**Operation Analytics and Investigating Metric Spike**

**Project Description**

This project consist of 2 Case Studies, One is for Operation Analytics and other one is for Investigating Metric Spike. In the Operation Analytics I have to find the ares on which company must improve on and I have to derive insights out of the data company collects.

In the Investigating Metric Spike, metric spike is the sudden increase in a particular metric. In this I have used different KPI’s and parameter to analyse product’s user engagement, user growth, weekly retention, weekly engagement and email engagement.

**Approach**

Approach for making this project is first undersatanding all the datasets. After that creating a database and insert all the tables or data in the database. I have used advance SQl queries like window functions, subqueries etc. to get the answers and gather insights to make good decisions for the company.

**Tech-Stack Used**

* I have used MySQL Version 8.0.33, MySql Workbench 8.0 CE for writing the queries.

**Insights**

* Through this project I have got the insights like number of jobs reviewed per hour per day, throughput, **Percentage share of each language.**
* Highest no. of active users each week.
* Maximum user engagement per week
* Maximum no. of email sent per month.
* Daily throughput is always better than weekly as we can see daily ups and downs.

**Result**

In this project of Operation Analytics and Investigating Metric Spike, I have

achieved various Analytics and logical skills as well as technical skills to

efficiently use MySQL

In this project of Operation Analytics and Investigating Metric Spike, I have

achieved various Analytics and logical skills as well as technical skills to

efficiently use MySQL

In this project of Operation Analytics and Investigating Metric Spike, I have

achieved various Analytics and logical skills as well as technical skills to

efficiently use MySQL

In this project of Operation Analytics and Investigating Metric Spike, I have

achieved various Analytics and logical skills as well as technical skills to

efficiently use MySQL

In this project of Operation Analytics and Investigating Metric Spike, I have achieved various Analytics and logical skills as well as technical skills to efficiently use MySQl. I have come to know the use of the window function, nested queries or sub-queries etc. It helped me to get more understanding of when and where to use appropriate functions. It will add to my portfolio by which this project help me to get good opportunities in the future.

**Commands for Queries**

**Case Study 1 (Job Data)**

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

SELECT

ds,

(COUNT(job\_id) / SUM(time\_spent) \* 3600) AS jobs\_per\_hour\_per\_day

FROM

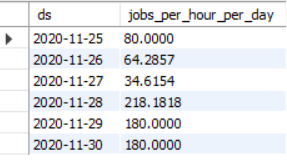
job\_data

WHERE

ds BETWEEN '2020-11-01' AND '2020-11-30'

GROUP BY ds

ORDER BY ds;



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

**-- Weekly Throghput (7 Day Rolling)**

SELECT

num\_of\_events / total\_time\_spent AS weekly\_throughput

FROM

(SELECT

COUNT(event) AS num\_of\_events,

SUM(time\_spent) AS total\_time\_spent

FROM

job\_data) j;

****

**-- Daily Throghput**

SELECT

ds, num\_of\_events / total\_time\_spent AS daily\_throughput

FROM

(SELECT

ds,

COUNT(event) AS num\_of\_events,

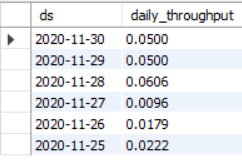
SUM(time\_spent) AS total\_time\_spent

FROM

job\_data

GROUP BY ds) j

GROUP BY ds;



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

SELECT

language,

ROUND((COUNT(language) \* 100 / (SELECT

COUNT(\*)

FROM

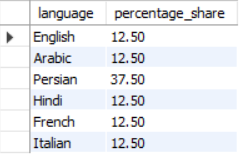
job\_data)),

2) AS percentage\_share

FROM

job\_data

GROUP BY language;



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

SELECT \*

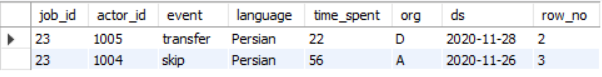
FROM (

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY job\_id) AS row\_no

FROM job\_data

) AS a

WHERE row\_no > 1;



**Case Study 2 (Investigating metric spike)**

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

SELECT

COUNT(DISTINCT user\_id) AS active\_user,

WEEK(occurred\_at) AS week\_num

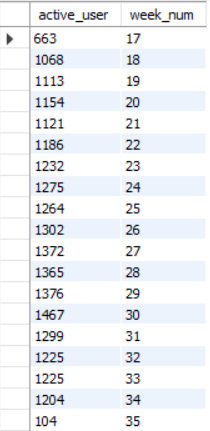
FROM

events

WHERE

event\_type = 'engagement'

GROUP BY week\_num;



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

select

year, week, num\_users,

sum(num\_users) over ( rows between unbounded preceding and current row) as active\_users

from

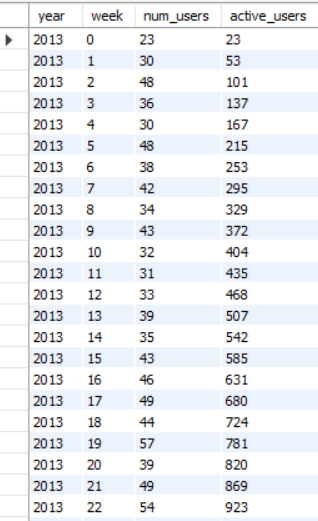
(select year(activated\_at) as year, week(activated\_at) as week, count(user\_id) as num\_users from users

where

state='active'

group by year, week

order by year, week) as a;



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

SELECT

YEAR(OCCURRED\_AT) AS YEAR,

WEEK(OCCURRED\_AT) AS WEEK,

DEVICE,

COUNT(DISTINCT user\_id) AS users\_retention

FROM

events

WHERE

EVENT\_TYPE = 'ENGAGEMENT'

GROUP BY year , week , device

ORDER BY year , week , device;



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

SELECT DISTINCT

device,

WEEK(occurred\_at) AS week,

COUNT(DISTINCT user\_id) AS users\_engagement

FROM

events

WHERE

event\_type = 'engagement'

GROUP BY week , device

ORDER BY week , device;



**5. Email Engagement: Users engaging with the email service.**

SELECT

YEAR(occurred\_at) AS year,

MONTH(occurred\_at) AS month,

action,

COUNT(action) AS email\_engagement

FROM

email\_events

GROUP BY year , month , action

ORDER BY year , month , action;

