

Operation Analytics and Investigating Metric Spike

Name : Aniket Ashok Ubale

Class : B. Tech

Year : 3rd

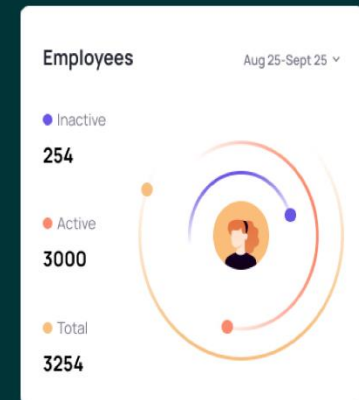
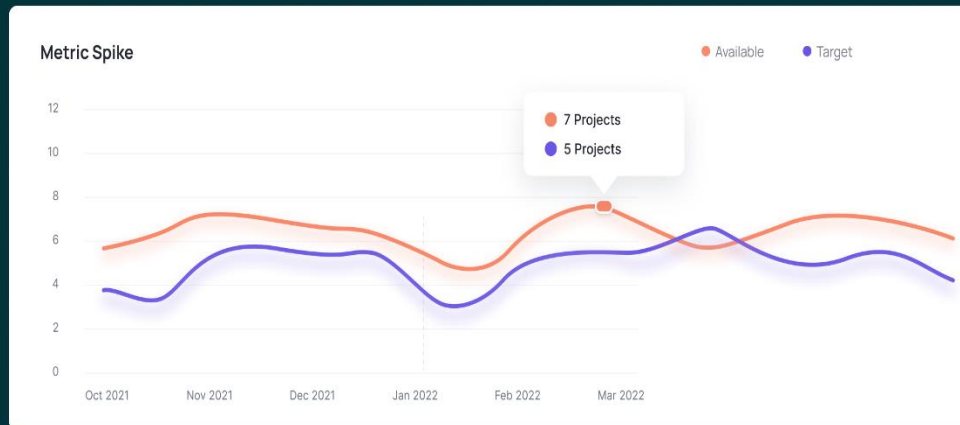
Branch : Artificial intelligence
and data science

Project : Data Analytics Process

Date : 10-04-2023

trainity

Operation Analytics & Investigating metric spike case study



Project description

Operational Analytics is a crucial process that involves analyzing a company's end-to-end operations. This analysis helps identify areas for improvement within the company. As a Data Analyst, you'll 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, you'll need to answer these questions daily, making it crucial to understand how to investigate these metric spikes.

In this project, you'll take on the role of a Lead Data Analyst at a company like Microsoft. You'll be provided with various datasets and tables, and your task will be to derive insights from this data to answer questions posed by different departments within the company. Your goal is to use your advanced SQL skills to analyze the data and provide valuable insights that can help improve the company's operations and understand sudden changes in key metrics.

Case Study 1: Job Data Analysis

You will be working with a table named `job_data` with the following columns:

- `Job_id`: Unique identifier of jobs
- `Actor_id`: Unique identifier of actor
- `event`: The type of event (decision/skip/transfer).
- `language`: The Language of the content
- `Time_spent`: Time spent to review the job in seconds.
- `org`: The Organization of the actor
- `ds`: The date in the format `yyyy/mm/dd` (stored as text).

Tasks:

A. Jobs Reviewed Over Time:

- Objective: 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.

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.

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.

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.

Case Study 2: Investigating Metric Spike

You will be working with three tables:

- users: Contains one row per user, with descriptive information about that user's account.
- events: Contains one row per event, where an event is an action that a user has taken (e.g., login, messaging, search).
- email_events: Contains events specific to the sending of emails.

Tasks:

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.

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.

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.

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.

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.

Please note that for each task, you should also provide insights and interpretations of the results obtained from your queries.

Operation and metric analytics

- Operation analytics is the subset of data analytics that mainly focuses on the improving efficiency of the operations in the company. It shows how currently different operations are working and how those operations can be improved for better profits.
- In this project, I have answered such questions which are generally asked by the different departments in a company like the ops team, support team, marketing team, etc. which are required to increase efficiency and streamline.

Approach-

- To start with the project first I understood all the problem statements. Tried to find out what tables will be required to find the best possible result and marked it to use while actual query writing.
- The queries should be easy to understand. I have written each of the primary and foreign keys for the tables. So, at the time of writing the query, I did have not to check again and again to get those columns. Studied unknown terms like throughput.

Tech-Stack used:-

- To solve these problems, I have used MYSQL Workbench 8.0 CE. Which is an open software and can be downloaded from <https://www.mysql.com/>
- I have created project reports using Microsoft Power Point.

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.
- I learned advanced sql functions.

Case study-1(job data)

➤ Table features

- job_id: unique identifier of jobs
- actor_id: unique identifier of actor
- event: decision/skip/transfer
- language: language of the content
- time_spent: time spent to review the job in seconds
- org: organization of the actor
- ds: date in the yyyy/mm/dd format. It is stored in the form of text and we use presto to run. no need for date function

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

Your task: Calculate the number of jobs reviewed per hour per day for November 2020?

- Query used-
- `SELECT ds AS Dates,
ROUND((COUNT(job_id)/SUM(time_spent))*3600) AS "Jobs Reviewed
per Hour per Day"FROM job data WHERE ds BETWEEN 2020-11-01'
AND '2020-11-30' GROUP BY ds;`

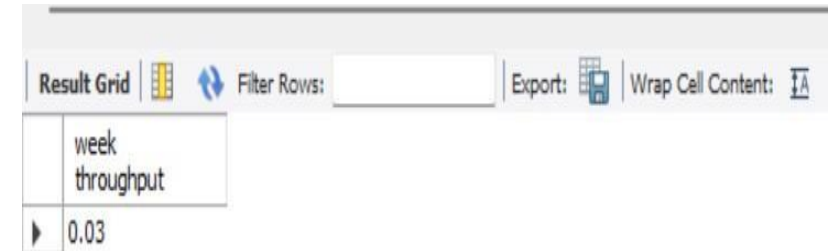
	Dates	Jobs Reviewed per Hour per Day
►	2020-11-30	180
	2020-11-29	180
	2020-11-28	218
	2020-11-27	35
	2020-11-26	64
	2020-11-25	80

Throughput: It is the no. of events happening per second.

Your task: 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?

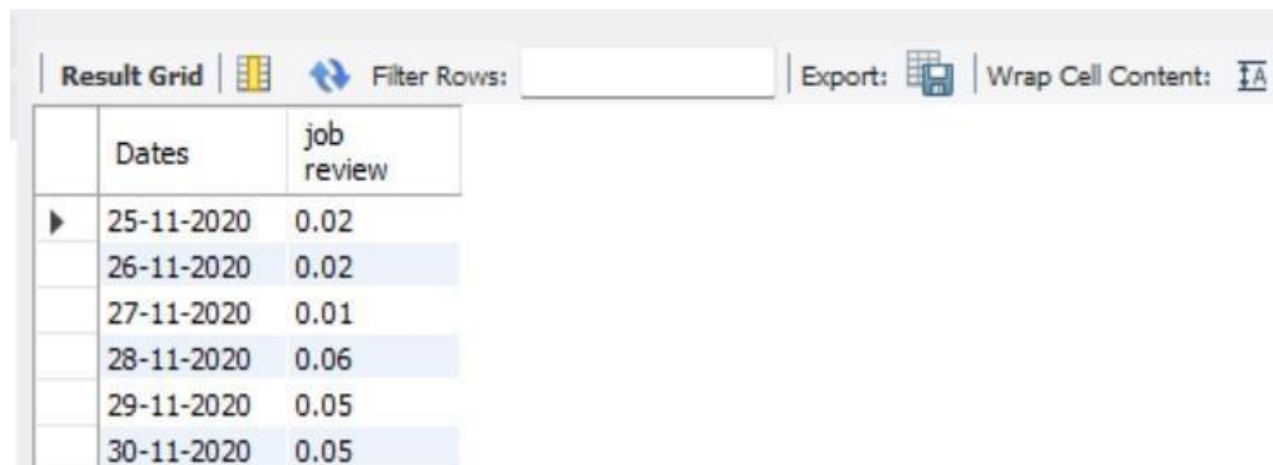
- Query used-

```
SELECT ROUND(COUNT(event)/SUM(time_spent),2)  
AS "week throughput" FROM job_data;
```



Result Grid	Filter Rows:	Export:	Wrap Cell Content:
week throughput			
0.03			

- ```
SELECT ds AS Dates,ROUND(COUNT(event)/SUM(time_spent),2) AS
"job review"FROM job_data GROUP BY ds ORDER BY ds;
```



|   |            |            |
|---|------------|------------|
|   | Dates      | job review |
| ▶ | 25-11-2020 | 0.02       |
|   | 26-11-2020 | 0.02       |
|   | 27-11-2020 | 0.01       |
|   | 28-11-2020 | 0.06       |
|   | 29-11-2020 | 0.05       |
|   | 30-11-2020 | 0.05       |

**Percentage share of each language:** Share of each language for different contents.

**Your task:** Calculate the percentage share of each language in the last 30 days?

- Query used-
- `SELECT language AS language ,ROUND(100*COUNT(*)/total,2) AS percentage FROM job_data CROSS JOIN(SELECT COUNT(*) AS total FROM job_data)sub GROUP BY language;`

|   | Languages | Percentage |
|---|-----------|------------|
| ► | English   | 12.50      |
|   | Arabic    | 12.50      |
|   | Persian   | 37.50      |
|   | Hindi     | 12.50      |
|   | French    | 12.50      |
|   | Italian   | 12.50      |

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

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

- Query used-
- `SELECT actor_id,COUNT(*) AS duplicates FROM job_data GROUP BY actor_id HAVING COUNT(*)>1;`



The screenshot shows a database interface with a 'Result Grid' tab. The grid contains one row of data. The first column is labeled 'actor\_id' and the second column is labeled 'duplicates'. The value in the first column is '1003' and the value in the second column is '2'. Above the grid, there are icons for 'Filter Rows' and 'Export'.

|   | actor_id | duplicates |
|---|----------|------------|
| ▶ | 1003     | 2          |

## Case study-2(investigating metric spike)

- Tables-
- Table-1: users  
This table includes one row per user, with descriptive information about that user's account.
- Table-2: events  
This table includes one row per event, where an event is an action that a user has taken. These events include login events, messaging events, search events, events logged as users progress through a signup funnel, events around received emails.
- Table-3: email\_events  
This table contains events specific to the sending of emails. It is similar in structure to the events table above.

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

**Your task:** Calculate the weekly user engagement?

- Query used-
- `SELECT EXTRACT(WEEK FROM occurred_at AS "Week Numbers",COUNT(DISTINCT user_id) AS "Weekly Active Users" FROM events WHERE event_type = 'engagement' GROUP BY 1;`

| Week Numbers | Weekly Active Users |
|--------------|---------------------|
| 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                 |

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

**Your task:** Calculate the user growth for product?

- Query used-
- `SELECT Months, Users, ROUND(((Users/LAG(Users,1) OVER (ORDER BY Months) - 1)*100), 2) AS "Growth in %"FROM(SELECT EXTRACT(MONTH FROM created_at) AS Months, COUNT(activated_at) AS Users FROM usersWHERE activated_at NOT IN(""))GROUP BY 1ORDER BY 1) sub;`

|   | Months | Users | Growth in % |
|---|--------|-------|-------------|
| ▶ | 1      | 712   | NULL        |
|   | 2      | 685   | -3.79       |
|   | 3      | 765   | 11.68       |
|   | 4      | 907   | 18.56       |
|   | 5      | 993   | 9.48        |
|   | 6      | 1086  | 9.37        |
|   | 7      | 1281  | 17.96       |
|   | 8      | 1347  | 5.15        |
|   | 9      | 330   | -75.50      |



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

```
2 • select first_week,
3 sum(case when week_num=1 then 1 else 0 end) as week_0,
4 sum(case when week_num=2 then 1 else 0 end) as week_1,
5 sum(case when week_num=3 then 1 else 0 end) as week_2,
6 sum(case when week_num=4 then 1 else 0 end) as week_3,
7 sum(case when week_num=5 then 1 else 0 end) as week_4,
8 sum(case when week_num=6 then 1 else 0 end) as week_5,
9 sum(case when week_num=7 then 1 else 0 end) as week_6,
10 sum(case when week_num=8 then 1 else 0 end) as week_7,
11 sum(case when week_num=9 then 1 else 0 end) as week_8,
12 sum(case when week_num=10 then 1 else 0 end) as week_9,
13 sum(case when week_num=11 then 1 else 0 end) as week_10
14 from (select a.user_id, week, first_week, (week-first_week)
15 as week_num from
16 (select user_id, week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i')) as week
17 from events group by user_id, week) a,
18 (select user_id, min(week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i')))
19 as first_week
20 from events group by user_id) b
21 where a.user_id=b.user_id) as with_week_number
22 group by first_week order by first_week;
```

**Weekly Retention:** Users getting retained weekly after signing-up for a product.

**Your task:** Calculate the weekly retention of users-sign up cohort?

|   | 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       |
|   | 32         | 188    | 94     | 8      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0       |
|   | 33         | 202    | 9      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0       |

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

**Your task:** Calculate the weekly engagement per device?

```
2 • select first_week,weekly_number.device,
3 sum(case when week_num=1 then 1 else 0 end) as week_0,
4 sum(case when week_num=2 then 1 else 0 end) as week_1,
5 sum(case when week_num=3 then 1 else 0 end) as week_2,
6 sum(case when week_num=4 then 1 else 0 end) as week_3,
7 sum(case when week_num=5 then 1 else 0 end) as week_4,
8 sum(case when week_num=6 then 1 else 0 end) as week_5,
9 sum(case when week_num=7 then 1 else 0 end) as week_6,
10 sum(case when week_num=8 then 1 else 0 end) as week_7,
11 sum(case when week_num=9 then 1 else 0 end) as week_8,
12 sum(case when week_num=10 then 1 else 0 end) as week_9,
13 sum(case when week_num=11 then 1 else 0 end) as week_10
14 from (select a.user_id,device, week, first_week,(week-first_week)
15 as week_num from
16 (select user_id,device, week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i'))
17 as week
18 from events where event_type='engagement' group by user_id, week,device) a,
19 (select user_id, min(week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i'))))
20 as first_week from events group by user_id) b
21 where a.user_id=b.user_id) as weekly_number
22 group by 1,2 order by 1,2
```

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





**Your task:** Calculate the weekly engagement per device?

|  | 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         | samsung 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      |



Email Engagement: Users engaging with the email service.  
Your task: Calculate the email engagement metrics?

```
2 • select week(occurred_at) as week,
3 count(case when action='sent_weekly_digest' then user_id else null end)
4 as weekly_digest,
5 count(case when action='email_open' then user_id else null end)
6 as email_open,
7 count(case when action='email_clickthrough' then user_id else null end)
8 as email_clickthrough,
9 count(case when action='sent_reengagement_email' then user_id else null end)
10 as reengagement_email
11 from email_events
12 group by 1
```

| Result Grid     Filter Rows: <input type="text"/>   Export:    Wrap Cell Content:  |      |               |            |                    |                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|------------|--------------------|--------------------|
|                                                                                                                                                                                                                                                                                                                                                                                                                            | 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                |

# Result-

- As a result of this project, I was able to provide a detailed report for the two operations mentioned in the project description. I was able to calculate the number of jobs reviewed per hour per day for November 2020, the 7-day rolling average of throughput, the percentage share of each language in the last 30 days, and how to display duplicate rows from the table.
- Additionally, I was able to calculate the weekly user engagement, user growth for a product, weekly retention of users-sign up cohort, and weekly engagement of users

Case study-1 dataset: [link](#)

Case study-2 dataset: [link](#)

Thank you..!!