

Project Description:

- **Operation Analytics** – Operation Analytics is a category of business analytics that's focused on syncing data from your warehouse directly to your business tools – thus ensuring that everyone across the organization has access to the same data regardless of their technical skills. Operations analytics is a subfield of data analytics that focuses on using data to optimize and improve operational processes within an organization.
- **Metric Spike** – A metric spike refers to a sudden and increase in a particular metric or measurement that is being tracked or analyzed. This can occur in a wide range of data sets and can be observed over a relatively short period of time.

Approach:

- Download all data provided
- Create database with the dataset provided for Case Study 1 (Job Data)
- Upload datasets of Case Study 2 (Investigating Metric Spike) to MySQL Workbench
- Write queries to find answers

Tech-Stack Used – MySQL Workbench 8.0

Case Study 1 (Job Data)

A. No. of jobs reviewed: Amount of jobs reviewed over time.

Task: Calculate the no. of jobs reviewed per hour per day of November 2020

Query - `SELECT ds, round(SUM(time_spent) / COUNT(*), 2) AS avg_time_spent FROM lang
WHERE ds BETWEEN '01-11-2020' AND '30-11-2020'
GROUP BY ds ORDER BY ds`

Output –

ds	avg_time_spent
25-11-2020	45.00
26-11-2020	56.00
27-11-2020	104.00
28-11-2020	16.50
29-11-2020	20.00
30-11-2020	20.00

B. Throughput: It is the no. of events happening per second.

Task: Assuming the above metric is called throughput. Calculate the 7 day rolling average of throughput.

Query - `SELECT ds, event_per_day,
AVG(event_per_day)OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW)
AS 7_day_rolling_avg FROM
(SELECT ds, COUNT(DISTINCT event) AS event_per_day FROM lang
WHERE ds BETWEEN '01-11-2020' and '30-11-2020' GROUP BY ds ORDER BY ds)a`

Output –

ds	event_per_day	7_day_rolling_avg
25-11-2020	1	1.0000
26-11-2020	1	1.0000
27-11-2020	1	1.0000
28-11-2020	2	1.2500
29-11-2020	1	1.2000
30-11-2020	2	1.3333

C. Percentage Share of each language – Share of each language for different contents.

Task – Calculate the percentage share of each language in the last 30 days.

Query - select language, count(lang.language) as total_of_each_language,
((count(lang.language)/(select count(*)
from lang
group by lang.language
percentage_share_of_each_language

Output –

language	total_of_each_language	percentage_share_of_each_language
English	1	12.5000
Arabic	1	12.5000
Persian	3	37.5000
Hindi	1	12.5000
French	1	12.5000
Italian	1	12.5000

Duplicate Rows: Rows that have the same value present in them.

Task – Display duplicate rows in the data

Query - select * from lang
where job_id in(
select job_id from lang
group by job_id
having count(*) > 1)

Output –

ds	job_id	actor_id	event	language	time_spent	org
29-11-2020	23	1003	decision	Persian	20	C
28-11-2020	23	1005	transfer	Persian	22	D
26-11-2020	23	1004	skip	Persian	56	A

Case Study 2 – INVESTIGATING METRIC SPIKE

A. Calculate Weekly User Engagement

Query: select week(occured_at) as Week,
count(distinct user_id) as Weekly_User_Engagement
from events
group by week(occured_at)
order by week(occured_at)

Output –

Week	Weekly_User_Engagement
NULL	9760

B. Task - Calculate weekly retention of users- signup cohort

Query - select user_id, occurred_at

from users

where occurred_at > '2014-05-01'

order by user_id;

Output –

user_id	occured_at
11801	2014-05-01 10:14:27
11802	2014-05-01 10:19:35
11803	2014-05-01 13:16:37
11804	2014-05-01 08:08:46
11805	2014-05-01 17:15:28
11806	2014-05-01 12:26:42
11807	2014-05-01 09:35:33
11808	2014-05-01 13:20:20
11809	2014-05-01 15:39:11
11810	2014-05-01 09:49:18
11811	2014-05-01 13:57:13
11812	2014-05-01 07:49:45
11813	2014-05-01 13:47:47

C. Task – Calculate weekly engagement per device

Query – SELECT week(occured_at) as Weeks,

device,

count(distinct user_id)as User_engagement

FROM events

GROUP BY device,

week(occured_at)

ORDER BY week(occured_at);

Output –

Weeks	device	User_engagement
17	acer aspire desktop	259
17	acer aspire notebook	437
17	amazon fire phone	118
17	asus chromebook	460
17	dell inspiron desktop	450
17	dell inspiron notebook	848
17	hp pavilion desktop	437
17	htc one	251
17	ipad air	590
17	ipad mini	378
17	iphone 4s	531
17	iphone 5	1291
17	iphone 5s	798
17	kindle fire	239
17	lenovo thinkpad	1727
17	mac mini	177
17	macbook air	1268
17	macbook pro	2559
17	nexus 10	347
17	nexus 5	819
17	nexus 7	442
17	nokia lumia 635	271
17	samsung galaxy table	135
17	samsung galaxy note	150
17	samsung galaxy s4	1027
17	windows surface	228

Task – Calculate email engagement metrics

```
Query – SELECT week(occured_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 email  
GROUP BY week(occured_at)  
ORDER BY week(occured_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
32	3897	200	1318	416
33	4012	264	1417	490
34	4111	261	1502	481
35	0	48	41	38