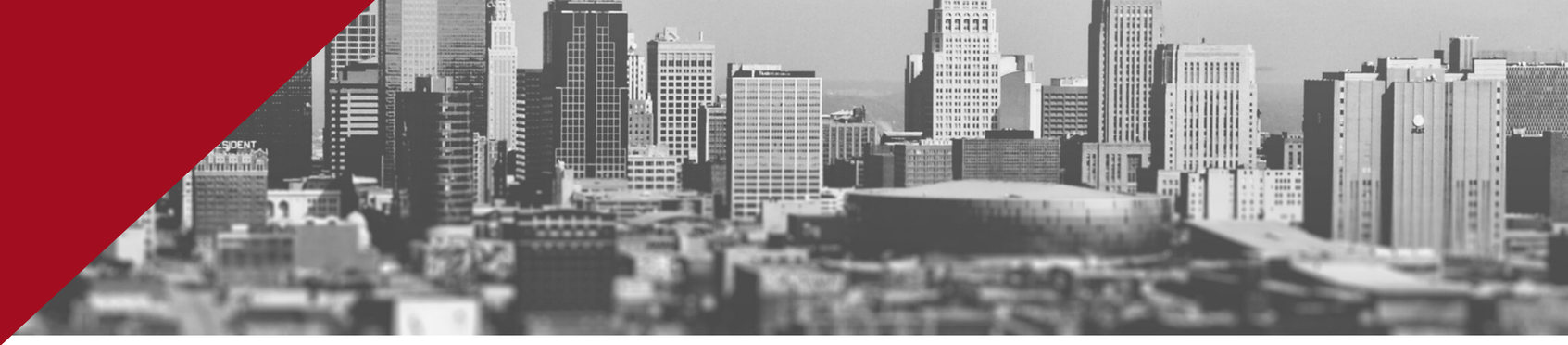
Task 3

Operation Analytics and Investigating Metric Spike

Yash Rajput
Yashrajput9232@gmail.com

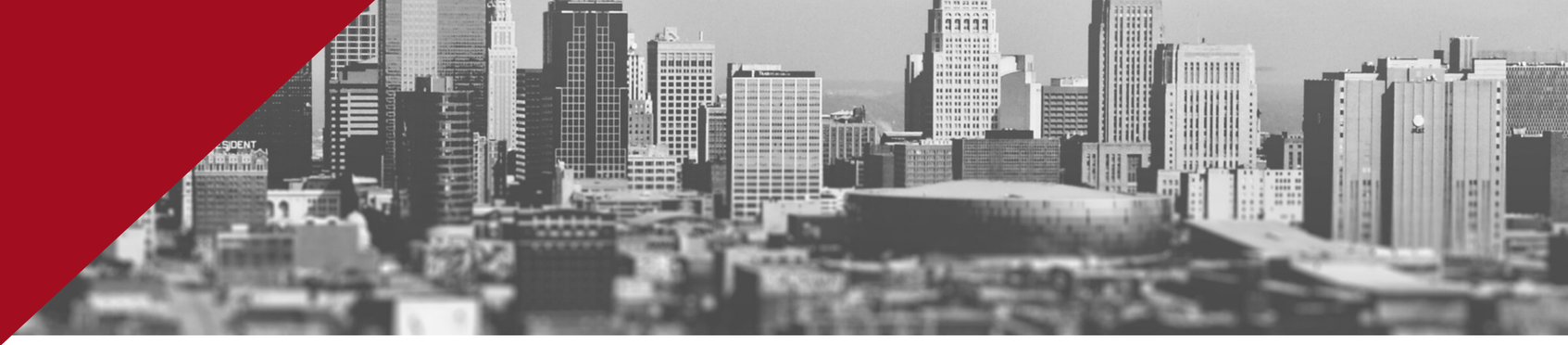
Analysis Breakdown



Case 1 : Job Data

1	**Number of jobs reviewed:** Amount of jobs reviewed over time.
2	**Throughput:** It is the no. of events happening per second
3	**Percentage share of each language:** Share of each language for different contents.
4	**Duplicate rows:** Rows that have the same value present in them.

Analysis Breakdown



Case 2: Investigating metric spike

1	**User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.
2	**User Growth:** Amount of users growing over time for a product
3	**Weekly Retention:** Users getting retained weekly after signing-up for a product
4	**Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.
5	**Email Engagement:** Users engage with the email service.

About the Tech Stack

- in the following task, the data sets used were from Trainity for the analysis of the given case study

- It is an Open-source database system for storing, managing, and retrieving data using SQL. Ideal for web apps, with security, scalability, and high performance.

Case Study 1

Job Data

Number of jobs reviewed:

Amount of jobs reviewed over time for the number of jobs reviewed per hour per day for November 2020

Query

```
SELECT
COUNT(job_id)/(30*24) AS no_of_jobs_reviewed_per_day_non_distinct
FROM task3.table_1;
```

Output

no_of_jobs_reviewed_per_day_non_distinct
0.0111

Job Data

Number of jobs reviewed:

Amount of jobs reviewed over time for the number of jobs reviewed per hour per day for November 2020

Query

```
SELECT
COUNT(DISTINCT job_id)/(30*24) AS no_of_jobs_reviewed_per_day_distinct
FROM task3.table_1;
```

Output

no_of_jobs_reviewed_per_day_distinct
0.0083

Job Data

Number of events happening per second.

It is the no. of events happening per second.

Query

```
SELECT ds as date_of_review, jobs_reviewed, AVG(jobs_reviewed)
OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS
throughput_7_rolling_average
FROM
(
SELECT ds, COUNT( DISTINCT job_id) AS jobs_reviewed
FROM job_data
GROUP BY ds ORDER BY ds
) a;
```

Output

date_of_review	jobs_reviewed	throughput_7_rolling_average
25-11-2020	1	1
26-11-2020	1	1
27-11-2020	1	1
28-11-2020	2	1.25
29-11-2020	1	1.2
30-11-2020	2	1.3333

Job Data

Percentage share of each language

Share of each language for different contents

Query

```
SELECT ds as date_of_review, jobs_reviewed, AVG(jobs_reviewed)
OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS
throughput_7_rolling_average_non_distinct_job_id
FROM
(
SELECT ds, COUNT(job_id) AS jobs_reviewed
FROM job_data
GROUP BY ds ORDER BY ds
) a;
```

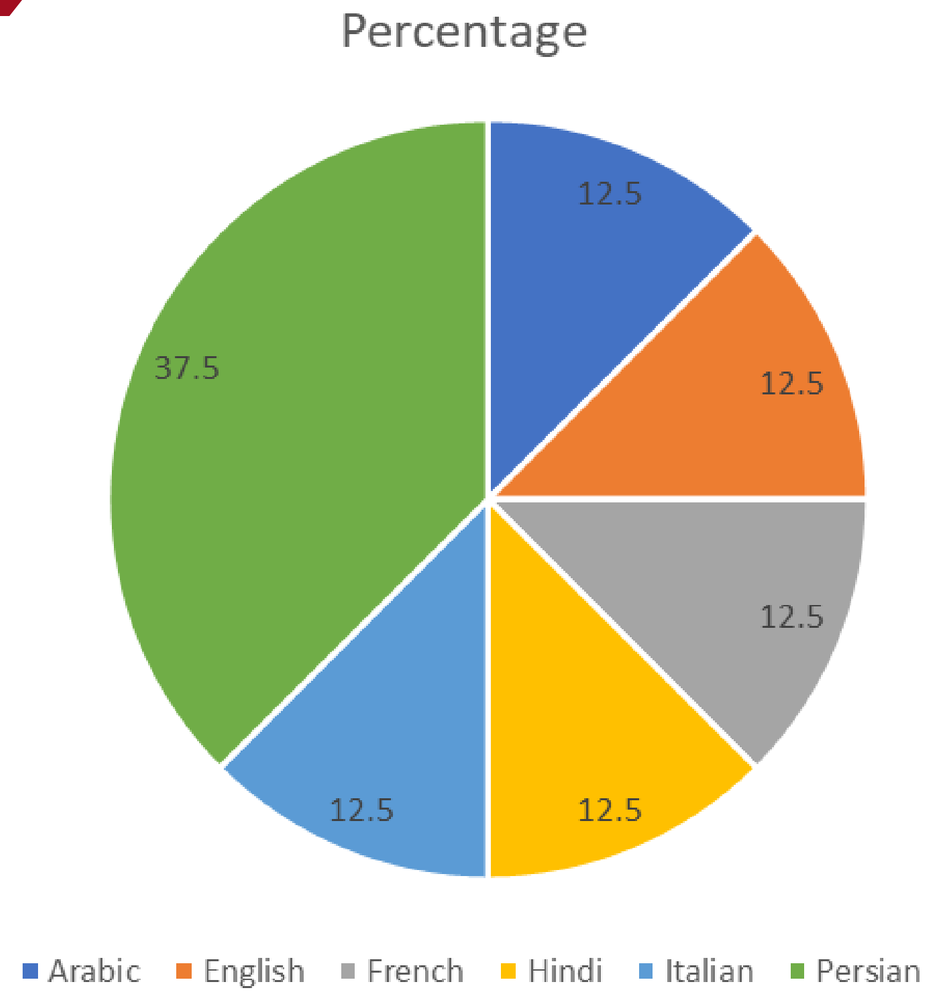
Output

job_id	language	total_of_each_language	percentage_share_of_each_language
21	English	1	12.5
22	Arabic	1	12.5
23	Persian	3	37.5
25	Hindi	1	12.5
11	French	1	12.5
20	Italian	1	12.5

Job Data

Chart

Percentage share of each language



Duplicate Rows

Rows that have the same value present in them.

Query

```
SELECT *
FROM
(
SELECT *, ROW_NUMBER()OVER(PARTITION BY job_id) AS row_num
FROM job_data
) a
WHERE row_num>1;
```

Output

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

Job Data

User Engagement

To measure the activeness of a user. Measuring if the user finds quality in a product/service.

Query

```
SELECT
extract (week from occurred_at) as week_number,
count(distinct user_id) as number_of_users
FROM
tutorial.yammer_events
group by
week_number;
```

Output

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

Investigating Metric Spike

User Engagement

To measure the activeness of a user. Measuring if the user finds quality in a product/service.

Investigating Metric Spike

User Growth

Amount of users growing over time for a product. Your task: Calculate the user growth for product? User Growth = Number of active users per week

Query

```
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;
```

Investigating Metric Spike

User Growth

Amount of users growing over time for a product. Your task: Calculate the user growth for product? User Growth = Number of active users per week

Output

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

User Growth

Amount of users growing over time for a product. Your task: Calculate the user growth for product? User Growth = Number of active users per week

Query

```
select count(*) from tutorial.yammer_users
where state = 'active';
```

Output

count
9381

Weekly Retention

Users getting retained weekly after signing-up for a product. Your task: Calculate the weekly retention of users-sign up cohort?

Query

```
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 tutorial.yammer_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 tutorial.yammer_events
where event_type = 'engagement'
)b
on a.user_id = b.user_id
)
)d
group by user_id
order by user_id
```

Output : Link

Investigating Metric Spike

Weekly Retention

Users getting retained weekly after signing-up for a product. Your task: Calculate the weekly retention of users-sign up cohort?

Query

```
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 tutorial.yammer_events
WHERE event_type = 'signup_flow'
and event_name = 'complete_signup'
and extract(week from occurred_at) = 18
)a
LEFT JOIN
(SELECT distinct user_id, extract (week from occurred_at) as engagement_week FROM tutorial.yammer_events
where event_type = 'engagement'
)b
on a.user_id = b.user_id
)
)d
group by user_id
order by user_id;
```

Output : Link

Investigating Metric Spike

Weekly Retention

To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly. Your task: Calculate the weekly engagement per device?

Query

```
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
tutorial.yammer_events
where event_type = 'engagement'
GROUP by 1,2,3
order by 1,2,3;
```

Output : Link

Investigating Metric Spike

Email Engagement

Users engaging with the email service. Your task: Calculate the email engagement metrics?

Query

```
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;
```

Output : Link

Investigating Metric Spike

Conclusion

Hence, all the questions given as part of Trainity Data Analytics Trainee Task 3 : Operation Analytics and Investigating Metric Spike have been provided with answers along with graphs.

In this task all the basic as well as advanced concepts related to SQL in Data Analytics have been implemented using the MySQL workbench 8.0 C