PROJECT TITLE

OPERATION ANALYTICS

AND

INVESTIGATING METRIC SPIKE

Table of contents

List of Tables

PROJECT DESCRIPTION

This project aims to leverage operation analytics and advanced SQL queries to derive valuable insights and provide data-driven solutions for a company, specifically focusing on various departments such as operations, support, marketing, and more.

Hypothetically, a situation is presented as part of internship at Trinity, I was assumed working as a Data Analyst Lead at a company similar to Microsoft, I have been entrusted with the responsibility of analysing different data sets and tables to address questions and challenges posed by these departments. The primary objective of this project is to conduct end-to-end analysis of the company's operations and identify areas for improvement. These insights will help the company make informed decisions, enhance automation processes, foster better cross-functional understanding, and optimize workflows for improved efficiency.

Given the importance of this project for the company's overall growth and success, the completion deadline is set at one week. Within this timeframe, will be expected to demonstrate the expertise in advanced SQL queries, data analysis techniques, and communication skills to deliver comprehensive insights and reports to the respective departments. Effective collaboration and timely delivery of accurate insights will be vital to ensure the project's success and enable the company to make informed decisions that positively impact its fortune.

Key Deliverables:

1. Conduct end-to-end operation analytics using advanced SQL queries.
2. Derive actionable insights from the provided data sets and tables.
3. Investigate metric spikes and provide detailed explanations for fluctuations.
4. Prepare comprehensive reports and presentations.
5. Deliver accurate and timely answers to the questions posed by different teams.
6. Propose recommendations for process improvements and optimizations.

APPROACH

Upon comprehending the requirements of both teams, the initial course of action involved accessing the data to check its consistency and acquiring familiarity with the diverse tables within the database. Understanding how these tables can be effectively utilized to address the queries posed by each team was a pivotal aspect of this step. It is of paramount importance to ensure that I am working with appropriate and accurate data sets. Failing to undertake this critical preliminary phase may yield undesirable outcomes and result in project delays. By ensuring that I am working with the correct database and have access to comprehensive data which results into completion of the project.

Timeline: The project is expected to be completed within one week. The following timeline is proposed:

- Day 1: Familiarize with the available data sets and tables, understand the objectives, and establish communication channels with relevant teams.

- Days 2-4: Conduct comprehensive analysis using advanced SQL queries, investigating metric spikes, and addressing daily questions from different departments.

- Day 5: Compile the findings, generate visualizations, and prepare the final report highlighting actionable insights.

- Day 6: Review and refine the report, ensuring accuracy and clarity of the presented information.

- Day 7: Finalize the project, deliver the report to stakeholders, and present the key findings to the relevant teams.

Tech Used

To obtain the necessary information required for the project or task, Oracle MySQL 8.0.33, a popular and widely used Relational Database Management System (RDBMS) was used to perform Structured Query Language (SQL) operations. Oracle MySQL 8.0.33 is a powerful and flexible RDBMS that provides users with various tools and functionalities to work with databases efficiently. In addition to using Oracle MySQL 8.0.33, an online database editor called "DB Fiddle" was also used. DB Fiddle is an online platform that allows users to execute SQL queries on various databases, including MySQL. Another, tools that are used to perform the analysis are Tableau and Ms-excel.

FINDINGS

CASE STUDY 1 (Operations analysis)

Finding 1: Number of Jobs Reviewed.

The task is to calculate the number of jobs reviewed per hour per day for November 2020. From the table job_data this insight has been derived. On November 28, 2020, there are maximum number of 218 job are reviewed.

Date	Jobs_per_day	time_spent_per_day_ in_secs	time_spent_per_day_in_ hours	Jobs_reviewed_per_day_ per_hour
2020-11-30	2	40	0.0111	180
2020-11-29	1	20	0.0056	180
2020-11-28	2	33	0.0092	218
2020-11-27	1	104	0.0289	35
2020-11-26	1	56	0.0156	64
2020-11-25	1	45	0.0125	80

Table 1:Number of jobs reviewed per hour per day.

The graphical representation of the above data is below. This representation is primarily between the different days in November and jobs reviewed per day per hour.
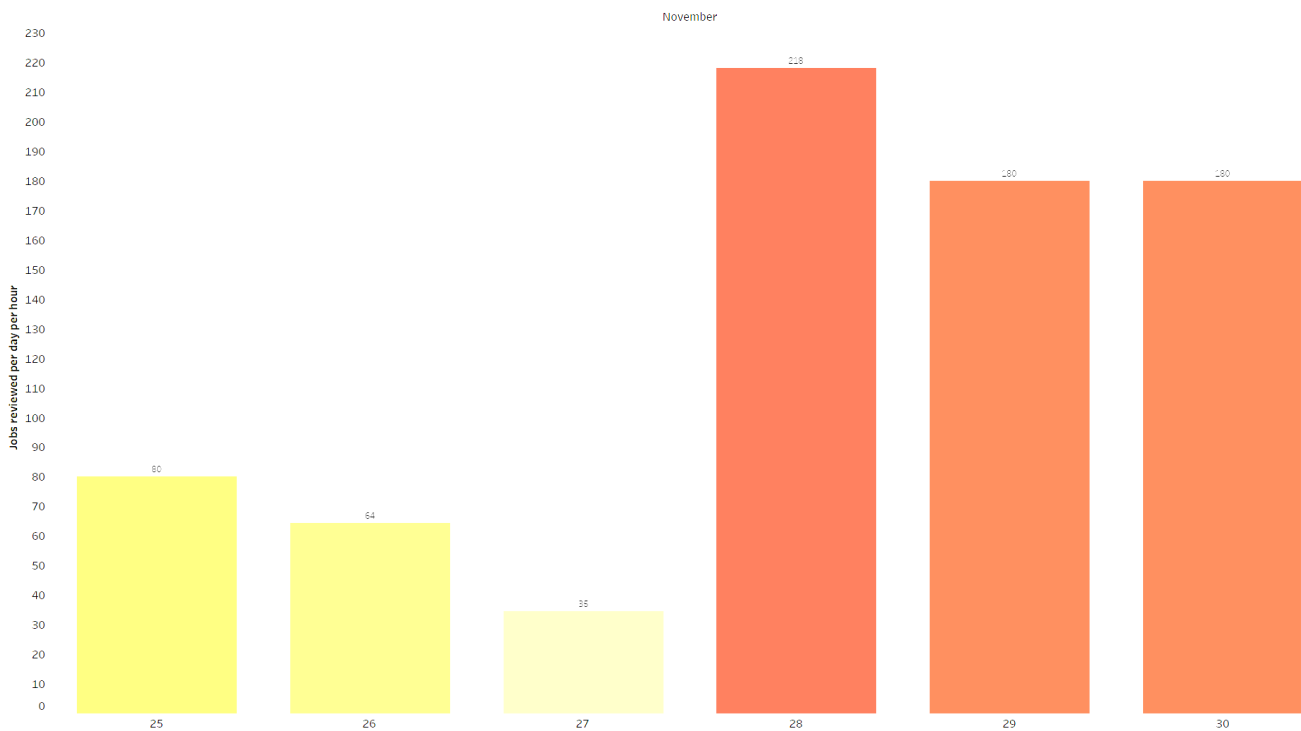
Finding 2: Throughput

Based on the data presented in the table, throughput refers to the rate at which events occur within a given time frame. The table contains various types of events, including transfers, skips, and decisions. Now, I have to examine the events happening per second and events per hour. The following table represent the findings.

Based on the analysis, we observe that the highest number of decision events occurred on November 28, 2020, with a frequency of 0.0909 events per second and 327 events per hour. Similarly, the most skipped events took place on November 30, 2020, at a rate of 0.0667 per second and 240 events per hour. Interestingly, the largest number of transfer events also happened on November 28, 2020.

Additionally, the team requires information on the daily moving average. The last column of the data represents the moving average, which helps in monitoring the daily targets.

date	Event_type	Total_events	Total_timespent_in_secs	Events_per_Second	Event_per_hour	Moving_average
2020-11-25	transfer	1	45	0.0222	80	80
2020-11-26	skip	1	56	0.0179	64	72
2020-11-27	decision	1	104	0.0096	35	49
2020-11-28	transfer	1	22	0.0455	164	99
2020-11-28	decision	1	11	0.0909	327	245
2020-11-29	decision	1	20	0.05	180	254
2020-11-30	skip	1	15	0.0667	240	210
2020-11-30	transfer	1	25	0.04	144	192

Table 2: Number of events happening per second and per hour.

Finding 3: Percentage Share of Each Language

The objective of this task is to determine the percentage share of each language used in the applications stored in the database. By calculating the share of each language, we can gain insights into the distribution of applicants across different languages. This information will be valuable in evaluating and understanding the applicant pool.

Persian language dominates the application pool with the highest share of 37.5%, indicating its popularity among applicants. On the other hand, the remaining languages have relatively similar shares, suggesting a more balanced distribution. This insight highlights the significance of Persian language and underscores the need for considering language-specific factors in applicant evaluation and selection processes.

CV_Language	Number_of_CV	Percnetage_Share(%)
Arabic	1	12.5
English	1	12.5
French	1	12.5
Hindi	1	12.5
Italian	1	12.5
Persian	3	37.5

Table 3: Percent share of different languages.

Finding 4: Duplicate Rows

The Job ID serves as the unique identifier for various activities performed by actors, including skip, transfer, and decision. By analyzing the application or Job identification number, three distinct outcomes are observed. Specifically, Job ID 23 underwent a review process involving three individuals. The first reviewer with ID 1004 skipped the application, while the second reviewer with ID 1005 transferred the application to actor ID 1003, who ultimately made the decision regarding the application.

Furthermore, it is worth noting that there are no duplicate rows present in the dataset, indicating that each entry is unique.

| Date	Unique_Job_ID	Actor_ID	Event_Type	Application_langauge
2020-11-26	23	1004	skip	Persian
2020-11-28	23	1005	transfer	Persian
2020-11-29	23	1003	decision	Persian

Table 4: Duplicate rows

CASE STUDY 2 (Investigating metric spike)

Finding 1: User Engagement

This finding pertains to the analysis of engagement levels on the application portal. It aims to measure the level of user activity and assess the extent to which users find value in the product or service being offered. This information is crucial for understanding user satisfaction, identifying areas for improvement, and enhancing the overall user experience. In 30th week, there was 1,467 maximum number activity happened on the platform whereas least activity happened in 35th week.

Week_of_year	Week_start_date	Week_end_date	Users_Logged
17	2014-05-01	2014-05-03	663
18	2014-05-04	2014-05-10	1068
19	2014-05-11	2014-05-17	1113
20	2014-05-18	2014-05-24	1154
21	2014-05-25	2014-05-31	1121
22	2014-06-01	2014-06-07	1186
23	2014-06-08	2014-06-14	1232
24	2014-06-15	2014-06-21	1275
25	2014-06-22	2014-06-28	1264
26	2014-06-29	2014-07-05	1302
27	2014-07-06	2014-07-12	1372
28	2014-07-13	2014-07-19	1365
29	2014-07-20	2014-07-26	1376
30	**2014-07-27**	**2014-08-02**	**1467**
31	2014-08-03	2014-08-09	1299
32	2014-08-10	2014-08-16	1225
33	2014-08-17	2014-08-23	1225
34	2014-08-24	2014-08-30	1204
35	**2014-08-31**	**2014-08-31**	**104**

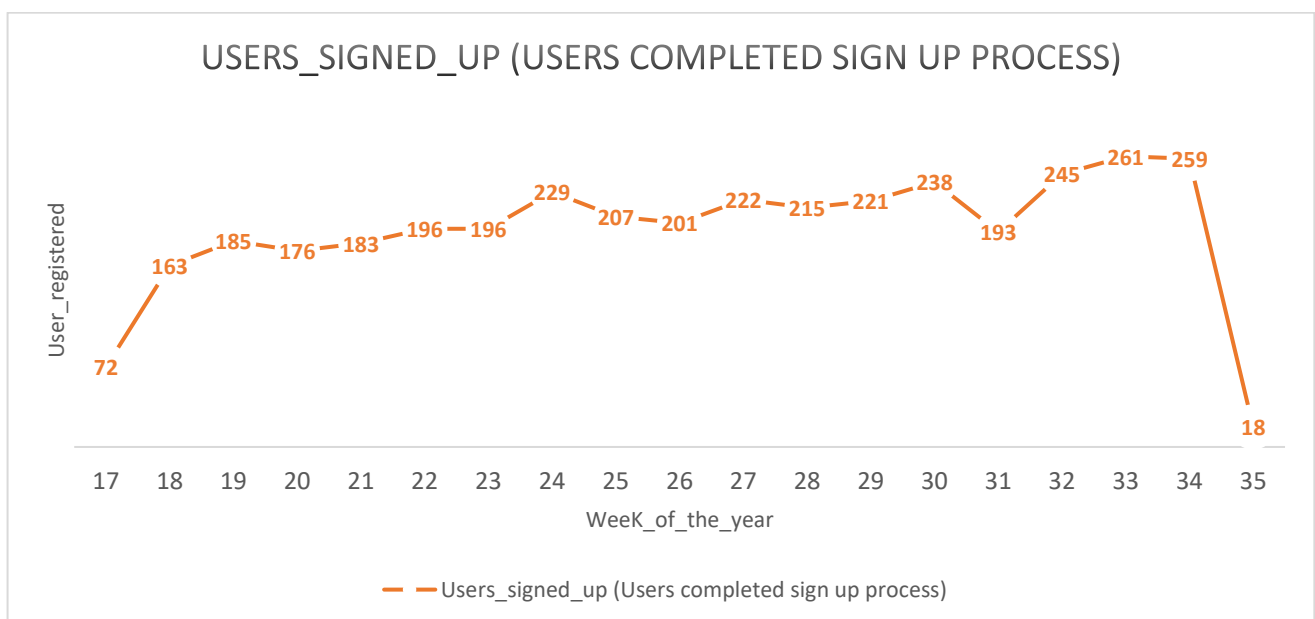*Table 5: Users logged on the platform*
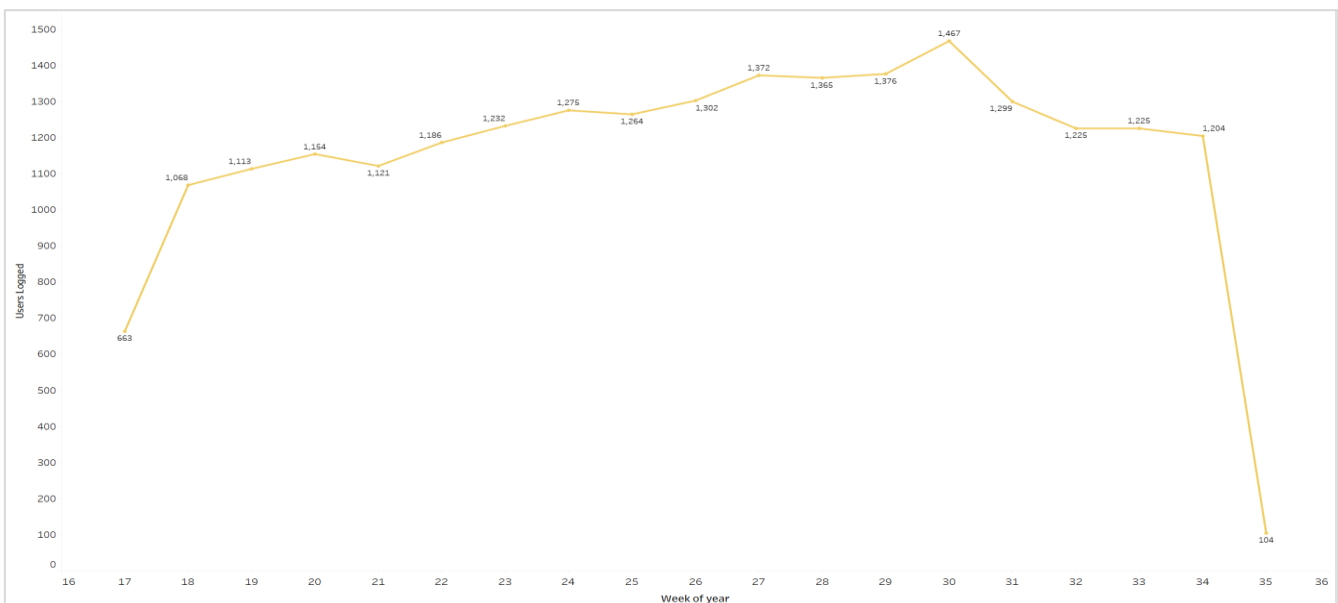
Finding 2: User Growth

Understanding the user registration pattern is vital, and this finding provides insights into the growth of user numbers over time for a specific product or platform. By analyzing the data, we can gauge the platform's popularity among users and evaluate its overall appeal. This information is valuable in assessing the platform's success, identifying user acquisition trends, and making informed decisions to enhance its user base and engagement.

Week_of_year (Calendar Week)	Users_signed_up (Users completed sign up process)
17	72
18	163
19	185
20	176
21	183
22	196
23	196
24	229
25	207
26	201
27	222
28	215
29	221
30	238
31	193
32	245
33	261
34	259
35	18

Table 6: Users Growth over time period

USERS_SIGNED_UP (USERS COMPLETED SIGN UP PROCESS)

Finding 3: Weekly Retention

The task is to calculate the weekly retention of users from the sign-up cohort to understand the number of users retained each week after signing up for the product. The table below illustrates the weekly retention numbers, providing valuable insights into user engagement and retention rates over time.

Week_of_the_year (Calendar Week)	User_registered	Retained_users	Retention_rate (%)
0	106	0	0
1	156	0	0
2	157	0	0
3	149	0	0
4	160	0	0
5	181	0	0
6	173	0	0
7	167	0	0
8	163	0	0
9	176	0	0
10	186	0	0
11	161	0	0
12	181	0	0
13	206	0	0
14	197	0	0
15	207	0	0
16	225	0	0
17	219	138	63
18	207	171	83
19	242	195	81
20	215	179	83
21	232	186	80
22	250	198	79
23	246	198	80
24	**274**	**234**	**85**
25	264	213	81
26	257	205	80
27	274	228	83
28	287	223	78
29	288	231	80
30	305	246	81
31	260	198	76
32	316	251	79
33	334	269	81
34	337	262	78
35	81	19	23

Table 7:Weekly User Rentention

Finding 4: Weekly Engagement

The table presented below showcases the weekly engagement per device. This analysis allows us to measure the level of user activity and assess the quality of the product or service on a weekly basis. By examining user engagement metrics per device, such as frequency and duration of interactions, we can gain insights into user satisfaction and determine the effectiveness of the product or service in meeting their needs.

Week_of_the_year	Total_Activity	Device_engaged	Weekly_average_engagement_per_device
17	8019	26	308
18	17341	26	667
19	17224	26	662
20	17911	26	689
21	17151	26	660
23	18280	26	703
22	18413	26	708
24	19052	26	733
25	18642	26	717
29	20067	26	772
26	19061	26	733
30	21533	26	828
28	20776	26	799
27	19881	26	765
31	18556	26	714
32	16612	26	639
33	16145	26	621
34	16127	26	620
35	784	23	34

Table 8:Weekly device engagement

Finding 5: Email Engagement

To determine the engagement of users with different types of weekly email events, the first I calculated the total number of events. This calculation provides an overview of the overall level of user interaction with the emails on a weekly basis.

Action_type	Total_events
sent_weekly_digest	3861172
email_open	1442899
email_clickthrough	714506
sent_reengagement_email	141706

Table 9: Different emails events count

Weekly calculation

Week_of_the_year	Weekly_digest	Re_engagement_email	User_opened_email	Email_clickthrough
17	908	73	310	166
18	2602	157	900	425
19	2665	173	961	476
20	2733	191	989	501
21	2822	164	996	436
22	2911	192	965	478
23	3003	197	1057	529
24	3105	226	1136	549
25	3207	196	1084	524
26	3302	219	1149	550
27	3399	213	1207	613
28	3499	213	1228	594
29	3592	213	1201	583
30	3706	231	1363	625
31	3793	222	1338	444
32	3897	200	1318	416
33	4012	264	1417	490
34	4111	261	1502	481
35	0	48	41	38

Table 10: Weekly email event type count

APPENDIX (SQL queries)

For the case study one the data was given in csv format and this data has been imported to mysql with the following queries.

/** CASE STUDY 1- SQL QUERY TO CREATE DATABASE AND INTER DATA INTO TABLE**/

CREATE DATABASE job;

use job;

CREATE TABLE job_data

(

   ds DATE,

   job_id INT NOT NULL,

   actor_id INT NOT NULL,

   event VARCHAR(15) NOT NULL,

   language VARCHAR(15) NOT NULL,

   time_spent INT NOT NULL,

   org CHAR(2)

);

INSERT INTO job_data(ds,job_id,actor_id,event,language,time_spent,org)

VALUES('2020-11-30','21','1001','skip','English','15','A'),

('2020-11-30','22','1006','transfer','Aracbic','25','B'),

('2020-11-29','23','1003','decision','Persian','20','C'),

('2020-11-28','23','1005','transfer','Persian','22','D'),

('2020-11-28','25','1002','decision','Hindi','11','B'),

('2020-11-27','11','1007','decision','French','104','D'),

('2020-11-26','23','1004','skip','Persian','56','A'),

('2020-11-25','20','1003','transfer','Italian','45','C');

CASE STUDY 1

```sql
-- Task 1: Number of Jobs Reviewed
SELECT
    ds AS Date,
    COUNT(job_id) AS Jobs_per_day,
    SUM(time_spent) AS time_spent_per_day_in_secs,
    SUM(time_spent) / 3600 AS time_spent_per_day_in_hours,
    COUNT(job_id) / (SUM(time_spent) / 3600) AS Jobs_reviewed_per_day_per_hour
FROM
    job.job_data
GROUP BY 1
```

```sql
-- TASK 2: Throughput
WITH MA AS -- MA represent for Moving Average
(
SELECT
    ds AS Date,
    event AS Event_type,
    COUNT(event) AS Total_events,
    SUM(time_spent) AS Total_timespent_in_secs,
    COUNT(event) / SUM(time_spent) AS Events_per_Second,
    COUNT(event) / (SUM(time_spent) / 3600) AS Events_per_hour
FROM job.job_data
GROUP BY 1 , 2
)
SELECT
        date,
        Event_type,
        Total_events,
        Total_timespent_in_secs,
        Events_per_Second,
        ROUND(Events_per_hour) as Event_per_hour,
        ROUND(AVG(Events_per_hour) OVER(ORDER BY Date ROWS BETWEEN 1 PRECEDING AND
CURRENT ROW))  AS Moving_average
 from    MA;
```

```sql
-- Task 3: Language Percentage Share
SELECT
language AS CV_Language,
count(job_id) AS Number_of_CV,
count(job_id)*100 / sum(count(*)) OVER() AS Percnetage_Share
```

```
FROM job.job_data
WHERE ds BETWEEN'2020-11-01' AND '2020-11-30'
GROUP BY 1
ORDER BY 1 ASC;
```

-- *Task 4: Duplicate Row*

```
WITH cte as
(
SELECT
    job_id AS JOB_ID,
    COUNT(actor_id) AS Application_count
FROM  job.job_data
GROUP BY job_id
)
SELECT
    j.ds AS Date, c.JOB_ID AS Unique_Job_ID, j.actor_id AS Actor_ID, j.event AS Event_Type, j.language AS
              Application_langauge
FROM
    cte c
      LEFT JOIN
    job.job_data j ON c.JOB_ID = j.job_id
WHERE
    Application_count = 3
```

CASE STUDY 2

-- *Task 1: User Engagement*

```
SELECT
    EXTRACT(WEEK FROM occurred_at) AS week_of_year,
    Date(MIN(occurred_at)) AS week_start_date,
    Date(MAX(occurred_at)) AS week_end_date,
    COUNT(DISTINCT user_id) AS Users_Logged
FROM
    job.events
WHERE
    event_type = 'en
gagement'
GROUP BY
    week_of_year
ORDER BY
```

```
        week_of_year;
```

```
SELECT
    EXTRACT(WEEK FROM occurred_at) AS week_of_year,
    COUNT(DISTINCT user_id) AS Users_signed_up
FROM job.events
WHERE
    event_type = 'signup_flow' ANd event_name='complete_signup'
GROUP BY
    week_of_year
ORDER BY
    week_of_year;
```

```
SELECT
    EXTRACT(WEEK FROM u.created_at) AS Week_of_the_year,
    COUNT(DISTINCT u.user_id) AS Users_registered,
    COUNT(DISTINCT e.user_id) AS Retained_users,
    COUNT(DISTINCT e.user_id)*100/COUNT(DISTINCT u.user_id) AS retention_rate
FROM
    users u
        LEFT JOIN
    events e ON u.user_id = e.user_id
        AND WEEK(e.occurred_at) = WEEK(u.created_at)
WHERE
    u.state = 'active'
GROUP BY Week_of_the_year
ORDER BY Week_of_the_year;
```

```
With WDT AS -- WDA means weekly device type
(
 SELECT
    e.device AS Device_name,
    EXTRACT(WEEK FROM e.occurred_at) AS Week_of_the_year,
    COUNT(e.device) AS Number_of_devices
FROM
    job.events e
WHERE
    e.event_type = 'Engagement'
```

```
GROUP BY 1,2
)
SELECT
    Week_of_the_year, AVG(Number_of_devices) AS Weekly_average_engagement_per_device
FROM
    WDT
GROUP BY 1;LECT
    device AS Device_name,
    EXTRACT(WEEK FROM occurred_at) AS Week_of_the_year,
    COUNT(device) AS Number_of_devices
FROM
    events
WHERE
    event_type = 'Engagement'
GROUP BY 1,2
ORDER BY 2;
```

```
-- Task 5: Email Engagement
SELECT
    WEEK(occurred_at) AS Week_of_the_year,
    COUNT(DISTINCT (CASE
        WHEN action = 'sent_weekly_digest' THEN user_id
      END)) AS Weekly_digest,
    COUNT(DISTINCT (CASE
        WHEN action = 'sent_reengagement_email' THEN user_id
      END)) AS Re_engagement_email,
    COUNT(DISTINCT (CASE
        WHEN action = 'email_open' THEN user_id
      END)) AS User_opened_email,
    COUNT(DISTINCT (CASE
        WHEN action = 'email_clickthrough' THEN user_id
      END)) AS Email_clickthrough
FROM
    job.email_events
GROUP BY WEEK(occurred_at)
ORDER BY WEEK(occurred_at);
```