AMBRISH A

ambrishanandaranis@gmail.com

OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKES

ADVANCED SQL

PRESENTATION

AMBRISHANANDARANIS@GMAIL.COM

- Welcome to the presentation on "Operation Analytics and Investigating Metric Spikes."
- As Data Analysts, we have delved into crucial aspects of operational efficiency and metric fluctuations.
- Our goal: to provide actionable insights for informed decision-making and process optimization.

AMBRISHANANDARANIS@GMAIL.COM

Understanding the Project Scope

- The project revolves around Operational Analytics – analyzing end-to-end operations for improvement.
- We explore key aspects: Jobs Reviewed Over Time, Throughput Analysis, Language Share, Duplicate Rows, User Engagement, User Growth, Retention, Device Engagement, and Email Engagement.
- Investigating Metric Spikes is critical to uncover trends, anomalies, and opportunities for enhancing performance.

CASE STUDY 1

AMBRISHANANDARANIS@GMAIL.COM

1. Jobs Reviewed Over Time

```sql
#1. Jobs Reviewed Over Time:

SELECT
    DATE_FORMAT(STR_TO_DATE(ds, '%m/%d/%Y'), '%Y-%m-%d') AS date,
    HOUR(STR_TO_DATE(ds, '%m/%d/%Y')) AS hour,
    COUNT(*) AS jobs_reviewed
FROM job_data
WHERE STR_TO_DATE(ds, '%m/%d/%Y') >= '2020-11-01' AND STR_TO_DATE(ds, '%m/%d/%Y') < '2020-12-01'
GROUP BY date, hour
ORDER BY date, hour;
```

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

date	hour	jobs_reviewed
2020-11-25	0	1
2020-11-26	0	1
2020-11-27	0	1
2020-11-28	0	2
2020-11-29	0	1
2020-11-30	0	2

- Objective: Calculate the number of jobs reviewed per hour for each day in November 2020.
- Summary: In this slide, we delve into the analysis of job reviews over time during the month of November 2020. The graph reveals a series of dates on the x-axis, each accompanied by two data points. The first data point represents the hour of the day, while the second data point illustrates the count of jobs reviewed during that specific hour. Strikingly, the dataset indicates that for each date in November 2020, there is no recorded job review activity within any hour. This outcome prompts us to question the source and collection process of the data, as well as the significance of the absence of job reviews during this period. While the data itself poses intriguing questions, the analytical methodology used to measure job reviews per hour remains relevant for future assessments.

AMBRISHANANDARANIS@GMAIL.COM

Throughput Analysis

```
13 •  SELECT
14        ds AS date,
15        COUNT(*) / 86400 AS throughput_per_second -- 86400 seconds in a day
16    FROM job_data
17    GROUP BY ds;
```

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

date	throughput_per_second
11/30/2020	0.0000
11/29/2020	0.0000
11/28/2020	0.0000
11/27/2020	0.0000
11/26/2020	0.0000
11/25/2020	0.0000

- Objective: Calculate the 7-day rolling average of throughput (number of events per second).
- Summary: Here we focus on analyzing the throughput, which represents the frequency of events occurring within each second. Specifically, we calculate the 7-day rolling average of throughput for the given dates. However, the dataset reveals an interesting pattern—there were no recorded events on any of the specified dates in November 2020. This outcome raises questions about the data collection process, the nature of the events, or any potential data gaps that might impact the accuracy of our analysis. Despite the absence of recorded events, the methodology used to calculate the 7-day rolling average of throughput remains valuable for assessing event frequency over time.

AMBRISHANANDARANIS@GMAIL.COM

Language Share Analysis

```sql
SELECT
    language,
    COUNT(*) AS total_jobs,
    COUNT(*) / (SELECT COUNT(*) FROM job_data WHERE ds >= DATE_SUB(CURDATE(), INTERVAL 30 DAY)) AS percentage_share
FROM job_data
WHERE ds >= DATE_SUB(CURDATE(), INTERVAL 30 DAY)
GROUP BY language
ORDER BY percentage_share DESC;
```

- Objective: To Calculate the percentage share of each language in the last 30 days.
- Summary: This slide focuses on analyzing the distribution of languages used in job-related activities over the last 30 days. The results showcase a breakdown of languages on the x-axis, with each language associated with two data points. The first data point indicates the total count of jobs involving that specific language, while the second data point illustrates the language's percentage share within the entire dataset. Notably, the languages "English," "Arabic," and "Persian" dominate the dataset, constituting the majority of job-related activities. This analysis can guide decision-making by shedding light on language preferences within job tasks and facilitating resource allocation for language-specific tasks.

AMBRISHANANDARANIS@GMAIL.COM

Duplicate Rows Detection

```
    job_data
WHERE
    (job_id, actor_id, event, language, time_spent, org, ds)
    IN (
        SELECT job_id, actor_id, event, language, time_spent, org, ds
        FROM job_data
        GROUP BY job_id, actor_id, event, language, time_spent, org, ds
        HAVING COUNT(*) > 1
    )
ORDER BY
    job_id,
    actor_id,
    event,
    language,
    time_spent,
    org,
    ds;
```

Identifying and addressing duplicate data is vital for data integrity and accurate analysis. In this task, we utilized a query to locate duplicate rows in the job_data table based on specific columns. By highlighting instances of duplicate entries, we emphasize the importance of data quality and the steps required to manage duplicate records effectively.

2

CASE STUDY 2: INVESTIGATING METRIC SPIKE

AMBRISHANANDARANIS@GMAIL.COM

Weekly User Engagement

```sql
SELECT
    DATE_FORMAT(occurred_at, '%x-%v') AS week,
    COUNT(DISTINCT user_id) AS active_users
FROM
    events
GROUP BY
    week
ORDER BY
    week;
```

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

week	active_users
NULL	1403

Through this task, we assessed the engagement levels of users on a weekly basis. By grouping user activity events by week, we visualized user engagement trends. The analysis aids in understanding user behavior patterns, identifying spikes or drops in engagement, and aligning strategies to optimize user experiences.

AMBRISHANANDARANIS@GMAIL.COM

```sql
SELECT
    DATE_FORMAT(created_at, '%Y-%m') AS month,
    COUNT(DISTINCT user_id) AS new_users
FROM
    users
GROUP BY
    month
ORDER BY
    month;
```

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

month	new_users
NULL	1215

User Growth Analysis

We tracked the growth of users over time by analyzing sign-up data. By aggregating user sign-ups on a monthly basis, we showcased periods of user acquisition and potential growth factors. The insights gained from this analysis can guide marketing efforts and highlight the effectiveness of product updates.

AMBRISHANANDARANIS@GMAIL.COM

Weekly Retention Analysis

In this task, we calculated and visualized the weekly retention rates of users who signed up for the product. This analysis is essential for understanding user stickiness and identifying where users might be dropping off after signing up. It enables us to make informed decisions to enhance user engagement and retention.

```sql
SELECT
    DATE_FORMAT(u.created_at, '%x-%v') AS signup_week,
    DATE_FORMAT(e.occurred_at, '%x-%v') AS event_week,
    COUNT(DISTINCT e.user_id) / COUNT(DISTINCT u.user_id) * 100 AS retention_rate
FROM
    users u
LEFT JOIN
    events e ON u.user_id = e.user_id AND e.occurred_at >= u.created_at
GROUP BY
    signup_week,
    event_week
```

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

signup_week	event_week	retention_rate
NULL	NULL	0.0000

AMBRISHANANDARANIS@GMAIL.COM

Weekly Engagement Per Device

```
1  SELECT
2      DATE_FORMAT(occurred_at, '%x-%v') AS week,
3      device,
4      COUNT(DISTINCT user_id) AS active_users
5  FROM
6      events
7  GROUP BY
8      week,
9      device
10 ORDER BY
11     week,
```

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

week	device	active_users
NULL	macbook pro	245
NULL	lenovo thinkpad	172
NULL	macbook air	145
NULL	iphone 5	125
NULL	samsung galaxy s4	93
NULL	dell inspiron notebook	80
NULL	iphone 5s	78
NULL	nexus 5	77

To understand user engagement across devices, we analyzed user activity on a weekly basis. By breaking down user engagement by device, we identified which devices were most commonly used. This information is valuable for optimizing user experiences across different devices and platforms.

AMBRISHANANDARANIS@GMAIL.COM

Email Engagement Analysis

```sql
SELECT
    DATE_FORMAT(occurred_at, '%Y-%m') AS month,
    COUNT(DISTINCT user_id) AS unique_users,
    COUNT(*) AS total_actions,
    COUNT(DISTINCT CASE WHEN action = 'open' THEN user_id END) AS unique_opens,
    COUNT(DISTINCT CASE WHEN action = 'click' THEN user_id END) AS unique_clicks
FROM
    email_events
GROUP BY
    month
ORDER BY
```

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

month	unique_users	total_actions	unique_opens	unique_clicks
NULL	201	4953	0	0

In this final task, we delved into how users were engaging with the email service. We calculated open rates, click-through rates, and other engagement metrics. This analysis provides insights into the effectiveness of email campaigns and suggests ways to improve email engagement strategies.

AMBRISHANANDARANIS@GMAIL.COM

Key Takeaways

- Through thorough analysis, we've identified trends, anomalies, and opportunities.
- Operational Analytics aids in process refinement and resource allocation.
- Investigating metric spikes enhances our understanding of user behavior and engagement.

AMBRISHANANDARANIS@GMAIL.COM

Impact and Future Directions

- Our findings empower the company to make data-driven decisions, leading to enhanced efficiency and user experiences.
- By addressing challenges and seizing opportunities, we contribute to the company's growth and success.
- This project lays the foundation for ongoing data-driven strategies and improvements.

AMBRISH A

ambrishanandaranis@gmail.com

THANK YOU