Operation Analytics and Investigating Metric Spike

PROJECT DESCRIPTION:

- The analysis for the complete end to end operations of a company.
- By analyzing data, the company then finds the areas on which it must improve upon
- Help other departments to derive insights out of the data they collect.
- Predict the overall growth or decline of a company’s fortune.

APPROACH:

- Get the information from given description about data and understand the problem.
- Go through the row data and understand the variable and attribute as given.
- Used MySQL Workbench to import files in new database and started writing my queries as required to achieve the desired result.
- Executed the Queries to find the output if any errors occurred in the code, the code is modified and fixed the error.

TECH STACK USED:

- MY SQL: To create database and generate queries.
- MICROSOFT OFFICE: To prepare presentation in MS POWERPOINT, temporarily store queries in MS WORD, open dataset files in MS EXCEL
- GOOGLE DRIVE: To upload the prepared presentation.

CASE STUDY :1 (JOB DATA)

A. **Number of jobs reviewed:** Amount of jobs reviewed over time.
Your task: Calculate the number of jobs reviewed per hour per day for November 2020?

Query:

```
SELECT ds, ROUND(1.0*COUNT(job_id)*3600/sum(time_spent),2) as Jobs_reviewed_Per_Hour
from job_data
where event IN('Transfer', 'Decision') AND ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY ds ;
```

Output:

ds	Jobs_reviewed_Per_Hour
2020-11-30	144.00
2020-11-29	180.00
2020-11-28	218.18
2020-11-27	34.62
2020-11-25	80.00

B. Throughput: It is the no. of events happening per second.
Your task: Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

Query:;

```
WITH x AS
(SELECT ds, COUNT(job_id) AS num_jobs, sum(time_spent) as total_time from job_data
where event IN('Transfer', 'Decision') AND ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY ds)
SELECT ds, ROUND(1.0*sum(num_jobs) OVER (ORDER BY ds rows between 6 preceding AND
current row)/sum(total_time) over (order by ds rows between 6 preceding AND current
row),2) as throughput_7d from x;
```

Output:

	ds	throughput_7d
▶	2020-11-25	0.02
	2020-11-27	0.01
	2020-11-28	0.02
	2020-11-29	0.02
	2020-11-30	0.03

C. Percentage share of each language: Share of each language for different contents.
Your task: Calculate the percentage share of each language in the last 30 days?

Query:
```
select language, count(*)*12.5 as Percentage_Share from job_data where ds between '2020-11-01' and '2020-11-30' group by language;
```

Output:

language	Percentage_Share
English	12.5
Arabic	12.5
Persian	37.5
Hindi	12.5
French	12.5
Italian	12.5

D. **Duplicate rows:** Rows that have the same value present in them.
Your task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

Query:

```
select * from
(select *,row_number()over(partition by job_id) as rownum from job_data)a
where rownum>1;
```

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

CASE STUDY :2 (INVESTIGATING METRIC SPIKE)

A. **User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.
Your task: Calculate the weekly user engagement?

Query:

```
SELECT EXTRACT(WEEK FROM a.occurred_at) AS week_num,
COUNT(DISTINCT a.user_id) AS users_engaged
FROM events a
GROUP BY week_num;
```

Output:

week_num	users_engaged
17	663
18	1068
19	1113
20	1154
21	1052
22	1051
23	1046
24	1060
25	1033
26	1035

week_num	users_engaged
27	1107
28	1074
29	1095
30	1169
31	996
32	949
33	905
34	900
35	45

B. User Growth: Amount of users growing over time for a product.
Your task: Calculate the user growth for product?

Query:

```
SELECT
year,week_num,active_users,
SUM(active_users) OVER (ORDER BY year,week_num ROWS BETWEEN UNBOUNDED PRECEDING AND
CURRENT ROW) AS total_active_users
FROM(SELECT EXTRACT(YEAR FROM u.activated_at) AS year,
EXTRACT(WEEK FROM u.activated_at) AS week_num,
COUNT(DISTINCT u.user_id) AS active_users FROM users u
WHERE state='active'
GROUP BY year,week_num
ORDER BY year,week_num
)a
```

Output:

year	week_num	active_users	total_active_users
2013	0	23	23
2013	1	30	53
2013	2	48	101
2013	3	36	137
2013	4	30	167
2013	5	48	215
2013	6	38	253
2013	7	42	295
2013	8	34	329
2013	9	43	372
2013	10	32	404
2013	11	31	435
2013	12	33	468
2013	13	39	507
2013	14	35	542
2013	15	43	585

C. 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 a.user_id, a.signup_week, b.engagement_week, b.engagement_week-a.signup_week as
retention_weekFROM((SELECT DISTINCT user_id, EXTRACT(WEEK FROM occurred_at) AS
signup_weekFROM events WHERE event_type = 'signup_flow'AND event_name =
'complete_signup'AND EXTRACT(WEEK FROM occurred_at) = 18) aLEFT JOIN(SELECT DISTINCT
user_id, EXTRACT(WEEK FROM occurred_at) AS engagement_weekFROM eventsWHERE event_type =
'engagement') b ON a.user_id = b.user_id)ORDER BY a.user_id;
```

Output:

user_id	signup_week	engagement_week	retention_week
11961	18	18	0
11962	18	18	0
11963	18	18	0
11971	18	18	0
11972	18	18	0
11973	18	18	0
11975	18	18	0
11981	18	18	0
11984	18	18	0
11986	18	18	0
11986	18	19	1
11990	18	18	0

D. Weekly Engagement: 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,
    EXTRACT(WEEK FROM occurred_at) AS week, device,
    COUNT(DISTINCT user_id) AS Count
    FROM events
    WHERE event_type = 'engagement'
    GROUP BY year,week,device
    ORDER BY year,week,device;
```

year	week	device	Count
2014	17	acer aspire desktop	9
2014	17	acer aspire notebook	20
2014	17	amazon fire phone	4
2014	17	asus chromebook	21
2014	17	dell inspiron desktop	18
2014	17	dell inspiron notebook	46
2014	17	hp pavilion desktop	14
2014	17	htc one	16
2014	17	ipad air	27
2014	17	ipad mini	19
2014	17	iphone 4s	21
2014	17	iphone 5	65
2014	17	iphone 5s	42
2014	17	kindle fire	6
2014	17	lenovo thinkpad	86

Output

E. 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_open' THEN 1 ELSE 0 END)/SUM(CASE WHEN
email_cat='email_sent' THEN 1 ELSE 0 END) AS email_open_rate,
100.0*SUM(CASE WHEN email_cat='email_click' THEN 1 ELSE 0 END)/SUM(CASE WHEN
email_cat='email_sent' THEN 1 ELSE 0 END) AS email_clicked_rate
FROM
(SELECT *,CASE WHEN action IN ('sent_weekly_digest','sent_reengagement_email') THEN 'email_sent’
WHEN action IN ('email_open') THEN 'email_open’
WHEN action in ('email_clickthrough') THEN 'email_click’
END AS email_cat
FROM email_events
)a
```

Output:

	email_open_rate	email_clicked_rate
▶	33.58339	14.78989