## What is Operation Analytics?

Operational analytics focuses on real-time measurement and assessment of a company's existing day-to-day operations. It allows the company to monitor its operations in real time, enabling timely actions to enhance customer satisfaction and improve the bottom line. Operation Analytics is a comprehensive analysis of an organization's end-to-end operations, providing insights that are crucial for identifying areas that require improvement. It plays a pivotal role in predicting a company's overall growth or decline.

Here are some practical examples:

- Ops: Developers can utilize real-time data to observe how customers are using their products and make on-the-fly adjustments.
- Marketing: Businesses can optimize user engagement in real time by leveraging operational analytics to offer personalized recommendations.

**Project Approach**

This project was developed using SQL Workbench and follows these steps:

1. Database Creation: A database is created using the dataset provided by the company.
2. Data Loading: The data is imported into SQL Workbench.
3. Data Analysis: Extensive analysis is conducted to answer critical questions, such as identifying reasons for fluctuations in daily engagement and sales dips. These questions need to be addressed daily, making it crucial to investigate metric spikes.

**Tech-Stack Used**

MySQL Workbench serves as the primary tool for this project. It provides a unified environment for data modeling, SQL development, and database administration. Key features include:

- Visual SQL Editor: Developers can visually design, edit, and run queries, allowing them to preview changes before applying them.
- Database Administration: MySQL Workbench offers a comprehensive suite for database administration, enabling tasks such as auditing, server configuration, and log viewing.
- Performance Monitoring: Users can monitor query status, client timing, network latency, and index usage through a user-friendly dashboard, facilitating the identification of opportunities for SQL performance optimization.

## Insights from Case Study 1 (Job Data)…

1. **Calculate the Number of Jobs Reviewed per Hour per Day for November 2020.**

To determine the number of jobs reviewed per hour per day in November 2020, the following SQL query was executed:

```
SELECT ds AS Dates, ROUND((COUNT(job_id) / SUM(time_spent)) * 3600)
AS "Jobs Reviewed per Hour per Day FROM job_data
WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY ds;
```

***The analysis revealed that on November 28, 2020, the highest number of jobs reviewed in a single day was 218.***

This information provides valuable insights into the daily job review activity for November 2020, with a notable peak on November 28th, where 218 jobs were reviewed, potentially indicating a significant event or activity on that day.

| Dates | Jobs Reviewed per Hour per Day |
|---|---|
| 2020-11-30 | 180 |
| 2020-11-29 | 180 |
| 2020-11-28 | 218 |
| 2020-11-27 | 35 |
| 2020-11-26 | 64 |
| 2020-11-25 | 80 |

## 2. Calculate 7-Day Rolling Average of Throughput

**To calculate the 7-day rolling average of throughput, two SQL queries were executed:**

### Weekly Throughput:

```
SELECT ROUND(COUNT(event) / SUM(time_spent), 2) AS "Weekly
Throughput" FROM job_data;
```

**The result is a weekly throughput of 0.03.**

### Daily Throughput:

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

**On November 28, 2020, the daily throughput was found to be the highest at 0.06.**

Preference for Daily Metric or 7-Day Rolling Metric

Metrics exhibit fluctuations on both daily and weekly bases. Daily metrics provide a granular view, with updates available every day or even every minute. However, 7-day rolling metrics offer a broader perspective by smoothing out daily variations, making them particularly useful for identifying trends over time.

The preference between daily and 7-day rolling metrics depends on the specific analytical goals:

**Daily Metrics:** These are ideal for monitoring short-term, day-to-day changes and identifying immediate issues or opportunities. They provide real-time insights.

**7-Day Rolling Metrics:** These are valuable for identifying longer-term trends and patterns, which are often less affected by daily fluctuations. They offer a more stable view of performance over time.

*In summary, while daily metrics are crucial for quick reactions to daily changes, 7-day rolling metrics are superior in revealing overall trends and performance stability, making them a preferred choice for strategic decision-making and trend analysis.*

**3. Calculate the Percentage Share of Each Language in the Last 30 Days**

To calculate the percentage share of each language in the last 30 days, the following SQL query was executed:

```
SELECT language AS Languages, ROUND(100 * COUNT(*) / total, 2)
AS Percentage FROM job_data CROSS JOIN (SELECT COUNT(*) AS
total FROM job_data) sub GROUP BY language;
```

***The analysis indicates that the Persian language has the highest percentage share, accounting for 37.5% of the total.***

**4. Display Duplicate Rows from the Table ,To identify and display duplicate rows in the data, the following SQL query was used:**

```
SELECT actor_id, COUNT(*) AS Duplicates FROM job_data
GROUP BY actor_id HAVING COUNT(*) > 1;
```

*This query identifies rows with duplicate values in the actor id column. It reveals that Actor ID 1003 has duplicate rows in the dataset.*

*These insights provide an understanding of the distribution of languages in the dataset, with Persian being the most prevalent. Additionally, the detection of duplicate rows helps ensure data quality and integrity.*

# Case Study 2 - Investigating Metric Spike

## 1. Calculate the Weekly User Engagement

**To calculate the weekly user engagement, the following SQL query was executed:**

```
SELECT EXTRACT(WEEK FROM occurred_at) AS "Week Numbers",
COUNT(DISTINCT user_id) AS "Weekly Active Users" FROM events
WHERE event_type = 'engagement' GROUP BY 1;
```

This query provides insights into the number of weekly active users engaged in events categorized under 'engagement.' It extracts the week numbers from the occurred_at column and counts the distinct user IDs for each week.

The result will give you a breakdown of weekly user engagement, which is valuable for understanding user activity trends over time.

| Week Numbers | Weekly Active Users |
|---|---|
| 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 |

## 2.Calculate the User Growth for Product

***To calculate the user growth for the product, the following SQL query was executed:***

```
SELECT Months, Users, ROUND(((Users / LAG(Users, 1) OVER
(ORDER BY Months) - 1) * 100), 2) AS "Growth in %" FROM (
SELECT EXTRACT(MONTH FROM created_at) AS Months,
```

COUNT(activated_at) AS Users FROM users WHERE activated_at IS NOT NULL GROUP BY 1 ORDER BY 1 ) sub;

| Week Numbers | 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 | 740 | 472 | 324 | 251 | 205 | 187 | 167 | 146 | 145 | 145 | 136 | 131 | 132 | 143 | 116 | 91 | 82 | 77 | 5 |
| 18 | 788 | 362 | 261 | 203 | 168 | 147 | 144 | 127 | 113 | 122 | 106 | 118 | 127 | 110 | 97 | 85 | 67 | 4 | 0 |
| 19 | 601 | 284 | 173 | 153 | 114 | 95 | 91 | 81 | 95 | 82 | 68 | 65 | 63 | 42 | 51 | 49 | 2 | 0 | 0 |
| 20 | 555 | 223 | 165 | 121 | 91 | 72 | 63 | 67 | 63 | 65 | 67 | 41 | 40 | 33 | 40 | 0 | 0 | 0 | 0 |
| 21 | 495 | 187 | 131 | 91 | 74 | 63 | 75 | 72 | 58 | 48 | 45 | 39 | 35 | 28 | 2 | 0 | 0 | 0 | 0 |
| 22 | 521 | 224 | 150 | 107 | 87 | 73 | 63 | 60 | 55 | 48 | 41 | 39 | 31 | 1 | 0 | 0 | 0 | 0 | 0 |
| 23 | 542 | 219 | 138 | 101 | 90 | 79 | 69 | 61 | 54 | 47 | 35 | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 535 | 205 | 143 | 102 | 81 | 63 | 65 | 61 | 38 | 39 | 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 25 | 500 | 218 | 139 | 101 | 75 | 63 | 50 | 46 | 38 | 35 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 26 | 495 | 181 | 114 | 83 | 73 | 55 | 47 | 43 | 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 27 | 493 | 199 | 121 | 106 | 68 | 53 | 40 | 36 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 28 | 486 | 194 | 114 | 69 | 46 | 30 | 28 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 29 | 501 | 186 | 102 | 65 | 47 | 40 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 30 | 533 | 202 | 121 | 78 | 53 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 31 | 430 | 145 | 76 | 57 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 32 | 496 | 188 | 94 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 33 | 499 | 202 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 34 | 518 | 44 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 35 | 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This query calculates the growth percentage of users over time by comparing the current month's user count with the previous month. It extracts the month from the created_at column, counts the activated users, and then calculates the growth in percentage.
The result will provide you with insights into how the user base for the product has been growing over time, allowing you to assess the product's performance and adoption rate.

1. Calculate the Weekly Retention of Users-Sign-Up Cohort
2. To calculate the weekly retention of users in the sign-up cohort, the following SQL query was executed.

*This query calculates the weekly retention rates for users in the sign-up cohort. It determines how many users from the initial cohort remain engaged in each subsequent week. The result provides insights into user retention over time, which is essential for understanding the product's ability to retain users after their initial sign-up.*

**4.Calculate the weekly engagement per device?**

SELECT first AS "Week Numbers",
SUM(CASE WHEN week_number = 0 THEN 1 ELSE 0 END) AS "Week 0",
SUM(CASE WHEN week_number = 1 THEN 1 ELSE 0 END) AS "Week 1",
SUM(CASE WHEN week_number = 2 THEN 1 ELSE 0 END) AS "Week 2",
SUM(CASE WHEN week_number = 3 THEN 1 ELSE 0 END) AS "Week 3",

```
SUM(CASE WHEN week_number = 4 THEN 1 ELSE 0 END) AS "Week 4",
SUM(CASE WHEN week_number = 5 THEN 1 ELSE 0 END) AS "Week 5",
SUM(CASE WHEN week_number = 6 THEN 1 ELSE 0 END) AS "Week 6",
SUM(CASE WHEN week_number = 7 THEN 1 ELSE 0 END) AS "Week 7",
SUM(CASE WHEN week_number = 8 THEN 1 ELSE 0 END) AS "Week 8",
SUM(CASE WHEN week_number = 9 THEN 1 ELSE 0 END) AS "Week 9",
SUM(CASE WHEN week_number = 10 THEN 1 ELSE 0 END) AS "Week 10",
SUM(CASE WHEN week_number = 11 THEN 1 ELSE 0 END) AS "Week 11",
SUM(CASE WHEN week_number = 12 THEN 1 ELSE 0 END) AS "Week 12",
SUM(CASE WHEN week_number = 13 THEN 1 ELSE 0 END) AS "Week 13",
SUM(CASE WHEN week_number = 14 THEN 1 ELSE 0 END) AS "Week 14",
SUM(CASE WHEN week_number = 15 THEN 1 ELSE 0 END) AS "Week 15",
SUM(CASE WHEN week_number = 16 THEN 1 ELSE 0 END) AS "Week 16",
SUM(CASE WHEN week_number = 17 THEN 1 ELSE 0 END) AS "Week 17",
SUM(CASE WHEN week_number = 18 THEN 1 ELSE 0 END) AS "Week 18"
FROM
(
   SELECT m.user_id, m.login_week, n.first, m.login_week - first AS
week_number
   FROM
   (
      SELECT user_id, EXTRACT(WEEK FROM occurred_at) AS login_week
      FROM events
      GROUP BY 1, 2
   ) m,
   (
      SELECT user_id, MIN(EXTRACT(WEEK FROM occurred_at)) AS first
      FROM events
      GROUP BY 1
   ) n
   WHERE m.user_id = n.user_id
) sub
GROUP BY first
ORDER BY first;
```

| Week Numbers | Dell Inspiron Notebook | iPhone 5 | iPhone 4S | Windows Surface | Macbook Air | iPhone 5S | Macbook Pro | Kindle Fire | iPad Mini | Nexus 7 | Nexus 5 | Samsung Galaxy S4 | Lenovo Thinkpad | Samsumg Galaxy Tablet |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 17 | 46 | 65 | 21 | 10 | 54 | 42 | 143 | 6 | 19 | 18 | 40 | 52 | 86 | 8 |
| 18 | 77 | 113 | 46 | 10 | 121 | 73 | 252 | 27 | 30 | 30 | 73 | 82 | 153 | 11 |
| 19 | 83 | 115 | 44 | 16 | 112 | 79 | 266 | 21 | 36 | 41 | 87 | 91 | 178 | 6 |
| 20 | 84 | 125 | 55 | 21 | 119 | 79 | 256 | 23 | 32 | 32 | 103 | 93 | 173 | 9 |
| 21 | 80 | 137 | 45 | 17 | 110 | 74 | 247 | 30 | 23 | 29 | 91 | 84 | 167 | 6 |
| 22 | 92 | 125 | 45 | 15 | 145 | 71 | 251 | 21 | 34 | 45 | 96 | 105 | 176 | 10 |
| 23 | 103 | 152 | 53 | 14 | 124 | 79 | 266 | 25 | 33 | 36 | 88 | 99 | 176 | 14 |
| 24 | 99 | 142 | 53 | 22 | 152 | 79 | 255 | 25 | 39 | 49 | 87 | 101 | 165 | 11 |
| 25 | 105 | 137 | 40 | 22 | 121 | 78 | 275 | 24 | 30 | 51 | 89 | 99 | 197 | 12 |
| 26 | 89 | 152 | 50 | 21 | 134 | 94 | 269 | 26 | 43 | 46 | 87 | 112 | 192 | 12 |
| 27 | 89 | 163 | 67 | 33 | 142 | 83 | 302 | 25 | 35 | 40 | 84 | 116 | 202 | 15 |
| 28 | 103 | 151 | 61 | 33 | 148 | 93 | 295 | 31 | 35 | 39 | 85 | 122 | 220 | 9 |
| 29 | 113 | 144 | 60 | 28 | 148 | 90 | 295 | 37 | 34 | 45 | 77 | 123 | 209 | 13 |
| 30 | 127 | 152 | 65 | 19 | 159 | 103 | 322 | 25 | 35 | 62 | 84 | 103 | 206 | 9 |
| 31 | 113 | 135 | 56 | 19 | 147 | 71 | 321 | 14 | 27 | 38 | 69 | 100 | 207 | 8 |
| 32 | 104 | 119 | 34 | 10 | 125 | 67 | 307 | 12 | 30 | 25 | 67 | 82 | 179 | 6 |
| 33 | 110 | 110 | 35 | 15 | 133 | 65 | 312 | 14 | 28 | 30 | 70 | 80 | 191 | 12 |
| 34 | 105 | 101 | 50 | 18 | 136 | 70 | 292 | 13 | 25 | 33 | 70 | 90 | 193 | 14 |
| 35 | 9 | 2 | 6 | 3 | 10 | 3 | 17 | 3 | 2 | 2 | 4 | 6 | 16 | 0 |

| Week Numbers | Acer Aspire Notebook | Asus Chromebook | HTC One | Nokia Lumia 635 | Samsung Galaxy Note | Acer Aspire Desktop | Mac Mini | HP Pavilion Desktop | Dell Inspiron Desktop | iPad Air | Amazon Fire Phone | Nexus 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 17 | 20 | 21 | 16 | 17 | 7 | 9 | 6 | 14 | 18 | 27 | 4 | 16 |
| 18 | 33 | 42 | 19 | 33 | 15 | 26 | 13 | 37 | 58 | 52 | 9 | 30 |
| 19 | 41 | 27 | 30 | 23 | 11 | 23 | 18 | 40 | 36 | 55 | 12 | 25 |
| 20 | 40 | 41 | 29 | 22 | 18 | 23 | 26 | 30 | 52 | 59 | 11 | 22 |
| 21 | 47 | 38 | 21 | 25 | 20 | 29 | 18 | 44 | 41 | 51 | 5 | 25 |
| 22 | 41 | 52 | 24 | 25 | 19 | 25 | 25 | 38 | 52 | 58 | 5 | 27 |
| 23 | 43 | 49 | 20 | 31 | 14 | 22 | 18 | 54 | 53 | 41 | 16 | 45 |
| 24 | 40 | 43 | 20 | 35 | 20 | 24 | 29 | 56 | 59 | 57 | 11 | 38 |
| 25 | 47 | 38 | 21 | 37 | 14 | 28 | 21 | 52 | 52 | 57 | 13 | 29 |
| 26 | 35 | 49 | 23 | 42 | 9 | 29 | 11 | 46 | 60 | 56 | 13 | 29 |
| 27 | 49 | 52 | 27 | 31 | 15 | 29 | 15 | 56 | 53 | 55 | 10 | 37 |
| 28 | 49 | 50 | 26 | 35 | 10 | 30 | 28 | 56 | 56 | 54 | 6 | 26 |
| 29 | 53 | 49 | 31 | 43 | 16 | 28 | 31 | 58 | 54 | 52 | 12 | 25 |
| 30 | 60 | 56 | 31 | 34 | 15 | 33 | 23 | 42 | 54 | 70 | 12 | 36 |
| 31 | 55 | 56 | 13 | 28 | 14 | 31 | 24 | 51 | 44 | 55 | 14 | 24 |
| 32 | 55 | 62 | 18 | 28 | 12 | 35 | 20 | 51 | 57 | 48 | 12 | 30 |
| 33 | 46 | 49 | 19 | 27 | 13 | 39 | 32 | 38 | 37 | 40 | 14 | 23 |
| 34 | 63 | 47 | 25 | 17 | 13 | 30 | 30 | 36 | 49 | 39 | 11 | 25 |
| 35 | 3 | 6 | 2 | 2 | 1 | 1 | 2 | 1 | 1 | 0 | 0 | 2 |

## 5.Calculate Email Engagement Metrics

To calculate email engagement metrics, the following SQL query was executed:

```
SELECT Week,
ROUND((weekly_digest / total * 100), 2) AS "Weekly Digest Rate",
ROUND((email_opens / total * 100), 2) AS "Email Open Rate",
ROUND((email_clickthroughs / total * 100), 2) AS "Email Clickthrough Rate",
ROUND((reengagement_emails / total * 100), 2) AS "Reengagement Email Rate"
FROM
(
   SELECT EXTRACT(WEEK FROM occurred_at) AS Week,
   COUNT(CASE WHEN action = 'sent_weekly_digest' THEN user_id ELSE NULL
END) AS weekly_digest,
   COUNT(CASE WHEN action = 'email_open' THEN user_id ELSE NULL END) AS
email_opens,
   COUNT(CASE WHEN action = 'email_clickthrough' THEN user_id ELSE NULL
END) AS email_clickthroughs,
   COUNT(CASE WHEN action = 'sent_reengagement_email' THEN user_id ELSE
NULL END) AS reengagement_emails,
   COUNT(user_id) AS total
   FROM email_events
   GROUP BY 1
) subre-engagement
GROUP BY 1
ORDER BY 1;
```

| Week | Weekly Digest Rate | Email Open Rate | Email Clickthrough Rate | Reengagement Email Rate |
|---|---|---|---|---|
| 17 | 62.32 | 21.28 | 11.39 | 5.01 |
| 18 | 63.45 | 22.24 | 10.49 | 3.83 |
| 19 | 62.16 | 22.67 | 11.13 | 4.04 |
| 20 | 61.62 | 22.64 | 11.43 | 4.31 |
| 21 | 63.52 | 22.82 | 9.97 | 3.69 |
| 22 | 63.59 | 21.56 | 10.66 | 4.19 |
| 23 | 62.39 | 22.34 | 11.18 | 4.09 |
| 24 | 61.61 | 22.92 | 10.99 | 4.48 |
| 25 | 63.77 | 21.79 | 10.54 | 3.90 |
| 26 | 62.99 | 22.22 | 10.61 | 4.18 |
| 27 | 62.24 | 22.49 | 11.37 | 3.90 |
| 28 | 62.92 | 22.48 | 10.77 | 3.83 |
| 29 | 63.98 | 21.71 | 10.51 | 3.79 |
| 30 | 62.29 | 23.24 | 10.59 | 3.88 |
| 31 | 65.27 | 23.25 | 7.66 | 3.82 |
| 32 | 66.59 | 22.85 | 7.14 | 3.42 |
| 33 | 64.73 | 23.10 | 7.91 | 4.26 |
| 34 | 64.33 | 23.91 | 7.67 | 4.08 |
| 35 | 0.00 | 32.28 | 29.92 | 37.80 |

*This query calculates various email engagement metrics on a weekly basis. It calculates the weekly digest rate, email open rate, email clickthrough rate, and reengagement email rate. The result provides insights into how users are engaging with email content over time, which is essential for assessing the effectiveness of email campaigns and user retention through email engagement.*

# Project Impact and Conclusion

**Project Impact:**

This project has provided valuable insights into the realm of operational analytics and the power of investigating metric spikes. It has illuminated how companies utilize these insights as a secret weapon to drive success. By adopting an informed and proactive approach, organizations can harness these insights to make data-backed decisions, optimize their strategies, and significantly enhance their return on investment (ROI). The project has served as a pivotal learning experience, highlighting the critical role of data analytics in modern business operations.

**Challenges Overcome:**

Throughout the project, several challenges were encountered and effectively addressed. Notably, the sheer volume of data in case study 2 posed a significant hurdle, causing performance issues within SQL Workbench. To tackle this situation, the use of LOAD DATA statements was employed, streamlining the data import process. Additionally, a critical issue involving the "user_type" column in the events table, which had an incompatible data type, was resolved by converting it to the appropriate data type (text). These challenges underscored the importance of adaptability and problem-solving in the world of data analytics.

**Conclusion:**

Operational analytics stands as a robust solution for synchronizing real-time data within an organization. It enables the aggregation of data from multiple sources, culminating in a comprehensive, organized, and actionable solution. This approach empowers companies to develop analytical models in real-time, create individual customer profiles, and gain a holistic view of their operations. The result is increased efficiency in operational routines and systems, ultimately leading to a significant positive impact on specific areas of business operations. Operational analytics is the key to unlocking untapped potential and driving excellence in modern organizations.