

OPERATION AND METRIC ANALYTICS

- **Project Description :**

The project, aptly titled "Operational Analytics and Investigating Metric Spikes," embarks on a multifaceted exploration of our data landscape, comprising two interconnected case studies. The initial case study, centered around Operational Analytics, involves a meticulous examination of the job_data table. Here, we dissect daily trends, throughput patterns, language utilization, and identify duplicate entries. Simultaneously, the second case study, "Investigating Metric Spikes," delves into user, event, and email data, decoding weekly user engagement, growth patterns, retention rates, device-specific metrics, and email interactions.

These analyses aim to empower cross-functional teams with actionable insights, fostering a culture of data-driven decision-making. By comprehending operational dynamics and user behavior, we lay the groundwork for strategic optimizations that enhance overall efficiency. This project marks a pivotal step toward harnessing the power of data to uncover operational intricacies and decode user interactions, strategically positioning our company in the competitive technological landscape.

- **Approach:**

- i. **Case Study 1:** Operational Analytics - Unveiling Operational Dynamics

The analytical journey commences with an in-depth exploration of the job_data table. Using SQL queries, we dissect daily trends, throughput patterns, language utilization, and meticulously address duplicate entries to ensure data integrity. Calculating job reviews per hour provides real-time operational insights, while the 7-day rolling average throughput offers a stable metric for strategic optimizations. Language share analysis guides content strategies, and our systematic approach to duplicate entry detection fortifies data reliability.

- ii. **Case Study 2:** Investigating Metric Spikes - Decoding User Behaviour

Here, user-centric methodology involves a thorough examination of user, event, and email data. Leveraging SQL queries, we quantify weekly user engagement, unveil user growth patterns, and conduct weekly retention analysis through cohort-based queries for a deeper understanding of user loyalty. Device-specific engagement metrics reveal nuanced user interactions, and email engagement analysis identifies critical user interactions. These outcomes form a strategic blueprint for refining user experiences and optimizing email engagement strategies, ensuring adaptability in a dynamic technological landscape.

- **Tech – stack used :**

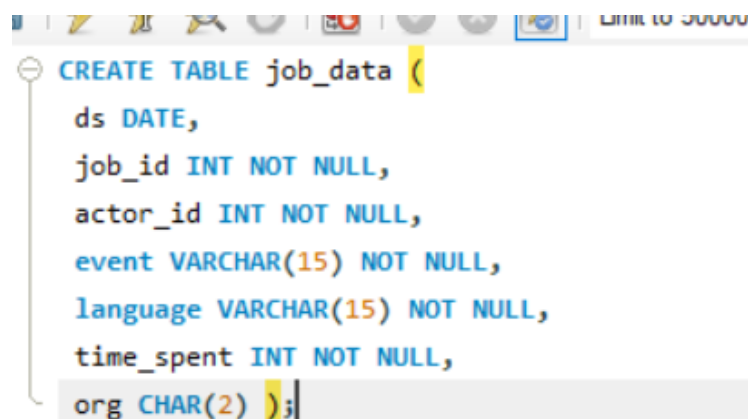
- i. MySQL WorkBench – User Optimised S/w for demonstrating SQL Queries.
- ii. SQL Server
- iii. SQL Shell

- **Case Study 1: Job Data Analysis**

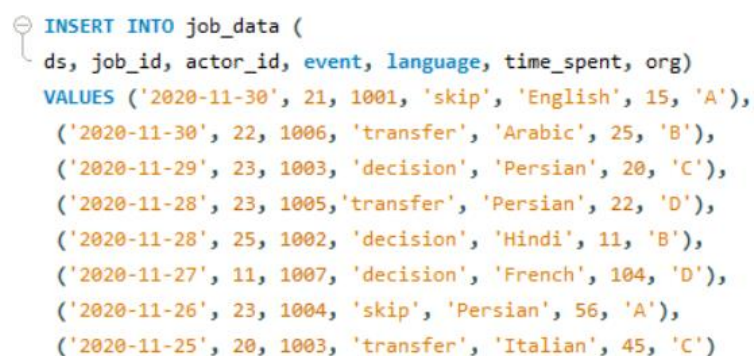
You will be working with a table named **job_data** with the following columns:

- **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).

- **Creating Tables :**



```
CREATE TABLE job_data (  
  ds DATE,  
  job_id INT NOT NULL,  
  actor_id INT NOT NULL,  
  event VARCHAR(15) NOT NULL,  
  language VARCHAR(15) NOT NULL,  
  time_spent INT NOT NULL,  
  org CHAR(2) );
```



```
INSERT INTO job_data (  
  ds, job_id, actor_id, event, language, time_spent, org)  
VALUES ('2020-11-30', 21, 1001, 'skip', 'English', 15, 'A'),  
('2020-11-30', 22, 1006, 'transfer', 'Arabic', 25, 'B'),  
('2020-11-29', 23, 1003, 'decision', 'Persian', 20, 'C'),  
('2020-11-28', 23, 1005, 'transfer', 'Persian', 22, 'D'),  
('2020-11-28', 25, 1002, 'decision', 'Hindi', 11, 'B'),  
('2020-11-27', 11, 1007, 'decision', 'French', 104, 'D'),  
('2020-11-26', 23, 1004, 'skip', 'Persian', 56, 'A'),  
('2020-11-25', 20, 1003, 'transfer', 'Italian', 45, 'C')
```

Tasks:

A. Jobs Reviewed Over Time:

- Objective: Calculate the number of jobs reviewed per hour for each day in November 2020.

```
1 • SELECT ds, COUNT(*)/24 AS no_of_job
2     FROM job_data
3     WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'
4     GROUP BY ds;
```

Result Grid			Filter Rows:
	ds	no_of_job	
▶	2020-11-30	0.0833	
	2020-11-29	0.0417	
	2020-11-28	0.0833	
	2020-11-27	0.0417	
	2020-11-26	0.0417	
	2020-11-25	0.0417	

B. Throughput Analysis:

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

```
1 • WITH DAILY_METRIC AS (
2     SELECT ds,
3           COUNT(job_id) AS job_review
4     FROM
5         job_data
6     GROUP BY
7         ds
8 )
9 SELECT ds, job_review,
10        AVG(job_review) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS throughput
11 FROM
12     DAILY_METRIC
13 ORDER BY
14     throughput DESC;
```

	ds	job_review	throughput
▶	2020-11-30	2	1.3333
	2020-11-28	2	1.2500
	2020-11-29	1	1.2000
	2020-11-25	1	1.0000
	2020-11-26	1	1.0000
	2020-11-27	1	1.0000

C. Language Share Analysis:

- Objective: Calculate the percentage share of each language in the last 30 days.

```
1 • SELECT
2     language,
3     COUNT(language) AS language_count,
4     (COUNT(language) / (SELECT COUNT(*) FROM job_data)) * 100 AS percentage_share
5 FROM job_data
6 GROUP BY language
7 ORDER BY language DESC;
8
```

	language	language_count	percentage_share
▶	Arabic	1	12.5000
	English	1	12.5000
	French	1	12.5000
	Hindi	1	12.5000
	Italian	1	12.5000
	Persian	3	37.5000

D. Duplicate Rows Detection:

- Objective: Identify duplicate rows in the data.

```
1 • SELECT * FROM (
2     SELECT *,
3         ROW_NUMBER() OVER (PARTITION BY JOB_ID) AS DUPLICATE_ROWS
4     FROM JOB_DATA
5 ) AS A_R
6 WHERE DUPLICATE_ROWS > 1;
```

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

• Case Study 2: Investigating Metric Spike

You will be working with three tables:

- **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.

• User Table:

```

1  create database pro3;
2  • use pro3;
3  • create table users (
4      user_id int,
5      created_at varchar(100),
6      company_id int,
7      language varchar(50),
8      activated_at varchar(100),
9      state varchar(50));
10 • show variables like 'secure_file_priv';
11 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.2/Uploads/users.csv"
12 INTO TABLE users
13 FIELDS TERMINATED BY ','
14 ENCLOSED BY '"'
15 LINES TERMINATED BY '\n'
16 IGNORE 1 ROWS;
17 • select * from users;
18 • alter table users add column temp_date_created datetime;
19 • update users set temp_date_created = str_to_date(created_at , '%d-%m-%y %h:%i');
20

```

	user_id	created_at	company_id	language	activated_at	state
▶	0	01-01-2013 20:59	5737	english	01-01-2013 21:01	active
	3	01-01-2013 18:40	2800	german	01-01-2013 18:42	active
	4	01-01-2013 14:37	5110	indian	01-01-2013 14:39	active
	6	01-01-2013 18:37	11699	english	01-01-2013 18:38	active
	7	01-01-2013 16:19	4765	french	01-01-2013 16:20	active
	8	01-01-2013 04:38	2698	french	01-01-2013 04:40	active
	11	01-01-2013 08:07	3745	english	01-01-2013 08:09	active
	13	02-01-2013 12:27	4025	english	02-01-2013 12:29	active
	15	02-01-2013 15:39	4259	english	02-01-2013 15:41	active
	17	02-01-2013 10:56	5025	japanese	02-01-2013 10:57	active
	19	02-01-2013 09:54	326	english	02-01-2013 09:55	active
	20	02-01-2013 09:41	7	italian	02-01-2013 09:43	active
	21	02-01-2013 09:29	2606	english	02-01-2013 09:30	active
	22	02-01-2013 17:36	545	german	02-01-2013 17:38	active
	27	03-01-2013 16:14	6	japanese	03-01-2013 16:15	active
	30	03-01-2013 08:28	4148	english	03-01-2013 08:29	active
	31	03-01-2013 15:45	39	arabic	03-01-2013 15:46	active
	33	03-01-2013 12:16	10768	english	03-01-2013 12:18	active
	35	03-01-2013 16:06	1891	english	03-01-2013 16:07	active
	36	03-01-2013 11:51	2	english	03-01-2013 11:53	active
	47	04-01-2013 10:39	1	indian	04-01-2013 10:41	active
	49	05-01-2013 14:33	8727	spanish	05-01-2013 14:34	active
	50	05-01-2013 17:41	12526	spanish	05-01-2013 17:43	active
	52	06-01-2013 13:35	6	english	06-01-2013 13:37	active

- Event Table:

```

create table events (
  user_id int null,
  occurred_at varchar(100) ,
  event_type varchar(50) ,
  event_name varchar(100) ,
  location varchar(50) ,
  device varchar(50) ,
  user_type int null );

select * from events;

LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.2/Uploads/events.csv"
INTO TABLE events
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'

```

	user_id	occured_at	event_type	event_name	location	device	user_ty
▶	10522	02-05-2014 11:02	engagement	login	Japan	dell inspiron notebook	3
	10522	02-05-2014 11:02	engagement	home_page	Japan	dell inspiron notebook	3
	10522	02-05-2014 11:03	engagement	like_message	Japan	dell inspiron notebook	3
	10522	02-05-2014 11:04	engagement	view_inbox	Japan	dell inspiron notebook	3
	10522	02-05-2014 11:03	engagement	search_run	Japan	dell inspiron notebook	3
	10522	02-05-2014 11:03	engagement	search_run	Japan	dell inspiron notebook	3
	10612	01-05-2014 09:59	engagement	login	Netherlands	iphone 5	1
	10612	01-05-2014 10:00	engagement	like_message	Netherlands	iphone 5	1
	10612	01-05-2014 10:00	engagement	send_message	Netherlands	iphone 5	1
	10612	01-05-2014 10:01	engagement	home_page	Netherlands	iphone 5	1
	10612	01-05-2014 10:01	engagement	like_message	Netherlands	iphone 5	1
	10612	01-05-2014 10:02	engagement	h home_page	Netherlands	iphone 5	1
	10612	01-05-2014 10:02	engagement	view_inbox	Netherlands	iphone 5	1
	10612	01-05-2014 10:03	engagement	like_message	Netherlands	iphone 5	1
	10612	01-05-2014 10:03	engagement	home_page	Netherlands	iphone 5	1
	10612	01-05-2014 10:04	engagement	send_message	Netherlands	iphone 5	1
	10612	01-05-2014 10:04	engagement	like_message	Netherlands	iphone 5	1
	10612	01-05-2014 10:05	engagement	send_message	Netherlands	iphone 5	1
	10736	09-05-2014 17:52	engagement	login	Austria	iphone 4s	2
	10736	09-05-2014 17:53	engagement	like_message	Austria	iphone 4s	2
	10736	09-05-2014 17:53	engagement	send_message	Austria	iphone 4s	2
	10965	15-05-2014 13:52	engagement	login	Finland	windows surface	3
	10965	15-05-2014 13:53	engagement	home_page	Finland	windows surface	3
	11020	08-05-2014 09:15	engagement	login	Japan	macbook air	2

- Email-Event Table:

```
create table emailEvent(
user_id int,
occured_at varchar(100),
action varchar(100),
user_type int );
```

```
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.2/Uploads/email_events.csv"
INTO TABLE emailEvent
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

user_id	occured_at	action	user_type
0	06-05-2014 09:30	sent_weekly_digest	1
0	13-05-2014 09:30	sent_weekly_digest	1
0	20-05-2014 09:30	sent_weekly_digest	1
0	27-05-2014 09:30	sent_weekly_digest	1
0	03-06-2014 09:30	sent_weekly_digest	1
0	03-06-2014 09:30	email_open	1
0	10-06-2014 09:30	sent_weekly_digest	1
0	10-06-2014 09:30	email_open	1
0	17-06-2014 09:30	sent_weekly_digest	1
0	17-06-2014 09:30	email_open	1
0	24-06-2014 09:30	sent_weekly_digest	1
0	01-07-2014 09:30	sent_weekly_digest	1
0	08-07-2014 09:30	sent_weekly_digest	1
0	15-07-2014 09:30	sent_weekly_digest	1
0	22-07-2014 09:30	sent_weekly_digest	1
0	29-07-2014 09:30	sent_weekly_digest	1
0	29-07-2014 09:30	email_open	1
0	05-08-2014 09:30	sent_weekly_digest	1
0	12-08-2014 09:30	sent_weekly_digest	1
0	19-08-2014 09:30	sent_weekly_digest	1
0	19-08-2014 09:30	email_open	1
0	26-08-2014 09:30	sent_weekly_digest	1

Tasks:

A. Weekly User Engagement:

- Objective: Measure the activeness of users on a weekly basis.

```
select extract(week from occurred_at) as week_number,  
       count(distinct user_id) as active_user  
from events_tbl  
where event_type='engagement'  
group by week_number  
order by week_number
```

week_num	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

B. User Growth Analysis:

- Objective: Analyze the growth of users over time for a product.

```
select year, week_num, num_users, sum(num_users)  
       over (order by year, week_num) as cum_users  
from (  
  select extract(year from created_at) as year, extract(week from created_at) as week_num, count(distinct user_id) as num_users  
  from users_tbl  
  where state='active'  
  group by year, week_num  
  order by year, week_num)sub
```

Result Grid Filter Rows:				
	year	week_num	num_users	cum_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

C. Weekly Retention Analysis:

- Objective: Analyze the retention of users on a weekly basis after signing up for a product.

```
with cte1 as (
  select distinct user_id,
  extract (week from occurred_at) as signup_week
  from events_tbl
  where event_type = 'signup_flow'
  and event_name = 'complete_signup and extract (week from occurred_at) = 18 ),
cte2 as (select distinct user_id,
  extract (week from occurred_at) as engagement_week
  from events_tbl
  where event_type = 'engagement')
select count(user_id) total_engaged_users,
  sum(case when retention_weeks < 5 then 1 else end) as retained_users
  from (select a.user_id, a.signup_week,
  b.engagement_week, b.engagement_week - a.signup_week as retention_week
  from cte1 a
  left join cte2 b
  on a.user_id = b.user_id) sub
order by a.user_id) sub
```

total_engaged_users	retained_users
317	236

D. Weekly Engagement Per Device:

- Objective: Measure the activeness of users on a weekly basis per device.

```
with cte as (select extract (year from occurred_at)||'-'||extract(week from occurred_at) as weeknum
  device, count(distinct user_id) as usercnt
  from events_tbl
  where event_type = 'engagement'
  group by weeknum, device
  order by weeknum)
select weeknum, device, usercnt
from cte
```

weeknum	device	usercnt
2014-18	acer aspire desktop	10
2014-18	acer aspire notebook	21
2014-18	amazon fire phone	4
2014-18	asus chromebook	23
2014-18	dell inspiron desktop	21

E. Email Engagement Analysis:

- Objective: Analyze how users are engaging with the email service.

```
select
100 * 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 * 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_click_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) sub
```

	email_open_rate	email_click_rate
▶	31.1921	10.4745

• Insights:

Case Study 1: Operational Analytics - Unveiling Operational Dynamics

1. Jobs Reviewed Over Time: Extracting real-time operational trends is achieved through the calculation of job reviews per hour from job_data.
2. Throughput Analysis: Strategic guidance is provided by a stabilized 7-day rolling average throughput, effectively smoothing variations in daily operational metrics.
3. Language Share Analysis: Content strategies benefit from insights gained through language share analysis over the last 30 days, ensuring targeted and effective communication.
4. Duplicate Rows Detection: Enhancing data integrity is prioritized by identifying and addressing duplicate entries, ensuring the reliability of operational insights.

Case Study 2: Investigating Metric Spikes - Decoding User Behavior

1. Weekly User Engagement: Insightful metrics for user interactions are derived through SQL queries, quantifying and understanding weekly user engagement.
2. User Growth Analysis: Understanding the chronological narrative of user acquisition is facilitated by parsing sign-up dates, providing insights into growth patterns over time.
3. Weekly Retention Analysis: Critical insights into user loyalty are unveiled through cohort-based analysis linking sign-up cohorts to ongoing engagement.
4. Weekly Engagement Per Device: Device-specific engagement metrics offer nuanced understanding of user interactions, guiding strategies tailored to different devices.
5. Email Engagement Analysis: Shaping strategic blueprints for refining user experiences and optimizing email engagement is achieved through analyzing crucial user interactions with email services.

• Result :

This project has been a valuable learning experience, providing practical insights into our operations and influencing more informed decision-making. In the Operational Analytics segment, we delved into job_data, gaining real-time insights into job trends and language use. Utilizing a 7-day average enhanced metric stability, while addressing duplicate data bolstered the reliability of our information. These measures streamlined decision-making, making it more agile and strategic.

In the Investigating Metric Spikes phase, a user-focused approach offered nuanced insights into user behavior. Assessing weekly engagement, user growth, and retention illuminated areas of strength. Exploring different devices and email interactions provided actionable insights. These learnings are actively shaping decisions, contributing to product enhancements and user satisfaction. Overall, this project has cultivated a data-driven mindset, empowering us to navigate the evolving tech landscape with clarity and informed decision-making.