

# **OPERATION & INVESTIGATING METRIC ANALYTICS**

**Project by : Mayur Rajput**

# CONTENT

- 01** PROJECT DESCRIPTION
- 02** APPROACH
- 03** TECH-STACK USED
- 04** JOB DATA ANALYSIS
- 05** INVESTIGATING METRIC SPIKE
- 06** INSIGHTS
- 07** RESULT

# PROJECT DESCRIPTION



In this project, we are supposed to use our 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.



Through this project we performed Job Data analysis and Investigating Metric Spike analysis.

# APPROACH



1. Database creation: Database is created in MySQL workbench by importing datasets provided by company



2. Insights extraction: After creating database, required insights are generated from database tables by running queries in MySQL workbench.

# TECH-STACK USED



1. MySQL: To perform the analysis MySQL workbench version 8.0.33 is used as it is free and open source relational database management system that used SQL.



2. Canva: Canva is used to prepare this presentation

# JOB DATA ANALYSIS

## Task 1: Jobs Reviewed Over Time

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

The screenshot shows a MySQL Workbench interface. The SQL editor contains the following code:

```
2 • Select * from job_data;
3
4 -- task 1: Find number of jobs reviewed per hour for each day in November 2020 --
5
6 • SELECT
7     ds AS dates,
8     ROUND((COUNT(job_id) / SUM(time_spent)) * 3600) AS jobs_received_per_hr
9 FROM
10    job_data
11 GROUP BY ds;
```

The results grid displays the following data:

dates	jobs_received_per_hr
2020-11-30	180
2020-11-29	180
2020-11-28	218
2020-11-27	35
2020-11-26	64
2020-11-25	80

On date 2020-11-28, 218 Jobs were reviewed which was maximum.

# JOB DATA ANALYSIS

## Task 2 : Throughput Analysis

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

```
12  
13      -- Task 2: Find 7-day rolling average of throughput (number of events per second) --  
14  
15 •   SELECT  
16      COUNT(event_e) / SUM(time_spent) AS '7-day throughput'  
17      FROM  
18      job_data;  
19  
20
```

Result Grid	
<input type="button" value="Filter Rows:"/>	<input type="button" value="Export:"/>
7-day throughput	Wrap Cell Content:
▶ 0.0268	

7-day rolling average of throughput is 0.03

I would prefer 7-day rolling average for throughput as it is better to represent the overall performance also It helps to reduce the impact of daily fluctuations or noise in the data so it provides a clearer trend over time than daily metric.

# JOB DATA ANALYSIS

## Task 3 : Language Share Analysis

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

```
26 -- Task 3: Find percentage share of each language in the last 30 days. --
27
28 • SELECT
29     language_1 AS languages,
30     ROUND(SUM(time_spent) / (SELECT
31                         SUM(time_spent)
32                     FROM
33                         job_data) * 100) AS percentage_share
34     FROM
35         job_data
36     GROUP BY language_1
37     ORDER BY percentage_share DESC;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

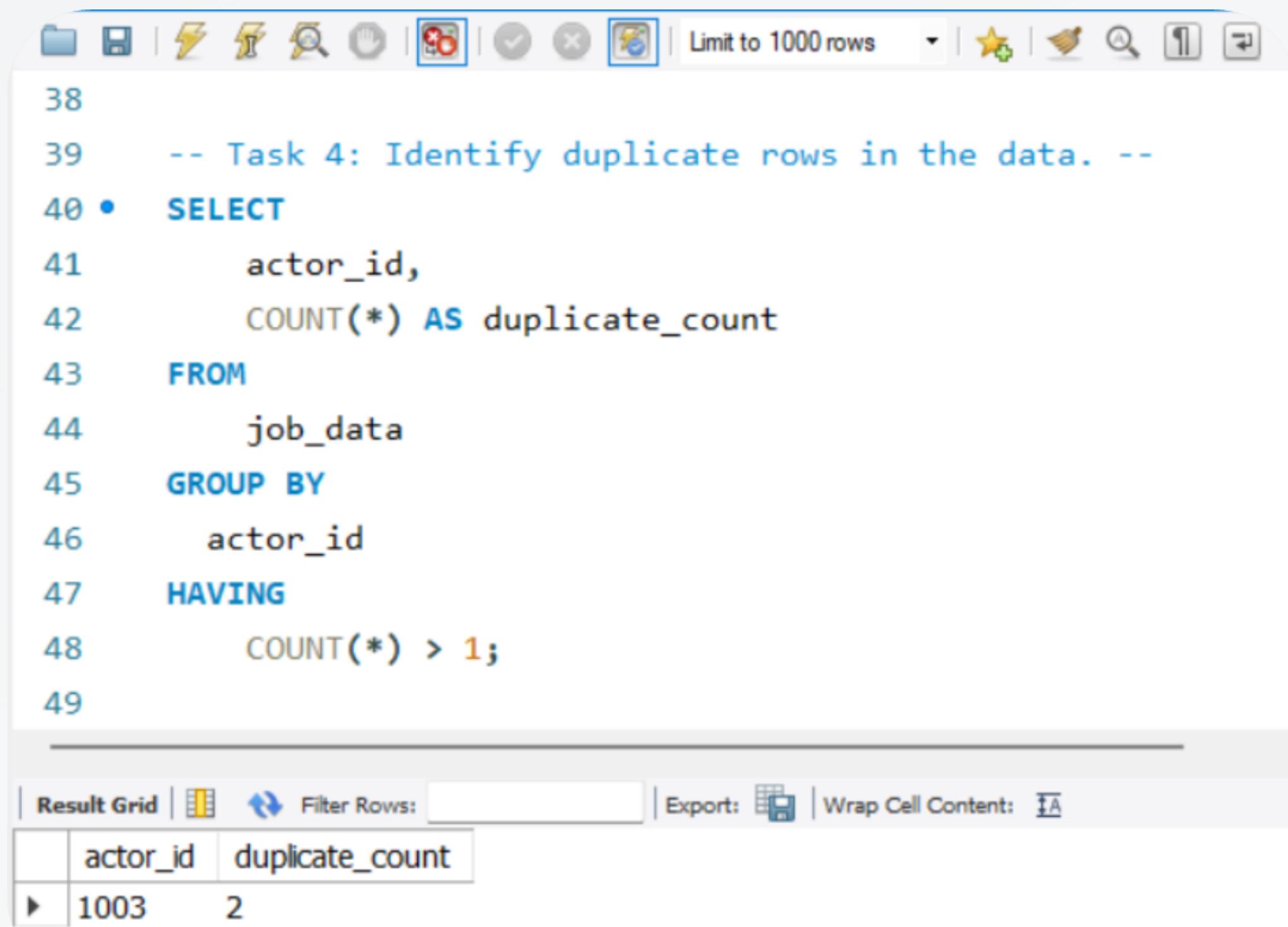
	languages	percentage_share
▶	French	35
	Persian	33
	Italian	15
	Arabic	8
	English	5
	Hindi	4

French language has highest percentage share 35%.

# JOB DATA ANALYSIS

## Task 4 : Duplicate Rows Detection

Identify duplicate rows in the data



The screenshot shows a MySQL query editor interface. At the top, there's a toolbar with various icons for file operations, search, and database management. Below the toolbar, the SQL query is displayed:

```
38
39 -- Task 4: Identify duplicate rows in the data.
40 • SELECT
41     actor_id,
42     COUNT(*) AS duplicate_count
43 FROM
44     job_data
45 GROUP BY
46     actor_id
47 HAVING
48     COUNT(*) > 1;
49
```

At the bottom of the editor, there are buttons for "Result Grid" and "Export". A result grid is shown below the editor area, containing the following data:

	actor_id	duplicate_count
▶	1003	2

Actor ID 1003 has duplicates rows in data

# INVESTIGATING METRIC SPIKE

## Task 1: Weekly User Engagement

Measure the activeness of users on a weekly basis

```
-- Task 1: Calculate activeness of users on a weekly basis. --  
  
SELECT  
    EXTRACT(WEEK FROM occurred_at) AS week_number,  
    COUNT(DISTINCT (user_id)) AS active_users  
FROM  
    events_table  
WHERE  
    event_type = 'engagement'  
GROUP BY week_number  
ORDER BY week_number ASC;
```

week_number	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

Week 30 has maximum active users while week 17 has minimum active users.

# INVESTIGATING METRIC SPIKE

## Task 2 : User Growth Analysis

Analyze the growth of users over time for a product.

```
-- Task 2: Analyze the growth of users over time for a product --
select year, week_number, users_number, sum(users_number)
over (order by year,week_number) as users_growth
from (
    select extract(year from created_at) as year,
           extract(week from created_at) as week_number,
           count(distinct user_id) as users_number from users
    where state ='active'
    group by year,week_number
    order by year,week_number) as temp;
```

Week 33 has highest user growth while week 35 has lowest user growth.

year	week_number	users_number	users_growth
2013	0	23	23
2013	1	30	53
2013	2	48	101
2013	3	36	137
2013	4	30	167
2013	5	48	215
2013	6	38	253
2013	7	42	295
2013	8	34	329
2013	9	43	372
2013	10	32	404
2013	11	31	435
2013	12	33	468
2013	13	39	507
2013	14	35	542
year	week_number	users_number	users_growth
2014	30	238	8405
2014	31	193	8598
2014	32	245	8843
2014	33	261	9104
2014	34	259	9363
2014	35	18	9381

# INVESTIGATING METRIC SPIKE

## Task 3 : Weekly Retention Analysis

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

```
-- Task 3: Analyze retention of users on weekly basis after signing up for a product. --
select first_ as 'week numb',
sum(case when week_numb= 0 then 1 else 0 end) as 'week 0',
sum(case when week_numb= 1 then 1 else 0 end) as 'week 1',
sum(case when week_numb= 2 then 1 else 0 end) as 'week 2',
sum(case when week_numb= 3 then 1 else 0 end) as 'week 3',
sum(case when week_numb= 4 then 1 else 0 end) as 'week 4',
sum(case when week_numb= 5 then 1 else 0 end) as 'week 5',
sum(case when week_numb= 6 then 1 else 0 end) as 'week 6',
sum(case when week_numb= 7 then 1 else 0 end) as 'week 7',
sum(case when week_numb= 8 then 1 else 0 end) as 'week 8',
sum(case when week_numb= 9 then 1 else 0 end) as 'week 9',
sum(case when week_numb= 10 then 1 else 0 end) as 'week 10',
sum(case when week_numb= 11 then 1 else 0 end) as 'week 11',
sum(case when week_numb= 12 then 1 else 0 end) as 'week 12',
sum(case when week_numb= 13 then 1 else 0 end) as 'week 13',
sum(case when week_numb= 14 then 1 else 0 end) as 'week 14',
sum(case when week_numb= 15 then 1 else 0 end) as 'week 15',
sum(case when week_numb= 16 then 1 else 0 end) as 'week 16',
sum(case when week_numb= 17 then 1 else 0 end) as 'week 17',
sum(case when week_numb= 18 then 1 else 0 end) as 'week 18'
from (
```

# INVESTIGATING METRIC SPIKE

```
select a.user_id, a.login_week, b.first_, a.login_week-b.first_ as week_numb  
from ( select user_id, extract(week from occurred_at) as login_week from events_table  
group by 1,2) a,  
(select user_id, min(extract(week from occurred_at)) as first_ from events_table  
group by 1) b  
where a.user_id=b.user_id) as temp  
group by first_  
order by first_;
```

# INVESTIGATING METRIC SPIKE

## Task 4: Weekly Engagement Per Device

Calculate activeness of users on a weekly basis per device.

```
-- Task 4: Calculate activeness of users on a weekly basis per device. --
SELECT
    EXTRACT(WEEK FROM occurred_at) AS weeks,
    device,
    COUNT(DISTINCT user_id) AS user_engagement
FROM
    events_table
GROUP BY device , weeks
ORDER BY weeks;
```

weeks	device	user_engagement	weeks	device	user_engagement
17	acer aspire desktop	9	17	lenovo thinkpad	86
17	acer aspire notebook	20	17	mac mini	6
17	amazon fire phone	4	17	macbook air	54
17	asus chromebook	21	17	macbook pro	143
17	dell inspiron desktop	18	17	nexus 10	16
17	dell inspiron notebook	46	17	nexus 5	40
17	hp pavilion desktop	14	17	nexus 7	18
17	htc one	16	17	nokia lumia 635	17
17	ipad air	27	17	samsung galaxy tablet	8
17	ipad mini	19	17	samsung galaxy note	7
17	iphone 4s	21	17	samsung galaxy s4	52
17	iphone 5	65	17	windows surface	10
17	iphone 5s	42	18	acer aspire desktop	26
17	kindle fire	6	18	acer aspire notebook	33

# INVESTIGATING METRIC SPIKE

## Task 5 : Email Engagement Analysis

Analyze how users are engaging with the email service.

-- Task 5: Analyze how users are engaging with the email service. --

```
SELECT week(occurred_at) as Week,  
       count( DISTINCT ( CASE WHEN actions = "sent_weekly_digest"  
                           THEN user_id end )) as weekly_digest,  
       count( distinct ( CASE WHEN actions = "sent_reengagement_email"  
                           THEN user_id end )) as reengagement_mail,  
       count( distinct ( CASE WHEN actions = "email_open"  
                           THEN user_id end )) as opened_email,  
       count( distinct ( CASE WHEN actions = "email_clickthrough"  
                           THEN user_id end )) as email_clickthrough  
  FROM email_events  
 GROUP BY week(occurred_at)  
 ORDER BY week(occurred_at);
```

Week	weekly_digest	reengagement_mail	opened_email	email_clickthrough
17	908	73	310	166
18	2602	157	900	425
19	2665	173	961	476
20	2733	191	989	501
21	2822	164	996	436
22	2911	192	965	478
23	3003	197	1057	529
24	3105	226	1136	549
25	3207	196	1084	524
26	3302	219	1149	550
27	3399	213	1207	613
28	3499	213	1228	594
29	3592	213	1201	583
30	3706	231	1363	625
31	3793	222	1338	444
32	3897	200	1318	416
33	4012	264	1417	490
34	4111	261	1502	481
35	0	48	41	38

Week 34 has highest engagement using email services while week 35 has lowest email engagement.

# INSIGHTS

- Around 218 Jobs were reviewed on 11 Nov 2010 which was maximum among other days.
- 7-day rolling average for throughput as it is better to represent the overall performance also it provides a clearer trend over time than daily metric.
- French language has highest percentage share (35%) among other languages.
- Week 30 has maximum active users while week 17 has minimum active users.
- Week 33 has highest user growth while week 35 has lowest user growth.
- Week 30 has maximum active users on device MacBook Pro while week 35 has minimum active users on device acer aspire desktop.
- Week 34 has highest engagement using email services while week 35 has lowest email engagement.



# RESULTS

This project helped me to advance my query writing techniques and problem solving ability. The problem I faced in this project was about importing large size Event CSV file into database which was taking lot of time to import so instead I learned to use load data statement to import large size csv files into database.

