Operation Analytics and

Investigating Metric Spike

Advanced SQL



# Introduction

This project focuses on analyzing the data which is provided by company. My task is to derive insights and answer the questions asked by different departments. So that these insights are then used by **ops**

**team**, **support team**, **marketing team**, etc to predict the overall growth or decline of a company’s

fortune. It means better automation, better understanding between cross-functional effective workflows.

teams, and more

In case study 1 there is **job\_data** table while in case study 2 there are **users**, **events** and **email\_events**

tables.

In case study 1 the insights are found based on following questions:

1. **Number of jobs reviewed:** Amount of jobs reviewed over time.
2. **Throughput:** Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?
3. **Percentage share of each language:** Share of each language for different contents.
4. **Duplicate rows:** Rows that have the same value present in them.

In case study 2 the insights are found based on following questions:

1. **User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.
2. **User Growth:** Amount of users growing over time for a product.
3. **Weekly Retention:** Users getting retained weekly after signing-up for a product.
4. **Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.
5. **Email Engagement:** Users engaging with the email service.

## What is Operation Analytics?

Operational analytics focuses on measuring the existing and real-time operations of the company so that the company can monitor their day-to-day operations basis which they can take the necessary actions to improve customer satisfaction and bottom line.

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 predict the overall growth or decline of a company’s fortune.

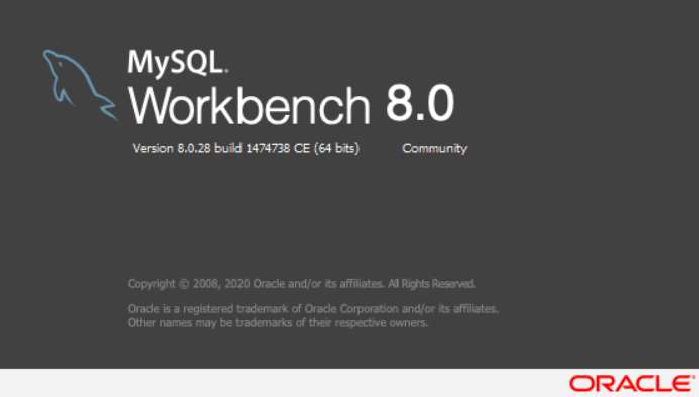
Here are few examples:

* **Ops:** Developers can use real-time data to look at how customers are using their products and make changes on the fly.
* **Marketing:** Businesses can optimize user engagement in real-time by using operational analytics to make personalized recommendations.

# Project Approach

This project is developed using SQL Workbench. First I need to create database by using dataset file which was provided by the company. Next step load the data into SQL Workben h then performed analysis and find the information that will the help the **ops team**, **support team**, **marketing team**, etc to 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 spike.

# Tech-Stack Used

MySQL Workbench is a visual editor that unifies data modeling, SQL development, and database administration in one interface.

It allows you to visually design, and manage databases.

generate,

MySQL Workbench is widely used to handle structured data. It is an open-source Relational Database Management System (RDBMS) developed by Oracle Corporation, Sun Microsystems that uses Structured Query language (SQL) to interact with databases.

MySQL Workbench offers database migration options, making it easier to move data to and from the Microsoft SQL Server, Microsoft Access and other RDBMS tables.

## Key Functionalities of MySQL Workbench

* **Visual SQL Editor:** MySQL Workbench is equipped with a visual SQL editor where developers can build, edit, and run queries. What’s great about this is that it allows you to preview your changes before applying them.
* **Database Administration:** Aside from providing you with SQL editing tools, MySQL Workbench also comes with a database administration suite. This makes it easy for you to audit your databases, configure servers, and view logs.
* **Performance Monitoring:** MySQL Workbench gives users a dashboard where they can view the status of their queries, client timing, network latency, and index usage. This allows for simpler identification of possible ways to optimize SQL performance.

# Insights

### Case Study 1 (Job Data)

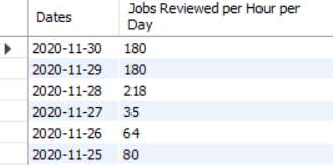
#### Calculate the number of jobs reviewed per hour per day for November 2020?

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;

On date 2020-11-28 there is maximum number of jobs reviewed that is **218**.



#### Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7- day rolling and why?

SELECT ROUND(COUNT(event)/SUM(time\_spent), 2) AS "Weekly Throughput" FROM job\_data;

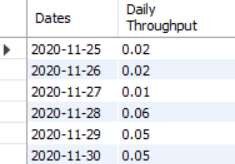
The weekly throughput is **0.03**.



SELECT ds AS Dates, ROUND(COUNT(event)/SUM(time\_spent), 2) AS "Daily Throughput" FROM job\_data

GROUP BY ds ORDER BY ds;

On date 2020-11-28 the throughput is highest **0.06**.



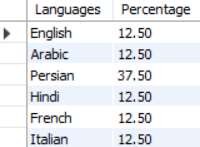
Metrics will always go up and down on a weekly and daily basis. You’ll get numbers faster every day or minute if you want. As a result, rolling metrics are superb at showing if your metrics are trending up or down on a daily level.

#### Calculate the percentage share of each language in the last 30 days?

SELECT language AS Languages, ROUND(100 \* COUNT(\*)/total, 2) AS Percentage FROM job\_data

CROSS JOIN (SELECT COUNT(\*) AS total FROM job\_data) sub GROUP BY language;

Persian language is highest with **37.5%** total.



#### Let’s say you see some duplicate rows in the data. How will you display duplicates from the table?

SELECT actor\_id, COUNT(\*) AS Duplicates FROM job\_data

GROUP BY actor\_id HAVING COUNT(\*) > 1;

Actor ID **1003** has duplicate rows.



### Case Study 2 (Investigating Metric Spike)

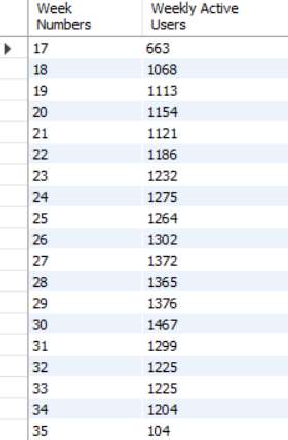


#### Calculate the weekly user engagement?

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;



#### Calculate the user growth for product?

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 users

WHERE activated\_at NOT IN("") GROUP BY 1

ORDER BY 1

) sub;

#### Calculate the weekly retention of users-sign up cohort?

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", FROM

(

SELECT m.user\_id, m.login\_week, n.first, m.login\_week - first AS week\_number FROM

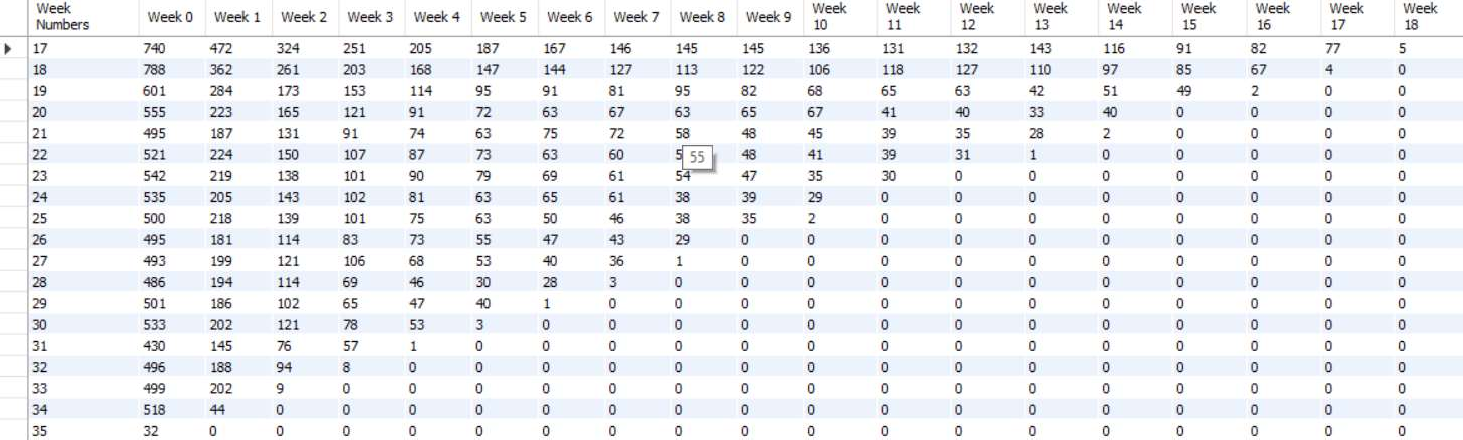
(SELECT user\_id, EXTRACT(WEEK FROM occurred\_at) AS login\_week FROM events GROUP BY 1, 2) m,

(SELECT user\_id, MIN(EXTRACT(WEEK FROM occurred\_at)) AS first FROM events GROUP BY 1) n

WHERE m.user\_id = n.user\_id

) sub

GROUP BY first ORDER BY first;



#### Calculate the weekly engagement per device?

SELECT EXTRACT(WEEK FROM occurred\_at) AS "Week Numbers",

COUNT(DISTINCT CASE WHEN device IN('dell inspiron notebook') THEN user\_id ELSE NULL END) AS "Dell Inspiron Notebook",

COUNT(DISTINCT CASE WHEN device IN('iphone 5') THEN user\_id ELSE NULL END) AS

"iPhone 5",

COUNT(DISTINCT CASE WHEN device IN('iphone 4s') THEN user\_id ELSE NULL END) AS

"iPhone 4S",

COUNT(DISTINCT CASE WHEN device IN('windows surface') THEN user\_id ELSE NULL END) AS "Windows Surface",

COUNT(DISTINCT CASE WHEN device IN('macbook air') THEN user\_id ELSE NULL END) AS "Macbook Air",

COUNT(DISTINCT CASE WHEN device IN('iphone 5s') THEN user\_id ELSE NULL END) AS

"iPhone 5S",

COUNT(DISTINCT CASE WHEN device IN('macbook pro') THEN user\_id ELSE NULL END) AS "Macbook Pro",

COUNT(DISTINCT CASE WHEN device IN('kindle fire') THEN user\_id ELSE NULL END) AS "Kindle Fire",

COUNT(DISTINCT CASE WHEN device IN('ipad mini') THEN user\_id ELSE NULL END) AS "iPad Mini",

COUNT(DISTINCT CASE WHEN device IN('nexus 7') THEN user\_id ELSE NULL END) AS

"Nexus 7",

COUNT(DISTINCT CASE WHEN device IN('nexus 5') THEN user\_id ELSE NULL END) AS

"Nexus 5",

COUNT(DISTINCT CASE WHEN device IN('samsung galaxy s4') THEN user\_id ELSE NULL END) AS "Samsung Galaxy S4",

COUNT(DISTINCT CASE WHEN device IN('lenovo thinkpad') THEN user\_id ELSE NULL END) AS "Lenovo Thinkpad",

COUNT(DISTINCT CASE WHEN device IN('samsumg galaxy tablet') THEN user\_id ELSE NULL END) AS "Samsumg Galaxy Tablet",

COUNT(DISTINCT CASE WHEN device IN('acer aspire notebook') THEN user\_id ELSE NULL END) AS "Acer Aspire Notebook",

COUNT(DISTINCT CASE WHEN device IN('asus chromebook') THEN user\_id ELSE NULL END) AS "Asus Chromebook",

COUNT(DISTINCT CASE WHEN device IN('htc one') THEN user\_id ELSE NULL END) AS "HTC One",

COUNT(DISTINCT CASE WHEN device IN('nokia lumia 635') THEN user\_id ELSE NULL END) AS "Nokia Lumia 635",

COUNT(DISTINCT CASE WHEN device IN('samsung galaxy note') THEN user\_id ELSE NULL END) AS "Samsung Galaxy Note",

COUNT(DISTINCT CASE WHEN device IN('acer aspire desktop') THEN user\_id ELSE NULL END) AS "Acer Aspire Desktop",

COUNT(DISTINCT CASE WHEN device IN('mac mini') THEN user\_id ELSE NULL END) AS "Mac Mini",

COUNT(DISTINCT CASE WHEN device IN('hp pavilion desktop') THEN user\_id ELSE NULL END) AS "HP Pavilion Desktop",

COUNT(DISTINCT CASE WHEN device IN('dell inspiron desktop') THEN user\_id ELSE NULL END) AS "Dell Inspiron Desktop",

COUNT(DISTINCT CASE WHEN device IN('ipad air') THEN user\_id ELSE NULL END) AS "iPad Air",

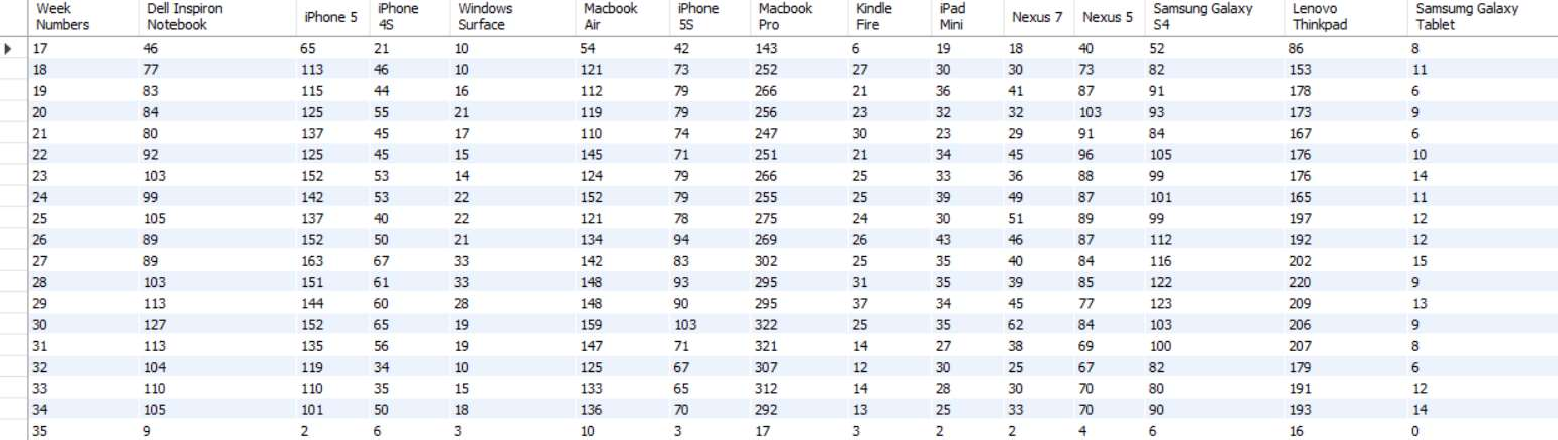
COUNT(DISTINCT CASE WHEN device IN('amazon fire phone') THEN user\_id ELSE NULL END) AS "Amazon Fire Phone",

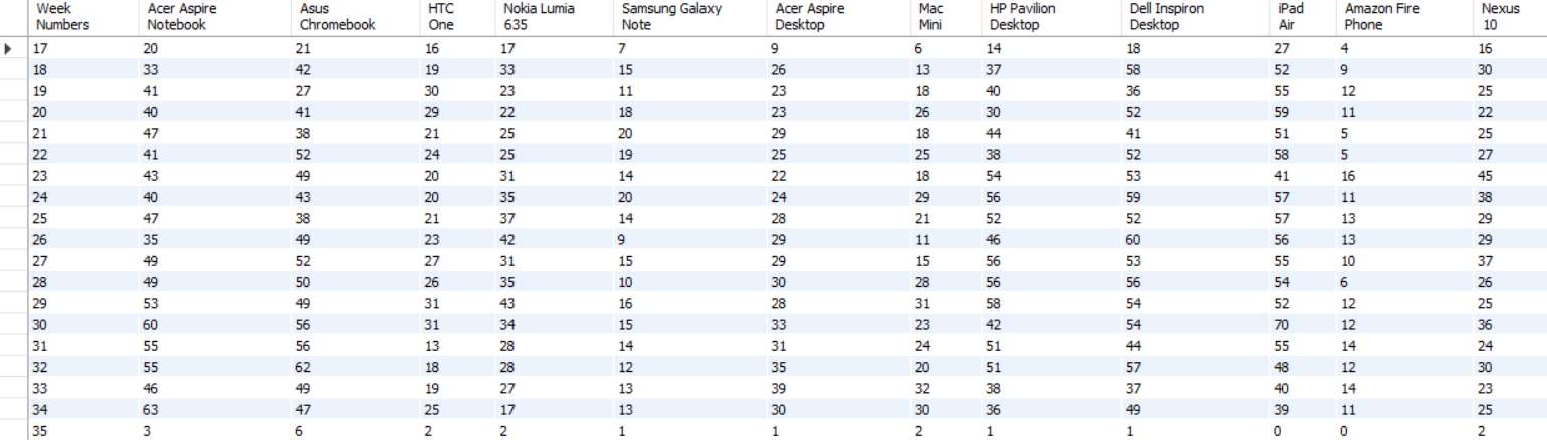
COUNT(DISTINCT CASE WHEN device IN('nexus 10') THEN user\_id ELSE NULL END) AS

"Nexus 10" FROM events

WHERE event\_type = 'engagement' GROUP BY 1

ORDER BY 1;





#### Calculate the email engagement metrics?

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 occurred\_at) AS Week,

COUNT(CASE WHEN action = 'sent\_weekly\_digest' THEN user\_id ELSE NULL END) AS weekly\_digest,

COUNT(CASE WHEN

email\_opens,

action = 'email\_open' THEN user\_id ELSE

NULL END) AS

COUNT(CASE WHEN action = 'email\_clickthrough' THEN user\_id ELSE NULL END) AS email\_clickthroughs,

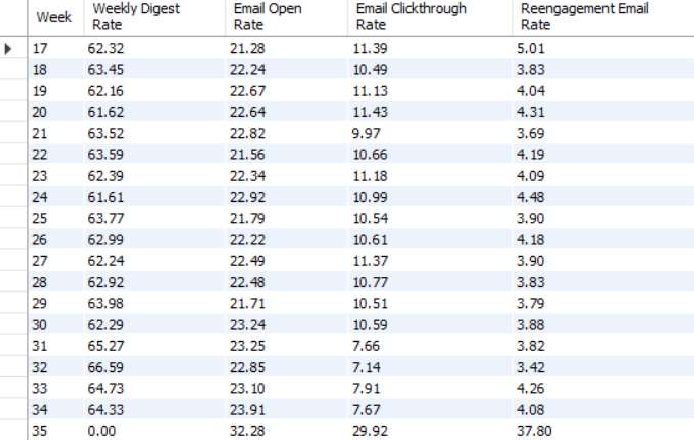
COUNT(CASE WHEN action = 'sent\_reengagement\_email' THEN user\_id ELSE NULL END) AS reengagement\_emails,

COUNT(user\_id) AS total FROM email\_events GROUP BY 1

) sub

GROUP BY 1

ORDER BY 1;



# Result

**How this project helped me:** This project helps me to understand the importance of operation analytics. Through this project I am able to understand how the companies use metric spike as a secret weapon. With an informed and proactive approach, they can leverage insights to make data-backed decisions that optimize their strategy and boost ROI.

**Challenges that I faced in this project:** The challenge here is that the data in **case study 2** is very huge, as the huge amount of data SQL Workbech is very slow to import it. To tackle this situation I have to use **LOAD DATA** statements. Now, there is another problem arises in the column **user\_type** in **events** table that has datatype **int** which is stopping the process of importing. First I need to change its datatype to **text** then restart the process of loading the data into **events** table.

**Conclusion:** Operational Analytics tackles the problem by synchronizing real-time data. Operational

Analytics has the capability to aggregate data from multiple data sources into a cumulative, organized, actionable solution capable of delivering analytical models in real-time to create individual customer profiles and a holistic view of operations for a company. This guarantees that your operational routines

and systems are used efficiently. significant positive effect on our efficiency of specific areas.

Whenever utilized correctly, operational analyt cs can achieve a general public and world everywhere and increment the general