

# OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

-MRUNALI PETAKAR

---

## *PROJECT DESCRIPTION*

---

As the Lead Data Analyst in this operational analytics project at Microsoft, I will collaborate with various departments to analyze data, identify areas for improvement, and understand sudden changes in key metrics. By leveraging advanced SQL skills, statistical analysis, and data visualization techniques, I will uncover insights that can drive operational enhancements. Through a comprehensive approach involving data collection, analysis, hypothesis testing, and predictive modeling, I will provide actionable recommendations to optimize business processes, foster informed decision-making, and contribute to the company's overall success.

---

## *Approach*

---

To handle the tasks and derive insights, I will use SQL queries to retrieve relevant information from the provided database.

---

## *Tech-stack used*

---

I am using MySQL Workbench for this project. Choosing MySQL Workbench for this project aligns well with database requirements,

my expertise, and the need for efficient data analysis, making it a reliable and suitable tool for extracting valuable insights from the Instagram user database.

---

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

---

In case study 1 the insights are found based on following questions:

#### **1] Number of jobs reviewed: jobs reviewed over time(in hour).**

*Query-*

```
select
ds,
((count(job_id)*3600)/sum(time_spent)) as
jobs_reviewed_per_hr
from
job_data
where
month(ds)=11
group by ds;
```

*output-*

Result Grid			Filter Rows:
	ds	jobs_reviewed_per_hr	
▶	2020-11-30	180.0000	
	2020-11-29	180.0000	
	2020-11-28	218.1818	
	2020-11-27	34.6154	
	2020-11-26	64.2857	
	2020-11-25	80.0000	

above output shows no. of jobs reviewed per hour.

This task highlighted the importance of analyzing temporal trends to understand variations in user behavior, which can aid in resource allocation and optimization.

## 2] Throughput: Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

*Query-*

#For daily metric

```
select ds, round(count(j_event)/sum(time_spent),2) as "daily
metric"
from job_data
group by ds
order by ds;
```

*Output-*

```
44 • SELECT
45     ds,
46     ROUND(COUNT(j_event) / SUM(time_spent), 2) AS 'daily metric'
47 FROM
48     job_data
49 GROUP BY ds
50 ORDER BY ds;
```

ds	daily metric
2020-11-25	0.02
2020-11-26	0.02
2020-11-27	0.01
2020-11-28	0.06
2020-11-29	0.05
2020-11-30	0.05

Query –

# for weekly throughput

```
SELECT  
    ROUND(COUNT(j_event) / SUM(time_spent), 2) AS 'weekly  
throughput'  
FROM  
    job_data;
```

Output-



The screenshot shows a SQL query editor with the following code:

```
41 #weekly throughput  
42 • SELECT  
43     ROUND(COUNT(j_event) / SUM(time_spent), 2) AS 'weekly throughput'  
44 FROM  
45     job_data;  
46
```

Below the editor is a toolbar with options: Result Grid, Filter Rows, Export, and Wrap Cell Content. The Result Grid shows the output of the query:

	weekly throughput
▶	0.03

We prefer 7-day rolling average. Because, rolling averages are useful for finding long-term trends otherwise disguised by occasional fluctuations

### 3] Percentage share of each language: Share of each language for different contents.

Query-

```
select count(distinct j_language) from job_data as total;
```

#this will give output 6 which we will use for further calculations.

```

select j_language as Languages, round(100 *count(*)/6, 2) as
Percentage
from job_data
cross join (SELECT COUNT(*) FROM job_data) jobs
group by j_language;

```

*Output-*

	Languages	Percentage
►	English	12.50
	Arabic	12.50
	Persian	37.50
	Hindi	12.50
	French	12.50
	Italian	12.50

This task showcased how tracking language trends can provide insights into user preferences and potentially guide content creation efforts.

**4] Duplicate rows: Rows that have the same value present in them.**

*Query-*

```

select actor_id, count(*) as times
from job_data
group by actor_id
having count(*)>1;

```

*Output-*

```

15     # duplicates
16 •   select actor_id, count(*) as times
17     from job_data
18     group by actor_id
19     having count(*)>1;

```

Result Grid			Filter Rows:	Export:
	actor_id	times		
▶	1003	2		

Detecting and addressing duplicates is crucial for ensuring accurate insights and preventing skewed conclusions.

In case study 2 the insights are found based on following questions:

---

### *Case Study 2: Investigating metric spike*

---

Below are the queries for creating database and importing dataset which I used-

#creating database

drop database project3;

create database Project3;

use project3;

#table 1 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';
```

```
#files should be at path described in above querie's output
```

```
#importing dataset
```

```
load data infile"C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users.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;
```

```
update users set temp_created_at = STR_TO_DATE(created_at, '%d-%m-%Y  
%H:%i'); # after doing sql_safe_updates=0
```

```
alter table users drop column created_at;
```

```
alter table users change column temp_created_at 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';
```

```
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;
```

```
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'); # after doing sql_safe_updates=0  
alter table events drop column occurred_at;  
alter table events change column temp_occurred_at occurred_at datetime;
```

#table-3

```
create table EmailEvents(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 EmailEvents
```



fields terminated by','  
enclosed by ''''  
lines terminated by '\n'  
ignore 1 rows;

```
alter table EmailEvents add column temp_occurred_at datetime;  
SET SQL_SAFE_UPDATES = 0;  
update EmailEvents set temp_occurred_at = STR_TO_DATE(occurred_at, '%d-  
%m-%Y %H:%i'); # after doing sql_safe_updates=0  
alter table EmailEvents drop column occurred_at;  
alter table EmailEvents change column temp_occurred_at occurred_at  
datetime;
```

**1] User Engagement: To measure the activeness of a user.  
Measuring if the user finds quality in a product/service.**

Write an SQL query to calculate weekly user engagement

*Query-*

```
SELECT  
    EXTRACT(WEEK FROM occurred_at) AS 'week',  
    COUNT(DISTINCT user_id) AS 'active users'  
FROM  
    events  
WHERE  
    event_type = 'engagement'  
GROUP BY week;
```

*Output-*

	week	active 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
	33	1225
	34	1204
	35	104

Tracking user engagement over time provides insights into user satisfaction, feature adoption, and potential points of improvement.

## 2]User Growth: Amount of users growing over time for a product.

*Query used-*

```
select months, Users, round((((Users/LAG(Users,1) over (order
by months) - 1) *100),2)as "growth in %" from (
select extract(MONTH FROM created_at) as months,
count(activated_at)as Users
from users
where activated_at not in("")
group by 1
order by 1
)growth;
```

*Output-*

	months	Users	growth in %
▶	1	712	NULL
	2	685	-3.79
	3	765	11.68
	4	907	18.56
	5	993	9.48
	6	1086	9.37
	7	1281	17.96
	8	1347	5.15
	9	330	-75.50
	10	390	18.18
	11	399	2.31
	12	486	21.80

I recognized the value of analyzing user growth trends over time to gauge the success of product adoption.

Understanding user growth helps assess marketing strategies, product improvements, and overall company growth potential.

### **3] Weekly Retention: Users getting retained weekly after signing-up for a product.**

*Query-*

SELECT first AS "Week Numbers",

SUM(CASE WHEN week\_number=0 THEN 1 ELSE 0 END) AS "Week 0",

SUM(CASE WHEN week\_number=1 THEN 1 ELSE 0 END) AS "Week 1",

SUM(CASE WHEN week\_number=2 THEN 1 ELSE 0 END) AS "Week 2",

SUM(CASE WHEN week\_number=3 THEN 1 ELSE 0 END) AS "Week 3",

SUM(CASE WHEN week\_number=4 THEN 1 ELSE 0 END) AS "Week 4",

SUM(CASE WHEN week\_number=5 THEN 1 ELSE 0 END) AS "Week 5",

SUM(CASE WHEN week\_number=6 THEN 1 ELSE 0 END) AS "Week 6",

SUM(CASE WHEN week\_number=7 THEN 1 ELSE 0 END) AS "Week 7",

SUM(CASE WHEN week\_number=8 THEN 1 ELSE 0 END) AS "Week 8",

SUM(CASE WHEN week\_number=9 THEN 1 ELSE 0 END) AS "Week 9",

SUM(CASE WHEN week\_number=10 THEN 1 ELSE 0 END) AS "Week 10",

SUM(CASE WHEN week\_number=11 THEN 1 ELSE 0 END) AS "Week 11",

SUM(CASE WHEN week\_number=12 THEN 1 ELSE 0 END) AS "Week 12",

SUM(CASE WHEN week\_number=13 THEN 1 ELSE 0 END) AS "Week 13",

SUM(CASE WHEN week\_number=14 THEN 1 ELSE 0 END) AS "Week 14",

SUM(CASE WHEN week\_number=15 THEN 1 ELSE 0 END) AS "Week 15",

SUM(CASE WHEN week\_number=16 THEN 1 ELSE 0 END) AS "Week 16",

SUM(CASE WHEN week\_number=17 THEN 1 ELSE 0 END) AS "Week 17",

SUM(CASE WHEN week\_number=18 THEN 1 ELSE 0 END) AS "Week 18"

FROM

(

SELECT m.user\_id, m.login\_week, n.first, m.login\_week-first AS week\_number

FROM

(SELECT user\_id, EXTRACT( WEEK FROM occurred\_at) AS login\_week  
FROM events

GROUP BY 1,2)m,

(SELECT user\_id, MIN(EXTRACT(WEEK FROM occurred\_at)) AS first  
FROM events

GROUP BY 1)n

WHERE m.user\_id=n.user\_id)

sub

GROUP BY first

ORDER BY first;

## Output-

	Week Numbers	Week 0	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18
17		663	472	324	251	205	187	167	146	145	145	136	131	132	143	116	91	82	77	5
18		596	362	261	203	168	147	144	127	113	122	106	118	127	110	97	85	67	4	0
19		427	284	173	153	114	95	91	81	95	82	68	65	63	42	51	49	2	0	0
20		358	223	165	121	91	72	63	67	63	65	67	41	40	33	40	0	0	0	0
21		317	187	131	91	74	63	75	72	58	48	45	39	35	28	2	0	0	0	0
22		326	224	150	107	87	73	63	60	55	48	41	39	31	1	0	0	0	0	0
23		328	219	138	101	90	79	69	61	54	47	35	30	0	0	0	0	0	0	0
24		339	205	143	102	81	63	65	61	38	39	29	0	0	0	0	0	0	0	0
25		305	218	139	101	75	63	50	46	38	35	2	0	0	0	0	0	0	0	0
26		288	181	114	83	73	55	47	43	29	0	0	0	0	0	0	0	0	0	0
27		292	199	121	106	68	53	40	36	1	0	0	0	0	0	0	0	0	0	0
28		274	194	114	69	46	30	28	3	0	0	0	0	0	0	0	0	0	0	0
29		270	186	102	65	47	40	1	0	0	0	0	0	0	0	0	0	0	0	0
30		294	202	121	78	53	3	0	0	0	0	0	0	0	0	0	0	0	0	0
31		215	145	76	57	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32		267	188	94	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33		286	202	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34		279	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35		18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I learned how cohort-based analysis can provide insights into user retention patterns after sign-up.

Examining retention helps identify engagement bottlenecks and supports efforts to enhance user satisfaction and loyalty.

**4]Weekly Engagement: To measure the activeness of a user.  
Measuring if the user finds quality in a product/service weekly.**

*Query used-*

```
SELECT EXTRACT(WEEK FROM occurred_at) AS "Week Numbers",  
COUNT(DISTINCT CASE WHEN device IN('dell inspiron notebook')  
THEN user_id ELSE  
NULL END) AS "Dell Inspiron Notebook",  
COUNT(DISTINCT CASE WHEN device IN('iphone 5') THEN user_id  
ELSE NULL END) AS  
"iPhone 5",  
COUNT(DISTINCT CASE WHEN device IN('iphone 4s') THEN user_id  
ELSE NULL END) AS
```

```
"iPhone 4S",  
COUNT(DISTINCT CASE WHEN device IN('windows surface') THEN  
user_id ELSE NULL  
END) AS "Windows Surface",  
COUNT(DISTINCT CASE WHEN device IN('macbook air') THEN user_id  
ELSE NULL END)  
AS "Macbook Air",  
COUNT(DISTINCT CASE WHEN device IN('iphone 5s') THEN user_id  
ELSE NULL END) AS  
"iPhone 5S",  
COUNT(DISTINCT CASE WHEN device IN('macbook pro') THEN  
user_id ELSE NULL END)  
AS "Macbook Pro",  
COUNT(DISTINCT CASE WHEN device IN('kindle fire') THEN user_id  
ELSE NULL END)  
AS "Kindle Fire",  
COUNT(DISTINCT CASE WHEN device IN('ipad mini') THEN user_id  
ELSE NULL END) AS  
"iPad Mini",  
COUNT(DISTINCT CASE WHEN device IN('nexus 7') THEN user_id ELSE  
NULL END) AS  
"Nexus 7",  
COUNT(DISTINCT CASE WHEN device IN('nexus 5') THEN user_id ELSE  
NULL END) AS  
"Nexus 5",  
COUNT(DISTINCT CASE WHEN device IN('samsung galaxy s4') THEN  
user_id ELSE NULL  
END) AS "Samsung Galaxy S4",
```

```
COUNT(DISTINCT CASE WHEN device IN('lenovo thinkpad') THEN
user_id ELSE NULL
END) AS "Lenovo Thinkpad",
COUNT(DISTINCT CASE WHEN device IN('samsung galaxy tablet')
THEN user_id ELSE
NULL END) AS "Samsung Galaxy Tablet",
COUNT(DISTINCT CASE WHEN device IN('acer aspire notebook')
THEN user_id ELSE
NULL END) AS "Acer Aspire Notebook", COUNT(DISTINCT CASE
WHEN device IN('asus chromebook') THEN user_id ELSE NULL
END) AS "Asus Chromebook",
COUNT(DISTINCT CASE WHEN device IN('htc one') THEN user_id ELSE
NULL END) AS
"HTC One",
COUNT(DISTINCT CASE WHEN device IN('nokia lumia 635') THEN
user_id ELSE NULL END) AS "Nokia Lumia 635", COUNT(DISTINCT
CASE WHEN device IN('samsung galaxy note') THEN user_id ELSE
NULL END) AS "Samsung Galaxy Note",
COUNT(DISTINCT CASE WHEN device IN('acer aspire desktop') THEN
user_id ELSE NULL END) AS "Acer Aspire Desktop",
COUNT(DISTINCT CASE WHEN device IN('mac mini') THEN user_id
ELSE NULL END) AS
"Mac Mini",
COUNT(DISTINCT CASE WHEN device IN('hp pavilion desktop') THEN
user_id ELSE NULL END) AS "HP Pavilion Desktop",

COUNT(DISTINCT CASE WHEN device IN('dell inspiron desktop') THEN
user_id ELSE
```



```

NULL END) AS "Dell Inspiron Desktop",

COUNT(DISTINCT CASE WHEN device IN('ipad air') THEN user_id ELSE
NULL END) AS "iPad Air",

COUNT(DISTINCT CASE WHEN device IN('amazon fire phone') THEN
user_id ELSE NULL END) AS "Amazon Fire Phone", COUNT(DISTINCT
CASE WHEN device IN('nexus 10') THEN user_id ELSE NULL END) AS

"Nexus 10"

FROM events

WHERE event_type = "engagement"

GROUP BY 1

ORDER BY 1;

```

## Output-

Week	Nur	Dell Inspi	iPhone 5	iPhone 4S	Windows	Macbook	iPhone 5S	Macbook	Kindle Fire	iPad Mini	Nexus 7	Nexus 5	Samsung	Lenovo Th	Samsung	Acer Aspir	Asus Chroi	HTC One	Nokia Lum	Samsung	Acer Aspir	Mac Mini	HP Pavilio	Dell Inspi	iPad Air	Amazon F	Nexus 10
17	46	65	21	10	54	42	143	6	19	18	40	52	86	0	20	21	16	17	7	9	6	14	18	27	4	16	
18	77	113	46	10	121	73	252	27	30	30	73	82	153	0	33	42	19	33	15	26	13	37	58	52	9	30	
19	83	115	44	16	112	79	266	21	36	41	87	91	178	0	41	27	30	23	11	23	18	40	36	55	12	25	
20	84	125	55	21	119	79	256	23	32	32	103	93	173	0	40	41	29	22	18	23	26	30	52	59	11	22	
21	80	137	45	17	110	74	247	30	23	29	91	84	167	0	47	38	21	25	20	29	18	44	41	51	5	25	
22	92	125	45	15	145	71	251	21	34	45	96	105	176	0	41	52	24	25	19	25	25	38	52	58	5	27	
23	103	152	53	14	124	79	266	25	33	36	88	99	176	0	43	49	20	31	14	22	18	54	53	41	16	45	
24	99	142	53	22	152	79	255	25	39	49	87	101	165	0	40	43	20	35	20	24	29	56	59	57	11	38	
25	105	137	40	22	121	78	275	24	30	51	89	99	197	0	47	38	21	37	14	28	21	52	52	57	13	29	
26	89	152	50	21	134	94	269	26	43	46	87	112	192	0	35	49	23	42	9	29	11	46	60	56	13	29	
27	89	163	67	33	142	83	302	25	35	40	84	116	202	0	49	52	27	31	15	29	15	56	53	55	10	37	
28	103	151	61	33	148	93	295	31	35	39	85	122	220	0	49	50	26	35	10	30	28	56	56	54	6	26	
29	113	144	60	28	148	90	295	37	34	45	77	123	209	0	53	49	31	43	16	28	31	58	54	52	12	25	
30	127	152	65	19	159	103	322	25	35	62	84	103	206	0	60	56	31	34	15	33	23	42	54	70	12	36	
31	113	135	56	19	147	71	321	14	27	38	69	100	207	0	55	56	13	28	14	31	24	51	44	55	14	24	
32	104	119	34	10	125	67	307	12	30	25	67	82	179	0	55	62	18	28	12	35	20	51	57	48	12	30	
33	110	110	35	15	133	65	312	14	28	30	70	80	191	0	46	49	19	27	13	39	32	38	37	40	14	23	
34	105	101	50	18	136	70	292	13	25	33	70	90	193	0	63	47	25	17	13	30	30	36	49	39	11	25	
35	9	2	6	3	10	3	17	3	2	2	4	6	16	0	3	6	2	2	1	1	2	1	1	0	0	2	

I gained insight into measuring user engagement across different devices on a weekly basis.

Device-specific engagement analysis assists in tailoring user experiences for different platforms and optimizing the company's multi-device strategy.

## 5] calculate email engagement metrics-

*Query-*

```
SELECT Week,  
ROUND((weekly_digest/total* 100),2) AS "Weekly Digest Rate",  
ROUND((email_opens/total* 100),2) AS "Email Open Rate",  
ROUND((email_clickthroughs/total*100),2) AS "Email Clickthrough  
Rate", ROUND((reengagement_emails/total*100),2) AS  
"Reengagement Email Rate"  
  
FROM(  
  
SELECT EXTRACT(WEEK FROM occurred_at) AS Week, COUNT(CASE  
WHEN action = 'sent weekly digest' THEN user_id ELSE NULL END) AS  
weekly_digest,  
  
COUNT(CASE WHEN action = 'email_open' THEN user_id ELSE NULL  
END) AS email_opens,  
  
COUNT(CASE WHEN action = 'email_clickthrough' THEN user_id ELSE  
NULL END) AS email_clickthroughs,  
  
COUNT(CASE WHEN action = 'sent_reengagement_email' THEN  
user_id ELSE NULL END) AS reengagement_emails,  
  
COUNT(user_id) AS total  
  
FROM EmailEvents  
  
GROUP BY 1  
  
) sub  
  
GROUP BY 1  
  
ORDER BY 1;
```

*Output-*

	Week	Weekly Digest Rate	Email Open Rate	Email Clickthrough Rate	Reengagement Email Rate
▶	17	0.00	21.28	11.39	5.01
	18	0.00	22.24	10.49	3.83
	19	0.00	22.67	11.13	4.04
	20	0.00	22.64	11.43	4.31
	21	0.00	22.82	9.97	3.69
	22	0.00	21.56	10.66	4.19
	23	0.00	22.34	11.18	4.09
	24	0.00	22.92	10.99	4.48
	25	0.00	21.79	10.54	3.90
	26	0.00	22.22	10.61	4.18
	27	0.00	22.49	11.37	3.90
	28	0.00	22.48	10.77	3.83
	29	0.00	21.71	10.51	3.79
	30	0.00	23.24	10.59	3.88
	31	0.00	23.25	7.66	3.82
	32	0.00	22.85	7.14	3.42
	33	0.00	23.10	7.91	4.26
	34	0.00	23.91	7.67	4.08
	35	0.00	32.28	29.92	37.80

I understood how to assess user interactions with email services, which can inform email marketing strategies.

Analyzing email engagement metrics aids in crafting personalized communication and improving user engagement through targeted campaigns.

---

## *RESULTS :*

---

Throughout this project focused on operational analytics and investigating metric spikes, I gained a deeper understanding of the critical role data analysis plays in improving a company's performance and decision-making. By working on various tasks, I encountered real-world challenges and had the opportunity to apply advanced SQL skills to derive actionable insights from data.

Overall, this project provided invaluable hands-on experience in operational analytics, from understanding business processes to

performing SQL-based data analysis. The tasks highlighted the significance of data-driven decision-making, trend identification, and understanding sudden changes in key metrics. These insights are immensely helpful in guiding data-driven strategies to improve company operations and enhance overall performance.

---

*Thank you !*

---