Operation Analytics and Investigating Metric Spike

TRAINITY PROJECT-3

Project Description

- Operational analysis is the process of examining and understanding a company's day-to-day operations with the goal of identifying areas for improvement and optimization. It involves using data, insights, and analytical techniques to assess various aspects of a company's operations.
- One of the key aspects of Operational Analytics is investigating metric spikes. This involves understanding and explaining sudden changes in key metrics, such as a dip in daily user engagement or a drop in sales. In this project,2 cases are given “CASE STUDY 1” and “CASE STUDY 2” respectively. The goal of this project is to use your advanced SQL skills to analyze the data and provide valuable insights that can help improve the company's operations and understand sudden changes in key metrics.

APPROACH

I imported the dataset given by the team and extracted csv file and cleaned the data in excel sheet to improve data quality and analyze accurately. After importing dataset in MYSQL Workbench I carefully examined the values placed in each column of each table . Done some modification in tables in order to achieve accurate results.

Finally, run the appropriate MYSQL queries according to the given problem statement provided by team to find desirable output tables which will help the different departments of the organization/company.

TECH-STACK USED

MY SQL WOREKBENCH 8.0 CE:

- MySQL Workbench 8.0 CE is a free, open-source tool. It provides a user-friendly interface for visually designing and editing database schemas, writing and executing SQL queries, importing and exporting data, managing multiple servers, and analyzing query performance.

MICROSOFT EXCEL:

- Microsoft Excel is a ubiquitous spreadsheet application that empowers users to organize, analyze, and visualize data .Its versatility and widespread adoption make it an indispensable tool for individuals and businesses across various industries.

MICOSOFT POWER POINT:

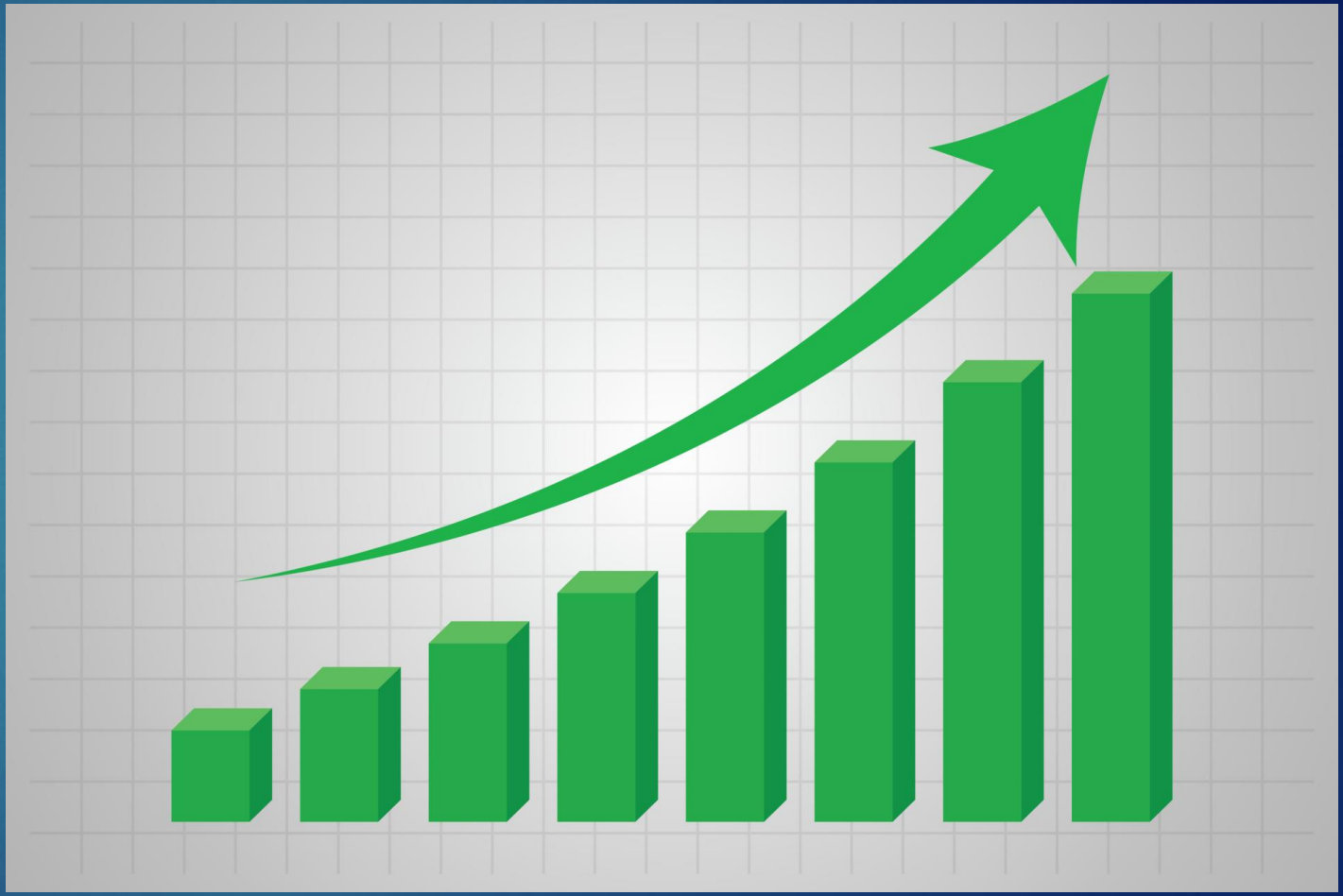
- Microsoft PowerPoint is a presentation software that allows you to create visually engaging slideshows.

INSIGHTS

I completed this project by using advance SQL learned from TRAINITY platform and gained knowledge about various advance SQL queries which are used in real-world problems. This project helped me to understand how the role of a Lead Data Analyst at a company like Microsoft works and uses his/her data analyst skills to contribute to the company's growth . Furthermore, I understood how to work with various datasets and tables and find out meaningful insights from that. Overall, the project contributes to the deeper understanding of key performance indicators and facilitates data-driven decision-making within the company/organization.

Case study-1

Job Data Analysis



A) JOBS REVIEWED OVER TIME

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

```
#A) jobs reviewed over time

select ds as date,
count(job_id) as job_review_counts,
round(count(job_id)/(sum(time_spent)/(60*60)),2) as job_review_per_hour_each_day
from jobdata
where ds between '01-11-2020' and '30-11-2020'
group by ds
order by ds;
```

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

date	job_review_counts	job_review_per_hour_each_
11/25/2020	1	80.00
11/26/2020	1	64.29
11/27/2020	1	34.62
11/28/2020	2	218.18
11/29/2020	1	180.00
11/30/2020	2	180.00

The most number of jobs are reviewed on 28th and 30th November, however the time taken by both is varied.

B) THROUGHPUT ANALYSIS

Task: Write an SQL query to calculate the 7-day rolling average of throughput. (number of events per second).

```
#B) Throughput analysis
WITH total_events AS ( SELECT date(ds)
AS event_date, COUNT(job_id) AS
total_jobs, SUM(time_spent) AS
total_time_spent
FROM jobdata
GROUP BY DATE(ds))
SELECT total_jobs / (total_time_spent / 3600
AS throughput, AVG(total_jobs /
(total_time_spent / 3600)) OVER ( ORDER
BY event_date ROWS BETWEEN 6
PRECEDING AND CURRENT ROW ) AS
rolling_average
FROM total_events;
```

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

throughput	rolling_average
96.6443	96.64430000

C) LANGUAGE SHARE ANALYSIS

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

```sql
#C) language analysi

SELECT
    language,
    (COUNT(language) / (SELECT
            COUNT(language)
        FROM
            jobdata)) * 100 AS percentage
FROM
    jobdata
GROUP BY language;
```

sult Grid | Filter Rows:

language	percentage
English	12.5000
Arabic	12.5000
Persian	37.5000
Hindi	12.5000
French	12.5000
Italian	12.5000

D) DUPLICATE ROWS DETECTION

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

```
#D) duplicate row  detetction

SELECT
    job_id, COUNT(*) AS duplicate
FROM
    jobdata
GROUP BY job_id
HAVING COUNT(*) > 1;
```

Result Grid | Filter Rows:

job_id	duplicate
23	3

There are 3 duplicates whose job_id is 23.

IMPORTING LARGE DATA-SETS

```sql
#table1 : users
create table users(
user_id int,
created_at varchar(100),
company_id int,
language varchar(50),
activated_at varchar(100),
state varchar(50) );
Show variables like "secure_file_priv";
LOAD DATA INFILE "C:/users (1).csv"
into table users
fields terminated by ","
ENCLOSED BY '"'
LINES TERMINATED BY "\n"
IGNORE 1 ROWS;
select * from users;
ALTER TABLE users ADD COLUMN temp_created_at DATETIME;
ALTER TABLE users RENAME COLUMN temp_created_at TO temp_created_at_old;
SELECT created_at FROM users
WHERE STR_TO_DATE(created_at, '%d-%m-%Y %H:%i:%s') IS NULL;
UPDATE users SET temp_created_at_old = STR_TO_DATE(created_at, '%d-%m-%Y %H:%i:%s')
WHERE STR_TO_DATE(created_at, '%d-%m-%Y %H:%i:%s') IS NOT NULL;
alter table users drop column created_at;
alter table users change column temp_created_at_old created_at datetime;
```

USER TABLE

```sql
#creating table-2 events
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 );

LOAD DATA INFILE "C:/events.csv"
INTO TABLE events
FIELDS TERMINATED BY ","
ENCLOSED BY '"'
LINES TERMINATED BY "\n"
IGNORE 1 ROWS;
select*from events;
ALTER TABLE events ADD COLUMN temp_occured_at DATETIME;
SET SQL_SAFE_UPDATES = 0;
UPDATE events
SET temp_occured_at = STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i:%s')
WHERE STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i:%s') IS NOT NULL;
alter table events drop column occurred_at;
alter table events change column temp_occured_at occurred_at datetime;
```

EVENTS TABLE

```sql
# table-3 emailEvents
create table emailEvents(
user_id int,
occurred_at varchar(100),
action varchar(100),
user_type int
);
LOAD DATA INFILE "C:/email_events.csv"
INTO TABLE emailevents
FIELDS TERMINATED BY ","
ENCLOSED BY '"'
LINES TERMINATED BY "\n"
ignore 1 rows;

select*from emailevents;
ALTER TABLE emailevents ADD COLUMN temp_occured_at DATETIME;
UPDATE emailevents
SET temp_occured_at = STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i:%s')
WHERE STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i:%s') IS NOT NULL;

alter table emailevents drop column occurred_at;
alter table emailevents change column temp_occured_at occurred_at datetime;
```

EMAIL_EVENTS TABLE

CASE-STUDY 2

INVESTIGATING METRIC SPIKE

A) Weekly User Engagement

Task: Write an SQL query to calculate the weekly user engagement.

```sql
#1) weekly user engagemenet

SELECT extract(week from occurred_at) as week_number,
count(distinct user_id) as num_users
from events
where event_type = 'engagement'
group by week_number
order by week_number;
```

week_number	num_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

Result 1

B) User Growth Analysis

Task: Write an SQL query to calculate the user growth for the product.

```sql
#B) user grouth analysis

WITH totalUsers AS ( SELECT
WEEK(created_at) AS week,
COUNT(*) AS users_2 FROM users
GROUP BY WEEK(created_at))
SELECT week,users_2,((users_2-
LAG(users_2) OVER (ORDER BY
week)) / LAG(users_2) OVER (ORDER
BY week))*100 AS percentage_growth
FROM
totalUsers;
```

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

week	users_2	percentage_growth
0	424	NULL
1	624	47.1698
2	628	0.6410
3	596	-5.0955
4	640	7.3826
5	724	13.1250
6	692	-4.4199
7	668	-3.4682
8	652	-2.3952
9	704	7.9755
10	744	5.6818
11	644	-13.4409
12	724	12.4224
13	824	13.8122
14	788	-4.3689

Result 4

Result Grid | Filter Rows: | Export: | Wrap Ce

week	users_2	percentage_growth
15	828	5.0761
16	900	8.6957
17	876	-2.6667
18	828	-5.4795
19	968	16.9082
20	860	-11.1570
21	928	7.9070
22	1000	7.7586
23	984	-1.6000
24	1096	11.3821
25	1056	-3.6496
26	1028	-2.6515
27	1096	6.6148
28	1148	4.7445
29	1152	0.3484

Result 4

C) Weekly Retention Analysis

Task: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

```sql
SELECT users.user_id, Date(activated_at) as Date, week(activated_at) as week, COUNT(distinct users.user_id) as num_users
FROM users JOIN events ON events.user_id=users.user_id
WHERE event_type='signup_flow'
GROUP BY users.user_id,activated_at,week;
```

Result Grid | Filter Rows: | Export:

user_id	Date	week	users
11768	NULL	NULL	1
11770	NULL	NULL	1
11775	NULL	NULL	1
11778	NULL	NULL	1
11779	NULL	NULL	1
11780	NULL	NULL	1
11785	NULL	NULL	1
11787	NULL	NULL	1
11791	NULL	NULL	1
	NULL	NULL	

Result 3

D) Weekly Engagement Per Device

Task: Write an SQL query to calculate the weekly engagement per device.

```sql
#D) weekly engagement per device

SELECT COUNT(DISTINCT user_id) as total_users,
       COUNT(*) as engagements,
       WEEK(occurred_at) as week_no,
       device
FROM events
WHERE event_type = "engagement"
GROUP BY week_no, device;
```

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

total_users	engagements	week_no	device
9	67	17	acer aspire desktop
20	206	17	acer aspire notebook
4	83	17	amazon fire phone
21	251	17	asus chromebook
18	187	17	dell inspiron desktop
46	503	17	dell inspiron notebook
14	132	17	hp pavilion desktop
16	190	17	htc one
27	330	17	ipad air
19	205	17	ipad mini
21	217	17	iphone 4s
65	706	17	iphone 5
42	473	17	iphone 5s
6	57	17	kindle fire
86	793	17	lenovo thinkpad
6	59	17	mac mini

Result 1

E) Email Engagement Analysis

Task: Write an SQL query to calculate the email engagement metrics.

```sql
#E) email engagement analysis

SELECT week(occurred_at) as week_no,
count(distinct user_id) as userID,
SUM(case when action="email_open" then 1
else 0 end) as total_emails_opened,
SUM(case when action="email_clickthrough"
then 1 else 0 end) as total_emails_clicked,
SUM(case when action="sent_weekly_digest"
then 1 else 0 end) as
total_emails_sent,user_type
FROM emailevents
GROUP BY week_no, user_type;
```

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

week_no	userID	total_emails_opened	total_emails_clicked	total_emails_sent	user_type
17	329	115	57	309	1
17	266	81	46	243	2
17	386	114	63	356	3
18	935	323	148	906	1
18	720	247	119	687	2
18	1059	342	163	1009	3
19	956	331	161	913	1
19	741	279	130	708	2
19	1090	362	186	1044	3
20	960	331	164	928	1
20	764	265	128	728	2
20	1150	408	215	1077	3
21	982	341	146	951	1
21	781	287	130	752	2

RESULT

In this project, I learned advance SQL queries on real-life like datasets. This project yielded valuable insights into operational metrics, including job review activity, throughput trends, language preferences, and data quality issues. Moreover, the important task was 'importing the large data sets' into MYSQL workbench using both 'Table import wizard' and 'Load and Infile'command helped me to understand how to import large datasets within seconds into database. Additionally, we identified and addressed duplicate rows in the dataset.

By analyzing user engagement on a weekly basis, understanding user growth trends, and examining retention rates over time, I've identified patterns and opportunities for improvement in product usage and user retention strategies.

Additionally, analyzing engagement per device and email engagement metrics has provided insights into device preferences, communication effectiveness, and opportunities for optimizing engagement strategies.

Overall, I realized the complexity of real-world problems and how data analysis is applied to solve complicated problems.

Thank You

PROJECT MADE BY: GITANJALI PEKAMWAR