PROJECT –2

OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

NAME: BHAVYA SRI DUGGINA

PROJECT DESCRIPTION

- This project focuses on leveraging advanced SQL skills to conduct Operational Analytics at a company similar to Microsoft.
- The main goal is to analyze various datasets and tables to derive valuable insights that can help improve the company's operations and understand sudden changes in key metrics.
- As the Lead Data Analyst, the tasks involve investigating metric spikes, measuring user engagement, growth, retention, and analyzing email engagement.
- The scope includes tackling tasks encompassing investigating metric spikes, gauging user engagement, evaluating growth and retention, and scrutinizing email engagement.

PURPOSE OF CASE STUDY 1 –JOB DATA ANALYSIS

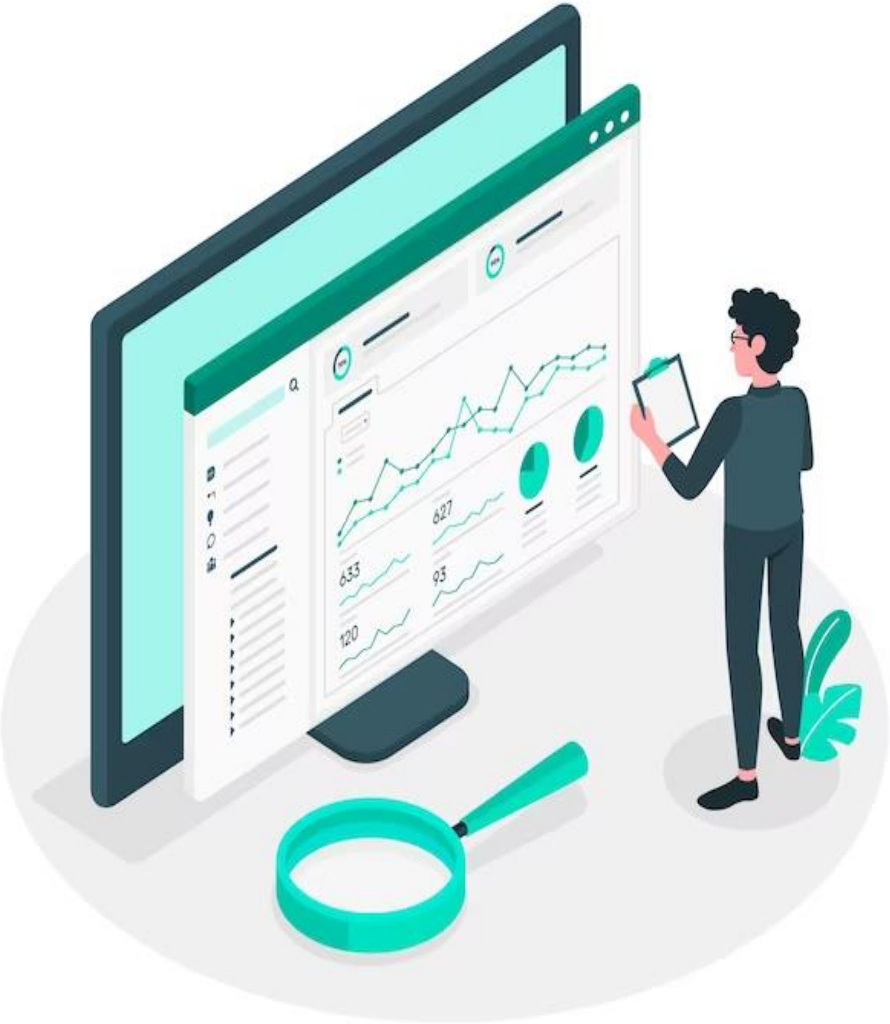
- The purpose of the "Job Data Analysis" case study is to gain insights into the company's job review process and identify opportunities for improving operational efficiency.
- By analyzing the provided job_data table, the project aims to address key questions related to job review patterns, language distribution, data quality, and overall process optimization.

PURPOSE OF CASE STUDY 2 -INVESTIGATING METRIC SPIKE

- The purpose of the "Investigating Metric Spike" case study is to analyze user and event data to understand sudden changes (spikes) in key metrics and gain insights into user behavior and engagement.
- This case study aims to address questions related to user engagement, growth, retention, device usage, and email engagement.

APPROCAH

- **Create a Database:** Create and set up a necessary database using the appropriate commands and database file. This database will serve as the foundation for our analysis.
- **Explore Data:** Familiarize yourself with the data structure, tables, and relationships. Understand the attributes that will enable us to assess user engagement.
- **Develop optimized query:** Develop SQL queries that target specific questions posed by the management team. Use these queries to extract meaningful information from the database, ensuring accuracy and efficiency.
- **Insights Extraction:** Analyze the query outputs to derive insights related to user behavior, engagement patterns, and app usage. Identify trends, preferences, and potential areas for improvement.
- **Generate efficient report:** Compile the findings into a comprehensive report. This report will be presented to the leadership team and should include detailed explanations of the SQL queries used, along with accompanying outputs in the form of snapshots.



TECH – STACK USED:

Tech-stack used in this project are MY-SQL workbench ,database and Microsoft PowerPoint

1. MySQL Workbench ,Database : MySQL Workbench was chosen as the primary tool for this project due to its efficient capabilities for SQL development, database management, and visualization. Its user-friendly interface allows for efficient query development, data exploration, and result visualization, which are crucial for our data analysis tasks. MySQL was selected as the database system for this project due to its open-source nature, performance, and widespread usage in various industries. It provides reliable data storage and retrieval, making it well-suited for handling large datasets associated with user interactions and engagement on the Instagram app.

2. PowerPoint Software - Microsoft PowerPoint: For creating the final report to present to the leadership team, we will use PDF or PowerPoint software. These tools allow us to compile our analysis, SQL query outputs, and insights into a visually appealing and organized format, facilitating clear communication of our findings and recommendations.

EXECUTION OF THE QUERIES:

Step 1: Extract the dataset from the given pdf in the dashboard

Step 2:create a database
Syntax: create database database_name

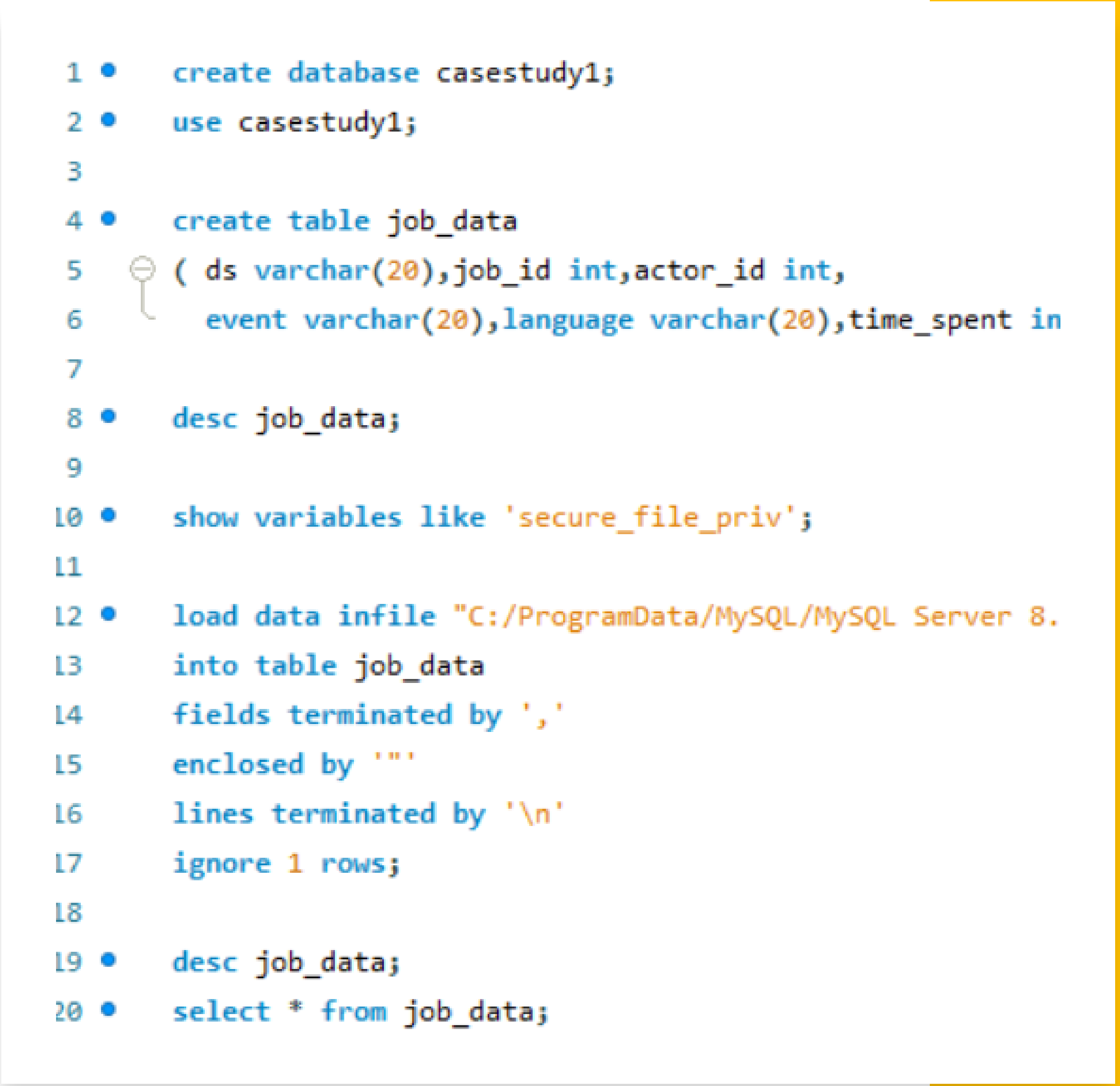
Step3: Now use the database for further execution
Syntax: use database_name;

Step 4: Create the required tables and insert data into them

Step5: Now we are good to go .Perform the required operations according to the given queries

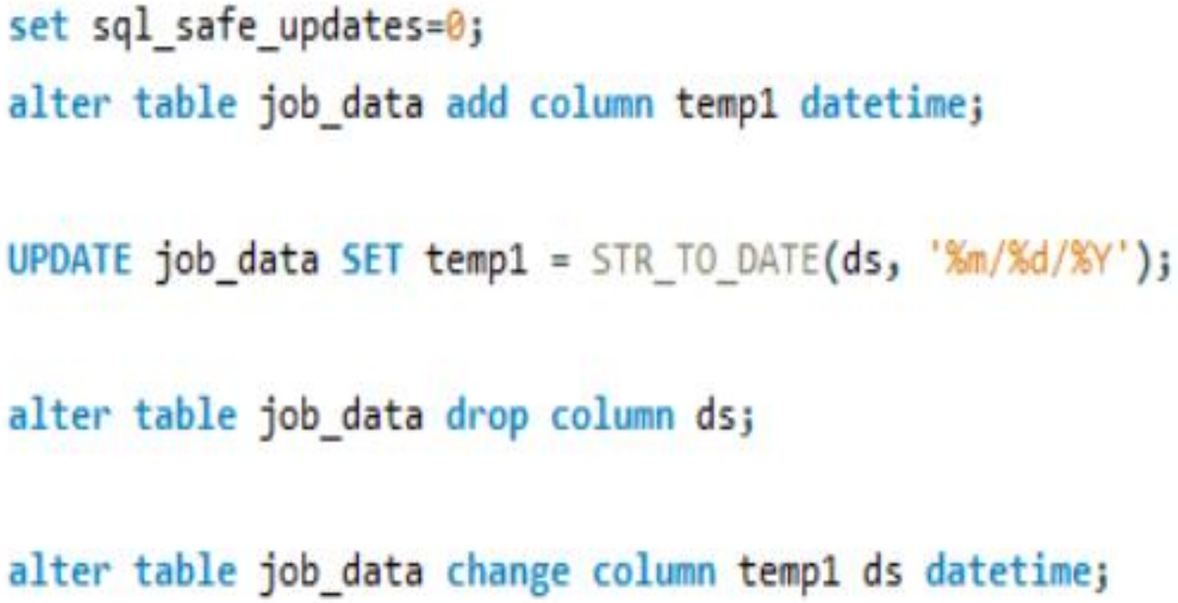
CASE STUDY 1: JOB DATA ANALYSIS

Database creation and extracting data from csv file to MY-SQL



```
create database casestudy1;
use casestudy1;

create table job_data
( ds varchar(20),job_id int,actor_id int,
  event varchar(20),language varchar(20),time_spent in

desc job_data;

show variables like 'secure_file_priv';

load data infile "C:/ProgramData/MySQL/MySQL Server 8.
into table job_data
fields terminated by ','
enclosed by '"'
lines terminated by '\n'
ignore 1 rows;

desc job_data;
select * from job_data;
```

CHANGE THE VARCHAR DATATYPE OF DS TO DATE TIME:



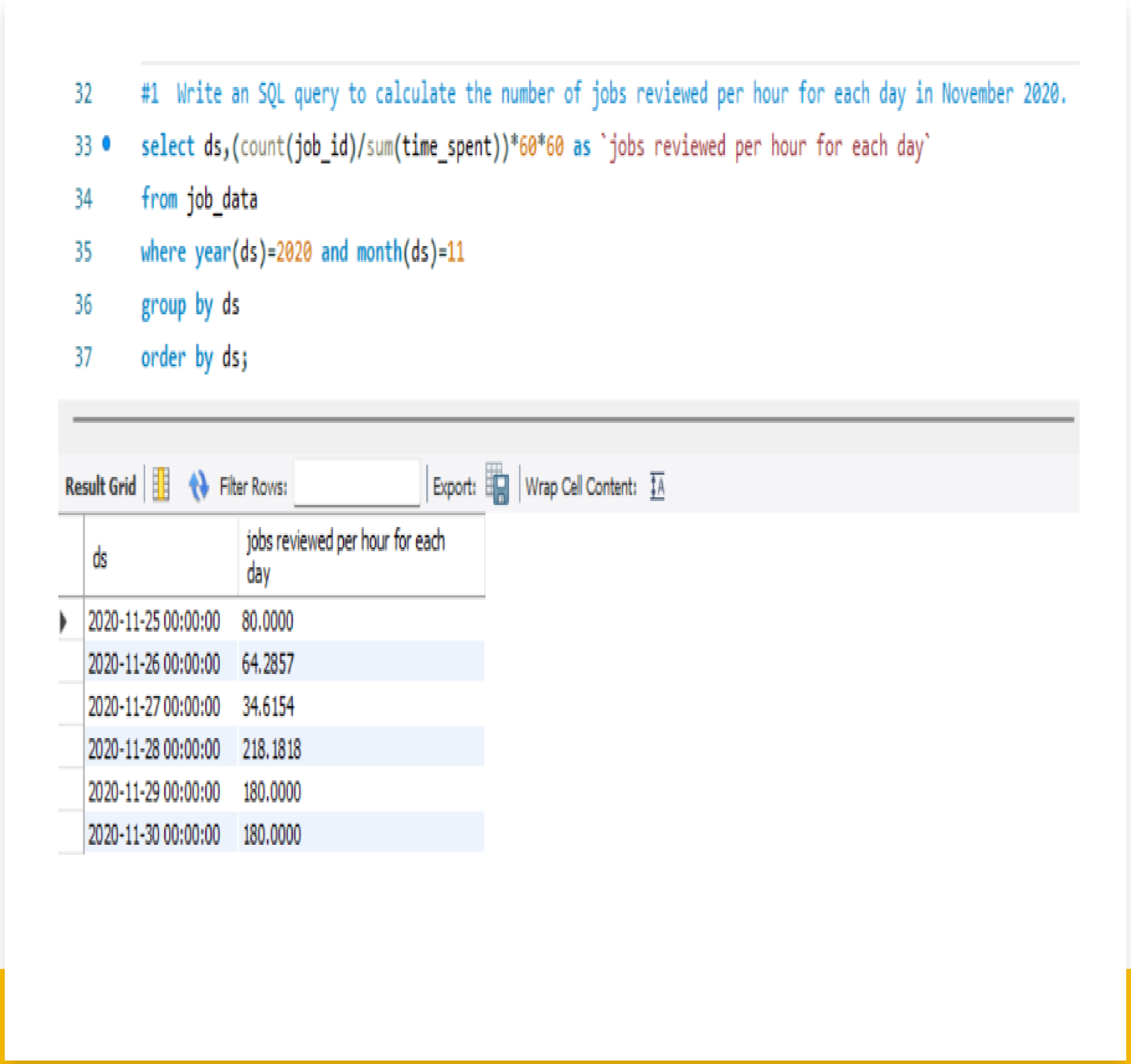
```sql
set sql_safe_updates=0;
alter table job_data add column temp1 datetime;

UPDATE job_data SET temp1 = STR_TO_DATE(ds, '%m/%d/%Y');

alter table job_data drop column ds;

alter table job_data change column temp1 ds datetime;
```

TASKS

A. Jobs Reviewed Over Time:

Objective: Calculate the number of jobs reviewed per hour for each day in November 2020.

Task: Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.



```
#1  Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.
select ds,(count(job_id)/sum(time_spent))*60*60 as `jobs reviewed per hour for each day`
from job_data
where year(ds)=2020 and month(ds)=11
group by ds
order by ds;
```

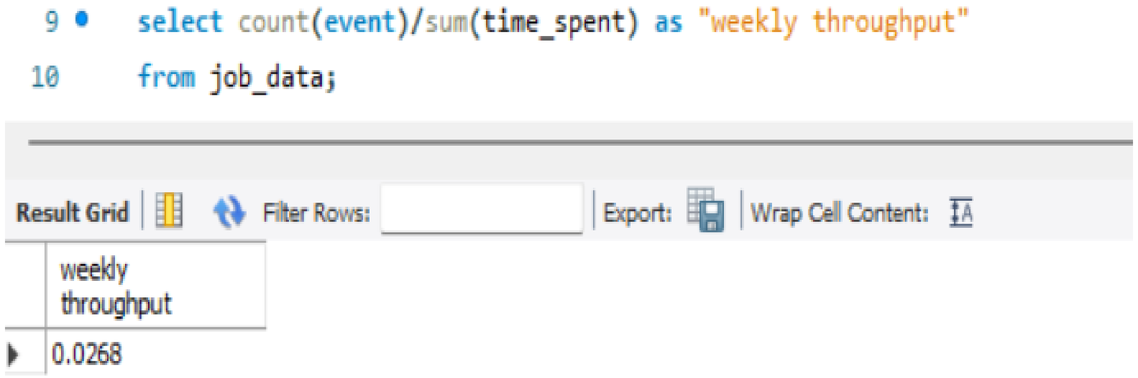
ds	jobs reviewed per hour for each day
2020-11-25 00:00:00	80.0000
2020-11-26 00:00:00	64.2857
2020-11-27 00:00:00	34.6154
2020-11-28 00:00:00	218.1818
2020-11-29 00:00:00	180.0000
2020-11-30 00:00:00	180.0000

B. Throughput Analysis:

Objective: Calculate the 7-day rolling average of throughput (number of events per second).

Your Task: Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

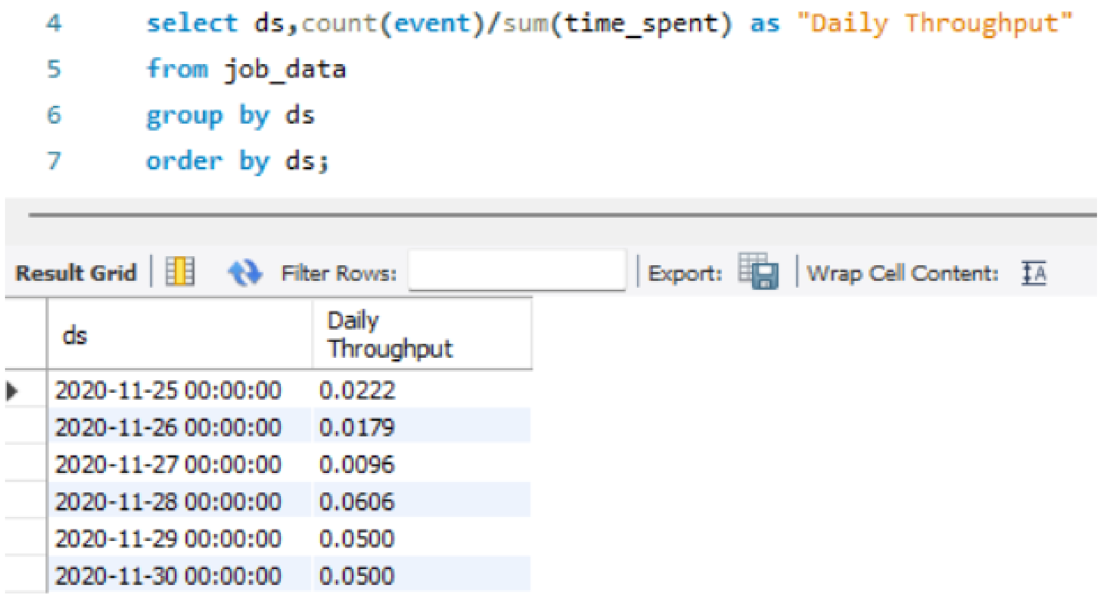
Weekly throughput



```sql
select count(event)/sum(time_spent) as "weekly throughput"
from job_data;
```

weekly throughput
0.0268

Daily throughput

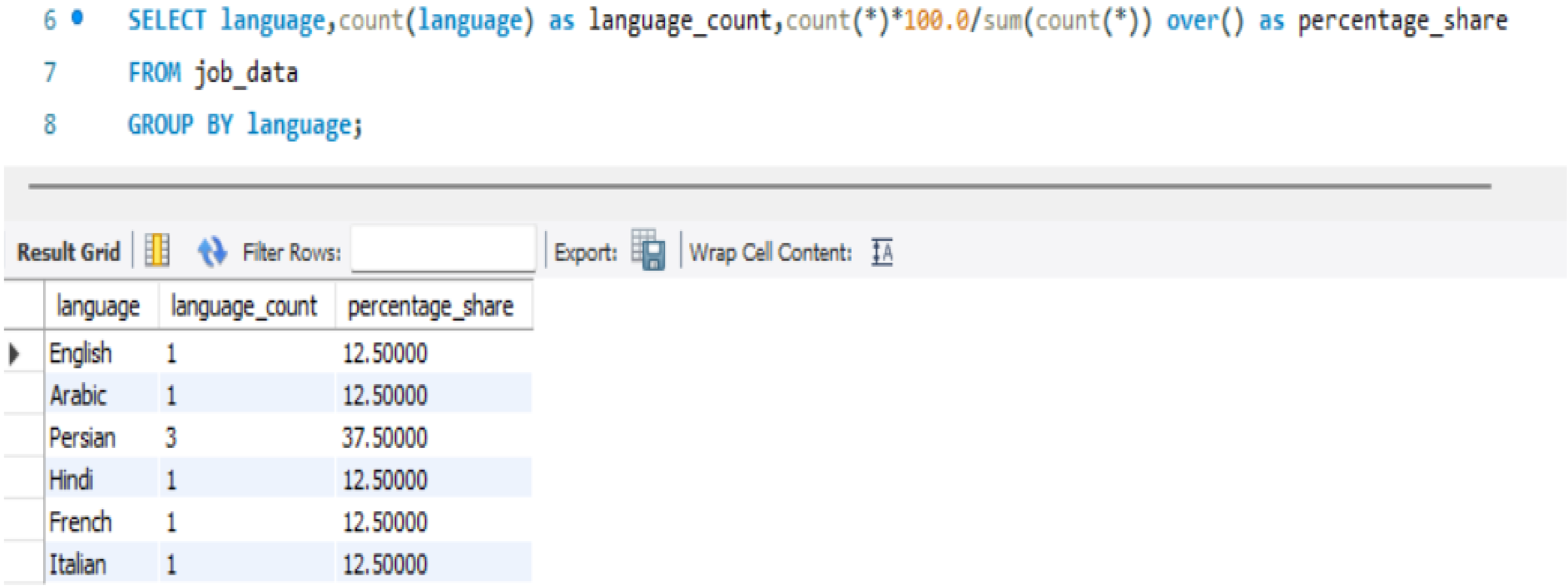


```sql
select ds,count(event)/sum(time_spent) as "Daily Throughput"
from job_data
group by ds
order by ds;
```

ds	Daily Throughput
2020-11-25 00:00:00	0.0222
2020-11-26 00:00:00	0.0179
2020-11-27 00:00:00	0.0096
2020-11-28 00:00:00	0.0606
2020-11-29 00:00:00	0.0500
2020-11-30 00:00:00	0.0500

C. Language Share Analysis:

- Objective: Calculate the percentage share of each language in the last 30 days.
- Your Task: Write an SQL query to calculate the percentage share of each language over the last 30 days.

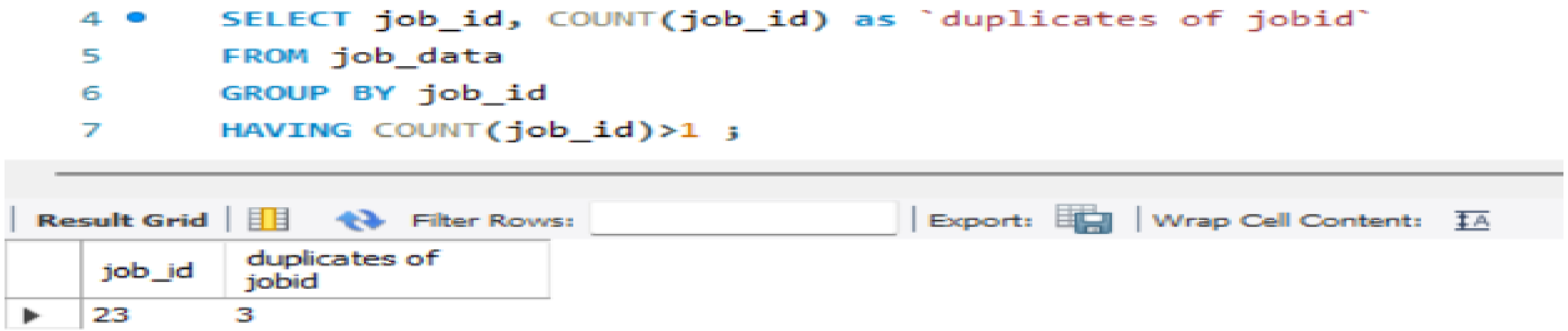


```sql
SELECT language,count(language) as language_count,count(*)*100.0/sum(count(*)) over() as percentage_share
FROM job_data
GROUP BY language;
```

language	language_count	percentage_share
English	1	12.50000
Arabic	1	12.50000
Persian	3	37.50000
Hindi	1	12.50000
French	1	12.50000
Italian	1	12.50000

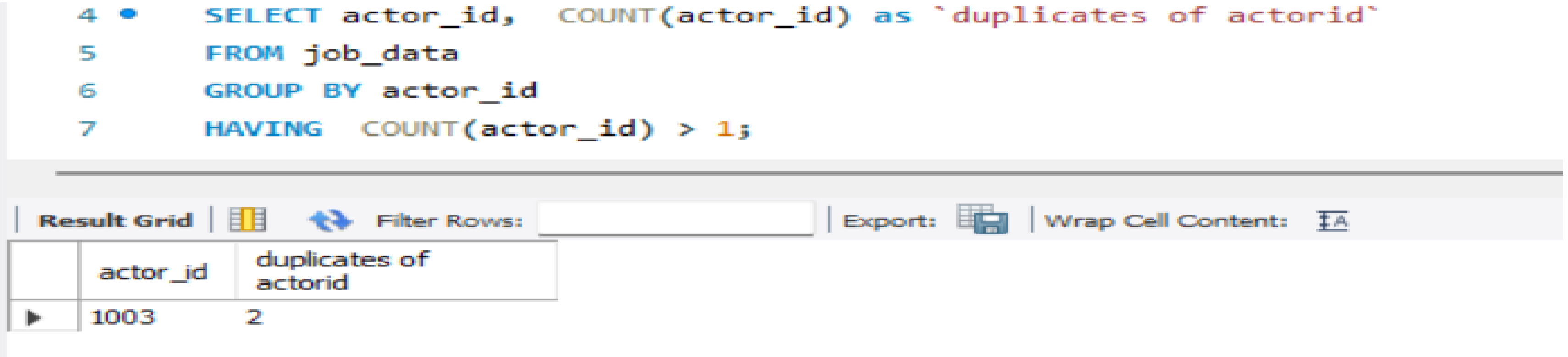
D. Duplicate Rows Detection:

- Objective: Identify duplicate rows in the data.
- Your Task: Write an SQL query to display duplicate rows from the job_data table.



```sql
SELECT job_id, COUNT(job_id) as `duplicates of jobid`
FROM job_data
GROUP BY job_id
HAVING COUNT(job_id)>1 ;
```

job_id	duplicates of jobid
23	3



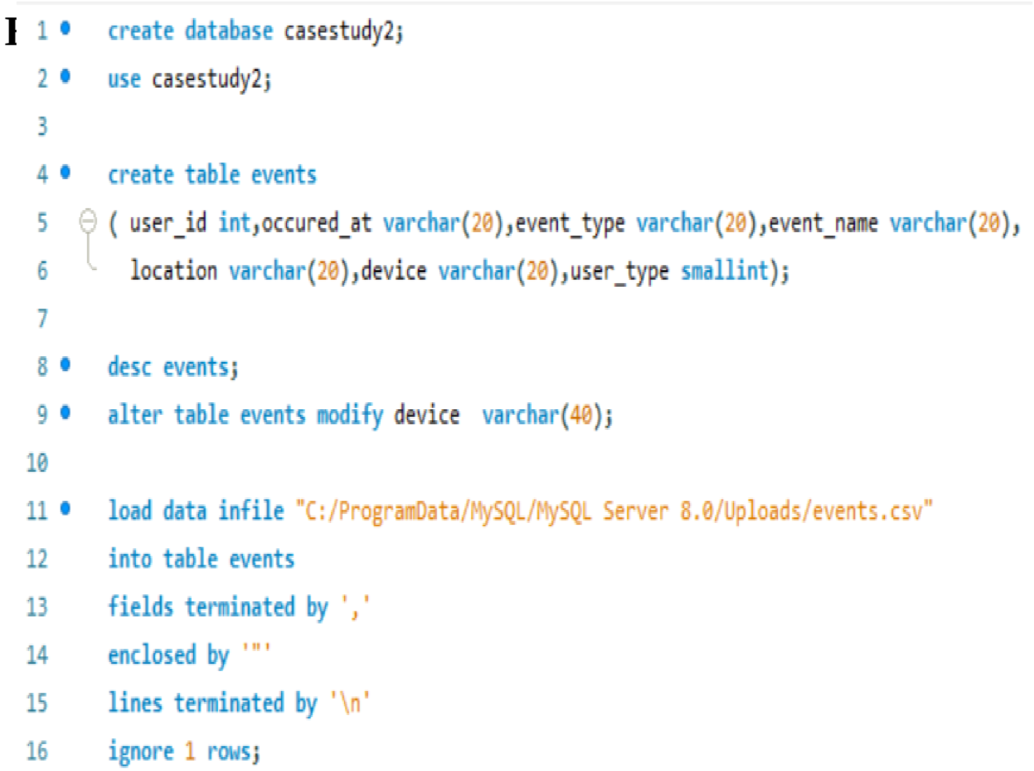
```sql
SELECT actor_id,  COUNT(actor_id) as `duplicates of actorid`
FROM job_data
GROUP BY actor_id
HAVING  COUNT(actor_id) > 1;
```

actor_id	duplicates of actorid
1003	2

CASE STUDY 2: INVESTIGATING METRIC SPIKE

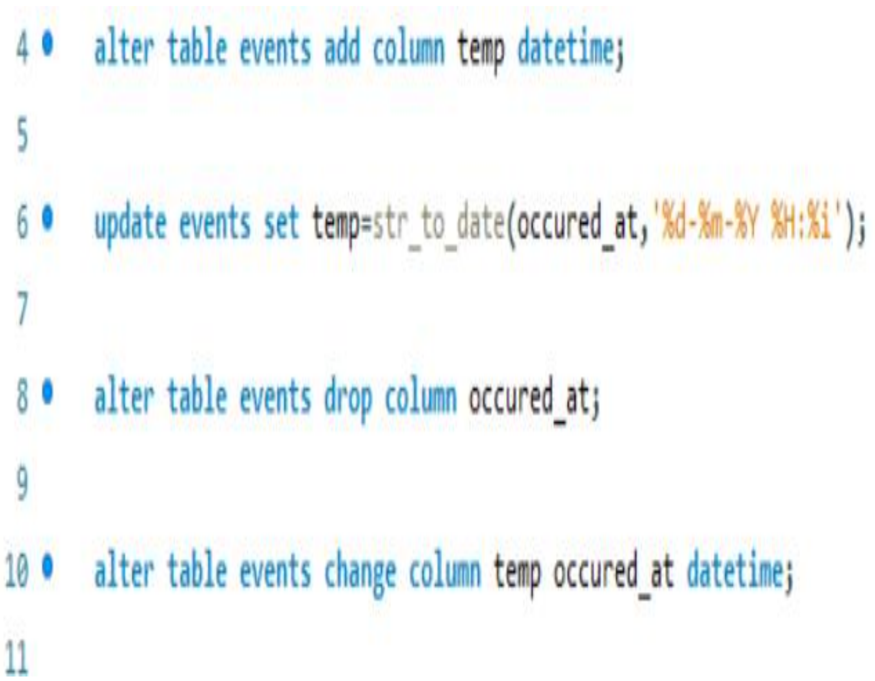
In this case study we will use three tables users ,events ,email_events.

Database creation and extracting data from csv file to MY-SQL



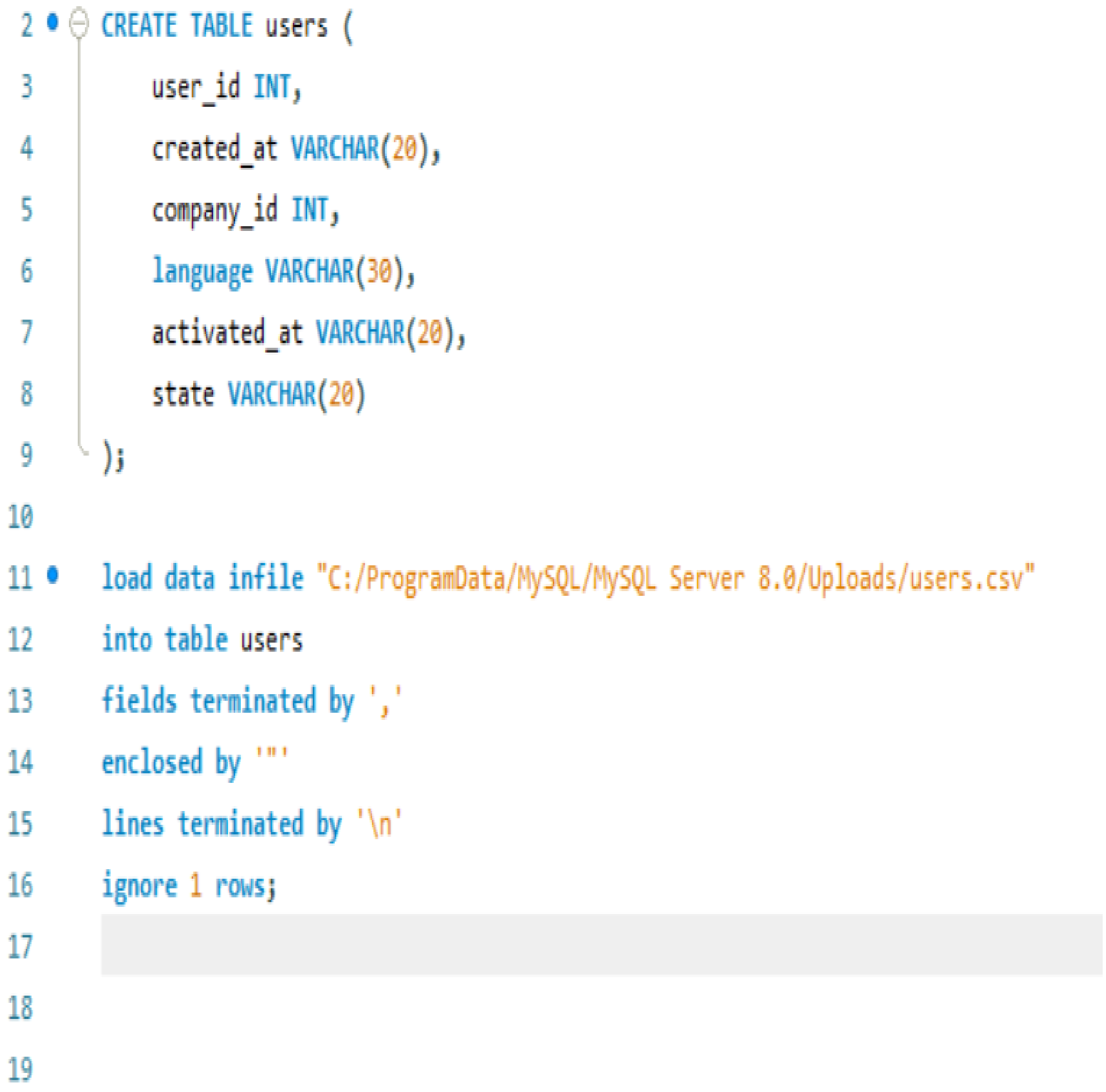
```
create database casestudy2;
use casestudy2;

create table events
( user_id int,occured_at varchar(20),event_type varchar(20),event_name varchar(20),
  location varchar(20),device varchar(20),user_type smallint);

desc events;
alter table events modify device  varchar(40);

load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/events.csv"
into table events
fields terminated by ','
enclosed by '"'
lines terminated by '\n'
ignore 1 rows;
```

Change the varchar datatype to datatime

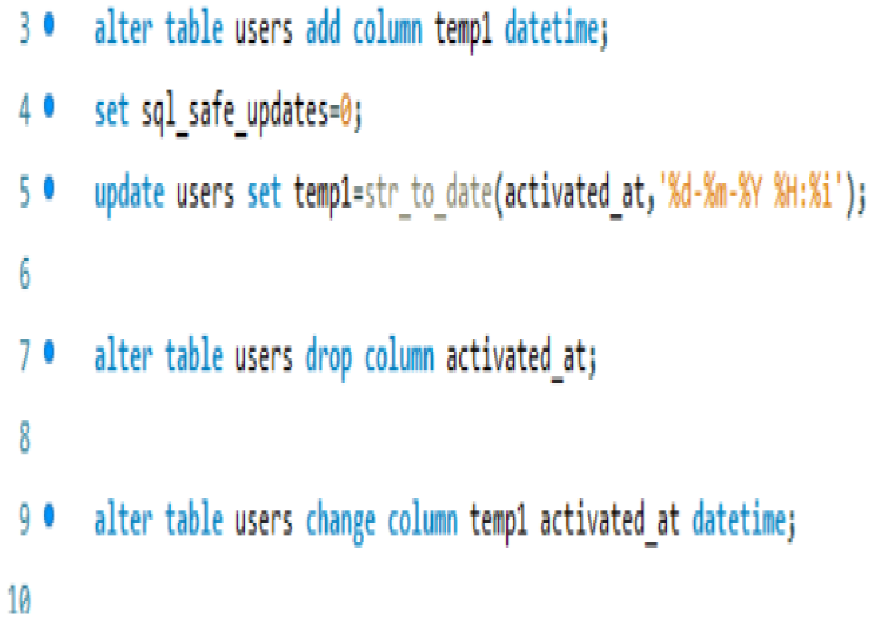


```
alter table events add column temp datetime;

update events set temp=str_to_date(occured_at,'%d-%m-%Y %H:%i');

alter table events drop column occured_at;

alter table events change column temp occured_at datetime;

```

USERS

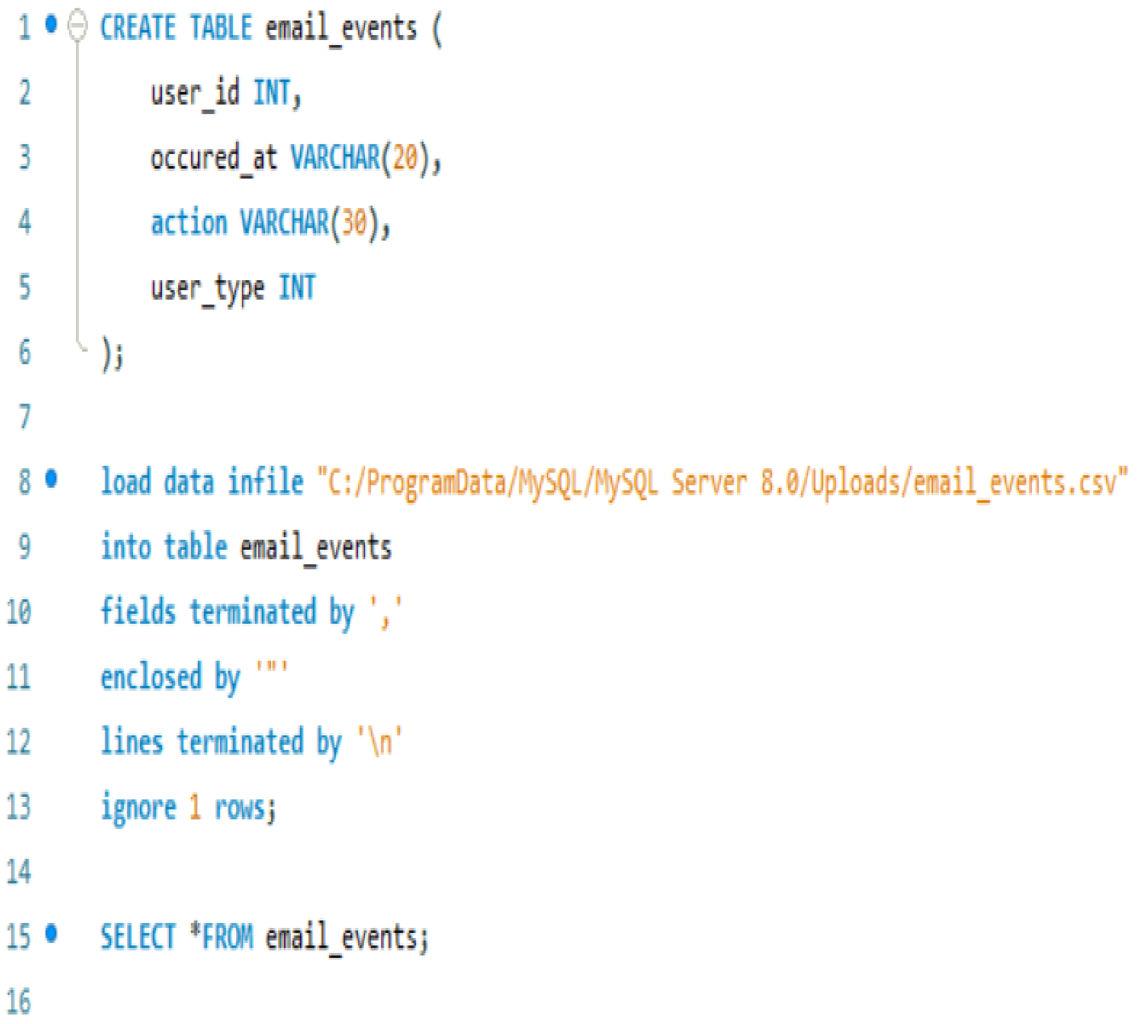


```
CREATE TABLE users (
    user_id INT,
    created_at VARCHAR(20),
    company_id INT,
    language VARCHAR(30),
    activated_at VARCHAR(20),
    state VARCHAR(20)
);

load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users.csv"
into table users
fields terminated by ','
enclosed by '"'
lines terminated by '\n'
ignore 1 rows;
```

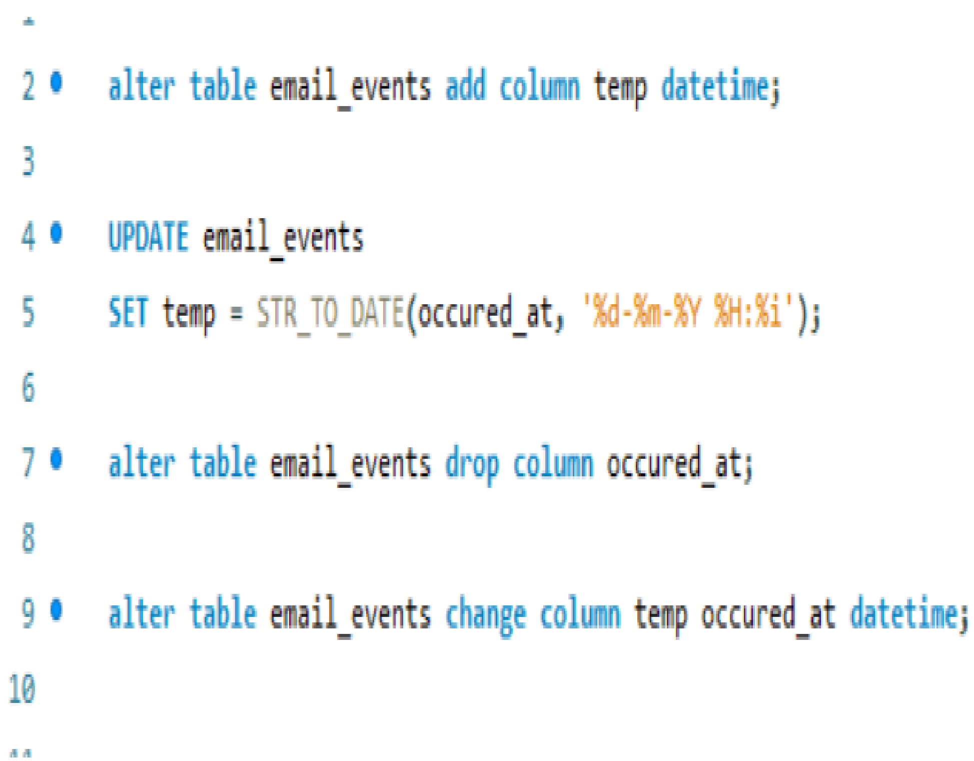


```
alter table users add column temp1 datetime;
set sql_safe_updates=0;
update users set temp1=str_to_date(activated_at,'%d-%m-%Y %H:%i');

alter table users drop column activated_at;

alter table users change column temp1 activated_at datetime;
```

EMAIL_EVENTS



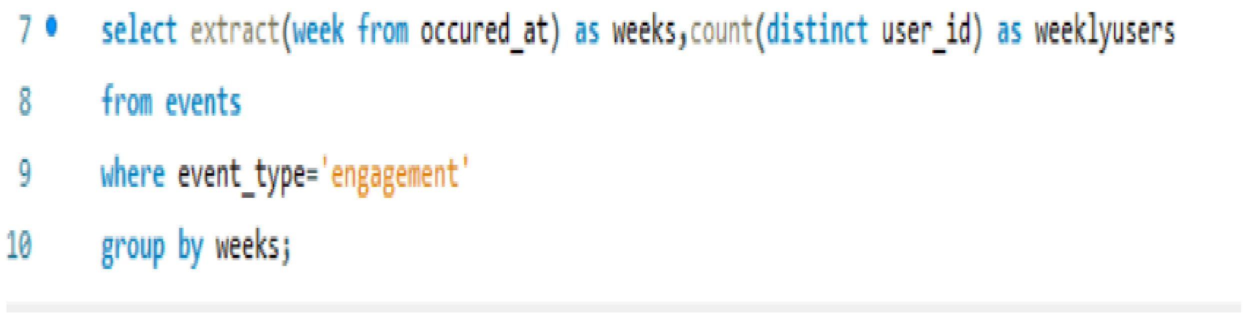
```sql
CREATE TABLE email_events (
    user_id INT,
    occured_at VARCHAR(20),
    action VARCHAR(30),
    user_type INT
);

load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/email_events.csv"
into table email_events
fields terminated by ','
enclosed by '"'
lines terminated by '\n'
ignore 1 rows;

SELECT *FROM email_events;
```



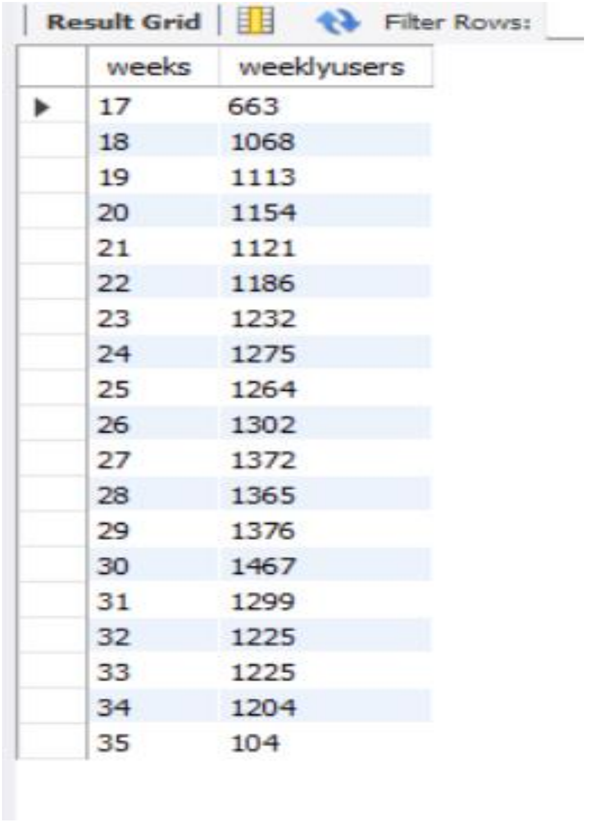
```sql
alter table email_events add column temp datetime;

UPDATE email_events
SET temp = STR_TO_DATE(occured_at, '%d-%m-%Y %H:%i');

alter table email_events drop column occured_at;

alter table email_events change column temp occured_at datetime;
```

A. Weekly User Engagement:

- Objective: Measure the activeness of users on a weekly basis.
- Your Task: Write an SQL query to calculate the weekly user engagement.



```sql
select extract(week from occured_at) as weeks,count(distinct user_id) as weeklyusers
from events
where event_type='engagement'
group by weeks;
```



weeks	weeklyusers
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

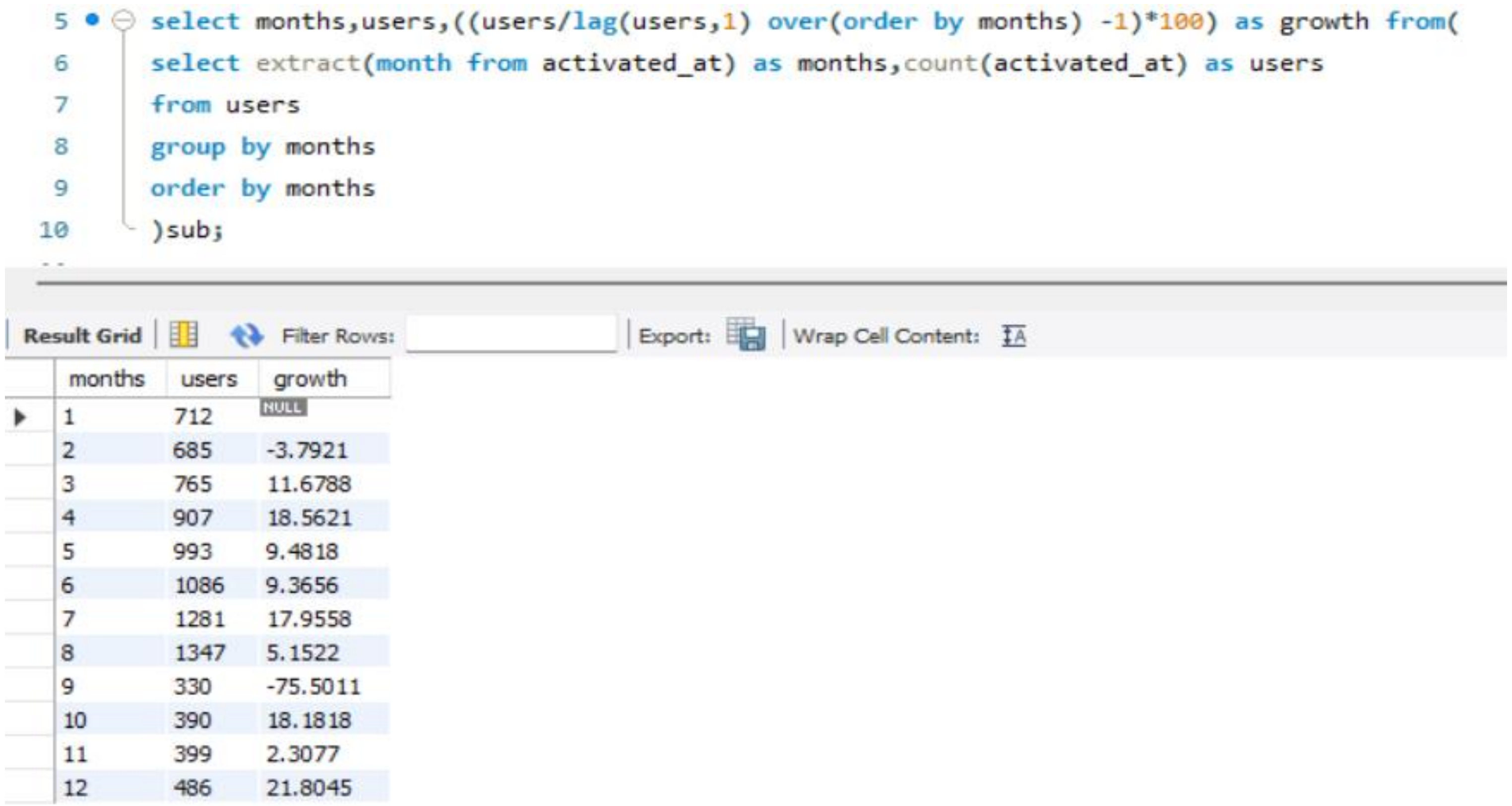
B. User Growth Analysis:

Objective: Analyze the growth of users over time for a product.

Your Task: Write an SQL query to calculate the user growth for the product.

This question I solved by two approaches

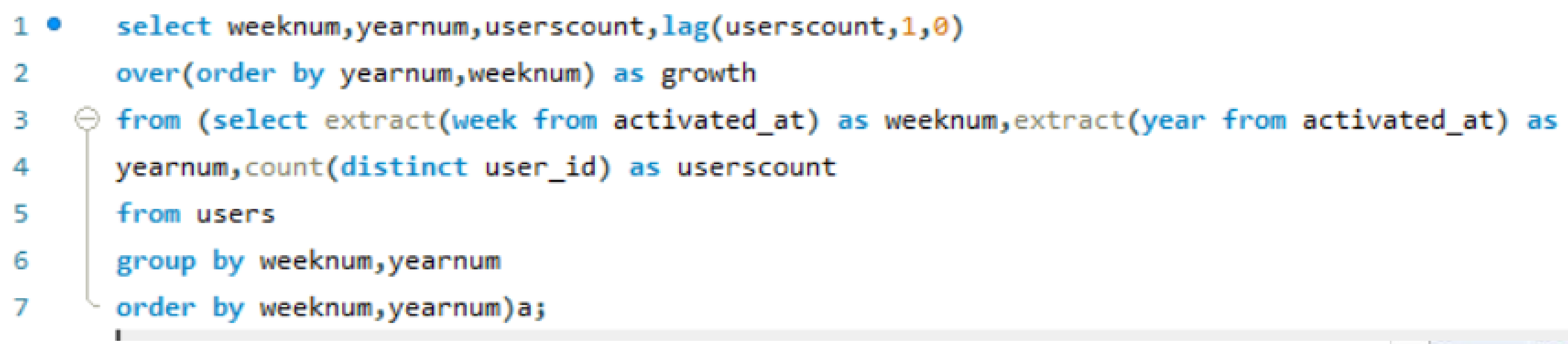
APPROACH-1



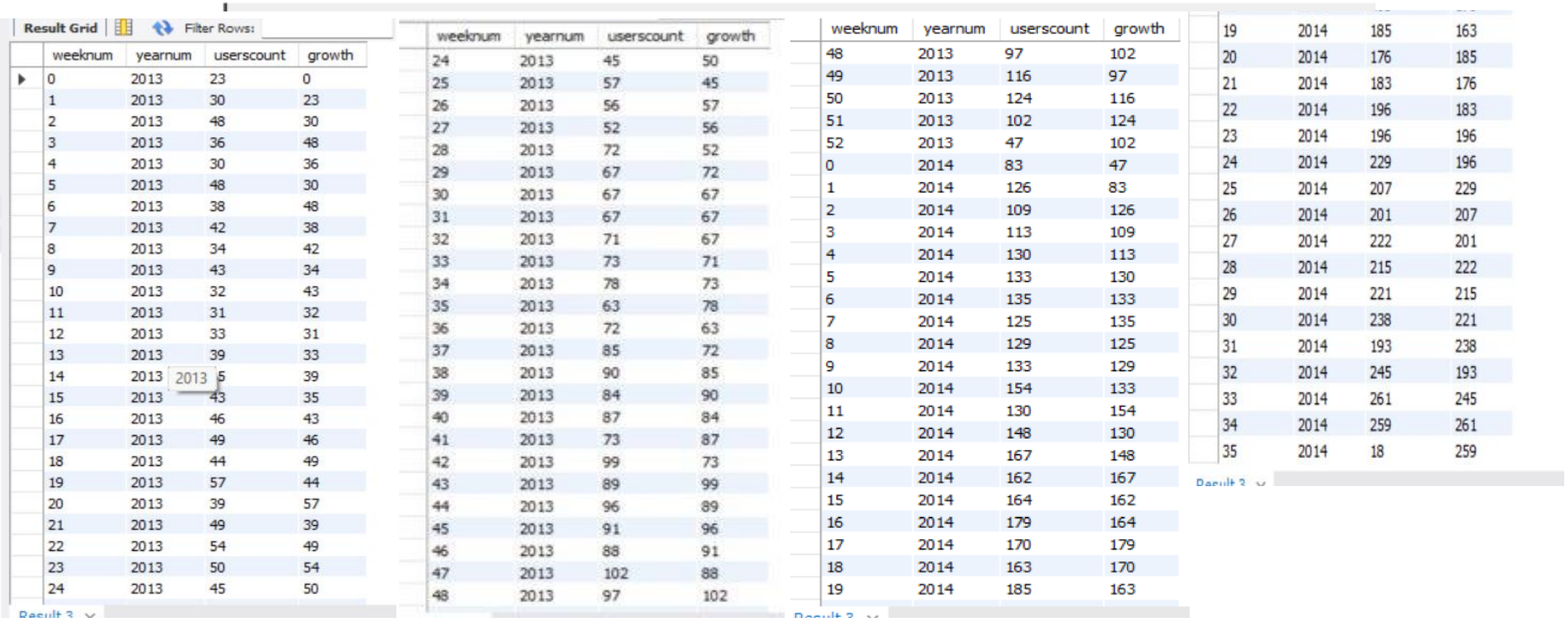
```sql
select months,users,((users/lag(users,1) over(order by months) -1)*100) as growth from(
select extract(month from activated_at) as months,count(activated_at) as users
from users
group by months
order by months
)sub;
```

months	users	growth
1	712	NULL
2	685	-3.7921
3	765	11.6788
4	907	18.5621
5	993	9.4818
6	1086	9.3656
7	1281	17.9558
8	1347	5.1522
9	330	-75.5011
10	390	18.1818
11	399	2.3077
12	486	21.8045

APPROACH-2



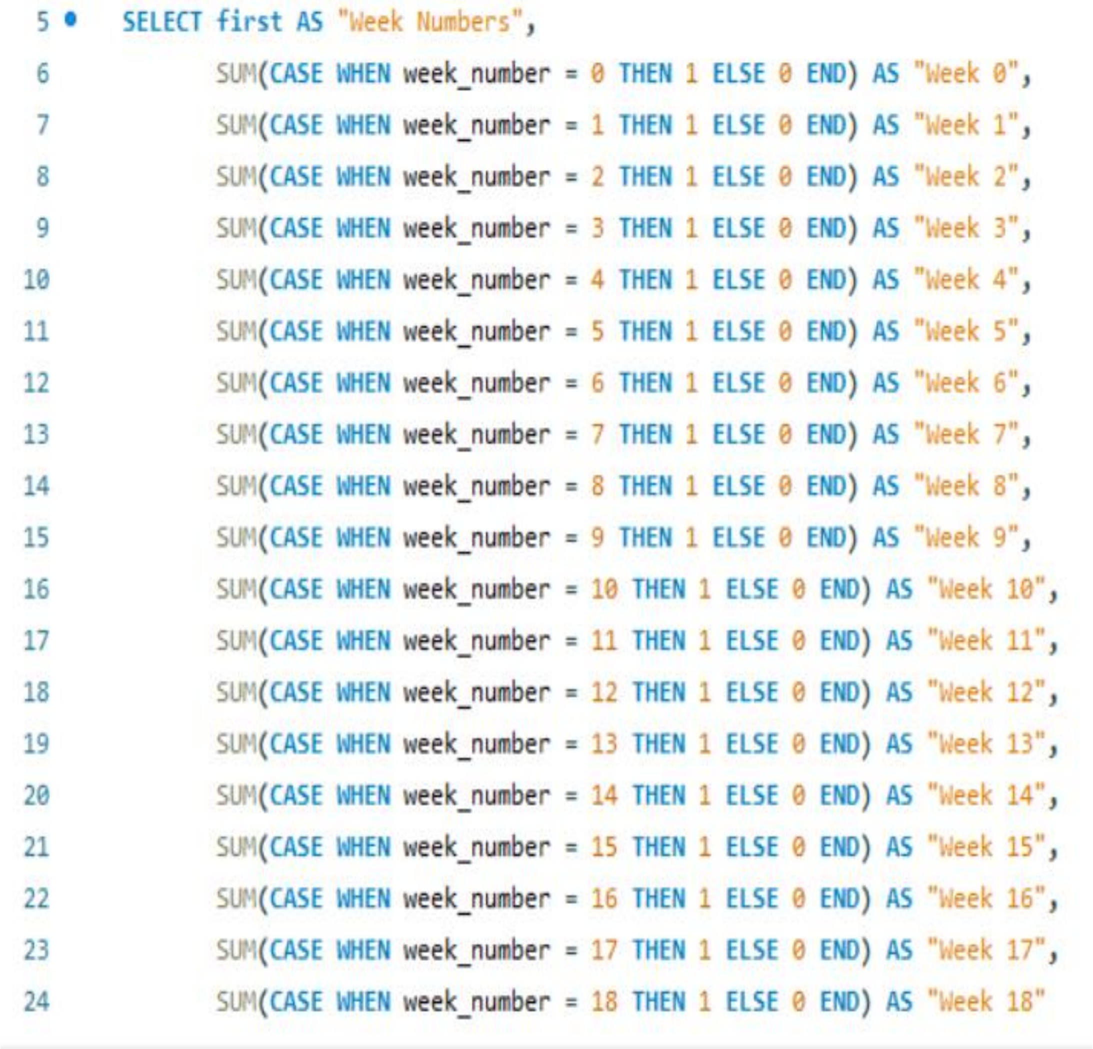
```
select weeknum,yearnum,userscount,lag(userscount,1,0)
over(order by yearnum,weeknum) as growth
from (select extract(week from activated_at) as weeknum,extract(year from activated_at) as
yearnum,count(distinct user_id) as userscount
from users
group by weeknum,yearnum
order by weeknum,yearnum)a;
```



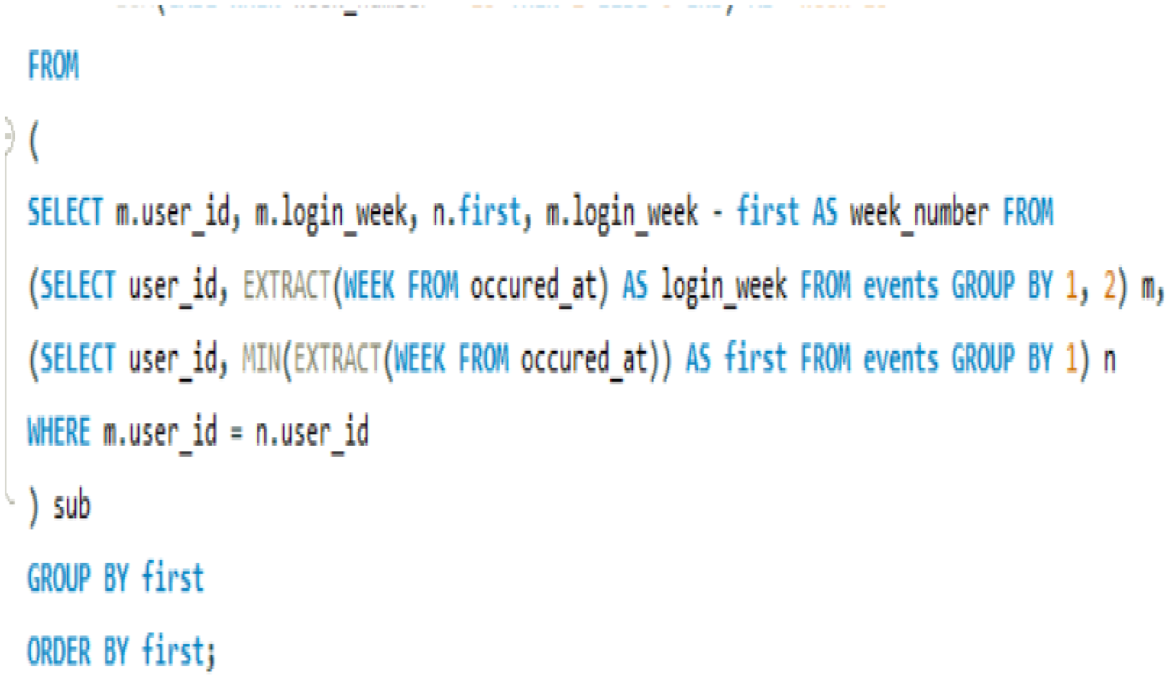
weeknum	yearnum	userscount	growth
0	2013	23	0
1	2013	30	23
2	2013	48	30
3	2013	36	48
4	2013	30	36
5	2013	48	30
6	2013	38	48
7	2013	42	38
8	2013	34	42
9	2013	43	34
10	2013	32	43
11	2013	31	32
12	2013	33	31
13	2013	39	33
14	2013	5	39
15	2013	43	35
16	2013	46	43
17	2013	49	46
18	2013	44	49
19	2013	57	44
20	2013	39	57
21	2013	49	39
22	2013	54	49
23	2013	50	54
24	2013	45	50
25	2013	57	45
26	2013	56	57
27	2013	52	56
28	2013	72	52
29	2013	67	72
30	2013	67	67
31	2013	67	67
32	2013	71	67
33	2013	73	71
34	2013	78	73
35	2013	63	78
36	2013	72	63
37	2013	85	72
38	2013	90	85
39	2013	84	90
40	2013	87	84
41	2013	73	87
42	2013	99	73
43	2013	89	99
44	2013	96	89
45	2013	91	96
46	2013	88	91
47	2013	102	88
48	2013	97	102
49	2013	116	97
50	2013	124	116
51	2013	102	124
52	2013	47	102
0	2014	83	47
1	2014	126	83
2	2014	109	126
3	2014	113	109
4	2014	130	113
5	2014	133	130
6	2014	135	133
7	2014	125	135
8	2014	129	125
9	2014	133	129
10	2014	154	133
11	2014	130	154
12	2014	148	130
13	2014	167	148
14	2014	162	167
15	2014	164	162
16	2014	179	164
17	2014	170	179
18	2014	163	170
19	2014	185	163
20	2014	176	185
21	2014	183	176
22	2014	196	183
23	2014	196	196
24	2014	229	196
25	2014	207	229
26	2014	201	207
27	2014	222	201
28	2014	215	222
29	2014	221	215
30	2014	238	221
31	2014	193	238
32	2014	245	193
33	2014	261	245
34	2014	259	261
35	2014	18	259

C. Weekly Retention Analysis:

- Objective: Analyze the retention of users on a weekly basis after signing up for a product.
- Your Task: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

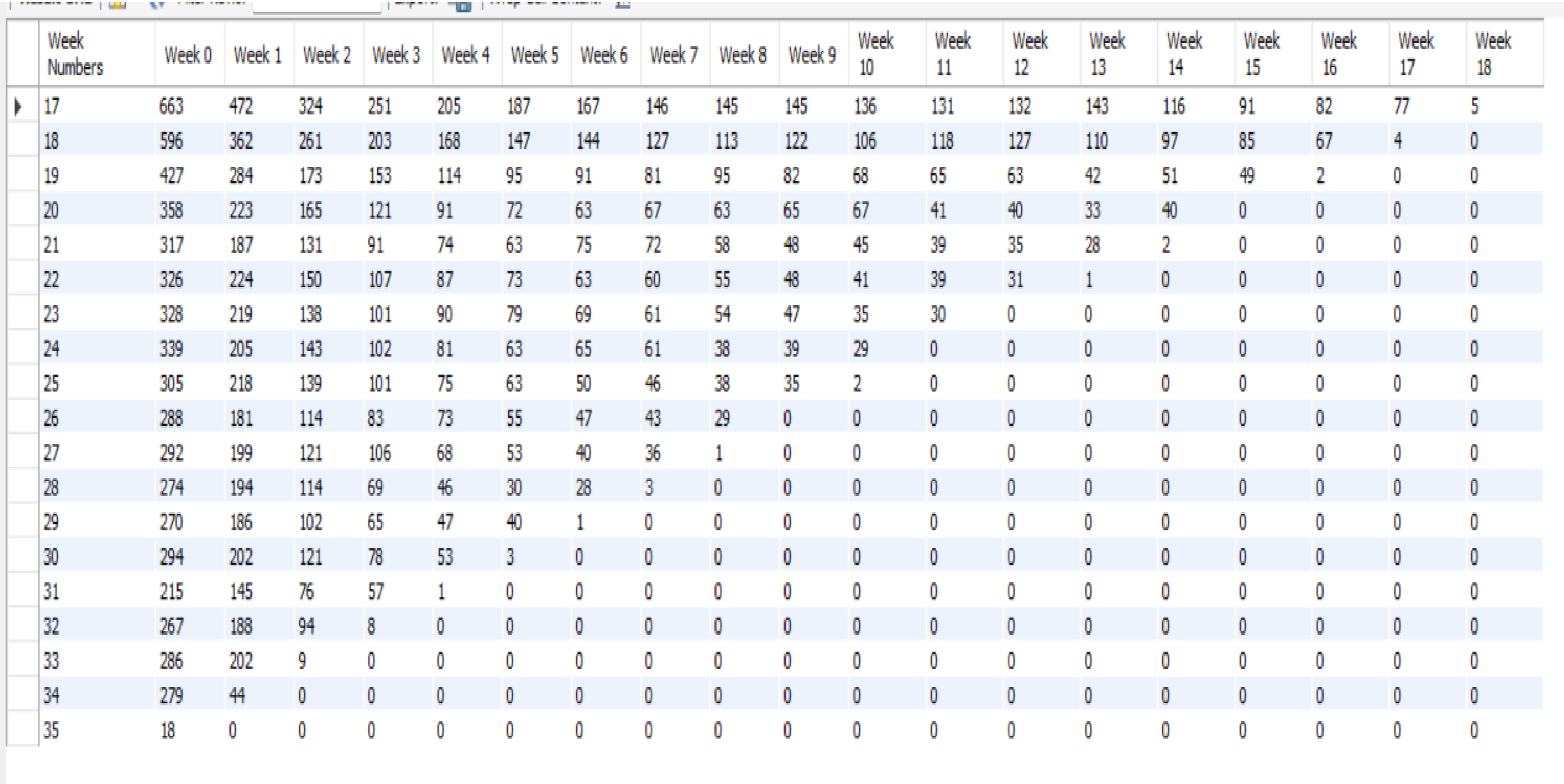


```sql
SELECT first AS "Week Numbers",
        SUM(CASE WHEN week_number = 0 THEN 1 ELSE 0 END) AS "Week 0",
        SUM(CASE WHEN week_number = 1 THEN 1 ELSE 0 END) AS "Week 1",
        SUM(CASE WHEN week_number = 2 THEN 1 ELSE 0 END) AS "Week 2",
        SUM(CASE WHEN week_number = 3 THEN 1 ELSE 0 END) AS "Week 3",
        SUM(CASE WHEN week_number = 4 THEN 1 ELSE 0 END) AS "Week 4",
        SUM(CASE WHEN week_number = 5 THEN 1 ELSE 0 END) AS "Week 5",
        SUM(CASE WHEN week_number = 6 THEN 1 ELSE 0 END) AS "Week 6",
        SUM(CASE WHEN week_number = 7 THEN 1 ELSE 0 END) AS "Week 7",
        SUM(CASE WHEN week_number = 8 THEN 1 ELSE 0 END) AS "Week 8",
        SUM(CASE WHEN week_number = 9 THEN 1 ELSE 0 END) AS "Week 9",
        SUM(CASE WHEN week_number = 10 THEN 1 ELSE 0 END) AS "Week 10",
        SUM(CASE WHEN week_number = 11 THEN 1 ELSE 0 END) AS "Week 11",
        SUM(CASE WHEN week_number = 12 THEN 1 ELSE 0 END) AS "Week 12",
        SUM(CASE WHEN week_number = 13 THEN 1 ELSE 0 END) AS "Week 13",
        SUM(CASE WHEN week_number = 14 THEN 1 ELSE 0 END) AS "Week 14",
        SUM(CASE WHEN week_number = 15 THEN 1 ELSE 0 END) AS "Week 15",
        SUM(CASE WHEN week_number = 16 THEN 1 ELSE 0 END) AS "Week 16",
        SUM(CASE WHEN week_number = 17 THEN 1 ELSE 0 END) AS "Week 17",
        SUM(CASE WHEN week_number = 18 THEN 1 ELSE 0 END) AS "Week 18"
```



```sql
FROM
(
SELECT m.user_id, m.login_week, n.first, m.login_week - first AS week_number FROM
(SELECT user_id, EXTRACT(WEEK FROM occured_at) AS login_week FROM events GROUP BY 1, 2) m,
(SELECT user_id, MIN(EXTRACT(WEEK FROM occured_at)) AS first FROM events GROUP BY 1) n
WHERE m.user_id = n.user_id
) sub
GROUP BY first
ORDER BY first;
```

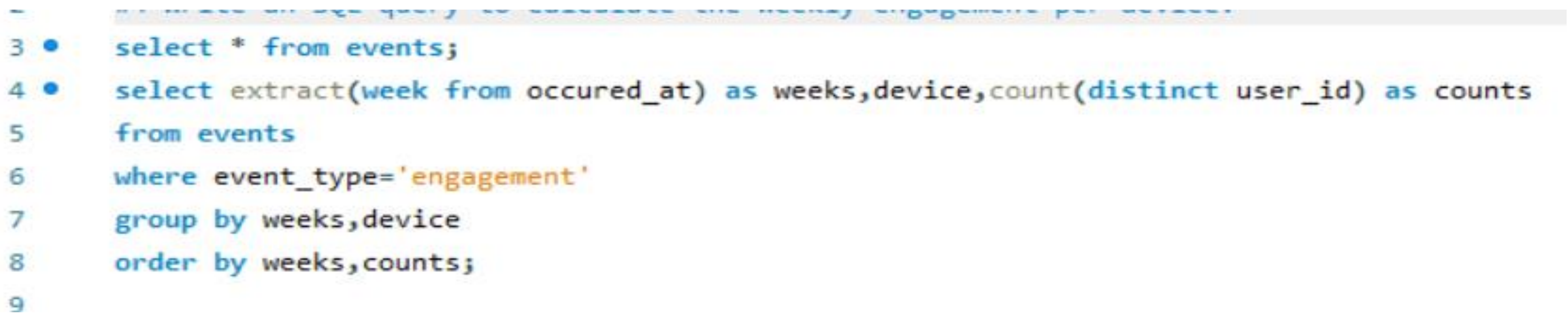
CODE:



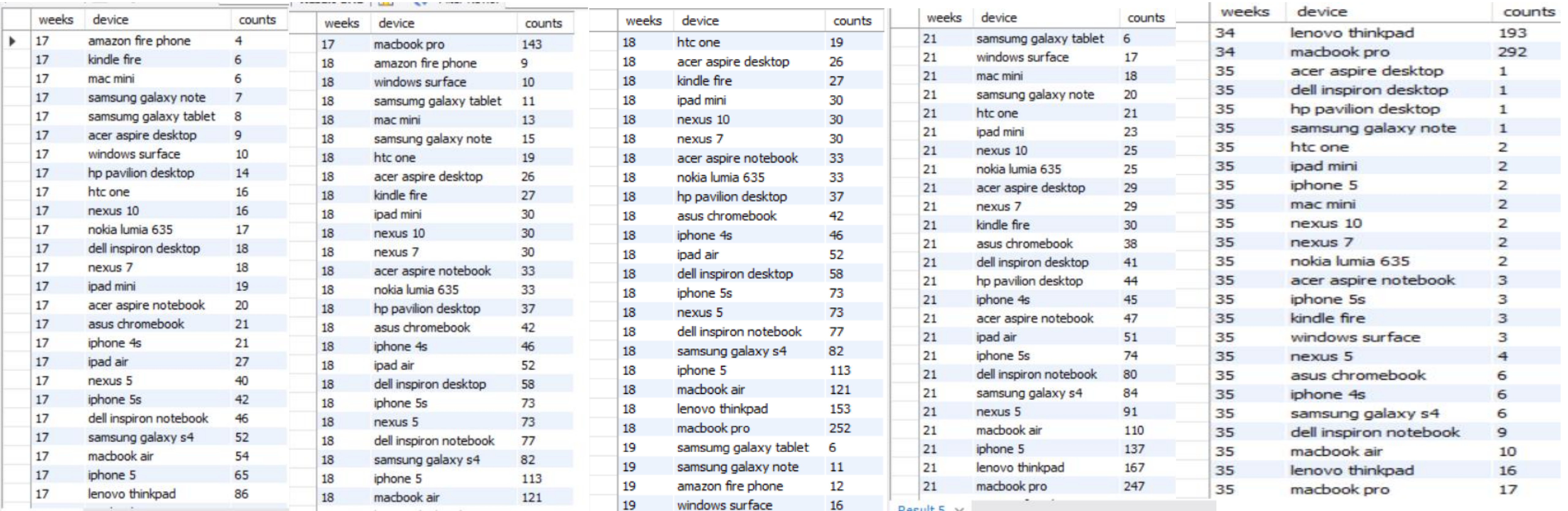
Week Numbers	Week 0	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18
17	663	472	324	251	205	187	167	146	145	145	136	131	132	143	116	91	82	77	5
18	596	362	261	203	168	147	144	127	113	122	106	118	127	110	97	85	67	4	0
19	427	284	173	153	114	95	91	81	95	82	68	65	63	42	51	49	2	0	0
20	358	223	165	121	91	72	63	67	63	65	67	41	40	33	40	0	0	0	0
21	317	187	131	91	74	63	75	72	58	48	45	39	35	28	2	0	0	0	0
22	326	224	150	107	87	73	63	60	55	48	41	39	31	1	0	0	0	0	0
23	328	219	138	101	90	79	69	61	54	47	35	30	0	0	0	0	0	0	0
24	339	205	143	102	81	63	65	61	38	39	29	0	0	0	0	0	0	0	0
25	305	218	139	101	75	63	50	46	38	35	2	0	0	0	0	0	0	0	0
26	288	181	114	83	73	55	47	43	29	0	0	0	0	0	0	0	0	0	0
27	292	199	121	106	68	53	40	36	1	0	0	0	0	0	0	0	0	0	0
28	274	194	114	69	46	30	28	3	0	0	0	0	0	0	0	0	0	0	0
29	270	186	102	65	47	40	1	0	0	0	0	0	0	0	0	0	0	0	0
30	294	202	121	78	53	3	0	0	0	0	0	0	0	0	0	0	0	0	0
31	215	145	76	57	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	267	188	94	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	286	202	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	279	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

D. Weekly Engagement Per Device:

- Objective: Measure the activeness of users on a weekly basis per device.
- Your Task: Write an SQL query to calculate the weekly engagement per device.



```sql
select * from events;
select extract(week from occured_at) as weeks,device,count(distinct user_id) as counts
from events
where event_type='engagement'
group by weeks,device
order by weeks,counts;
```



weeks	device	counts
17	amazon fire phone	4
17	kindle fire	6
17	mac mini	6
17	samsung galaxy note	7
17	samsumg galaxy tablet	8
17	acer aspire desktop	9
17	windows surface	10
17	hp pavilion desktop	14
17	htc one	16
17	nexus 10	16
17	nokia lumia 635	17
17	dell inspiron desktop	18
17	nexus 7	18
17	ipad mini	19
17	acer aspire notebook	20
17	asus chromebook	21
17	iphone 4s	21
17	ipad air	27
17	nexus 5	40
17	iphone 5s	42
17	dell inspiron notebook	46
17	samsung galaxy s4	52
17	macbook air	54
17	iphone 5	65
17	lenovo thinkpad	86

weeks	device	counts
17	macbook pro	143
18	amazon fire phone	9
18	windows surface	10
18	samsumg galaxy tablet	11
18	mac mini	13
18	samsung galaxy note	15
18	htc one	19
18	acer aspire desktop	26
18	kindle fire	27
18	ipad mini	30
18	nexus 10	30
18	nexus 7	30
18	acer aspire notebook	33
18	nokia lumia 635	33
18	hp pavilion desktop	37
18	asus chromebook	42
18	iphone 4s	46
18	ipad air	52
18	dell inspiron desktop	58
18	iphone 5s	73
18	nexus 5	73
18	dell inspiron notebook	77
18	samsung galaxy s4	82
18	iphone 5	113
18	macbook air	121

weeks	device	counts
18	htc one	19
18	acer aspire desktop	26
18	kindle fire	27
18	ipad mini	30
18	nexus 10	30
18	nexus 7	30
18	acer aspire notebook	33
18	nokia lumia 635	33
18	hp pavilion desktop	37
18	asus chromebook	42
18	iphone 4s	46
18	ipad air	52
18	dell inspiron desktop	58
18	iphone 5s	73
18	nexus 5	73
18	dell inspiron notebook	77
18	samsung galaxy s4	82
18	iphone 5	113
18	macbook air	121
18	lenovo thinkpad	153
18	macbook pro	252
19	samsumg galaxy tablet	6
19	samsung galaxy note	11
19	amazon fire phone	12
19	windows surface	16

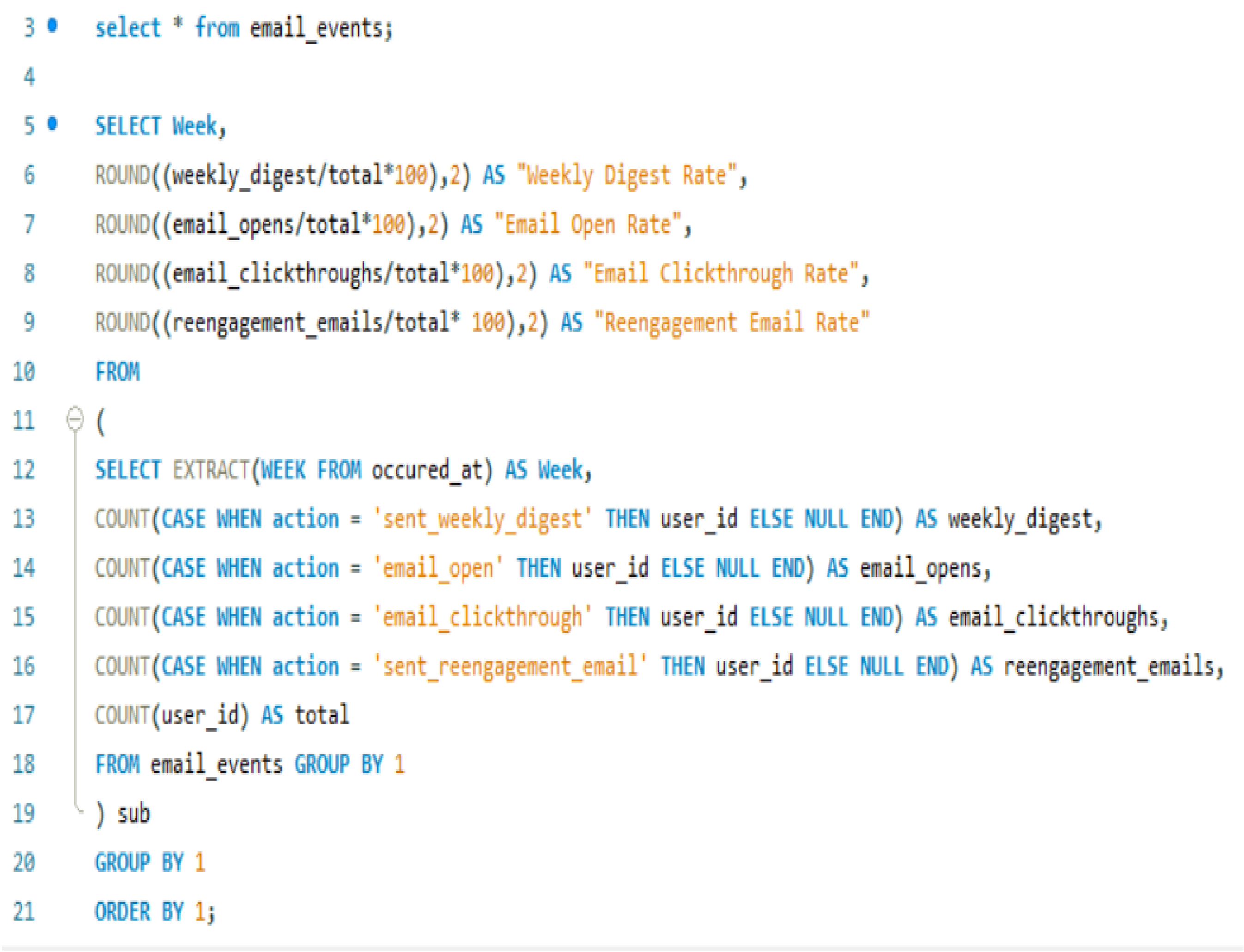
weeks	device	counts
21	samsumg galaxy tablet	6
21	windows surface	17
21	mac mini	18
21	samsung galaxy note	20
21	htc one	21
21	ipad mini	23
21	nexus 10	25
21	nokia lumia 635	25
21	acer aspire desktop	29
21	nexus 7	29
21	kindle fire	30
21	asus chromebook	38
21	dell inspiron desktop	41
21	hp pavilion desktop	44
21	iphone 4s	45
21	acer aspire notebook	47
21	ipad air	51
21	iphone 5s	74
21	dell inspiron notebook	80
21	samsung galaxy s4	84
21	nexus 5	91
21	macbook air	110
21	iphone 5	137
21	lenovo thinkpad	167
21	macbook pro	247

Result 5

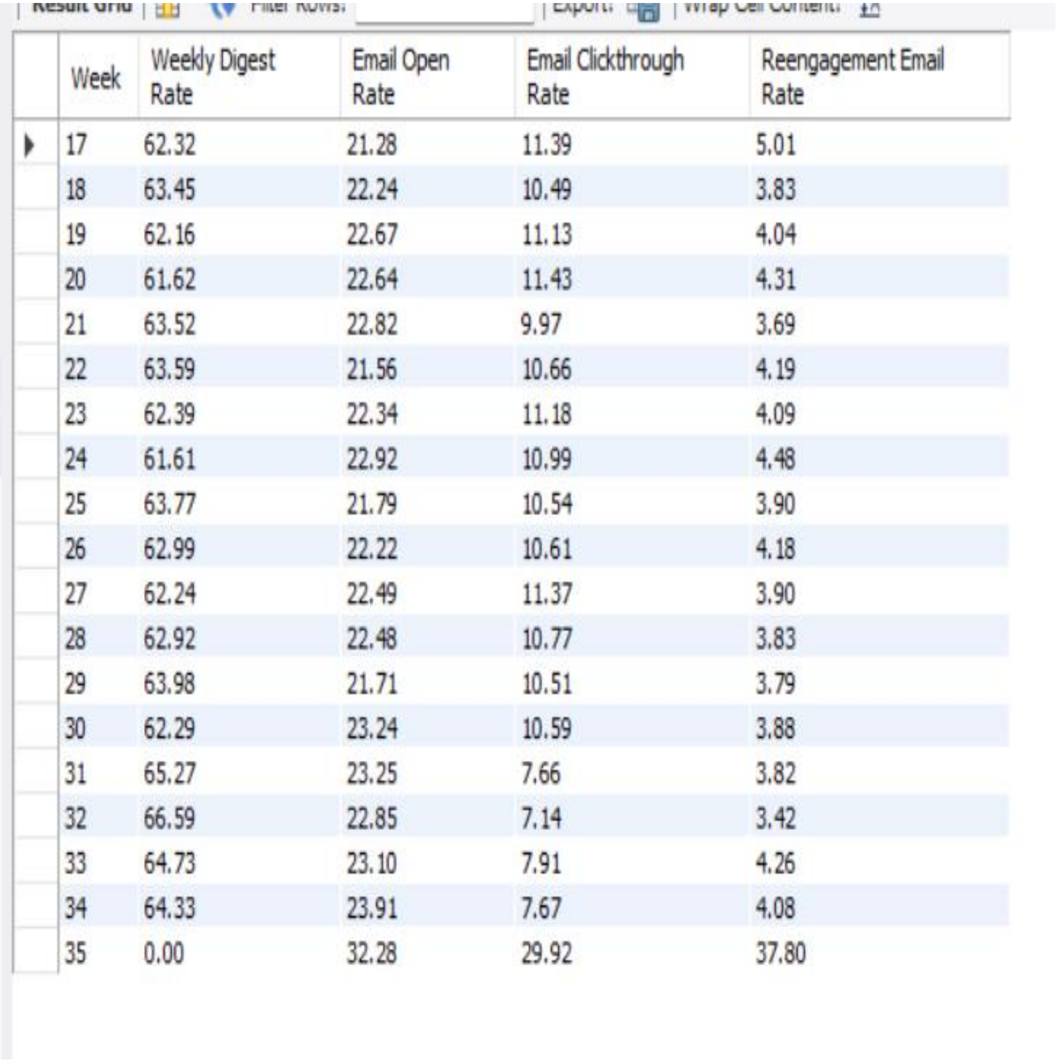
weeks	device	counts
34	lenovo thinkpad	193
34	macbook pro	292
35	acer aspire desktop	1
35	dell inspiron desktop	1
35	hp pavilion desktop	1
35	samsung galaxy note	1
35	htc one	2
35	ipad mini	2
35	iphone 5	2
35	mac mini	2
35	nexus 10	2
35	nexus 7	2
35	nokia lumia 635	2
35	acer aspire notebook	3
35	iphone 5s	3
35	kindle fire	3
35	windows surface	3
35	nexus 5	4
35	asus chromebook	6
35	iphone 4s	6
35	samsung galaxy s4	6
35	dell inspiron notebook	9
35	macbook air	10
35	lenovo thinkpad	16
35	macbook pro	17

E. Email Engagement Analysis:

- Objective: Analyze how users are engaging with the email service.
- Your Task: Write an SQL query to calculate the email engagement metrics.



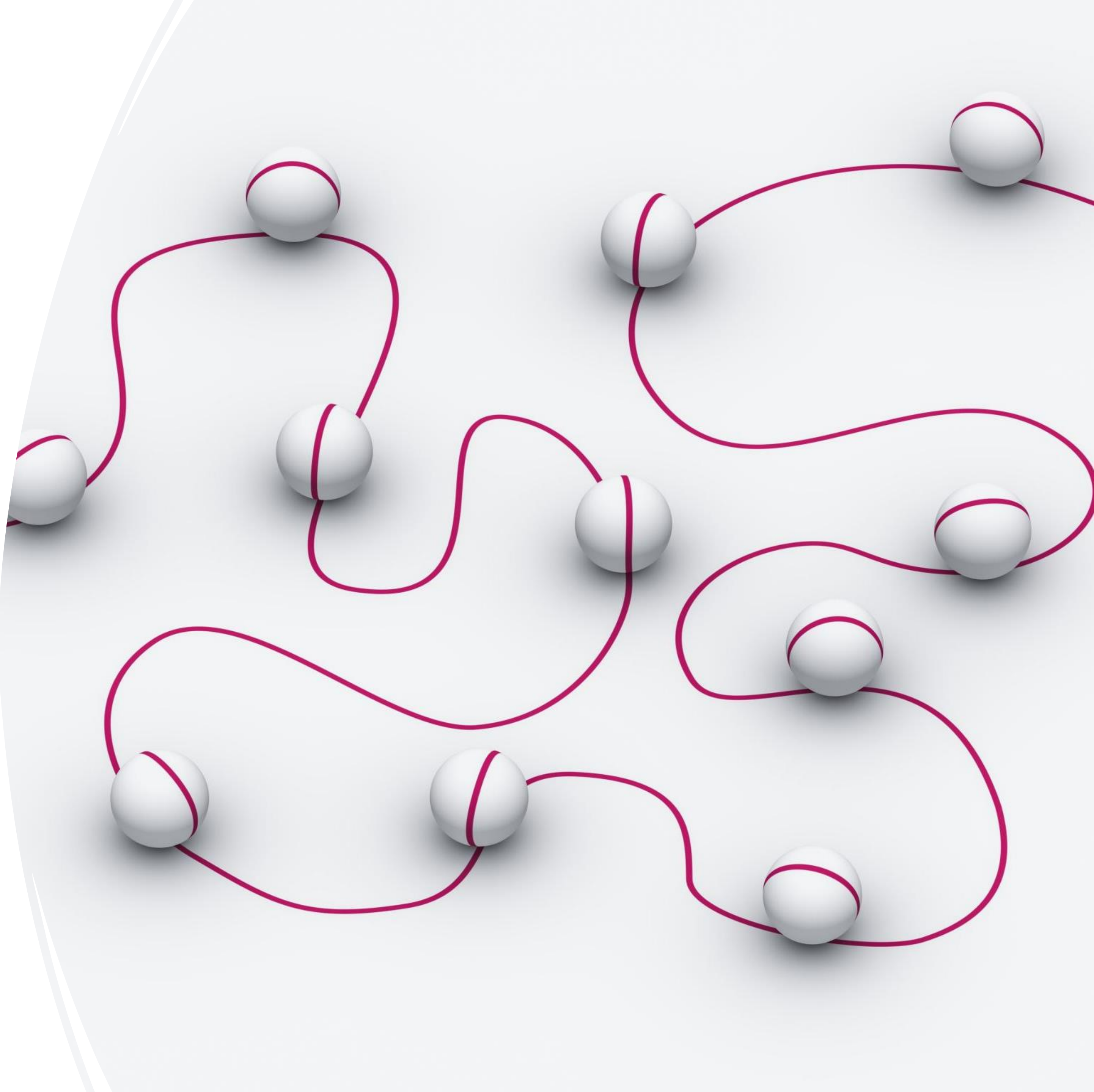
```
select * from email_events;

SELECT Week,
ROUND((weekly_digest/total*100),2) AS "Weekly Digest Rate",
ROUND((email_opens/total*100),2) AS "Email Open Rate",
ROUND((email_clickthroughs/total*100),2) AS "Email Clickthrough Rate",
ROUND((reengagement_emails/total* 100),2) AS "Reengagement Email Rate"
FROM
(
SELECT EXTRACT(WEEK FROM occured_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_opens,
COUNT(CASE WHEN action = 'email_clickthrough' THEN user_id ELSE NULL END) AS email_clickthroughs,
COUNT(CASE WHEN action = 'sent_reengagement_email' THEN user_id ELSE NULL END) AS reengagement_emails,
COUNT(user_id) AS total
FROM email_events GROUP BY 1
) sub
GROUP BY 1
ORDER BY 1;
```



Week	Weekly Digest Rate	Email Open Rate	Email Clickthrough Rate	Reengagement Email Rate
17	62.32	21.28	11.39	5.01
18	63.45	22.24	10.49	3.83
19	62.16	22.67	11.13	4.04
20	61.62	22.64	11.43	4.31
21	63.52	22.82	9.97	3.69
22	63.59	21.56	10.66	4.19
23	62.39	22.34	11.18	4.09
24	61.61	22.92	10.99	4.48
25	63.77	21.79	10.54	3.90
26	62.99	22.22	10.61	4.18
27	62.24	22.49	11.37	3.90
28	62.92	22.48	10.77	3.83
29	63.98	21.71	10.51	3.79
30	62.29	23.24	10.59	3.88
31	65.27	23.25	7.66	3.82
32	66.59	22.85	7.14	3.42
33	64.73	23.10	7.91	4.26
34	64.33	23.91	7.67	4.08
35	0.00	32.28	29.92	37.80

INSIGHTS:

- Through this project, we extracted valuable insights that can shape strategic decisions across multiple departments within the organization.
- Our journey into user analysis provided us with crucial understandings about user behavior, engagement patterns, and potential opportunities for growth.



INSIGHTS FOR JOB DATA ANALYSIS

Jobs Reviewed Over Time:

Analyzing the "Jobs Reviewed Over Time" dataset has uncovered intriguing patterns in job review activity during November 2020. The hourly breakdown revealed specific peak periods, aiding in optimized resource allocation and enhancing user engagement. Additionally, observing variations between weekdays and weekends offered insights into user behavior trends that can be leveraged for strategic notifications and scheduling adjustments.

Throughput Analysis:

By conducting the "Throughput Analysis" and calculating the 7-day rolling average, we gained a holistic perspective on event activity trends. This technique effectively smoothed out daily fluctuations, allowing for better trend identification and anomaly detection. This insight not only aids in operational planning by anticipating activity spikes but also serves as a robust monitoring tool for assessing the impact of campaigns or external factors on event throughput.

Insights from Project: Language Share Analysis:

Through the "Language Share Analysis," we've gained valuable insights into user language preferences over the past month. This data not only informs content localization strategies but also enables targeted content creation, optimizing user engagement and satisfaction. The analysis holds potential for market expansion as well, as shifts in language distribution could signify growth opportunities in specific regions.

Duplicate Rows Detection:

The "Duplicate Rows Detection" effort has enhanced data quality and reliability. By addressing duplicate entries within the dataset, we've fortified data integrity and eliminated potential sources of skewed analysis. This endeavor ensures accurate decision-making, aids in root cause analysis for data duplication, and streamlines database efficiency, marking a crucial step toward maintaining a robust and efficient data environment.

INSIGHTS FOR INVESTIGATING METRIC SPIKE

Weekly User Engagement:

Analyzing weekly engagement patterns reveals when users are most active, aiding in strategic resource allocation. Peaks denote optimal times for communication, while troughs might signal usability issues requiring attention.

User Growth Analysis:

Understanding user growth trends helps correlate successful strategies with spikes in acquisition. This knowledge informs resource allocation for future initiatives and ensures a sustained growth trajectory, guided by data-backed decisions.

Weekly Retention Analysis:

Tracking weekly retention based on sign-up cohorts uncovers user loyalty insights. Identifying retention drop-offs guides improvements in user experiences, solidifying product value and user satisfaction over time.

Weekly Engagement Per Device:

Device-specific engagement analysis facilitates tailored user experiences. By optimizing engagement on various platforms, the company enhances user satisfaction and ensures seamless interactions across all devices.

Email Engagement Analysis:

Examining email engagement metrics provides a clear view of user interaction with email campaigns. High open and click-through rates indicate compelling content and effective timing, leading to increased user engagement. Insights from this analysis enable the company to refine email strategies, delivering more relevant and impactful content that fosters stronger connections with users and drives desired actions.

RESULT

- Throughout this project, the role of Lead Data Analyst has been instrumental in driving data-driven decision-making within the organization, resembling the operations of a company like Microsoft. Through meticulous analysis of diverse datasets and rigorous application of advanced SQL techniques, this project has yielded actionable insights that transcend departmental boundaries

- Engaged in the role of a Lead Data Analyst, the delivered insights extend far beyond numerical figures. They form the bedrock of informed strategic choices, user-centric experience optimizations, and holistic comprehension of the company's operational landscape.

- As a result, this project stands as a testament to the potency of data analysis in driving improvements across a wide spectrum of company functions, much like the influential technology giants in the industry.

THANK YOU
BHAVYA SRI DUGGINA