OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

SUBRAT SINDHU

Project Description

- Operational Analytics, a vital process in business, entails end-to-end operations analysis, enabling Data Analysts to collaborate with diverse teams and extract valuable insights for continuous improvement.
- Operational analytics depends on the skillful investigation of metric spikes, which reveals sudden changes in critical indicators such as a rise in daily user engagement or a fall in sales. It is crucial for a Data Analyst to develop the ability to investigate these variations on a daily basis, which calls for a thorough understanding of the causes of these metric spikes.
- Working as a Lead Data Analyst for a corporation such as Microsoft , I will use SQL knowledge to draw conclusions from a variety of datasets and respond to requests from different departments. The primary goal is to improve operational effectiveness and shed light on the reasons for changes in important indicators so that decisions can be made with knowledge.

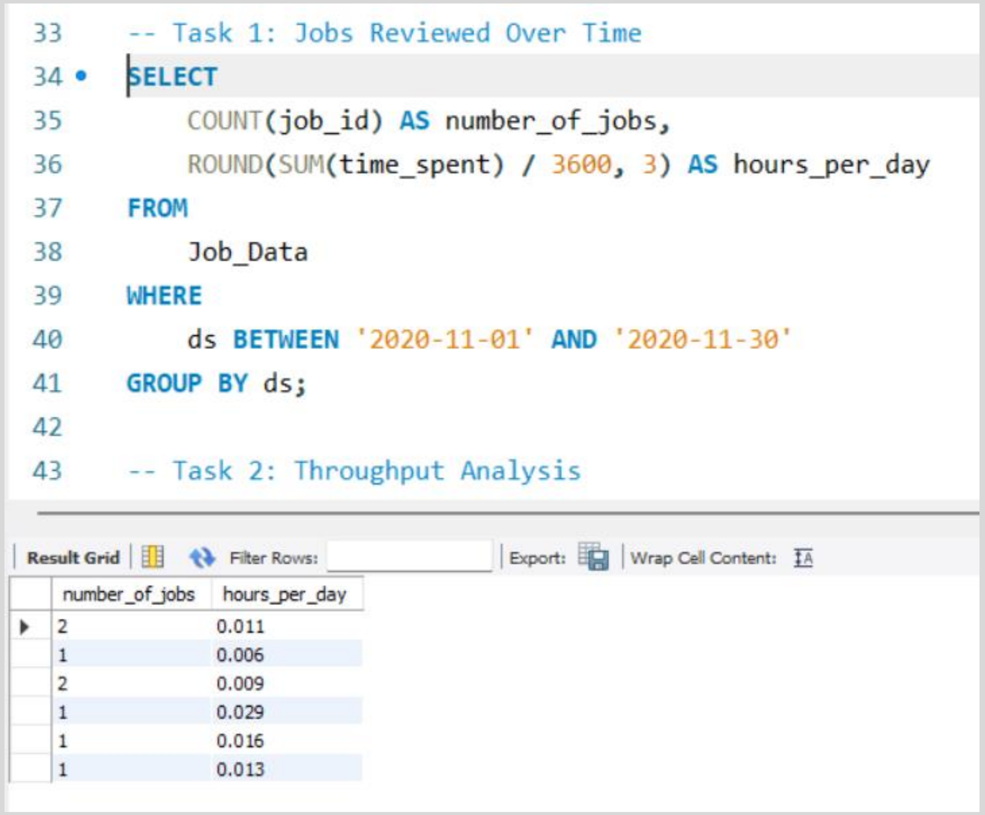
Approach

- Importing the Dataset: The first step is maintaining a file path to export data (.csv) into the SQL workbench.
- Understanding the Schema: The next step is to examine the structure of the table holding the data (job_data, events, etc).
- Identifying the Key tables: Identification of the primary key from each of the tables of job_data, email_events, events, users etc.
- Checking for null values: Before the analysis, it is necessary to check for null values in the given tables
- Visually Appealing: The SQL Queries need to be properly formatted so that they can be understood by any user.

CASE STUDY 1: Job Data Analysis

A. Jobs Reviewed Over Time:

Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.

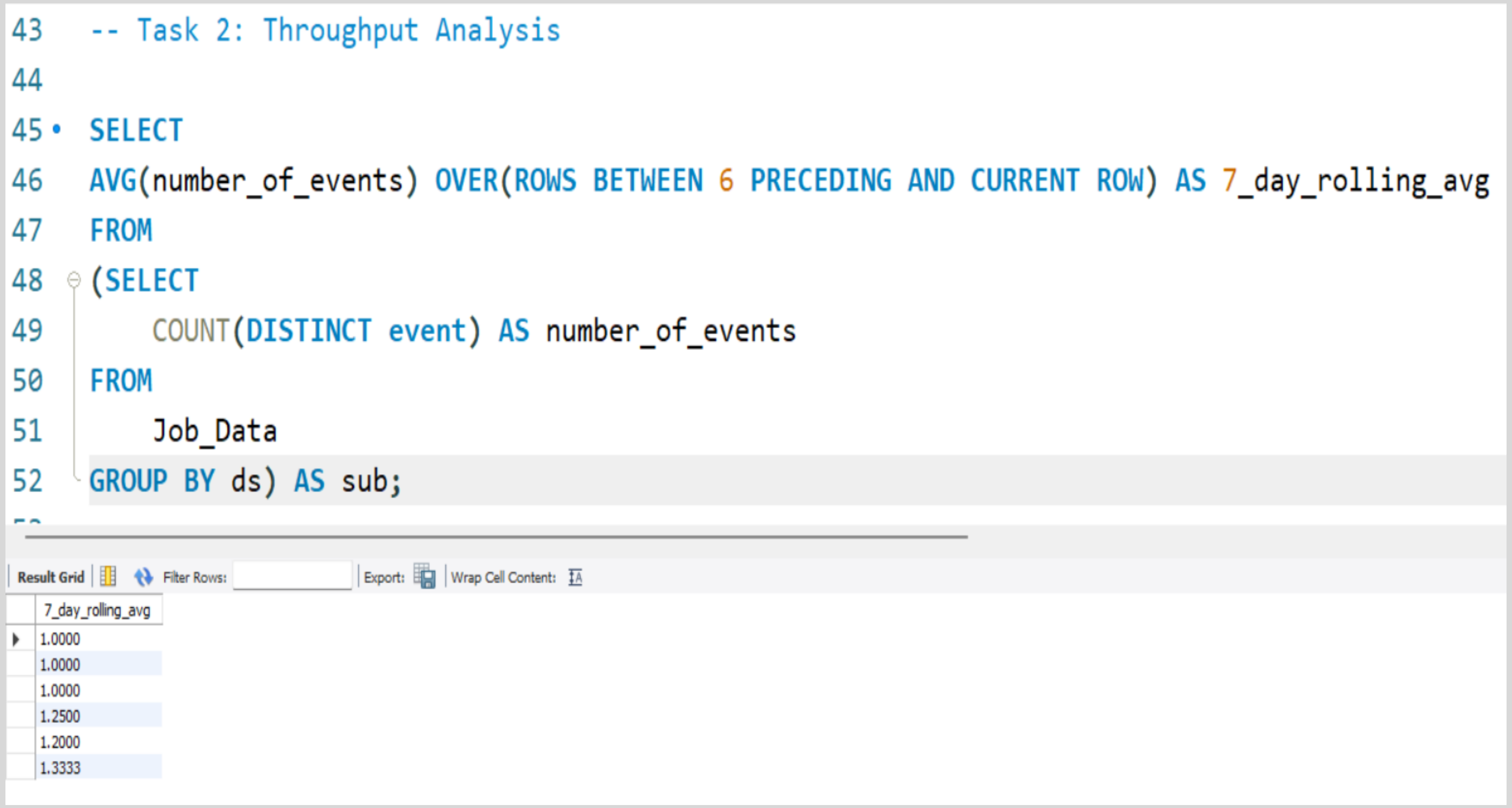


```sql
-- Task 1: Jobs Reviewed Over Time
SELECT
    COUNT(job_id) AS number_of_jobs,
    ROUND(SUM(time_spent) / 3600, 3) AS hours_per_day
FROM
    Job_Data
WHERE
    ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY ds;

-- Task 2: Throughput Analysis
```

number_of_jobs	hours_per_day
2	0.011
1	0.006
2	0.009
1	0.029
1	0.016
1	0.013

B. Throughput Analysis:

Calculate the 7-day rolling average of throughput (number of events per second).



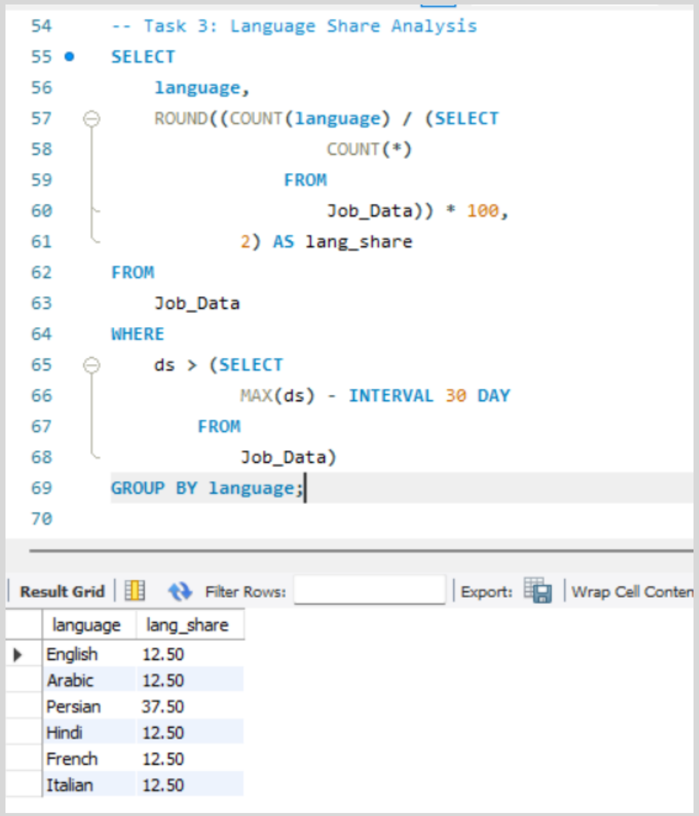
```sql
-- Task 2: Throughput Analysis

SELECT
AVG(number_of_events) OVER(ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS 7_day_rolling_avg
FROM
(SELECT
    COUNT(DISTINCT event) AS number_of_events
FROM
    Job_Data
GROUP BY ds) AS sub;
```

7_day_rolling_avg
1.0000
1.0000
1.0000
1.2500
1.2000
1.3333

C. Language Share Analysis:

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

Persian language has most of the share among other languages in the past 30 days



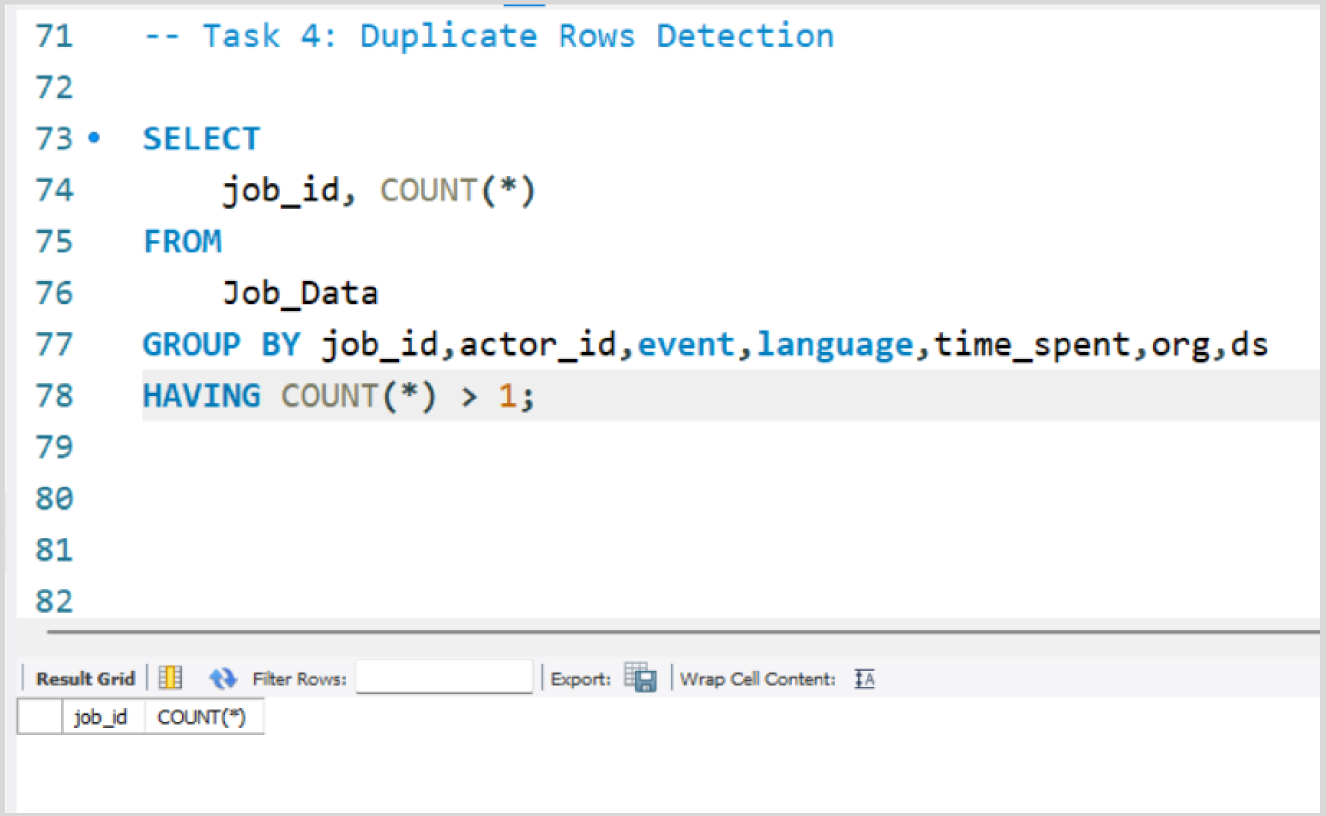
```
-- Task 3: Language Share Analysis
SELECT
    language,
    ROUND((COUNT(language) / (SELECT
                COUNT(*)
            FROM
                Job_Data)) * 100,
            2) AS lang_share
FROM
    Job_Data
WHERE
    ds > (SELECT
            MAX(ds) - INTERVAL 30 DAY
        FROM
            Job_Data)
GROUP BY language;
```

language	lang_share
English	12.50
Arabic	12.50
Persian	37.50
Hindi	12.50
French	12.50
Italian	12.50

D. Duplicate Rows Detection:

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

There are no duplicate rows in the given Job_Data table. Hence there is no output.



```
-- Task 4: Duplicate Rows Detection

SELECT
    job_id, COUNT(*)
FROM
    Job_Data
GROUP BY job_id,actor_id,event,language,time_spent,org,ds
HAVING COUNT(*) > 1;
```

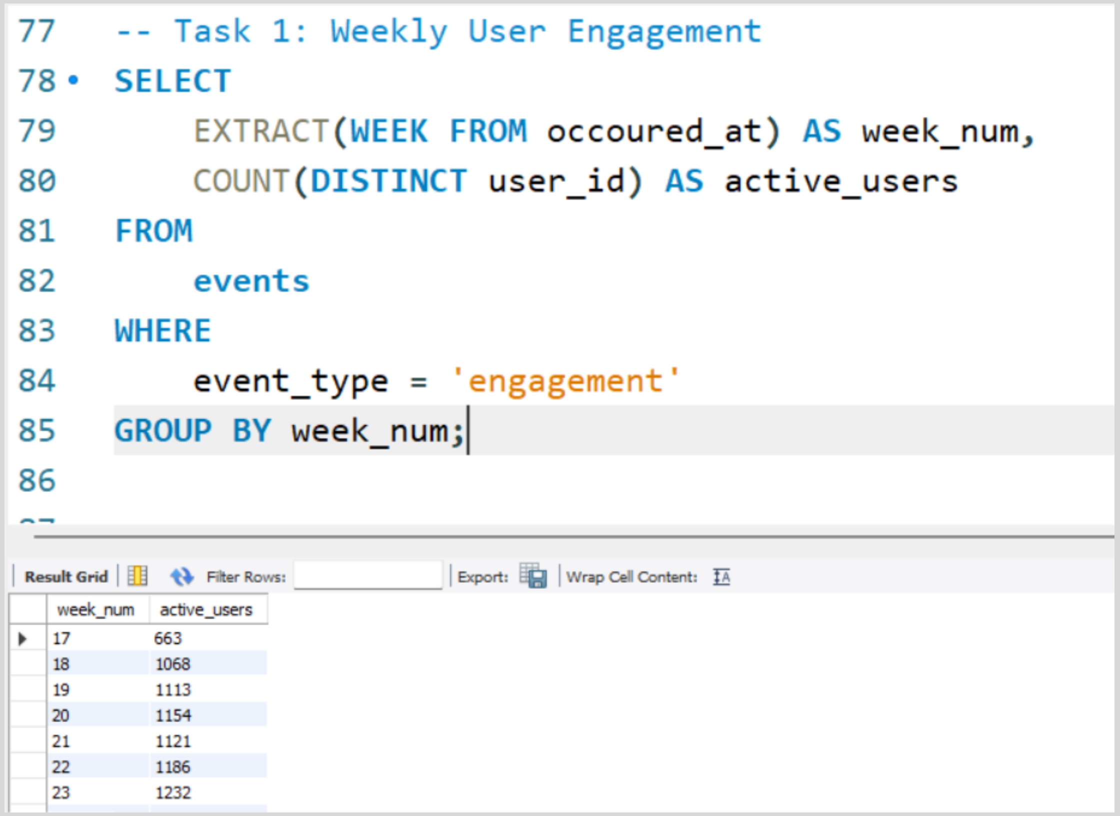
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

job_id	COUNT(*)

CASE STUDY 2: Investigating Metric Spike

A. Weekly User Engagement:

Write an SQL query to calculate the weekly user engagement.



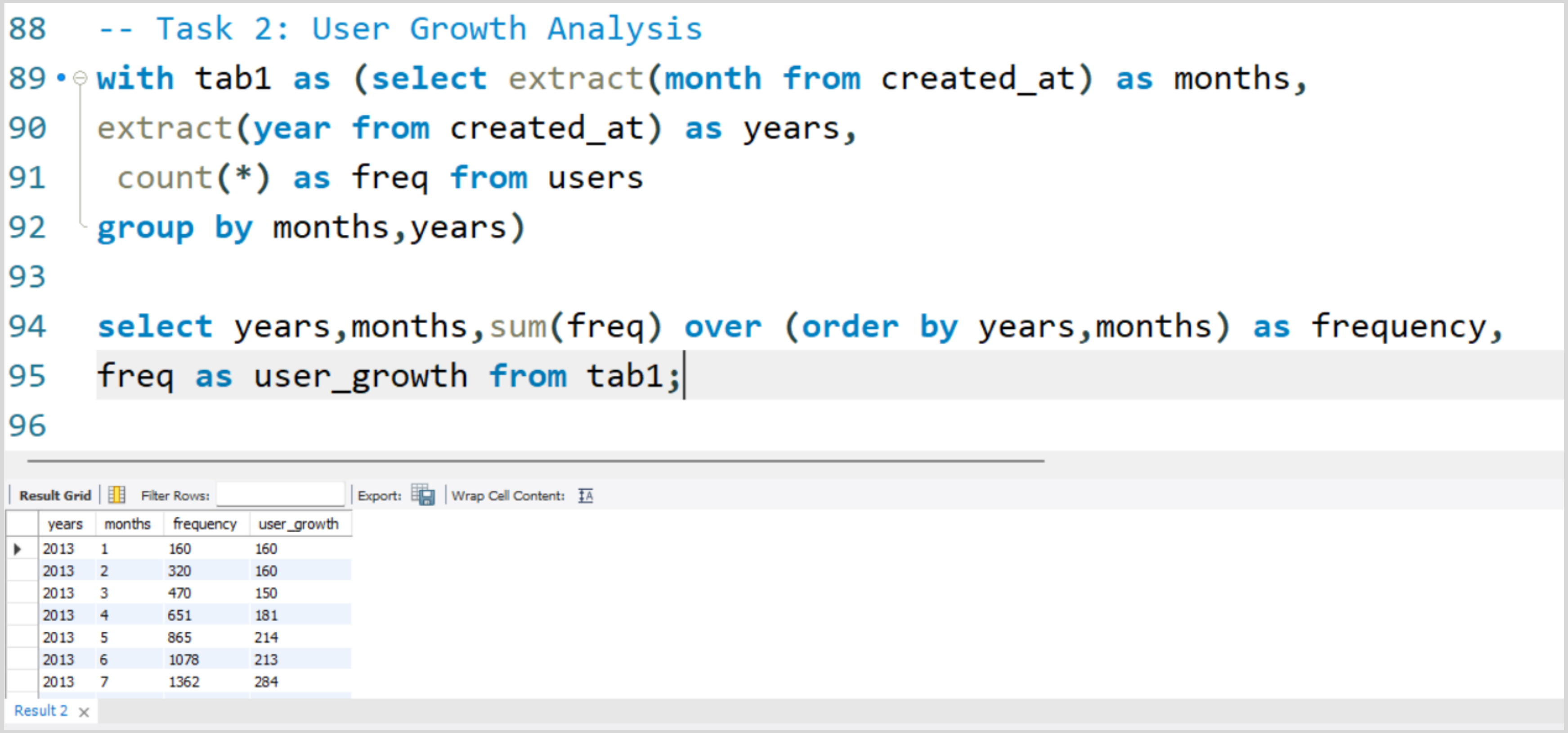
```
-- Task 1: Weekly User Engagement
SELECT
    EXTRACT(WEEK FROM occoured_at) AS week_num,
    COUNT(DISTINCT user_id) AS active_users
FROM
    events
WHERE
    event_type = 'engagement'
GROUP BY week_num;
```

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

week_num	active_users
17	663
18	1068
19	1113
20	1154
21	1121
22	1186
23	1232

B. User Growth Analysis:

Write an SQL query to calculate the user growth for the product.

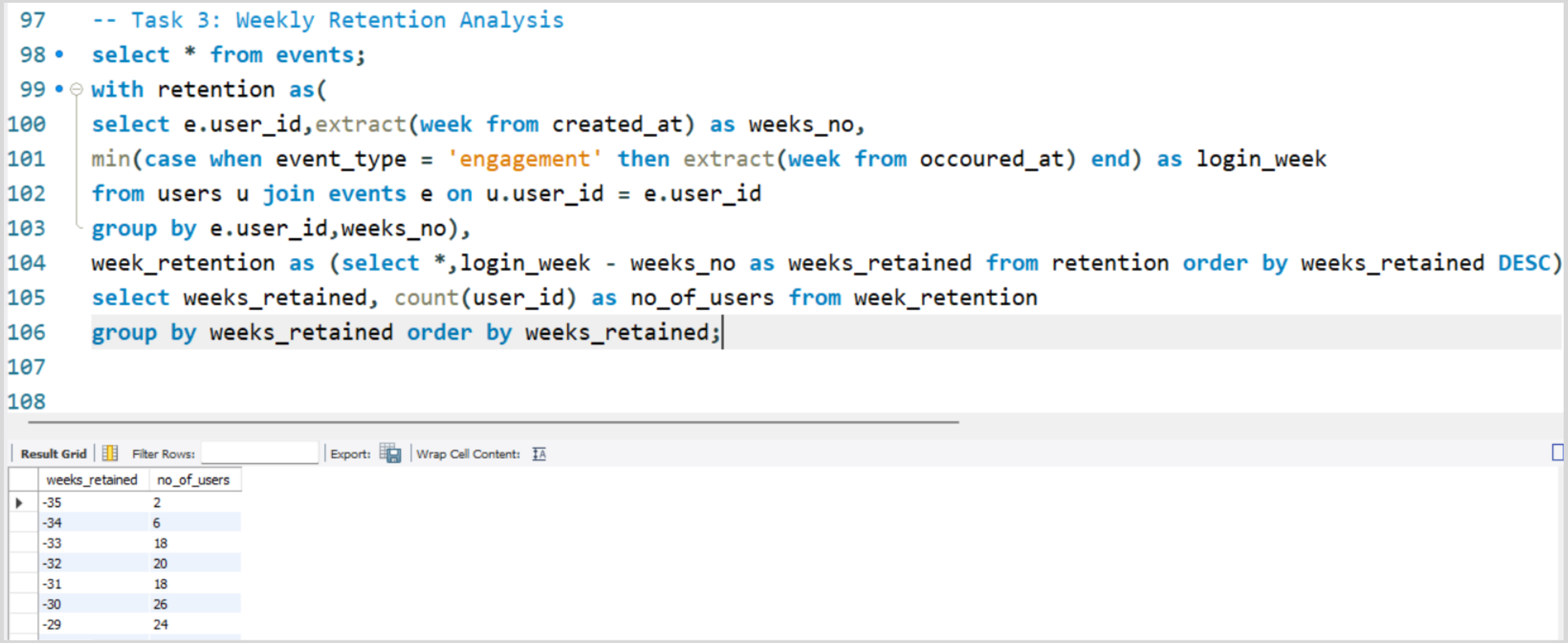


```sql
-- Task 2: User Growth Analysis
with tab1 as (select extract(month from created_at) as months,
extract(year from created_at) as years,
 count(*) as freq from users
group by months,years)

select years,months,sum(freq) over (order by years,months) as frequency,
freq as user_growth from tab1;
```

years	months	frequency	user_growth
2013	1	160	160
2013	2	320	160
2013	3	470	150
2013	4	651	181
2013	5	865	214
2013	6	1078	213
2013	7	1362	284

C. Weekly Retention Analysis:

Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

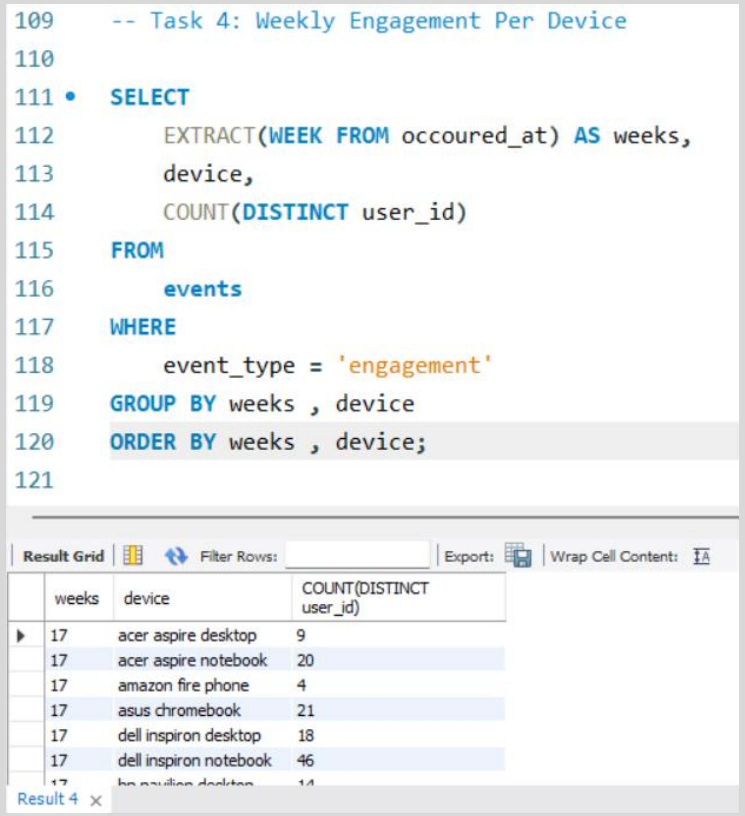


```sql
-- Task 3: Weekly Retention Analysis
select * from events;
with retention as(
select e.user_id,extract(week from created_at) as weeks_no,
min(case when event_type = 'engagement' then extract(week from occoured_at) end) as login_week
from users u join events e on u.user_id = e.user_id
group by e.user_id,weeks_no),
week_retention as (select *,login_week - weeks_no as weeks_retained from retention order by weeks_retained DESC)
select weeks_retained, count(user_id) as no_of_users from week_retention
group by weeks_retained order by weeks_retained;
```

weeks_retained	no_of_users
-35	2
-34	6
-33	18
-32	20
-31	18
-30	26
-29	24

D. Weekly Engagement Per Device:

Write an SQL query to calculate the weekly engagement per device.



```sql
-- Task 4: Weekly Engagement Per Device

SELECT
    EXTRACT(WEEK FROM occoured_at) AS weeks,
    device,
    COUNT(DISTINCT user_id)
FROM
    events
WHERE
    event_type = 'engagement'
GROUP BY weeks , device
ORDER BY weeks , device;
```

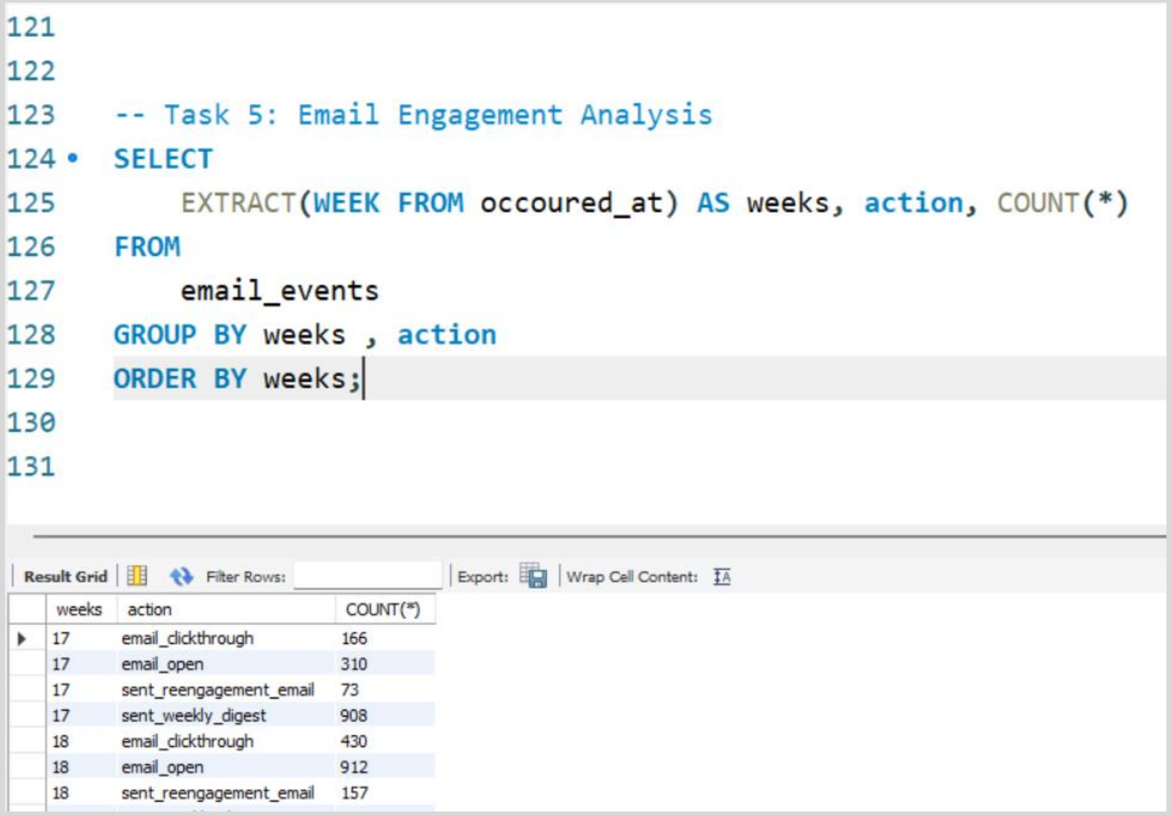
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

```
weeks  device                 COUNT(DISTINCT
                              user_id)
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
```

Result 4 ×

E. Email Engagement Analysis:

Write an SQL query to calculate the email engagement metrics.



```
-- Task 5: Email Engagement Analysis
SELECT
    EXTRACT(WEEK FROM occoured_at) AS weeks, action, COUNT(*)
FROM
    email_events
GROUP BY weeks , action
ORDER BY weeks;
```

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

weeks	action	COUNT(*)
17	email_clickthrough	166
17	email_open	310
17	sent_reengagement_email	73
17	sent_weekly_digest	908
18	email_clickthrough	430
18	email_open	912
18	sent_reengagement_email	157

Tech-Stack

- **MySQL Workbench:** The main interactive development environment for SQL queries is MySQL Workbench (8.0.34). For data analysis, it makes query creation, execution, and debugging more efficient.
- **Windows Function:** Does calculation across a set of rows that are related to the current row. These functions are used when we want to calculate Average Running Price, Running Total Orders, Running Sum Sales, Rank and Percentile.

Insights

1. Job Data Analysis

In case study 1, we were able to analyze job reviewed per hour for each day in the month of November, 2020. Also some other insights like removal of duplicate rows, percentage share of each language in the past 30 days, and 7-day rolling average throughput.

2. Investigating Metric Spike

In Case Study 2, we were able to understand how different users engage with email events in the application. Were able to analyse growth of users over time for a product, were able to analyse and measure activeness of a user on a weekly basis per device.

Result

- Remembering to adapt these queries on specific database schemas.
- These learned insights helped me understand specific business questions which were addressed by SQL queries.
- Learning about the SQL clauses such as the join clauses and sub-queries. The importance of order by and group by and many more.
- We were able to import the dataset from a .CSV file into the SQL workbench for performing analysis.
- Achieving the ability to learn and write SQL queries to execute different business questions.
- Solving business related problems using Windows functions of SQL.

Thankyou