



OPERATIONAL ANALYTICS: INVESTIGATING AND IMPROVING BUSINESS METRICS



Submitted by:
DHANUSH HEGDE

Table of Contents

TABLE OF CONTENTS

01. **INTRODUCTION**

02. **PROJECT DESCRIPTION**

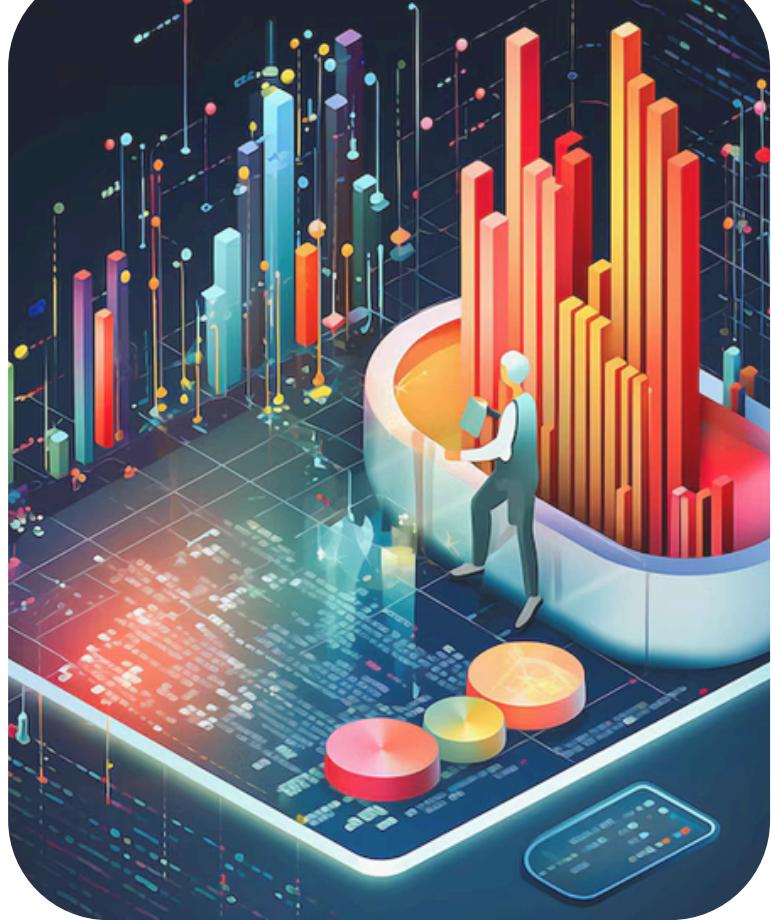
03. **APPROACH**

04. **TASKS & INSIGHTS**

05. **TECH-STACK USED**

06. **RESULT**

INTRODUCTION



In this project, Operational Analytics was utilized to analyze a company's end-to-end operations, focusing on identifying areas for improvement and investigating sudden changes in key metrics. As a Lead Data Analyst at a company like Microsoft, I worked with various datasets and tables, collaborating with teams such as operations, support, and marketing. Advanced SQL skills were applied to derive actionable insights, addressing critical questions like explaining dips in user engagement or sales. The analysis provided valuable recommendations, enhancing the company's understanding of operational inefficiencies and guiding strategies for improved performance.

PROJECT DESCRIPTION

This project is focused on conducting a comprehensive analysis of user interactions and engagement within the Instagram app, leveraging SQL and MySQL Workbench as the primary tools for data extraction, transformation, and visualization. As a Data Analyst, my objective is to generate actionable insights that drive strategic decision-making across multiple key departments, including marketing, product development, and user experience design.

By systematically tracking and analyzing user behavior patterns, this study aims to uncover critical trends, pain points, and opportunities that can enhance platform functionality, optimize feature performance, and improve overall user engagement. Through deep-diving into large datasets, I will identify factors influencing user retention, content interaction, and engagement metrics, such as likes, shares, comments, time spent on posts, and story interactions.



"Driving growth through data-driven insights to shape the future of innovation."



APPROACH

I started by downloading the Case Study 1 dataset and creating the project named 'Project3' in MySQL. I then created the table job_data and inserted the column names and data provided in CSV format. Using SQL, I performed exploratory data analysis to understand the data structure and identify key metrics.

01

Data Collection

Dataset: Extracted data from the provided dataset to get the resulting insights.

02

Exploratory Data Analysis (EDA)

Analyzed key metrics: likes, comments, users, follower, photos, tags & identified correlations and patterns across various variables.

03

Insights Derivation

Using MySQL Workbench and SQL, we cleaned and transformed raw data to uncover meaningful insights. This analysis provided actionable recommendations for optimizing business strategies and improving decision-making.

TASKS PERFORMED & INSIGHTS

A) JOBS REVIEWED OVER TIME:

TASK 1:

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

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

SQL QUERY:

```
SELECT
    DATE(ds) AS review_date,
    COUNT(job_id) AS jobs_per_day,
    SUM(time_spent) / 3600 AS hours_spent
FROM
    job_data
WHERE
    ds >= '2020-11-01'
    AND ds <= '2020-11-30'
GROUP BY
    review_date;
```

OUTPUT:

	review_date	jobs_per_day	hours_spent
▶	2020-11-30	2	0.0111
	2020-11-29	1	0.0056
	2020-11-28	2	0.0092
	2020-11-27	1	0.0289
	2020-11-26	1	0.0156
	2020-11-25	1	0.0125

INSIGHTS:

The results indicate the daily job review activity in November 2020, showing that on average, very few jobs were reviewed per day, with hours spent per job being minimal, suggesting low activity levels during that period.

1.Throughput Analysis:

- Objective: Calculate the 7-day rolling average of throughput (number of events per second).
 - Your Task: Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.
-

T A S K 2 :

B) THROUGHPUT ANALYSIS :

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

Task: Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

SQL QUERY:

SELECT

```
    DATE(ds) AS event_date,  
    COUNT(event) AS total_events,  
    SUM(time_spent) AS total_time_spent,  
    COUNT(event) / SUM(time_spent) AS events_per_second,  
    ROUND(  
        AVG(COUNT(event) / SUM(time_spent)) OVER (  
            ORDER BY DATE(ds)  
            ROWS BETWEEN 6 PRECEDING AND CURRENT ROW  
        ), 2  
    ) AS rolling_avg_7_day  
FROM  
    job_data  
GROUP BY  
    event_date  
ORDER BY  
    event_date;
```

OUTPUT:

event_date	total_events	total_time_spent	events_per_second	rolling_avg_7_day
2020-11-25	1	45	0.0222	0.02
2020-11-26	1	56	0.0179	0.02
2020-11-27	1	104	0.0096	0.02
2020-11-28	2	33	0.0606	0.03
2020-11-29	1	20	0.0500	0.03
2020-11-30	2	40	0.0500	0.04

INSIGHTS:

The 7-day rolling average shows a steady and gradual increase in throughput, smoothing out daily fluctuations, which helps provide a clearer view of long-term trends in event processing efficiency.

1. Language Share Analysis:

- Objective: Calculate the percentage share of each language in the last 30 days.
- Your Task: Write an SQL query to calculate the percentage share of each language over the last 30 days.

T A S K 3 :

C) LANGUAGE SHARE ANALYSIS:

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

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

SQL QUERY:

```

WITH JD AS (
    SELECT
        LANGUAGE,
        COUNT(JOB_ID) AS NUM_JOBS
    FROM
        JOB_DATA
    WHERE
        EVENT IN ('TRANSFER', 'DECISION')
        AND DS BETWEEN '2020-11-01' AND '2020-11-30'
    GROUP BY
        LANGUAGE
),
TOTAL AS (
    SELECT
        COUNT(JOB_ID) AS TOTAL_JOBS
    FROM
        JOB_DATA
    WHERE
        EVENT IN ('TRANSFER', 'DECISION')
        AND DS BETWEEN '2020-11-01' AND '2020-11-30'
)
SELECT
    JD.LANGUAGE,
    ROUND(100.0 * JD.NUM_JOBS / TOTAL.TOTAL_JOBS, 2) AS PERC_JOB
FROM
    JD
CROSS JOIN
    TOTAL
ORDER BY
    PERC_JOB DESC;

```

OUTPUT:

	language	perc_job
▶	Persian	33.33
	Arabic	16.67
	Hindi	16.67
	French	16.67
	Italian	16.67

INSIGHTS:

The language share analysis indicates that Persian accounts for the highest percentage of jobs at 33.33%, while Arabic, Hindi, French, and Italian each share an equal 16.67% of the total jobs in the last 30 days, reflecting a diverse but slightly skewed distribution toward Persian.

1. Duplicate Rows Detection:

- Objective: Identify duplicate rows in the data.
- Your Task: Write an SQL query to display duplicate rows from the job_data table.

TASK 4:

D) DUPLICATE ROWS DETECTION:

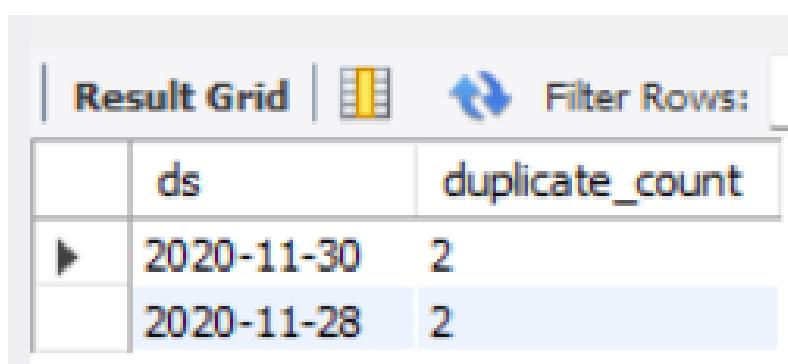
Objective: Identify duplicate rows in the data

Task: Write an SQL query to display duplicate rows from the job_data table.

SQL QUERY:

```
SELECT
    DS,
    COUNT(*) AS DUPLICATE_COUNT
FROM
    JOB_DATA
GROUP BY
    DS
HAVING
    COUNT(*) > 1;
```

OUTPUT:



	ds	duplicate_count
▶	2020-11-30	2
	2020-11-28	2

INSIGHTS:

The duplicate rows detection shows that there are two duplicate entries in the job_data table for the dates 30-11-2020 and 28-11-2020, indicating potential data redundancy on those specific days.

CASE STUDY 2: INVESTIGATING METRIC SPIKE

TASKS:

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.

SQL QUERY:

```
SELECT
DATE(DATE_SUB(occurred_at, INTERVAL WEEKDAY(occurred_at) DAY))
AS week_start_date,
WEEK(occurred_at) AS week_number,
COUNT(DISTINCT user_id) AS weekly_active_users
FROM
events
WHERE
event_type = 'engagement'
AND event_name = 'login'
GROUP BY
week_start_date, week_number
ORDER BY
week_start_date;
```

OUTPUT:

	week_start_date	week_number	weekly_active_users
▶	2014-04-28	17	663
	2014-04-28	18	79
	2014-05-05	18	1044
	2014-05-05	19	92
	2014-05-12	19	1081
	2014-05-12	20	83
	2014-05-19	20	1136
	2014-05-19	21	76
	2014-05-26	21	1100
	2014-05-26	22	86
	2014-06-02	22	1157
	2014-06-02	23	100
	2014-06-09	23	1201
	2014-06-09	24	98
	2014-06-16	24	1253

INSIGHTS:

User engagement shows a consistent upward trend from April to August, with notable spikes in activity during certain weeks, likely driven by external factors such as promotions or feature releases. However, fluctuations in later weeks suggest periodic drops in user activity, warranting deeper analysis of potential causes like seasonal or operational factors.

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.

SQL QUERY:

```
SELECT  
DATE_FORMAT(created_at, '%Y-%m') AS month,  
COUNT(*) AS total_users  
FROM  
users  
GROUP BY  
month  
ORDER BY  
month;
```

OUTPUT:

Result Grid		
	month	total_users
▶	2013-01	160
	2013-02	160
	2013-03	150
	2013-04	181
	2013-05	214
	2013-06	213
	2013-07	284
	2013-08	316
	2013-09	330
	2013-10	390
	2013-11	399
	2013-12	486
	2014-01	552
	2014-02	525
	2014-03	615
	2014-04	726
	2014-05	779
	2014-06	873
	2014-07	997

INSIGHTS:

The product experienced steady user growth from 160 in January 2013 to 1,031 by August 2014, with significant increases in mid-2013 and a major surge from December 2013 onwards. Notable growth periods include July 2013 to January 2014, indicating possible product improvements or successful marketing strategies during that time.

3) Weekly Retention Analysis:

- Objective: Analyze the retention of users on a weekly basis after signing up for a product.
- Your Task: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

SQL QUERY:

```
SELECT
DATE(DATE_SUB(occurred_at, INTERVAL WEEKDAY(occurred_at) DAY))
AS week,
COUNT(CASE WHEN e.event_type = 'engagement' THEN e.user_id END)
AS engagement,
COUNT(CASE WHEN e.event_type = 'signup_flow' THEN e.user_id END)
AS signup
FROM
events e
GROUP BY
week
ORDER BY
week;
```

OUTPUT:

	week	engagement	signup
▶	2014-04-28	8709	81
	2014-05-05	17532	160
	2014-05-12	17047	186
	2014-05-19	17890	177
	2014-05-26	17193	186
	2014-06-02	18608	197
	2014-06-09	18233	198
	2014-06-16	18976	222
	2014-06-23	18859	210
	2014-06-30	18959	199
	2014-07-07	19965	223
	2014-07-14	20723	215
	2014-07-21	20132	228
	2014-07-28	21472	234
	2014-08-04	18341	189
	2014-08-11	16612	250
	2014-08-18	16158	259
	2014-08-25	16166	266

INSIGHTS:

User engagement and signups show a steady increase from April to August 2014, indicating consistent user retention over time. Engagement tends to track closely with signups, suggesting that a high percentage of new users remain active in the following weeks. However, the drop in engagement in early August (from 21,472 to 16,158) despite increasing signups suggests a potential dip in user retention that may need further investigation.

4) Weekly Engagement Per Device:

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

SQL QUERY:

```
SELECT
DATE(DATE_SUB(occurred_at, INTERVAL WEEKDAY(occurred_at) DAY))
AS week_start_date,
SUM(CASE
WHEN e.device IN ('Amazon Fire Phone', 'HTC One', 'iPhone 4s', 'iPhone 5',
'iPhone 5s', 'Nokia Lumia 635', 'Samsung Galaxy Note', 'Samsung Galaxy
S4') THEN 1
ELSE 0
END) AS mobile, -- Count of distinct mobile users
SUM(CASE
WHEN e.device IN ('Amazon Kindle Fire', 'iPad Air', 'iPad Mini', 'Nexus 10',
'Nexus 7', 'Samsung Galaxy Tablet', 'Windows Surface') THEN 1
ELSE 0
END) AS tablet, -- Count of distinct tablet users
SUM(CASE
WHEN e.device IN ('Acer Aspire Desktop', 'Acer Aspire Notebook', 'Asus
Chromebook', 'Dell Inspiron Desktop', 'Dell Inspiron Notebook', 'HP Pavilion
Desktop', 'Lenovo ThinkPad', 'Mac Mini', 'MacBook Air', 'MacBook Pro')
THEN 1
ELSE 0
END) AS pc -- Count of distinct PC users
FROM
events e
WHERE
e.event_type = 'engagement'
and e.event_name = 'login'
GROUP BY
week_start_date
ORDER BY
week_start_date;
```

OUTPUT:

Result Grid | Filter Rows:

week_start_date	mobile	tablet	pc
2014-04-28	268	109	525
2014-05-05	517	182	1146
2014-05-12	536	203	1128
2014-05-19	571	184	1166
2014-05-26	547	174	1106
2014-06-02	559	218	1227
2014-06-09	611	200	1213
2014-06-16	609	237	1261
2014-06-23	590	229	1285
2014-06-30	634	234	1232
2014-07-07	663	234	1345
2014-07-14	667	234	1436
2014-07-21	698	221	1363
2014-07-28	684	262	1466
2014-08-04	559	190	1395
2014-08-11	480	169	1358
2014-08-18	448	159	1343
2014-08-25	465	168	1306

INSIGHTS:

User engagement across devices shows that PC usage dominates weekly activity, with the highest engagement seen in July 2014. Mobile usage steadily increases until late July but drops in August, while tablet usage remains relatively stable. The decline in mobile and tablet engagement in August suggests a shift in user preference or a possible issue with mobile engagement during that period.

5) Email Engagement Analysis:

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

SQL QUERY:

- **WEEKLY BASIS:**

```
SELECT
    DATE(DATE_SUB(occurred_at, INTERVAL WEEKDAY(occurred_at)
DAY)) AS week_start_date,
    COUNT(DISTINCT user_id) AS weekly_active_user,
    COUNT(CASE WHEN action = 'email_open' THEN user_id END) AS
weekly_email_open,
    COUNT(CASE WHEN action = 'email_clickthrough' THEN user_id END)
AS weekly_email_clickthrough,
    COUNT(CASE WHEN action = 'sent_reengagement_email' THEN
user_id END) AS weekly_sent_reengagement_email,
    COUNT(CASE WHEN action = 'sent_weekly_digest' THEN user_id END)
AS weekly_sent_weekly_digest
FROM
emailevents
GROUP BY
week_start_date
ORDER BY
week_start_date;
```

OUTPUT:

week_start_date	weekly_active_user	weekly_email_open	weekly_email_clickthrough	weekly_sent_reengagement_email	weekly_sent_weekly_digest
2014-04-28	1006	332	187	98	908
2014-05-05	2724	919	434	164	2602
2014-05-12	2801	971	479	175	2665
2014-05-19	2876	995	498	179	2733
2014-05-26	2945	1026	453	179	2822
2014-06-02	3047	993	492	199	2911
2014-06-09	3143	1070	533	190	3003
2014-06-16	3272	1161	563	234	3105
2014-06-23	3340	1090	524	187	3207
2014-06-30	3461	1168	559	222	3302
2014-07-07	3557	1230	622	214	3399
2014-07-14	3675	1260	607	226	3499
2014-07-21	3748	1211	584	206	3592
2014-07-28	3883	1386	633	230	3706
2014-08-04	3953	1336	432	206	3793
2014-08-11	4061	1357	430	224	3897
2014-08-18	4209	1421	487	257	4012
2014-08-25	4309	1533	493	263	4111

INSIGHTS:

Email engagement shows a consistent increase in weekly active users, email opens, and clickthroughs from April to August 2014, indicating strong user interaction with the service. The steady rise in re-engagement and weekly digest emails suggests successful retention efforts. However, the slight drop in clickthroughs in August, despite increasing email opens, may indicate a need to optimize email content for better engagement.

- **MONTHLY BASIS:**

```
SELECT
    DATE_FORMAT(occurred_at, '%Y-%m') AS month_year,
    COUNT(DISTINCT user_id) AS monthly_active_user,
    COUNT(CASE WHEN action = 'email_open' THEN user_id END) AS
monthly_email_open,
    COUNT(CASE WHEN action = 'email_clickthrough' THEN user_id END)
AS monthly_email_clickthrough,
    COUNT(CASE WHEN action = 'sent_reengagement_email' THEN
user_id END) AS monthly_sent_reengagement_email,
    COUNT(CASE WHEN action = 'sent_weekly_digest' THEN user_id END)
AS monthly_sent_weekly_digest
FROM
emailevents
GROUP BY
month_year
ORDER BY
month_year;
```

OUTPUT:

Result Grid					
Filter Rows: <input type="button" value="Filter"/>					
Export: <input type="button" value="CSV"/> Wrap Cell Content: <input type="checkbox"/>					
month_year	monthly_active_user	monthly_email_open	monthly_email_clickthrough	monthly_sent_reengagement_email	monthly_sent_weekly_diges
2014-05	3289	4212	2023	758	11730
2014-06	3736	4658	2274	889	13155
2014-07	4195	5611	2721	933	15902
2014-08	4766	5978	1992	1073	16480

INSIGHTS:

Monthly email engagement shows strong growth in active users, email opens, and clickthroughs from May to August 2014. However, despite an increase in monthly active users and emails sent, clickthrough rates drop significantly in August (from 2,721 to 1,992), indicating that while more users are interacting with emails, fewer are engaging with the content. This suggests a need to improve email relevance or calls-to-action to boost engagement.

TECHSTACK

Software and Versions Used

- **MySQL Workbench (Version 8)**

- MySQL Workbench was chosen for its robust capabilities in database management and visualization. Its user-friendly interface made it easier to design schemas, execute SQL queries, and visualize data relationships effectively.
- The version used was compatible with the dataset size and provided necessary features like query optimization and ER diagrams, streamlining the data analysis process.

- **SQL**

- SQL was used for its efficiency in handling large datasets and performing complex queries, such as filtering, aggregation, and joining tables.
- It allowed me to analyze data and perform the required tasks to derive actionable insights effectively.

- **Canva**

- Canva was chosen for its ease of use and professional templates, enabling us to create a visually appealing report summarizing the project findings.
- Its intuitive interface helped in presenting technical insights in an organized and engaging manner, suitable for stakeholders.

Why These Tools Were Chosen

The combination of MySQL Workbench and SQL provided a powerful and efficient platform for managing and analyzing relational data. Canva complemented this by facilitating the creation of a polished and visually engaging report, ensuring effective communication of the project results.

RESULT

This project focused on analyzing operational metrics to identify areas for improvement using SQL-based data analysis. Key insights derived from the analysis include:

- Job Reviews Over Time: The daily job review activity in November 2020 was relatively low, with minimal hours spent per job, indicating a potential slowdown in operations.
- Throughput Analysis: The 7-day rolling average provided a clearer trend of increasing event processing efficiency, smoothing out daily fluctuations.
- Language Share Analysis: Persian accounted for the highest percentage of jobs (33.33%), while Arabic, Hindi, French, and Italian had equal shares (16.67%), suggesting a diverse but slightly skewed distribution.
- Duplicate Rows Detection: Data redundancy was observed on specific dates (30-11-2020 and 28-11-2020), highlighting potential data quality issues.
- Weekly User Engagement: User engagement showed an upward trend from April to August, with periodic fluctuations likely due to external factors like promotions or seasonal trends.
- User Growth Analysis: The number of users steadily increased over time, indicating positive adoption of the product, with some months experiencing significant growth spurts

These insights helped in identifying inefficiencies, optimizing operations, and improving overall business performance.