# Operation Analytics and Investigating Metric Spike

## Project Description:

The purpose of this project is to analyse the company's end to end operations. The primary objective is to work closely with various teams like operations, support and marketing to derive valuable insights from various datasets to improve operational efficiency and understand sudden changes in key metrics and also using advanced SQL skills to investigate and explain metric spikes such as fluctuation in user engagement or sales figures. The analysis will help organization to proactively address the challenges and ultimately leading to continuous growth and improvement. The project will address the problem statements in two case studies as given below:

A) **Case Study 1: Job Data Analysis:** The project will analyse job data to improve operational efficiency and derive insights from the following problem statements posed by different departments within the company:

1. **Jobs reviewed over time.**
2. **Throughput analysis.**
3. **Language share analysis.**
4. **Duplicate rows detection.**

B) **Case Study 2: Investigating metric spike:** Using the datasets provided, the project will explain the key metric spikes by analysing and providing insights of the following problem statements:

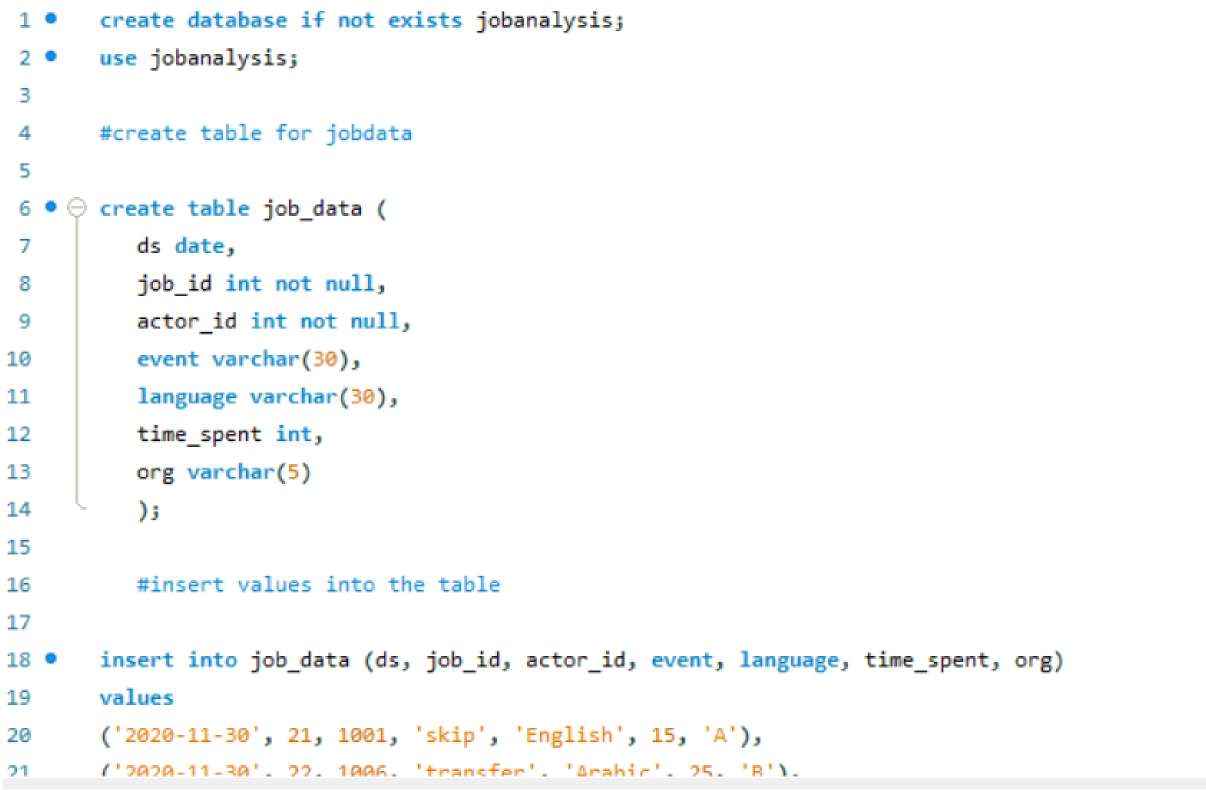
1. **Weekly user engagement.**
2. **User growth analysis.**
3. **Weekly retention analysis.**
4. **Weekly engagement per device.**
5. **Email engagement analysis.**

## Approach:

To accomplish the necessary tasks and finalize the project, SQL queries were employed using the MySQL Command Line client. Following provided instructions, the database and corresponding tables were created, data was imported into MySQL, and relevant queries were executed to derive the required insights.

1. **Creating database:** The dataset file for both the case studies has been provided by the team. The database has been created using DML and DDL SQL queries. The following analyzation can be done from the provided database:
   **Case Study 1 (Job data analysis):** SQL codes have been used to create table called job_data. The table consists some relevant column for analysis as given below:
   - job_id: Unique identifier of jobs.
   - actor_id: Unique identifier of actor.
   - event: The type of event (decision/skip/transfer).

- language: The language of the content.
- time_spent: Time spent to review the job in seconds.
- org: The organization of the actor.
- ds: The date in the format yyyy/mm/dd (stored as text).

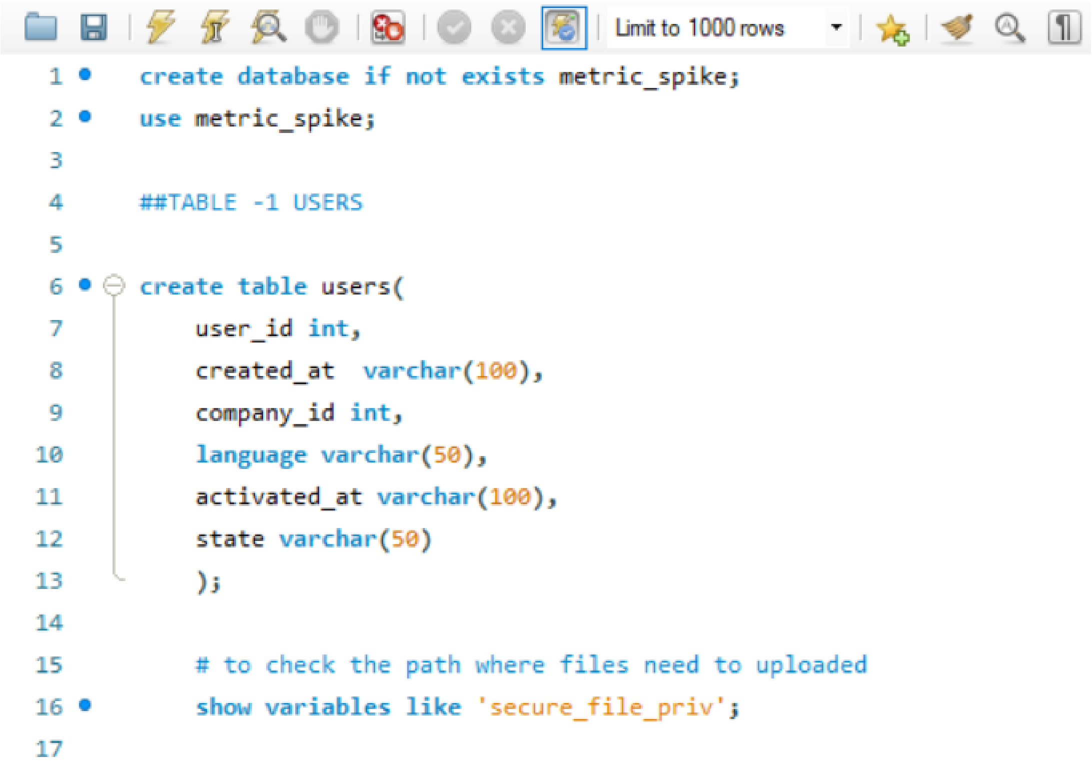


```
create database if not exists jobanalysis;
use jobanalysis;

#create table for jobdata

create table job_data (
   ds date,
   job_id int not null,
   actor_id int not null,
   event varchar(30),
   language varchar(30),
   time_spent int,
   org varchar(5)
   );

   #insert values into the 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'),
```

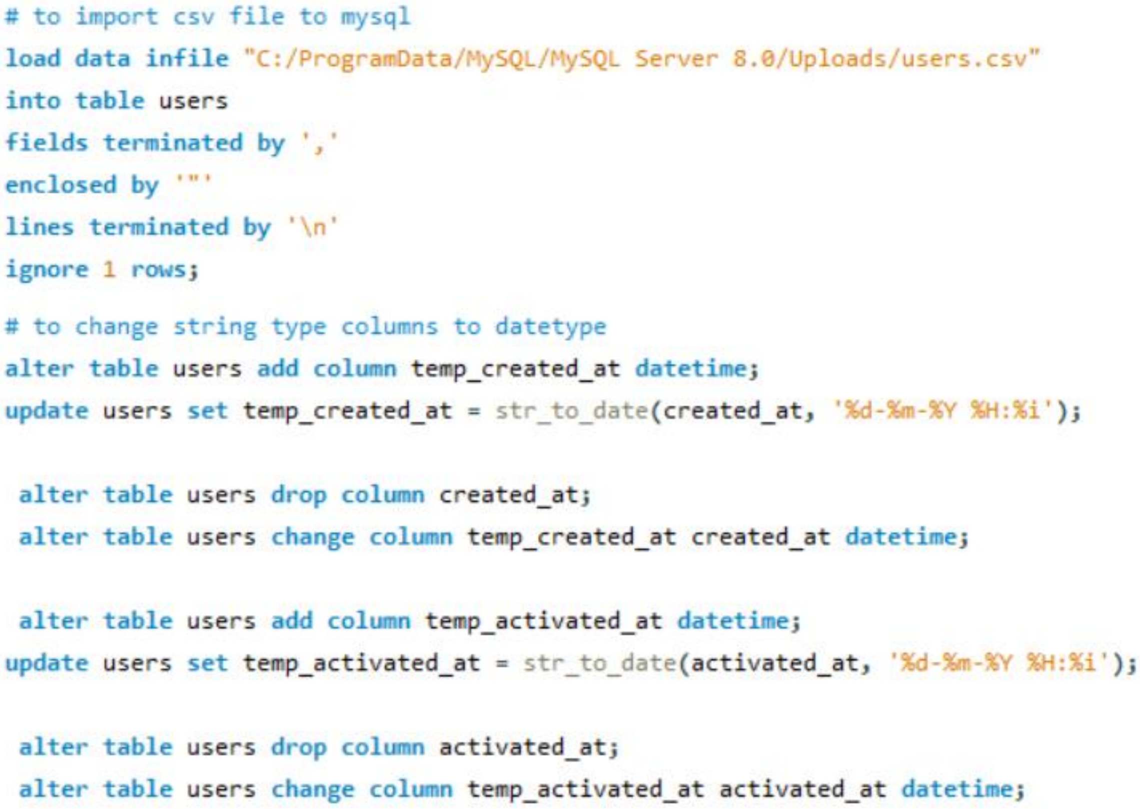
**Case Study 2 (Investigating metric spike):** SQL queries have been used to create three tables to work on. They are:

- users: Contains one row per user, with descriptive information about that users account.
- events: Contains one row per event, where an event is an action that a user has taken.
- Email_events: Contains events specific to the sending of emails.

Each table consists of some relevant columns for analysis. The datasets for all the three tables were provided in csv file. After creating all the three tables the provided csv file was imported into MySQL Workbench.



```sql
create database if not exists metric_spike;
use metric_spike;

##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)
    );

    # to check the path where files need to uploaded
    show variables like 'secure_file_priv';

```



```sql
# to import csv file to mysql
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;

# to change string type columns to datetype
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;

 alter table users add column temp_activated_at datetime;
update users set temp_activated_at = str_to_date(activated_at, '%d-%m-%Y %H:%i');

 alter table users drop column activated_at;
 alter table users change column temp_activated_at activated_at datetime;
```

Similarly, events and email_events table were created and datasets were imported into MySQL Workbench. Necessary changes were made in the tables after importing the datasets.

2. **Derive insights:** After creating the database, key insights were derived from the tables using SQL queries for both the case studies.

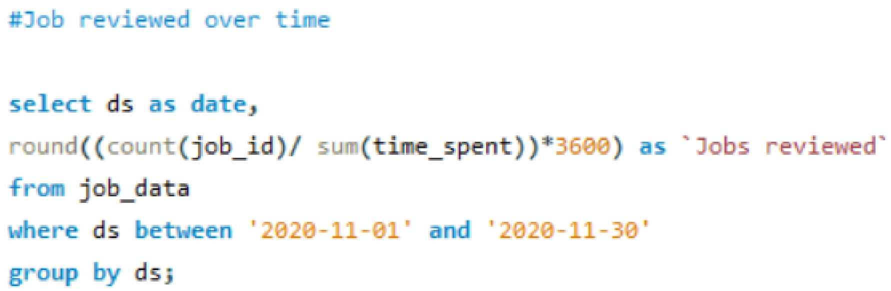
- ## Tech-Stack used:

**MySQL 8.0.37 community version** has been used to create database and to derive key insights. The reason behind using MySQL is that it is the most popular relational database management system (RDBMS) which is easy to use and understand and also the community version is free and open-source. The tool helps in writing and running SQL queries to extract specific data which is needed for analysis. The tool not only helps analyst explore the structure of database bust also gives faster results. It also ensures data security making it reliable for handling large datasets and supports various analysis tasks which helps in generating insights that improve company operations.

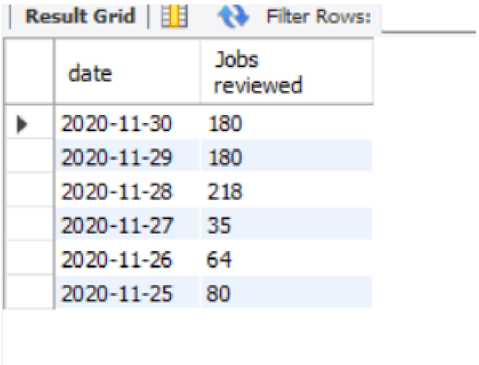
- ## Insights:

### A) Case Study 1 (Job data analysis):

1. **Jobs reviewed over time:** The task is to calculate the number of jobs reviewed per hour for each day in November 2020.



```
#Job reviewed over time

select ds as date,
round((count(job_id)/ sum(time_spent))*3600) as `Jobs reviewed`
from job_data
where ds between '2020-11-01' and '2020-11-30'
group by ds;
```

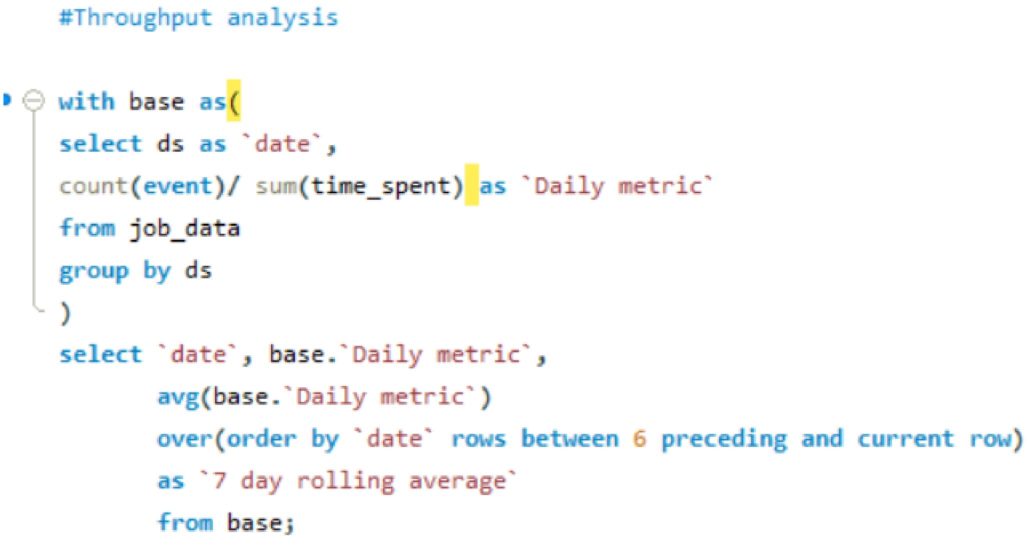


| date | Jobs reviewed |
|---|---|
| 2020-11-30 | 180 |
| 2020-11-29 | 180 |
| 2020-11-28 | 218 |
| 2020-11-27 | 35 |
| 2020-11-26 | 64 |
| 2020-11-25 | 80 |

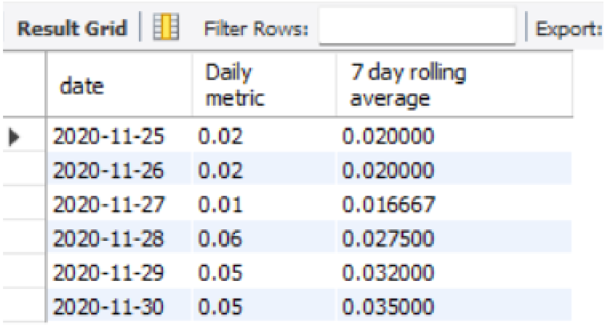
**Insights:** From the data gathered above it can be observed that number of jobs reviewed per hour in November 2020 varies. The highest number of jobs were reviewed on 28 November 2020 with 218 jobs review per hour and lowest number of jobs were reviewed on 27 November 2020 with 35 jobs review per hour. It can also be observed that on average 126 jobs were reviewed per hour in November 2020.

Identifying the peak review days can help in scheduling tasks and staffing accordingly. High review counts indicate active user engagement and satisfaction whereas low counts might indicate some issues which needs attention. Overall analysing the above data will help in optimizing operations, improve user experience and overall performance.

2. **Throughput analysis:** The task is to calculate the 7-day rolling average of throughput and explain preference between using daily metric or the 7-day rolling average for throughput.



```
#Throughput analysis

with base as(
select ds as `date`,
count(event)/ sum(time_spent) as `Daily metric`
from job_data
group by ds
)
select `date`, base.`Daily metric`,
       avg(base.`Daily metric`)
       over(order by `date` rows between 6 preceding and current row)
       as `7 day rolling average`
       from base;
```

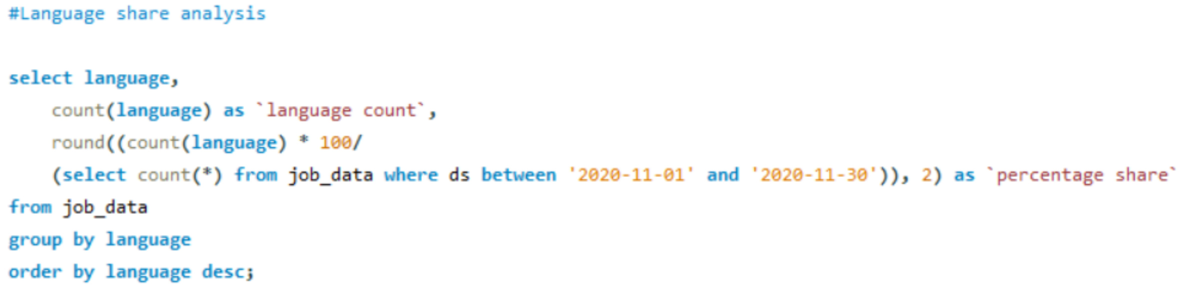


| date | Daily metric | 7 day rolling average |
|---|---|---|
| 2020-11-25 | 0.02 | 0.020000 |
| 2020-11-26 | 0.02 | 0.020000 |
| 2020-11-27 | 0.01 | 0.016667 |
| 2020-11-28 | 0.06 | 0.027500 |
| 2020-11-29 | 0.05 | 0.032000 |
| 2020-11-30 | 0.05 | 0.035000 |

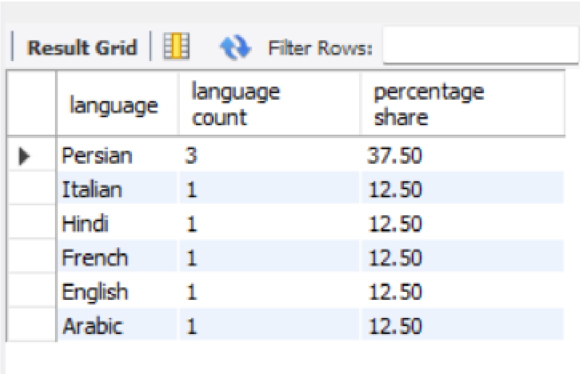
**Insights:** The above data shows daily metric throughput as well as 7-day rolling average of throughput. It can be observed in case of daily metric number of events per second for each day is between 0.01 and 0.06 approximately. It provides a clear picture of throughput which is useful for focusing specific days of high or low activity. It basically shows immediate changes.
In case of 7-day rolling average number of events per second is somewhat stable between 0.02 and 0.04 approximately. It basically smooths out daily fluctuations and provides a longer term and stable picture of throughput.
Although both metrics serve different purposes but 7 -day rolling average is preferred more because daily metric fluctuations can happen by factors which sometimes cannot be controlled by organizations like holidays or special events, etc whereas 7- day rolling

average minimizes the impact of such factors and provides a more comprehensive view for making strategic decisions and long-term planning.

3. **Language share analysis:** The task is to calculate the percentage share of each language over the last 30 days.



```
#Language share analysis

select language,
    count(language) as `language count`,
    round((count(language) * 100/
    (select count(*) from job_data where ds between '2020-11-01' and '2020-11-30')), 2) as `percentage share`
from job_data
group by language
order by language desc;
```

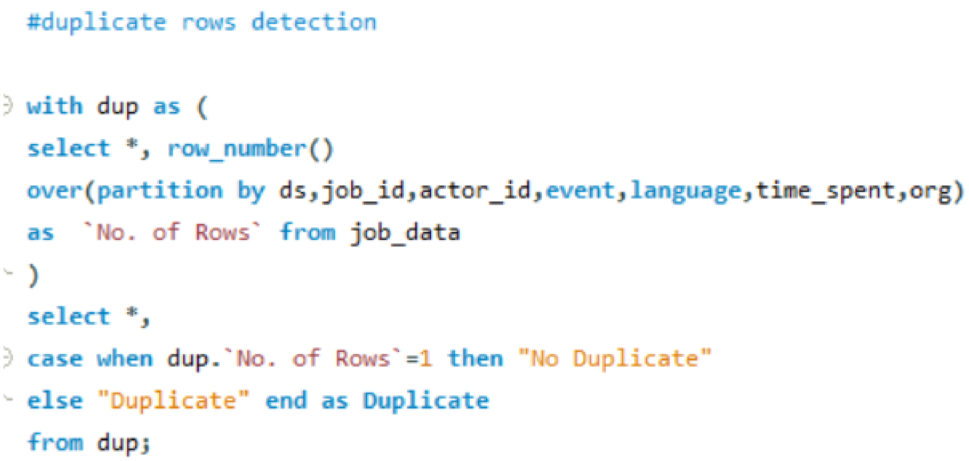


| | language | language count | percentage share |
|---|---|---|---|
| ▶ | Persian | 3 | 37.50 |
| | Italian | 1 | 12.50 |
| | Hindi | 1 | 12.50 |
| | French | 1 | 12.50 |
| | English | 1 | 12.50 |
| | Arabic | 1 | 12.50 |

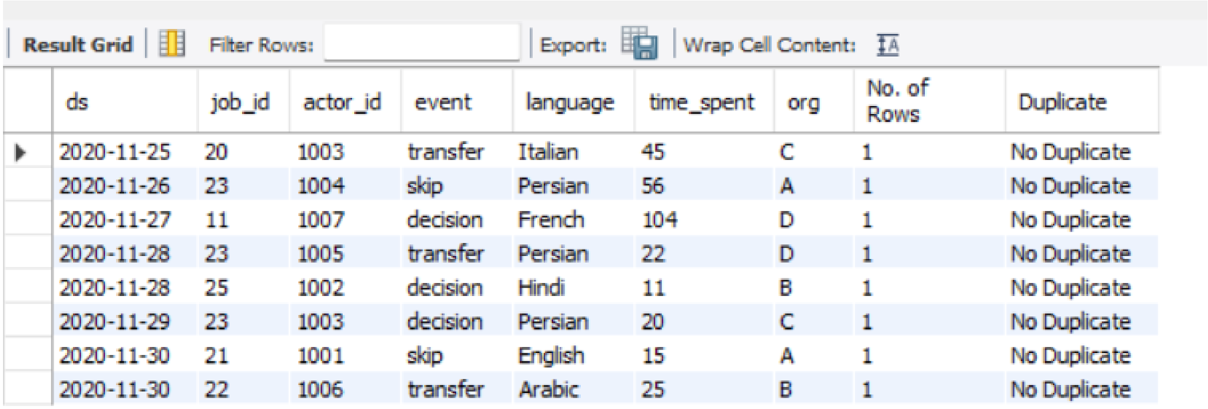
**Insights:** The above data reveals distribution of content and activities in different languages. From the above data it can be observed that Persian language has the highest percentage share of 37.5 whereas rest of the languages have the same percentage share of 12.5.

Analysing the above data shows that Persian language is much more in demand compared to other languages. Hence, allocating more support and resources like customer support, marketing efforts for Persian language can improve operational efficiency. If the language activity is content or feature based then the marketing team can focus more on user engagement and customer satisfaction in Persian language and simultaneously find ways to improve user engagement in other languages with lower share. Businesses can consider investing more in Persian dominated market to capitalize on growth opportunities.

4. **Duplicate rows detection:** The task is to display duplicate rows from the job_data table.



```
#duplicate rows detection

with dup as (
select *, row_number()
over(partition by ds,job_id,actor_id,event,language,time_spent,org)
as  `No. of Rows` from job_data
)
select *,
case when dup.`No. of Rows`=1 then "No Duplicate"
else "Duplicate" end as Duplicate
from dup;
```



| ds | job_id | actor_id | event | language | time_spent | org | No. of Rows | Duplicate |
|---|---|---|---|---|---|---|---|---|
| 2020-11-25 | 20 | 1003 | transfer | Italian | 45 | C | 1 | No Duplicate |
| 2020-11-26 | 23 | 1004 | skip | Persian | 56 | A | 1 | No Duplicate |
| 2020-11-27 | 11 | 1007 | decision | French | 104 | D | 1 | No Duplicate |
| 2020-11-28 | 23 | 1005 | transfer | Persian | 22 | D | 1 | No Duplicate |
| 2020-11-28 | 25 | 1002 | decision | Hindi | 11 | B | 1 | No Duplicate |
| 2020-11-29 | 23 | 1003 | decision | Persian | 20 | C | 1 | No Duplicate |
| 2020-11-30 | 21 | 1001 | skip | English | 15 | A | 1 | No Duplicate |
| 2020-11-30 | 22 | 1006 | transfer | Arabic | 25 | B | 1 | No Duplicate |

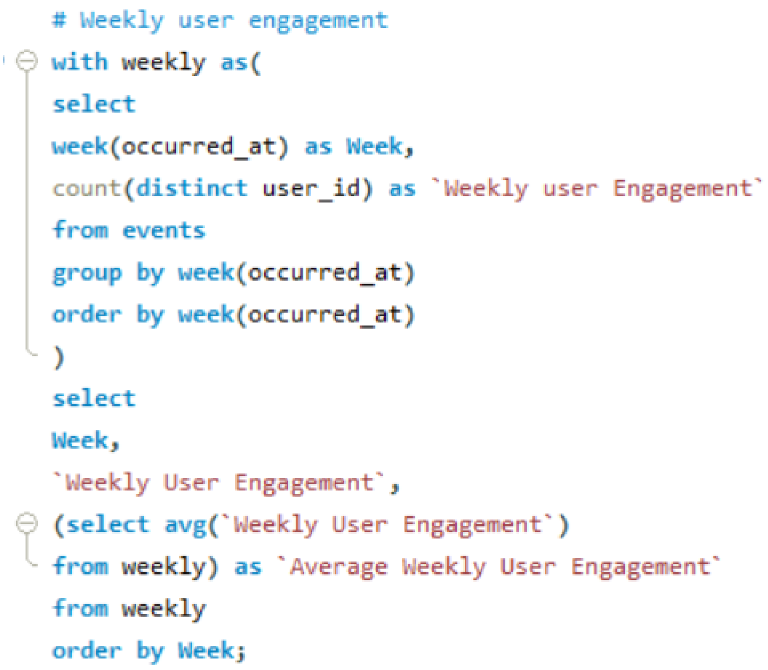
**Insights:** From the above data it can be observed that there are no duplicate rows and all rows are unique when entire table is considered. This means that there are no redundant data entries which helps in maintaining data consistency.

If ever duplicate rows are observed then fixing it becomes very crucial because time and resources will be wasted on processing redundant and unnecessary data. It can also distort metrics like event frequency, language distribution, time spent which will lead to unreliable analysis and insights of outcomes.

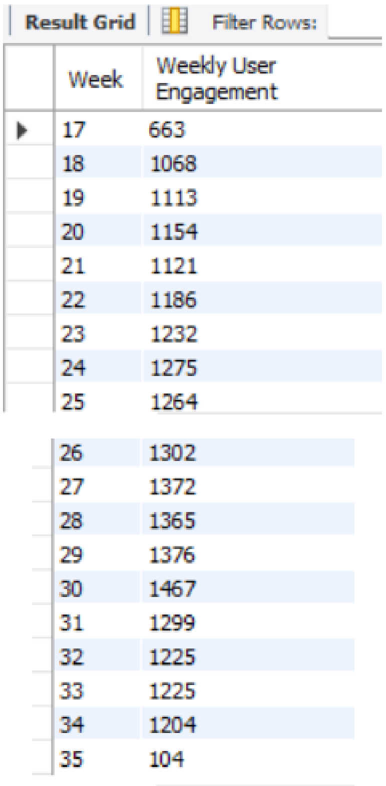
Hence addressing duplicate rows ensures that the data is clean and accurate which help in making informed business decisions.

B) <u>**Case Study 2 (Investigating metric spike):**</u>

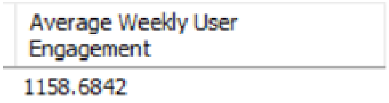
1. **Weekly user engagement:** The task is to measure the activeness of users on a weekly basis by calculating the weekly user engagement.



```
# Weekly user engagement
with weekly as(
select
week(occurred_at) as Week,
count(distinct user_id) as `Weekly user Engagement`
from events
group by week(occurred_at)
order by week(occurred_at)
)
select
Week,
`Weekly User Engagement`,
(select avg(`Weekly User Engagement`)
from weekly) as `Average Weekly User Engagement`
from weekly
order by Week;
```



| Week | Weekly User Engagement |
|---|---|
| 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 |



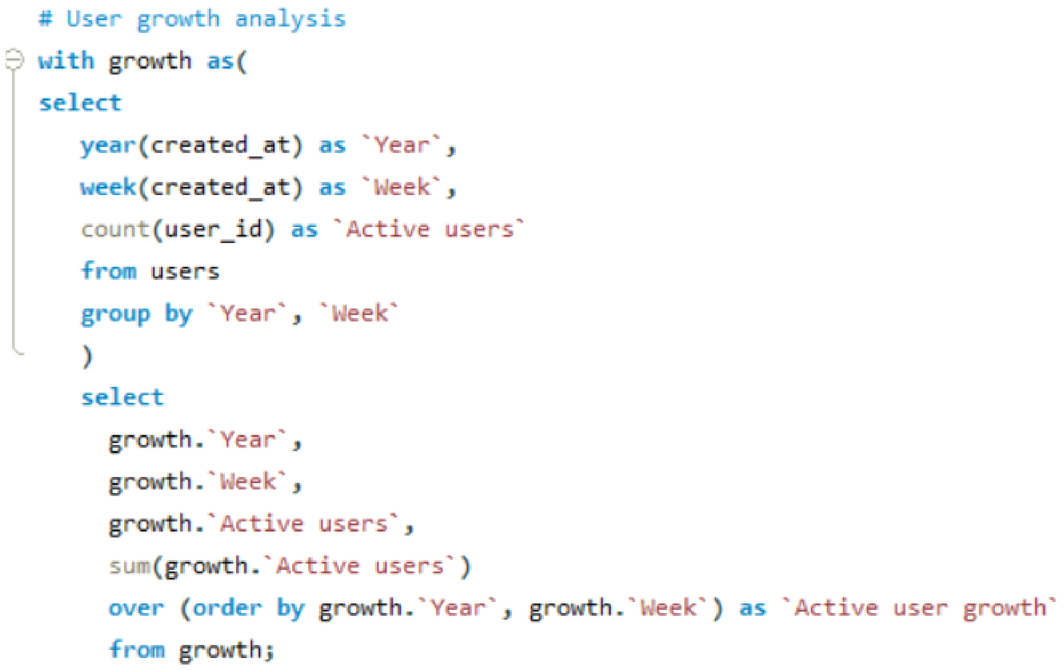
| Average Weekly User Engagement |
|---|
| 1158.6842 |

**Insights:** From the above data it can be observed that on average users engage in 1159 events approximately on weekly basis. The highest user engagement is observed around week 30 with 1467 users whereas the lowest engagement is observed on week 35 with 104 users.

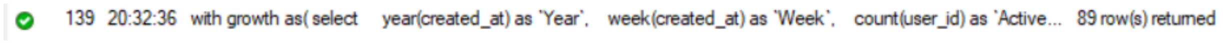
Analysing the above data helped in identifying the peak week when user engagement is highest which will help in scheduling important updates and launch marketing campaigns or promotional events to maximize engagement and visibility. It will also help in analysing the type of event which attracts the most user.

Low user engagement during specific weeks can occur due to certain external factors such as holidays, vacations or events that divert user's attention away from the specified platform. Another reason for low engagement can be certain technical glitches that might have occurred during those weeks. The team can review user feedback during those low engagement weeks and can analyse why users were less engaged during those weeks.

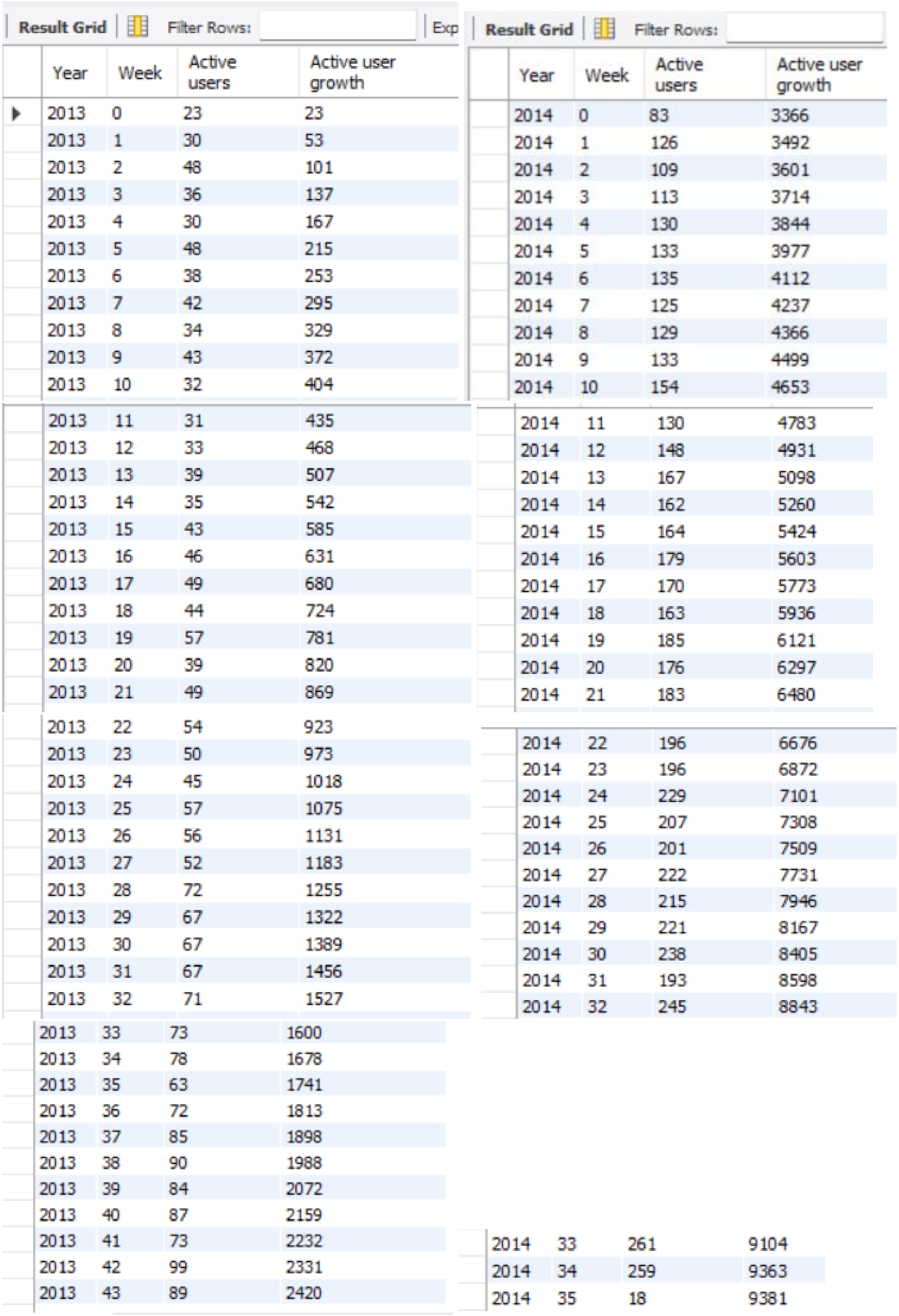
2. **User growth analysis:** The task is to analyse the growth of users over time for a product by calculating the user growth for the product.



```
# User growth analysis
with growth as(
select
   year(created_at) as `Year`,
   week(created_at) as `Week`,
   count(user_id) as `Active users`
   from users
   group by `Year`, `Week`
   )
   select
     growth.`Year`,
     growth.`Week`,
     growth.`Active users`,
     sum(growth.`Active users`)
     over (order by growth.`Year`, growth.`Week`) as `Active user growth`
     from growth;
```

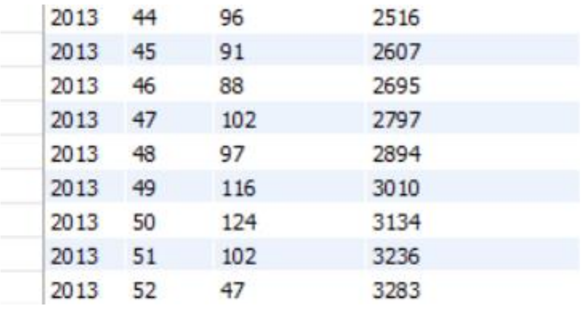


```
139  20:32:36  with growth as( select    year(created_at) as `Year`,    week(created_at) as `Week`,    count(user_id) as `Active...   89 row(s) returned
```



| Year | Week | Active users | Active user growth |
|---|---|---|---|
| 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 |
| 2013 | 16 | 46 | 631 |
| 2013 | 17 | 49 | 680 |
| 2013 | 18 | 44 | 724 |
| 2013 | 19 | 57 | 781 |
| 2013 | 20 | 39 | 820 |
| 2013 | 21 | 49 | 869 |
| 2013 | 22 | 54 | 923 |
| 2013 | 23 | 50 | 973 |
| 2013 | 24 | 45 | 1018 |
| 2013 | 25 | 57 | 1075 |
| 2013 | 26 | 56 | 1131 |
| 2013 | 27 | 52 | 1183 |
| 2013 | 28 | 72 | 1255 |
| 2013 | 29 | 67 | 1322 |
| 2013 | 30 | 67 | 1389 |
| 2013 | 31 | 67 | 1456 |
| 2013 | 32 | 71 | 1527 |
| 2013 | 33 | 73 | 1600 |
| 2013 | 34 | 78 | 1678 |
| 2013 | 35 | 63 | 1741 |
| 2013 | 36 | 72 | 1813 |
| 2013 | 37 | 85 | 1898 |
| 2013 | 38 | 90 | 1988 |
| 2013 | 39 | 84 | 2072 |
| 2013 | 40 | 87 | 2159 |
| 2013 | 41 | 73 | 2232 |
| 2013 | 42 | 99 | 2331 |
| 2013 | 43 | 89 | 2420 |

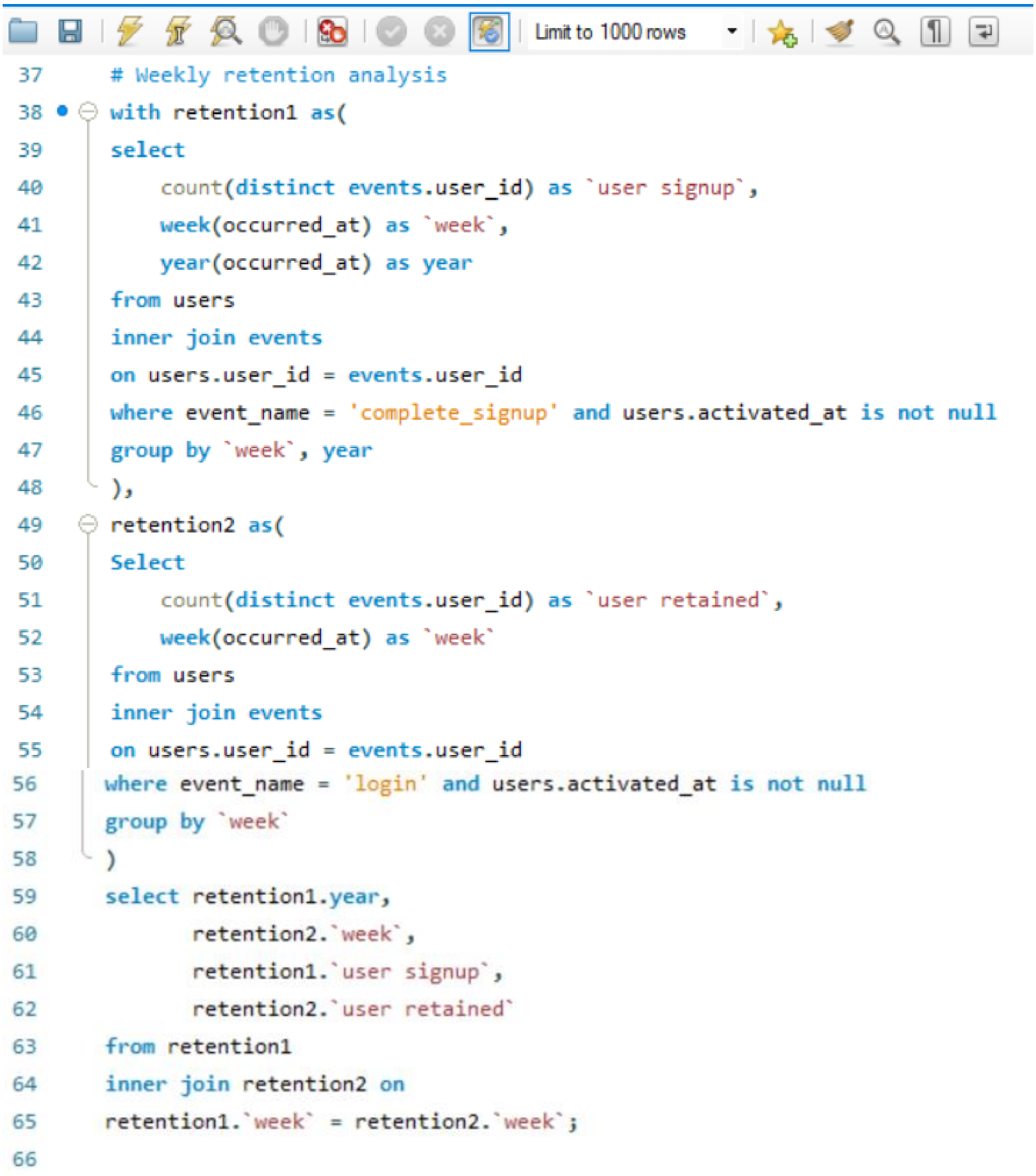
| Year | Week | Active users | Active user growth |
|---|---|---|---|
| 2014 | 0 | 83 | 3366 |
| 2014 | 1 | 126 | 3492 |
| 2014 | 2 | 109 | 3601 |
| 2014 | 3 | 113 | 3714 |
| 2014 | 4 | 130 | 3844 |
| 2014 | 5 | 133 | 3977 |
| 2014 | 6 | 135 | 4112 |
| 2014 | 7 | 125 | 4237 |
| 2014 | 8 | 129 | 4366 |
| 2014 | 9 | 133 | 4499 |
| 2014 | 10 | 154 | 4653 |
| 2014 | 11 | 130 | 4783 |
| 2014 | 12 | 148 | 4931 |
| 2014 | 13 | 167 | 5098 |
| 2014 | 14 | 162 | 5260 |
| 2014 | 15 | 164 | 5424 |
| 2014 | 16 | 179 | 5603 |
| 2014 | 17 | 170 | 5773 |
| 2014 | 18 | 163 | 5936 |
| 2014 | 19 | 185 | 6121 |
| 2014 | 20 | 176 | 6297 |
| 2014 | 21 | 183 | 6480 |
| 2014 | 22 | 196 | 6676 |
| 2014 | 23 | 196 | 6872 |
| 2014 | 24 | 229 | 7101 |
| 2014 | 25 | 207 | 7308 |
| 2014 | 26 | 201 | 7509 |
| 2014 | 27 | 222 | 7731 |
| 2014 | 28 | 215 | 7946 |
| 2014 | 29 | 221 | 8167 |
| 2014 | 30 | 238 | 8405 |
| 2014 | 31 | 193 | 8598 |
| 2014 | 32 | 245 | 8843 |
| 2014 | 33 | 261 | 9104 |
| 2014 | 34 | 259 | 9363 |
| 2014 | 35 | 18 | 9381 |



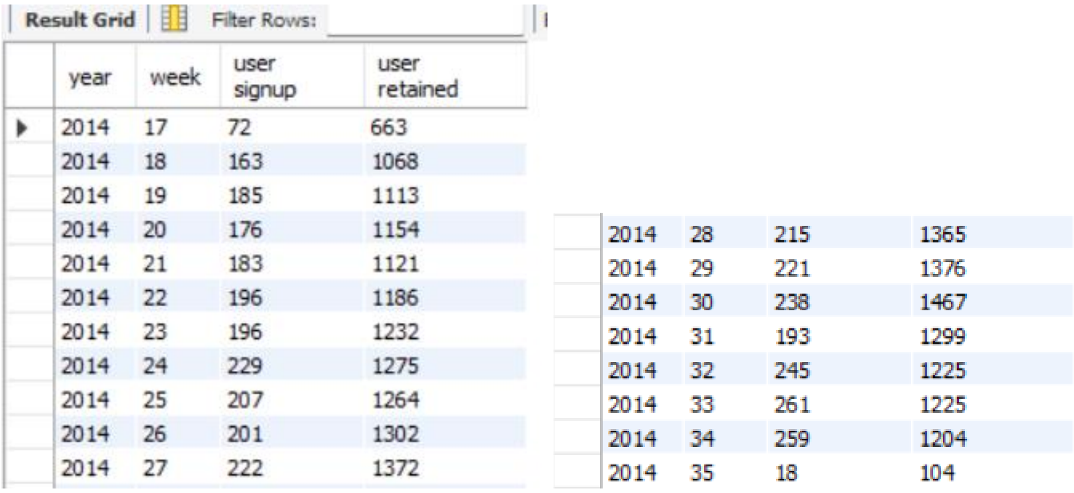
| | | | |
|---|---|---|---|
| 2013 | 44 | 96 | 2516 |
| 2013 | 45 | 91 | 2607 |
| 2013 | 46 | 88 | 2695 |
| 2013 | 47 | 102 | 2797 |
| 2013 | 48 | 97 | 2894 |
| 2013 | 49 | 116 | 3010 |
| 2013 | 50 | 124 | 3134 |
| 2013 | 51 | 102 | 3236 |
| 2013 | 52 | 47 | 3283 |

**Insights:** The above data shows growth of active users in 2013 and 2014 respectively.
In the year 2013 it is observed that total number of active users is 3283. The highest number of active user registration is seen in week 50 which is 124.
In the year 2014 it is observed that total number of active users is 9381. The highest number of active user registration is observed in week 34 which is 259. It is also observed that there is sharp drop in user registration in week 35 with only 18 new users being registered that week which is the lowest in that whole year.
Overall the user growth from year 2013 to 2014 has generally been positive over weeks with some fluctuations. This positive growth indicates a good sign for the organization. The team can focus on the factors which stimulated the user growth over time by analyzing both higher growth periods as well as weeks where growth was low compared to other weeks. The team can analyze which particular feature update or trending contents and marketing campaigns lead to the user growth. This can help oragnization in sustainable growth and success for their service as well as platform.

3. **Weekly retention analysis:** The task is to analyse the retention of users on a weekly basis after signing up for a product by calculating the weekly retention of users based on their sign-up cohort.



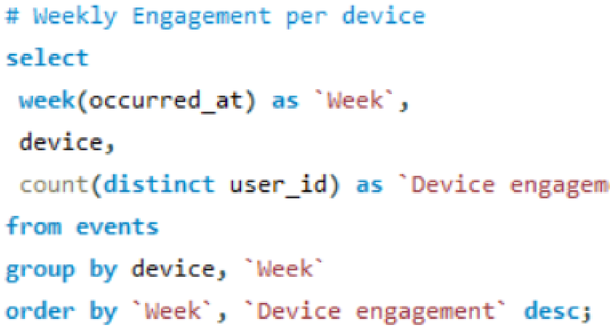
```
# Weekly retention analysis
with retention1 as(
select
    count(distinct events.user_id) as `user signup`,
    week(occurred_at) as `week`,
    year(occurred_at) as year
from users
inner join events
on users.user_id = events.user_id
where event_name = 'complete_signup' and users.activated_at is not null
group by `week`, year
),
retention2 as(
Select
    count(distinct events.user_id) as `user retained`,
    week(occurred_at) as `week`
from users
inner join events
on users.user_id = events.user_id
where event_name = 'login' and users.activated_at is not null
group by `week`
)
select retention1.year,
       retention2.`week`,
       retention1.`user signup`,
       retention2.`user retained`
from retention1
inner join retention2 on
retention1.`week` = retention2.`week`;
```



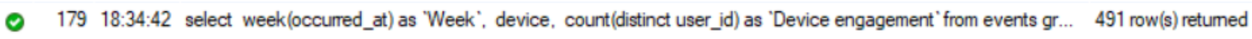
| year | week | user signup | user retained |
|---|---|---|---|
| 2014 | 17 | 72 | 663 |
| 2014 | 18 | 163 | 1068 |
| 2014 | 19 | 185 | 1113 |
| 2014 | 20 | 176 | 1154 |
| 2014 | 21 | 183 | 1121 |
| 2014 | 22 | 196 | 1186 |
| 2014 | 23 | 196 | 1232 |
| 2014 | 24 | 229 | 1275 |
| 2014 | 25 | 207 | 1264 |
| 2014 | 26 | 201 | 1302 |
| 2014 | 27 | 222 | 1372 |
| 2014 | 28 | 215 | 1365 |
| 2014 | 29 | 221 | 1376 |
| 2014 | 30 | 238 | 1467 |
| 2014 | 31 | 193 | 1299 |
| 2014 | 32 | 245 | 1225 |
| 2014 | 33 | 261 | 1225 |
| 2014 | 34 | 259 | 1204 |
| 2014 | 35 | 18 | 104 |

**Insights:** From the above data it can be observed that weekly user retention gradually increases till week 30 and then gradually start decreasing from week 31 onwards. There is a sharp drop observed in week 35 with lowest number of user signup as well as user retained.
Higher retention rate till week 30 might indicate users quickly adopted as well as engaged with new features. It could also indicate that the platform offered relevant content that kept the user engaged and returning for more during those weeks.
The decline after week 30 might indicate issues with initial user experience or onboarding. It could also indicate continuous technical problems or downtime during those weeks that lead to user frustration. Also, irrelevant content that did not align with user interests lead to failed user engagement.
Analysing user feedback can help in identifying factors which impacted the user retention positively as well as negatively. By addressing those factors, the organization can improve user retention over time.

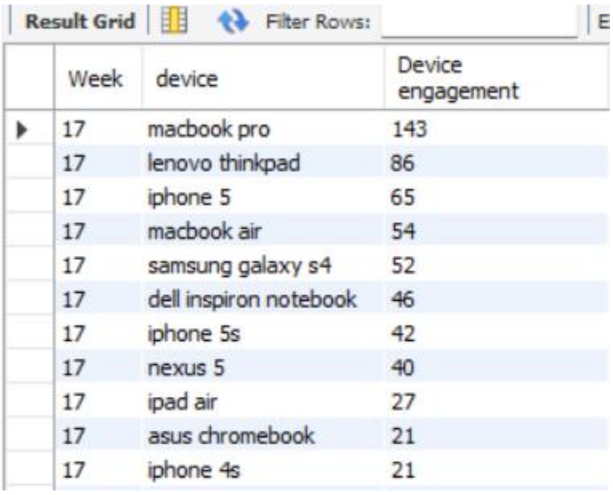
4. **Weekly engagement per device:** The task is to measure the activeness of users on a weekly basis per device by calculating the weekly engagement per device.



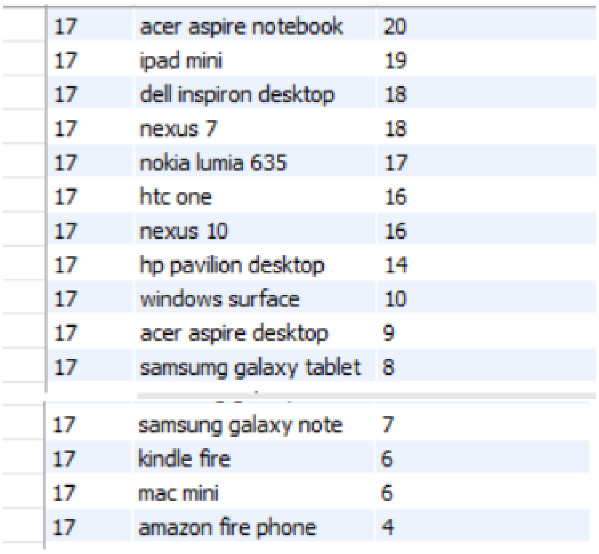
```
# Weekly Engagement per device
select
 week(occurred_at) as `Week`,
 device,
 count(distinct user_id) as `Device engagement`
from events
group by device, `Week`
order by `Week`, `Device engagement` desc;
```



```
179  18:34:42  select week(occurred_at) as `Week`, device, count(distinct user_id) as `Device engagement` from events gr...  491 row(s) returned
```



| Week | device | Device engagement |
|---|---|---|
| 17 | macbook pro | 143 |
| 17 | lenovo thinkpad | 86 |
| 17 | iphone 5 | 65 |
| 17 | macbook air | 54 |
| 17 | samsung galaxy s4 | 52 |
| 17 | dell inspiron notebook | 46 |
| 17 | iphone 5s | 42 |
| 17 | nexus 5 | 40 |
| 17 | ipad air | 27 |
| 17 | asus chromebook | 21 |
| 17 | iphone 4s | 21 |



| | | |
|---|---|---|
| 17 | acer aspire notebook | 20 |
| 17 | ipad mini | 19 |
| 17 | dell inspiron desktop | 18 |
| 17 | nexus 7 | 18 |
| 17 | nokia lumia 635 | 17 |
| 17 | htc one | 16 |
| 17 | nexus 10 | 16 |
| 17 | hp pavilion desktop | 14 |
| 17 | windows surface | 10 |
| 17 | acer aspire desktop | 9 |
| 17 | samsumg galaxy tablet | 8 |
| 17 | samsung galaxy note | 7 |
| 17 | kindle fire | 6 |
| 17 | mac mini | 6 |
| 17 | amazon fire phone | 4 |

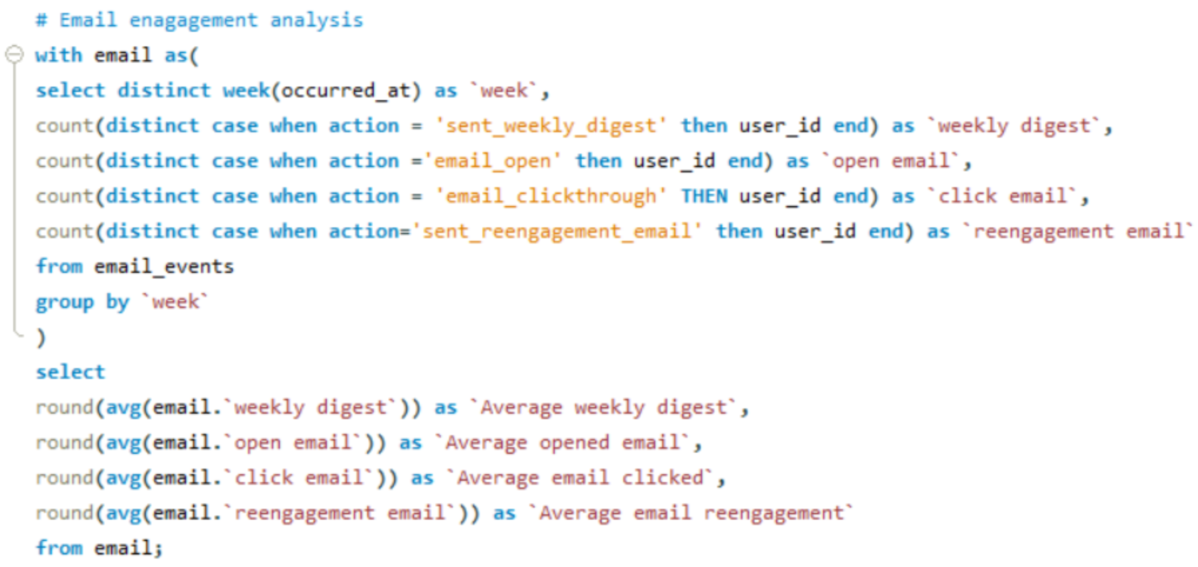
**Insights:** The above data shows weekly user engagement per device for week 17. The SQL query returns total 491 rows showing weekly user engagement per device from week 17 to week 35. The further analysis of user engagement per device is based on week 17.

From the above data it can be observed that MacBook pro has the highest device engagement whereas amazon fire phone has the lowest device engagement.

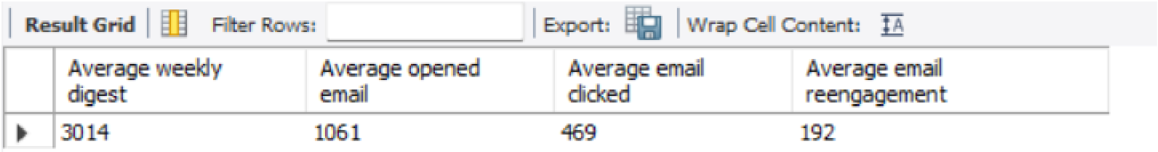
Since MacBook pro has highest device engagement, it indicates that it is predominantly used to access the platform compared to other devices. The reason for it can be that MacBook pro provides better user experience for the platform. The development team can make sure to fully optimise the platform for MacBook pro to maintain high performance and enhance visual experience.

Low user engagement on amazon fire phone might indicate that users using this device are not engaging with the platform as actively. The reason behind it can be compatibility issues on this device like display problems, difficulty with the interface. By addressing the issues for low engagement, the team can improve the overall user experience across all devices.

5. **Email engagement analysis:** The task is to analyse how users are engaging with email service by calculating the email engagement metrics.



```
# Email enagagement analysis
with email as(
select distinct week(occurred_at) as `week`,
count(distinct case when action = 'sent_weekly_digest' then user_id end) as `weekly digest`,
count(distinct case when action ='email_open' then user_id end) as `open email`,
count(distinct case when action = 'email_clickthrough' THEN user_id end) as `click email`,
count(distinct case when action='sent_reengagement_email' then user_id end) as `reengagement email`
from email_events
group by `week`
)
select
round(avg(email.`weekly digest`)) as `Average weekly digest`,
round(avg(email.`open email`)) as `Average opened email`,
round(avg(email.`click email`)) as `Average email clicked`,
round(avg(email.`reengagement email`)) as `Average email reengagement`
from email;
```



| Average weekly digest | Average opened email | Average email clicked | Average email reengagement |
|---|---|---|---|
| 3014 | 1061 | 469 | 192 |

**Insights:** From the above data it can be observed that the most email activity is related with weekly digest with an average of 3014. It indicates that an average of 3014 users received weekly email.

An average of 1061 users opened the email and an average of 469 users clicked on the links in the email. It can also be observed that an average of 192 users were sent re-engagement emails.

From the above metrics it can be analysed that conversion rate of email activity is below average. The average users who were sent the email and the average users who actually clicked through the email is quite low. These below average metrics points towards issues like poor subject lines, irrelevant content or poor product recommendation.

The team should analyse how different types of content impact email metrics like promotional offers, newsletter, educational content. Emails with personalised subject lines and holiday themed promotional offers achieve higher conversion rates which will help in improving email marketing strategies.

- **Results:**

Through this project as a Lead data analyst in Microsoft, I gained significant insights that proved to be valuable. It deepened my knowledge of data analysis in organisations and process of making data driven decisions. This experience helped in improving my analytical skills and SQL skills. I became comfortable in performing queries and analysing large datasets.

By analysing job data in case study 1, I found ways to make processes smoother and efficient. This included improving throughput, quickly reviewing job and identifying and fixing duplicate entries.

By investigating metric spike in case study 2 like user engagement and growth I gained valuable insights into what drives these changes which helped me in understanding what worked well and where improvement is needed.

Addressing these challenges not only helped in my problem-solving skills but also showed me the value of always looking to do better. I have become more confident in making decisions. Using advanced SQL skills and working with different teams has given me a clearer picture of what's going on and what needs to be done.

In conclusion this project was not just about analysing the data but also understanding company’s operations and market dynamics. This experience will further help in boosting my career as data analyst in shaping successful products and services in various organizations.