

PROJECT 3

OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

INTRODUCTION

Operation analytics involves analyzing and optimizing business operations to improve efficiency, productivity, and customer satisfaction.

Metric analytics focuses on tracking, measuring, and interpreting key performance indicators (KPIs) to gain valuable insights.

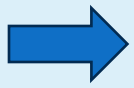
IMPORTANCE OF USING OPERATION AND METRIC ANALYTICS

- ✦ **Efficiency Boost:** Operations analytics optimizes processes
- ✦ **Performance Snapshot:** Metrics reveal strengths/weaknesses
- ✦ **Real-time Navigation:** Like a business GPS.
- ✦ **Actionable insights:** Turn data into decisions.

PROJECT DESCRIPTION

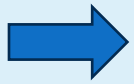
Welcome to our presentation on Operation and Metric Analytics. This project takes you through the core of efficient operations and strategic decision-making. From enhancing efficiency to predicting trends ,this analysis is done for the end to end operations of the company. It helps in spotting the areas to work upon for thee growth of the company.

APPROACH



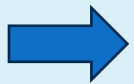
DATA COLLECTION -

Identifying relevant data from the data set provided



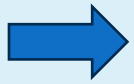
EXPLORATORY ANALYSIS-

Discovering patterns, trends, and correlations in the data to gain initial insights



DATA ANALYSIS-

Done via SQL queries to answer the business questions



DATA VISUALIZATION-

Bringing data to life with visuals and sharing final report

TECH STACK USED

MS EXCEL – Data is in the form of .csv file from which the data will be processed

MYSQL COMMUNITY EDITION –

- Free to use
- Good speed and performance
- Deals with the data in form of tables
- Can handle wide range of data

MS POWERPOINT –

Final report is visualized in the form of presentation



CASE STUDY - 1




A. JOBS REVIEWED OVER TIME

```
3 #TASK 1 JOBS REVIEWED OVER TIME
4 • select avg(h) as 'avg jobs reviewed per hour',
5    avg(s) as 'avg jobs reviewed per second'
6 from (
7     select ds,
8        ((count(job_id)*3600)/sum(time_spent)) as h,
9        ((count(job_id))/sum(time_spent)) as s
10    from job_data
11    where
12        month(ds)=11
13    group by ds)a;
```

Result Grid			Filter Rows:	Export:
	avg jobs reviewed per hour	avg jobs reviewed per second		
▶	126.18048333	0.03505000		

B. THROUGHPUT ANALYSIS

```
16 #TASK 2 THROUGHPUT ANALYSIS
17 • SELECT ds as review_date, jobs_reviewed,
18     AVG(jobs_reviewed)
19     OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS 7_rolling_days_avg
20 FROM ( SELECT ds, COUNT(job_id) AS jobs_reviewed FROM job_data GROUP BY ds ORDER BY ds ) a;
21
```


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

review_date	jobs_reviewed	7_rolling_days_avg
2020-11-25	1	1.0000
2020-11-26	1	1.0000
2020-11-27	1	1.0000
2020-11-28	2	1.2500
2020-11-29	1	1.2000
2020-11-30	2	1.3333

The daily metric might be more suitable if the goal is to capture short term trends or fluctuations whereas 7-day rolling average is suitable for the overall trend while minimizing the daily variations

C. LANGUAGE SHARE ANALYSIS

```
25      #task 3 language share analysis
26 •    SELECT language AS Languages, ROUND(100*COUNT(*)/total,2)
27      AS Percentage, sub.total
28      FROM job_data
29      CROSS JOIN (SELECT COUNT(*) AS total FROM job_data) AS sub
30      GROUP BY language, sub.total;
```

Result Grid  Filter Rows: <input type="text"/>			
	Languages	Percentage	total
▶	English	12.50	8
	Arabic	12.50	8
	Persian	37.50	8
	Hindi	12.50	8
	French	12.50	8
	Italian	12.50	8

D. DUPLICATE ROWS DETECTION

```
37      #TASK 4 DUPLICATE ROWS DETECTION
38 •    SELECT actor_id, COUNT(*) AS Duplicates FROM job_data
39      GROUP BY actor_id HAVING COUNT(*) >1;
40
```

Result Grid



Filter Rows:



	actor_id	Duplicates
▶	1003	2

CASE STUDY - 2

A. WEEKLY USER ENGAGEMENT

```
2  #TASK 1 weekly user engagement
3  •  select
4  extract(year from occurred_at) as Year,
5  extract(week from occurred_at) as weekNo,
6  count(distinct user_id) as User_Engagement
7  from events
8  group by 1,2
9  order by 1,2;
```

10

Result Grid   Filter Rows: <input type="text"/>			
	Year	weekNo	User_Engagement
▶	2014	17	663
	2014	18	1068
	2014	19	1113
	2014	20	1154
	2014	21	1121
	2014	22	1186
	2014	23	1232
	2014	24	1275
	2014	25	1264
	2014	26	1302
	2014	27	1372
	2014	28	1365
	2014	29	1376
	2014	30	1467
	2014	31	1299
	2014	32	1225
	2014	33	1225
	2014	34	1204
	2014	35	104


B. USER GROWTH ANALYSIS

```
13
14 #TASK 2 USER GROWTH ANALYSIS
15 • select
16     year,
17     weeknum,
18     num_active_user,
19     SUM(num_active_user)OVER
20     (ORDER BY year_num, week_num
21     ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
22     AS cum_active_user
23 from(
24     select
25         extract(year from a.activated_at) as year_num,
26         extract(week from a.activated_at) as week_num,
27         count(distinct user_id) as num_active_user
28     from
29         users a
30     WHERE
31         state = 'active'
32     group by year,weeknum
33     order by year,weeknum
34 ) a;
```

Result Grid					Filter Rows:	Export:
	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					×	

C. WEEKLY RETENTION ANALYSIS

```
39  #TASK 3 WEEKLY RETENTION ANALYSIS
40  • SELECT
41  distinct user_id,
42  COUNT(user_id) as total_users,
43  SUM(CASE WHEN retention_week = 1 Then 1 Else 0 END) as retention
44  FROM (
45  SELECT
46  a.user_id,
47  a.signup_week,
48  b.engagement_week,
49  b.engagement_week - a.signup_week as retention_week
50  FROM
51  (
52  (SELECT distinct user_id, extract(week from occurred_at) as signup_week from events
53   WHERE event_type = 'signup_flow'
54   and event_name = 'complete_signup'
55   and extract(week from occurred_at) = 18
56  )a
57  LEFT JOIN
58  (SELECT distinct user_id, extract(week from occurred_at) as engagement_week FROM events
59   where event_type = 'engagement'
60  )b
61  on a.user_id = b.user_id
62  )
63  )d
64  group by user_id
65  order by user_id
66  ;
```

Result Grid  Filter Rows: <input type="text"/>			
	user_id	total_users	retention
▶	11919	2	0
	11920	1	0
	11924	1	0
	11926	8	1
	11928	8	0
	11929	1	0
	11931	6	1
	11933	6	1
	11936	3	0
	11939	3	1
	11940	4	1
	11942	7	1
	11944	3	1
	11947	2	1

163 rows returned

D. Weekly Engagement Per Device

```
70 #TASK 4 WEEKLY ENGAGEMENT PER DEVICE
71 • SELECT
72     extract(year from occurred_at) as year,
73     extract(week from occurred_at) as week_num,
74     device,
75     COUNT(distinct user_id) as users
76 FROM
77     events
78 where event_type = 'engagement'
79 GROUP by 1,2,3
80 order by 1,2,3;
81
```

Result Grid			 Filter Rows:	<input type="text"/>	Export: 
	year	week_num	device	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	
	2014	17	lenovo thinkpad	86	
	2014	17	mac mini	6	
	2014	17	macbook air	54	
	2014	17	macbook pro	143	
	2014	17	nexus 10	16	
	2014	17	nexus 5	40	
	2014	17	nexus 7	18	
	2014	17	nokia lumia 635	17	
	2014	17	samsung galaxy tablet	8	
	2014	17	samsung galaxy note	7	
	2014	17	samsung galaxy s4	52	

491 Rows returned

E. Email Engagement Analysis:

```
85 #TASK 5
86 • SELECT
87     100.0 * SUM(CASE WHEN action IN ('email_open') THEN 1 ELSE 0 END) /
88     SUM(CASE WHEN action IN ('sent_weekly_digest', 'sent_reengagement_email', 'email_open') THEN 1 ELSE 0 END)
89     AS emails_rate_of_opening,
90     100.0 * SUM(CASE WHEN action = 'email_clickthrough' THEN 1 ELSE 0 END) /
91     SUM(CASE WHEN action IN ('sent_weekly_digest', 'sent_reengagement_email', 'email_clickthrough') THEN 1 ELSE 0 END)
92     AS emails_rate_of_opening
93 FROM email_events;
```

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

	emails_rate_of_opening	emails_rate_of_opening
▶	25.14039	12.88431

RESULT



- Jobs reviewed over time
- No. of events per second
- Percentage share of each language
- Duplicate rows detection
- Weekly user engagement
- User growth for a product
- Weekly retention
- Weekly engagement per device
- Email engagement