OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

TRAINITY PROJECT #3

Submission by :

ANANAYA LAL (Data Analyst Trainee)

PROJECT DESCRIPTION :

This analysis process involves analyzing a company's end-to-end operations. This analysis helps identify areas for improvement within the company. Data Analyst works closely with various teams, such as operations, support, and marketing, helping them derive valuable insights from the data they collect.

Our goal is to use 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.

This project involves conducting a comprehensive analysis of the end-to-end operations of a company based on two case studies. There were 2 case studies given to work upon. A brief description of both case studies and output required form both of them are given further.

Case Studies

Case Study 1: Job Data Analysis

Table name - job_data

Columns:

- **job_id:** Unique identifier of jobs
- **actor_id:** Unique identifier of actor
- **event:** The type of event (decision/skip/transfer).
- **language:** The Language of the content
- **time_spent:** Time spent to review the job in seconds.
- **org:** The Organization of the actor
- **ds:** The date in the format yyyy/mm/dd (stored as text).

Case Study 2: Investigating Metric Spike

Table names -

- **users**: Contains one row per user, with descriptive information about that user's account.
- **events**: Contains one row per event, where an event is an action that a user has taken (e.g., login, messaging, search).
- **email_events**: Contains events specific to the sending of emails.

Tasks

Case Study 1

- **Jobs Reviewed Over Time**
- **Throughput Analysis**
- **Language Share Analysis**
- **Duplicate Rows Detection**

Case Study 2

- **Weekly User Engagement**
- **User Growth Analysis**
- **Weekly Retention Analysis**
- **Weekly Engagement Per Device**
- **Email Engagement Analysis**

Important Informations

Approach :

- To conduct a comprehensive analysis of a company's end-to-end operations based on case studies, we would typically follow a structured approach that involves several key steps like , define the objective , collect data , analyse data , identify areas of improvement etc .

Tech-Stack Used :

- MySQL
- PostgreSQL
- MS Excel
- MS Word
- Powerpoint Ppt

C1T1 . Amount of jobs reviewed over time.

```
use opr_n_ma ;
SELECT ((COUNT(job_id))/sum(time_spent)/3600) as job_reviewed
 FROM job_data
```

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

job_reviewed
0.00000746

C1T2 . No. of events happening per second – to calculate 7 day rolling average.

```sql
select ds , job_nums, total_time,
sum(job_nums) over (order by ds rows between 6 preceding and current row)/sum(total_time) as 7day_rolling_average
from
(
select ds ,count(job_id) as job_nums,sum(time_spent) as total_time
from job_data where ds >='2020-11-01'  and ds <='2020-11-30'
group by ds
) a
group by
ds
```

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

ds	job_nums	total_time	7day_rolling_average
2020-11-25	1	45	0.0222
2020-11-26	1	56	0.0357
2020-11-27	1	104	0.0288
2020-11-28	2	33	0.1515
2020-11-29	1	20	0.3000
2020-11-30	2	40	0.2000

C1T3 . Share of each language for different contents.

```sql
-- select * from job_data
select language, lang_count,
sum(lang_count)/(sum(lang_count)over(order by language rows between unbounded preceding and unbounded following )) * 100.0 as perc_language
from
(
select language, count(language) as lang_count
from job_data
group by language
) a
```

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

language	lang_count	perc_language
Arabic	1	12.50000
English	1	12.50000
French	1	12.50000
Hindi	1	12.50000
Italian	1	12.50000
Persian	3	37.50000

C1T4 . Rows that have same values present in them.

```sql
select job_id, count(job_id) as duplicate_count
from job_data
group by job_id
having duplicate_count > 1
```

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

job_id	duplicate_count
23	3

C2T1 . To measure the activeness of a user.

```sql
use opr_n_ma ;

SELECT
    extract(year from occurred_at) as year,
    extract(week from occurred_at) as weeknum,
    count(distinct user_id) as user_enagagement
FROM
    events
GROUP BY
    year, weeknum
order by
    year, weeknum
```

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

year	weeknum	user_enagagement
2014	17	663
2014	18	1068
2014	19	1113
2014	20	1154
2014	21	1121

Result 2

C2T2 . Amount of users growing over time for a product – number of active users per week.

SQL File 4* | events | events

Don't Limit

```sql
use opr_n_ma ;

select
    year,
    weeknum,
    new_active_user,
    sum(new_active_user)over(order by year,weeknum rows between unbounded preceding and current row) as cum_active_user
FROM
(
SELECT
    extract(year from activated_at) as year,
    extract(week from activated_at) as weeknum,
    count(distinct user_id) as new_active_user
 FROM
    users
where
    state = 'active'
 group by
    year,
    weeknum
) a
```

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

year	weeknum	new_active_user	cum_active_user
2013	0	23	23
2013	1	30	53
2013	2	48	101
2013	3	36	137
2013	4	30	167

Result 47

Result Grid

Read Only

C2T3 . Users getting retained weekly after signing up for a product.

```sql
SELECT user_id, count(user_id),
sum(case when retention_week = 1 then 1 else 0 end) as week_1,
sum(case when retention_week = 2 then 1 else 0 end) as week_2,
sum(case when retention_week = 3 then 1 else 0 end) as week_3,
sum(case when retention_week = 4 then 1 else 0 end) as week_4,
sum(case when retention_week = 5 then 1 else 0 end) as week_5,
sum(case when retention_week = 6 then 1 else 0 end) as week_6,
sum(case when retention_week = 7 then 1 else 0 end) as week_7,
sum(case when retention_week = 8 then 1 else 0 end) as week_8,
sum(case when retention_week = 9 then 1 else 0 end) as week_9
from
(
SELECT
    a.user_id,
    a.signup_week,
    b.engagement_week,
    b.engagement_week - a.signup_week as retention_week
FROM
```

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

user_id	count(user_id)	week_1	week_2	week_3	week_4	week_5	week_6	week_7	week_8	week_9
11920	1	0	0	0	0	0	0	0	0	0
11924	1	0	0	0	0	0	0	0	0	0
11926	8	1	1	1	1	1	1	1	0	0
11928	8	0	0	0	0	0	0	0	0	1
11929	1	0	0	0	0	0	0	0	0	0
11931	6	1	1	1	1	0	0	0	0	0
11933	6	1	1	1	1	1	0	0	0	0
11936	3	0	0	1	0	0	0	0	0	0

Result 12

C2T4 . To measure the activeness of a user weekly - Calculate the weekly engagement per device.

SQL File 4* events SQL File 6*

Don't Limit

```sql
select
    extract(year from occurred_at) as year,
    extract(week from occurred_at) as week,
    device,
    count(distinct user_id ) as user_count
FROM
    events
where
    event_type = 'engagement'
GROUP BY
    year,week,device
order by
    year,week,device
```

Result Grid Filter Rows: Export: Wrap Cell Content:

year	week	device	user_count
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

Result 2

C2T5 . Users engaging with the email service.

SQL File 4* events SQL File 6* SQL File 7*

Don't Limit

```sql
SELECT
100.0*SUM(case when email_cat = 'email_open' then 1 else 0 END)/SUM(case when email_cat = 'email_sent' then 1 else 0 END) as email_open_rate,
100.0*SUM(case when email_cat = 'email_clicked' then 1 else 0 END)/SUM(case when email_cat = 'email_sent' then 1 else 0 END) as email_clicked_rate
FROM
(
select *,
case
when action IN ('sent_weekly_digest','sent_reengagement_email')
    THEN 'email_sent'
when action IN ('email_open')
    THEN 'email_open'
when action IN ('email_clickthrough')
    THEN 'email_clicked'
END as email_cat
FROM email_events
) a
```

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

email_open_rate	email_clicked_rate
33.58339	14.78989

Insights

- 0.00000746 jobs were reviewed over time.
- Highest share of language is of Persian i.e. 37.5 % followed by Arabic, English, French, Hindi and Italian
- Job Id : 23 has 3 duplicate count.
- The highest user engagement was 1154 in week number 20 in 2014.
- The highest weekly engagement as per device has user count 21 by Asus Chromebook in week 17 in 2014.

RESULTS

Results obtained are displayed on each slide respectively.

The project focused on various aspects of advanced SQL and enhanced the skills on the same.