OPERATIONAL ANALYTICS AND INVESTIGATING METRIC SPIKE

1.1 PROJECT DESCRIPTION

The primary goal of the project is to analyze a company's end-to-end operations. For the proper functioning of the organization, collaborating with teams including operations, support, and marketing to extract valuable insights from the collected data is a vital process. A key aspect involves investigating metric spikes and deciphering abrupt changes in crucial metrics like user engagement or sales. Advanced MySQL skills come in handy while analyzing provided datasets, offering insights that enhance operational efficiency and facilitate a deeper understanding of fluctuating metrics. This project aids in optimizing company operations and fostering informed decision-making across departments.

1.2 APPROACH

The approach involves storing the large data from csv files in the form of systematic tables related to each other in a MySQL database and using MySQL Workbench to convert the stored raw data into meaningful insights that can be used to form profitable strategies for the organization. Using advanced SQL skills leads to gaining a better understanding of the happenings within and related to the organization in a hassle-free and simpler form. As a result, strategies to further improvise and boost the well-being of the organization are formed.

1.3 TECH STACK USED

MySQL is Relational Database Management tool used for storing the raw data generated from the social media platform. MySQL Workbench Version 8.0.36 provides a user-friendly GUI that makes query processing effortless. Designing and modeling databases as well as writing and SQL queries becomes extremely simple for users who prefer a visual approach to database management. MS Excel is used to store a huge amount of data in the form of csv files before loading it into the MySQL database. MS Word is used to document the insights of the analysis efficiently.

1.4 INSIGHTS

1.4.1 Case Study 1: Job Data Analysis

- **Jobs Reviewed Over Time:**
  - Calculate the number of jobs reviewed per hour for each day in November 2020.
  - **Code:**

```sql
#Q1. Jobs Reviewed Over Time
SELECT
    *
FROM
    job_data;
SELECT
    AVG(hour) AS 'jobs reviewed per day per hour',
    AVG(sec) AS 'jobs reviewed per day per second'
FROM
    (SELECT
        ds,
            ((COUNT(job_id) * 3600) / SUM(time_spent)) AS hour,
            (COUNT(job_id) / SUM(time_spent)) AS sec
    FROM
        job_data
    WHERE
        MONTH(ds) = 11
    GROUP BY ds) job;
```

  - **Output:**

	jobs reviewed per day per hour	jobs reviewed per day per second
▶	126.18048333	0.03505000

  - **Inference:**
    - This gives an idea of how many jobs were reviewed per day every hour and every second for November. This data will help the company decide on whether this rate is suitable or can be improved.

- **Throughput Analysis:**
  - Calculate the 7-day rolling average of throughput (number of events per second).
  - **Code:**

```sql
#Q2. Throughput Analysis
SELECT
    ROUND(COUNT(event) / SUM(time_spent), 2) AS 'Weekly Throughput'
FROM
    job_data;

SELECT
    ds AS Date,
    ROUND(COUNT(event) / SUM(time_spent), 2) AS 'Daily Throughput'
FROM
    job_data
GROUP BY ds
ORDER BY ds;
```

- **Output:**

	Weekly Throughput
▶	0.03

	Date	Daily Throughput
▶	2020-11-25	0.02
	2020-11-26	0.02
	2020-11-27	0.01
	2020-11-28	0.06
	2020-11-29	0.05
	2020-11-30	0.05

- **Inference:**
  - Weekly throughput gives an overall idea of the events occurring per second in the entire week whereas daily gives a more detailed analysis. Daily throughput is an excellent choice as one can always take the average or sum of the data as per the requirements if accurate data is available.

⇨ **Language Share Analysis:**

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

```
38      #Q3. Language Share Analysis
39  •   SELECT
40          language,
41          COUNT(*) AS total_lang,
42          COUNT(*) * 100.0 / SUM(COUNT(*)) OVER () AS percentage
43      FROM
44          job_data
45      GROUP BY
46          language
47      ORDER BY
48          language;
```

- **Output:**

	language	total_lang	percentage
▶	Arabic	1	12.50000
	English	1	12.50000
	French	1	12.50000
	Hindi	1	12.50000
	Italian	1	12.50000
	Persian	3	37.50000

- **Inference:**
  - The query provides insights into the distribution of languages in the table, along with the percentage representation of each language out of the total count of languages.

⇨ **Duplicate Rows Detection:**

- Identify duplicate rows in the data.
- **Code:**

```
53      #Q4. Duplicate Rows Detection
54 •    SELECT
55          actor_id, COUNT(*) AS Duplicates
56      FROM
57          job_data
58      GROUP BY actor_id
59      HAVING COUNT(*) > 1;
```

- **Output:**

	actor_id	Duplicates
▶	1003	2

- **Inference:**
  - The records with duplicate ids are identified. Redundant data can be eliminated using this insight.

1.4.2 Case Study 2: Investigating Metric Spike

⇨ **Weekly User Engagement:**

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

```
#Q1. Weekly User Engagement
SELECT
    EXTRACT(WEEK FROM occurred_at) AS week_no,
    COUNT(DISTINCT user_id) AS active_user
FROM
    events
WHERE
    event_type = 'engagement'
GROUP BY week_no
ORDER BY week_no;
```

- **Output:**

week_no	active_user
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

- **Inference:**
  This tells which user was active during which week. The organization could introduce monthly contests to reward these users and bring in more engagement.

⇨ **User Growth Analysis:**

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

```sql
#Q2. User Growth Analysis
SELECT
    year_no,
    week_no,
    total_users,
    SUM(total_users) OVER (ORDER BY year_no, week_no) AS cumulative_users
FROM
    (SELECT
        EXTRACT(YEAR FROM activated_at) AS year_no,
        EXTRACT(WEEK FROM activated_at) AS week_no,
        COUNT(DISTINCT user_id) AS total_users
     FROM
        users
     WHERE
        state = 'active'
     GROUP BY
        year_no, week_no) AS a;
```

- **Output:**

	year_no	week_no	total_users	cumulative_users

- **Inference:**
  - This data indicated the growth of users for a given product in the course of time since its date of creation. It serves as feedback of the product and helps the organization keep a track of how the product is doing in the market.

⇨ **Weekly Retention Analysis:**

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

```sql
#Q3. Weekly Retention Analysis
WITH cte1 AS (
    SELECT DISTINCT
        user_id,
        EXTRACT(WEEK FROM occurred_at) AS signup_week
    FROM
        events
    WHERE
        event_type = 'signup_flow' AND event_name = 'complete_signup'
        AND EXTRACT(WEEK FROM occurred_at) = 18
),
cte2 AS (
    SELECT DISTINCT
        user_id,
        EXTRACT(WEEK FROM occurred_at) AS engagement_week
    FROM
        events
    WHERE
        event_type = 'engagement'
)
```

```sql
SELECT
    COUNT(user_id) AS total_engaged_users,
    SUM(CASE WHEN retention_week > 8 THEN 1 ELSE 0 END) AS retained_users
FROM
    (
    SELECT
        a.user_id,
        a.signup_week,
        b.engagement_week,
        b.engagement_week - a.signup_week AS retention_week
    FROM
        cte1 a
    LEFT JOIN
        cte2 b ON a.user_id = b.user_id
    ORDER BY
        a.user_id
    ) AS sub;
```

- **Output:**

	total_engaged_users	retained_users
▶	615	96

- **Inference:**
  - This allows the company to understand if the product is being used by users repeatedly or not. This helps the marketing team strategize their next move to bring the customers back.

⇨ **Weekly Engagement Per Device:**

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

```sql
#Q4. Weekly Engagement Per Device
WITH cte AS (
    SELECT
        CONCAT(EXTRACT(YEAR FROM occurred_at), '-', EXTRACT(WEEK FROM occurred_at)) AS week_no,
        device,
        COUNT(DISTINCT user_id) AS total_users
    FROM
        events
    WHERE
        event_type = 'engagement'
    GROUP BY
        week_no, device
    ORDER BY
        week_no
)
SELECT
    week_no,
    device,
    total_users
FROM
    cte;
```

- **Output:**

week_no	device	total_users
2014-17	acer aspire desktop	9
2014-17	acer aspire notebook	20
2014-17	amazon fire phone	4
2014-17	asus chromebook	21
2014-17	dell inspiron desktop	18
2014-17	dell inspiron notebook	46
2014-17	hp pavilion desktop	14
2014-17	htc one	16
2014-17	ipad air	27
2014-17	ipad mini	19
2014-17	iphone 4s	21
2014-17	iphone 5	65
2014-17	iphone 5s	42
2014-17	kindle fire	6

- **Inference:**
  - This data provides an insight on the category of users of the product. This allows the company to understand which device is being used by most users and to better understand and meet the requirements of those users.

⇨ **Email Engagement Analysis:**

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

```
#Q5. Email Engagement Analysis
SELECT
    100 * SUM(CASE
        WHEN email_cat = 'email_open' THEN 1
        ELSE 0
    END) / SUM(CASE
        WHEN email_cat = 'email_sent' THEN 1
        ELSE 0
    END) AS email_open_rate,
    100 * SUM(CASE
        WHEN email_cat = 'email_clicked' THEN 1
        ELSE 0
    END) / SUM(CASE
        WHEN email_cat = 'email_sent' THEN 1
        ELSE 0
    END) AS email_click_rate
FROM
    (SELECT
        *,
            CASE
                WHEN action IN ('sent_weekly_digest' , 'sent_reengagement_email') THEN 'email_sent'
                WHEN action IN ('email_open') THEN 'email_open'
                WHEN action IN ('email_clickthrough') THEN 'email_clicked'
            END AS email_cat
    FROM
        email_events) AS sub;
```

- **Output:**

email_open_rate	email_click_rate
33.5834	14.7899

- **Inference:**
  - Emails are an efficient way to track engagement and this tracking can be done by monitoring a number of parameters like clicks, emails sent, emails opened, etc. This helps the teams form a decision on how to pop an ad and increase engagement based on the most prominent email engagement form.

1.5 RESULTS

The user analysis generated helps in assessing various aspects across multiple dimensions. For the first case study, the throughput, the number of jobs reviewed in a given time proved to be helpful in making the operations more efficient and the overall functioning of the company better from a management point of view. The second case study looked into the engagement of the users with the products and services of the company. Investigation of such factors and improving based on them from time to time is what helps build a fortune from a seed.