Operation and Metric Analytics

By
Stan Pereira

Content

01 Project Description

02 Approach

03 Tech-Stack Used

04 Insights

05 Results

Project Description

Operational Analytics is a crucial process that involves analysing a company's end-to-end operations. This analysis helps identify areas for improvement within the company. As a Data Analyst, I will work closely with various teams, such as operations, support, and marketing, helping them derive valuable insights from the data they collect.

One of the key aspects of Operational Analytics is investigating metric spikes. This involves understanding and explaining sudden changes in key metrics, such as a dip in daily user engagement or a drop in sales. As a Data Analyst, I will need to answer these questions daily, making it crucial to understand how to investigate these metric spikes.

In this project, I will take on the role of a Lead Data Analyst at a company like Microsoft. I will be provided with various datasets and tables, and my task will be to derive insights from this data to answer questions posed by different departments within the company.

Project Description

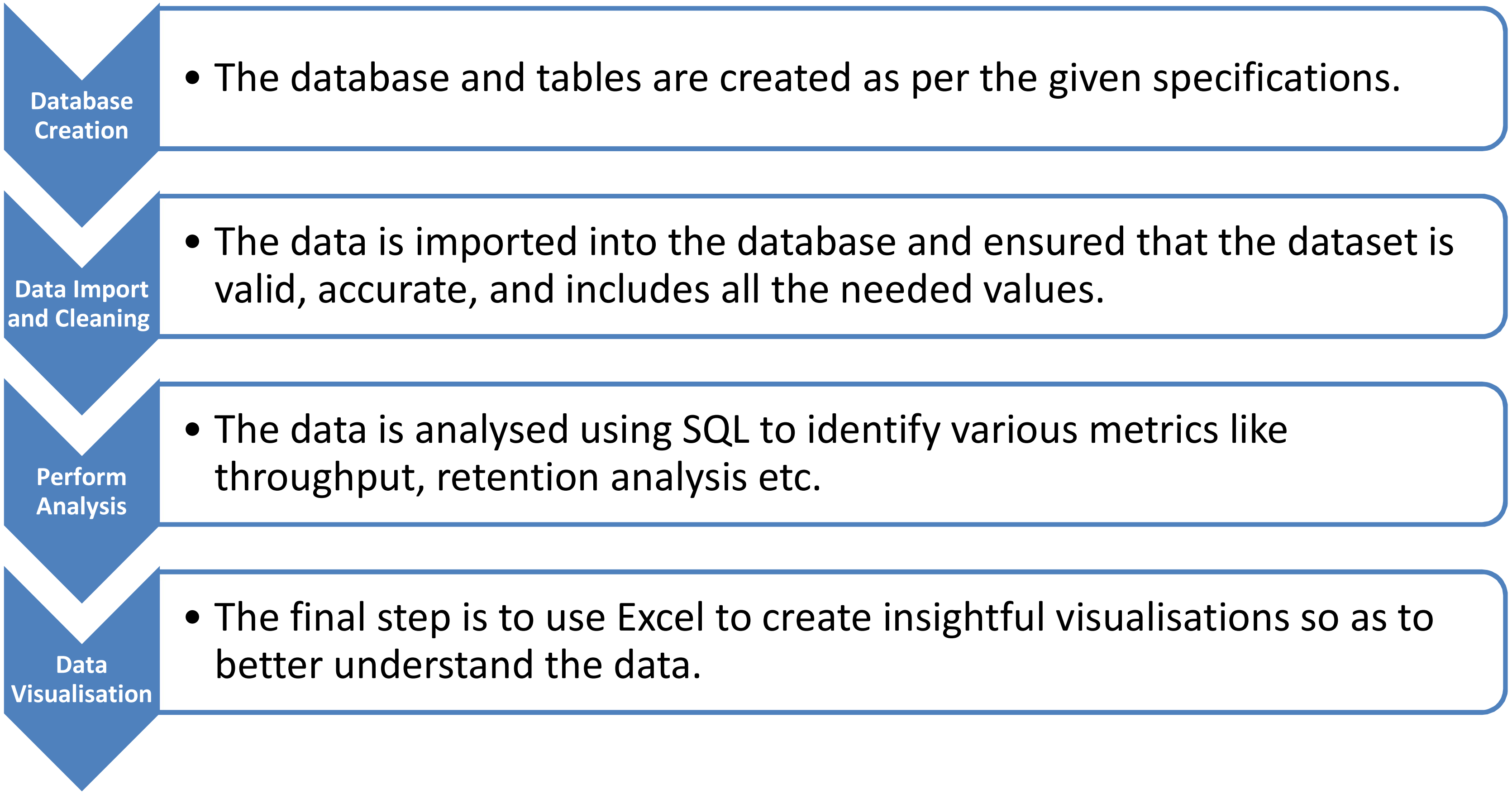
Case Study 1 (Operation Analytics)

- The first case involves an analysis of the job data to improve operational efficiency.
- Various metrics such as Throughput, Productivity, Percentage Share etc., are extracted from the provided data and recommendations for improvements are to be made

Case Study 2 (Metric Analytics)

- The second case study involves investigating data to draw better conclusions.
- Metrics to identify patterns and trends such as User Growth, User Engagement, Cohort Retention Analysis and Email Engagement Metrics are extracted and used to find the best ways to improve productivity.

Approach



Database Creation

- The database and tables are created as per the given specifications.

Data Import and Cleaning

- The data is imported into the database and ensured that the dataset is valid, accurate, and includes all the needed values.

Perform Analysis

- The data is analysed using SQL to identify various metrics like throughput, retention analysis etc.

Data Visualisation

- The final step is to use Excel to create insightful visualisations so as to better understand the data.

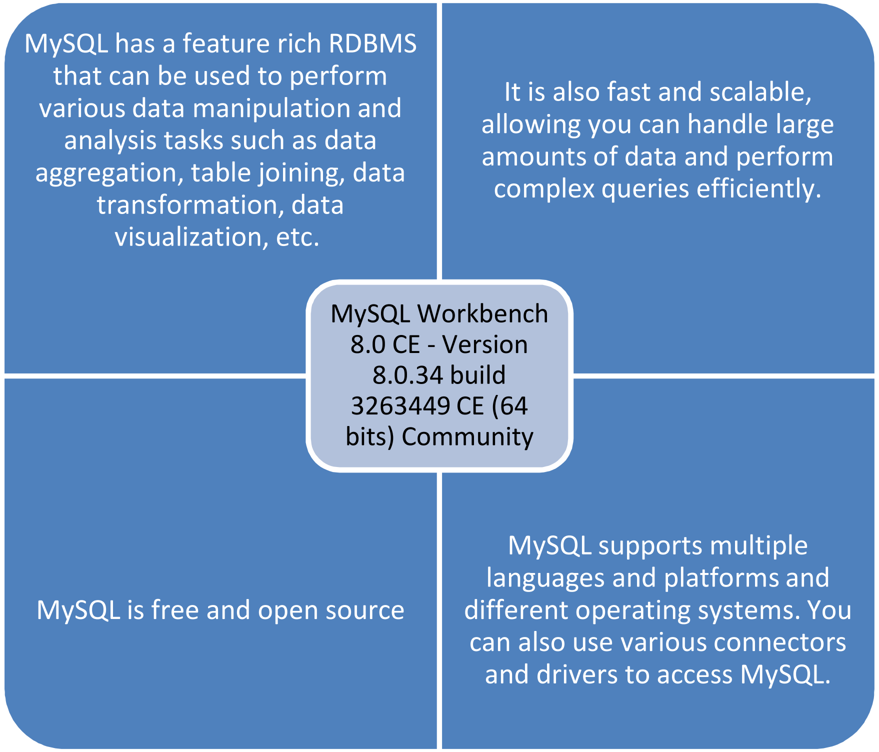
Approach

Notes:

- While creating the Job_Data Table, the datatype specified for the column 'ds' was ***varchar***, even though the better datatype would have been ***date***.
- As part of the data cleaning process, the csv file had the 'ds' column in the ***text*** format MM/DD/YYYY. But the required format was YYYY/MM/DD, and hence needed to be converted using SQL statements.
- To perform the analysis and ascertain insights, there was a need to research on functions/terms in SQL and Metrics like ***CAST*** function, ***LOAD DATA*** statement, ***ROWS BETWEEN***, ***Throughput***, ***Rolling Average***, ***Cohort Analysis***, ***Engagement metrics*** etc.

Tech Stack Used

The Softwares and their Version Utilized



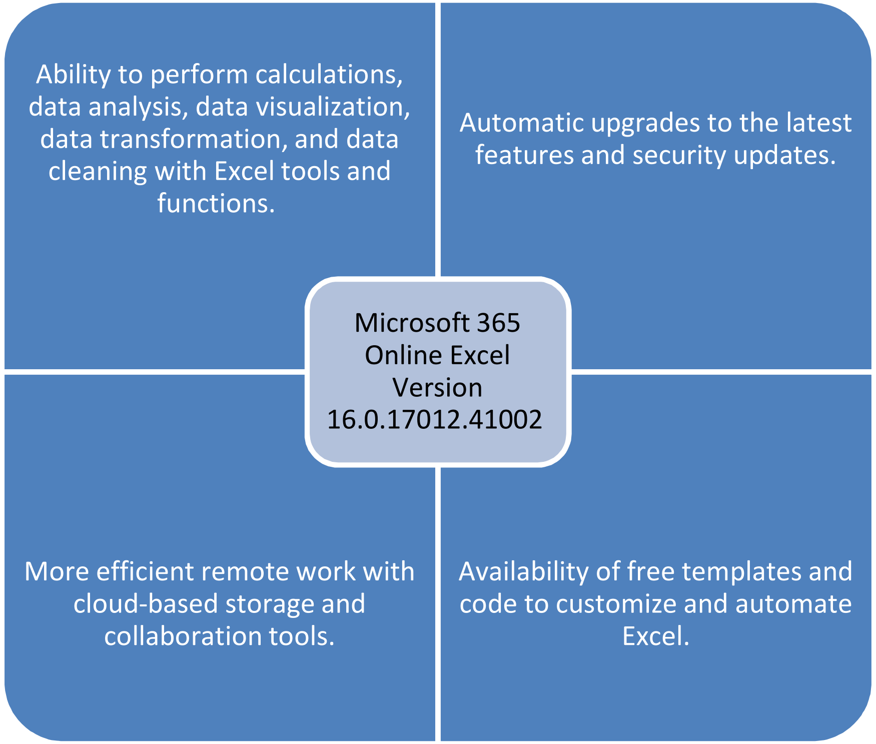
MySQL Workbench 8.0 CE - Version 8.0.34 build 3263449 CE (64 bits) Community

MySQL has a feature rich RDBMS that can be used to perform various data manipulation and analysis tasks such as data aggregation, table joining, data transformation, data visualization, etc.

It is also fast and scalable, allowing you can handle large amounts of data and perform complex queries efficiently.

MySQL is free and open source

MySQL supports multiple languages and platforms and different operating systems. You can also use various connectors and drivers to access MySQL.



Microsoft 365 Online Excel Version 16.0.17012.41002

Ability to perform calculations, data analysis, data visualization, data transformation, and data cleaning with Excel tools and functions.

Automatic upgrades to the latest features and security updates.

More efficient remote work with cloud-based storage and collaboration tools.

Availability of free templates and code to customize and automate Excel.

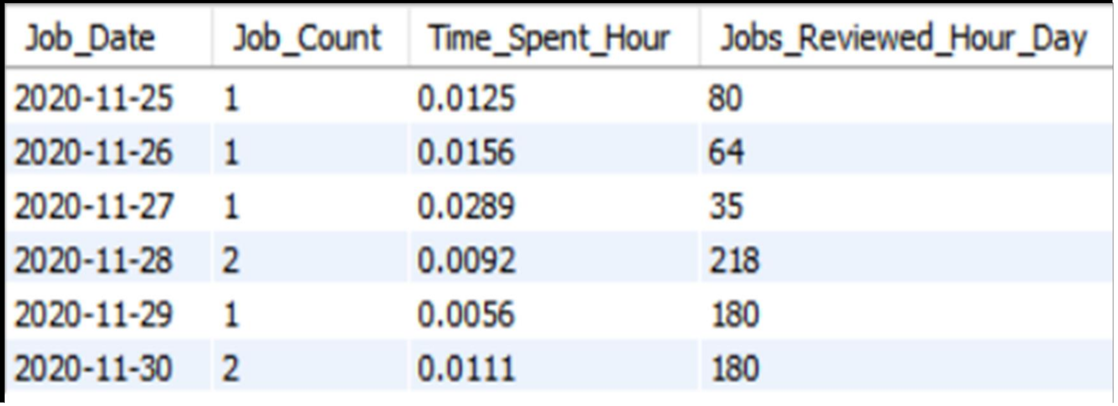
Case Study 1

Question 1 - SQL Statement

01) Jobs Reviewed Over Time: Calculate the number of jobs reviewed per hour for each day in November 2020.

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

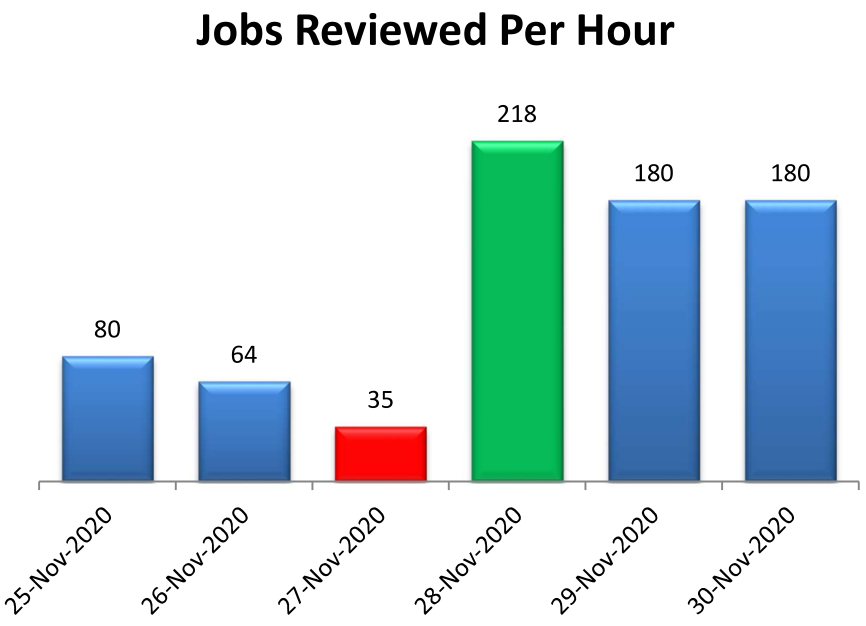
```
SELECT
    CAST(ds AS DATE) AS Job_Date,
    COUNT(job_id) AS Job_Count,
    SUM(time_spent)/3600 AS Time_Spent_Hour,
    ROUND(COUNT(job_id)/(SUM(time_spent)/3600)) AS Jobs_Reviewed_Hour_Day
FROM
    job_data
WHERE
    CAST(ds AS DATE) >= '2020/11/01' AND CAST(ds AS DATE) <= '2020/11/30'
GROUP BY Job_Date
ORDER BY Job_Date;
```



Job_Date	Job_Count	Time_Spent_Hour	Jobs_Reviewed_Hour_Day
2020-11-25	1	0.0125	80
2020-11-26	1	0.0156	64
2020-11-27	1	0.0289	35
2020-11-28	2	0.0092	218
2020-11-29	1	0.0056	180
2020-11-30	2	0.0111	180

Question 1 - Insights

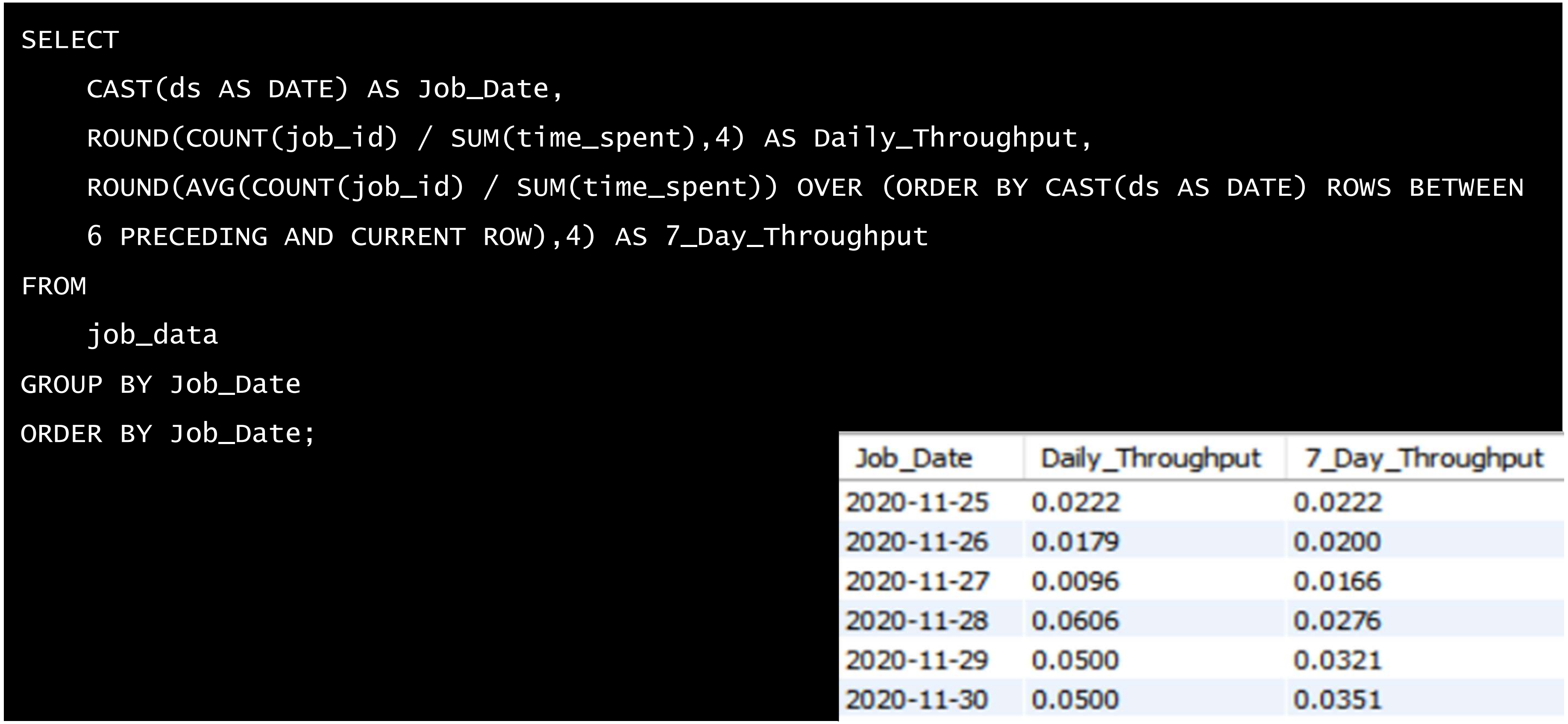
- 28th November 2020 had the highest number of jobs reviewed per hour at 218
- The lowest number of jobs reviewed per hour was on 27th November 2020 at 35.



Question 2 - SQL Statement

02) Throughput Analysis: 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.



```
SELECT
    CAST(ds AS DATE) AS Job_Date,
    ROUND(COUNT(job_id) / SUM(time_spent),4) AS Daily_Throughput,
    ROUND(AVG(COUNT(job_id) / SUM(time_spent)) OVER (ORDER BY CAST(ds AS DATE) ROWS BETWEEN
    6 PRECEDING AND CURRENT ROW),4) AS 7_Day_Throughput
FROM
    job_data
GROUP BY Job_Date
ORDER BY Job_Date;
```

Job_Date	Daily_Throughput	7_Day_Throughput
2020-11-25	0.0222	0.0222
2020-11-26	0.0179	0.0200
2020-11-27	0.0096	0.0166
2020-11-28	0.0606	0.0276
2020-11-29	0.0500	0.0321
2020-11-30	0.0500	0.0351

Question 2 - Insights

What is Throughput

Throughput analysis is a method of measuring the efficiency of a business process by calculating the rate at which units move through the process from start to finish. It can help a business make decisions that minimize costs and maximize profits. Throughput analysis involves two components: inventory and flow time. Inventory is the number of units that are involved in the process at a given time, and flow time is the amount of time a unit spends in the process from start to finish. The formula for throughput rate is:

$$R=I/T$$

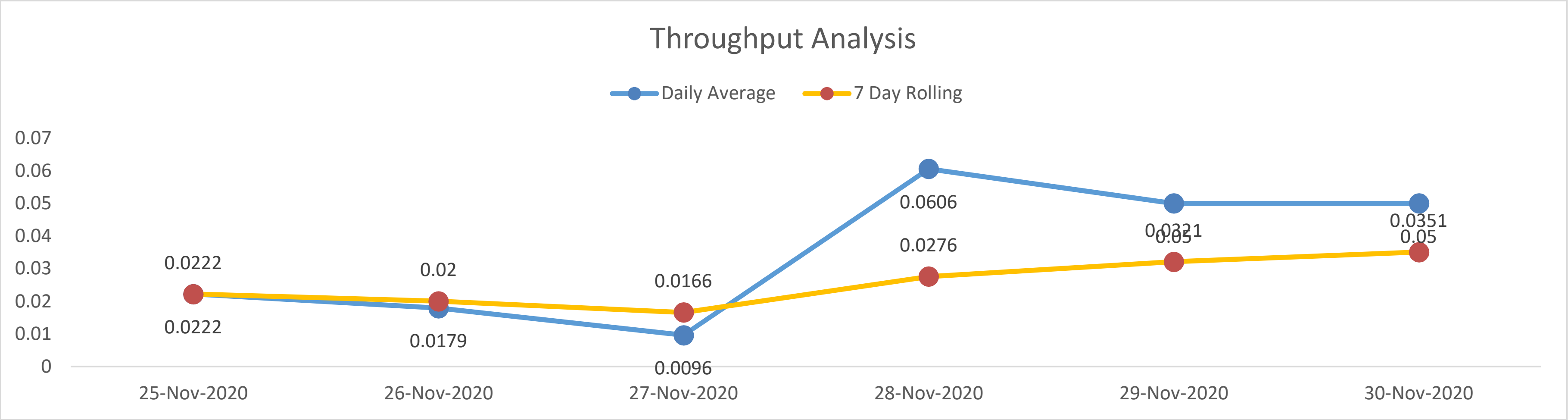
7-day rolling average v/s daily metric. Which is better in this situation?

A rolling average, sometimes referred to as a moving average, is a metric that calculates trends over short periods of time using a set of data. Specifically, it helps calculate trends when they might otherwise be difficult to detect.

An average/daily metric fluctuates wildly as each day can result in a different calculation. The rolling average creates a trend that can show a better picture over a long time.

In this situation, a 7-day rolling metric makes more sense, due to the fluctuation of the daily metric.

Question 2 - Insights



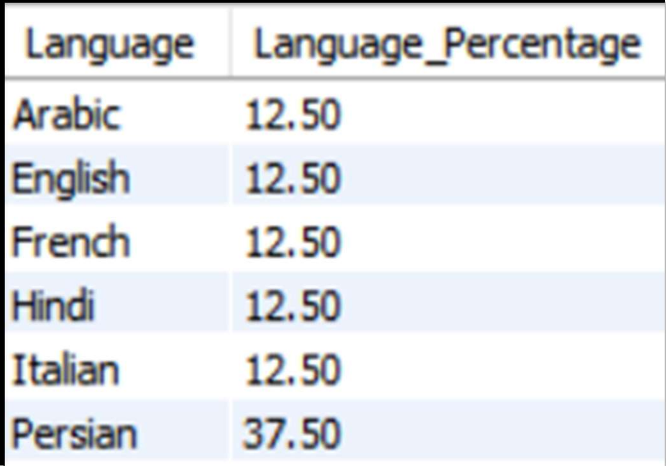
- The graph shows the daily average and the 7 day rolling average
- As observed via the graph, the rolling average is more consistent and less fluctuating. This will help to understand the trend better.

Question 3 - SQL Statement

03) Language Share Analysis: 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.

```
SELECT
     DISTINCT language AS Language,
     ROUND(((COUNT(language) OVER (PARTITION BY language)) / (COUNT(language) OVER ()) *
100),2) AS Language_Percentage
FROM job_data
WHERE
    CAST(ds AS DATE) >= '2020/11/01'
        AND CAST(ds AS DATE) <= '2020/11/30';
```

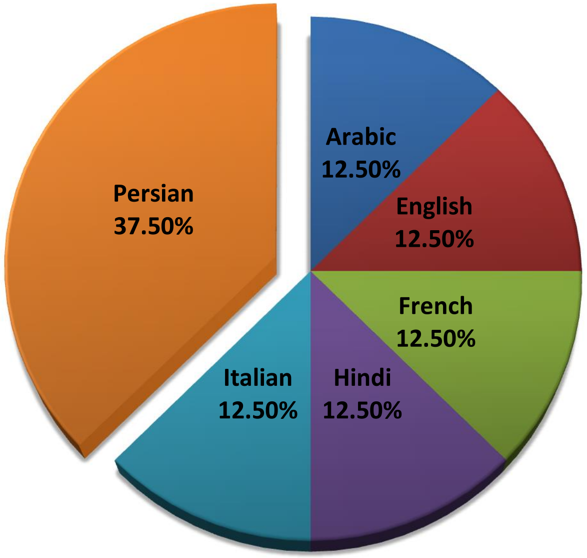


Language	Language_Percentage
Arabic	12.50
English	12.50
French	12.50
Hindi	12.50
Italian	12.50
Persian	37.50

Question 3 - Insights

- The Pie Chart shows the language percentage of jobs reviewed.
- As per the chart, **Persian** Language jobs are reviewed the most with a share of **37.50%**
- All the other languages have the same percentage share of **12.5%**

Language Percentage

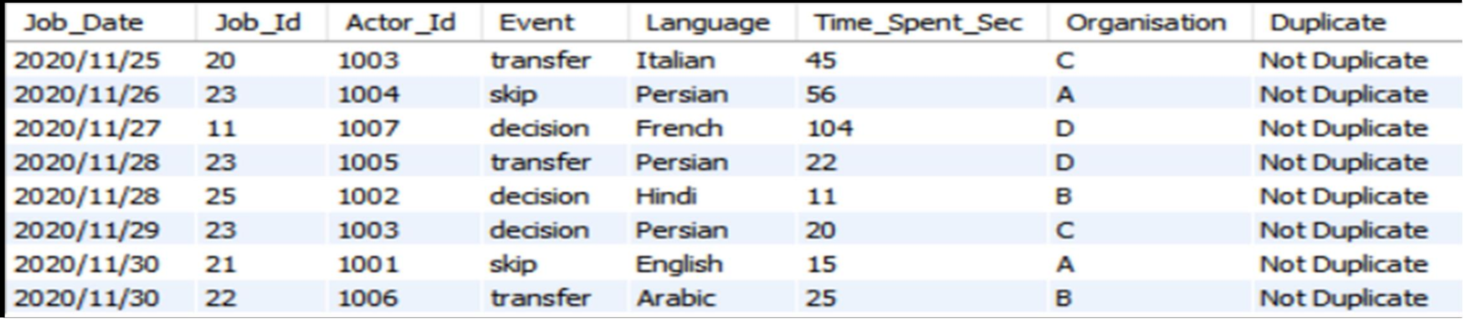


Question 4 - SQL Statement

04) Duplicate Rows Detection: Identify duplicate rows in the data.

Your Task: Write an SQL query to display duplicate rows from the job_data table.

```
SELECT ds AS Job_Date, job_id AS Job_Id, actor_id AS Actor_Id, event AS Event,
       language AS Language, time_spent AS Time_Spent_Sec, org AS Organisation,
       IF (Job_Count>1,"Duplicate","Not Duplicate") AS Duplicate
FROM
       (SELECT *,
               COUNT(*) OVER (PARTITION BY ds, job_id, actor_id, event, language,time_spent,
  org) AS Job_Count
      FROM job_data) AS Tot_Count
ORDER BY Job_Date, Duplicate;
```



Job_Date	Job_Id	Actor_Id	Event	Language	Time_Spent_Sec	Organisation	Duplicate
2020/11/25	20	1003	transfer	Italian	45	C	Not Duplicate
2020/11/26	23	1004	skip	Persian	56	A	Not Duplicate
2020/11/27	11	1007	decision	French	104	D	Not Duplicate
2020/11/28	23	1005	transfer	Persian	22	D	Not Duplicate
2020/11/28	25	1002	decision	Hindi	11	B	Not Duplicate
2020/11/29	23	1003	decision	Persian	20	C	Not Duplicate
2020/11/30	21	1001	skip	English	15	A	Not Duplicate
2020/11/30	22	1006	transfer	Arabic	25	B	Not Duplicate

Question 4 - Insights

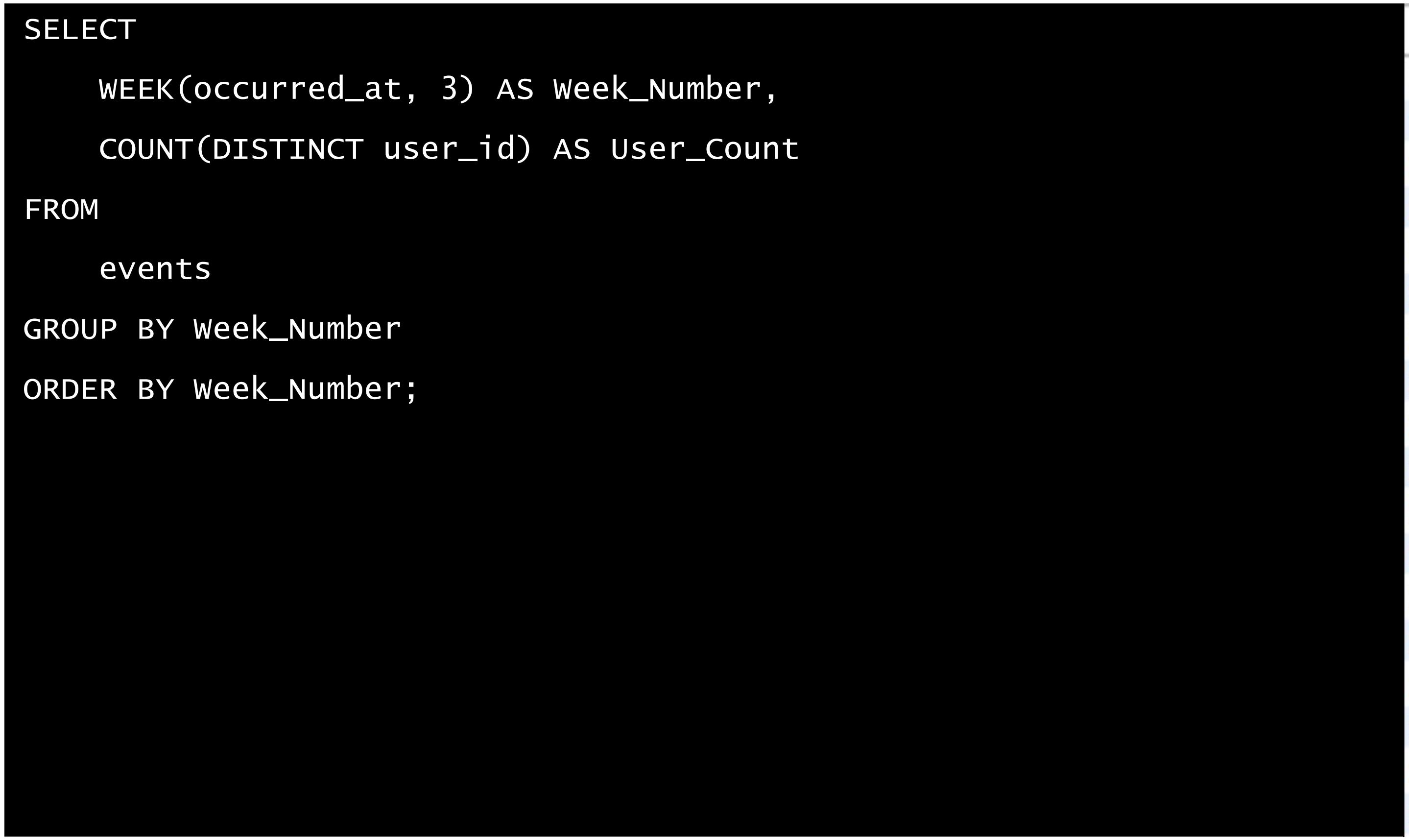
- None of the fields are duplicates.

Note : All the columns in the table can be repeated and can have duplicates, so we need to assign a combination of two or more columns to allow for duplicate checks. For this project, I used a combination of all fields for the duplicate check (even though, this is not an ideal check). If a time field was present, it could be used as a primary key).

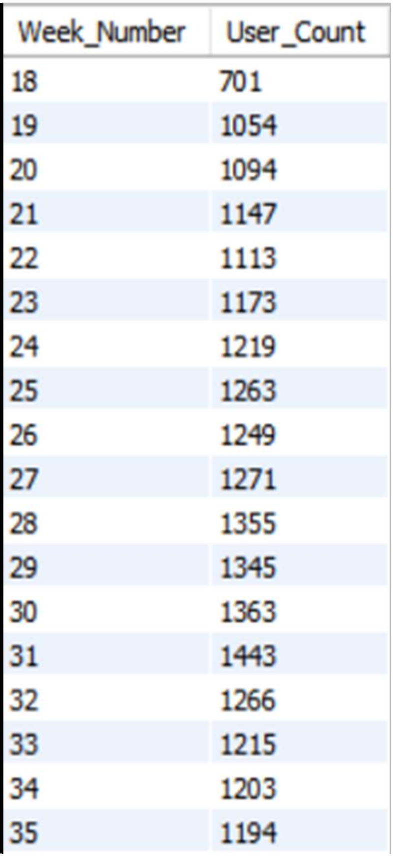
Case Study 2

Question 1 - SQL Statement

01) Weekly User Engagement: Measure the activeness of users on a weekly basis.
Your Task: Write an SQL query to calculate the weekly user engagement.

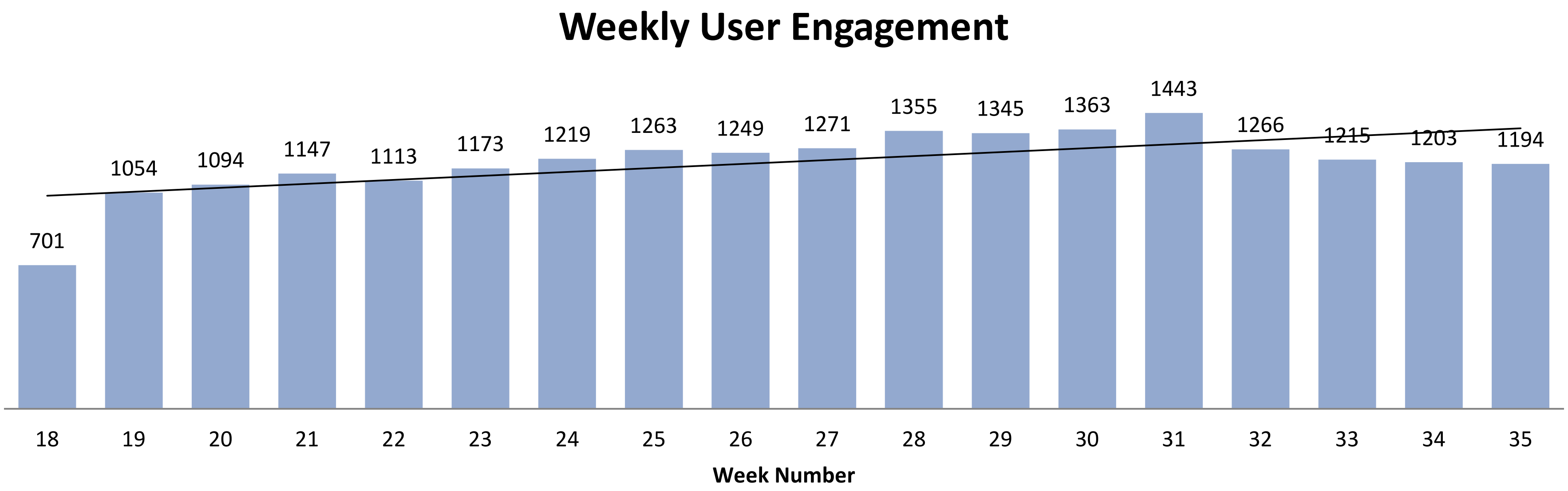


```
SELECT
    WEEK(occurred_at, 3) AS Week_Number,
    COUNT(DISTINCT user_id) AS User_Count
FROM
    events
GROUP BY Week_Number
ORDER BY Week_Number;
```



Week_Number	User_Count
18	701
19	1054
20	1094
21	1147
22	1113
23	1173
24	1219
25	1263
26	1249
27	1271
28	1355
29	1345
30	1363
31	1443
32	1266
33	1215
34	1203
35	1194

Question 1 - Insights



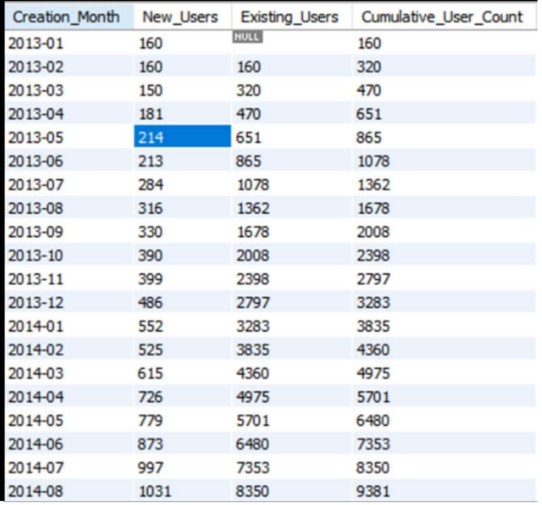
- The highest weekly user engagement was in **Week 31** at **1443** users
- There is a slight upward trend in user engagement over the weeks
- The lowest weekly user engagement was in **Week 19** at **1054** users. *(Week 18 is not taken into account as it does not have all days)*

Question 2 - SQL Statement

02) User Growth Analysis: Analyse the growth of users over time for a product.

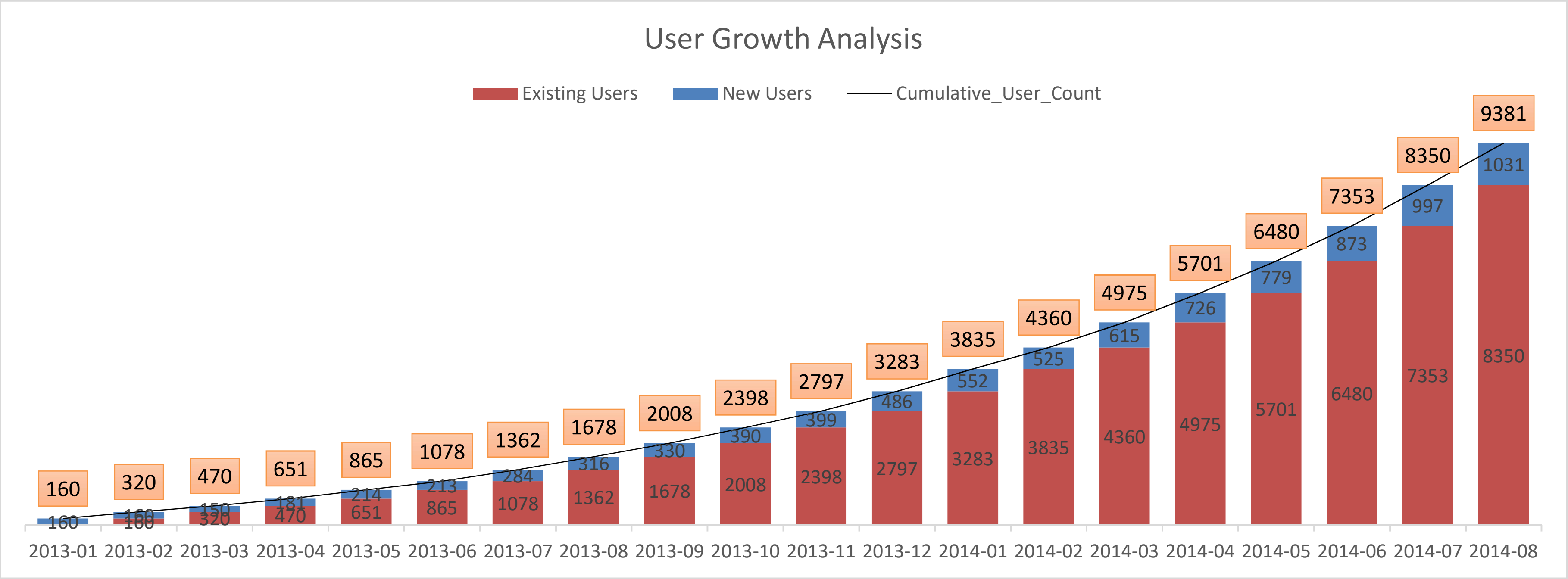
Your Task: Write an SQL query to calculate the user growth for the product.

```
SELECT
    DATE_FORMAT(created_at,'%Y-%m') AS Creation_Month,
   COUNT(user_id) AS New_Users,
   SUM(COUNT(user_id)) OVER (ORDER BY DATE_FORMAT(created_at,'%Y-%m') ROWS BETWEEN UNBOUNDED
    PRECEDING AND 1 PRECEDING) AS Existing_Users,
   SUM(COUNT(user_id)) OVER (ORDER BY DATE_FORMAT(created_at,'%Y-%m') ROWS BETWEEN UNBOUNDED
    PRECEDING AND CURRENT ROW) AS Cumulative_User_Count
FROM users
GROUP BY Creation_Month;
```



Creation_Month	New_Users	Existing_Users	Cumulative_User_Count
2013-01	160	NULL	160
2013-02	160	160	320
2013-03	150	320	470
2013-04	181	470	651
2013-05	214	651	865
2013-06	213	865	1078
2013-07	284	1078	1362
2013-08	316	1362	1678
2013-09	330	1678	2008
2013-10	390	2008	2398
2013-11	399	2398	2797
2013-12	486	2797	3283
2014-01	552	3283	3835
2014-02	525	3835	4360
2014-03	615	4360	4975
2014-04	726	4975	5701
2014-05	779	5701	6480
2014-06	873	6480	7353
2014-07	997	7353	8350
2014-08	1031	8350	9381

Question 2 - Insights

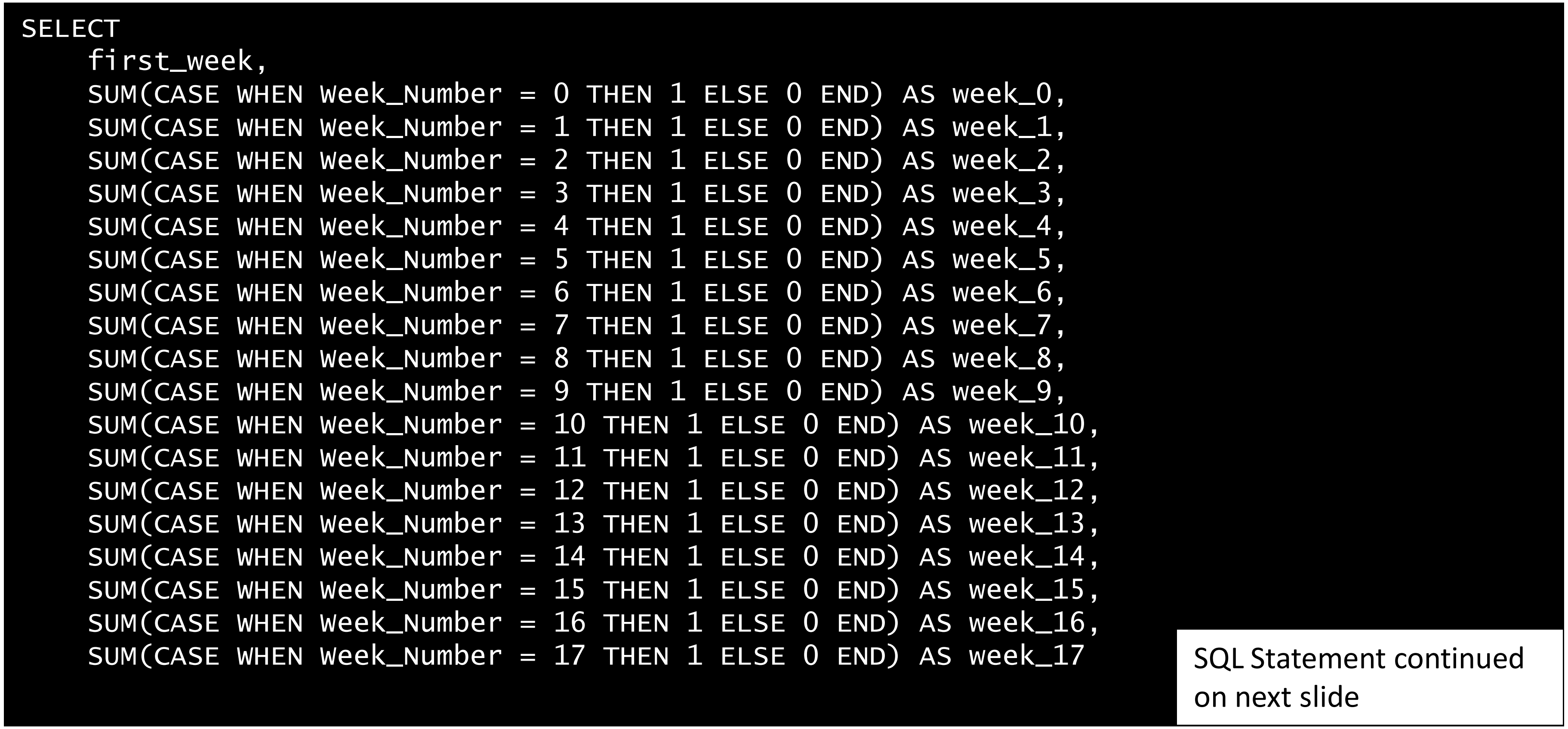


- We can notice a steady upward **increase** of **weekly user registrations**.
- The total number of users are **9381**.

Question 3 - SQL Statement

03) Weekly Retention Analysis: Analyse 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.

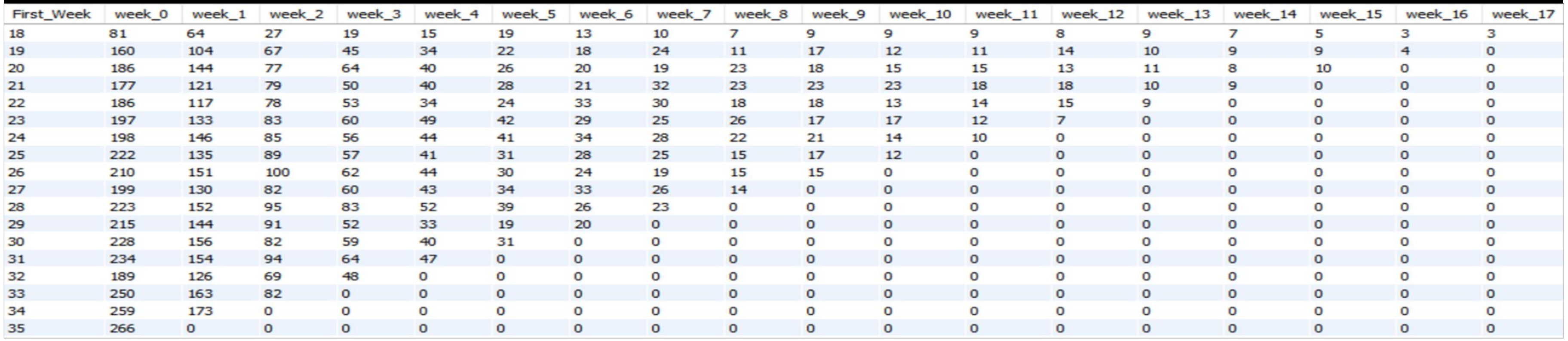


```
SELECT
    first_week,
    SUM(CASE WHEN Week_Number = 0 THEN 1 ELSE 0 END) AS week_0,
    SUM(CASE WHEN Week_Number = 1 THEN 1 ELSE 0 END) AS week_1,
    SUM(CASE WHEN Week_Number = 2 THEN 1 ELSE 0 END) AS week_2,
    SUM(CASE WHEN Week_Number = 3 THEN 1 ELSE 0 END) AS week_3,
    SUM(CASE WHEN 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
```

SQL Statement continued on next slide

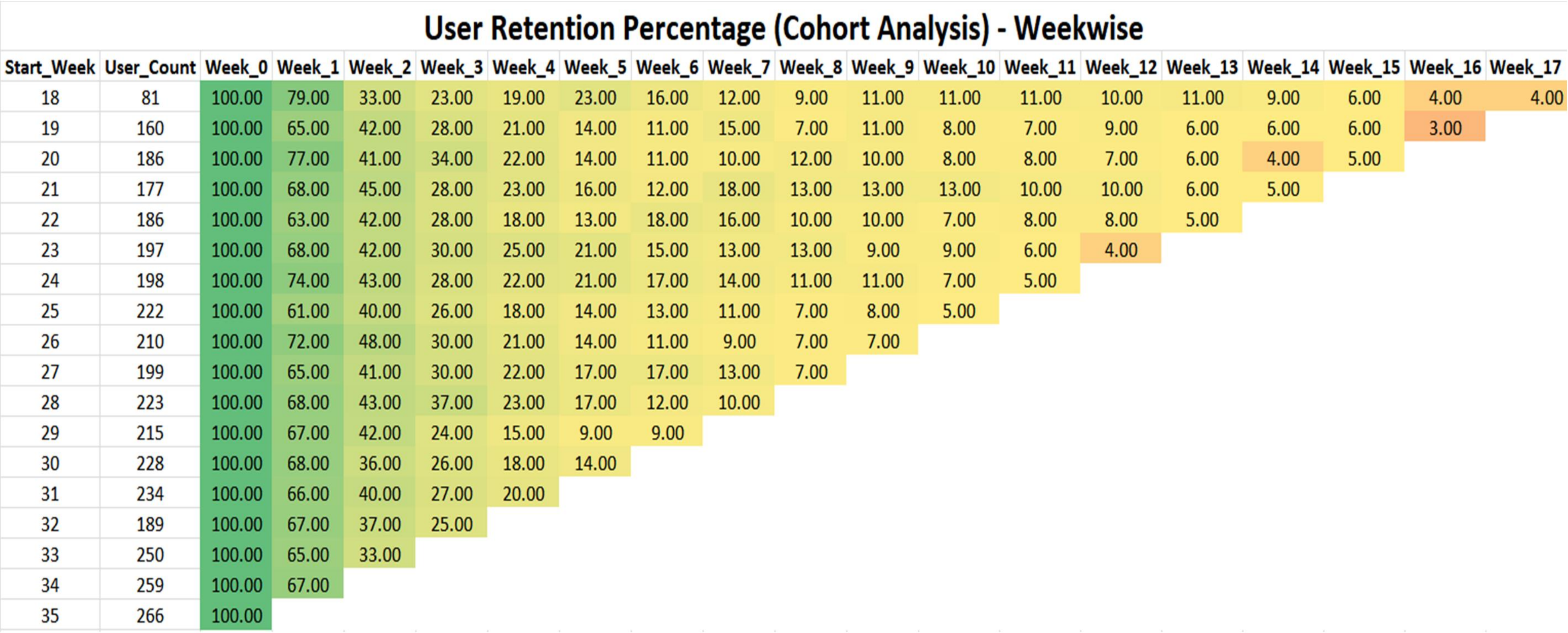
Question 3 - SQL Statement (Continued)

```
FROM
   (SELECT b.user_id, Week_Occurred, First_Week, (Week_Occurred-First_Week) AS Week_Number
    FROM
           (SELECT user_id, week(occurred_at,3) AS Week_Occurred
            FROM events
            GROUP BY user_id, Week_Occurred
            ORDER BY user_id, Week_Occurred) AS a
    RIGHT JOIN
           (SELECT user_id, min(week(occurred_at,3)) AS First_Week
            FROM events
            WHERE event_type='signup_flow'
            GROUP BY user_id
            ORDER BY user_id) AS b
    ON a.user_id = b.user_id) AS Week_Number_Select
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	week_11	week_12	week_13	week_14	week_15	week_16	week_17
18	81	64	27	19	15	19	13	10	7	9	9	9	8	9	7	5	3	3
19	160	104	67	45	34	22	18	24	11	17	12	11	14	10	9	9	4	0
20	186	144	77	64	40	26	20	19	23	18	15	15	13	11	8	10	0	0
21	177	121	79	50	40	28	21	32	23	23	23	18	18	10	9	0	0	0
22	186	117	78	53	34	24	33	30	18	18	13	14	15	9	0	0	0	0
23	197	133	83	60	49	42	29	25	26	17	17	12	7	0	0	0	0	0
24	198	146	85	56	44	41	34	28	22	21	14	10	0	0	0	0	0	0
25	222	135	89	57	41	31	28	25	15	17	12	0	0	0	0	0	0	0
26	210	151	100	62	44	30	24	19	15	15	0	0	0	0	0	0	0	0
27	199	130	82	60	43	34	33	26	14	0	0	0	0	0	0	0	0	0
28	223	152	95	83	52	39	26	23	0	0	0	0	0	0	0	0	0	0
29	215	144	91	52	33	19	20	0	0	0	0	0	0	0	0	0	0	0
30	228	156	82	59	40	31	0	0	0	0	0	0	0	0	0	0	0	0
31	234	154	94	64	47	0	0	0	0	0	0	0	0	0	0	0	0	0
32	189	126	69	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	250	163	82	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	259	173	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	266	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Question 3 - Insights



User Retention Percentage (Cohort Analysis) - Weekwise

Start_Week	User_Count	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
18	81	100.00	79.00	33.00	23.00	19.00	23.00	16.00	12.00	9.00	11.00	11.00	11.00	10.00	11.00	9.00	6.00	4.00	4.00
19	160	100.00	65.00	42.00	28.00	21.00	14.00	11.00	15.00	7.00	11.00	8.00	7.00	9.00	6.00	6.00	6.00	3.00	
20	186	100.00	77.00	41.00	34.00	22.00	14.00	11.00	10.00	12.00	10.00	8.00	8.00	7.00	6.00	4.00	5.00		
21	177	100.00	68.00	45.00	28.00	23.00	16.00	12.00	18.00	13.00	13.00	13.00	10.00	10.00	6.00	5.00			
22	186	100.00	63.00	42.00	28.00	18.00	13.00	18.00	16.00	10.00	10.00	7.00	8.00	8.00	5.00				
23	197	100.00	68.00	42.00	30.00	25.00	21.00	15.00	13.00	13.00	9.00	9.00	6.00	4.00					
24	198	100.00	74.00	43.00	28.00	22.00	21.00	17.00	14.00	11.00	11.00	7.00	5.00						
25	222	100.00	61.00	40.00	26.00	18.00	14.00	13.00	11.00	7.00	8.00	5.00							
26	210	100.00	72.00	48.00	30.00	21.00	14.00	11.00	9.00	7.00	7.00								
27	199	100.00	65.00	41.00	30.00	22.00	17.00	17.00	13.00	7.00									
28	223	100.00	68.00	43.00	37.00	23.00	17.00	12.00	10.00										
29	215	100.00	67.00	42.00	24.00	15.00	9.00	9.00											
30	228	100.00	68.00	36.00	26.00	18.00	14.00												
31	234	100.00	66.00	40.00	27.00	20.00													
32	189	100.00	67.00	37.00	25.00														
33	250	100.00	65.00	33.00															
34	259	100.00	67.00																
35	266	100.00																	

- The chart above shows the User Retention Percentage (Signup Cohort) - Weekwise.

Question 4 - SQL Statement

04) Weekly Engagement Per Device: Measure the activeness of users on a weekly basis per device.

Your Task: Write an SQL query to calculate the weekly engagement per device.

```
SELECT
   Device,
   SUM(CASE WHEN Week_Number = 18 THEN Tot_Count ELSE 0 END) AS Week_18,
   SUM(CASE WHEN Week_Number = 19 THEN Tot_Count ELSE 0 END) AS Week_19,
   SUM(CASE WHEN Week_Number = 20 THEN Tot_Count ELSE 0 END) AS Week_20,
   SUM(CASE WHEN Week_Number = 21 THEN Tot_Count ELSE 0 END) AS Week_21,
   SUM(CASE WHEN Week_Number = 22 THEN Tot_Count ELSE 0 END) AS Week_22,
   SUM(CASE WHEN Week_Number = 23 THEN Tot_Count ELSE 0 END) AS Week_23,
   SUM(CASE WHEN Week_Number = 24 THEN Tot_Count ELSE 0 END) AS Week_24,
   SUM(CASE WHEN Week_Number = 25 THEN Tot_Count ELSE 0 END) AS Week_25,
   SUM(CASE WHEN Week_Number = 26 THEN Tot_Count ELSE 0 END) AS Week_26,
   SUM(CASE WHEN Week_Number = 27 THEN Tot_Count ELSE 0 END) AS Week_27,
   SUM(CASE WHEN Week_Number = 28 THEN Tot_Count ELSE 0 END) AS Week_28,
   SUM(CASE WHEN Week_Number = 29 THEN Tot_Count ELSE 0 END) AS Week_29,
   SUM(CASE WHEN Week_Number = 30 THEN Tot_Count ELSE 0 END) AS Week_30,
   SUM(CASE WHEN Week_Number = 31 THEN Tot_Count ELSE 0 END) AS Week_31,
   SUM(CASE WHEN Week_Number = 32 THEN Tot_Count ELSE 0 END) AS Week_32,
   SUM(CASE WHEN Week_Number = 33 THEN Tot_Count ELSE 0 END) AS Week_33,
   SUM(CASE WHEN Week_Number = 34 THEN Tot_Count ELSE 0 END) AS Week_34,
   SUM(CASE WHEN Week_Number = 35 THEN Tot_Count ELSE 0 END) AS Week_35,
   SUM(Tot_Count) AS Total_Count
```

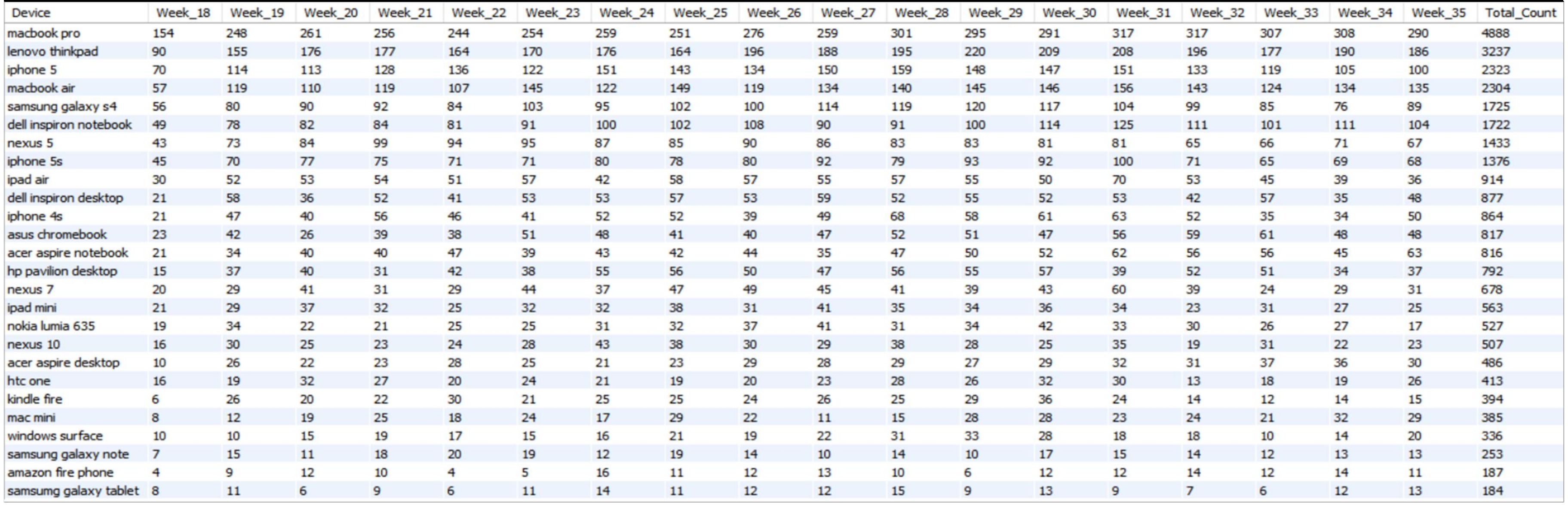
SQL Statement
continued on
next slide

Question 4 - SQL Statement (Continued)

04) Weekly Engagement Per Device: Measure the activeness of users on a weekly basis per device.

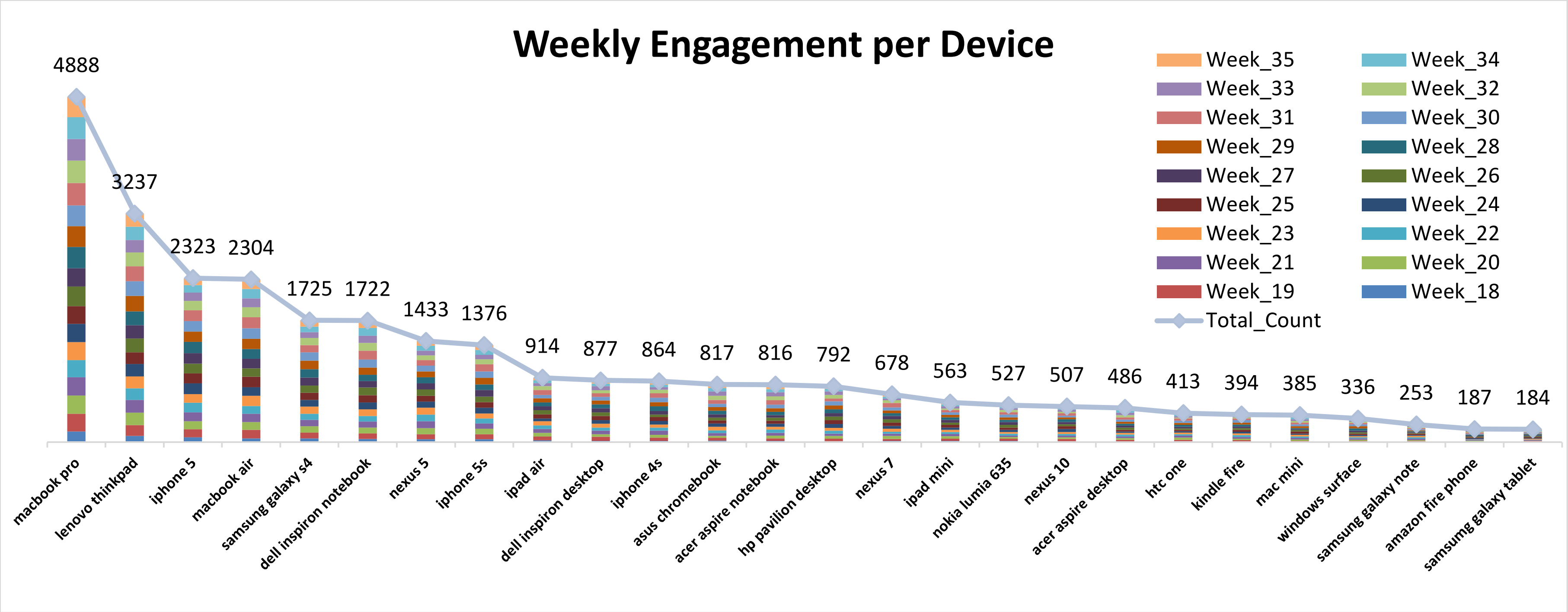
Your Task: Write an SQL query to calculate the weekly engagement per device.

```
FROM
   (SELECT WEEK(occurred_at,3) AS Week_Number,device AS Device,COUNT(DISTINCT user_id) AS Tot_Count
   FROM events WHERE event_type='engagement‘
   GROUP BY user_id, Device, Week_Number
   ORDER BY user_id, device) AS Query
GROUP BY Device
ORDER BY Total_Count DESC;
```



Device	Week_18	Week_19	Week_20	Week_21	Week_22	Week_23	Week_24	Week_25	Week_26	Week_27	Week_28	Week_29	Week_30	Week_31	Week_32	Week_33	Week_34	Week_35	Total_Count
macbook pro	154	248	261	256	244	254	259	251	276	259	301	295	291	317	317	307	308	290	4888
lenovo thinkpad	90	155	176	177	164	170	176	164	196	188	195	220	209	208	196	177	190	186	3237
iphone 5	70	114	113	128	136	122	151	143	134	150	159	148	147	151	133	119	105	100	2323
macbook air	57	119	110	119	107	145	122	149	119	134	140	145	146	156	143	124	134	135	2304
samsung galaxy s4	56	80	90	92	84	103	95	102	100	114	119	120	117	104	99	85	76	89	1725
dell inspiron notebook	49	78	82	84	81	91	100	102	108	90	91	100	114	125	111	101	111	104	1722
nexus 5	43	73	84	99	94	95	87	85	90	86	83	83	81	81	65	66	71	67	1433
iphone 5s	45	70	77	75	71	71	80	78	80	92	79	93	92	100	71	65	69	68	1376
ipad air	30	52	53	54	51	57	42	58	57	55	57	55	50	70	53	45	39	36	914
dell inspiron desktop	21	58	36	52	41	53	53	57	53	59	52	55	52	53	42	57	35	48	877
iphone 4s	21	47	40	56	46	41	52	52	39	49	68	58	61	63	52	35	34	50	864
asus chromebook	23	42	26	39	38	51	48	41	40	47	52	51	47	56	59	61	48	48	817
acer aspire notebook	21	34	40	40	47	39	43	42	44	35	47	50	52	62	56	56	45	63	816
hp pavilion desktop	15	37	40	31	42	38	55	56	50	47	56	55	57	39	52	51	34	37	792
nexus 7	20	29	41	31	29	44	37	47	49	45	41	39	43	60	39	24	29	31	678
ipad mini	21	29	37	32	25	32	32	38	31	41	35	34	36	34	23	31	27	25	563
nokia lumia 635	19	34	22	21	25	25	31	32	37	41	31	34	42	33	30	26	27	17	527
nexus 10	16	30	25	23	24	28	43	38	30	29	38	28	25	35	19	31	22	23	507
acer aspire desktop	10	26	22	23	28	25	21	23	29	28	29	27	29	32	31	37	36	30	486
htc one	16	19	32	27	20	24	21	19	20	23	28	26	32	30	13	18	19	26	413
kindle fire	6	26	20	22	30	21	25	25	24	26	25	29	36	24	14	12	14	15	394
mac mini	8	12	19	25	18	24	17	29	22	11	15	28	28	23	24	21	32	29	385
windows surface	10	10	15	19	17	15	16	21	19	22	31	33	28	18	18	10	14	20	336
samsung galaxy note	7	15	11	18	20	19	12	19	14	10	14	10	17	15	14	12	13	13	253
amazon fire phone	4	9	12	10	4	5	16	11	12	13	10	6	12	12	14	12	14	11	187
samsumg galaxy tablet	8	11	6	9	6	11	14	11	12	12	15	9	13	9	7	6	12	13	184

Question 4 - Insights



- The device with highest weekly engagement is Macbook Pro with an average engagement of 271.5.
- The device with lowest weekly engagement is Samsung Galaxy Tablet with an average engagement of 10.2.

Question 5 - SQL Statement

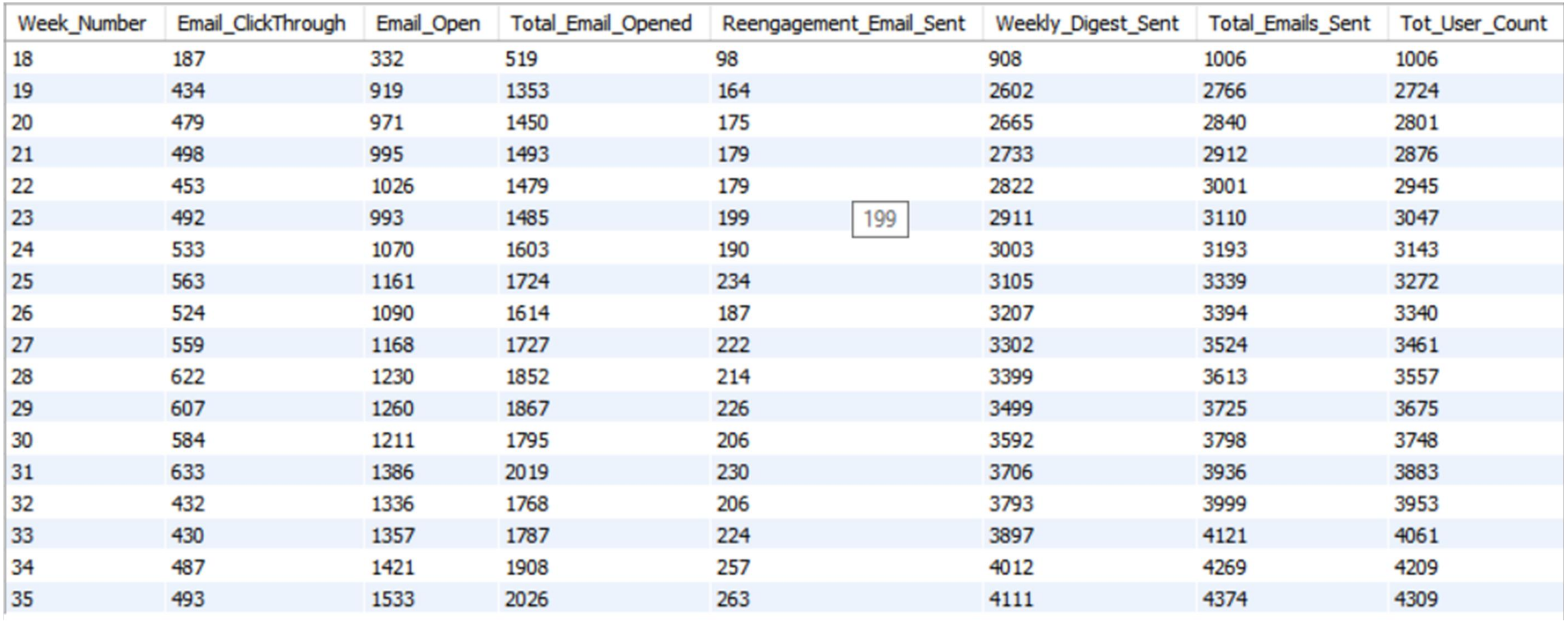
05) Email Engagement Analysis: Analyse how users are engaging with the email service.

Your Task: Write an SQL query to calculate the email engagement metrics.

```
SELECT
    WEEK(occurred_at, 3) AS Week_Number,
    COUNT((CASE WHEN action = 'email_clickthrough' THEN user_id END)) AS Email_ClickThrough,
    COUNT((CASE WHEN action = 'email_open' THEN user_id END)) AS Email_Open,
    COUNT((CASE WHEN action = 'email_clickthrough' THEN user_id END)) + COUNT((CASE WHEN
    action = 'email_open' THEN user_id END)) AS Total_Email_Opened,
    COUNT((CASE WHEN action = 'sent_reengagement_email' THEN user_id END)) AS
    Reengagement_Email_Sent,
    COUNT((CASE WHEN action = 'sent_weekly_digest' THEN user_id END)) AS Weekly_Digest_Sent,
    COUNT((CASE WHEN action = 'sent_reengagement_email' THEN user_id END)) + COUNT((CASE WHEN
    action = 'sent_weekly_digest' THEN user_id END)) AS Total_Emails_Sent,
    COUNT(DISTINCT user_id) AS Tot_User_Count
FROM
    email_events
GROUP BY Week_Number
ORDER BY Week_Number;
```

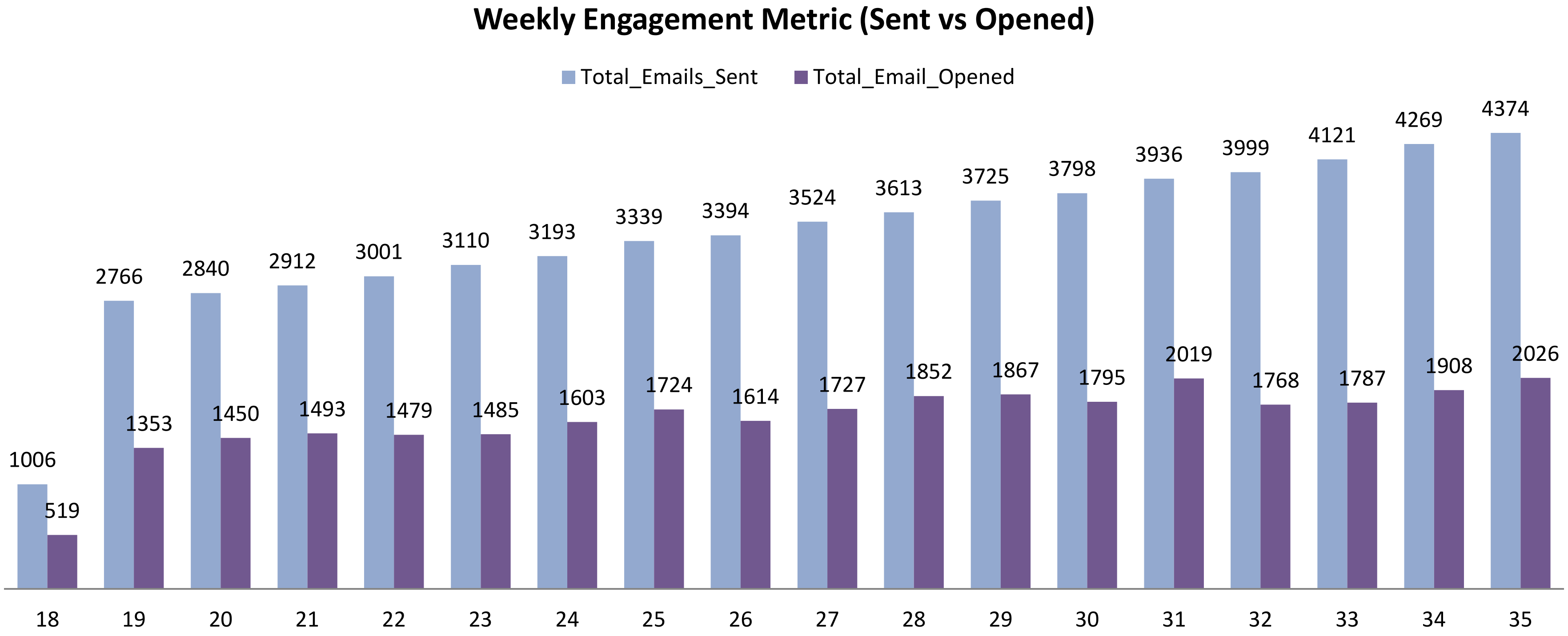
Result on Next Slide

Question 5 - SQL Statement (Continued)



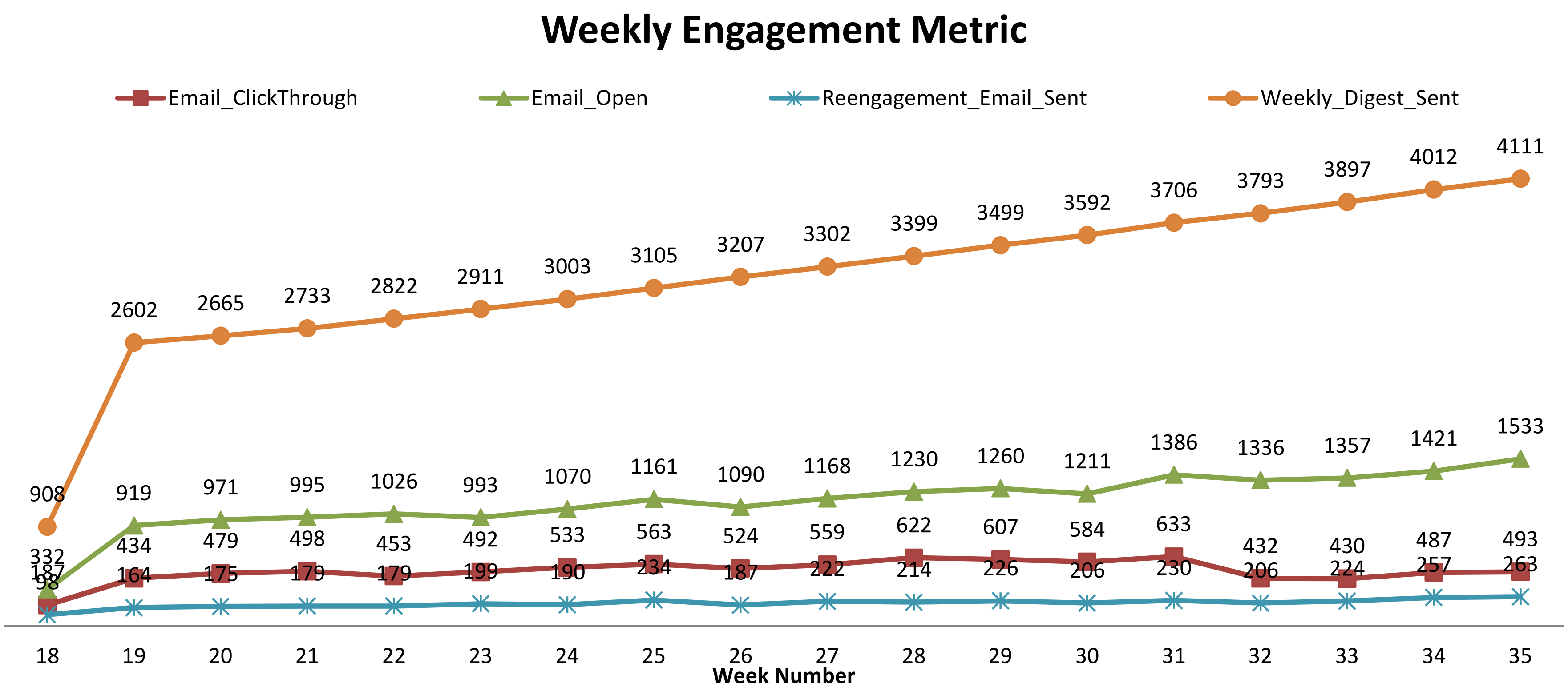
Week_Number	Email_ClickThrough	Email_Open	Total_Email_Opened	Reengagement_Email_Sent	Weekly_Digest_Sent	Total_Emails_Sent	Tot_User_Count
18	187	332	519	98	908	1006	1006
19	434	919	1353	164	2602	2766	2724
20	479	971	1450	175	2665	2840	2801
21	498	995	1493	179	2733	2912	2876
22	453	1026	1479	179	2822	3001	2945
23	492	993	1485	199	2911	3110	3047
24	533	1070	1603	190	3003	3193	3143
25	563	1161	1724	234	3105	3339	3272
26	524	1090	1614	187	3207	3394	3340
27	559	1168	1727	222	3302	3524	3461
28	622	1230	1852	214	3399	3613	3557
29	607	1260	1867	226	3499	3725	3675
30	584	1211	1795	206	3592	3798	3748
31	633	1386	2019	230	3706	3936	3883
32	432	1336	1768	206	3793	3999	3953
33	430	1357	1787	224	3897	4121	4061
34	487	1421	1908	257	4012	4269	4209
35	493	1533	2026	263	4111	4374	4309

Question 5 - Insights



- Less than 50% of the emails sent are opened by users

Question 5 – Insights 2



- Most of the emails sent are Weekly Digests
- Less than 50% Users are opening emails, out of which less than 50% are clicking through.

Results

The project allowed me to advance my SQL skills through research and further learning. It has also allowed me to provide valuable insights based on the given data.

The project was a great learning experience as I was able to research new concepts in SQL, Excel and Data Visualisation. I also gained hands on experience in a real-world project, which allowed me to learn new business concepts related to various metrics.

Through the use of SQL queries, I was able to extract insightful analysis from operational data. The insight gained from this data will help improve the company's operations and understand sudden changes in key metrics.

THANK YOU