Drive Link for the project-
https://drive.google.com/drive/u/0/folders/11pqMw_Pzh1u3olNzV9UzXr0boF1fOU-v

Project by –

Mohd Faiz

faiz.mohd340@gmail.com

Instagram User Analytics

Project Description

In order to improve automation, cross-functional team understanding, and workflow efficiency as well as to help other teams understand why daily engagement is declining or why sales are declining, this project will go over some of the crucial questions the operations team, support team, marketing team, etc. have to predict the overall growth or decline of a company's fortune.

Project Approach

To answer the questions, SQL was used. Using SQL, we created the database using the raw data provided to us for this project. Once, the database was set up, we performed various operations like selecting, sorting, joining tables, etc. to get the insights we needed.

Tech Stack Used

MySQL Workbench v8.0.32, MS Word, Google Drive.

Project Insight

Case Study -1

While using job_data table to have detailed result, we insertedmore values. Command for that is as follows-

```
CREATE SCHEMA project3;

USE project3;
```

```
CREATE TABLE job_data

(

   ds        VARCHAR(30),

   job_id  INT,

   actor_id            INT,

   event   VARCHAR(30),

   language            VARCHAR(30),

   time_spent          INT,

   org       VARCHAR(30)

);
```

```sql
CREATE SCHEMA project3;
USE project3;

CREATE TABLE job_data
(
    ds  VARCHAR(30),
    job_id  INT,
    actor_id    INT,
    event   VARCHAR(30),
    language    VARCHAR(30),
    time_spent  INT,
    org VARCHAR(30)
);
```

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-30', '21', '1001', 'skip', 'English', '15', 'A');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-30', '22', '1006', 'transfer', 'Arabic', '25', 'B');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-29', '23', '1003', 'decision', 'Persian', '20', 'C');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-28', '23', '1005', 'transfer', 'Persian', '22', 'D');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-28', '25', '1002', 'decision', 'Hindi', '11', 'B');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-27', '11', '1007', 'decision', 'French', '104', 'D');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-26', '23', '1004', 'skip', 'Persian', '56', 'A');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-25', '20', '1003', 'transfer', 'Italian', '45', 'C');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-24', '18', '1017', 'skip', 'English', '79', 'A');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-23', '26', '1008', 'transfer', 'Arabic', '74', 'B');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-29', '30', '1008', 'decision', 'Persian', '63', 'C');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-20', '24', '1010', 'transfer', 'Persian', '19', 'D');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-21', '19', '1014', 'decision', 'Hindi', '89', 'B');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-13', '27', '1011', 'decision', 'French', '67', 'D');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-10', '12', '1006', 'skip', 'Persian', '29', 'A');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-21', '16', '1004', 'transfer', 'Italian', '53', 'C');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-18', '22', '1015', 'skip', 'English', '40', 'A');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-20', '17', '1018', 'transfer', 'Arabic', '22', 'B');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-17', '14', '1009', 'decision', 'Persian', '86', 'C');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-16', '24', '1016', 'transfer', 'Persian', '102', 'D');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-15', '28', '1016', 'decision', 'Hindi', '27', 'B');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-10', '21', '1019', 'decision', 'French', '63', 'D');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-13', '12', '1020', 'skip', 'Persian', '46', 'A');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-12', '18', '1020', 'transfer', 'Italian', '102', 'C');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-11', '25', '1018', 'skip', 'English', '106', 'A');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-02', '14', '1016', 'transfer', 'Arabic', '18', 'B');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-03', '16', '1011', 'decision', 'Persian', '32', 'C');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-13', '24', '1019', 'transfer', 'Persian', '68', 'D');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-19', '28', '1013', 'decision', 'Hindi', '105', 'B');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-21', '30', '1016', 'skip', 'English', '87', 'B');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-18', '16', '1014', 'transfer', 'Arabic', '10', 'C');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-20', '20', '1004', 'decision', 'Persian', '72', 'D');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-17', '13', '1019', 'transfer', 'Persian', '31', 'B');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-16', '28', '1016', 'decision', 'Hindi', '69', 'D');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-15', '11', '1004', 'decision', 'French', '109', 'A');

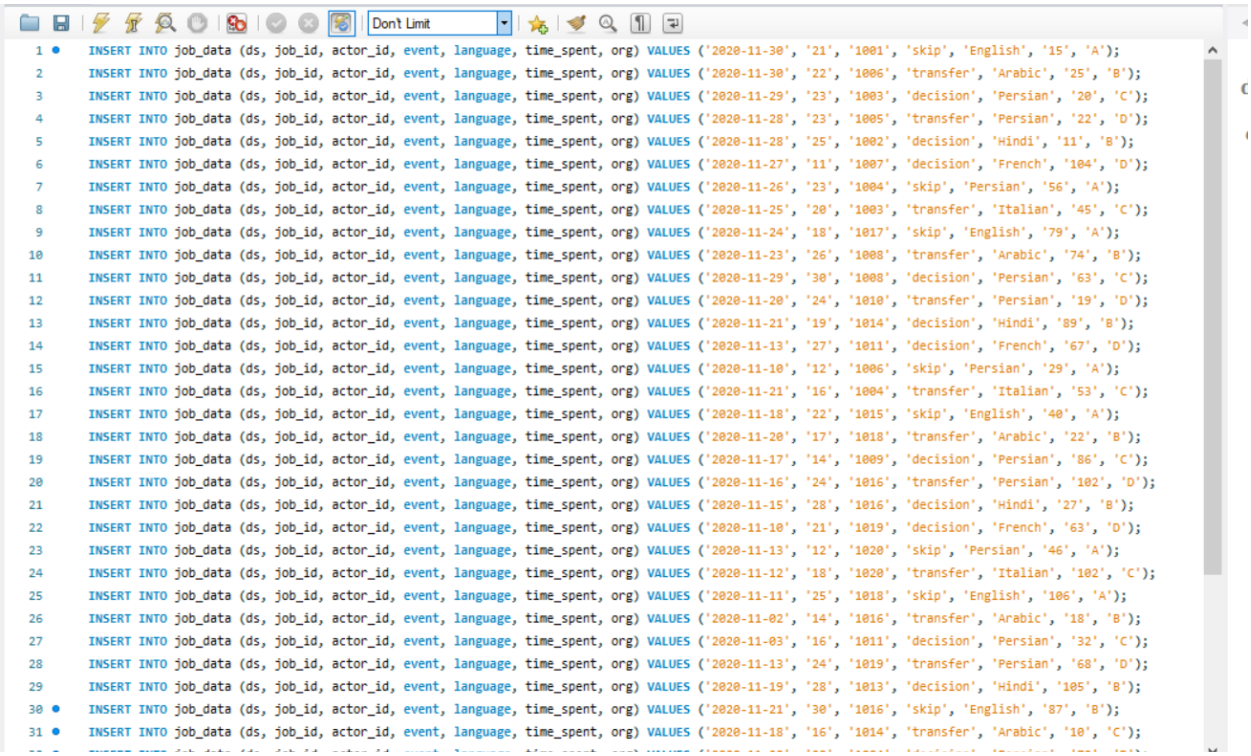
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-10', '27', '1007', 'skip', 'Persian', '44', 'C');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-13', '16', '1016', 'transfer', 'Italian', '21', 'A');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-12', '23', '1010', 'skip', 'English', '27', 'B');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-11', '29', '1011', 'transfer', 'Arabic', '68', 'C');

INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-02', '21', '1020', 'decision', 'Persian', '52', 'D');



```
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-30', '21', '1001', 'skip', 'English', '15', 'A');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-30', '22', '1006', 'transfer', 'Arabic', '25', 'B');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-29', '23', '1003', 'decision', 'Persian', '20', 'C');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-28', '23', '1005', 'transfer', 'Persian', '22', 'D');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-28', '25', '1002', 'decision', 'Hindi', '11', 'B');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-27', '11', '1007', 'decision', 'French', '104', 'D');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-26', '23', '1004', 'skip', 'Persian', '56', 'A');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-25', '20', '1003', 'transfer', 'Italian', '45', 'C');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-24', '18', '1017', 'skip', 'English', '79', 'A');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-23', '26', '1008', 'transfer', 'Arabic', '74', 'B');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-29', '30', '1008', 'decision', 'Persian', '63', 'C');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-20', '24', '1010', 'transfer', 'Persian', '19', 'D');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-21', '19', '1014', 'decision', 'Hindi', '89', 'B');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-13', '27', '1011', 'decision', 'French', '67', 'D');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-10', '12', '1006', 'skip', 'Persian', '29', 'A');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-21', '16', '1004', 'transfer', 'Italian', '53', 'C');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-18', '22', '1015', 'skip', 'English', '40', 'A');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-20', '17', '1018', 'transfer', 'Arabic', '22', 'B');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-17', '14', '1009', 'decision', 'Persian', '86', 'C');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-16', '24', '1016', 'transfer', 'Persian', '102', 'D');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-15', '28', '1016', 'decision', 'Hindi', '27', 'B');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-10', '21', '1019', 'decision', 'French', '63', 'D');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-13', '12', '1020', 'skip', 'Persian', '46', 'A');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-12', '18', '1020', 'transfer', 'Italian', '102', 'C');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-11', '25', '1018', 'skip', 'English', '106', 'A');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-02', '14', '1016', 'transfer', 'Arabic', '18', 'B');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-03', '16', '1011', 'decision', 'Persian', '32', 'C');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-13', '24', '1019', 'transfer', 'Persian', '68', 'D');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-19', '28', '1013', 'decision', 'Hindi', '105', 'B');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-21', '30', '1016', 'skip', 'English', '87', 'B');
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org) VALUES ('2020-11-18', '16', '1014', 'transfer', 'Arabic', '10', 'C');
```

1. **Number of jobs reviewed:** Amount of jobs reviewed over time.

SQL Query:

```
-- Calculating the number of jobs per hour per day for November 2020

SELECT
        ds, count(job_id) as Jobs_per_day,
sum(time_spent)/3600 as Hours_per_day
FROM
        job_data
WHERE
        ds >='2020-11-01' AND ds <='2020-11-30'
GROUP BY
        ds ;
```

ds	Jobs_per_day	Hours_per_day
2020-11-30	2	0.0111
2020-11-29	2	0.0231
2020-11-28	2	0.0092
2020-11-27	1	0.0289
2020-11-26	1	0.0156
2020-11-25	1	0.0125
2020-11-24	1	0.0219
2020-11-23	1	0.0206
2020-11-20	3	0.0314
2020-11-21	3	0.0636
2020-11-13	4	0.0561
2020-11-10	3	0.0378
2020-11-18	2	0.0139
2020-11-17	2	0.0325
2020-11-16	2	0.0475
2020-11-15	2	0.0378
2020-11-12	2	0.0358
2020-11-11	2	0.0483
2020-11-02	2	0.0194
2020-11-03	1	0.0089
2020-11-19	1	0.0292

Result-

The number of jobs reviewed per day and also per hour per day is shown in the result image for the different dates of November month of 2020.

2. **Throughput:** Number of events happening per second.

SQL Query-

```
-- calculating 7-day rolling average of throughput

WITH job_data_cte AS
        ( SELECT ds, count(job_id) as Jobs_per_day,
sum(time_spent) as time_per_day
        FROM job_data
  GROUP BY ds
  )
SELECT
        ds, Jobs_per_day, SUM(Jobs_per_day)
OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING
AND CURRENT ROW) /
  SUM(time_per_day) OVER (ORDER BY ds ROWS
BETWEEN 6 PRECEDING AND CURRENT ROW)
  AS rolling_avg_7d FROM job_data_cte
```

ds	Jobs_per_day	rolling_avg_7d
2020-11-02	2	0.0286
2020-11-03	1	0.0294
2020-11-10	3	0.0252
2020-11-11	2	0.0194
2020-11-12	2	0.0185
2020-11-13	4	0.0188
2020-11-15	2	0.0182
2020-11-16	2	0.0163
2020-11-17	2	0.0160
2020-11-18	2	0.0163
2020-11-19	1	0.0165
2020-11-20	3	0.0179
2020-11-21	3	0.0163
2020-11-23	1	0.0163
2020-11-24	1	0.0169
2020-11-25	1	0.0173
2020-11-26	1	0.0157
2020-11-27	1	0.0157
2020-11-28	2	0.0161
2020-11-29	2	0.0190
2020-11-30	2	0.0227

Result-

7 days rolling throughput result is given in the image provided.

Almost always, metrics will fluctuate on a daily and weekly basis with little consideration for or connection to the actions you perform. Using a rolling average can help you eliminate this unwanted noise and get a better understanding of how your activities are impacting your metrics (7-day metric).

3. **Percentage share of each language:** Share of each language.

SQL Query-

```
-- percentage share of each language in the last
30 days

WITH job_num_cte AS
(
        SELECT language, COUNT(job_id) AS
num_jobs
        FROM job_data
        WHERE ds BETWEEN "2020-11-01" AND
"2020-11-30"
        GROUP BY language
        ),
total_jobs_cte AS
(
        SELECT COUNT(job_id) AS total_jobs
  FROM job_data
  WHERE ds BETWEEN "2020-11-01" AND "2020-
11-30"
  )
 SELECT language,
ROUND(100.0*num_jobs/total_jobs,2) AS
Jobs_perecentage
 FROM job_num_cte
 CROSS JOIN total_jobs_cte
 ORDER BY  Jobs_perecentage DESC
```

Result Grid | Filter Rows: | Export:

language	Jobs_perecentage
Persian	37.50
English	15.00
Arabic	15.00
Hindi	12.50
French	10.00
Italian	10.00

Result-

Share of each language of November 2020 is derived in the result. With 37.50% Persian language came out on top and was used most often.

4. **Duplicate Rows:** Rows that have the same values present in them

SQL Query-

```
-- displaying duplicate in the table

WITH duplicate_val AS
(
        SELECT *,

        ROW_NUMBER() OVER (PARTITION BY ds, job_id, actor_id, event, language) AS rownum
   FROM job_data
   )
SELECT
        *
FROM
        duplicate_val
WHERE
        rownum > 1
```

Here, we can see that 0 rows are returned which means no duplicate row is found.

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ds	job_id	actor_id	event	language	time_spent	org	rownum

340905 21:29:27 WITH duplicate_val AS (SELECT *, ROW_NUMBER() OVER (PARTITION BY ds, job_id, actor_id, e... 0 row(s) returned

Here, is the result snippet of column "rownum"-

ds	job_id	actor_id	event	language	time_spent	org	rownum
2020-11-02	14	1016	transfer	Arabic	18	B	1
2020-11-02	21	1020	decision	Persian	52	D	1
2020-11-03	16	1011	decision	Persian	32	C	1
2020-11-10	12	1006	skip	Persian	29	A	1
2020-11-10	21	1019	decision	French	63	D	1
2020-11-10	27	1007	skip	Persian	44	C	1
2020-11-11	25	1018	skip	English	106	A	1
2020-11-11	29	1011	transfer	Arabic	68	C	1
2020-11-12	18	1020	transfer	Italian	102	C	1
2020-11-12	23	1010	skip	English	27	B	1
2020-11-13	12	1020	skip	Persian	46	A	1
2020-11-13	16	1016	transfer	Italian	21	A	1
2020-11-13	24	1019	transfer	Persian	68	D	1
2020-11-13	27	1011	decision	French	67	D	1

Result- if for any row, the value of "rownum" is greater than 1 then we would have duplicate rows in the table. Here now value is greater than 1 hence, no duplicate row is found in the table

Case Study -2 Investigating Metric Spike

For this case study, we used 3 tables- users, events, and email_events.

1. **User Engagement:** To measure whether the user is active or not or measure if the user finds quality in products/services.

SQL Query-

```
-- calculating weekly user engagement

SELECT
        WEEK(e.occurred_at) AS week_num,
   COUNT(DISTINCT e.user_id) AS
weekly_active_users
FROM
        events AS e
WHERE
        e.event_type = "engagement" AND
e.event_name = "login"
GROUP BY 1
ORDER BY 1
```

week_num	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

Results-

Weekly user engagement with total number of users on that particular week is displayed in the result.

2. **User Growth:** Amount of users growing over time for a product.

SQL Query-

```
-- calculating user growth for a product over time
(monthly)

WITH month_users AS (SELECT
        month(u.created_at) AS month_num,
        COUNT(*) AS all_users
FROM
        users AS u
GROUP BY 1
ORDER BY 1
)
SELECT *,
all_users - LAG(all_users) OVER(ORDER BY
month_num ASC) AS users_growth
FROM month_users
```

Result Grid | Filter Rows:

month_num	all_users	users_growth
1	1415	NULL
2	1382	-33
3	1614	232
4	1829	215
5	2083	254
6	2213	130
7	2591	378
8	2626	35
9	699	-1927
10	826	127
11	816	-10
12	972	156

Results- Calculated user growth/dip monthly and the result is displayed in the table.

3. **Weekly Retention:** Number of users getting retained weekly after signing up for a product.

SQL Query-

```
-- Calculating weekly retention of users-signup cohort

WITH new_table AS (
SELECT
        us.user_id,
   us.created_at,
   -- WEEK(us.created_at) AS created_at_week,
   e.occurred_at,
   -- WEEK(e.occurred_at) AS occurred_at_week,

datediff(e.occurred_at,us.created_at) AS user_age,
   e.event_name
FROM
        users AS us
JOIN
        events AS e
   ON us.user_id = e.user_id
)
SELECT
        WEEK(u.occurred_at) AS week_num,
   count(CASE WHEN (u.user_age >= 0 AND u.user_age <8) then u.user_id ELSE null END) AS "1 week",
   count(CASE WHEN (u.user_age > 7 AND u.user_age <15) then u.user_id ELSE null END) AS "2 weeks",
   count(CASE WHEN (u.user_age > 14 AND u.user_age <22) then u.user_id ELSE null END) AS "3 weeks",
   count(CASE WHEN (u.user_age > 21 AND u.user_age <29) then u.user_id ELSE null END) AS "4 weeks",
   count(CASE WHEN (u.user_age > 28 AND u.user_age <36) then u.user_id ELSE null END) AS "5 weeks",
```

week_num	1 week	2 weeks	3 weeks	4 weeks	5 weeks	6 weeks	7 weeks	8 weeks	9 weeks	10 weeks	10+ weeks
19	432	150	94	81	50	53	41	44	43	93	1014
20	453	144	105	71	59	52	46	38	49	101	1040
21	424	148	103	68	45	46	43	46	43	109	974
22	470	160	113	84	65	29	36	36	34	98	1088
23	508	164	109	72	52	46	27	24	29	68	1129
24	535	166	135	66	56	39	29	16	20	65	1165
25	484	181	158	79	50	41	35	35	37	82	1117
26	490	172	124	99	48	47	37	28	31	71	1161
27	514	206	118	80	81	53	49	33	38	97	1184
17	162	81	45	51	35	30	14	14	12	33	428
18	397	165	144	72	59	46	42	30	44	87	959
30	576	172	143	97	66	47	47	52	49	111	1297
31	510	161	114	82	74	45	37	42	32	85	1150
32	520	157	121	69	73	61	49	25	25	71	962
28	534	182	136	86	69	60	61	45	33	79	1261
33	610	143	110	87	36	35	52	32	32	73	909
34	630	172	89	89	55	47	27	36	28	56	859
29	532	183	118	84	64	60	54	36	43	86	1230
35	61	8	8	1	3	1	0	0	0	0	22

count(CASE WHEN (u.user_age > 35 AND u.user_age <43) then u.user_id ELSE null END) AS "6 weeks", count(CASE WHEN (u.user_age > 42 AND u.user_age <50) then u.user_id ELSE null END) AS "7 weeks", count(CASE WHEN (u.user_age > 49 AND u.user_age <57) then u.user_id ELSE null END) AS "8 weeks", count(CASE WHEN (u.user_age > 56 AND u.user_age <64) then u.user_id ELSE null END) AS "9 weeks", count(CASE WHEN (u.user_age > 53 AND u.user_age <71) then u.user_id ELSE null END) AS "10 weeks", count(CASE WHEN (u.user_age > 70) then u.user_id ELSE null END) AS "10+ weeks" FROM new_table AS u WHERE u.event_name = "login" GROUP BY 1	

Results- Calculated weekly retention of user sign-up cohort and the result is displayed in the pic.

4. **Weekly Engagement:** To measure the activeness of users weekly for a product.

SQL Query-

```
-- calculating weekly user engagement per device

SELECT
        WEEK(occurred_at) AS week_num,
        count(CASE WHEN device in ('acer aspire desktop', 'dell inspiron desktop', 'hp pavilion desktop') then user_id ELSE null END) AS desktop_users,
   count(CASE WHEN device in ('macbook pro','lenovo thinkpad','dell inspiron notebook','macbook air', 'asus chromebook') then user_id ELSE null END) AS laptop_users,
   count(CASE WHEN device in ('iphone 5s','samsung galaxy note','nokia lumia 635','nexus 7','amazon fire phone', 'nexus 5','iphone 4s','htc one','iphone 5','samsung galaxy s4') then user_id ELSE null END) AS phone_users,
   count(CASE WHEN device in ('kindle fire','acer aspire notebook','mac mini','ipad mini','nexus 10', 'samsumg galaxy tablet','windows surface','ipad air') then user_id ELSE null END) AS tablet_users,
   count(user_id) AS total_users
FROM
        events AS e
GROUP BY 1
ORDER BY 1
```

week_num	desktop_users	laptop_users	phone_users	tablet_users	total_users
17	416	3705	3088	1195	8404
18	1420	8535	5973	2314	18242
19	1099	8507	6126	2446	18178
20	1075	8625	6681	2485	18866
21	1439	8097	6126	2450	18112
22	1423	8706	6735	2591	19455
23	1497	8624	6627	2597	19345
24	1797	8619	6977	2817	20210
25	1580	8449	6948	2740	19717
26	1586	9190	6876	2474	20126
27	1494	9409	7294	2824	21021
28	1755	9884	7264	3005	21908
29	1595	9512	7121	3005	21233
30	1639	10427	7703	3006	22775
31	1509	9642	5995	2439	19585
32	1502	8916	5293	2161	17872
33	1229	9143	4908	2165	17445
34	1171	8769	5202	2324	17466
35	25	462	221	164	872

Results- Calculated weekly active users per device (devices are grouped categorically) and the result is displayed in the table. The above result can be found by running the query mentioned.

5. **Email Engagement:** Number of users engaging with the email service.

SQL Query-

```
-- calculating email engagement metrics

SELECT
        WEEK(e.occurred_at) AS week1,
        COUNT(CASE WHEN e.action =
"sent_weekly_digest" THEN e.user_id ELSE
NULL END) AS weekly_digest_emails,
   COUNT(CASE WHEN e.action =
"sent_reengagement_email" THEN
e.user_id ELSE NULL END) AS
reengagement_emails,
   COUNT(CASE WHEN e.action =
"email_open" THEN e.user_id ELSE NULL
END) AS email_opens,
   COUNT(CASE WHEN e.action =
"email_clickthrough" THEN e.user_id  ELSE
NULL END) AS email_clickthroughs
FROM
        email_events AS e
GROUP BY 1
```

week1	weekly_digest_emails	reengagement_emails	email_opens	email_clickthroughs
18	2602	157	912	430
19	2665	173	972	477
20	2733	191	1004	507
21	2822	164	1014	443
22	2911	192	987	488
23	3003	197	1075	538
24	3105	226	1155	554
25	3207	196	1096	530
26	3302	219	1165	556
27	3399	213	1228	621
28	3499	213	1250	599
29	3592	213	1219	590
30	3706	231	1383	630
31	3793	222	1351	445
32	3897	200	1337	418
33	4012	264	1432	490
34	4111	261	1528	490
17	908	73	310	166
35	0	48	41	38

Results- Calculated email engagement metrics and the result is displayed in the table.

Project Conclusion

While analyzing the dataset provided, several meaningful insights were discovered that could not have been discovered by manually searching the dataset for insights. These insights can help investors and marketing teams make better judgments in the future, saving them a good amount of time and money.

We could also leverage the MySQL tool and got a little more experienced in using the tool and also injecting different types of queries to look for insights.