



OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

A large, abstract graphic element consisting of numerous thin, light blue lines that curve and overlap to create a sense of depth and motion, resembling waves or a network.

sakhipatel20@gmail.com

A smaller, abstract graphic element located at the bottom left, consisting of several thin, light teal lines that intersect and form a diagonal pattern, creating a grid-like effect.

PROJECT DESCRIPTION

The project aims to leverage advanced SQL skills to conduct operational analytics and investigate metric spikes within a company's data. As a Lead Data Analyst, the objective is to provide valuable insights derived from the analysis to improve the company's operations and understand sudden changes in key metrics.



APPROACH

Approach towards the project and explain how I executed the analysis.

1

Database Setup & Exploration

Create and import tables into the database, and explore their structures, column meanings, and interrelationships.

2

Analysis Execution

Utilize SQL queries to conduct the required analysis tasks as outlined in the case studies.

3

Insightful Recommendations

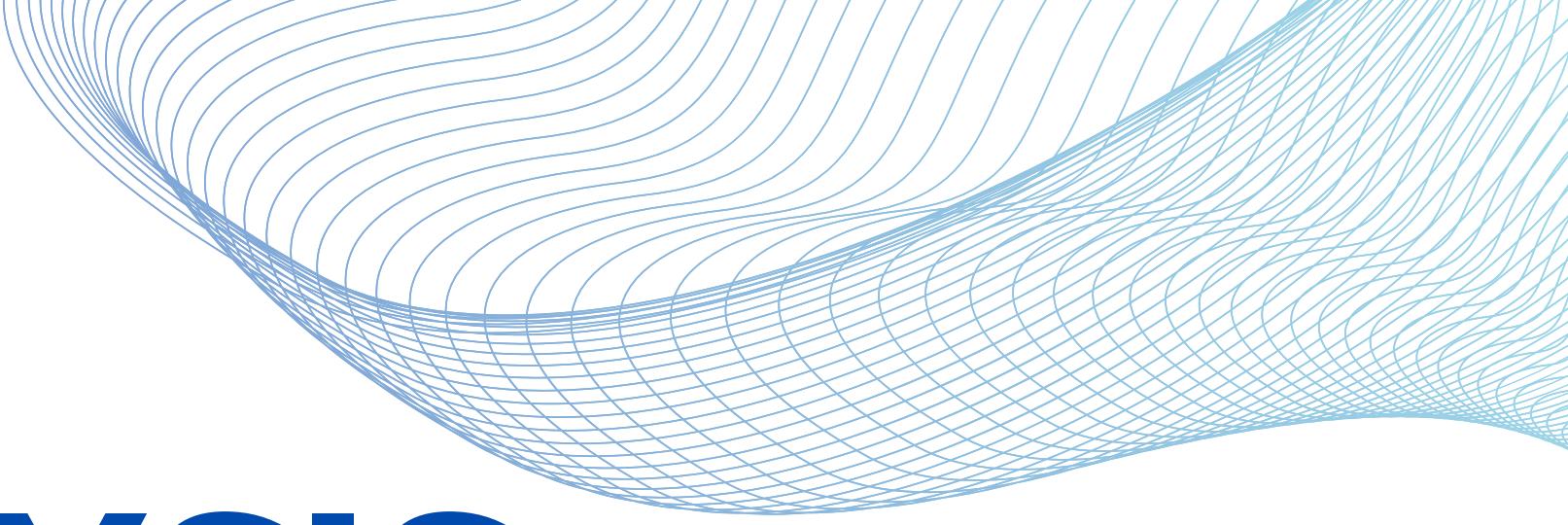
Interpret the analysis results to derive meaningful insights and provide actionable recommendations.

TECH-STACK USED

MySQL Workbench:

**Used for database creation,
table setup, and SQL query
execution.**





CASE STUDY 1: JOB DATA ANALYSIS

Tasks:

A. Jobs Reviewed Over Time: Calculate the number of jobs reviewed per hour for each day in November 2020.

B. Throughput Analysis: Calculate the 7-day rolling average of throughput (number of events per second).

C. Language Share Analysis: Calculate the percentage share of each language in the last 30 days.

D. Duplicate Rows Detection: Identify duplicate rows in the data.

A. JOBS REVIEWED OVER TIME

```
select count(distinct job_id)/(30*24) as number_of_jobs_reviewed_per_day  
from job_data;
```

number_of_jobs_reviewed_per_day
0.0083

B. THROUGHPUT ANALYSIS

```
select ds as date_of_review, jobs_reviewed, avg(jobs_reviewed)
over(order by ds rows between 6 preceding and current row) as throughput_7_rolling_average
from ( select ds, count( distinct job_id) as jobs_reviewed
from job_data group by ds order by ds) a;
```

Result Grid Filter Rows: Search Export:

	date_of_review	jobs_reviewed	throughput_7_rolling_average
	11/25/2020	1	1.0000
	11/26/2020	1	1.0000
	11/27/2020	1	1.0000
	11/28/2020	2	1.2500
	11/29/2020	1	1.2000
	11/30/2020	2	1.3333

C. LANGUAGE SHARE ANALYSIS

```
select job_data.job_id,  
       job_data.language,  
       count(distinct job_data.language) as total_of_each_language,  
       (count(job_data.language) / (select count(*) from job_data)) * 100 as  
       percentage_share_of_each_distinct_language  
  from job_data  
 group by job_data.job_id, job_data.language;
```

Result Grid Filter Rows: Search Export:

job_id	language	total_of_each_language	percentage_share_of_each_distinct_language
11	French	1	12.5000
20	Italian	1	12.5000
21	English	1	12.5000
22	Arabic	1	12.5000
23	Persian	1	37.5000
25	Hindi	1	12.5000

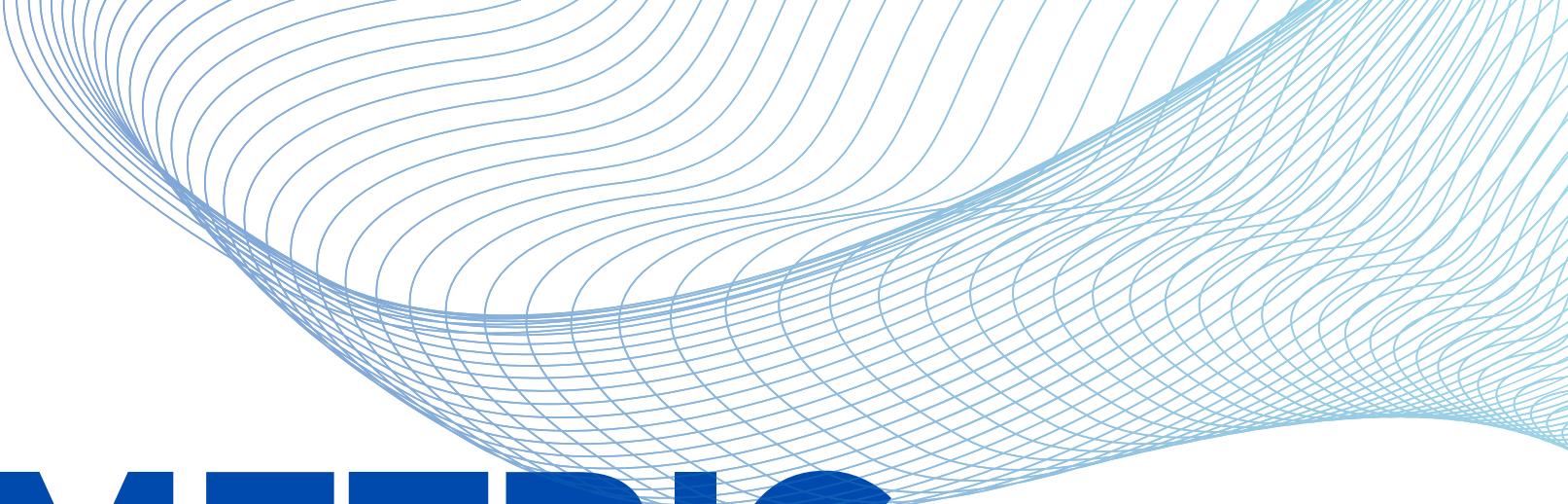
D. DUPLICATE ROWS DETECTION:

```
select * from  
(select *, row_number()over(partition by job_id) as row_num  
from job_data) a  
where row_num>1;
```

Result Grid Filter Rows: Search Export:

	ds	job_id	actor_id	event	language	time_spent	org	row_num
	11/28/2020	23	1005	transfer	Persian	22	D	2
	11/26/2020	23	1004	skip	Persian	56	A	3

CASE STUDY 2: INVESTIGATING METRIC SPIKE



- A. Weekly User Engagement:** Measure the activeness of users on a weekly basis.
- B. User Growth Analysis:** Analyze the growth of users over time for a product.
- C. Weekly Retention Analysis:** Analyze the retention of users on a weekly basis after signing up for a product.
- D. Weekly Engagement Per Device:** Measure the activeness of users on a weekly basis per device.
- E. Email Engagement Analysis:** Analyze how users are engaging with the email service.

A. WEEKLY USER ENGAGEMENT

```
select extract(week from str_to_date(occurred_at, '%d-%m-%Y %H:%i')) as num_week,  
count(distinct user_id) as no_of_distinct_user  
from tutorial.events  
group by extract(week from str_to_date(occurred_at, '%d-%m-%Y %H:%i'));
```

num_week	no_of_distinct_user
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

B. USER GROWTH ANALYSIS

```
select
    extract(week from str_to_date(occurred_at, '%d-%m-%Y %H:%i')) as num_week,
    count(distinct user_id) as no_of_distinct_user
from tutorial.events
group by extract(week from str_to_date(occurred_at, '%d-%m-%Y %H:%i'));
select
    year(str_to_date(substring_index(a.activated_at, '', 1), '%d-%m-%Y')) as year_num,
    week(str_to_date(substring_index(a.activated_at, '', 1), '%d-%m-%Y')) as week_num,
    count(distinct a.user_id) as num_active_users,
    @cum_active_users := @cum_active_users + count(distinct a.user_id) as cum_active_users
from tutorial.users a
cross join (select @cum_active_users := 0) r
where a.state = 'active'
group by year_num, week_num
order by year_num, week_num;
select count(*) from tutorial.users
where state = 'active';
```

B. USER GROWTH ANALYSIS

year_num	week_num	num_active_users	cum_active_users
2013	0	23	23
2013	1	30	53
2013	2	48	101
2013	3	36	137
2013	4	30	167
2013	5	48	215
2013	6	38	253
2013	7	42	295
2013	8	34	329
2013	9	43	372
2013	10	32	404
2013	11	31	435
2013	12	33	468
2013	13	39	507
2013	14	35	542
2013	15	43	585
2013	16	46	631
2013	17	49	680
2013	18	44	724
2013	19	57	781
2013	20	39	820
2013	21	49	869
2013	22	54	923
2013	23	50	973
2013	24	45	1018
2013	25	57	1075
2013	26	56	1131
2013	27	52	1183
2013	28	72	1255
2013	29	67	1322
2013	30	67	1389
2013	31	67	1456
2013	32	71	1527
2013	33	73	1600
2013	34	78	1678
2013	35	63	1741
2013	36	72	1813
2013	37	85	1898
2013	38	90	1988
2013	39	84	2072
2013	40	87	2159
2013	41	73	2232
2013	42	99	2331
2013	43	89	2420
2013	44	96	2516

year_num	week_num	num_active_users	cum_active_users
2013	44	96	2516
2013	45	91	2607
2013	46	88	2695
2013	47	102	2797
2013	48	97	2894
2013	49	116	3010
2013	50	124	3134
2013	51	102	3236
2013	52	47	3283
2014	0	83	3366
2014	1	126	3492
2014	2	109	3601
2014	3	113	3714
2014	4	130	3844
2014	5	133	3977
2014	6	135	4112
2014	7	125	4237
2014	8	129	4366
2014	9	133	4499
2014	10	154	4653
2014	11	130	4783
2014	12	148	4931
2014	13	167	5098
2014	14	162	5260
2014	15	164	5424
2014	16	179	5603
2014	17	170	5773
2014	18	163	5936
2014	19	185	6121
2014	20	176	6297
2014	21	183	6480
2014	22	196	6676
2014	23	196	6872
2014	24	229	7101
2014	25	207	7308
2014	26	201	7509
2014	27	222	7731
2014	28	215	7946
2014	29	221	8167
2014	30	238	8405
2014	31	193	8598
2014	32	245	8843
2014	33	261	9104
2014	34	259	9363
2014	35	18	9381

count(*)

9381

C. WEEKLY RETENTION ANALYSIS

```
select
    distinct d.user_id, count(d.user_id),
    sum(case when d.retention_week = 1 then 1 else 0 end) as per_week_retention
from (
    select
        a.user_id, extract(week from str_to_date(a.occurred_at, '%d-%m-%Y %H:%i')) as signup_week,
        b.engagement_week,
        b.engagement_week - extract(week from str_to_date(a.occurred_at, '%d-%m-%Y %H:%i')) as retention_week
    from ((select distinct user_id, occurred_at from tutorial.events
            where event_type = 'signup_flow' and event_name = 'complete_signup') a
        left join (select distinct user_id, extract(week from str_to_date(occurred_at, '%d-%m-%Y %H:%i')) as engagement_week
                  from tutorial.events
                  where event_type = 'engagement') b
        on a.user_id = b.user_id)) d
group by d.user_id
order by d.user_id
```

C. WEEKLY RETENTION ANALYSIS

Results in the following link for Google Drive:

<https://drive.google.com/file/d/1eTiElaPMfz3-9OK7yUZt5Z8t44kMLVtb/view?usp=sharing>

Preview:

user_id	count(d.user_id)	per_week_retention
11768	1	0
11770	1	0
11775	2	1
11778	3	0
11779	5	1
11780	2	1
11785	1	0
11787	3	1
11791	2	1
11793	6	1
11795	2	1
11798	6	1
11799	10	1

D. WEEKLY ENGAGEMENT PER DEVICE

```
select
    distinct d.user_id, count(d.user_id),
    sum(case when d.retention_week = 1 then 1 else 0 end) as per_week_retention
from (
    select
        a.user_id, extract(week from str_to_date(a.occurred_at, '%d-%m-%Y %H:%i')) as signup_week,
        b.engagement_week,
        b.engagement_week - extract(week from str_to_date(a.occurred_at, '%d-%m-%Y %H:%i')) as retention_week
    from ((select distinct user_id, occurred_at from tutorial.events
            where event_type = 'signup_flow' and event_name = 'complete_signup') a
            left join (select distinct user_id, extract(week from str_to_date(occurred_at, '%d-%m-%Y %H:%i')) as engagement_week
            from tutorial.events
            where event_type = 'engagement') b
            on a.user_id = b.user_id)) d
group by d.user_id
order by d.user_id
```

D. WEEKLY ENGAGEMENT PER DEVICE

Results in the following link for Google Drive:

https://drive.google.com/file/d/1_oem1BOWNUxrcRE54r3psAhagY6XAiev/view?usp=sharing

Preview:

year_num	week_num	device	no_of_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

E. EMAIL ENGAGEMENT ANALYSIS

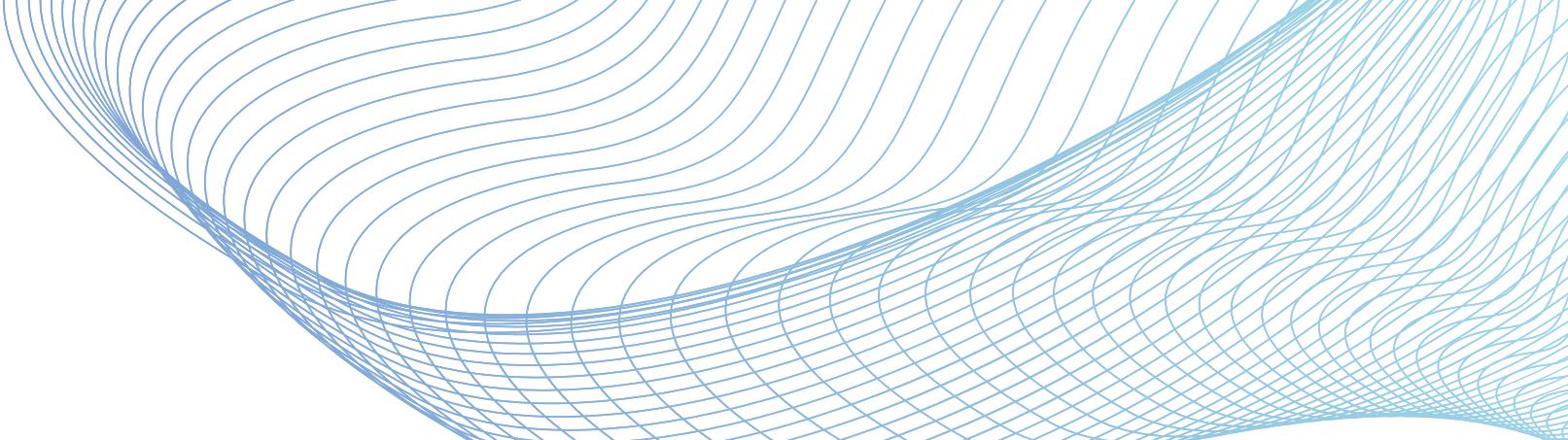
```
select
    100.0*sum(case when email_cat = 'email_opened' then 1 else 0 end)/sum(case when email_cat = 'email_sent'
then 1 else 0 end) as email_opening_rate,
    100.0*sum(case when email_cat = 'email_clicked' then 1 else 0 end)/sum(case when email_cat = 'email_sent'
then 1 else 0 end) as email_clicking_rate
from (
select *,
case
    when action in ('sent_weekly_digest','sent_reengagement_email')
        then 'email_sent'
    when action in ('email_open')
        then 'email_opened'
    when action in ('email_clickthrough')
        then 'email_clicked'
    end as email_cat
from tutorial.email_events
) a;
```

E. EMAIL ENGAGEMENT ANALYSIS

Results:

email_opening_rate	email_clicking_rate
33.58339	14.78989

INSIGHTS:

- 
- 1. Jobs Reviewed Over Time:** Monitoring the number of jobs reviewed per hour can help identify peak times and potential bottlenecks in the review process.
 - 2. Throughput Analysis:** The 7-day moving average offers a smoother long-term trend, while the daily metric is more volatile but useful for short-term monitoring.
 - 3. Language Share Analysis:** Understanding the percentage share of each language can inform localisation efforts and content strategy.
 - 4. User Engagement and Growth Analysis:** Analyzing user engagement and growth trends over time provides insights into the product's performance and user adoption.
 - 5. Retention Analysis:** Assessing weekly retention rates helps evaluate user onboarding and product stickiness.
 - 6. Email Engagement Analysis:** Understanding how users engage with email communications can guide improvements in email marketing strategies.

RESULTS

Through this project, valuable insights have been derived from the analysis of operational data. By understanding the patterns in job reviews, user engagement, and other metrics, actionable recommendations can be made to optimize operations, enhance user experience, and drive business growth. The project contributes to a deeper understanding of the company's operations and facilitates data-driven decision-making across departments.

sakhipatel20@gmail.com