

# OPERATION & METRIC ANALYSIS

## Project Discription

Opeartion Analytics is the analysi done for the complete end to end operations of the company.

Investing Metric spike is also an important parts of Opeartion Analytics as being a Data Analyst we must be able to understand or make our team understood.

In the **first Case study** It involves analyzing job data to imporve operational efficiency.

And the **second Case** study looking at product information to find out some important facts and show key performance measures.This includes analyzing the data to indentify pattern and trends such as user growth overtime,user engagement,cohort analysis and email metrics.

## Approach

First I prepared the data and analyze using SQL, I use many SQL fucntion to extract the meaningful information form the given dataset.

After that once the data is cleaned I create the database and then the tables using the structures and link provided by the trainity team.

Then, I performed analysis to generate valuable insights for the company.

## Tech-Used

To compelete this project the software that I used is

**My-SQL ver 8.0.33** because of it's better performance for read/write workloads and anotherone is **WPS office** to making the report for the leadership team.

## CASE STUDY 1: JOB DATA

### **1. Calculate the number of jobs reviewed per hour per day for November 2020?**

```
SELECT DISTINCT ds AS days,  
Count(job_id) / (Sum(time_spent) / 3600) AS no_of_jobs_reviewed  
FROM task1  
GROUP BY days;
```

**Output:**

days	no_of_jobs_reviewed
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

**2. Let's how the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput do you prefer daily metric or 7-days rolling and why?**

```
SELECT a.ds AS day,
a.throughput,
avg(a.throughput) OVER ( ORDER BY ds rows BETWEEN 6 PRECEDING AND CURRENT
row ) AS
7_day_avg_of_throughput
FROM
( SELECT ds, count(job_id) / sum(time_spent) AS throughput FROM task1 GROUP BY
ds ) AS a
GROUP BY ds;
```

**Output:**

day	throughput	7_day_avg_of_throughput
2020-11-25	0.0222	0.02220000
2020-11-26	0.0179	0.02005000
2020-11-27	0.0096	0.01656667
2020-11-28	0.0606	0.02757500
2020-11-29	0.0500	0.03206000
2020-11-30	0.0500	0.03505000

**3. Calculate the percentage share of each language in the last 30 days?**

```
SELECT language,
count(job_id) as no_of_jobs,
count(job_id)*100 / sum(count(*)) OVER() as percentage_share
FROM task1
WHERE ds between '2020-11-01' and '2020-11-30'
GROUP by language;
```

**Output:**

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

**4. Let's say you see some duplicate rows in data. How will you display duplicates from the table?**

```
SELECT a.ds,
a.job_id,
a.actor_id,
a.event,
a.language,
a.time_spent,
a.org,
CASE when a.duplicates = 1 then "No Duplicate" else "Duplicate" end as Duplicate
FROM
( SELECT *, row_number() OVER (partition by ds, job_id, actor_id, event, language,
time_spent, org)
as duplicates FROM task1 ) as a ;
```

**Output:**

ds	job_id	actor_id	event	language	time_spent	org	Duplicate
2020-11-25	20	1003	transfer	Italian	45	C	No Duplicate
2020-11-26	23	1004	skip	Persian	56	A	No Duplicate
2020-11-27	11	1007	decision	French	104	D	No Duplicate
2020-11-28	23	1005	transfer	Persian	22	D	No Duplicate
2020-11-28	25	1002	decision	Hindi	11	B	No Duplicate
2020-11-29	23	1003	decision	Persian	20	C	No Duplicate
2020-11-30	21	1001	skip	English	15	A	No Duplicate
2020-11-30	22	1006	transfer	Arabic	25	B	No Duplicate

## CASE STUDY 2: INVESTING METRIC SPIKE

**1. Calculate the weekly user engagement?**

```

SELECT week(occurred_at) as Week,
count(DISTINCT user_id)as Weekly_User_engagement
FROM events
GROUP BY week(occurred_at)
ORDER BY week(occurred_at);

```

**Output:**

Week	Weekly_User_engagement
17	740
18	1260
19	1287
20	1351
21	1299
22	1381
23	1446
24	1471
25	1459
26	1509
27	1573
28	1577
29	1607
30	1706
31	1514
32	1454
33	1438
34	1443
35	118

**2. Calculate the user growth for product?**

```

SET @g := 0;
SELECT a.no_of_users, a.date,
( @g := @g + a.no_of_users ) as user_growth
FROM
( SELECT count(user_id) as no_of_users,
date(created_at) as date
FROM users WHERE state = "active"
GROUP BY date(created_at) ) a;

```

**Output:**

no_of_users	date	user_growth
7	2013-01-01	7
7	2013-01-02	14
6	2013-01-03	20
1	2013-01-04	21
2	2013-01-05	23
3	2013-01-06	26
4	2013-01-07	30
2	2013-01-08	32
6	2013-01-09	38
6	2013-01-10	44
6	2013-01-11	50
3	2013-01-12	53

### 3. Calculate the weekly retention of user-sign up cohort?

```
SELECT DISTINCT u.user_id,
e.occurred_at
FROM users u join events e on u.user_id = e.user_id
WHERE u.activated_at > '2014-05-01' and e.event_name = 'login'
GROUP BY week( e.occurred_at)
ORDER BY e.occurred_at;
```

#### Output:

user_id	occurred_at
11768.0	2014-05-01 08:03:12
11775.0	2014-05-09 12:21:24
11787.0	2014-05-13 15:59:11
11799.0	2014-05-20 07:56:03
11778.0	2014-05-28 14:10:56
11901.0	2014-06-03 15:12:16
11778.0	2014-06-09 07:06:11
12741.0	2014-06-20 14:31:18
12882.0	2014-06-22 18:46:04
13441.0	2014-07-04 10:33:32
13942.0	2014-07-12 08:53:20
13757.0	2014-07-18 20:39:28
13317.0	2014-07-20 12:18:05
13743.0	2014-07-30 06:10:40
14237.0	2014-08-04 23:54:39

### 4. Calculate the weekly engagement per device?

```
SELECT DISTINCT u.user_id,
e.occurred_at
FROM users u join events e on u.user_id = e.user_id
WHERE u.activated_at > '2014-05-01' and e.event_name = 'login'
```

```
GROUP BY week( e.occurred_at)
ORDER BY e.occurred_at;
```

**Output:**

Weeks	device	User_engagement
17	acer aspire desktop	12
17	acer aspire notebook	23
17	amazon fire phone	4
17	asus chromebook	23
17	dell inspiron desktop	20
17	dell inspiron notebook	48
17	hp pavilion desktop	17
17	htc one	19
17	ipad air	29
17	ipad mini	19
17	iphone 4s	27
17	iphone 5	69
17	iphone 5s	47
17	kindle fire	6
17	lenovo thinkpad	94
17	mac mini	7
17	macbook air	61
17	macbook pro	154

**5. Calculate email engagement metrics?**

```
SELECT week(occurred_at) as Week,
count( DISTINCT ( CASE WHEN action = "sent_weekly_digest"
THEN user_id end )) as weekly_digest,
count( distinct ( CASE WHEN action = "sent_reengagement_email"
THEN user_id end )) as reengagement_mail,
count( distinct ( CASE WHEN action = "email_open"
THEN user_id end )) as opened_email,
count( distinct ( CASE WHEN action = "email_clickthrough"
THEN user_id end )) as email_clickthrough
FROM emails
GROUP BY week(occurred_at)
ORDER BY week(occurred_at);
```

### Output:

Week	weekly_digest	reengagement_mail	opened_email	email_clickthrough
17	908	73	310	166
18	2602	157	900	425
19	2665	173	961	476
20	2733	191	989	501
21	2822	164	996	436
22	2911	192	965	478
23	3003	197	1057	529
24	3105	226	1136	549
25	3207	196	1084	524
26	3302	219	1149	550
27	3399	213	1207	613
28	3499	213	1228	594
29	3592	213	1201	583
30	3706	231	1363	625
31	3793	222	1338	444

## Insights

### Case study1

- > On average 126 jobs, highest 218 jobs and the lowest 34 jobs were reviewed per year in November.
- > We used 7 days rolling average of throughput as it gives the avg for all the days right from day 1 to 7 whereas daily metrics gives the average for only that particular day itself.
- > The percentage of share of persian language is the most 37.5%.
- > No duplicate rows are present in the data.

### Case study2

- > The weekly user engagement increased from week 18<sup>th</sup> to week 31<sup>st</sup> and then declining from then onwards.
- > Total number of users in august 2014 is 19066.
- > The average retention rate after 1st week of sign-up is 72.5%.
- > The device with highest weekly engagement is Macbook Pro
- > On average, 1117 users open emails in which 493 click on links in email. Weekly digest has the highest weekly engagement with an average of 3181.

## Results

Through this project I was able to get hands on experience and also improve my skill in SQL.

This project also helps me to learn the new business concepts like KPIs and other metrics.