OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

DESCRIPTION

Operational Analytics is a crucial process that involves analyzing a company's end-to-end operations. This analysis helps identify areas for improvement within the company. As a Data Analyst, you'll work closely with various teams, such as operations, support, and marketing, helping them derive valuable insights from the data they collect.

One of the key aspects of Operational Analytics is investigating metric spikes. This involves understanding and explaining sudden changes in key metrics, such as a dip in daily user engagement or a drop in sales. As a Data Analyst, you'll need to answer these questions daily, making it crucial to understand how to investigate these metric spikes.

In this project, you'll take on the role of a Lead Data Analyst at a company like Microsoft. You'll be provided with various datasets and tables, and your task will be to derive insights from this data to answer questions posed by different departments within the company. Your goal is to use your advanced SQL skills to analyze the data and provide valuable insights that can help improve the company's operations and understand sudden changes in key metrics.

APPROACH

- Understanding the data
- Getting familiar with the data values and attributes
- For doing analysis, I have been given datasets for each case study. These tables have data related to different metrics. I will be running SQL queries on these tables to find the required information.

TECH STACK Used

- Microsoft Excel
- MySQL 8.0 Workbench

Case Study 1: Job Data Analysis

I 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).

JOBS REVIEWED OVER TIME

Calculate the number of jobs reviewed per hour for each day in November 2020

SQL Query :- select *, round(3600/ time_spent) no_Jobs_per_Hour from job_data;

ds	job_id	actor_id	event	language	time_spent	org	no_Jobs_per_Hour
11/30/2020	21	1001	skip	English	15	A	240
11/30/2020	22	1006	transfer	Arabic	25	B	144
11/29/2020	23	1003	decision	Persian	20	C	180
11/28/2020	23	1005	transfer	Persian	22	D	164
11/28/2020	25	1002	decision	Hindi	11	B	327
11/27/2020	11	1007	decision	French	104	D	35
11/26/2020	23	1004	skip	Persian	56	A	64
11/25/2020	20	1003	transfer	Italian	45	C	80

THROUGHPUT ANALYSIS

Calculate the 7-day rolling average of throughput.

SQL Query :- select *, count(job_id) over(order by ds) Rolling_job_count, sum(time_spent) over(order by ds) time_sum, round (count(job_id) over(order by ds) /sum(time_spent) over(order by ds) ,3) Jobs_per_sec from job_data;

ds	job_id	actor_id	event	language	time_spent	org	Rolling_job_count	time_sum	Jobs_per_sec
11/25/2020	20	1003	transfer	Italian	45	C	1	45	0.022
11/26/2020	23	1004	skip	Persian	56	A	2	101	0.020
11/27/2020	11	1007	decision	French	104	D	3	205	0.015
11/28/2020	23	1005	transfer	Persian	22	D	5	238	0.021
11/28/2020	25	1002	transfer	Hindi	11	B	5	238	0.021
11/29/2020	23	1003	decision	Persian	20	C	6	258	0.023
11/30/2020	21	1001	skip	English	15	A	8	298	0.027
11/30/2020	22	1006	transfer	Arabic	25	B	8	298	0.027

LANGUAGE SHARE ANALYSIS

Calculate the percentage share of each language over the last 30 days.

SQL Query :- select round(count(*)*100/c,1) percent_share, language from job_data cross join (select count(*) c from job_data) a group by language,c ;

select distinct language, count(*) over(),count(*) over(partition by language),round(count(*) over(partition by language) *100/count(*) over(),1) per_share from job_data;

language	count(*) over()	count(*) over(partition by language)	per_share
Arabic	8	1	12.5
English	8	1	12.5
French	8	1	12.5
Hindi	8	1	12.5
Italian	8	1	12.5
Persian	8	3	37.5

DUPLICATE ROWS DETECTION

Display duplicate rows from the job_data table.

SQL Query:- select ds, count(ds) from job_data group by ds having count(ds) > 1;

select * from (select *,(row_number() over(partition by ds)) b from job_data) a where b > 1 ;;

ds	job_id	actor_id	event	language	time_spent	org	b
11/28/2020	25	1002	decision	Hindi	11	B	2
11/30/2020	22	1006	transfer	Arabic	25	B	2

ds	count(ds)
11/30/2020	2
11/28/2020	2

CASE STUDY 2: INVESTIGATING METRIC SPIKE

I 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.

WEEKLY USER ENGAGEMENT

Calculate the weekly user engagement.

SQL Query :- select YEARWEEK(occurred_at) week,count(distinct user_id) from events where event_type = 'engagement' group by week ;

week	count(distinct user_id)
201417	663
201418	1068
201419	1113
201420	1154
201421	1121
201422	1186
201423	1232
201424	1275
201425	1264
201426	1302
201427	1372
201428	1365
201429	1376
201430	1467
201431	1299
201432	1225
201433	1225
201434	1204
201435	104

USER GROWTH ANALYSIS

Calculate the user growth for the product.

SQL Query :- select year(created_at) year, month(created_at) month,count(*) user_per_month from users group by year,month;

year	month	user_per_month
2013	1	160
2013	2	160
2013	3	150
2013	4	181
2013	5	214
2013	6	213
2013	7	284
2013	8	316
2013	9	330
2013	10	390
2013	11	399
2013	12	486
2014	1	552
2014	2	525
2014	3	615
2014	4	726
2014	5	779
2014	6	873
2014	7	997
2014	8	1031

WEEKLY ENGAGEMENT PER DEVICE

calculate the weekly engagement per device.

SQL Query :- select yearweek(occurred_at) week, count(distinct user_id), device from events where event_type ='engagement' group by week,device;

week	count(distinct user_id)	device
201417	9	acer aspire desktop
201417	20	acer aspire notebook
201417	4	amazon fire phone
201417	21	asus chromebook
201417	18	dell inspiron desktop
201417	46	dell inspiron notebook
201417	14	hp pavilion desktop
201417	16	htc one
201417	27	ipad air
201417	19	ipad mini
201417	21	iphone 4s
201417	65	iphone 5
201417	42	iphone 5s
201417	6	kindle fire

EMAIL ENGAGEMENT ANALYSIS

Calculate the email engagement metrics.

SQL Query :- select month(occurred_at) month,action,count(distinct user_id) from emailevents group by month,action;

month	action	count(distinct user_id)
5	email_clickthrough	1703
5	email_open	2681
5	sent_reengagement_email	758
5	sent_weekly_digest	2822
6	email_clickthrough	1915
6	email_open	3037
6	sent_reengagement_email	889
6	sent_weekly_digest	3231
7	email_clickthrough	2267
7	email_open	3457
7	sent_reengagement_email	933
7	sent_weekly_digest	3685
8	email_clickthrough	1804
8	email_open	3879
8	sent_reengagement_email	1073
8	sent_weekly_digest	4111