Operation Analytics and Investigating Metric Spike

PROJECT DESCRIPTION

The project widely referred to as "Operational Analytics and Investigating Metric Spikes," conducted comprehensive analysis into the end-to-end operations of an organization, using advanced SQL analysis techniques. The primary function of the use-case contained within the data, was for department use, specifically operations and marketing, with the aim of helping them uncover learning and draw ideas from the respective departments. The main concepts of the analysis included discovering and interpretation of key performance metrics, and the additional phase of investigating any unanticipated changes or "spikes" to these metrics. The purpose of this investigation was to help improve overall productivity and make data-driven decisions.

APPROACH

My approach consisted of two separate case studies. Each case study followed the same procedure, as described below:

- **Data Import:**
  The first phase consisted of getting the provided CSV files into a MySQL database. I utilized the Table Data Import Wizard, and where possible for CSV imports, I utilized the LOAD DATA INFILE command to develop efficiency in importing the data. I also imported the data into the MySQL Server to make adjustments to correct challenges associated with the import; for example, many dates had incorrect formatting for data type as a VARCHAR type due to characters used in the process preventing it from importing it as predicted DATETIME types and most dates needed to be adjusted to upload the type to VARCHAR and then use different SQL functions like the STR_TO_DATE() function to adopt a DATETIME or timestamp format to complete the file analysis.

- **SQL Processing:**
  I performed some advance SQL processing to address the specific business questions posed for each case study. I utilized aggregate functions, window functions (for rolling averages), and common table expressions (CTEs) in a few cases to enable complex calculations and cohort analysis.

- **Insights:**
After I had all the outputs from the queries, I examined them closely and thought carefully about how these outputs translate into actionable insights. For each task, I provided a concise interpretation of the outputs, describing the information the data provided with respect to operational efficiency and user engagement.

TECHNOLOGY-STACK UTILIZED

- **MySQL Workbench:** The core application utilized for the project as the primary database management tool, as well as the data import tool, and executing SQL queries.

- **MySQL Server:** The database server that was the backend utilized for the data storage and manipulation.

INSIGHTS

Case Study 1: Job Data Analysis

- Operational Efficiency: The use of jobs_per_hour was a great indicator of operational efficiency for job reviewers. This can help in the development of performance marks and to discover any restrictors.
- Throughput Volatility: Once comparing the throughput on a day-to-day basis against the 7-day rolling average, I understood that the 7-day rolling average was a much more stable and reliable indicator of throughput. This was an important differentiation to make because it ultimately will rub against the concept of overreacting to daily volatility and approaching long-term data driven decisions.
- Duplicate Data: The query for duplicate rows proved that the dataset provided has no duplicate rows, which is key to the integrity of the data.

Task 1: Jobs Reviewed Over Time

- **Query**:

```
SELECT ds,
    COUNT(job_id) AS total_jobs,
    SUM(time_spent) / 3600 AS total_hours_spent,
```

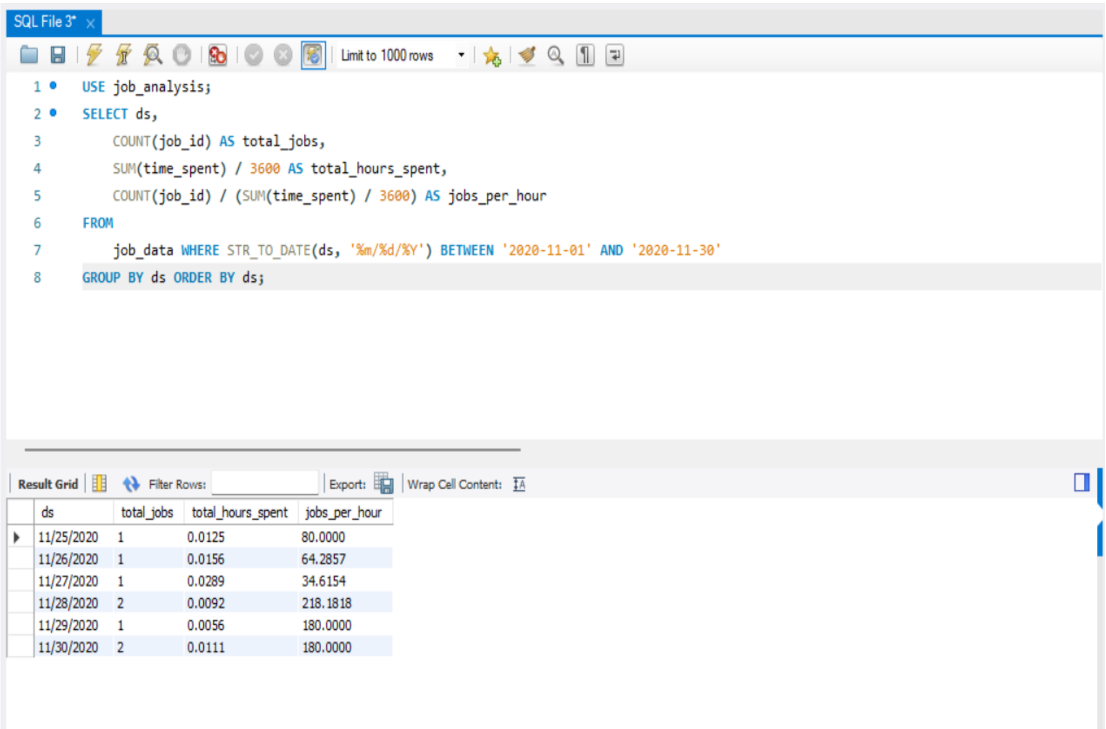
COUNT(job_id) / (SUM(time_spent) / 3600) AS jobs_per_hour

FROM job_data WHERE STR_TO_DATE(ds, '%m/%d/%Y') BETWEEN '2020-11-01' AND '2020-11-30'

GROUP BY ds ORDER BY ds;

- **Insight:**
  This query provides a clear operational efficiency metric by presenting the number of jobs reviewed per hour each day. The degree of consistency tells us if the workflow is steady, while any considerable inconsistencies could indicate a need to investigate the efficiency of the workflow or resource allocation.

- **Output:**



Task 2: Throughput Analysis

- **Query**:

WITH DailyThroughput AS (

SELECT

```
        ds, COUNT(event) AS events, SUM(time_spent) AS total_time,
        COUNT(event) / SUM(time_spent) AS daily_throughput
    FROM job_data GROUP BY ds
)
SELECT
    ds, daily_throughput,
    AVG(daily_throughput) OVER ( ORDER BY ds ROWS BETWEEN 6
PRECEDING AND CURRENT ROW ) AS rolling_avg_throughput
FROM DailyThroughput ORDER BY ds;
```

- **Insight**:
  The rolling average over 7 days is a much better long-term efficiency metric than the daily one. The rolling average smooths out the day-to-day inconsistencies and allows us to avoid overreacting to day-to-day spikes or drops and provides a better illustration of the direction and trend of job review efficiency.

- **Output:**

ds	daily_throughput	rolling_avg_throughput
11/25/2020	0.0222	0.02220000
11/26/2020	0.0179	0.02005000
11/27/2020	0.0096	0.01656667
11/28/2020	0.0606	0.02757500
11/29/2020	0.0500	0.03206000
11/30/2020	0.0500	0.03505000

Task 3: Language Share Analysis:

- **Query:**

```
SELECT
    language,
```

```
    (COUNT(job_id) * 100.0) / (SELECT COUNT(job_id) FROM job_data) AS percentage_share
FROM
    job_data GROUP BY language ORDER BY percentage_share DESC;
```

- **Insight:**
  The analysis provides an understanding of how all of the jobs are distributed by language, which is a potentially useful metric for product and content teams. It provides the teams with an understanding of which languages are most frequently represented in the job queue as well as an opportunity to inform content strategy, or identify possible language performance issues if one language has a disproportionately low level of throughput.

- **Output:**

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

language	percentage_share
Persian	37.50000
English	12.50000
Arabic	12.50000
Hindi	12.50000
French	12.50000
Italian	12.50000

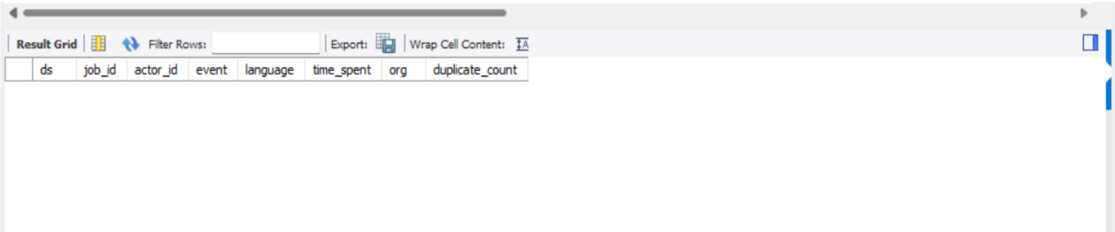
Task 4: Duplicate Rows Detection

- **Query**:

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

- **Insight:** The query completed successfully and produced no output. This means there were no duplicate rows of data in the dataset that was provided. This is an important discovery as it confirms that the data is of high quality and that it is trustworthy, which is the most important foundational aspect of reliable analysis.

- **Output:**



Case Study 2: Investigating Metric Spike

- User Engagement: Tracking weekly user metric engagement and engagement by device is critical for understanding user activity. Any drops or spikes in these metrics could indicate a bug, a successful feature launch, or changing user behavior.
- User Retention: The retention analysis, which looks at sign-ups cohorts, is a critical measure of how "sticky" this product is. A declining retention rate over time, may indicate the need for improved user experience or more compelling features.
- Email Effectiveness: This analysis of email engagement helps understand the effectiveness of campaigns, which may include open rates and click-throughs. This can help the marketing team think through their content decisions, timing, and audience segmentation.

Loading csv:

Query:

Use metric_spike;

Alter table users add column temp_occurred_at2 DATETIME;

Update users set temp_occurred_at2 =STR_TO_DATE(created_at, '%d-%m-%Y %H:%i');

```
Alter table users DROP column created_at;
Alter table users change column temp_occurred_at2 created_at DATETIME;

Create table events(
  user_id INT,
  occurred_at varchar(100),
  event_type varchar(50),
  event_name varchar(100),
  location varchar(50),
  device varchar(50),
  user_type INT
  );

  SHOW VARIABLES LIKE 'secure_file_priv';

  SET GLOBAL local_infile = 1;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/events.csv'
INTO TABLE events
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

Select *from events;
ALTER TABLE events
RENAME COLUMN occurred_at TO occured_at;
```

```
ALTER TABLE events
RENAME COLUMN occured_at TO occurred_at;

Alter table events add column temp_occurred_at DATETIME;
SET SQL_SAFE_UPDATES = 0;
Update events set temp_occurred_at =STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i');
Alter table events DROP column occurred_at;
Alter table events change column temp_occurred_at occurred_at DATETIME;

Create table email_events(
user_id INT,
occurred_at varchar(100),
action varchar(100),
user_type int);

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/email_events.csv'
INTO TABLE email_events
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

Alter table email_events add column temp_occurred_at1 DATETIME;
Update email_events set temp_occurred_at1 =STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i');
```

Alter table email_events DROP column occurred_at;

Alter table email_events change column temp_occurred_at1 occurred_at DATETIME;

Select *from email_events ;

Task 1: Weekly User Engagement

- **Query:**

  SELECT

  YEAR(occurred_at) AS year,

  WEEK(occurred_at) AS week_number,

  COUNT(DISTINCT user_id) AS weekly_active_users

  FROM events

  GROUP BY year, week_number

  ORDER BY year, week_number;

- **Insights:**
  A clear measure of product health. When there is a steady or confirming number of weekly active users (WAU), it means the user base is using the product with some level of engagement. Decreasing WAU levels would show up as a metric spike that would warrant immediate investigation into the problem.

- **Output:**

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

year	week_number	weekly_active_users
2014	17	663
2014	18	1068
2014	19	1113
2014	20	1154
2014	21	1121
2014	22	1186
2014	23	1232
2014	24	1275
2014	25	1264
2014	26	1302
2014	27	1372
2014	28	1365
2014	29	1376
2014	30	1467
2014	31	1299
2014	32	1225
2014	33	1225
2014	34	1204
2014	35	104

Task 2: User Growth Analysis

- **Query:**

```
SELECT
   YEAR(created_at) AS year,
   WEEK(created_at) AS week_number,
   COUNT(user_id) AS new_users
FROM users
   GROUP BY year, week_number
ORDER BY year,week_number;
```

- **Insights:**
  This metric is a straight measure of acquisition channel success. As the WAU number fluctuates in response to your marketing campaigns or product updates, the marketing and product teams can connect other data like changes in sign-ups with changes in WAU. A drop-off in WAU levels is a warning sign that something is going on with either the onboarding funnel for that sign up cohort or optimization in marketing is losing effectiveness.

- **Output:**

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

year	week_number	new_users
2013	0	23
2013	1	30
2013	2	48
2013	3	36
2013	4	30
2013	5	48
2013	6	38
2013	7	42
2013	8	34
2013	9	43
2013	10	32
2013	11	31
2013	12	33
2013	13	39
2013	14	35
2013	15	43
2013	16	46
2013	17	49
2013	18	44

Task 3: Weekly Retention Analysis

- **Query:**

```
WITH user_cohort AS (
   SELECT
      user_id,
      created_at AS signup_date
   FROM
```

```
        users
),
weekly_activity AS (
    SELECT
        user_id,
        occurred_at AS activity_date
    FROM
        events
)
SELECT
    YEAR(uc.signup_date) AS signup_year,
    WEEK(uc.signup_date) AS signup_week,
    FLOOR(DATEDIFF(wa.activity_date, uc.signup_date) / 7) AS
weeks_since_signup,
    COUNT(DISTINCT wa.user_id) AS retained_users
FROM
    user_cohort uc
JOIN
    weekly_activity wa ON uc.user_id = wa.user_id
WHERE
    uc.signup_date IS NOT NULL AND wa.activity_date IS NOT NULL
GROUP BY
    signup_year,
    signup_week,
    weeks_since_signup
ORDER BY
    signup_year,
    signup_week,
    weeks_since_signup;
```

- **Insights:**
  A meaningful metric in gauging your product targeting and stickiness. If most of your users come from one cohort and they undoubtfully retain their engagement, it is a strong sign that they see long-term value in the product. A drop-off in retention for that cohort would tell you a certain feature or poor UX may have led to them disengaging from your product.

- **Output:**

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

signup_year	signup_week	weeks_since_signup	retained_users
2013	0	69	3
2013	0	70	3
2013	0	71	3
2013	0	72	3
2013	0	73	3
2013	0	74	3
2013	0	75	3
2013	0	76	6
2013	0	77	4
2013	0	78	1
2013	0	79	2
2013	0	80	1

Task 4: Weekly Engagement Per Device

- **Query:**

```
SELECT
   YEAR(occurred_at) AS year,
   WEEK(occurred_at) AS week_number,
   device,
   COUNT(DISTINCT user_id) AS weekly_active_users
FROM events
GROUP BY year, week_number, device
ORDER BY year, week_number, device;
```

- **Insights:**

A good analysis identifying which platforms provide the most engagement. If big differences are observed in the engagement level of a given device (i.e., on iPhone it is high, but not on Dell Inspiron), then opportunities have been identified to reduce friction and improve user experience on the platform(s) that are falling short.

- **Output:**

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

year	week_number	device	weekly_active_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
2014	17	lenovo thinkpad	86
2014	17	mac mini	6
2014	17	macbook air	54
2014	17	macbook pro	143
2014	17	nexus 10	16

Task 5: Email Engagement Analysis

- **Query:**

```
SELECT
    YEAR(occurred_at) AS year,
    WEEK(occurred_at) AS week_number,
    action,
    COUNT(user_id) AS total_actions
FROM email_events
GROUP BY year, week_number, action
ORDER BY year, week_number, action;
```

- **Insights:**
  This metric is very important for the marketing group as it helps them quantify the performance of their email campaigns. A low open rate and/or low click-through rate could mean the subject line, content, or timing of the email needs altered or optimized.

- **Output:**

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

year	week_number	action	total_actions
2014	17	email_clickthrough	166
2014	17	email_open	310
2014	17	sent_reengagement_email	73
2014	17	sent_weekly_digest	908
2014	18	email_clickthrough	430
2014	18	email_open	912
2014	18	sent_reengagement_email	157
2014	18	sent_weekly_digest	2602
2014	19	email_clickthrough	477
2014	19	email_open	972
2014	19	sent_reengagement_email	173
2014	19	sent_weekly_digest	2665
2014	20	email_clickthrough	507
2014	20	email_open	1004
2014	20	sent_reengagement_email	191
2014	20	sent_weekly_digest	2733
2014	21	email_clickthrough	443
2014	21	email_open	1014
2014	21	sent_reengagement_email	164

CONCLUSION:

This project was able to successfully illustrate the essential nature of data analytics for better understanding and improving business processes. By engaging with two separate case studies, I was able to mould raw data into insight for different areas of the business.

The analysis of Case Study 1 (Researchers Job Data Analysis) allowed me to develop an appreciation of operational effectiveness with some useful operational metrics including throughput and job review counts. Further to this, I learned the importance of observing smoothed metrics such as a rolling 7-day average in order to make observations when trends begin to emerge and to avoid overreacting to daily changes in comparisons as they may lead us to inaccurate conclusions. My analysis was able to confer the integrity of the dataset I was provided and provide a basis for forming an intuition that will be built upon going forward as my group continues to monitor performance.

Case Study 2 (Investigating Metric Spikes) took a different focus in that it concentrated more closely on user behavior and engagement. By measuring cohorts of users, interested activity by device, and email engagement - covered several key metrics including Weekly Active Users (WAU) and retention rates. The most important contribution here was the development of a sustainable framework that provides a basis to identify and investigate "metric spikes" - or sudden deviations of key performance indicators (KPI) that occur without expectation of validity or being flagged prior the observation being made.

To summarize, this project reiterated the importance of a structured, data-driven approach to problems of operational relevance. Skills in preparing and cleansing data; querying in a more advanced SQL format; and developing insights can be employed to help businesses extract meaningful intelligence from their data to improve business performance.