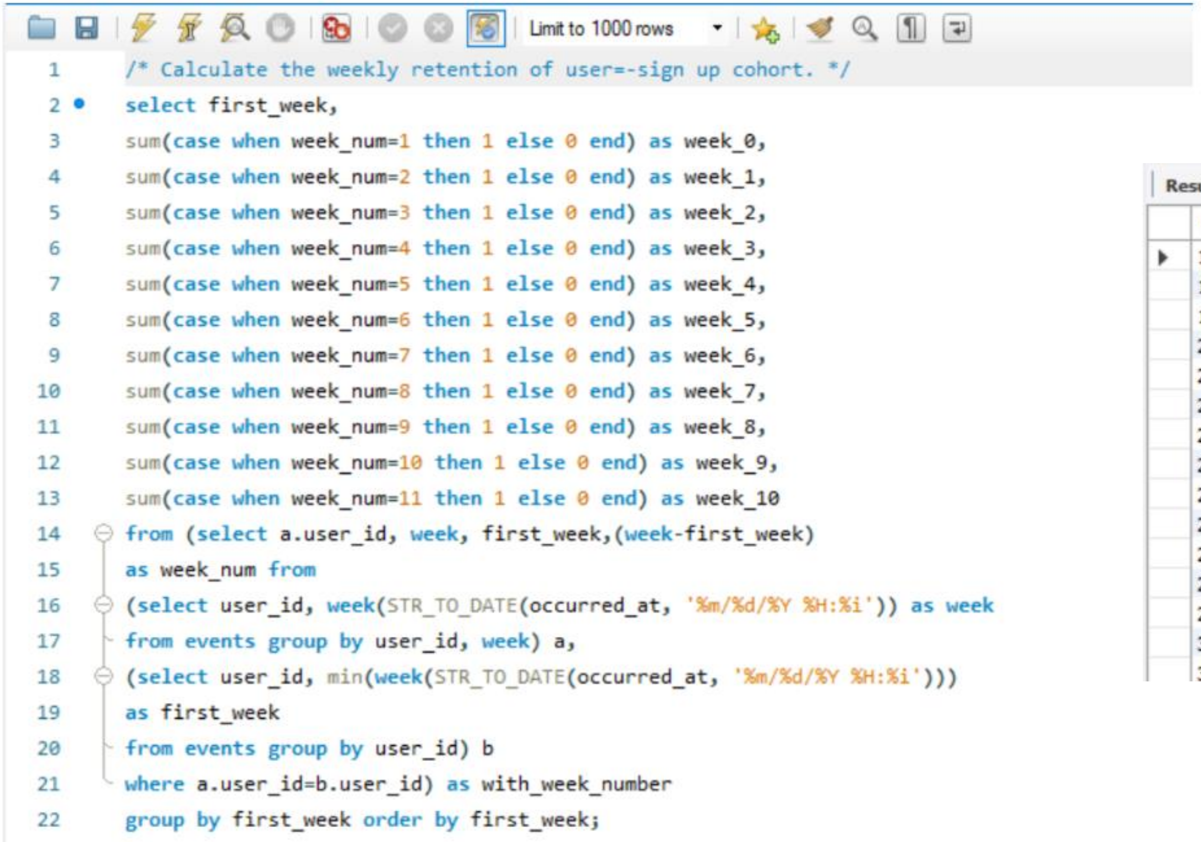


```
/* Calculate user growth for product.*/
select day(created_at) as day,
count(*) as all_users,
count(activated_at) as activated_users
from users u
where created_at>='2013-03-01'
and created_at<'2013-03-31'
group by 1 order by 1
```



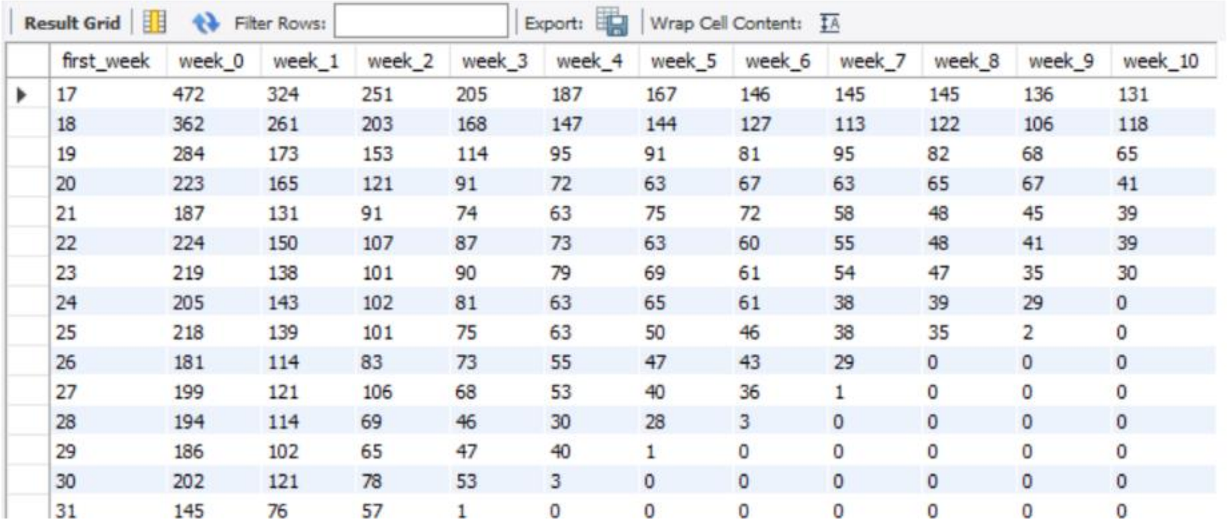
day	all_users	activated_users
1	15	8
2	4	1
3	4	0
4	18	9
5	13	7
6	15	7
7	15	9
8	15	8
9	4	3
10	5	0
11	16	8
12	17	5
13	15	6
14	16	8
15	15	4
16	4	1
17	4	1
18	17	6
19	17	4
20	17	5
21	18	7
22	18	8

C. Weekly Retention Analysis:



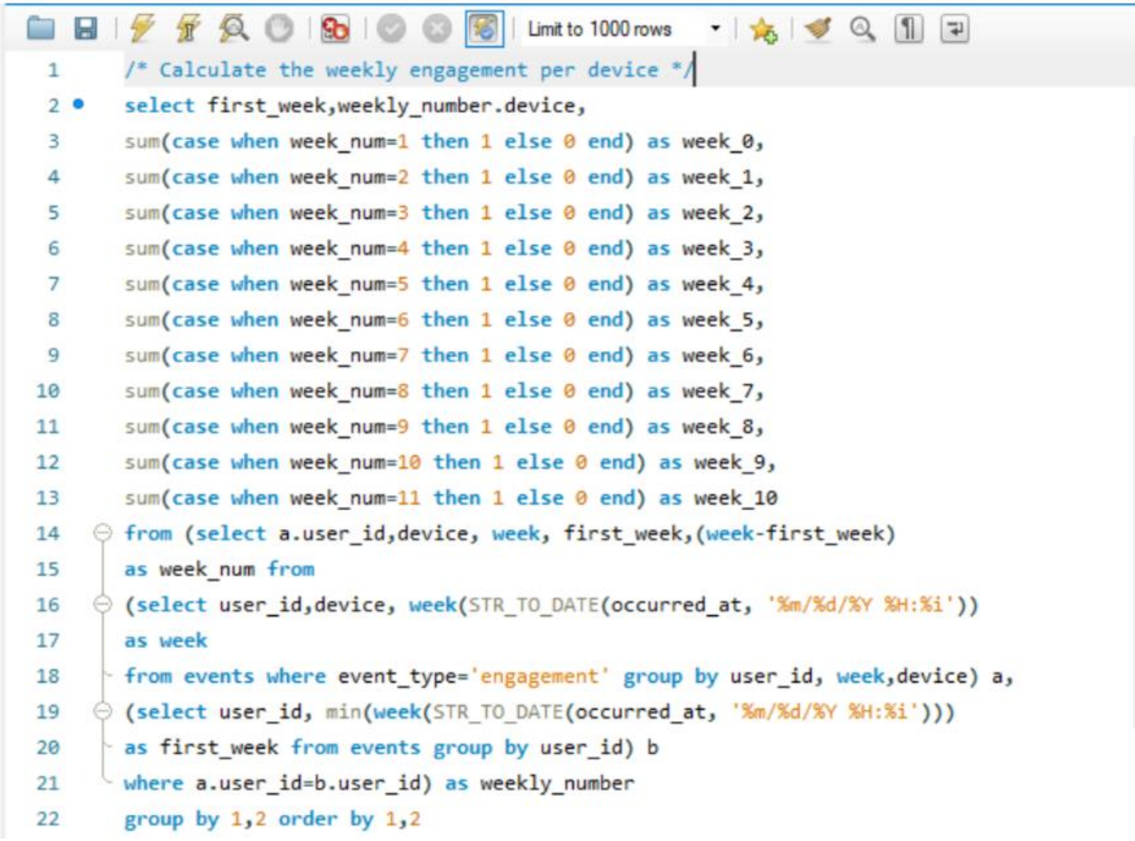
Limit to 1000 rows

```sql
/* Calculate the weekly retention of user=-sign up cohort. */
select first_week,
sum(case when week_num=1 then 1 else 0 end) as week_0,
sum(case when week_num=2 then 1 else 0 end) as week_1,
sum(case when week_num=3 then 1 else 0 end) as week_2,
sum(case when week_num=4 then 1 else 0 end) as week_3,
sum(case when week_num=5 then 1 else 0 end) as week_4,
sum(case when week_num=6 then 1 else 0 end) as week_5,
sum(case when week_num=7 then 1 else 0 end) as week_6,
sum(case when week_num=8 then 1 else 0 end) as week_7,
sum(case when week_num=9 then 1 else 0 end) as week_8,
sum(case when week_num=10 then 1 else 0 end) as week_9,
sum(case when week_num=11 then 1 else 0 end) as week_10
from (select a.user_id, week, first_week,(week-first_week)
as week_num from
(select user_id, week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i')) as week
from events group by user_id, week) a,
(select user_id, min(week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i')))
as first_week
from events group by user_id) b
where a.user_id=b.user_id) as with_week_number
group by first_week order by first_week;
```

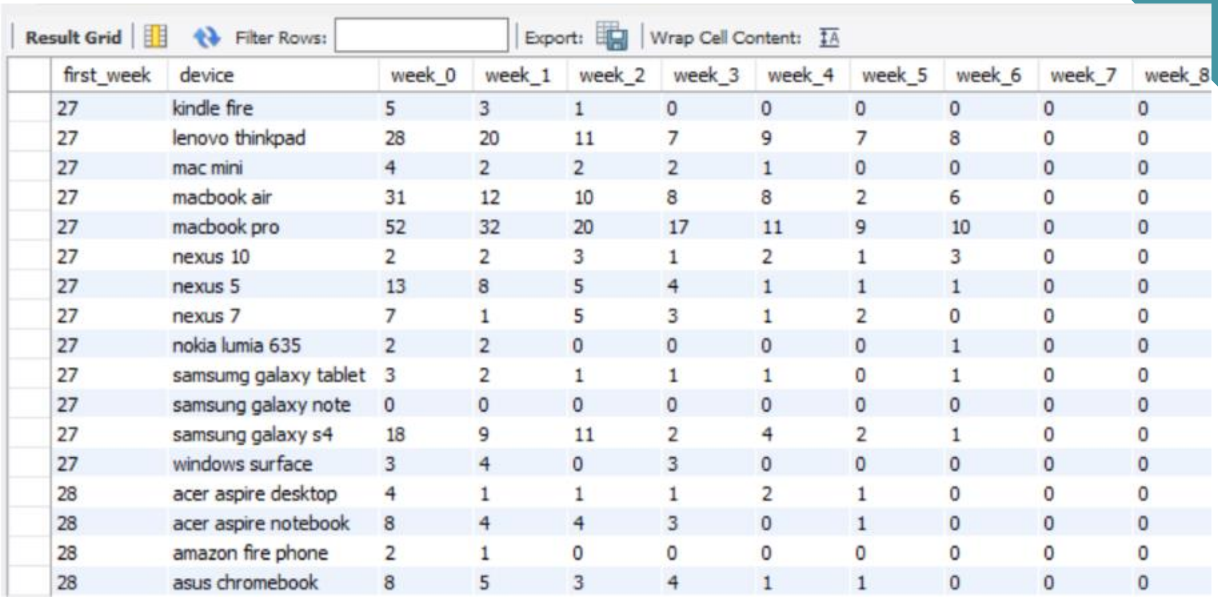


first_week	week_0	week_1	week_2	week_3	week_4	week_5	week_6	week_7	week_8	week_9	week_10
17	472	324	251	205	187	167	146	145	145	136	131
18	362	261	203	168	147	144	127	113	122	106	118
19	284	173	153	114	95	91	81	95	82	68	65
20	223	165	121	91	72	63	67	63	65	67	41
21	187	131	91	74	63	75	72	58	48	45	39
22	224	150	107	87	73	63	60	55	48	41	39
23	219	138	101	90	79	69	61	54	47	35	30
24	205	143	102	81	63	65	61	38	39	29	0
25	218	139	101	75	63	50	46	38	35	2	0
26	181	114	83	73	55	47	43	29	0	0	0
27	199	121	106	68	53	40	36	1	0	0	0
28	194	114	69	46	30	28	3	0	0	0	0
29	186	102	65	47	40	1	0	0	0	0	0
30	202	121	78	53	3	0	0	0	0	0	0
31	145	76	57	1	0	0	0	0	0	0	0

D. Weekly Engagement Per Device:

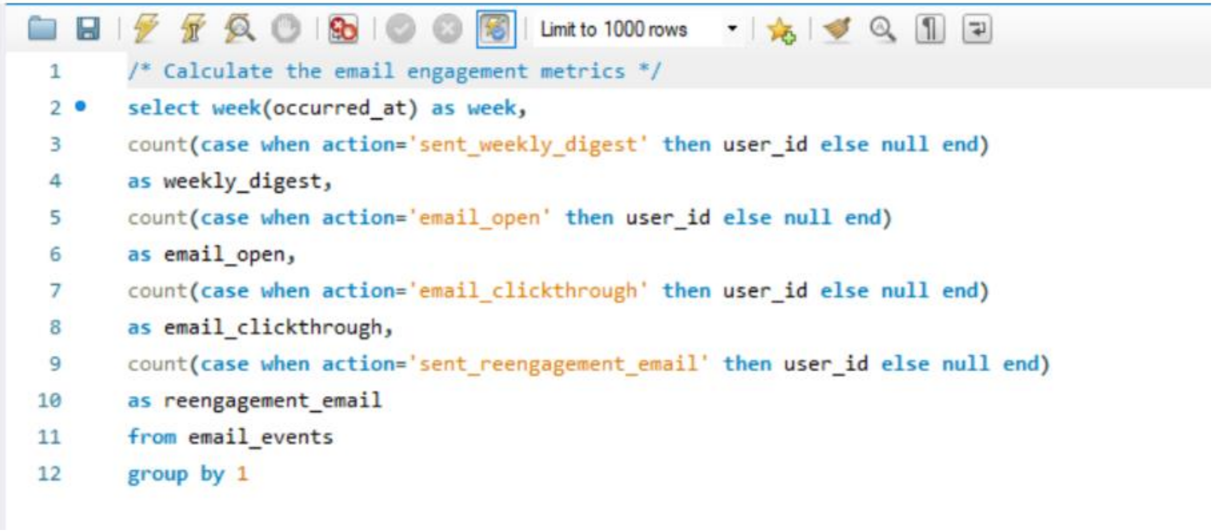


```sql
/* Calculate the weekly engagement per device */
select first_week,weekly_number.device,
sum(case when week_num=1 then 1 else 0 end) as week_0,
sum(case when week_num=2 then 1 else 0 end) as week_1,
sum(case when week_num=3 then 1 else 0 end) as week_2,
sum(case when week_num=4 then 1 else 0 end) as week_3,
sum(case when week_num=5 then 1 else 0 end) as week_4,
sum(case when week_num=6 then 1 else 0 end) as week_5,
sum(case when week_num=7 then 1 else 0 end) as week_6,
sum(case when week_num=8 then 1 else 0 end) as week_7,
sum(case when week_num=9 then 1 else 0 end) as week_8,
sum(case when week_num=10 then 1 else 0 end) as week_9,
sum(case when week_num=11 then 1 else 0 end) as week_10
from (select a.user_id,device, week, first_week,(week-first_week)
as week_num from
(select user_id,device, week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i'))
as week
from events where event_type='engagement' group by user_id, week,device) a,
(select user_id, min(week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i')))
as first_week from events group by user_id) b
where a.user_id=b.user_id) as weekly_number
group by 1,2 order by 1,2
```

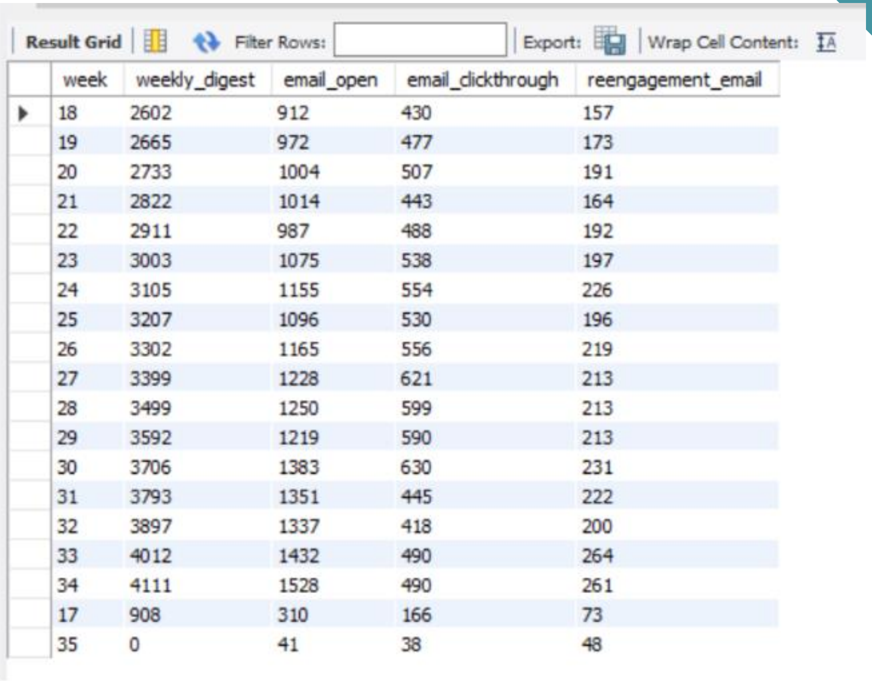


first_week	device	week_0	week_1	week_2	week_3	week_4	week_5	week_6	week_7	week_8
27	kindle fire	5	3	1	0	0	0	0	0	0
27	lenovo thinkpad	28	20	11	7	9	7	8	0	0
27	mac mini	4	2	2	2	1	0	0	0	0
27	macbook air	31	12	10	8	8	2	6	0	0
27	macbook pro	52	32	20	17	11	9	10	0	0
27	nexus 10	2	2	3	1	2	1	3	0	0
27	nexus 5	13	8	5	4	1	1	1	0	0
27	nexus 7	7	1	5	3	1	2	0	0	0
27	nokia lumia 635	2	2	0	0	0	0	1	0	0
27	samsumg galaxy tablet	3	2	1	1	1	0	1	0	0
27	samsung galaxy note	0	0	0	0	0	0	0	0	0
27	samsung galaxy s4	18	9	11	2	4	2	1	0	0
27	windows surface	3	4	0	3	0	0	0	0	0
28	acer aspire desktop	4	1	1	1	2	1	0	0	0
28	acer aspire notebook	8	4	4	3	0	1	0	0	0
28	amazon fire phone	2	1	0	0	0	0	0	0	0
28	asus chromebook	8	5	3	4	1	1	0	0	0

E. Email Engagement Analysis:



```sql
/* Calculate the email engagement metrics */
select week(occurred_at) as week,
count(case when action='sent_weekly_digest' then user_id else null end)
as weekly_digest,
count(case when action='email_open' then user_id else null end)
as email_open,
count(case when action='email_clickthrough' then user_id else null end)
as email_clickthrough,
count(case when action='sent_reengagement_email' then user_id else null end)
as reengagement_email
from email_events
group by 1
```



week	weekly_digest	email_open	email_clickthrough	reengagement_email
18	2602	912	430	157
19	2665	972	477	173
20	2733	1004	507	191
21	2822	1014	443	164
22	2911	987	488	192
23	3003	1075	538	197
24	3105	1155	554	226
25	3207	1096	530	196
26	3302	1165	556	219
27	3399	1228	621	213
28	3499	1250	599	213
29	3592	1219	590	213
30	3706	1383	630	231
31	3793	1351	445	222
32	3897	1337	418	200
33	4012	1432	490	264
34	4111	1528	490	261
17	908	310	166	73
35	0	41	38	48

Through this project, I was able to understand how important **Operational Analytics** is for an organizations as it helps in identifying and understanding areas where **improvement** is required.
In this project I was able to get insights about various questions like rate of job reviews, share of languages, patterns of user engagement on weekly basis, growth of users etc. which can be **communicated** to the management team as per the requirements using which they can make proper **data-driven decisions.**

Thank you

JONNALAGADDA SUSRITHA

susrithaj2102@gmail.com