Operations Analytics & Investigating Metric Spike

Task- 3

Email id: dhritimalyadutta@gmail.com
Data Analytics Trainee

Introduction

In this project, I'll take on the role of a Lead Data Analyst at a company like Microsoft. I have two different datasets.

My task will be to derive insights from this datasets to answer questions posed by different departments within the company.

My goal is to use 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.

Software Used : MySQl Workbench 8.0

Project Description

Case Study 1 : Job Data Analysis

A. Number of jobs reviewed:Calculate the number of jobs reviewed per hour for each day in November 2020.

B. Throughput: Calculate the 7-day rolling average of throughput (number of events per second).

C. Percentage share of each language: Calculate the percentage share of each language in the last 30 days.

D. Duplicate rows: Identify duplicate rows in the data

Insights - Job Data Analysis

A. Number of jobs reviewed:Calculate the number of jobs reviewed per hour for each day in November 2020.

Task: Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.

To find the number of jobs reviewed per hour per day of November 2020:

1. We will use the data from job_id columns of the job_data table.

2. Then we will divide the total count of job_id by (30 days X 24 hours)for finding the number of jobs reviewed per day

3. There are some common values in job_id so we will write two program by taking & not taking common values.

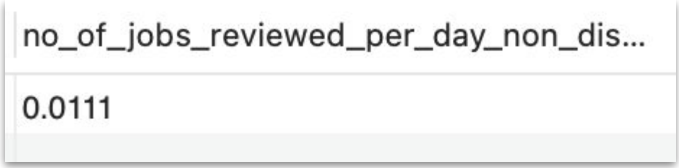
Program and Output

Non Distinct :

```
SELECT count(job_id)/(30*24)

AS
no_of_jobs_reviewed_per_day_non_distinct

FROM job_data;
```



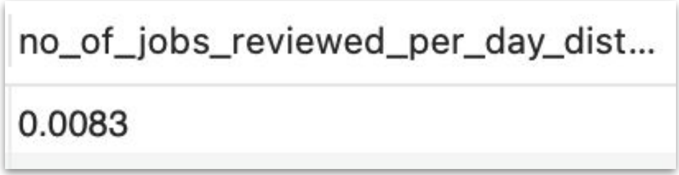
```
no_of_jobs_reviewed_per_day_non_dis...
0.0111
```

Distinct :

```
SELECT count(DISTINCT job_id)/(30*24)

AS
no_of_jobs_reviewed_per_day_distinct

FROM job_data;
```



```
no_of_jobs_reviewed_per_day_dist...
0.0083
```

Insights - Job Data Analysis

B. Throughput: Calculate the 7-day rolling average of throughput (number of events per second).

Task : Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

7-day rolling gives us the average for all the 7 days but daily metric gives us average for only that particular day itself for that reason 7-day rolling is preferred.

For calculating the 7-day rolling average of throughput:-

1. We will be first taking the count of job_id(distinct and non-distinct) and ordering them according to ds (date of interview)

2. Then by using the ROW function. We will be considering the rows between 6 preceding rows and the current row

3. Then we will be taking the average of the jobs_reviewed

Program and Output

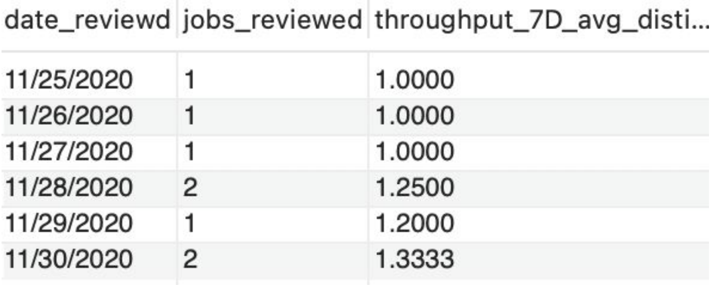
Distinct :

```
SELECT ds AS date_reviewd, jobs_reviewed, AVG(jobs_reviewed)

OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW)

AS throughput_7D_avg FROM ( SELECT ds, COUNT( DISTINCT job_id)

AS jobs_reviewed FROM job_data GROUP BY ds ORDER BY ds )a;
```



date_reviewd	jobs_reviewed	throughput_7D_avg_disti...
11/25/2020	1	1.0000
11/26/2020	1	1.0000
11/27/2020	1	1.0000
11/28/2020	2	1.2500
11/29/2020	1	1.2000
11/30/2020	2	1.3333

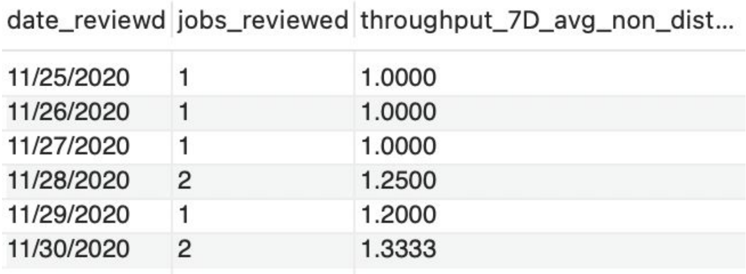
Non Distinct :

```
SELECT ds as date_reviewd, jobs_reviewed,

AVG(jobs_reviewed)

OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW)
AS throughput_7D_avg_non_distinct

FROM ( SELECT ds, COUNT(job_id) AS jobs_reviewed FROM job_data
GROUP BY ds ORDER BY ds ) a;
```



date_reviewd	jobs_reviewed	throughput_7D_avg_non_dist...
11/25/2020	1	1.0000
11/26/2020	1	1.0000
11/27/2020	1	1.0000
11/28/2020	2	1.2500
11/29/2020	1	1.2000
11/30/2020	2	1.3333

Insights - Job Data Analysis

C. Percentage share of each language: Calculate the percentage share of each language in the last 30 days.

Task: Write an SQL query to calculate the percentage share of each language over the last 30 days.

To calculate the percentage share of each language :-

1. We will first divide the total number of languages by the total number of rows presents in the table

2. Then we will do the GROUP BY based on the languages.

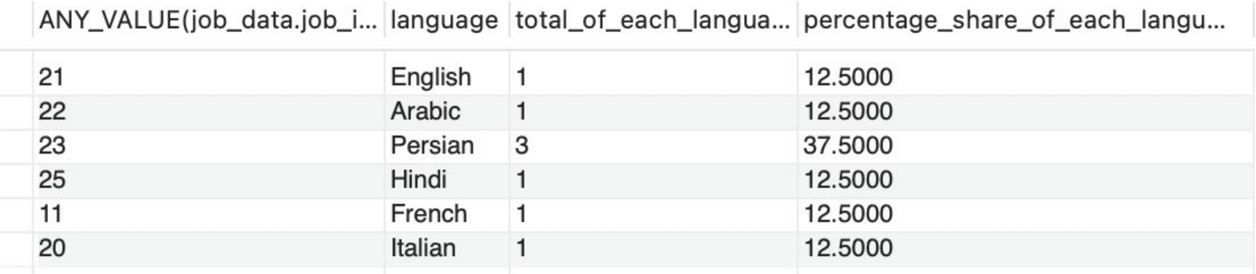
Program and Output

```
SELECT ANY_VALUE(job_data.job_id), job_data.language, COUNT(job_data.language) AS
total_of_each_language,

((COUNT(job_data.language) / (SELECT COUNT(*) FROM job_data)) * 100) AS
percentage_share_of_each_language

FROM job_data

GROUP BY job_data.language;
```



ANY_VALUE(job_data.job_i...	language	total_of_each_langua...	percentage_share_of_each_langu...
21	English	1	12.5000
22	Arabic	1	12.5000
23	Persian	3	37.5000
25	Hindi	1	12.5000
11	French	1	12.5000
20	Italian	1	12.5000

Insights - Job Data Analysis

D. Duplicate rows: Identify duplicate rows in the data

Task: Write an SQL query to display duplicate rows from the job_data table.

To find the duplicate rows :-

1. First find in which do we need to find the duplicate row values

2. Then the column we will use the ROW_NUMBER function to find the row numbers having the same value

3. Then we will portioning the ROW_NUMBER function over the column that we decided

4. Then using the WHERE function to find the row number having value greater than 1

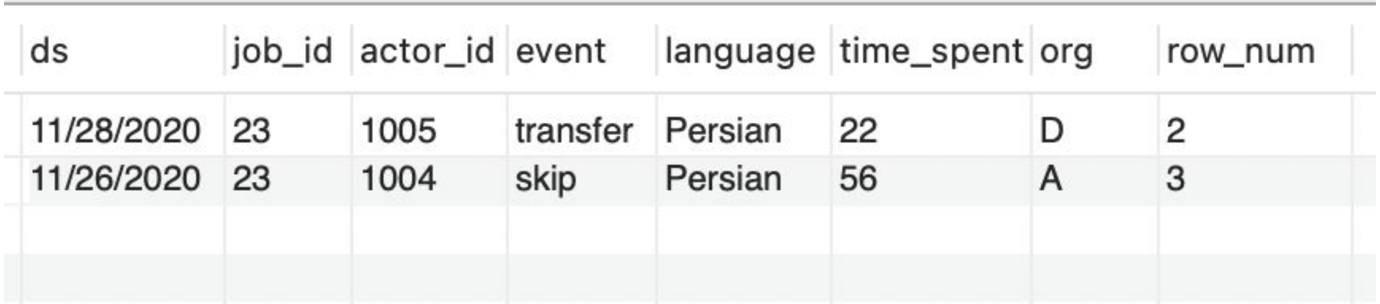
Program and Output

```
SELECT * FROM

(SELECT *, ROW_NUMBER()OVER(PARTITION BY job_id) AS row_num

FROM job_data) a

WHERE row_num>1;
```



ds	job_id	actor_id	event	language	time_spent	org	row_num
11/28/2020	23	1005	transfer	Persian	22	D	2
11/26/2020	23	1004	skip	Persian	56	A	3

Project Description

Case Study 2 : Investigating metric spike

A. **Weekly User Engagement**: Measure the activeness of users on a weekly basis.

B. **User Growth Analysis**: Analyze the growth of users over time for a product.

C. **Weekly Retention Analysis**: Analyze the retention of users on a weekly basis after signing up for a product.

D. **Weekly Engagement Per Device**: Measure the activeness of users on a weekly basis per device.

E. **Email Engagement Analysis**: Analyze how users are engaging with the email service.

Insights - Investigating metric spike

A.Weekly User Engagement: Measure the activeness of users on a weekly basis.

Task: Write an SQL query to calculate the weekly user engagement.

To find the weekly user engagement:-

1. We will extract the week from the occurred_at column of the events table using the EXTRACT function and WEEK function

2. Then we will be counting the number of distinct user_id from the events table

3. Then we will use the GROUP BY function to group the output week from occurred_at

Program and Output

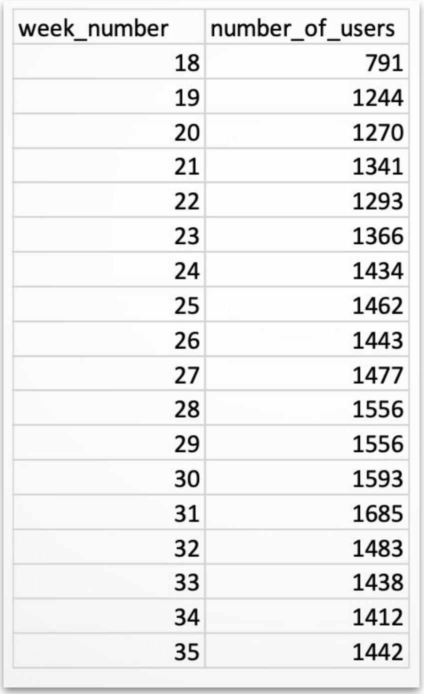
```
SELECT

  extract (week from occurred_at)

 as week_num_user,count(distinct
user_id)

FROM events group by

  week_num_user;
```



week_number	number_of_users
18	791
19	1244
20	1270
21	1341
22	1293
23	1366
24	1434
25	1462
26	1443
27	1477
28	1556
29	1556
30	1593
31	1685
32	1483
33	1438
34	1412
35	1442

Insights - Investigating metric spike

B. User Growth Analysis: Analyze the growth of users over time for a product.

Task: Write an SQL query to calculate the user growth for the product.

User Growth = Number of active users per week

1. First we will the extract the year and week for the occurred_at column of the users table using the extract, year and week functions

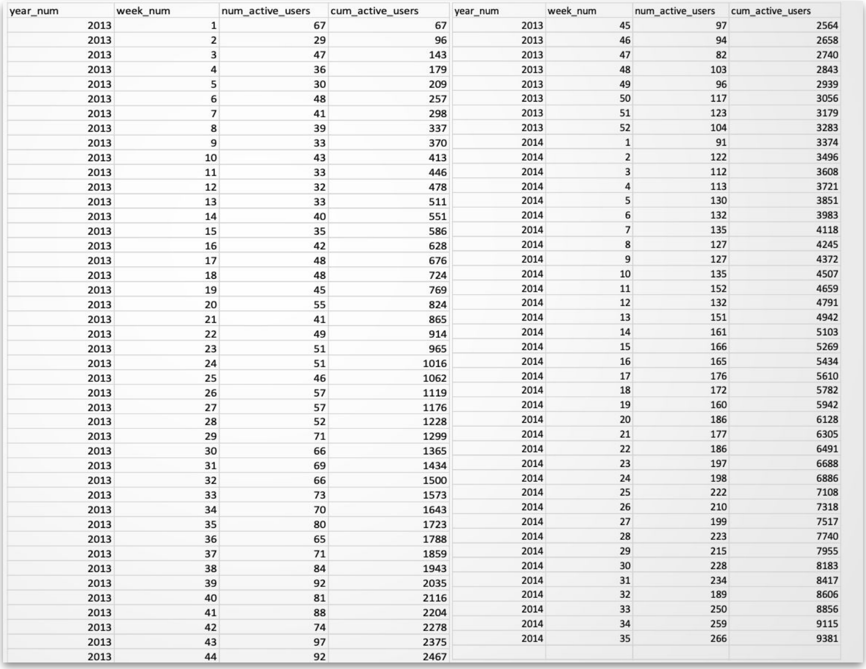
2. Then we will group the extracted week and year on the basis of year and week number

3. Then we ordered the result on the basis of year and week number

4. Then we will find the cumm_active_users using the SUM, OVER and ROW function between unbounded preceding and current row

Program and Output

```
select year_num, week_num,
num_active_users,
SUM(num_active_users)OVER(ORDER BY
year_num, week_num ROWS BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW)
AS cum_active_users from ( select extract (year
from a.activated_at) as year_num, extract
(week from a.activated_at) as week_num,
count(distinct user_id) as num_active_users
from tutorial.yammer_users a WHERE state =
'active' group by year_num,week_num order by
year_num,week_num ) a
```



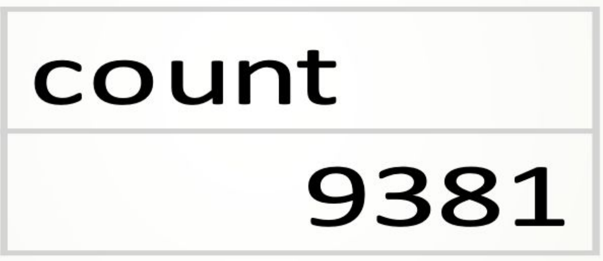
year_num	week_num	num_active_users	cum_active_users
2013	1	67	67
2013	2	29	96
2013	3	47	143
2013	4	36	179
2013	5	30	209
2013	6	48	257
2013	7	41	298
2013	8	39	337
2013	9	33	370
2013	10	43	413
2013	11	33	446
2013	12	32	478
2013	13	33	511
2013	14	40	551
2013	15	35	586
2013	16	42	628
2013	17	48	676
2013	18	48	724
2013	19	45	769
2013	20	55	824
2013	21	41	865
2013	22	49	914
2013	23	51	965
2013	24	51	1016
2013	25	46	1062
2013	26	57	1119
2013	27	57	1176
2013	28	52	1228
2013	29	71	1299
2013	30	66	1365
2013	31	69	1434
2013	32	66	1500
2013	33	73	1573
2013	34	70	1643
2013	35	80	1723
2013	36	65	1788
2013	37	71	1859
2013	38	84	1943
2013	39	92	2035
2013	40	81	2116
2013	41	88	2204
2013	42	74	2278
2013	43	97	2375
2013	44	92	2467
2013	45	97	2564
2013	46	94	2658
2013	47	82	2740
2013	48	103	2843
2013	49	96	2939
2013	50	117	3056
2013	51	123	3179
2013	52	104	3283
2014	1	91	3374
2014	2	122	3496
2014	3	112	3608
2014	4	113	3721
2014	5	130	3851
2014	6	132	3983
2014	7	135	4118
2014	8	127	4245
2014	9	127	4372
2014	10	135	4507
2014	11	152	4659
2014	12	132	4791
2014	13	151	4942
2014	14	161	5103
2014	15	166	5269
2014	16	165	5434
2014	17	176	5610
2014	18	172	5782
2014	19	160	5942
2014	20	186	6128
2014	21	177	6305
2014	22	186	6491
2014	23	197	6688
2014	24	198	6886
2014	25	222	7108
2014	26	210	7318
2014	27	199	7517
2014	28	223	7740
2014	29	215	7955
2014	30	228	8183
2014	31	234	8417
2014	32	189	8606
2014	33	250	8856
2014	34	259	9115
2014	35	266	9381

Program and Output

```
-- counting users from user table having state
as active

select count(*) from tutorial.yammer_users

where state = 'active';
```



count
9381

Hence, there are in total 9381 active users from 1st week of 2013 to the 35th week of 2014

Insights - Investigating metric spike

C. Weekly Retention Analysis: Analyze the retention of users on a weekly basis after signing up for a product.

Task: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

1. Firstly we will extract the week from occurred_at column using the extract,week functions

2. Then, we will select out those rows in which event_type = 'signup_flow' and event_name = 'complete_signup'

3. If finding for a specific week we will specify the week number using the extract function

4. Then using the left join we will join the two tables on the basis of user_id where event_type = 'engagement'

5. Then we will use the Group By function to group the output table on the basis of user_id

6. Then we will use the Order By function to order the result table on the basis of user_id

Program and Output

```
SELECT distinct user_id, COUNT(user_id), SUM(CASE WHEN retention_week = 1 Then 1 Else 0 END) as
per_week_retention

FROM ( SELECT a.user_id, a.signup_week, b.engagement_week, b.engagement_week - a.signup_week
as retention_week FROM ( (SELECT distinct user_id, extract(week from occurred_at) as signup_week
from events WHERE event_type = 'signup_flow' and event_name = 'complete_signup' )a

LEFT JOIN (SELECT distinct user_id, extract (week from occurred_at) as engagement_week FROM
events where event_type = 'engagement' )b on a.user_id = b.user_id ) )d group by user_id order by
user_id
```

Output CSV -
https://drive.google.com/drive/folders/1h3-_Mj7gy02tRyT4K5tjznZDbRVRYXAV?usp=sharing

Insights - Investigating metric spike

D.Weekly Engagement Per Device: Measure the activeness of users on a weekly basis per device.

Task: Write an SQL query to calculate the weekly engagement per device.

To find the weekly user engagement per device:-

1. Firstly we will extract the year_num and week_num from the occurred_at column of the events table using the extract, year and week function

2. Then we will select those rows where event_type = 'engagement’ using the WHERE clause

3. Then by using the Group By and Order By function we will group and order the result on the basis of year_num, week_num and devic

Program and Output

```
SELECT extract(year from occurred_at) as year_num,

extract(week from occurred_at) as week_num, device,

COUNT(distinct user_id) as no_of_users

 FROM events where event_type = 'engagement' GROUP by 1,2,3 order by 1,2,3;
```

Output CSV -
https://drive.google.com/file/d/1G3CvqAijaKs-irXzSYCK7sbc-4clCDH4/view?usp=sharing

Insights - Investigating metric spike

E.Email Engagement Analysis: Analyze how users are engaging with the email service.

Task: Write an SQL query to calculate the email engagement metrics.

To find the email engagement metrics(rate) of users:-

1. We will first categorize the action on the basis of email_sent, email_opened and email_clicked using the CASE, WHEN, THEN functions

2. Then we select the sum of category of email_opened divide by the sum of the category of email_sent and multiply the result by 100.0 and name is as email_opening_rate

3. Then we select the sum of category of email_clicked divide by the sum of the category of email_sent and multiply the result by 100.0 and name is as email_clicking_rate

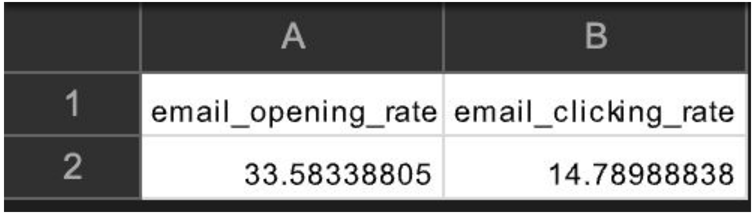
4. email_sent = ('sent_weekly_digest','sent_reengagement_email')

5. email_opened = 'email_open' 6. email_clicked = 'email_clickthrough'

Program and Output

```
SELECT 100.0*SUM(CASE when email_cat = 'email_opened' then 1 else 0 end)/SUM(CASE when
email_cat = 'email_sent' then 1 else 0 end) as email_opening_rate, 100.0*SUM(CASE when email_cat =
'email_clicked' then 1 else 0 end)/SUM(CASE when email_cat = 'email_sent' then 1 else 0 end) as
email_clicking_rate FROM ( SELECT *, CASE

WHEN action in ('sent_weekly_digest','sent_reengagement_email') then 'email_sent' WHEN action in
('email_open') then 'email_opened' WHEN action in ('email_clickthrough') then 'email_clicked' end as
email_cat from tutorial.yammer_emails ) a;
```



	A	B
1	email_opening_rate	email_clicking_rate
2	33.58338805	14.78988838

Result

Operational Analytics is a crucial process that involves analyzing a company's end-to-end operations.

As a Data Analyst, helping my team to use these valuable insights from the data they collect.

This analysis helps identify areas for improvement within the company.

Thank You

Task- 3

Email id: dhritimalyadutta@gmail.com
Data Analytics Trainee