PROJECT-3

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 understand cross-functional teams, and more effective workflows.

Investigating metric spikes is also an important part of operation analytics as being a Data Analyst we 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 it's very important to investigate metric spikes.

The things that we are going to find out through the projects are:

- Number of jobs reviewed
- Throughput
- Percentage share of each language
- Duplicate rows
- User Engagement
- User Growth
- Weekly Retention
- Weekly Engagement
- Email Engagement

APPROACH

Firstly, I spent some time on understanding the data/table given. I cleared the questions which were in my mind like what does the job_id, actor_id, event means and what are the things to consider while reviewing the data. I use SQL to derive different insights from the dataset provided by the management team. I first created a database "operation_analytics" and then the tables using the structure and links provided by the team. Then, we performed analysis to generate valuable insights for the company.

Tach stack Used :

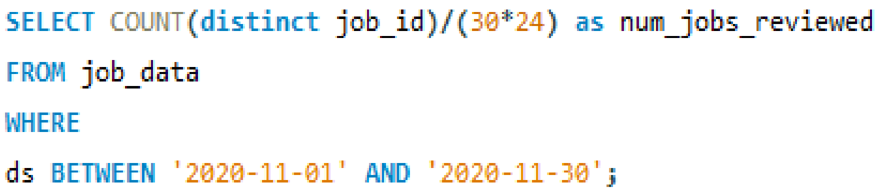
1. MySQL Workbench (Version 8.0.36) for working ,analysing and reporting insights.
2. Microsoft Word (for presenting the detailed analysis Report)

Insights :

<u>Case Study 1 (Job Data):</u>

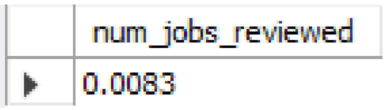
A. Number of jobs reviewed: Amount of jobs reviewed over time.
My task: Calculate the number of jobs reviewed per hour per day for November 2020?

<u>Query:</u>



```
SELECT COUNT(distinct job_id)/(30*24) as num_jobs_reviewed
FROM job_data
WHERE
ds BETWEEN '2020-11-01' AND '2020-11-30';
```

<u>RESULT:</u>

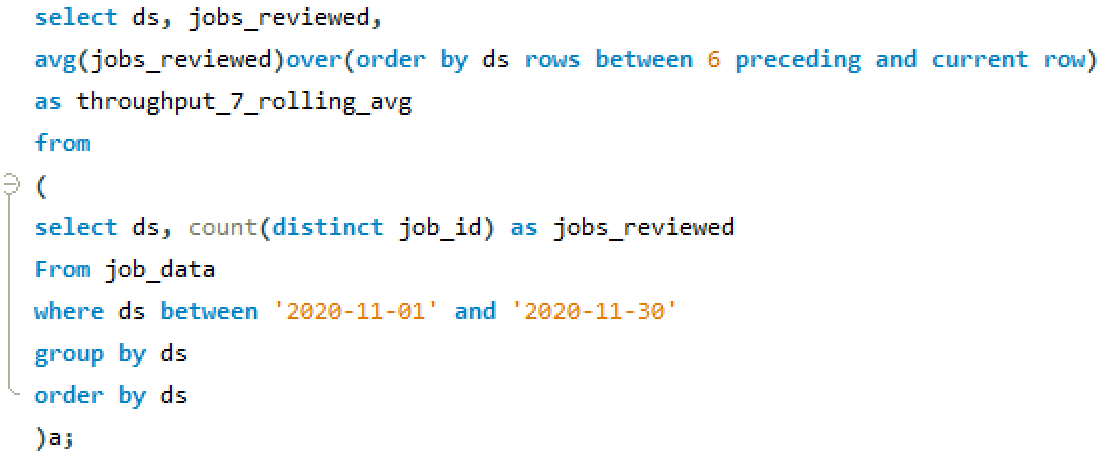


	num_jobs_reviewed
▶	0.0083

Less than 0.01 jobs were reviewed each hour of the day throughout the month of November.

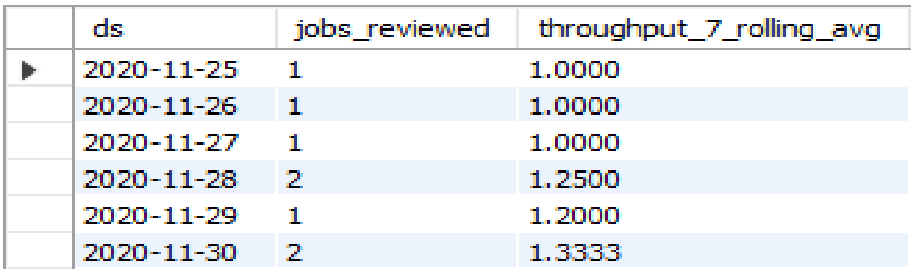
B. **Throughput:** It is the no. of events happening per second.
My task: Let's say the above metric is called throughput. Calculate 7 day rolling average of thoroughput. For throughput, do you prefer daily metric or 7-day rolling and why?

Query:



```
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;
```

RESULT:

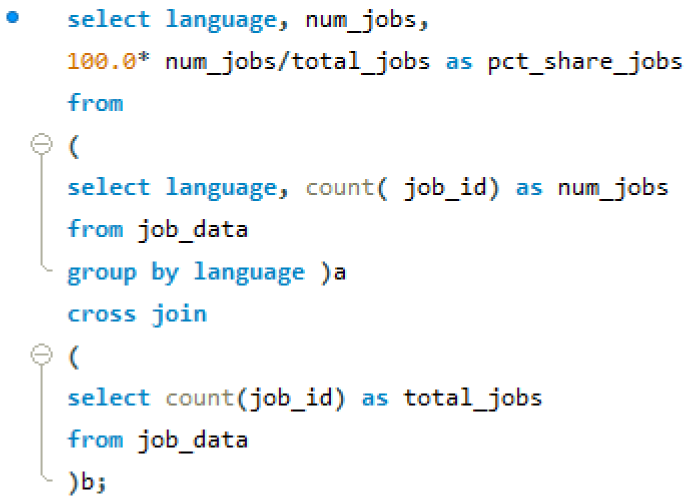


	ds	jobs_reviewed	throughput_7_rolling_avg
▶	2020-11-25	1	1.0000
	2020-11-26	1	1.0000
	2020-11-27	1	1.0000
	2020-11-28	2	1.2500
	2020-11-29	1	1.2000
	2020-11-30	2	1.3333

Using a 7-day rolling average for throughput can be helpful in understanding trends over time, as it provides a longer- term perspective compared to a daily metric. This can help to smooth out any short-term fluctuations in the data and provide a clear picture of the overall trend.

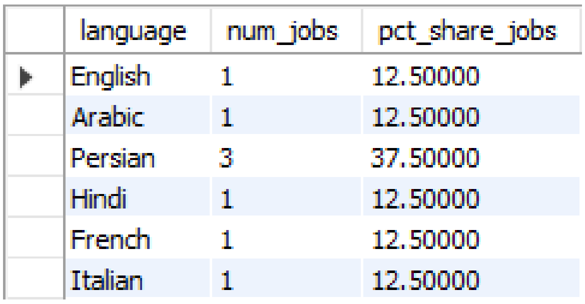
C. **Percentage share of each language:** Share of each language for different contents.
My task: Calculate the percentage share of each language in the last 30 days?

Query:



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

RESULT:



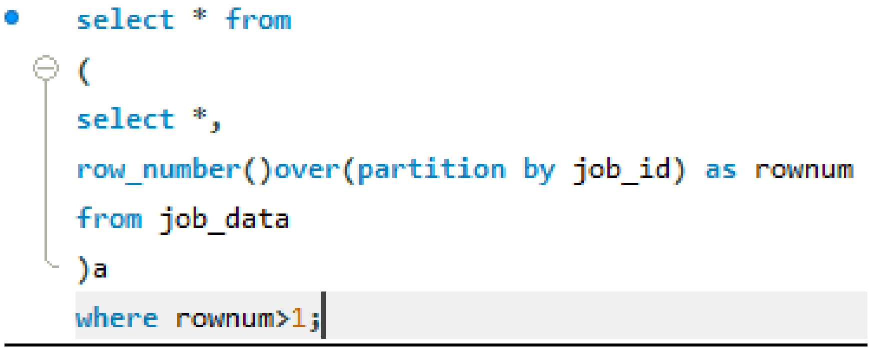
	language	num_jobs	pct_share_jobs
▶	English	1	12.50000
	Arabic	1	12.50000
	Persian	3	37.50000
	Hindi	1	12.50000
	French	1	12.50000
	Italian	1	12.50000

Persian Language had the highest share among other languages.

D. **Duplicate rows:** Rows that have the same value present in them.

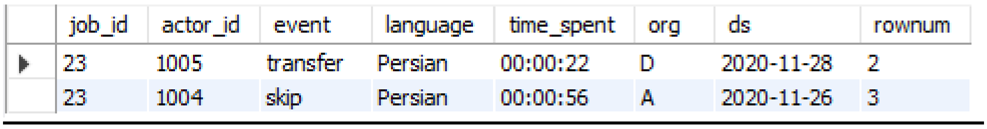
My task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

Query:



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

RESULT:



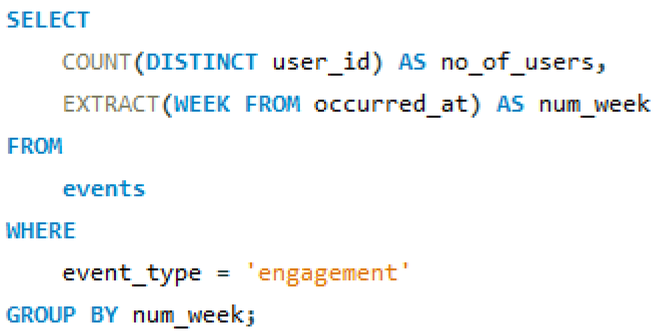
job_id	actor_id	event	language	time_spent	org	ds	rownum
23	1005	transfer	Persian	00:00:22	D	2020-11-28	2
23	1004	skip	Persian	00:00:56	A	2020-11-26	3

The output showed two records as there were two duplicate job id in the dataset.

Case Study 2 (Investigating metric spike):

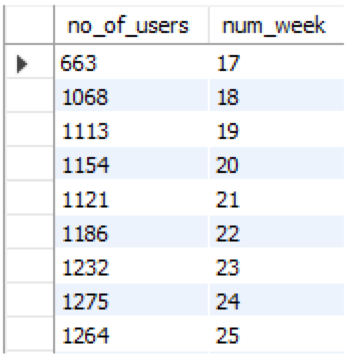
A. **User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.
My task: Calculate the weekly user engagement?

Query:

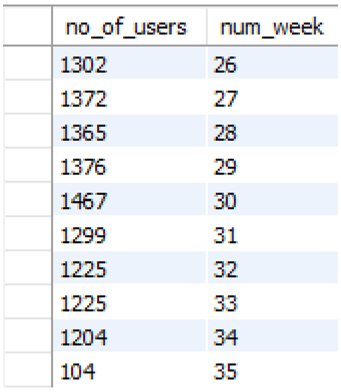


```
SELECT
    COUNT(DISTINCT user_id) AS no_of_users,
    EXTRACT(WEEK FROM occurred_at) AS num_week
FROM
    events
WHERE
    event_type = 'engagement'
GROUP BY num_week;
```

RESULT:



	no_of_users	num_week
▶	663	17
	1068	18
	1113	19
	1154	20
	1121	21
	1186	22
	1232	23
	1275	24
	1264	25



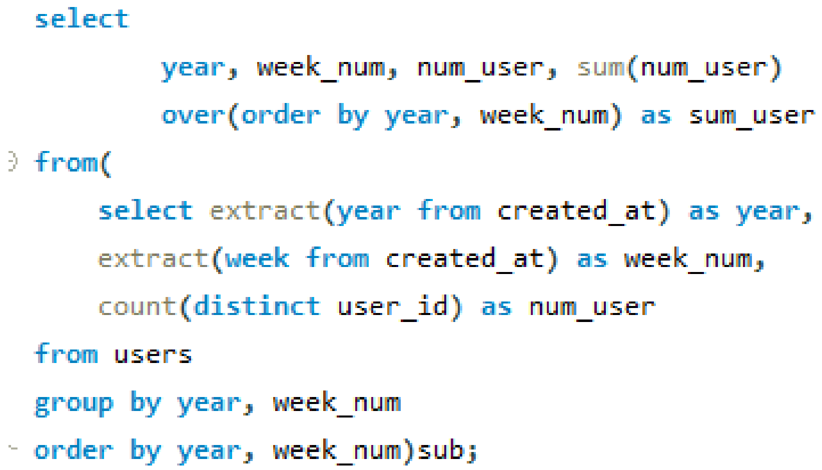
	no_of_users	num_week
	1302	26
	1372	27
	1365	28
	1376	29
	1467	30
	1299	31
	1225	32
	1225	33
	1204	34
	104	35

Week 30 posted the highest user engagement and week 17 posted the minimum user engagement

B. User Growth: Amount of users growing over time for a product.

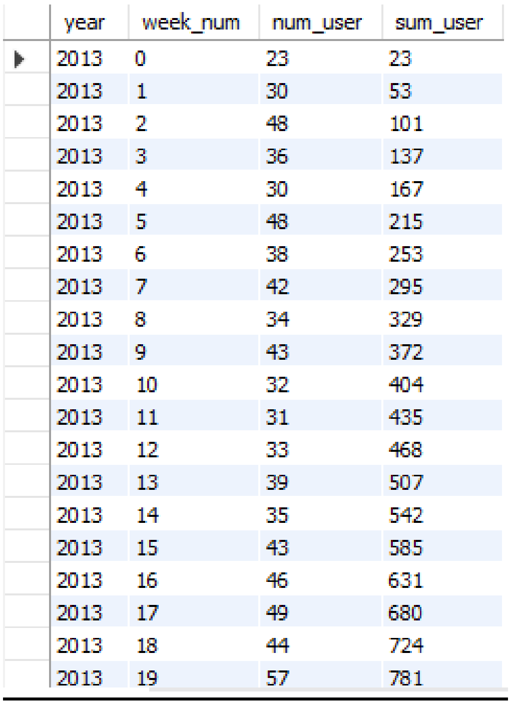
My task: Calculate the user growth for the product?

Query:



```
select
        year, week_num, num_user, sum(num_user)
        over(order by year, week_num) as sum_user
from(
    select extract(year from created_at) as year,
    extract(week from created_at) as week_num,
    count(distinct user_id) as num_user
from users
group by year, week_num
order by year, week_num)sub;
```

RESULT:

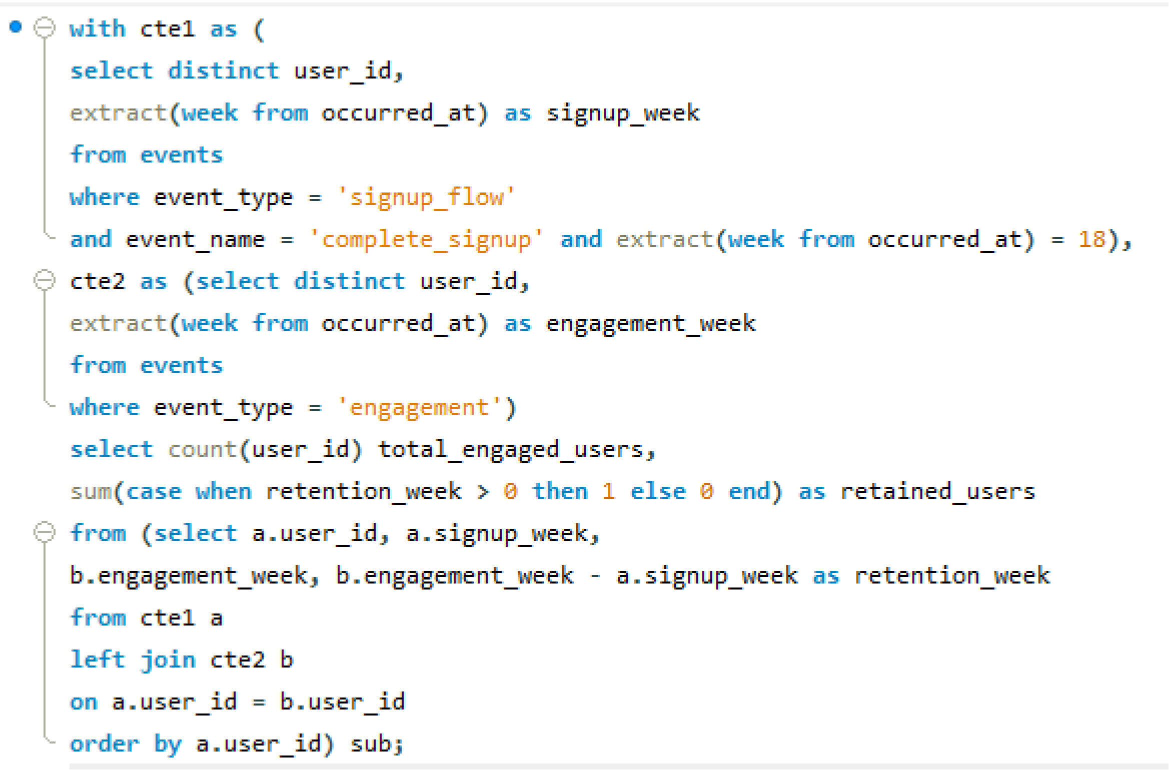


year	week_num	num_user	sum_user
2013	0	23	23
2013	1	30	53
2013	2	48	101
2013	3	36	137
2013	4	30	167
2013	5	48	215
2013	6	38	253
2013	7	42	295
2013	8	34	329
2013	9	43	372
2013	10	32	404
2013	11	31	435
2013	12	33	468
2013	13	39	507
2013	14	35	542
2013	15	43	585
2013	16	46	631
2013	17	49	680
2013	18	44	724
2013	19	57	781

The 33th week of 2014 saw the greatest number of users actively engaging with the product or service, while the 35th week of 2014 had the lowest number of active users.

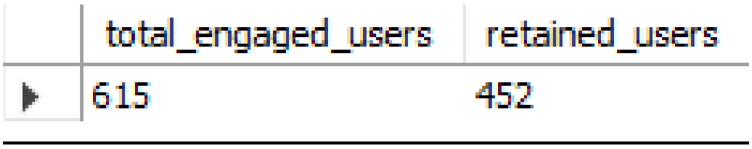
C. **Weekly Retention:** Users getting retained weekly after signing-up for a product.
My task: Calculate the weekly retention of users-sign up cohort?

Query:



```
with cte1 as (
select distinct user_id,
extract(week from occurred_at) as signup_week
from events
where event_type = 'signup_flow'
and event_name = 'complete_signup' and extract(week from occurred_at) = 18),
cte2 as (select distinct user_id,
extract(week from occurred_at) as engagement_week
from events
where event_type = 'engagement')
select count(user_id) total_engaged_users,
sum(case when retention_week > 0 then 1 else 0 end) as retained_users
from (select a.user_id, a.signup_week,
b.engagement_week, b.engagement_week - a.signup_week as retention_week
from cte1 a
left join cte2 b
on a.user_id = b.user_id
order by a.user_id) sub;
```

RESULT:

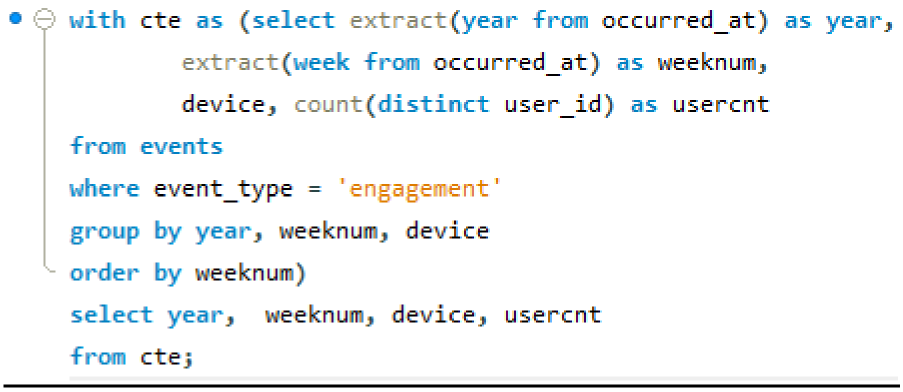


total_engaged_users	retained_users
615	452

D. **Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.

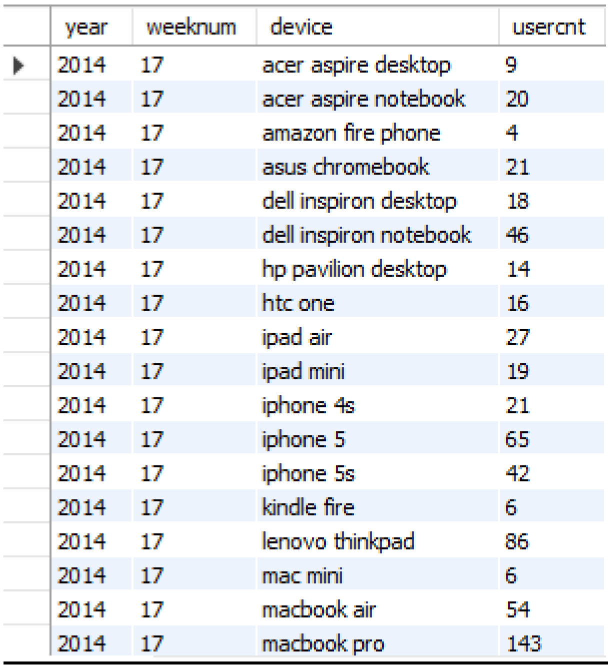
My task: Calculate the weekly engagement per device?

Query:



```
with cte as (select extract(year from occurred_at) as year,
        extract(week from occurred_at) as weeknum,
        device, count(distinct user_id) as usercnt
from events
where event_type = 'engagement'
group by year, weeknum, device
order by weeknum)
select year,  weeknum, device, usercnt
from cte;
```

RESULT:

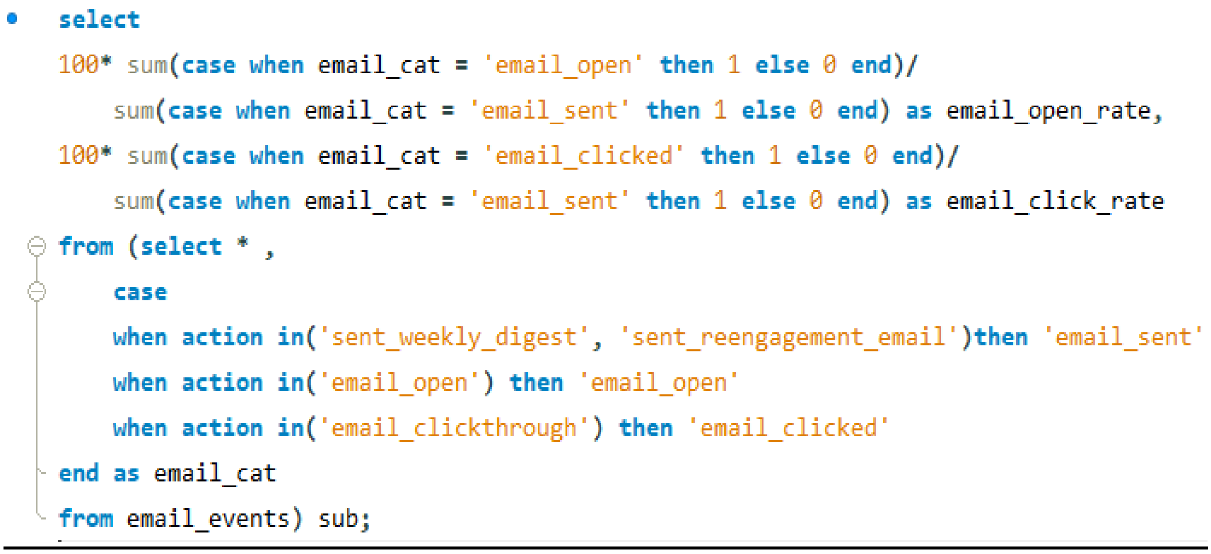


year	weeknum	device	usercnt
2014	17	acer aspire desktop	9
2014	17	acer aspire notebook	20
2014	17	amazon fire phone	4
2014	17	asus chromebook	21
2014	17	dell inspiron desktop	18
2014	17	dell inspiron notebook	46
2014	17	hp pavilion desktop	14
2014	17	htc one	16
2014	17	ipad air	27
2014	17	ipad mini	19
2014	17	iphone 4s	21
2014	17	iphone 5	65
2014	17	iphone 5s	42
2014	17	kindle fire	6
2014	17	lenovo thinkpad	86
2014	17	mac mini	6
2014	17	macbook air	54
2014	17	macbook pro	143

Week30 of the year 2014 had the highest user engagement of 322 users for the product and device being usedwas ‘Macbook Pro’.

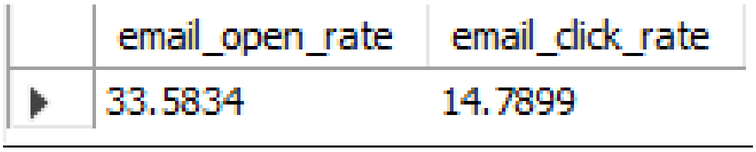
E. **Email Engagement:** Users engaging with the email service.
My task: Calculate the email engagement metrics?

Query:



```
select
100* 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* 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_click_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 email_events) sub;
```

RESULT:



	email_open_rate	email_click_rate
▶	33.5834	14.7899

Out of the total emails sent around 34% of them were opened and only 15% of those emails were clicked.

OVERALL RESULT

In this project, I learned how to apply advanced SQL concepts like Windows Functions, etc. I understood how the real-world industry works. It helped me in mastering my SQL concepts. I learned how to ask the right questions given the circumstances. From the given data and questions, which columns to consider and how to find the valuable insights which help the business to grow. I learned how the company finds different areas related to the company to improve it further. I got to know about investigating metric spikes (why there is a boom and why there is a dip)

Drive Link:

https://drive.google.com/drive/folders/1FgN8iw8N0MxK8wdh9-6KztYqjQGoA_MK?usp=sharing