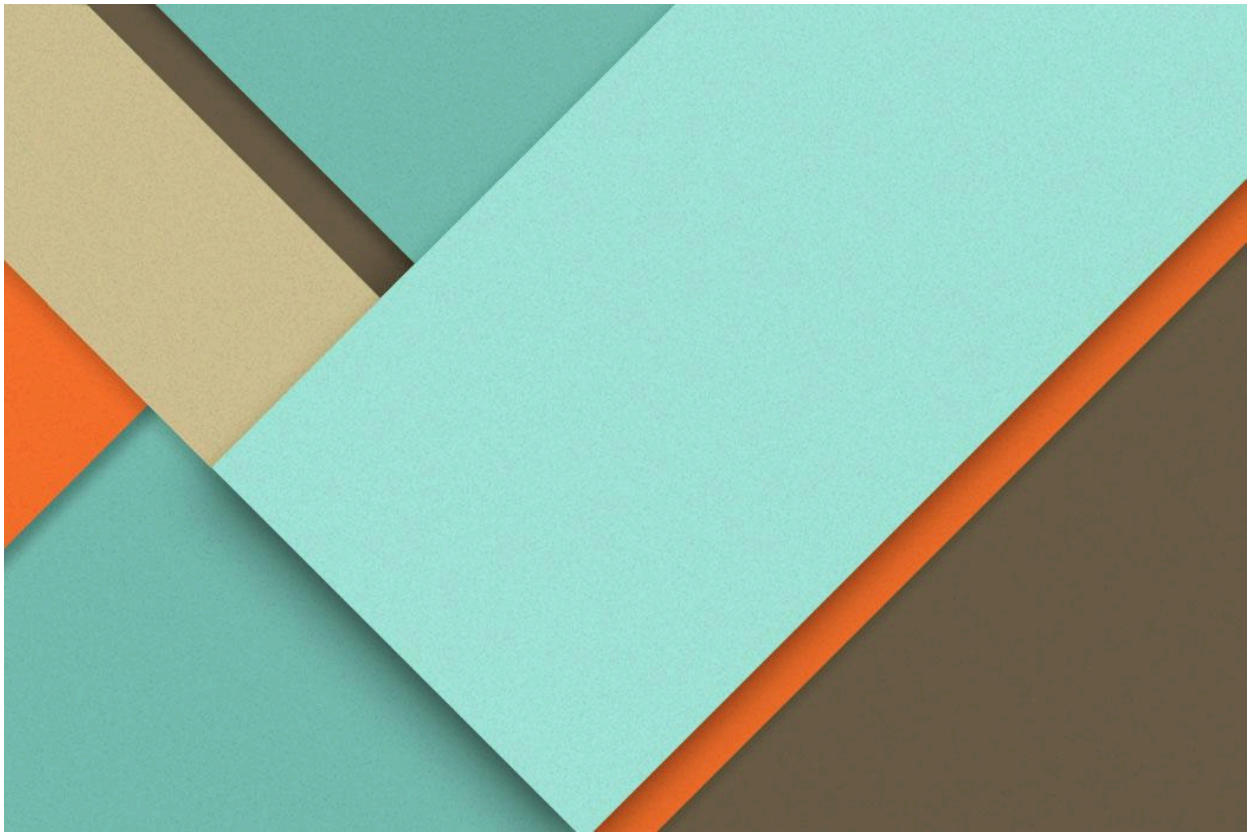


Operation Analytics and Investigating Metric Spike

Case Study 1: Job Data Analysis

Case Study 2: Investigating Metric Spike

Project Description

In this project, we will leverage advanced SQL skills and queries to address operational challenges, enabling us to effectively monitor and respond to sudden changes in key metrics. This approach will provide valuable insights into performance fluctuations and help in proactively identifying and solving potential issues.

Methodology and Technology Stack Implemented

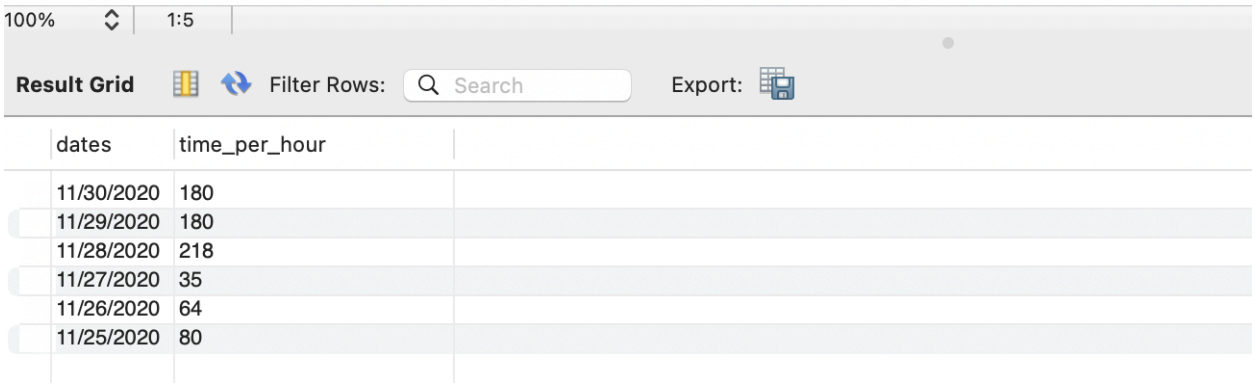
1. Methodology-
    - Conduct a thorough review of the tasks and get familiar with the provided data file by importing it into MySQL for in-depth analysis. This will ensure a clear understanding of the data structure and content before proceeding with further tasks.
    - Following the creation of the database, the necessary insights are generated from the database tables by executing queries in MySQL Workbench.
2. Technology Stack Implemented-
    - In order to perform the task, MySQL Workbench has been used, as it is considered to be a idol software to run the

Case Study 1: Job Data Analysis

Jobs Reviewed Over Time

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

```sql
SELECT ds AS dates,
ROUND((COUNT(job_id) / SUM(time_spent))*3600) AS time_per_hour
FROM records
GROUP BY dates
```



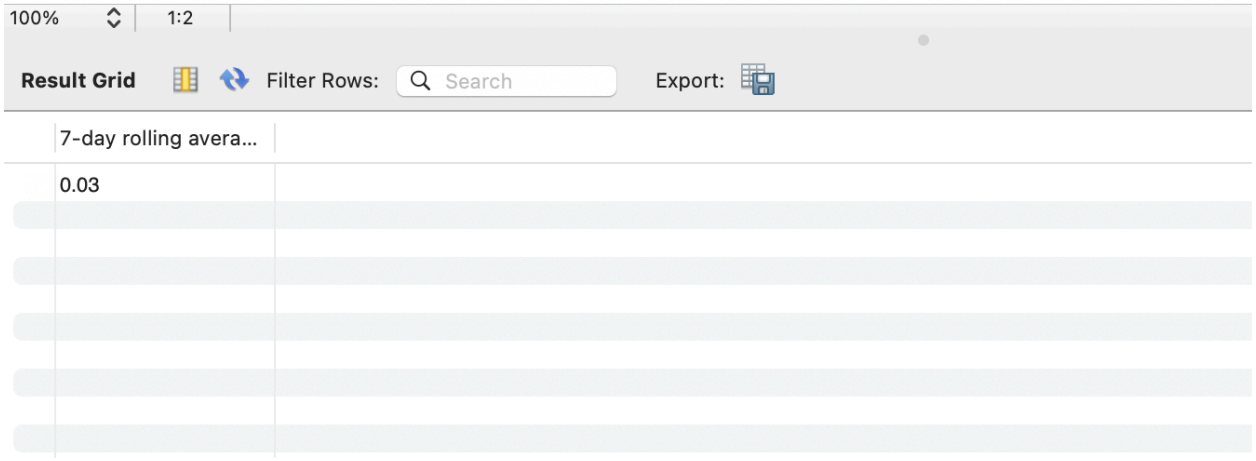
100% 1:5

Result Grid Filter Rows: Search Export:

dates	time_per_hour
11/30/2020	180
11/29/2020	180
11/28/2020	218
11/27/2020	35
11/26/2020	64
11/25/2020	80

Throughput Analysis:

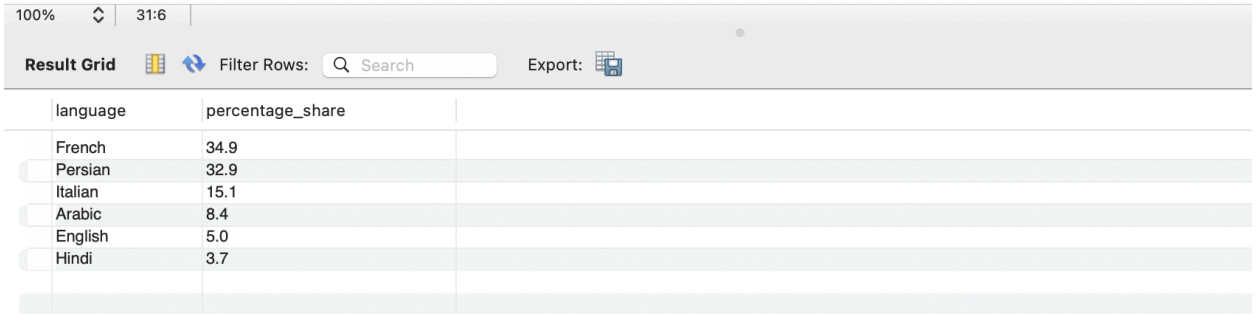
```sql
SELECT
ROUND(COUNT(event)/SUM(time_spent),2) AS `7-day rolling average`
FROM records;
```



Calculating a 7-day rolling average provides a more comprehensive and clearer view compared to using daily metrics alone.

Language Share Analysis

```sql
SELECT
    language,
    ROUND(SUM(time_spent) / (SELECT SUM(time_spent) FROM records) * 100, 1) AS percentage_share
FROM records
GROUP BY language
ORDER BY percentage_share DESC
```

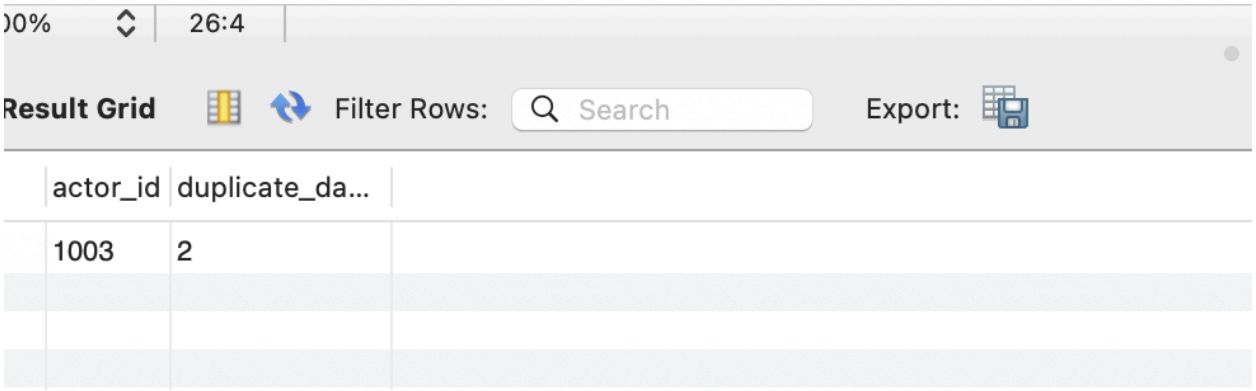


language	percentage_share
French	34.9
Persian	32.9
Italian	15.1
Arabic	8.4
English	5.0
Hindi	3.7

The highest percentage share is 34.9% for French, while the lowest percentage share is 3.7% for Hindi.

Duplicate Rows Detection

```
1  SELECT actor_id,COUNT(*) AS duplicate_data
2  FROM records
3  GROUP BY actor_id
4  HAVING duplicate_data > 1
```



Actor ID 1003 has duplicate entries in the database.

Case Study 2: Investigating Metric Spike

Weekly User Engagement

```sql
SELECT EXTRACT(WEEK FROM occurred_at) AS week_number,
       COUNT(DISTINCT user_id) AS active_users
FROM events
WHERE event_type = "engagement"
GROUP BY week_number
ORDER BY active_users DESC
```

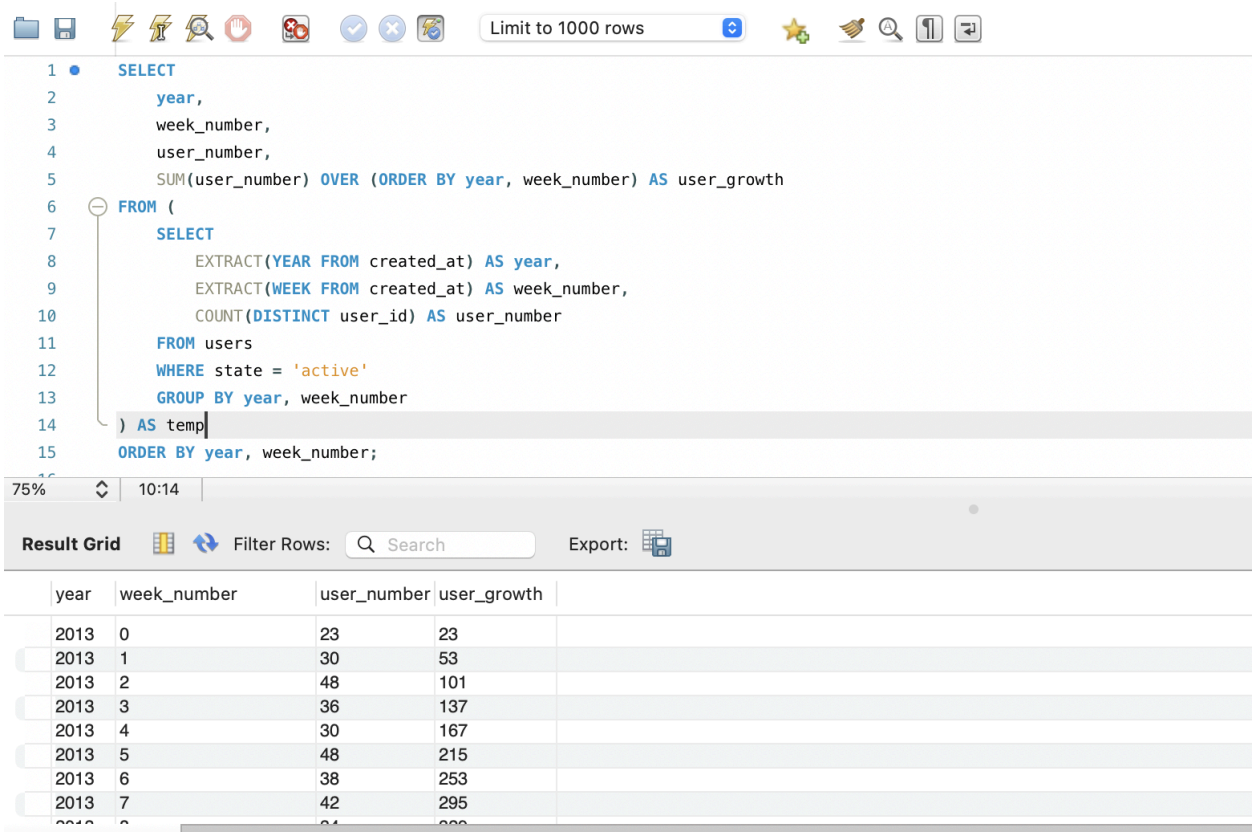
100% 27:6

Result Grid Filter Rows: Search Export:

week_number	active_users
30	1467
29	1376
27	1372
28	1365
26	1302
31	1299
24	1275
25	1264
23	1232
32	1225
33	1225
34	1204
22	1186
20	1154

Week 30 has the highest number of users, while week 35 sees the lowest user activity.

User Growth Analysis



```sql
SELECT
    year,
    week_number,
    user_number,
    SUM(user_number) OVER (ORDER BY year, week_number) AS user_growth
FROM (
    SELECT
        EXTRACT(YEAR FROM created_at) AS year,
        EXTRACT(WEEK FROM created_at) AS week_number,
        COUNT(DISTINCT user_id) AS user_number
    FROM users
    WHERE state = 'active'
    GROUP BY year, week_number
) AS temp
ORDER BY year, week_number;
```

Result Grid

year	week_number	user_number	user_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

Week 35 of 2014 saw the highest user growth rate on record.

Weekly Retention Analysis

To analyze weekly retention effectively, we want to measure the number of users who return each week following their sign-up week. Here's a streamlined query with clear steps for a **Weekly Retention Analysis**

```sql
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 (
    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 user_id, login_week
    ) a
    JOIN (
        SELECT
            user_id,
            MIN(EXTRACT(WEEK FROM occurred_at)) AS first_
        FROM events_table
        GROUP BY user_id
    ) b ON a.user_id = b.user_id
) AS temp
GROUP BY first_
ORDER BY first_;
```

week_numb	week 0	week 1	week 2	week 3	week 4	week 5	week 6	week 7	week 8	week 9	week 10	week 11	week 12	week 13	week 14	week 15	week 16	week 17	week 18
17	663	472	324	251	205	187	167	146	145	145	136	131	132	143	116	91	82	77	5
18	596	362	261	203	168	147	144	127	113	122	106	118	127	110	97	85	67	4	0
19	427	284	173	153	114	95	91	81	95	82	68	65	63	42	51	49	2	0	0
20	358	223	165	121	91	72	63	67	63	65	67	41	40	33	40	0	0	0	0
21	317	187	131	91	74	63	75	72	58	48	45	39	35	28	2	0	0	0	0
22	326	224	150	107	87	73	63	60	55	48	41	39	31	1	0	0	0	0	0
23	328	219	138	101	90	79	69	61	54	47	35	30	0	0	0	0	0	0	0
24	339	205	143	102	81	63	65	61	38	39	29	0	0	0	0	0	0	0	0
25	305	218	139	101	75	63	50	46	38	35	2	0	0	0	0	0	0	0	0
26	288	181	114	83	73	55	47	43	29	0	0	0	0	0	0	0	0	0	0
27	292	199	121	106	68	53	40	36	1	0	0	0	0	0	0	0	0	0	0
28	274	194	114	69	46	30	28	3	0	0	0	0	0	0	0	0	0	0	0
29	270	186	102	65	47	40	1	0	0	0	0	0	0	0	0	0	0	0	0
30	294	202	121	78	53	3	0	0	0	0	0	0	0	0	0	0	0	0	0
31	215	145	76	57	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	267	188	94	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	286	202	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	279	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Examine weekly user retention following their initial product sign-up.

Weekly Engagement Per Device

```sql
SELECT EXTRACT(WEEK FROM occurred_at) AS weeks,
       device,
       COUNT(DISTINCT user_id) AS weekly_user_engagement
FROM events_table
GROUP BY weeks, device
ORDER BY weekly_user_engagement DESC;
```

75% 37:6

Result Grid Filter Rows: Search Export:

weeks	device	weekly_user_engageme...
30	macbook pro	322
31	macbook pro	321
33	macbook pro	312
32	macbook pro	307
27	macbook pro	302
28	macbook pro	295
29	macbook pro	295

Week 30 saw the highest number of active users on the MacBook Pro, while Week 35 had the lowest number of active users on the Acer Aspire desktop.

Email Engagement Analysis

```sql
SELECT WEEK(occurred_at) AS week_number,
       COUNT(DISTINCT CASE WHEN action = 'sent_weekly_digest' THEN user_id END) AS weekly_digest,
       COUNT(DISTINCT CASE WHEN action = 'sent_reengagement_email' THEN user_id END) AS reengagement_email,
       COUNT(DISTINCT CASE WHEN action = 'email_open' THEN user_id END) AS email_open,
       COUNT(DISTINCT CASE WHEN action = 'email_clickthrough' THEN user_id END) AS email_clickthrough
FROM email_events
GROUP BY WEEK(occurred_at)
ORDER BY week_number;
```

22:8

Result Grid Filter Rows: Search Export:

week_number	weekly_digest	reengagement_email	email_open	email_clickthrou...
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

Result 23

Week 34 had the highest engagement with email services, while Week 35 recorded the lowest email engagement.

Weekly Performance and Engagement Insights

- On November 11, 2010, approximately 218 jobs were reviewed, the highest count compared to other days.

- A 7-day rolling average is recommended for throughput, as it better reflects overall performance and highlights trends more effectively than daily metrics.

- The French language holds the largest percentage share among languages, with 35%.

- Week 30 recorded the highest number of active users, while Week 17 had the fewest.

- User growth peaked in Week 33, with Week 35 experiencing the lowest growth rate.

- The highest number of active users on the MacBook Pro was observed in Week 30, whereas Week 35 saw the lowest number of active users on the Acer Aspire desktop.

- Engagement with email services was highest in Week 34 and lowest in Week 35.

Result

This project enhanced my query-writing skills and strengthened my problem-solving abilities. One of the challenges I encountered was the time required to clean and prepare the data to ensure it could be fully utilized.