TRAINITY PROJECT 03

Operation Analytics and Investigating Metric Spike

Project Description:

Operational Analytics involves analysing the company's end-to-end operations to identify areas for improvement and helping teams optimize their processes and performance using data-driven insights.

Investigating Metric Spikes means monitoring key metrics like daily user engagement, sales etc. It also includes investigating and explaining sudden changes or anomalies in these metrics.

My role as a designated data analyst is to ensure that the company makes informed decision by leveraging data to understand operational inefficiencies, investigate unusual changes in key metrics and support teams with actionable insights for improvement.

The objective of this project is to offer insights into two distinct scenarios and metric analytics.

Objectives:

Case study 1: Job Data Analytics

- To calculate the number of jobs reviewed per hour for each day in November 2020.
- Calculate the 7-day rolling average of throughput.
- Calculate the percentage share of each language in the last 30 days.
- Identify duplicate rows in the data.

Case study 2: Investigating Metric Spike

- Measure the activeness of users on a weekly basis.
- Analyse the growth of users over time for a product.
- Analyse the retention of users on a weekly basis after signing up for a product.
- Measure the activeness of users on a weekly basis per device.
- Analyse how users are engaging with the email service.

Tech-Stack Used:

- MySQL Workbench 8.0
- Ms Excel

Approach:

To start the project I first downloaded the datasets from the given link. Next I created the required tables and imported the data from the excel sheets. I then analysed the tables and understood their respective columns. I also went through the tasks thoroughly and gained an understanding of what I needed to do. I then started using SQL queries to to further analyse the data and solve the problems.

Case Study 1: Job Data Analysis

The process of creating a new database, creating the job_data table and inserting values into it.

```
#case study 1: job data analysis

#create table job_data
create database project3;

CREATE TABLE job_data(
    ds DATE,
    job_id INT NOT NULL,
    actor_id INT NOT NULL,
    event VARCHAR(10) NOT NULL,
    language VARCHAR(10) NOT NULL,
    time_spent INT NOT NULL,
    org CHAR(2)
);
```

```
#inserting the data into the job_data table
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', 'Arabic', 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');
```

Tasks:

A. **Jobs Reviewed Over Time:**
   - **Objective:** Calculate the number of jobs reviewed per hour for each day in November 2020.
   - **Your Task:** Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.

```
/*
Objective: Calculate the number of jobs reviewed per hour for
each day in November 2020.
Your Task: Write an SQL query to calculate the number of jobs reviewed
per hour for each day in November 2020.
*/
select ds as date, round(count(job_id)/sum(time_spent)*3600) as jobs_reviewed_perhour
from job_data
where ds between '2020-11-01' and '2020-11-30'
group by ds
order by dsj
```

Output:

date	jobs_reviewed_perhour
2020-11-25	80
2020-11-26	64
2020-11-27	35
2020-11-28	218
2020-11-29	180
2020-11-30	180

Insights:

- The highest number of jobs were reviewed on 28 November 2020 with 218 jobs per hour
- The lowest number of jobs were reviewed on 27 November 2020 with 35 jobs per hour

B. Throughput Analysis:

- **Objective:** Calculate the 7-day rolling average of throughput (number of events per second).
- **Your Task:** Write an SQL query 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.

```
/*
Objective: Calculate the 7-day rolling average of throughput (number of events per second).
Your Task: Write an SQL query 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.
*/
SELECT ds as dates,
COUNT(event) / SUM(time_spent) AS daily_throughput,
ROUND(AVG(COUNT(event) / SUM(time_spent)) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW), 2) AS 7_day_rolling_avg_throughput
FROM job_data
GROUP BY ds
ORDER BY ds;
```

Output:

dates	daily_throughput	7_day_rolling_avg_throughput
2020-11-25	0.0222	0.02
2020-11-26	0.0179	0.02
2020-11-27	0.0096	0.02
2020-11-28	0.0606	0.03
2020-11-29	0.0500	0.03
2020-11-30	0.0500	0.04

Insights:

- The throughput is the highest on 28 November 2020 at 0.06
- The throughput is the lowest on 27 November 2020 at 0.009
- The metrics will keep going up or down on a weekly and daily basis

C. Language Share Analysis:

- **Objective:** Calculate the percentage share of each language in the last 30 days.
- **Your Task:** Write an SQL query to calculate the percentage share of each language over the last 30 days.

```
/*
Objective: Calculate the percentage share of each language in the last 30 days.
Your Task: Write an SQL query to calculate the percentage share of each language over the last 30 days.
*/
select language,
count(*) /
(select count(*) from job_data) * 100 as percentage_share
from job_data
group by language
order by language desc;
```

Output:

language	percentage_share
Persian	37.5000
Italian	12.5000
Hindi	12.5000
French	12.5000
English	12.5000
Arabic	12.5000

Insights:

- The most used language is Persian with a percentage share of 37.5%
- All the other languages like Italian, Hindi, French, English and Arabic are same with the percentage share of 12.5%

D. **Duplicate Rows Detection:**
   - **Objective:** Identify duplicate rows in the data.
   - **Your Task:** Write an SQL query to display duplicate rows from the job_data table.

```
/*
Objective: Identify duplicate rows in the data.
Your Task: Write an SQL query to display duplicate rows from the job_data table.
*/
select actor_id, count(*) as duplicate_rows
from job_data
group by actor_id
having count(*) > 1;
```

Output:

actor_id	duplicate_rows
1003	2

Insights:

- The actor id 1003 has a total of 2 duplicate rows

Case study 2: Investigation Metric Spike

This case study required me to create three new table namely, users, events and email_events.

It also included importing data from the excel sheets.

```
#case study 2

#creating table 1 - users
create table users(
    user_id int,
    created_at  varchar(100),
    company_id  int,
    language varchar(50),
    activated_at varchar(100),
    state varchar (50)
);

show variables like 'secure_file_priv';

#importing data from users.csv
load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users.csv"
into table users
fields terminated by ','
enclosed by '"'
lines terminated by '\n'
ignore 1 rows;

select * from users;

#altering table users
alter table users add column temp_created_at datetime;

update users set temp_created_at = STR_TO_DATE(created_at, '%d-%m-%Y %H:%i');

alter table users drop column created_at;

alter table users change column temp_created_at created_at datetime;
```

```sql
#creating table 2 - events
create table events(
    user_id int,
    occurred_at varchar(100),
    event_type  varchar(50),
    event_name  varchar(100),
    location varchar(50),
    device varchar(50),
    user_type int
);

#importing data from events.csv
load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/events.csv"
into table events
fields terminated by ','
enclosed by '"'
lines terminated by '\n'
ignore 1 rows;

select * from events;
#altering table events
alter table events add column temp_occured_at datetime;

update events set temp_occured_at = STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i');

alter table events drop column occurred_at;

alter table events change column temp occured at occurred at datetime;
```

```sql
# creating table 3 - email-events
create table email_events(
    user_id int,
    occurred_at varchar(100),
    action varchar(100),
    user_type int
);

#importing data from email_events.csv
load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/email_events.csv'
into table email_events
fields terminated by ','
enclosed by '"'
lines terminated by '\n'
ignore 1 rows;

select * from email_events;
#altering table email_events
alter table email_events add column temp_occured_at datetime;

update email_events set temp_occured_at = STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i');

alter table email_events drop column occurred_at;

alter table email_events change column temp_occured_at occurred_at datetime;
```

Tasks:

A. **Weekly User Engagement:**
   - **Objective:** Measure the activeness of users on a weekly basis.
   - **Your Task:** Write an SQL query to calculate the weekly user engagement.

```sql
/*
Objective: Measure the activeness of users on a weekly basis.
Your Task: Write an SQL query to calculate the weekly user engagement.
*/
select extract(week from occurred_at) as week_number,
count(distinct user_id) as number_of_users
from events
where event_type = 'engagement'
group by week_number;
```

Output:

week_number	number_of_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

Insights:

- Week 30 saw the highest user engagement with 1467 active users.
- The user engagement increased after week 17 but later started dropping again after being the highest on week 30

B. **User Growth Analysis:**
   - **Objective:** Analyze the growth of users over time for a product.
   - **Your Task:** Write an SQL query to calculate the user growth for the product.

```
/*
Objective: Analyze the growth of users over time for a product.
Your Task: Write an SQL query to calculate the user growth for the product.
*/
select months, new_users, ((new_users / lag(new_users,1) over(
order by months) - 1) * 100) as growth
from
(
select month(created_at) as months,
count(distinct user_id) as new_users
from users
where users.activated_at is not null
group by months
order by months
) as sub_query;
```

Output:

months	new_users	growth
1	712	NULL
2	685	-3.7921
3	765	11.6788
4	907	18.5621
5	993	9.4818
6	1086	9.3656
7	1281	17.9558
8	1347	5.1522
9	330	-75.5011
10	390	18.1818
11	399	2.3077
12	486	21.8045

Insights:

- There was significant growth in months 4 and 7 but that was followed by a sharp decline in month 9.
- The trend shows a recovery with steady growth in the later months.

C. **Weekly Retention Analysis:**

- **Objective:** Analyze the retention of users on a weekly basis after signing up for a product.
- **Your Task:** Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

```
/*
Objective: Analyze the retention of users on a weekly basis after signing up for a product.
Your Task: Write an SQL query to calculate the weekly retention of users based on
their sign-up cohort.
*/
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
```

```
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 events
group by 1) n
where m.user_id = n.user_id
) sub
group by first
order by first;
```

Output:

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	663	472	324	251	205	187	167	146	145	145	136	131	132	143	116	91	82	77	5
18	596	362	261	203	168	147	144	127	113	122	106	118	127	110	97	85	67	4	0
19	427	284	173	153	114	95	91	81	95	82	68	65	63	42	51	49	2	0	0
20	358	223	165	121	91	72	63	67	63	65	67	41	40	33	40	0	0	0	0
21	317	187	131	91	74	63	75	72	58	48	45	39	35	28	2	0	0	0	0
22	326	224	150	107	87	73	63	60	55	48	41	39	31	1	0	0	0	0	0
23	328	219	138	101	90	79	69	61	54	47	35	30	0	0	0	0	0	0	0
24	339	205	143	102	81	63	65	61	38	39	29	0	0	0	0	0	0	0	0
25	305	218	139	101	75	63	50	46	38	35	2	0	0	0	0	0	0	0	0
26	288	181	114	83	73	55	47	43	29	0	0	0	0	0	0	0	0	0	0
27	292	199	121	106	68	53	40	36	1	0	0	0	0	0	0	0	0	0	0
28	274	194	114	69	46	30	28	3	0	0	0	0	0	0	0	0	0	0	0
29	270	186	102	65	47	40	1	0	0	0	0	0	0	0	0	0	0	0	0
30	294	202	121	78	53	3	0	0	0	0	0	0	0	0	0	0	0	0	0
31	215	145	76	57	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	267	188	94	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	286	202	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	279	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Insights:

- The retention rate is high initially but, as weeks go by it starts to go down.

D. Weekly Engagement Per Device:

- **Objective:** Measure the activeness of users on a weekly basis per device.
- **Your Task:** Write an SQL query to calculate the weekly engagement per device.

Output:

week	device	user_engagement
17	acer aspire desktop	9
17	acer aspire notebook	20
17	amazon fire phone	4
17	asus chromebook	21
17	dell inspiron desktop	18
17	dell inspiron notebook	46
17	hp pavilion desktop	14
17	htc one	16
17	ipad air	27
17	ipad mini	19
17	iphone 4s	21
17	iphone 5	65
17	iphone 5s	42
17	kindle fire	6
17	lenovo thinkpad	86
17	mac mini	6
17	macbook air	54
17	macbook pro	143
17	nexus 10	16
17	nexus 5	40
17	nexus 7	18
17	nokia lumia 635	17
17	samsumg galaxy tablet	8
17	samsung galaxy note	7
17	samsung galaxy s4	52
17	windows surface	10
18	acer aspire desktop	26
18	acer aspire notebook	33
18	amazon fire phone	9
18	asus chromebook	42
18	dell inspiron desktop	58
18	dell inspiron notebook	77
18	hp pavilion desktop	37
18	htc one	19
18	ipad air	52
18	ipad mini	30
18	iphone 4s	46
18	iphone 5	113
18	iphone 5s	73
18	kindle fire	27
18	lenovo thinkpad	153
18	mac mini	13
18	macbook air	121
18	macbook pro	252
18	nexus 10	30
18	nexus 5	73
18	nexus 7	30
18	nokia lumia 635	33
18	samsumg galaxy tablet	11
18	samsung galaxy note	15
18	samsung galaxy s4	82
18	windows surface	10
19	acer aspire desktop	23
19	acer aspire notebook	41
19	amazon fire phone	12
19	asus chromebook	27
19	dell inspiron desktop	36
19	dell inspiron notebook	83
19	hp pavilion desktop	40
19	htc one	30
19	ipad air	55
19	ipad mini	36
19	iphone 4s	44
19	iphone 5	115
19	iphone 5s	79
19	kindle fire	21
19	lenovo thinkpad	178
19	mac mini	18
19	macbook air	112
19	macbook pro	266
19	nexus 10	25
19	nexus 5	87
19	nexus 7	41
19	nokia lumia 635	23
19	samsumg galaxy tablet	6
19	samsung galaxy note	11
19	samsung galaxy s4	91
19	windows surface	16
20	acer aspire desktop	23
20	acer aspire notebook	40
20	amazon fire phone	11
20	asus chromebook	41
20	dell inspiron desktop	52
20	dell inspiron notebook	84
20	hp pavilion desktop	30
20	htc one	29
20	ipad air	59
20	ipad mini	32
20	iphone 4s	55
20	iphone 5	125
20	iphone 5s	79
20	kindle fire	23
20	lenovo thinkpad	173
20	mac mini	26
20	macbook air	119
20	macbook pro	
20	nexus 10	22
20	nexus 5	103
20	nexus 7	32
20	nokia lumia 635	22
20	samsumg galaxy tablet	9
20	samsung galaxy note	18
20	samsung galaxy s4	93
20	windows surface	21
21	acer aspire desktop	29
21	acer aspire notebook	47
21	amazon fire phone	5
21	asus chromebook	38
21	dell inspiron desktop	41
21	dell inspiron notebook	80
21	hp pavilion desktop	44
21	htc one	21
21	ipad air	51
21	ipad mini	23
21	iphone 4s	45
21	iphone 5	137
21	iphone 5s	74
21	kindle fire	30
21	lenovo thinkpad	167
21	mac mini	18

week	device	user_engagement
21	macbook air	110
21	macbook pro	247
21	nexus 10	25
21	nexus 5	91
21	nexus 7	29
21	nokia lumia 635	25
21	samsumg galaxy tablet	6
21	samsung galaxy note	20
21	samsung galaxy s4	84
21	windows surface	17
22	acer aspire desktop	25
22	acer aspire notebook	41
22	amazon fire phone	5
22	asus chromebook	52
22	dell inspiron desktop	52
22	dell inspiron notebook	92
22	hp pavilion desktop	38
22	htc one	24
22	ipad air	58
22	ipad mini	34
22	iphone 4s	45
22	iphone 5	125
22	iphone 5s	71
22	kindle fire	21
22	lenovo thinkpad	176
22	mac mini	25
22	macbook air	145
22	macbook pro	251
22	nexus 10	27
22	nexus 5	96
22	nexus 7	45
22	nokia lumia 635	25
22	samsumg galaxy tablet	10
22	samsung galaxy note	19
22	samsung galaxy s4	105
22	windows surface	15
23	acer aspire desktop	22
23	acer aspire notebook	43
23	amazon fire phone	16
23	asus chromebook	49
23	dell inspiron desktop	53
23	dell inspiron notebook	103
23	hp pavilion desktop	54
23	htc one	20
23	ipad air	41
23	ipad mini	33
23	iphone 4s	53
23	iphone 5	152
23	dell inspiron notebook	103
23	hp pavilion desktop	54
23	htc one	20
23	ipad air	41
23	ipad mini	33
23	iphone 4s	53
23	iphone 5	152
23	iphone 5s	79
23	kindle fire	25
23	lenovo thinkpad	176
23	mac mini	18
23	macbook air	124
23	macbook pro	266
23	nexus 10	45
23	nexus 5	88
23	nexus 7	36
23	nokia lumia 635	31
23	samsumg galaxy tablet	14
23	samsung galaxy note	14
23	samsung galaxy s4	99
23	windows surface	14
24	acer aspire desktop	24
24	acer aspire notebook	40
24	amazon fire phone	11
24	lenovo thinkpad	165
24	mac mini	29
24	macbook air	152
24	macbook pro	255
24	nexus 10	38
24	nexus 5	87
24	nexus 7	49
24	nokia lumia 635	35
24	samsumg galaxy tablet	11
24	samsung galaxy note	20
24	samsung galaxy s4	101
24	windows surface	22
25	acer aspire desktop	28
25	acer aspire notebook	47
25	amazon fire phone	13
25	asus chromebook	38
25	dell inspiron desktop	52
25	dell inspiron notebook	105
25	hp pavilion desktop	52
25	htc one	21
25	ipad air	57
25	ipad mini	30
25	iphone 4s	40
25	iphone 5	137
25	iphone 5s	78
25	kindle fire	24
25	lenovo thinkpad	197
25	mac mini	21
25	macbook air	121
25	macbook pro	275
25	nexus 10	29
25	nexus 5	89
25	nexus 7	51
25	nokia lumia 635	37
25	samsumg galaxy tablet	12
25	samsung galaxy note	14
25	samsung galaxy s4	99
25	windows surface	22
26	acer aspire desktop	29
26	acer aspire notebook	35
26	amazon fire phone	13
26	asus chromebook	49
26	dell inspiron desktop	60
26	dell inspiron notebook	89
26	hp pavilion desktop	46
26	htc one	23
26	ipad air	56
26	ipad mini	43
26	iphone 4s	50
26	iphone 5	152
26	iphone 5s	94
26	kindle fire	26
26	lenovo thinkpad	192
26	mac mini	11
26	macbook air	134
26	macbook pro	269
26	nexus 10	29
26	nexus 5	87
26	nexus 7	46
26	nokia lumia 635	42
26	samsumg galaxy tablet	12
26	samsung galaxy note	9
26	samsung galaxy s4	112
26	windows surface	21
27	acer aspire desktop	29
27	acer aspire notebook	49
27	amazon fire phone	10
27	asus chromebook	52
27	dell inspiron desktop	53
27	dell inspiron notebook	89
27	hp pavilion desktop	56
27	htc one	22

week	device	user_engagement
27	ipad air	55
27	ipad mini	35
27	iphone 4s	67
27	iphone 5	163
27	iphone 5s	83
27	kindle fire	25
27	lenovo thinkpad	202
27	mac mini	15
27	macbook air	142
27	macbook pro	302
27	nexus 10	37
27	nexus 5	84
27	nexus 7	40
27	nokia lumia 635	31
27	samsumg galaxy tablet	15
27	samsung galaxy note	15
27	samsung galaxy s4	116
27	windows surface	33
28	acer aspire desktop	30
28	acer aspire notebook	49
28	amazon fire phone	6
28	asus chromebook	50
28	dell inspiron desktop	56
28	dell inspiron notebook	103
28	hp pavilion desktop	56
28	htc one	26
28	ipad air	54
28	ipad mini	35
28	iphone 4s	61
28	iphone 5	151
28	iphone 5s	93
28	kindle fire	31
28	lenovo thinkpad	220
28	mac mini	28
28	macbook air	148
28	macbook pro	295
28	nexus 10	26
28	nexus 5	85
28	nexus 7	39
28	nokia lumia 635	35
28	samsumg galaxy tablet	9
28	samsung galaxy note	10
28	samsung galaxy s4	122
28	windows surface	33
29	acer aspire desktop	28
29	acer aspire notebook	53
29	amazon fire phone	12
29	asus chromebook	49
29	htc one	31
29	ipad air	52
29	ipad mini	34
29	iphone 4s	60
29	iphone 5	144
29	iphone 5s	90
29	kindle fire	37
29	lenovo thinkpad	209
29	mac mini	31
29	macbook air	148
29	macbook pro	295
29	nexus 10	25
29	nexus 5	77
29	nexus 7	45
29	nokia lumia 635	43
29	samsumg galaxy tablet	13
29	samsung galaxy note	16
29	samsung galaxy s4	123
29	windows surface	28
30	acer aspire desktop	33
30	acer aspire notebook	60
30	amazon fire phone	12
30	asus chromebook	56
30	dell inspiron desktop	54
30	nexus 10	36
30	nexus 5	84
30	nexus 7	62
30	nokia lumia 635	34
30	samsumg galaxy tablet	9
30	samsung galaxy note	15
30	samsung galaxy s4	103
30	windows surface	19
31	acer aspire desktop	31
31	acer aspire notebook	55
31	amazon fire phone	14
31	asus chromebook	56
31	dell inspiron desktop	44
31	dell inspiron notebook	113
31	hp pavilion desktop	51
31	htc one	13
31	ipad air	55
31	ipad mini	27
31	iphone 4s	56
31	iphone 5	135
31	iphone 5s	71
31	kindle fire	14
31	lenovo thinkpad	207
31	mac mini	24
31	macbook air	147
31	macbook pro	321
31	nexus 10	24
31	nexus 5	69
31	nexus 7	38
31	nokia lumia 635	28
31	samsumg galaxy tablet	8
31	samsung galaxy note	14
31	samsung galaxy s4	100
31	windows surface	19
32	acer aspire desktop	35
32	acer aspire notebook	55
32	amazon fire phone	12
32	asus chromebook	62
32	dell inspiron desktop	57
32	dell inspiron notebook	104
32	hp pavilion desktop	51
32	htc one	18
32	ipad air	48
32	ipad mini	30
32	iphone 4s	34
32	iphone 5	119
32	iphone 5s	67
32	kindle fire	2
32	lenovo thinkpad	179
32	mac mini	20
32	macbook air	125
32	macbook pro	307
32	nexus 10	30
32	nexus 5	67
32	nexus 7	25
32	nokia lumia 635	28
32	samsumg galaxy tablet	6
32	samsung galaxy note	12
32	samsung galaxy s4	82
32	windows surface	10
33	acer aspire desktop	39
33	acer aspire notebook	46
33	amazon fire phone	14
33	asus chromebook	49
33	dell inspiron desktop	37
33	dell inspiron notebook	110
33	hp pavilion desktop	38
33	htc one	19
33	ipad air	40
33	ipad mini	28
33	iphone 4s	35
33	iphone 5	110

week	device	user_engagement
33	iphone 5s	65
33	kindle fire	14
33	lenovo thinkpad	191
33	mac mini	32
33	macbook air	133
33	macbook pro	312
33	nexus 10	23
33	nexus 5	70
33	nexus 7	30
33	nokia lumia 635	27
33	samsumg galaxy tablet	12
33	samsung galaxy note	13
33	samsung galaxy s4	80
33	windows surface	15
34	acer aspire desktop	30
34	acer aspire notebook	63
34	amazon fire phone	11
34	asus chromebook	47
34	dell inspiron desktop	49
34	dell inspiron notebook	105
34	hp pavilion desktop	36
34	htc one	25
34	ipad air	39
34	ipad mini	25
34	iphone 4s	50
34	iphone 5	101
34	iphone 5s	70
34	kindle fire	13
34	lenovo thinkpad	193
34	mac mini	30
34	macbook air	136
34	macbook pro	292
34	nexus 10	25
34	nexus 5	70
34	nexus 7	33
34	nokia lumia 635	17
34	samsumg galaxy tablet	14
34	samsung galaxy note	13
34	samsung galaxy s4	90
34	windows surface	18
35	acer aspire desktop	1
35	acer aspire notebook	3
35	asus chromebook	6
35	dell inspiron desktop	1
35	dell inspiron notebook	9
35	hp pavilion desktop	1
35	htc one	2
35	ipad mini	2
35	iphone 4s	6
35	iphone 5	2
35	iphone 5s	3
35	kindle fire	3
35	lenovo thinkpad	16
35	mac mini	2
35	macbook air	10
35	macbook pro	17
35	nexus 10	2
35	nexus 5	4
35	nexus 7	2
35	nokia lumia 635	2
35	samsung	
35	samsung galaxy s4	6
35	windows surface	3

samsung galaxy note

E. **Email Engagement Analysis:**

- **Objective:** Analyze how users are engaging with the email service.
- **Your Task:** Write an SQL query to calculate the email engagement metrics.

```
/*
Objective: Analyze how users are engaging with the email service.
Your Task: Write an SQL query to calculate the email engagement metrics.
*/

select distinct week(occurred_at) as week_num,
count(distinct case when action = 'sent_weekly_digest' then user_id end) as email_digest,
count(distinct case when action='sent_reengagement_email' then user_id end) as reengagement_emails,
count(distinct case when action = 'email_clickthrough' THEN user_id end) as click_through,
count(distinct case when action ='email_open' then user_id end) as email_open
from email_events
group by week(occurred_at);
```

Output:

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

Insights:

- It is observed that most of the email activity pertains to the email_digest.

Conclusion:

Through this project I was able to learn and understand how important Operation Analytics is for organizations. I was able to answer various questions which can help the management team as per the requirements. The teams can use these insights and make proper data driven decisions.

This project has allowed me to test my SQL skills and gain valuable experience in data analysis. It has helped me understand Advance SQL concepts better.