OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE



Agenda

- Project Description
- Approach
- Tech-Stack Used
- Insights
- Result



PROJECT DESCRIPTION

- Operational Analytics focuses on investigating metric spikes and analyzing key business metrics to derive actionable insights. In this project, we assume the role of a Lead Data Analyst at a company like Microsoft, where we analyze datasets to answer crucial questions posed by different departments. We leverage advanced SQL techniques to identify trends and provide meaningful recommendations.
- This project focuses on two areas of analysis:
 - A) Job Data Analysis.
 - B) Investigating Metric Spikes.



TECH STACK USED

1.MySQL Workbench

1. Version: 8.0

2. Why I Used It:

MySQL Workbench made it super easy to manage the database and run SQL queries. Its simple interface and visual tools helped me analyze the data quickly and efficiently.

2.MySQL Server

1. Version: 8.0

2. Why I Used It:

MySQL Server is reliable and works smoothly with MySQL Workbench. It handled all the data without any issues, making the whole process hassle-free.

3. Operating System

1. Used: Windows 11

2. Why I Used It:

I worked on Windows 11 because it's easy to set up and works well with the tools I needed for the project.



APPROACH

- To do this project the team has given 1 dataset for case-study 1 and 3 datasets for case-study 2
- At first I have created a database called **job_data** then I have imported the 4 datasets called
- job_data as jobsdata,
- users,
- events,
- email_events in Table data import wizard and then I have gone through the columns and changed few columns datatypes into date format.



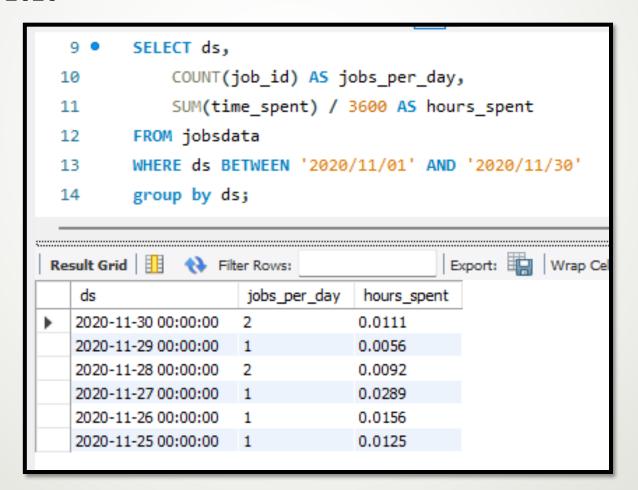
CASE STUDY 1: JOB DATA ANALYSIS

1. Write SQL Queries for the given Questions:

☐ Jobs Reviewed Over Time:

Your Task: Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020

ANS:





☐ Throughput Analysis:

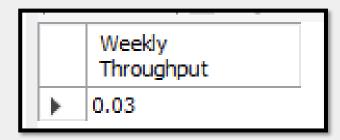
Your Task: Calculate the 7-day rolling average of throughput (number of events per second)

The task was to find the 7-day rolling average of throughput and also daily metric. I would prefer the 7-day rolling average because it gives proper insights for the company future than the daily metric.

ANS:

```
SELECT ds, ROUND(COUNT(event) /SUM(time_spent), 2) AS "Daily Throughput" FROM jobsdata
GROUP BY ds
ORDER BY ds;
SELECT ROUND(COUNT(event) /(SUM(time_spent)), 2) AS "Weekly Throughput" FROM jobsdata;
```

	ds	Daily Throughput
•	2020-11-25 00:00:00	0.02
	2020-11-26 00:00:00	0.02
	2020-11-27 00:00:00	0.01
	2020-11-28 00:00:00	0.06
	2020-11-29 00:00:00	0.05
	2020-11-30 00:00:00	0.05

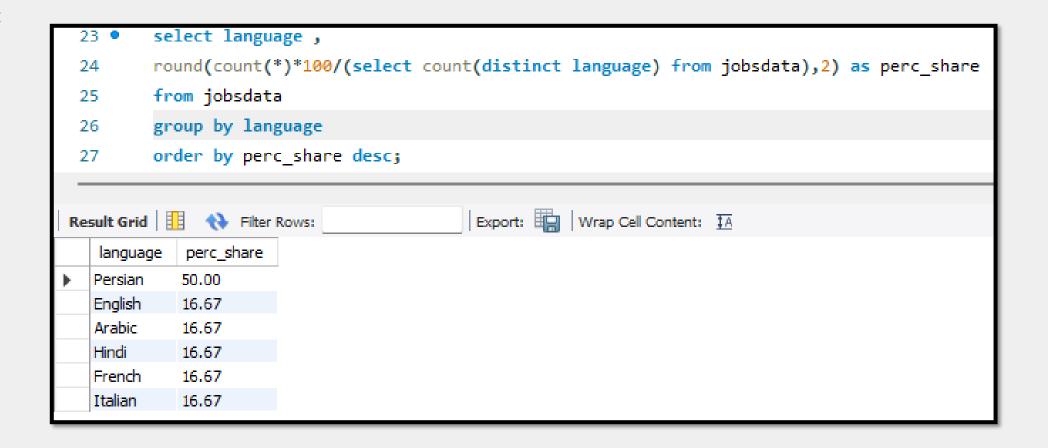




□ Language Share Analysis:

Your Task: Write an SQL query to calculate the percentage share of each language over the last 30 days

ANS:





☐ **Duplicate Rows Detection :**Your Task: Write an SQL query to display duplicate rows from the job_data table

ANS:

There are no duplicates in the table

```
SELECT job_id, actor_id, event, language, time_spent,
 31
            org, ds, COUNT(*) AS duplicate count
        FROM jobsdata
 32
        GROUP BY job_id, actor_id, event, language, time_spent, org, ds
 33
        HAVING COUNT(*) > 1;
 34
Result Grid
                                          Export: Wrap Cell Content: IA
              Filter Rows:
                  event language
                                                        duplicate_count
        actor_id
                                  time_spent
```



Case Study 2: Investigating Metric Spike

Here we will work on the table's users, events, email_events

1. 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.

ANS:

92 • select extract(week from occurred_at) as week_no,

93 count(distinct user_id) as active_user from events

94 group by week_no

95 order by week_no;

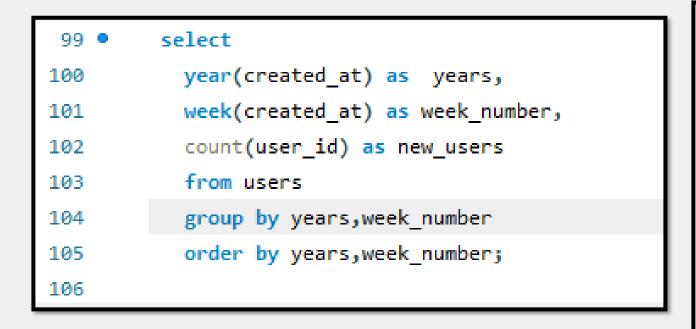
Re	sult Grid	Note:
	week_no	active_user
•	17	136
	18	278
	19	300
	20	286
	21	302
	22	335
	23	316
	24	358
	25	340
	26	318
	27	347
	28	344
	29	350
	30	377
	31	312
	32	339
	33	356
	34	360
	35	21

2.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

ANS:



	years	week_number	new_users
>	2013	0	23
	2013	1	30
	2013	2	48
	2013	3	36
	2013	4	30
	2013	5	48
	2013	6	38
	2013	7	42
	2013	8	34
	2013	9	43
	2013	10	32
	2013	11	31
	2013	12	33
	2013	13	39
	2013	14	35
	2013	15	43
	2013	16	46
	2013	17	49
	2013	18	44
	2013	19	57
	2013	20	39
	2013	21	49
	2013	22	54
	2013	23	50
	2013	24	45

years	week_number	new_users
2013	25	57
2013	26	56
2013	27	52
2013	28	72
2013	29	67
2013	30	67
2013	31	67
2013	32	71
2013	33	73
2013	34	78
2013	35	63
2013	36	72
2013	37	85
2013	38	90
2013	39	84
2013	40	87
2013	41	73
2013	42	99
2013	43	89
2013	44	96
2013	45	91
2013	46	88
2013	47	102
2013	48	97
2013	49	116

3. Weekly Retention Analysis:

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

B. Your Task: Write an SQL query to calculate the weekly retention of users based on their sign-

up cohort.

```
109 • \ominus WITH cohort AS (
             SELECT
110
111
                user_id,
                WEEK(created_at) AS sign_up_week,
112
                YEAR(created_at) AS sign_up_year
113
114
             FROM users
115
        activity AS (
116
117
             SELECT
118
                user_id,
                WEEK(activated at) AS activity week,
119
                YEAR(activated_at) AS activity_year
120
121
             FROM users
            WHERE activated at IS NOT NULL
122
123
124
        SELECT
125
            c.sign_up_year,
            c.sign up week,
126
            COUNT(DISTINCT a.user_id) AS active_users_in_week,
127
            COUNT(DISTINCT c.user_id) AS total_signups_in_week,
128
            ROUND(COUNT(DISTINCT a.user id) / COUNT(DISTINCT c.user id) * 100, 2) AS retention percentage
129
        FROM cohort c
130
        LEFT JOIN activity a
131
            ON c.user_id = a.user_id
132
            AND a.activity year = c.sign_up year
133
            AND a.activity week >= c.sign up week
134
        GROUP BY c.sign_up_year, c.sign_up_week
135
        ORDER BY c.sign up year, c.sign up week;
136
```



	sign_up_year	sign_up_week	active_users_in_week	total_signups_in_week	retention_percentage
•	2013	0	23	23	100.00
	2013	1	30	30	100.00
	2013	2	48	48	100.00
	2013	3	36	36	100.00
	2013	4	30	30	100.00
	2013	5	48	48	100.00
	2013	6	38	38	100.00
	2013	7	42	42	100.00
	2013	8	34	34	100.00
	2013	9	43	43	100.00
	2013	10	32	32	100.00
	2013	11	31	31	100.00
	2013	12	33	33	100.00
	2013	13	39	39	100.00
	2013	14	35	35	100.00
	2013	15	43	43	100.00
	2013	16	46	46	100.00
	2013	17	49	49	100.00
	2013	18	44	44	100.00
	2013	19	57	57	100.00
	2013	20	39	39	100.00
	2013	21	49	49	100.00
	2013	22	54	54	100.00
	2013	23	50	50	100.00
	2013	24	45	45	100.00
	2013	25	57	57	100.00
	2013	26	56	56	100.00

I				
sign_up_year	sign_up_week	active_users_in_week	total_signups_in_week	retention_percentage
2013	27	52	52	100.00
2013	28	72	72	100.00
2013	29	67	67	100.00
2013	30	67	67	100.00
2013	31	67	67	100.00
2013	32	71	71	100.00
2013	33	73	73	100.00
2013	34	78	78	100.00
2013	35	63	63	100.00
2013	36	72	72	100.00
2013	37	85	85	100.00
2013	38	90	90	100.00
2013	39	84	84	100.00
2013	40	87	87	100.00
2013	41	73	73	100.00
2013	42	99	99	100.00
2013	43	89	89	100.00
2013	44	96	96	100.00
2013	45	91	91	100.00
2013	46	88	88	100.00
2013	47	102	102	100.00
2013	48	97	97	100.00
2013	49	116	116	100.00
2013	50	124	124	100.00
2013	51	102	102	100.00
2013	52	47	47	100.00
2014	0	83	83	100.00

sign_up_year	sign up week	active_users_in_week	total_signups_in_week	retention_percentage
	sign_up_week 0	83	83	
2014	_			100.00
2014	1	126	126	100.00
2014	2	109	109	100.00
2014	3	113	113	100.00
2014	4	130	130	100.00
2014	5	133	133	100.00
2014	6	135	135	100.00
2014	7	125	125	100.00
2014	8	129	129	100.00
2014	9	133	133	100.00
2014	10	154	154	100.00
2014	11	130	130	100.00
2014	12	148	148	100.00
2014	13	167	167	100.00
2014	14	162	162	100.00
2014	15	164	164	100.00
2014	16	179	179	100.00
2014	17	170	170	100.00
2014	18	163	163	100.00
2014	19	185	185	100.00
2014	20	176	176	100.00
2014	21	183	183	100.00
2014	22	196	196	100.00
2014	23	196	196	100.00
2014	24	229	229	100.00
2014	25	207	207	100.00
2014	26	201	201	100.00

2014	27	222	222	100.00
2014	28	215	215	100.00
2014	29	221	221	100.00
2014	30	238	238	100.00
2014	31	193	193	100.00
2014	32	245	245	100.00
2014	33	261	261	100.00
2014	34	259	259	100.00
2014	35	18	18	100.00



4. Weekly Engagement Per Device:

- A. Objective: Measure the activeness of users on a weekly basis per device.
- B. Your Task: Write an SQL query to calculate the weekly engagement per device.

ANS:

```
select extract(week from occurred_at) as week_nos,device,

count(distinct user_id) as active_user from events

group by device,week_nos

order by week_nos,device;
```

	week_nos	device	active_user
•	17	acer aspire desktop	2
	17	acer aspire notebook	4
	17	amazon fire phone	1
	17	asus chromebook	6
	17	dell inspiron desktop	2
	17	dell inspiron notebook	7
	17	hp pavilion desktop	4
	17	htc one	3
	17	ipad air	3
	17	ipad mini	4
	17	iphone 4s	5
	17	iphone 5	16
	17	iphone 5s	10
	17	lenovo thinkpad	15
17		mac mini	2
	17	macbook air	6
	17	macbook pro	29
	17	nexus 10	1
	17	nexus 5	6
	17	nexus 7	5
	17	nokia lumia 635	4
	17	samsumg galaxy tablet	2
	17	samsung galaxy note	3
	17	samsung galaxy s4	12
	18	acer aspire desktop	5
	18	acer aspire notebook	7

week_nos	device	active_user
18	amazon fire phone	3
18	asus chromebook	5
18	dell inspiron desktop	6
18	dell inspiron notebook	18
18	hp pavilion desktop	9
18	htc one	3
18	ipad air	10
18	ipad mini	9
18	iphone 4s	9
18	iphone 5	18
18	iphone 5s	14
18	kindle fire	7
18	lenovo thinkpad	46
18	mac mini	2
18	macbook air	30
18	macbook pro	62
18	nexus 10	5
18	nexus 5	11
18	nexus 7	7
18	nokia lumia 635	8
18	samsung galaxy note	6
18	samsung galaxy s4	20
18	windows surface	2
19	acer aspire desktop	3
19	acer aspire notebook	14
19	amazon fire phone	5

week_no	os device	active_user
19	asus chromebook	4
19	dell inspiron desktop	3
19	dell inspiron notebook	15
19	hp pavilion desktop	8
19	htc one	7
19	ipad air	14
19	ipad mini	4
19	iphone 4s	11
19	iphone 5	27
19	iphone 5s	19
19	kindle fire	4
19	lenovo thinkpad	44
19	mac mini	7
19	macbook air	32
19	macbook pro	57
19	nexus 10	5
19	nexus 5	24
19	nexus 7	12
19	nokia lumia 635	6
19	samsung galaxy note	6
19	samsung galaxy s4	23
19	windows surface	4
20	acer aspire desktop	2
20	acer aspire notebook	6
20	amazon fire phone	1
20	asus chromebook	9



5. Email Engagement Analysis:

- A. Objective: Analyze how users are engaging with the email service.
- B. Your Task: Write an SQL query to calculate the email engagement metrics.

ANS:

```
147 • ⊖ WITH weekly_stats AS (
148
            SELECT
                YEAR(occurred at) AS activity year,
149
                WEEK(occurred at) AS activity week,
150
151
                user_id,
                MAX(action = 'sent weekly digest') AS sent weekly digest,
152
                MAX(action = 'email_open') AS email_opened,
153
                MAX(action = 'email clickthrough') AS email clicked
154
155
            FROM email events
            WHERE action IN ('sent_weekly_digest', 'email_open', 'email_clickthrough')
156
            GROUP BY activity year, activity week, user_id
157
158
159
        SELECT
160
            activity year,
161
            activity week,
            SUM(sent_weekly_digest) AS emails_sent,
162
            SUM(email opened) AS emails opened,
163
164
            SUM(email_clicked) AS emails_clicked,
            ROUND((SUM(email opened) * 100.0 / NULLIF(SUM(sent_weekly_digest), 0)), 2) AS open_rate,
165
            ROUND((SUM(email_clicked) * 100.0 / NULLIF(SUM(sent_weekly_digest), 0)), 2) AS click_through_rate
166
        FROM weekly stats
167
168
        GROUP BY activity year, activity week
        ORDER BY activity year, activity week;
169
```



	activity_year	activity_week	emails_sent	emails_opened	emails_clicked	open_rate	click_through_rate
•	2014	17	908	310	166	34.14	18.28
	2014	18	2602	900	425	34.59	16.33
	2014	19	2665	961	476	36.06	17.86
	2014	20	2733	989	501	36.19	18.33
	2014	21	2822	996	436	35.29	15.45
	2014	22	2911	965	478	33.15	16.42
	2014	23	3003	1057	529	35.20	17.62
	2014	24	3105	1136	549	36.59	17.68
	2014	25	3207	1084	524	33.80	16.34
	2014	26	3302	1149	550	34.80	16.66
	2014	27	3399	1207	613	35.51	18.03
	2014	28	3499	1228	594	35.10	16.98
	2014	29	3592	1201	583	33.44	16.23
	2014	30	3706	1363	625	36.78	16.86
	2014	31	3793	1338	444	35.28	11.71
	2014	32	3897	1318	416	33.82	10.67
	2014	33	4012	1417	490	35.32	12.21
	2014	34	4111	1502	481	36.54	11.70
	2014	35	0	41	38	NULL	NULL



INSIGHTS

• CASE STUDY-1:Job Data Analysis

- A. Jobs Reviewed Over Time: I identified the hours when the highest number of jobs were reviewed.
- B. Throughput Analysis: I calculated the 7-day rolling average to understand the changes in job processing speed.
- C. Language Share Analysis: I found the percentage share of all languages used in job reviewing.
- D. Duplicate Rows Detection: I learned how to find the duplicate rows in the dataset to decrease the data redundancy.



• CASE STUDY-2:Investigating Metric Spikes

- A. Weekly User Engagement: I measured the activeness of user on weekly basis.
- B. User Growth Analysis: I found out how many new users are registered per year and per week.
- C. Weekly Retention Analysis: I calculated how long users stay active after signing up.
- D. Weekly Engagement Per Device: I measured how users are engaged through different devices.
- E. Email Engagement Analysis: I analysed email engagement to determine how users are using the services.



RESULT

By the Project ,I have analysed Job data, User engagement metrics and extracted the actionable insights for given questions.

I Identified areas of operational improvement and also improved my advanced SQL skills, learned new functions.

This project had good impact on improving decision making for operations , marketing, and support teams.





