THE

OPERATION & METRIC ANALYTICS

2023 July 23

Shruthi Nambissan

TABLE OF CONTENTS

OVERVIEW

Utilize advanced SQL skills to extract insights from diverse datasets..
The ultimate goal is to improve company operations and understand sudden changes in key metrics as well as address questions from different company departments to make data-driven decisions.

APPROACH

01

DATABASE CREATION

Create tables using the description provided. Use dataset csv to upload data in the respective tables.

02

DERIVING INSIGHTS

Using SQL concepts to address questions from different company departments and understand sudden changes in key metrics.

03

ANALYSIS

Gaining an understanding of what the data represents. Presenting it in a user friendly, easy to understand format to further help decision makers

TECH STACK USED

Oracle Database WINDOWS.X64_1930 00_db_home

SQL Developer 23.1.0

Mode.com

CASE STUDY 1

Job Data Analysis

JOBS REVIEWED OVER TIME

NUMBER OF JOBS REVIEWED: AMOUNT OF JOBS REVIEWED OVER TIME.

```
SELECT
COUNT(JOB_ID) JOBSPERDAY,
DS
FROM JOB_DATA
GROUP BY DS;
```

	JOBSPERDAY	DS
1	1	29-NOV-20
2	1	25-NOV-20
3	2	30-NOV-20
4	1	26-NOV-20
5	1	27-NOV-20
6	2	28-NOV-20

THROUGHPUT ANALYSIS

7-DAY ROLLING AVERAGE OF THROUGHPUT (NUMBER OF EVENTS PER SECOND).

```
SELECT B.DS,
((SUM(B.JOBSPERDAY) OVER (ORDER BY DS
ROWS BETWEEN 6 PRECEDING AND CURRENT
ROW))
/(SUM(B.TIMESPENT) OVER (ORDER BY DS
ROWS BETWEEN 6 PRECEDING AND CURRENT
ROW))) ROLLING_AVG
FROM (SELECT COUNT(JOB_ID)AS
JOBSPERDAY,
SUM(TIME_SPENT) AS TIMESPENT,
DS
FROM JOB_DATA
GROUP BY DS) B;
```

	DS	ROLLING_AVG
1	25-NOV-20	0.022222222222222222222222222222222222222222
2	26-NOV-20	0.019801980198019801980198019801980198019801980198
3	27-NOV-20	0.014634146341463414634146341463414634146341463
4	28-NOV-20	0.021008403361344537815126050420168067226 9
5	29-NOV-20	0.023255813953488372093023255813953488372 1
6	30-NOV-20	0.026845637583892617449664429530201342281 9

DAILY METRIC OR 7-DAY ROLLING ?

Daily Metric for Throughput:

- The daily metric for throughput represents the number of jobs reviewed in a single day.
- This method provides a precise and granular view of the throughput on a day-to-day basis, allowing for real-time monitoring and quick identification of any sudden changes or spikes in performance.
- But daily metrics can be affected by short-term fluctuations and irregularities, making it harder to discern underlying trends and patterns.

7-Day Rolling Average for Throughput:

- The 7-day rolling average is calculated by taking the average throughput over the past seven days at any given point in time.
- Using a rolling average helps to smooth out short-term fluctuations, offering a more stable and representative view of throughput trends over time.
- It helps to capture the overall performance and mitigates the impact of random daily variations but might not provide real-time insights as it lags by several days.

DAILY METRIC OR 7-DAY ROLLING ?

Since this particular project aims at understanding sudden changes in key metrics and goal is to monitor and respond to daily variations and identify immediate issues or opportunities, using the daily metric would be more suitable.

LANGUAGE SHARE ANALYSIS

PERCENTAGE SHARE OF EACH LANGUAGE IN THE LAST 30 DAYS

```
SELECT LANGUAGE,
((COUNT(LANGUAGE)*100)/
(SELECT COUNT(LANGUAGE)
FROM JOB_DATA))
 PERCENTAGE_SHARE
FROM JOB_DATA
GROUP BY LANGUAGE;
```

	LANG...	PERCENTAGE_SHARE
1	Hindi	12.5
2	English	12.5
3	French	12.5
4	Persian	37.5
5	Italian	12.5
6	Arabic	12.5

DUPLICATE ROWS

ROWS THAT HAVE THE SAME VALUE PRESENT IN THEM.

```
SELECT JOB_ID,
ACTOR_ID,
COUNT(*)
FROM JOB_DATA
GROUP BY
JOB_ID,ACTOR_ID
HAVING COUNT(*) >= 1
```

	JOB_ID	ACTOR_ID	COUNT(*)
1	23	1003	1
2	25	1002	1
3	11	1007	1
4	23	1004	1
5	23	1005	1
6	21	1001	1
7	22	1006	1
8	20	1003	1

Here, for each job id there will be a corresponding actor_id , date, language and event. Hence to find the duplicate rows we must consider the combination of more than one column as shown above. In this particular scenario there are no duplicates.

CASE STUDY 2

Investigating Metric Spike

WEEKLY USER ENGAGEMENT

MEASURE THE ACTIVENESS OF USERS ON A WEEKLY BASIS.

```
SELECT EXTRACT(WEEK OF OCCURRED_AT) AS WEEK,
COUNT(DISTINCT(USER_ID)) USER_NO
FROM METRICSPIKE.EVENTS
ORDER BY WEEK;
```

Row	week	user_no
1	17	663
2	18	1068
3	19	1113
4	20	1154
5	21	1121
6	22	1186
7	23	1232
8	24	1275

USER GROWTH ANALYSIS

ANALYZE THE GROWTH OF USERS OVER TIME FOR A PRODUCT.

```
select week,year,noofusers,
(sum(noofusers) over (order by year,week
rows between 1 preceding and current
row)) totalincrease
from
(
select
extract(week from created_at) as week,
extract(year from created_at) as year,
count(distinct(user_id)) noofusers
from MetricSpike.users where state
='active'
group by year,week) order by year,week;
```

Row	week	year	noofusers	totalincrease
1	0	2013	23	23
2	1	2013	30	53
3	2	2013	48	78
4	3	2013	36	84
5	4	2013	30	66
6	5	2013	48	78
7	6	2013	38	86
8	7	2013	42	80
9	8	2013	34	76
10	9	2013	43	77
11	10	2013	32	75

WEEKLY RETENTION ANALYSISIS

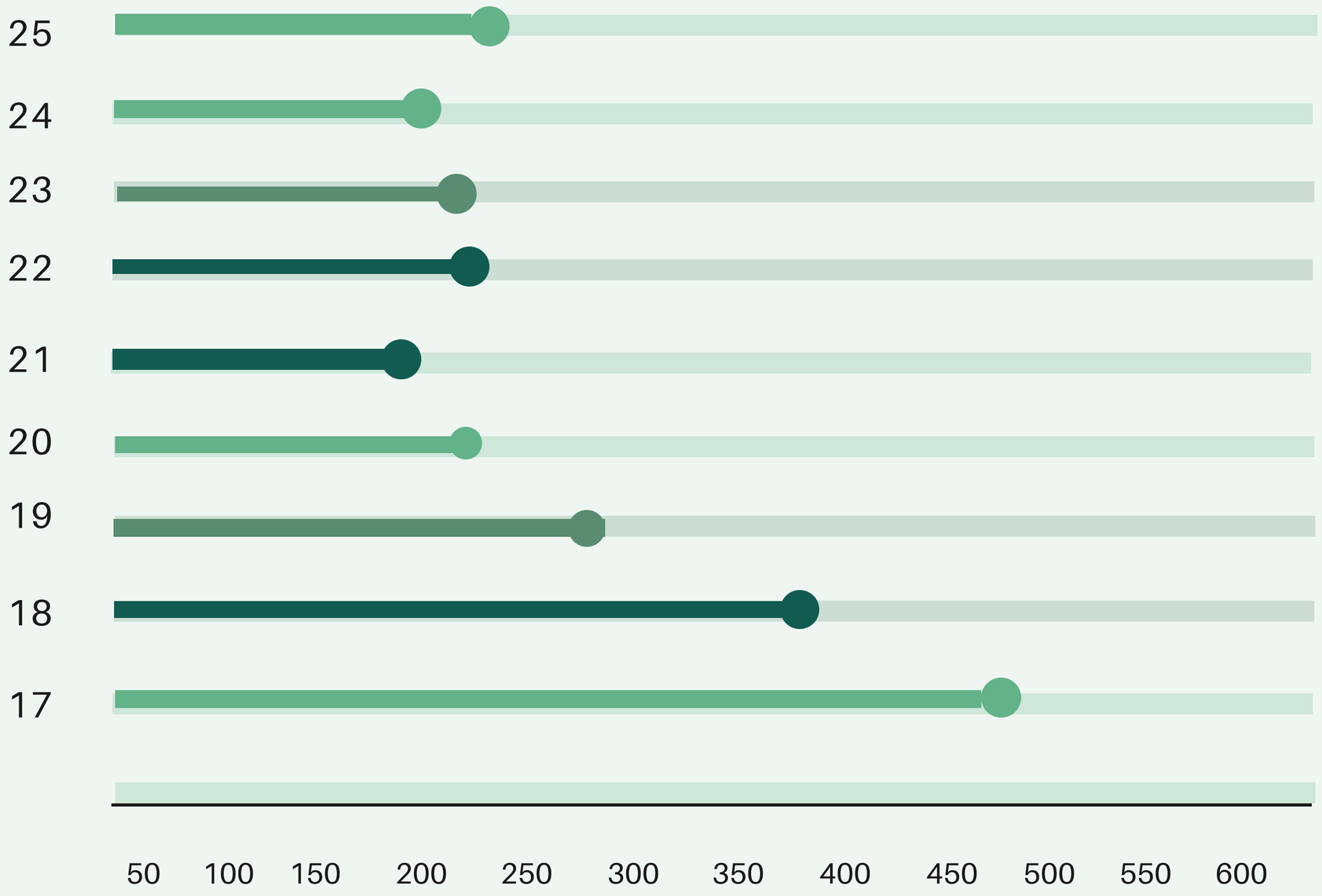
RETENTION OF USERS ON A WEEKLY BASIS AFTER SIGNING UP FOR A PRODUCT

```
 SELECT
FIRSTLOG,
    COUNT(USER_ID),
      SUM(CASE WHEN WEEK_NUMBER = 1 THEN 1 ELSE 0 END) AS
RETENTIONPERWEEK
FROM (
SELECT M.USER_ID, M.SIGNUPWEEK, N.FIRSTLOG, M.SIGNUPWEEK -
FIRSTLOG AS WEEK_NUMBER
FROM
(SELECT  USER_ID, EXTRACT(WEEK FROM OCCURRED_AT) AS SIGNUPWEEK
FROM METRICSPIKE.EVENTS GROUP BY USER_ID,SIGNUPWEEK)M,
(SELECT  USER_ID, MIN(EXTRACT(WEEK FROM OCCURRED_AT)) AS FIRSTLOG
FROM METRICSPIKE.EVENTS GROUP BY USER_ID) N
WHERE M.USER_ID = N.USER_ID
)
GROUP BY FIRSTLOG
ORDER BY FIRSTLOG;
```

Row	week	noofusers	retentionperweek
1	17	3618	472
2	18	2957	362
3	19	1935	284
4	20	1509	223
5	21	1265	187
6	22	1305	224
7	23	1251	219
8	24	1165	205
9	25	1072	218
10	26	913	181
11	27	916	199

Weekly Retention Analysisis

Retention of users on a weekly basis after signing up for a product



WEEKLY ENGAGEMENT PER DEVICE

ANALYZE HOW USERS ARE ENGAGING WITH THE EMAIL SERVICE.

```
SELECT
EXTRACT(YEAR FROM OCCURRED_AT) AS YEAR,
EXTRACT(WEEK FROM OCCURRED_AT) AS WEEK,
DEVICE,
COUNT(DISTINCT USER_ID) AS NOOFUSERS
FROM METRICSPIKE.EVENTS
WHERE EVENT_TYPE = 'ENGAGEMENT'
GROUP BY YEAR,WEEK,DEVICE
ORDER BY YEAR,WEEK,DEVICE;
```

Row	year	week	device	noofusers
1	2014	17	acer aspire desktop	9
2	2014	17	acer aspire notebook	20
3	2014	17	amazon fire phone	4
4	2014	17	asus chromebook	21
5	2014	17	dell inspiron desktop	18
6	2014	17	dell inspiron notebook	46
7	2014	17	hp pavilion desktop	14
8	2014	17	htc one	16
9	2014	17	ipad air	27
10	2014	17	ipad mini	19
11	2014	17	iphone 4s	21

EMAIL ENGAGEMENT ANALYSIS

MEASURE THE ACTIVENESS OF USERS ON A WEEKLY BASIS PER DEVICE.

```
select
(sum(case when email_cat = 'email_opened' then 1 else 0 end)
    /sum(case when email_cat = 'email_sent' then 1 else 0 end)) *100 as
viewingrate,
(sum(case when email_cat = 'email_clicked' then 1 else 0 end)
    /sum(case when email_cat = 'email_sent' then 1 else 0 end))*100 as
clickingrate
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 MetricSpike.email_events
)a;

;
```

Row	viewingrate	clickingrate
1	33.583388049901508	14.789888378200919

DERIVED INSIGHTS

Job Data

Since the requirement of the project is to identify immediate issues or opportunities, using the daily metric would be more suitable.
To find duplicate rows its necessary to use the other column to ensure that the same job_id having different Actor_id is not considered as a duplicate .

Investigating Metric Spike

Macbook Pro seems to be the most preferred device.
Email opening rate is around 34% and email clicking rate is around 15%

THANK YOU