

# OPERATION **ANALYTICS** AND INVESTIGATING METRIC SPIKE



Project By Shashank Haritwal





# CONTENTS

Introduction

Case Study 1  
(Job Data Analysis)

Case Study 2  
(Investigating Metric Spike)

Approach and Tech-stack used

Insight and Results

# Introduction

In this project, we are supposed to provide detailed report to answer questions posed by different departments within the company . **Advanced SQL skills** are used 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



### Jobs Reviewed Over Time

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

### Throughput Analysis

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

### Language Share Analysis

- Calculate the percentage share of each language in the last 30 days.

### Duplicate rows detection

- Identify duplicate rows in the data.



# 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.

## Query :

```
SELECT  
    ds AS Date_, COUNT(job_id) / 24 AS  
Jobs_Reviewed_Per_Hour  
FROM  
    job_data  
GROUP BY Date_;
```

## Output Table :

	Date_	Jobs_Reviewed_Per_Hour
▶	11/30/2020	0.0833
	11/29/2020	0.0417
	11/28/2020	0.0833
	11/27/2020	0.0417
	11/26/2020	0.0417
	11/25/2020	0.0417

**From the output** it can be seen that the maximum jobs are reviewed per hour on 30<sup>th</sup> and 28<sup>th</sup> of November 2020.



## B. Throughput Analysis

**Objective :** Calculate the 7-day rolling average of throughput.

**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..

### Query :

```
SELECT
    COUNT(event) / (SELECT
        COUNT(DISTINCT ds)
    FROM
        job_data) AS Avg_Throughput
FROM
    job_data;
```

### Output Table :

	Avg_Throughput
▶	1.3333

Since the given table is having record for 6 days only , the 6 day rolling average throughput is calculated by dividing the throughput of each day with the total no. of days

## B. Throughput Analysis



**Objective :** Calculate the 7-day rolling average of throughput.

**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..

Explanation:

To notice trends and fluctuation in records on a long term basis 7-days rolling averages are preferable , a rolling average is also called as moving average .Rolling average can also help you to determine the factors causing trends

## 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.

### Query :

```
SELECT
    language,
    COUNT(*) AS frequency,
    (COUNT(*) * 100) / (SELECT
        COUNT(*)
    FROM
        job_data) AS percentage_share
FROM
    job_data
GROUP BY language;
```

### Output Table :

	language	frequency	percentage_share
▶	English	1	12.5000
	Arabic	1	12.5000
	Persian	3	37.5000
	Hindi	1	12.5000
	French	1	12.5000
	Italian	1	12.5000

**From the output** we can say that Persian language is the most frequently used language , with a percentage share of 37.5%





## 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.

**Query :**

```
select * from job_data;  
SELECT  
    *  
FROM  
    job_data  
GROUP BY ds , job_id , actor_id , event , language , time_spent , org  
HAVING COUNT(*) > 1;
```

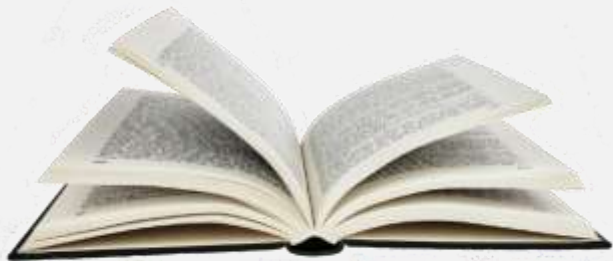
**Output Table :**

ds	job_id	actor_id	event	language	time_spent	org

**From the above output** it can be interpreted that there are **no duplicate records** in the data

# **CASE STUDY 2**

## **INVESTIGATING METRIC SPIKE**



### Weekly User Engagement

- Measure the activeness of users on a weekly basis.

### User Growth Analysis

- Analyze the growth of users over time for a product.

### Weekly Retention Analysis

- Analyze the retention of users on a weekly basis after signing up for a product.

### Weekly Engagement Per Device

- Measure the activeness of users on a weekly basis per device.

### Email Engagement Analysis

- Analyze how users are engaging with the email service.

# 1. WEEKLY USER ENGAGEMENT.

**Objective :** Measure the activeness of users on a weekly basis.

**Query :**

```
SELECT
    CONCAT(WEEKOFYEAR(activated_at),
           ' of ',
           YEAR(activated_at)) AS weeks,
    COUNT(user_id) AS users_active
FROM
    users
WHERE
    state = 'active'
GROUP BY weeks;
```



# 1. WEEKLY USER ENGAGEMENT.

**Objective :** Measure the activeness of users on a weekly basis.

**Output :**

	weeks	users_active
►	1 of 2013	67
	2 of 2013	29
	3 of 2013	47
	4 of 2013	36
	5 of 2013	30
	6 of 2013	48
	7 of 2013	41
	8 of 2013	39
	9 of 2013	33
	10 of 2013	43
	11 of 2013	33
	12 of 2013	32
	13 of 2013	33
	14 of 2013	40
	15 of 2013	35
	16 of 2013	42
	17 of 2013	48
	18 of 2013	48
	19 of 2013	45
	20 of 2013	55

There are total 87 rows in the output

	weeks	users_active
	48 of 2013	103
	49 of 2013	96
	50 of 2013	117
	51 of 2013	123
	52 of 2013	104
	1 of 2014	91
	2 of 2014	122
	3 of 2014	112
	4 of 2014	113
	5 of 2014	130
	6 of 2014	132
	7 of 2014	135
	8 of 2014	127
	9 of 2014	127
	10 of 2014	135
	11 of 2014	152
	12 of 2014	132
	13 of 2014	151
	14 of 2014	161
	15 of 2014	166



# 1. WEEKLY USER ENGAGEMENT.

## Interpretation of result : -

- The results shows the Number of active users on weekly basis , and result is shown for both the years(2013 and 2014) separately.
- It can be seen that the number of active users per week in 2014 is more than that in 2013 .
- And hence the engagement of users has been increased





## 2. User Growth Analysis.

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

**Query :**

```
SELECT
    device,
    MONTH(occurred_at) AS `Month`,
    COUNT(DISTINCT (user_id)) AS Users_Signed_Up
FROM
    events
WHERE
    event_type = 'signup_flow'
GROUP BY device , `Month`;
```



# 2. User Growth Analysis.

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

**Output :**

	device	Month	Users_Signed_Up
	dell inspiron notebook	6	57
	dell inspiron notebook	7	51
	dell inspiron notebook	8	54
	hp pavilion desktop	5	22
	hp pavilion desktop	6	24
	hp pavilion desktop	7	25
	hp pavilion desktop	8	22
	htc one	5	15
	htc one	6	11
	htc one	7	7
	htc one	8	15
	ipad air	5	30
	ipad air	6	28
	ipad air	7	34
	ipad air	8	36
	ipad mini	5	16
	ipad mini	6	15
	ipad mini	7	18
	ipad mini	8	17
	iphone 4s	5	22
	iphone 4s	6	19

There are  
total 104  
rows in the  
output .....

	device	Month	Users_Signed_Up
	mac mini	8	16
	macbook air	5	63
	macbook air	6	90
	macbook air	7	83
	macbook air	8	68
	macbook pro	5	131
	macbook pro	6	130
	macbook pro	7	159
	macbook pro	8	205
	nexus 10	5	13
	nexus 10	6	18
	nexus 10	7	20
	nexus 10	8	21
	nexus 5	5	46
	nexus 5	6	36
	nexus 5	7	44
	nexus 5	8	44
	nexus 7	5	16
	nexus 7	6	23
	nexus 7	7	22
	nexus 7	8	23



## 2. User Growth Analysis.

### Interpretation of result : -

- The result shows the name of the device and the months for which the number of users signed up is given.
- It can be seen that more users are signing up in the successive months and hence the growth of a particular product can be seen.



# 3. Weekly Retention Analysis:.



**Objective :** Analyze the retention of users on a weekly basis after signing up for a product.

**Query :**

```
SELECT
    us.weeks,
    us.user_s AS users_signup,
    ur.user_r AS users_retain,
    (ur.user_r) / (us.user_s) * 100 AS percentage_of_user_retention
FROM
    (SELECT
        WEEK(occurred_at) AS weeks,
        COUNT(DISTINCT user_id) AS user_s
    FROM
        events
    WHERE
        event_type = 'engagement'
    GROUP BY weeks) AS us
JOIN
```

```
(SELECT
    WEEK(occurred_at) AS weeks,
    COUNT(DISTINCT user_id) AS user_r
FROM
    events
WHERE
    event_type = 'signup_flow'
GROUP BY weeks) AS ur ON us.weeks = ur.weeks
GROUP BY weeks;
```

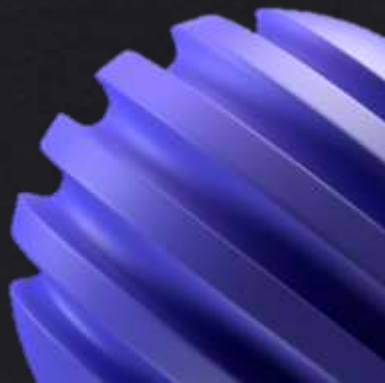


# 3. Weekly Retention Analysis:.

**Objective :** Analyze the retention of users on a weekly basis after signing up for a product.

**Output :**

	weeks	users_signup	users_retain	percentage_of_user_retention
▶	17	663	72	10.8597
	18	1068	163	15.2622
	19	1113	185	16.6217
	20	1154	176	15.2513
	21	1121	183	16.3247
	22	1186	196	16.5261
	23	1232	196	15.9091
	24	1275	229	17.9608
	25	1264	207	16.3766
	26	1302	201	15.4378
	27	1372	222	16.1808
	28	1365	215	15.7509
	29	1376	221	16.0610
	30	1467	238	16.2236
	31	1299	193	14.8576
	32	1225	245	20.0000
	33	1225	261	21.3061
	34	1204	259	21.5116
	35	104	18	17.3077





# 3. Weekly Retention Analysis:.



## **Interpretation of result : -**

- **The result shows the number of users that signed up and the number of users that are active after signing up in the particular week .**
- **From percentage\_of\_user\_retention it can be seen the weekly retention of users .**

# D. Weekly Engagement Per Device.

**Objective :** Measure the activeness of users on a weekly basis per device.

**Query :**

```
SELECT
    device,
    CONCAT(WEEKOFYEAR(occurred_at),
           ' of ',
           YEAR(occurred_at)) AS week_,
    COUNT(DISTINCT user_id) AS users_engaged
FROM
    events
GROUP BY week_ , device;
```



# D. Weekly Engagement Per Device.

**Objective :** Measure the activeness of users on a weekly basis per device.

**Output :**

	device	week_	users_engaged
▶	acer aspire desktop	18 of 2014	10
	acer aspire desktop	19 of 2014	26
	acer aspire desktop	20 of 2014	22
	acer aspire desktop	21 of 2014	23
	acer aspire desktop	22 of 2014	28
	acer aspire desktop	23 of 2014	25
	acer aspire desktop	24 of 2014	21
	acer aspire desktop	25 of 2014	23
	acer aspire desktop	26 of 2014	29
	acer aspire desktop	27 of 2014	28
	acer aspire desktop	28 of 2014	29
	acer aspire desktop	29 of 2014	27
	acer aspire desktop	30 of 2014	29
	acer aspire desktop	31 of 2014	32
	acer aspire desktop	32 of 2014	31
	acer aspire desktop	33 of 2014	37
	acer aspire desktop	34 of 2014	36
	acer aspire desktop	35 of 2014	30
	acer aspire notebook	18 of 2014	21
	acer aspire notebook	19 of 2014	34
	acer aspire notebook	20 of 2014	40
	acer aspire notebook	21 of 2014	40

There are 468 rows  
in the output....

	device	week_	users_engaged
	acer aspire notebook	30 of 2014	52
	acer aspire notebook	31 of 2014	62
	acer aspire notebook	32 of 2014	56
	acer aspire notebook	33 of 2014	56
	acer aspire notebook	34 of 2014	45
	acer aspire notebook	35 of 2014	63
	amazon fire phone	18 of 2014	4
	amazon fire phone	19 of 2014	9
	amazon fire phone	20 of 2014	12
	amazon fire phone	21 of 2014	10
	amazon fire phone	22 of 2014	4
	amazon fire phone	23 of 2014	5
	amazon fire phone	24 of 2014	16
	amazon fire phone	25 of 2014	11
	amazon fire phone	26 of 2014	12
	amazon fire phone	27 of 2014	13
	amazon fire phone	28 of 2014	10
	amazon fire phone	29 of 2014	6
	amazon fire phone	30 of 2014	12
	amazon fire phone	31 of 2014	12
	amazon fire phone	32 of 2014	14
	amazon fire phone	33 of 2014	12
	amazon fire phone	34 of 2014	14
	amazon fire phone	35 of 2014	11
	asus chromebook	18 of 2014	23
	asus chromebook	19 of 2014	40
	asus chromebook	20 of 2014	40
	asus chromebook	21 of 2014	40



# D. Weekly Engagement Per Device.

## Interpretation of result : -

- The result contains column for week , device name and users engagement for that particular device in that particular week .
- ie. For Acer aspire desktop the number of users engaged in 18<sup>th</sup> and 19<sup>th</sup> week are 10 and 26 respectively.
- A trend can be noticed here , that is the no. of users engaging per device are increasing with the increasing week number. That means more user are participating with time.





# E. Email Engagement Analysis.

**Objective :** Analyze how users are engaging with the email service..



**Query :**

```
SELECT
    MONTH(occurred_at) AS Month,
    action,
    COUNT(user_id) AS `No._of_Users`
FROM
    email_events
GROUP BY month , action;
```



# E. Email Engagement Analysis.

**Objective :** Analyze how users are engaging with the email service..

**Output :**

	Month	action	No._of_Users
▶	5	sent_weekly_digest	11730
	5	email_open	4212
	5	email_clickthrough	2023
	5	sent_reengageme...	758
	6	sent_weekly_digest	13155
	6	email_open	4658
	6	email_clickthrough	2274
	6	sent_reengageme...	889
	7	sent_weekly_digest	15902
	7	email_open	5611
	7	email_clickthrough	2721
	7	sent_reengageme...	933
	8	sent_weekly_digest	16480
	8	email_open	5978
	8	email_clickthrough	1992
	8	sent_reengageme...	1073
	8	sent_reengageme...	1013



# E. Email Engagement Analysis.

## Interpretation of result : -

- The result shows the no. of users with there action and the trend is shown on monthly basis .
- It can be seen here that over months more number of emails are sent to the users hence the action of opening email is done by more users .



## Approach :

- **1. Accessed the database provided for the analysis that contains the various required Information.**
- **2. And then thoroughly studied the database containing various tables and records.**

## Tech stack used :



MySQL Workbench 8.0

# Insights and Result

- From this project I got an idea about how the datasets are maintained and analyzed in businesses and companies.
- While going through this project I learn how to deal with huge datasets containing millions or records .
- Advanced MySQL skills are used to do this project , to understand the analysis process well, and to provide insights for the best decision possible

