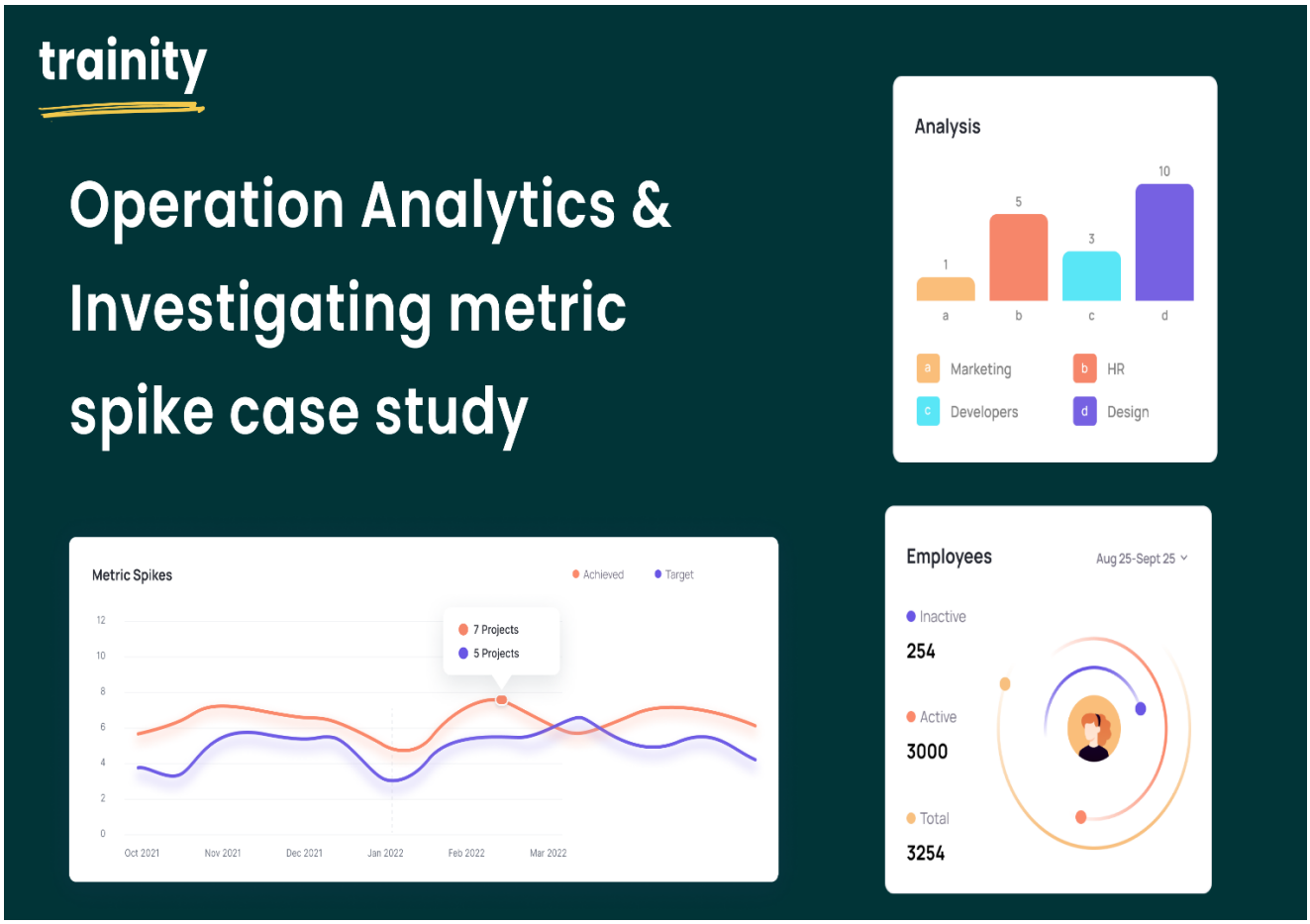
Operation Analytics and Investigating Metric Spike

Advanced SQL



Project Description: -

"Operation Analytics involves thoroughly examining a company's entire operations. By doing this, the company identifies areas that need improvement. I collaborate closely with various teams like operations, support, and marketing, assisting them in uncovering valuable insights from the data they gather.

This analysis plays a crucial role in predicting the overall growth or decline of a company's fortunes. It leads to improved automation, better understanding between different teams, and more efficient workflows.

As a Data Analyst Lead at a company like Microsoft, I work on intriguing projects. One such project involves Job Data Analysis, where I analyze different datasets and tables to provide insights and answer questions from various departments.

Another exciting project is Investigating Metric Spikes. In this, I delve into questions like why there's a dip in daily engagement or why sales have taken a hit. Addressing these questions daily is essential, requiring a thorough investigation into metric spikes.

Overall, my role involves working with diverse data sets, collaborating with teams, and deriving meaningful insight.

Project Approach: -

"At the start, I took some time to get a good grip on the data and tables provided. I clarified questions in my mind, like understanding what job_id, actor_id, and event really meant, and considered the key aspects while reviewing the data. Using SQL, I dug into the dataset given by the management team.

This project is developed using SQL Workbench. First, I need to create database by using dataset file which was provided by the company. Next step loads the data into SQL Workbench then performed analysis and find the information that will the help the operations team**,** support team**,** marketing team, etc to understand questions like – Why is there a dip in daily engagement? Why have sales taken a dip.Questions like these must be answered daily and for that it’s very important to investigate metric spike.”

Tech-Stack Used: -

"I have used MySQL Workbench v8.0 - Renowned for its versatility it also stands as a visual editor seamlessly integrating data modelling where developers can build, edit, and run queries. It allows us to preview our changes before applying them, SQL development, and database administration within a single, user-friendly interface. MySQL Workbench also comes with a database administration suite. This makes it easy for us to audit our databases, configure servers, and view logs. It also shows us the Performance Monitoring."

Insights: -

Case Study 1 – Job Data Analysis

a. **Number of Jobs Reviewed** - Jobs Reviewed Over Time

My Task - Calculate the number of jobs reviewed per hour for each day in November 2020?

Answer - On date 28th November 2020 there is maximum number of jobs reviewed is **218**. Likewise, on 29th and 30th of November 2020 Jobs reviewed were **180** and on 25th Nov 2020 it was **80** and on 26th Nov 2020 it was **64** and on 27th Nov 2020 it was **35**.

	Dates	Jobs Reviewed per Hour per Day
▶	2020-11-30	180
	2020-11-29	180
	2020-11-28	218
	2020-11-27	35
	2020-11-26	64
	2020-11-25	80

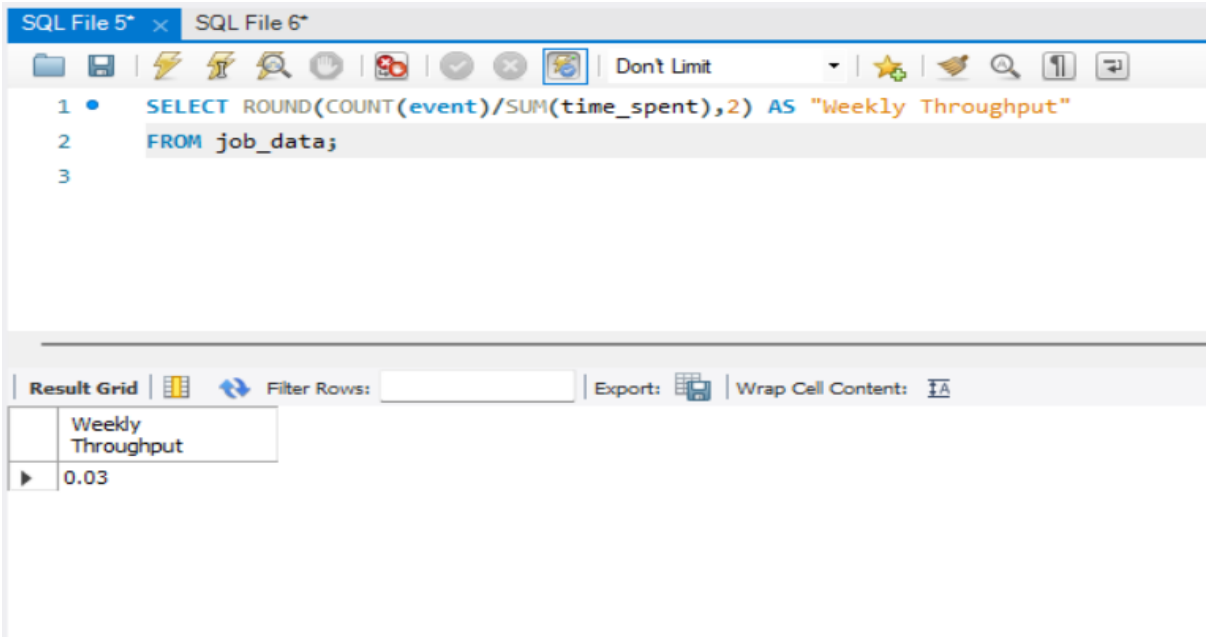
SQL Query: -

```
SELECT ds AS Dates, ROUND((COUNT(job_id)/SUM(time_Spent))*3600)
AS "Jobs reviewed per Hour per Day"

FROM job_data

WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'

GROUP BY ds;
```

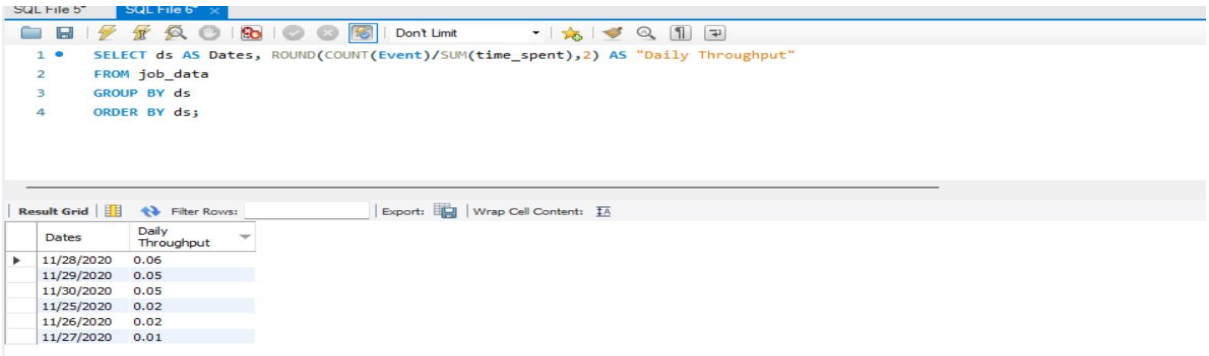
b. **Throughput Analysis**: - It is the no. of events happening per second.

My Task: - 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.

The Weekly throughput is **0.03**



On date 2020-11-28 the throughput is highest **0.06**.

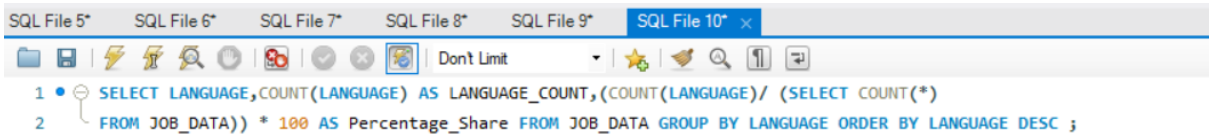


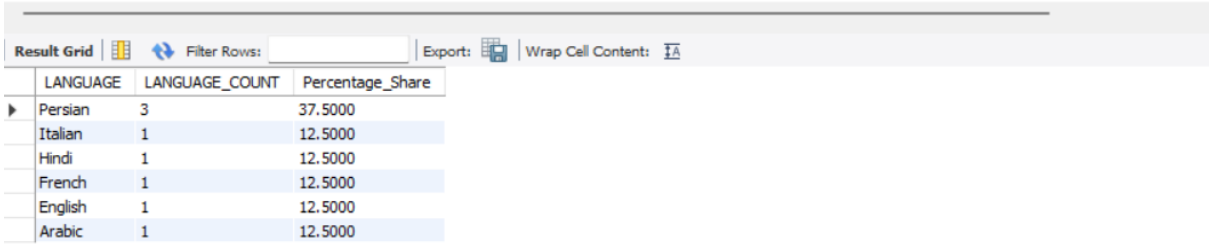
Metric values fluctuate regularly on both a weekly and daily basis. If you need quicker insights, you can obtain numbers daily or even by the minute.

Consequently, rolling metrics excel at illustrating whether your metrics are exhibiting an upward or downward trend on a daily level.

C. **Language Share Analysis: -** To Calculate the percentage share of each language in the last 30 days.

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



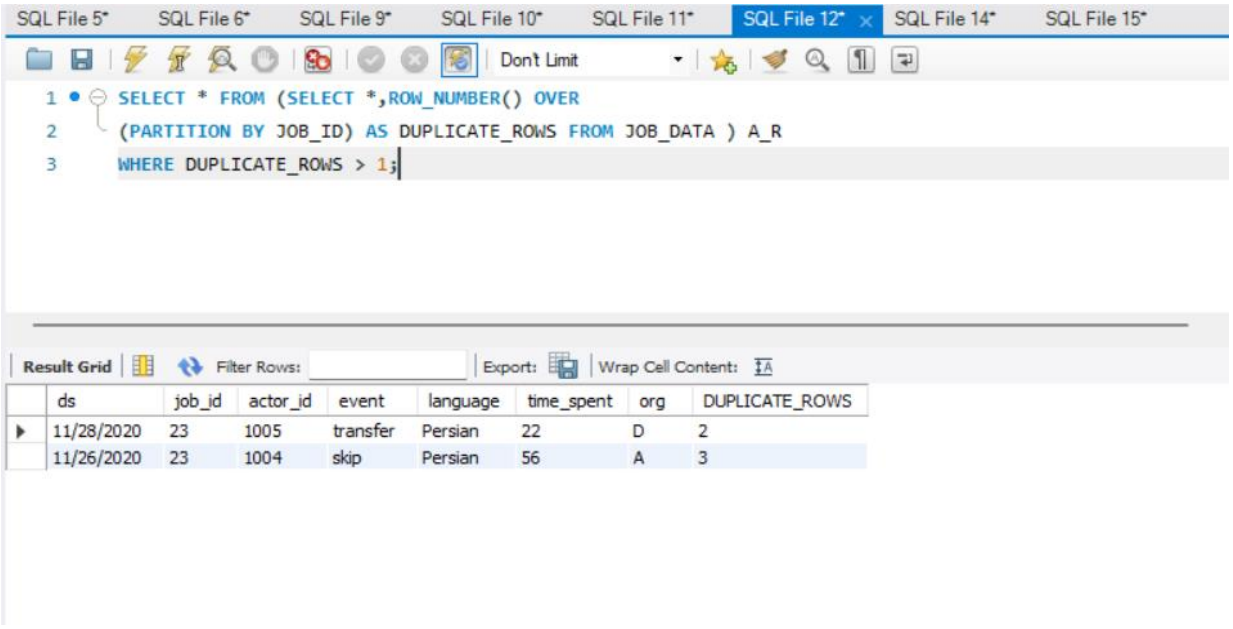


LANGUAGE	LANGUAGE_COUNT	Percentage_Share
Persian	3	37.5000
Italian	1	12.5000
Hindi	1	12.5000
French	1	12.5000
English	1	12.5000
Arabic	1	12.5000

The **Persian** language has the highest percentage share **37.5%. Italian, Hindi, French, English,** and **Arabic** are having the same percentage share at **12.5%.**

d. **Duplicate Rows Detection: -** Identify duplicate rows in the data

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



These rows are having the Duplicate values.

Case Study 2: - Investigating Metric Spike

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

   My Task: - To calculate the weekly user engagement.

Query: -

```
SELECT EXTRACT(WEEK FROM occurred_at) AS “Week Numbers”, COUNT(DISTINCT user_id)
AS “Weekly Active Users”

FROM events

WHERE event_type = ‘engagement’

GROUP BY 1;
```

Week Numbers	Weekly Active Users
17	663
18	1068
19	1113
20	1154
21	1121
22	1186
23	1232
24	1275
25	1264
26	1302
27	1372
28	1365
29	1376
30	1467
31	1299
32	1225
33	1225
34	1204
35	104

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

My Task: - To calculate the user growth for the product.

SQL Query: -

```
SELECT Months, Users, ROUND(((Users/LAG(Users, 1) OVER (ORDER BY Mo
nths) –1)*100), 2) AS “Growth in %”

FROM

(

SELECT EXTRACT(MONTH FROM created_at) AS Months,
COUNT(activated_at) AS Users

FROM users

WHERE activated_at NOT IN(“”)

GROUP BY 1

ORDER BY 1)

) sub;
```

	Months	Users	Growth in %
▶	1	712	NULL
	2	685	-3.79
	3	765	11.68
	4	907	18.56
	5	993	9.48
	6	1086	9.37
	7	1281	17.96
	8	1347	5.15
	9	330	-75.50
	10	390	18.18
	11	399	2.31
	12	486	21.80

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

   My Task: - To calculate the weekly retention of users based on their sign-up cohort.

   SQL Query: -

```
SELECT first AS “Week Numbers”,

SUM(CASE WHEN week_number = 0 THEN 1 ELSE 0 END) AS “Week 0”,

SUM(CASE WHEN week_number = 1 THEN 1 ELSE 0 END) AS “Week 1”,

SUM(CASE WHEN week_number = 2 THEN 1 ELSE 0 END) AS “Week 2”,

SUM(CASE WHEN week_number = 3 THEN 1 ELSE 0 END) AS “Week 3”,

SUM(CASE WHEN week_number = 4 THEN 1 ELSE 0 END) AS “Week 4”,

SUM(CASE WHEN week_number = 5 THEN 1 ELSE 0 END) AS “Week 5”,

SUM(CASE WHEN week_number = 6 THEN 1 ELSE 0 END) AS “Week 6”,

SUM(CASE WHEN week_number = 7 THEN 1 ELSE 0 END) AS “Week 7”,

SUM(CASE WHEN week_number = 8 THEN 1 ELSE 0 END) AS “Week 8”,

SUM(CASE WHEN week_number = 9 THEN 1 ELSE 0 END) AS “Week 9”,

SUM(CASE WHEN week_number = 10 THEN 1 ELSE 0 END) AS “Week 10”,

SUM(CASE WHEN week_number = 11 THEN 1 ELSE 0 END) AS “Week 11”,

SUM(CASE WHEN week_number = 12 THEN 1 ELSE 0 END) AS “Week 12”,

SUM(CASE WHEN week_number = 13 THEN 1 ELSE 0 END) AS “Week 13”,

SUM(CASE WHEN week_number = 14 THEN 1 ELSE 0 END) AS “Week 14”,
```

```
SUM(CASE WHEN week_number = 15 THEN 1 ELSE 0 END) AS “Week 15”,

SUM(CASE WHEN week_number = 16 THEN 1 ELSE 0 END) AS “Week 16”,

SUM(CASE WHEN week_number = 17 THEN 1 ELSE 0 END) AS “Week 17”,

SUM(CASE WHEN week_number = 18 THEN 1 ELSE 0 END) AS “Week
18”, FROM

(

SELECT m.user_id, m.login_week, n.first, m.login_week – first AS
week_number FROM

(SELECT user_id, EXTRACT(WEEK FROM occurred_at) AS login_week FROM
events GROUP BY 1, 2) m,

(SELECT user_id, MIN(EXTRACT(WEEK FROM occurred_at)) AS first FROM ev
ents GROUP BY 1) n

WHERE m.user_id = n.user_id

) sub

GROUP BY first ORDER BY first;
```

Week Numbers	Week 0	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18
17	740	472	324	251	205	187	167	146	145	145	136	131	132	143	116	91	82	77	5
18	788	362	261	203	168	147	144	127	113	122	106	118	127	110	97	85	67	4	0
19	601	284	173	153	114	95	91	81	95	82	68	65	63	42	51	49	2	0	0
20	555	223	165	121	91	72	63	67	63	65	67	41	40	33	40	0	0	0	0
21	495	187	131	91	74	63	75	72	58	48	45	39	35	28	2	0	0	0	0
22	521	224	150	107	87	73	63	60	55	48	41	39	31	1	0	0	0	0	0
23	542	219	138	101	90	79	69	61	54	47	35	30	0	0	0	0	0	0	0
24	535	205	143	102	81	63	65	61	38	39	29	0	0	0	0	0	0	0	0
25	500	218	139	101	75	63	50	46	38	35	2	0	0	0	0	0	0	0	0
26	495	181	114	83	73	55	47	43	29	0	0	0	0	0	0	0	0	0	0
27	493	199	121	106	68	53	40	36	1	0	0	0	0	0	0	0	0	0	0
28	486	194	114	69	46	30	28	3	0	0	0	0	0	0	0	0	0	0	0
29	501	186	102	65	47	40	1	0	0	0	0	0	0	0	0	0	0	0	0
30	533	202	121	78	53	3	0	0	0	0	0	0	0	0	0	0	0	0	0
31	430	145	76	57	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	496	188	94	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	499	202	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	518	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

My Task: - To calculate the weekly engagement per device.

SQL Query: -

```
SELECT EXTRACT(WEEK FROM occurred_at) AS “Week Numbers”,

COUNT(DISTINCT CASE WHEN device IN(‘dell inspiron notebook’) THEN use
r_id ELSE NULL END) AS “Dell Inspiron Notebook”,

COUNT(DISTINCT CASE WHEN device IN(‘iphone 5’) THEN user_id ELSE NUL
L END) AS

“iPhone 5”,

COUNT(DISTINCT CASE WHEN device IN(‘iPhone 4s’)
THEN user_id ELSE NULL END) AS

“iPhone 4S”,

COUNT(DISTINCT CASE WHEN device IN(‘windows surface’) THEN user_id E
LSE NULL END) AS “Windows Surface”,

COUNT(DISTINCT CASE WHEN device IN(‘macbook air’) THEN user_id ELSE
NULL END) AS “Macbook Air”,

COUNT(DISTINCT CASE WHEN device IN(‘iPhone 5s’)
THEN user_id ELSE NULL END) AS

“iPhone 5S”,

COUNT(DISTINCT CASE WHEN device IN(‘macbook pro’) THEN user_id ELSE
NULL END) AS “Macbook Pro”,

COUNT(DISTINCT CASE WHEN device IN(‘kindle fire’) THEN user_id ELSE NU
LL END) AS “Kindle Fire”,
```

```
COUNT(DISTINCT CASE WHEN device IN('iPad mini') THEN user_id ELSE
NULL END) AS "iPad Mini",

COUNT(DISTINCT CASE WHEN device IN('nexus 7') THEN user_id ELSE NULL
 END) AS

"Nexus 7",

COUNT(DISTINCT CASE WHEN device IN('nexus 5') THEN user_id ELSE NULL
 END) AS

"Nexus 5",

COUNT(DISTINCT CASE WHEN device IN('samsung galaxy s4') THEN user_id
 ELSE NULL END) AS "Samsung Galaxy S4",

COUNT(DISTINCT CASE WHEN device IN('lenovo thinkpad') THEN user_id E
LSE NULL END) AS "Lenovo Thinkpad",

COUNT(DISTINCT CASE WHEN device IN('Samsung galaxy
tablet') THEN user_id ELSE NULL END) AS "Samsung Galaxy Tablet",

COUNT(DISTINCT CASE WHEN device IN('acer aspire notebook') THEN user
_id ELSE NULL END) AS "Acer Aspire Notebook",

COUNT(DISTINCT CASE WHEN device IN('asus chromebook') THEN user_id
ELSE NULL END) AS "Asus Chromebook",

COUNT(DISTINCT CASE WHEN device IN('htc one') THEN user_id ELSE NULL
 END) AS "HTC One",

COUNT(DISTINCT CASE WHEN device IN('Nokia lumia 635') THEN
user_id ELSE NULL END) AS "Nokia Lumia 635",

COUNT(DISTINCT CASE WHEN device IN('samsung galaxy note') THEN user
_id ELSE NULL END) AS "Samsung Galaxy Note",

COUNT(DISTINCT CASE WHEN device IN('acer aspire desktop') THEN user_i
d ELSE NULL END) AS "Acer Aspire Desktop",
```

```
COUNT(DISTINCT CASE WHEN device IN('mac mini') THEN user_id ELSE NU
LL END) AS "Mac Mini",

COUNT(DISTINCT CASE WHEN device IN('hp pavilion
desktop') THEN user_id ELSE NULL END) AS "HP Pavilion Desktop",

COUNT(DISTINCT CASE WHEN device IN('dell inspiron desktop') THEN user
_id ELSE NULL END) AS "Dell Inspiron Desktop",

COUNT(DISTINCT CASE WHEN device IN('ipad air') THEN user_id ELSE NULL
 END) AS "iPad Air",

COUNT(DISTINCT CASE WHEN device IN('amazon fire phone') THEN user_id
 ELSE NULL END) AS "Amazon Fire Phone",

COUNT(DISTINCT CASE WHEN device IN('nexus 10') THEN user_id ELSE NU
LL END) AS

"Nexus 10" FROM events

WHERE event_type = 'engagement' GROUP BY 1

ORDER BY 1;
```

Week Numbers	Dell Inspiron Notebook	iPhone 5	iPhone 4S	Windows Surface	Macbook Air	iPhone 5S	Macbook Pro	Kindle Fire	iPad Mini	Nexus 7	Nexus 5	Samsung Galaxy S4	Lenovo Thinkpad	Samsung Galaxy Tablet
17	46	65	21	10	54	42	143	6	19	18	40	52	86	8
18	77	113	46	10	121	73	252	27	30	30	73	82	153	11
19	83	115	44	16	112	79	266	21	36	41	87	91	178	6
20	84	125	55	21	119	79	256	23	32	32	103	93	173	9
21	80	137	45	17	110	74	247	30	23	29	91	84	167	6
22	92	125	45	15	145	71	251	21	34	45	96	105	176	10
23	103	152	53	14	124	79	266	25	33	36	88	99	176	14
24	99	142	53	22	152	79	255	25	39	49	87	101	165	11
25	105	137	40	22	121	78	275	24	30	51	89	99	197	12
26	89	152	50	21	134	94	269	26	43	46	87	112	192	12
27	89	163	67	33	142	83	302	25	35	40	84	116	202	15
28	103	151	61	33	148	93	295	31	35	39	85	122	220	9
29	113	144	60	28	148	90	295	37	34	45	77	123	209	13
30	127	152	65	19	159	103	322	25	35	62	84	103	206	9
31	113	135	56	19	147	71	321	14	27	38	69	100	207	8
32	104	119	34	10	125	67	307	12	30	25	67	82	179	6
33	110	110	35	15	133	65	312	14	28	30	70	80	191	12
34	105	101	50	18	136	70	292	13	25	33	70	90	193	14
35	9	2	6	3	10	3	17	3	2	2	4	6	16	0

Week Numbers	Acer Aspire Notebook	Asus Chromebook	HTC One	Nokia Lumia 635	Samsung Galaxy Note	Acer Aspire Desktop	Mac Mini	HP Pavilion Desktop	Dell Inspiron Desktop	iPad Air	Amazon Fire Phone	Nexus 10
17	20	21	16	17	7	9	6	14	18	27	4	16
18	33	42	19	33	15	26	13	37	58	52	9	30
19	41	27	30	23	11	23	18	40	36	55	12	25
20	40	41	29	22	18	23	26	30	52	59	11	22
21	47	38	21	25	20	29	18	44	41	51	5	25
22	41	52	24	25	19	25	25	38	52	58	5	27
23	43	49	20	31	14	22	18	54	53	41	16	45
24	40	43	20	35	20	24	29	56	59	57	11	38
25	47	38	21	37	14	28	21	52	52	57	13	29
26	35	49	23	42	9	29	11	46	60	56	13	29
27	49	52	27	31	15	29	15	56	53	55	10	37
28	49	50	26	35	10	30	28	56	56	54	6	26
29	53	49	31	43	16	28	31	58	54	52	12	25
30	60	56	31	34	15	33	23	42	54	70	12	36
31	55	56	13	28	14	31	24	51	44	55	14	24
32	55	62	18	28	12	35	20	51	57	48	12	30
33	46	49	19	27	13	39	32	38	37	40	14	23
34	63	47	25	17	13	30	30	36	49	39	11	25
35	3	6	2	2	1	1	2	1	1	0	0	2

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

   My Task: - To calculate the email engagement metrics.

   SQL Query: -

```
SELECT Week,

ROUND((weekly_digest/total*100),2) AS “Weekly Digest
Rate”, ROUND((email_opens/total*100),2) AS “Email Open
Rate”, ROUND((email_clickthroughs/total*100),2) AS “Email Clickthrough
Rate”, ROUND((reengagement_emails/total*100),2) AS “Reengagement
Email Rate”

FROM

(SELECT EXTRACT(WEEK FROM occurred_at) AS Week,

COUNT(CASE WHEN action = ‘sent_weekly_digest’ THEN user_id ELSE NULL
 END) AS weekly_digest,

COUNT(CASE WHEN

email_opens,

action = ’email_open’ THEN user_id   ELSE

NULL END)   AS

COUNT(CASE WHEN action = ’email_clickthrough’ THEN user_id ELSE NULL
END) AS email_clickthroughs,

COUNT(CASE WHEN action = ‘sent_reengagement_email’ THEN user_id ELS
E NULL END) AS reengagement_emails,

COUNT(user_id) AS total FROM email_events GROUP BY 1

) sub
```

GROUP BY 1

ORDER BY 1;

Week	Weekly Digest Rate	Email Open Rate	Email Clickthrough Rate	Reengagement Email Rate
17	62.32	21.28	11.39	5.01
18	63.45	22.24	10.49	3.83
19	62.16	22.67	11.13	4.04
20	61.62	22.64	11.43	4.31
21	63.52	22.82	9.97	3.69
22	63.59	21.56	10.66	4.19
23	62.39	22.34	11.18	4.09
24	61.61	22.92	10.99	4.48
25	63.77	21.79	10.54	3.90
26	62.99	22.22	10.61	4.18
27	62.24	22.49	11.37	3.90
28	62.92	22.48	10.77	3.83
29	63.98	21.71	10.51	3.79
30	62.29	23.24	10.59	3.88
31	65.27	23.25	7.66	3.82
32	66.59	22.85	7.14	3.42
33	64.73	23.10	7.91	4.26
34	64.33	23.91	7.67	4.08
35	0.00	32.28	29.92	37.80

Insights

Case Study 1 - (Job Data)

1. Analysing the data on job reviews in November 2020, we found that, on average, there were 83% more distinct jobs reviewed per hour each day.
2. Utilizing the 7-day rolling average of throughput provides a comprehensive average spanning from day 1 to day 7, unlike the daily metric, which only reflects the average for that specific day.
3. In the past 30 days, the percentage share of languages is notable. Persian holds the highest at 37.5%, while Italian, Hindi, French, English, and Arabic each share an equal portion at 12.5%.
4. Examining the data for duplicates, we found two identical rows when the data is partitioned by job_id. However, considering all columns collectively, each row is unique.

Insights

Case Study 2 - (Investigating Metric Spike)

1. In the case study, we observed that more people were using the product or service from the 18th to the 31st week. However, after that, the number of users started going down. This suggests that some users might not be satisfied with the quality of the product or service in the later weeks.
2. The user base for the product has grown steadily over time. From the 17th week of 2013 to the 35th week of 2014, there were a total of 9381 active users.
3. Looking at how users stick around after signing up for the product each week, we found that MacBook and iPhone users have the highest weekly engagement per device.
4. Looking at how users interact with the email service, we see that about 34% of them open the emails, and approximately 15% click on the content. This level of engagement is positive for the company and indicates potential for expansion.

RESULT

Through this project, I gained proficiency in implementing advanced SQL concepts, including Windows Functions. This experience provided valuable insights into the workings of the real-world industry, enhancing my mastery of SQL concepts. I developed the skill of posing relevant questions tailored to specific situations, determining which columns to analyse from provided data, and extracting insights crucial for business growth. Additionally, I learned how companies identify and enhance various aspects related to their operations. Exploring metric spikes, understanding both surges and dips, became a part of my acquired knowledge.