Operation Analytics and Investigating Metric Spike

PROJECT DESCRIPTION

This project analyzes job data and investigates metric spikes to improve a company's operations. I have uncovered valuable insights using SQL queries, such as **user engagement**, **retention rates**, and **workflow optimizations**. These data-driven findings will guide better decision-making and enhance overall performance.

APPROACH

1. **Data Understanding:**
   I carefully **reviewed the data sets**, grasping table structures, and column meanings.
2. **SQL Analysis:**
   Utilizing **SQL queries**, I **extracted valuable information** and insights from the data.
3. **Case Study 1:**
   Calculated metrics like **jobs reviewed per hour, 7-day rolling average of throughput**, and **language percentage share**.
4. **Case Study 2:**
   Analyzed user data to derive metrics such as **weekly engagement, growth**, and **email engagement**.
5. **Data Visualization:**
   Presented findings **using charts and graphs** for clear and easy understanding.

TECH-STACK USED

I have mainly used 3 tools:-

- MS Excel (Microsoft 365)
  I used it for visualization purposes.
- MySQL Workbench 8.0.33
  I used it for analyzing the dataset provided.
- PowerPoint (Microsoft 365)
  I used it for the presentation.

INSIGHTS

Case Study 1: Job Data Analysis

A. Jobs reviewed overtime
B. Throughput analysis
C. Language share analysis
D. Duplicate rows detection

INSIGHTS

Case Study 2: Investigating Metric Spike

A. Weekly User Engagement
B. User Growth Analysis
C. Weekly Retention Analysis
D. Weekly Engagement Per Device
E. Email Engagement Analysis

INSIGHTS

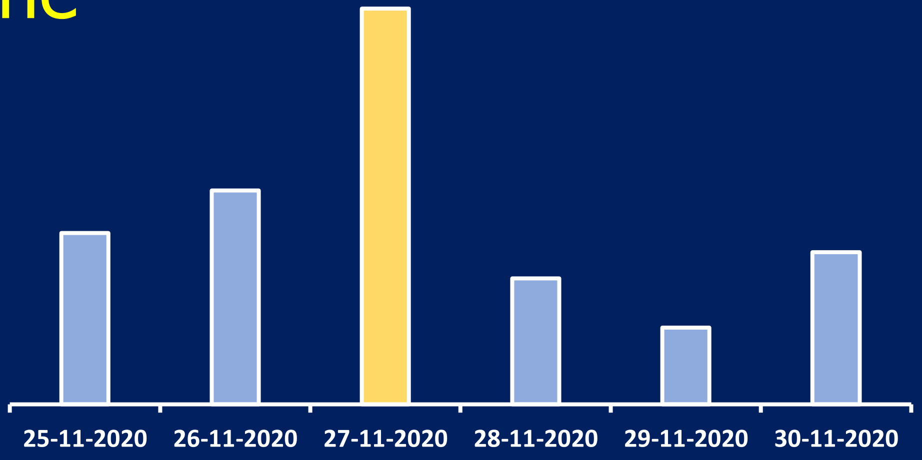
Case Study 1: Job Data Analysis

A. Jobs reviewed overtime

The data shows the number of jobs reviewed per day and the time taken to review the jobs per hour for each day. It shows that on 27th November 2020 one job took maximum time to review.

```sql
SELECT
    ds,
    COUNT(job_id) AS No_of_jobs,
    SUM(time_spent)/3600 AS per_hour_per_day
FROM
    job_data
GROUP BY ds;
```

ds	No_of_jobs	per_hour_per_day
25-11-2020	1	0.0125
26-11-2020	1	0.0156
27-11-2020	1	0.0289
28-11-2020	2	0.0092
29-11-2020	1	0.0056
30-11-2020	2	0.0111



INSIGHTS

Case Study 1: Job Data Analysis

B. Throughput Analysis

The data shows the throughput and 7-day rolling average of the throughput. I would prefer 7-day rolling average of throughput rather than daily metric because it provides a more stable trend over time, indicating the overall efficiency of the job review process.

```sql
SELECT
    ds,
    COUNT(job_id) AS jobs_reviewed,
    SUM(time_spent) AS Time_taken_to_review,
    COUNT(job_id) / SUM(time_spent) AS throughput,
    AVG(COUNT(job_id) / SUM(time_spent)) over(ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) as Rolling_7_day
FROM
    job_data
GROUP BY ds;
```

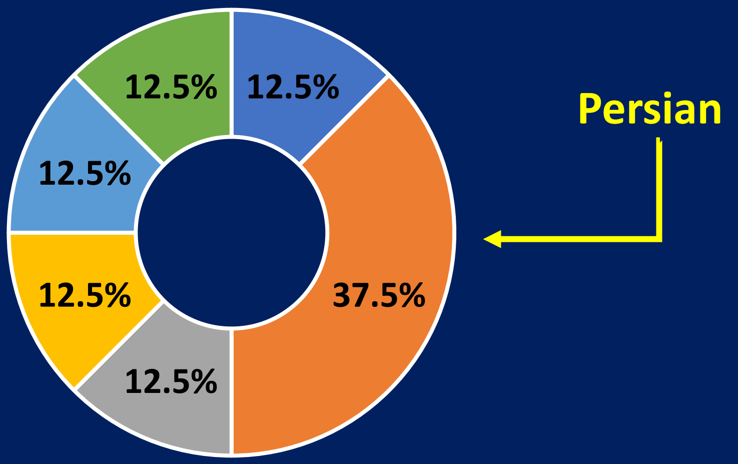
ds	jobs_reviewed	Time_taken_to_review	throughput	Rolling_7_day
25-11-2020	1	45	0.0222	0.02222222
26-11-2020	1	56	0.0179	0.02003968
27-11-2020	1	104	0.0096	0.01656492
28-11-2020	2	33	0.0606	0.02757520
29-11-2020	1	20	0.0500	0.03206016
30-11-2020	2	40	0.0500	0.03505013

INSIGHTS

Case Study 1: Job Data Analysis

C. Language share analysis

With this query, I identified that Persian was the most used language within the last 30 days. It holds 37.5% of the total share among all the languages.




```sql
WITH Count_lang AS( SELECT  language,
                            COUNT(job_id) AS Lang_used
                    FROM job_data
                    GROUP BY language
                  )
SELECT  language,
        Lang_used,
        ROUND((Lang_used/(SELECT COUNT(*) FROM job_data))*100,2) AS Percentage
FROM Count_lang;
```

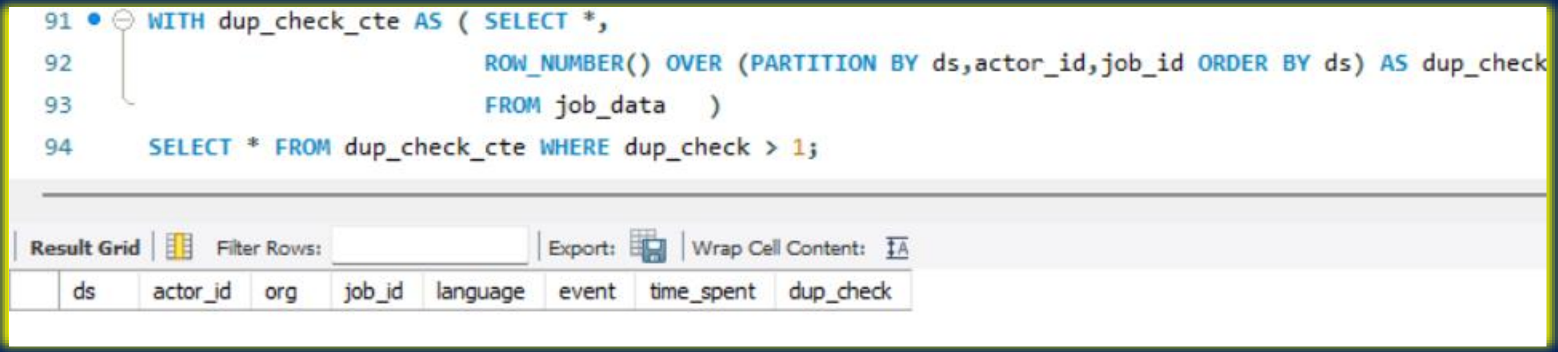
language	Lang_used	Percentage
Italian	1	12.50
Persian	3	37.50
French	1	12.50
Hindi	1	12.50
English	1	12.50
Arabic	1	12.50

INSIGHTS

Case Study 1: Job Data Analysis

D. Duplicate rows detection

From the below query I identified that there is no duplicate rows in job_data table.



```
WITH dup_check_cte AS ( SELECT *,
                        ROW_NUMBER() OVER (PARTITION BY ds,actor_id,job_id ORDER BY ds) AS dup_check
                        FROM job_data   )
SELECT * FROM dup_check_cte WHERE dup_check > 1;
```

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

ds	actor_id	org	job_id	language	event	time_spent	dup_check

INSIGHTS

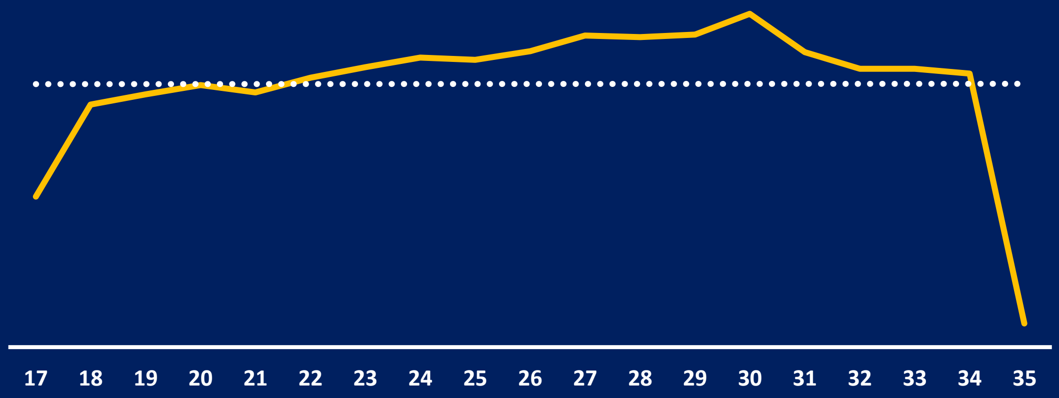
Case Study 2: Investigating Metric Spike

A. Weekly user engagement

```
SELECT
    EXTRACT(WEEK FROM STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i')) AS Week_no,
    COUNT(DISTINCT event.user_id) AS engagement
FROM
    event LEFT JOIN user ON event.user_id = user.user_id
WHERE state = 'active'
GROUP BY Week_no;
```

The information that I got from the data using this query is that the engagement was constant but there was a sudden dip in engagement in week 35.

Week_no	engagement
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
32	1225
33	1225
34	1204
35	104



INSIGHTS

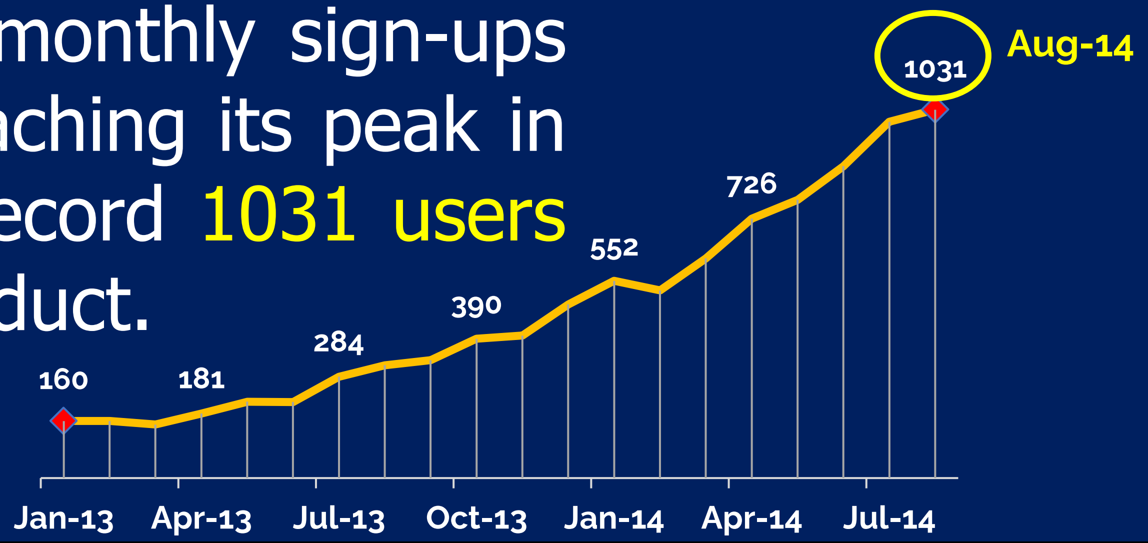
Case Study 2: Investigating Metric Spike

B. User growth analysis

```
SELECT
    DATE_FORMAT(STR_TO_DATE(created_at, '%d-%m-%Y %H:%i'),"%M-%Y") AS MONTH_NO,
    COUNT(user_id) AS created_users
FROM user
WHERE    state = 'active'
GROUP BY MONTH_NO;
```

In January 2013, the product witnessed a sign-up of 160 users. Over time, the number of monthly sign-ups steadily increased, reaching its peak in August 2014 with a record 1031 users registering for the product.

MONTH_NO	created_users
January-2013	160
February-2013	160
March-2013	150
April-2013	181
May-2013	214
June-2013	213
July-2013	284
August-2013	316
September-2013	330
October-2013	390
November-2013	399
December-2013	486
January-2014	552
February-2014	525
March-2014	615
April-2014	726
May-2014	779
June-2014	873
July-2014	997
August-2014	1031



INSIGHTS

Case Study 2: Investigating Metric Spike

C. Weekly retention Analysis

The given output indicates that a significant number of users (3743) were retained within the first week of signing up for the product. However, retention gradually declines over time, emphasizing the importance of early user engagement and onboarding. To improve long-term retention, the product team should focus on enhancing the user experience, implementing feature updates, and employing targeted marketing strategies beyond the critical first few weeks.

INSIGHTS

Case Study 2: Investigating Metric Spike

```sql
WITH retention_cal AS (SELECT
                            event.user_id,
                            EXTRACT(WEEK FROM STR_TO_DATE(created_at, '%d-%m-%Y %H:%i')) AS Created_Week_no,
                            MIN(CASE WHEN event_type = 'engagement' THEN EXTRACT(WEEK FROM STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i'))END) AS login_week_no
                        FROM user RIGHT JOIN event ON user.user_id = event.user_id
                        WHERE
                            EXTRACT(YEAR FROM STR_TO_DATE(created_at, '%d-%m-%Y %H:%i')) = 2014 AND
                            user.state = 'active'
                        GROUP BY user.user_id , Created_Week_no),
-- It is an addition to the previous cte it will also return the difference between the first login week and account created week
-- The difference is the time taken to retain that user
weeks_retain_user_count AS (SELECT
                                *, login_week_no - Created_Week_no AS weeks_to_retain
                            FROM retention_cal
                            ORDER BY weeks_to_retain DESC)
-- It will return the weeks_to_retain and no_of_users
-- weeks_to_retain is the number of weeks taken to retain the user
-- no_of_user is the count of users retained within the week
SELECT weeks_to_retain, COUNT(user_id) AS no_of_users
FROM weeks_retain_user_count
GROUP BY weeks_to_retain
ORDER BY weeks_to_retain;
```

weeks_to_retain	no_of_users
0	3743
1	77
2	82
3	58
4	66
5	44
6	39
7	40
8	44
9	48
10	50
11	40
12	55
13	54
14	54
15	45
16	45
17	52
18	48
19	37
20	27
21	11
22	20
23	13
24	8
25	11
26	10
27	7
28	6
29	6
30	4

INSIGHTS

Case Study 2: Investigating Metric Spike

D. Weekly engagement per device

This SQL query helped me to find how much engagement of users was there every week on each device.

```sql
SELECT
    EXTRACT(WEEK FROM STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i')) AS Week_No,
    device,
    COUNT(event_type) AS engagement
FROM event
WHERE event_type = 'engagement'
GROUP BY Week_No , device
ORDER BY Week_No;
```

Week_No	device	engagement
17	acer aspire desktop	67
17	acer aspire notebook	206
17	amazon fire phone	83
17	asus chromebook	251
17	dell inspiron desktop	187
17	dell inspiron notebook	503
17	hp pavilion desktop	132
17	htc one	190
⋮	⋮	⋮
35	macbook air	64
35	macbook pro	122
35	nexus 10	15
35	nexus 5	34
35	nexus 7	17
35	nokia lumia 635	7
35	samsung galaxy note	6
35	samsung galaxy s4	29
35	windows surface	30

491 rows

INSIGHTS

Case Study 2: Investigating Metric Spike

E. Email engagement analysis

```sql
SELECT
    MONTHNAME(STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i')) AS Month_Name,
    action,
    COUNT(user_id) AS No_of_users
FROM
    email_event
GROUP BY Month_Name , action
ORDER BY action DESC;
```

Month_Name	action	No_of_users
May	sent_weekly_digest	11730
June	sent_weekly_digest	13155
July	sent_weekly_digest	15902
August	sent_weekly_digest	16480
May	sent_reengagement_email	758
June	sent_reengagement_email	889
July	sent_reengagement_email	933
August	sent_reengagement_email	1073
June	email_open	4658
July	email_open	5611
August	email_open	5978
May	email_open	4212
May	email_clickthrough	2023
June	email_clickthrough	2274
July	email_clickthrough	2721
August	email_clickthrough	1992

The company is sending more emails each week, but they are not seeing a corresponding increase in click-throughs. This suggests that they need to improve the content of their emails or the way they are targeting their audience.