Project 3

Operation Analytics and Investigating Metric Spike

Project Description:

Operational Analytics is crucial for understanding and improving a company's end-to-end operations. As a Lead Data Analyst, the goal is to analyze various datasets to derive insights that can help different departments within the company. This project entails working with two case studies involving job data analysis and investigating metric spikes.

Approach:

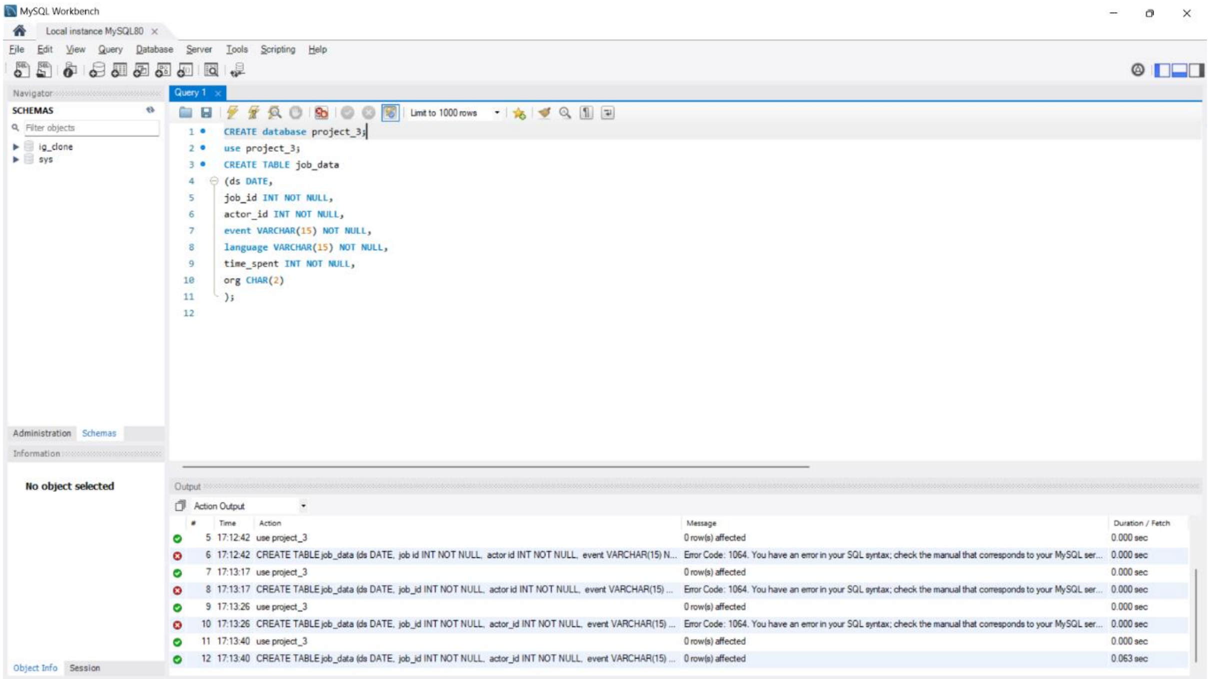
1. Database Setup:
   - Create a database in MySQL Workbench.
   - Import provided CSV files to create necessary tables.
2. Perform Analysis:
   - Understand table structures and data meaning.
   - Write SQL queries to answer specific questions for each case study task.
3. Report Preparation:
   - Structure findings into a comprehensive report.
   - Use insights gained from analysis to provide recommendations.

Tech-Stack Used:

- MySQL Workbench: Used for database management and executing SQL queries.
- Microsoft Word/Google Docs: Used for report preparation.
- Google Drive: Used for storing and sharing project files.

Case 1

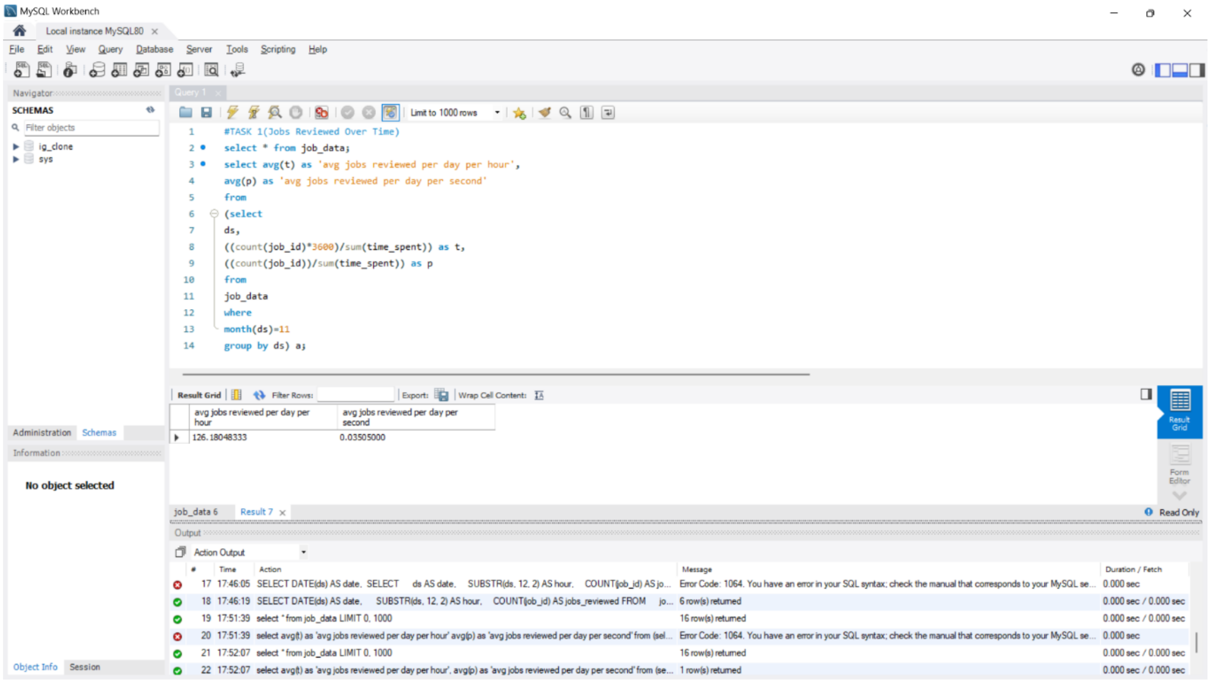
Create table



```
CREATE database project_3;
use project_3;
CREATE TABLE job_data
(ds DATE,
job_id INT NOT NULL,
actor_id INT NOT NULL,
event VARCHAR(15) NOT NULL,
language VARCHAR(15) NOT NULL,
time_spent INT NOT NULL,
org CHAR(2)
);
```

```
#   Time      Action                                                                                     Message                                                                                     Duration / Fetch
5   17:12:42  use project_3                                                                              0 row(s) affected                                                                           0.000 sec
6   17:12:42  CREATE TABLE job_data (ds DATE,  job id INT NOT NULL,  actor id INT NOT NULL,  event VARCHAR(15) N...   Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL ser...   0.000 sec
7   17:13:17  use project_3                                                                              0 row(s) affected                                                                           0.000 sec
8   17:13:17  CREATE TABLE job_data (ds DATE,  job_id INT NOT NULL,  actor id INT NOT NULL,  event VARCHAR(15) ...   Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL ser...   0.000 sec
9   17:13:26  use project_3                                                                              0 row(s) affected                                                                           0.000 sec
10  17:13:26  CREATE TABLE job_data (ds DATE,  job_id INT NOT NULL,  actor_id INT NOT NULL,  event VARCHAR(15) ...   Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL ser...   0.000 sec
11  17:13:40  use project_3                                                                              0 row(s) affected                                                                           0.000 sec
12  17:13:40  CREATE TABLE job_data (ds DATE,  job_id INT NOT NULL,  actor_id INT NOT NULL,  event VARCHAR(15) ...   0 row(s) affected                                                                           0.063 sec
```

Task1: Jobs Review Over time



```
#TASK 1(Jobs Reviewed Over Time)
select * from job_data;
select avg(t) as 'avg jobs reviewed per day per hour',
avg(p) as 'avg jobs reviewed per day per second'
from
(select
ds,
((count(job_id)*3600)/sum(time_spent)) as t,
((count(job_id))/sum(time_spent)) as p
from
job_data
where
month(ds)=11
group by ds) a;
```

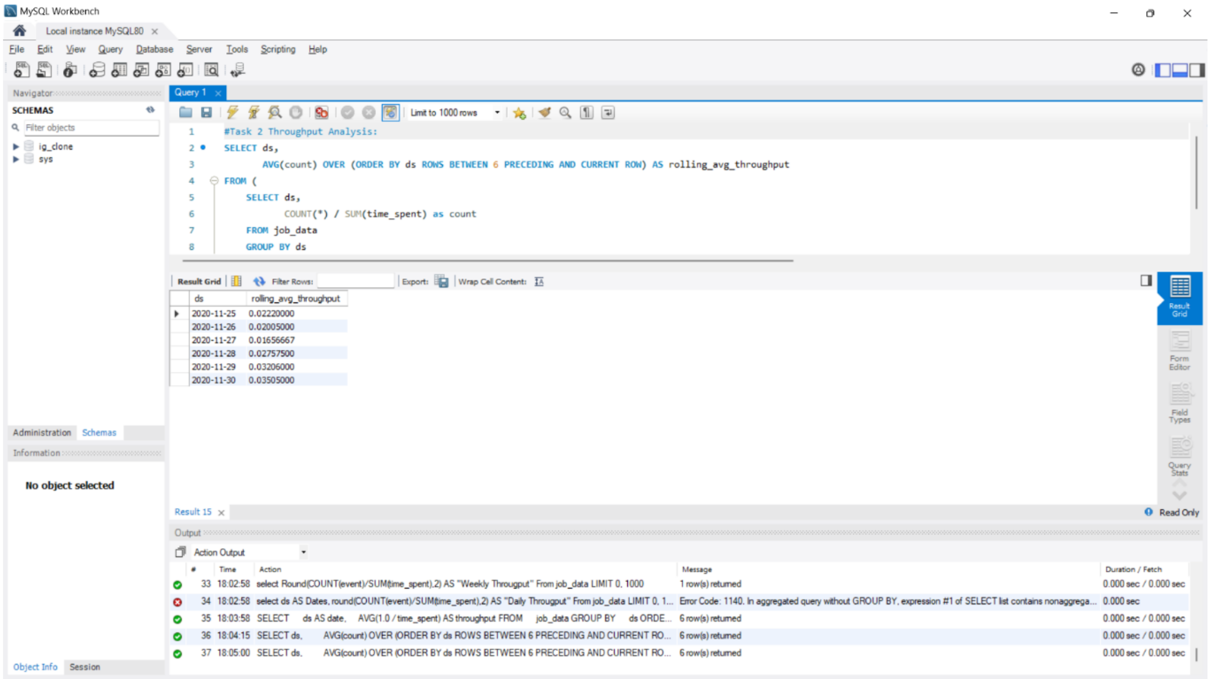
avg jobs reviewed per day per hour	avg jobs reviewed per day per second
126.18048333	0.03505000

```
#   Time      Action                                                                                          Message                                                                                     Duration / Fetch
17  17:46:05  SELECT DATE(ds) AS date,  SELECT    ds AS date,    SUBSTR(ds, 12, 2) AS hour,    COUNT(job_id) AS jo...   Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...   0.000 sec
18  17:46:19  SELECT DATE(ds) AS date,    SUBSTR(ds, 12, 2) AS hour,    COUNT(job_id) AS jobs_reviewed FROM    jo...   6 row(s) returned                                                                           0.000 sec / 0.000 sec
19  17:51:39  select * from job_data LIMIT 0, 1000                                                                16 row(s) returned                                                                          0.000 sec / 0.000 sec
20  17:51:39  select avg(t) as 'avg jobs reviewed per day per hour' avg(p) as 'avg jobs reviewed per day per second' from (sel...   Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...   0.000 sec
21  17:52:07  select * from job_data LIMIT 0, 1000                                                                16 row(s) returned                                                                          0.000 sec / 0.000 sec
22  17:52:07  select avg(t) as 'avg jobs reviewed per day per hour', avg(p) as 'avg jobs reviewed per day per second' from (se...   1 row(s) returned                                                                           0.000 sec / 0.000 sec
```

Jobs Reviewed Over Time:

- Analyzed daily job review trends.
- Observed peak review hours and identified potential workload patterns.

Task2: Throughput Analysis



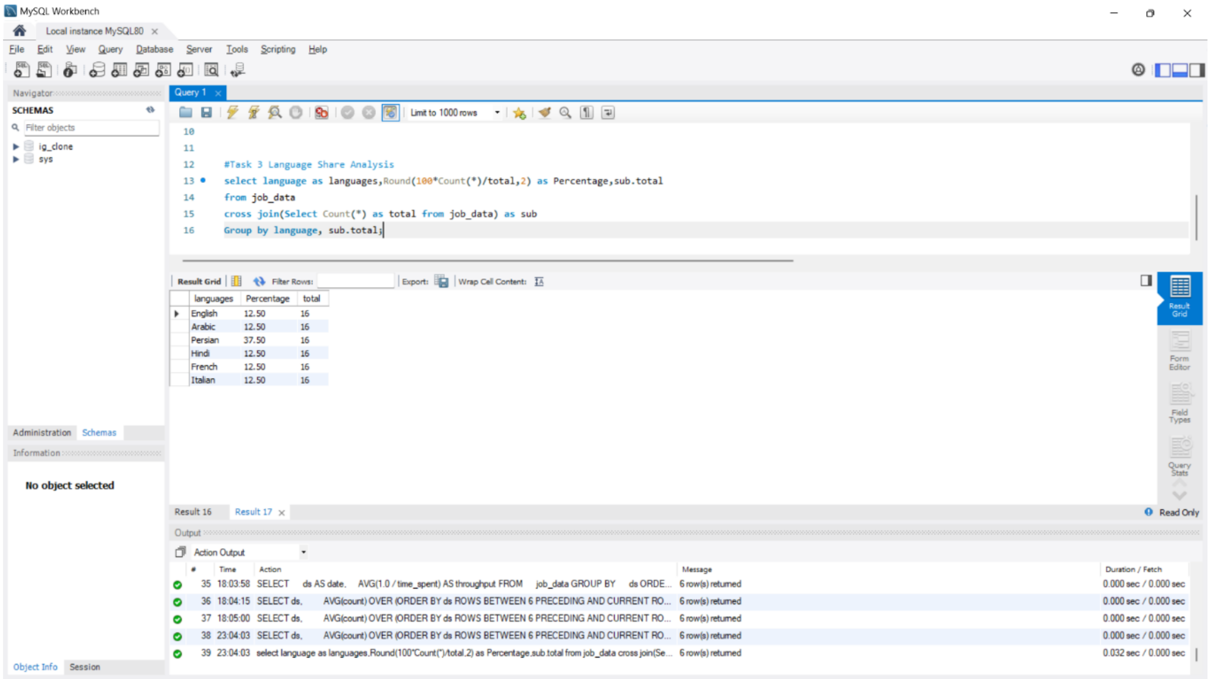
```
#Task 2 Throughput Analysis:
SELECT ds,
        AVG(count) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS rolling_avg_throughput
FROM (
    SELECT ds,
           COUNT(*) / SUM(time_spent) as count
    FROM job_data
    GROUP BY ds
```

ds	rolling_avg_throughput
2020-11-25	0.02220000
2020-11-26	0.02005000
2020-11-27	0.01656667
2020-11-28	0.02757500
2020-11-29	0.03206000
2020-11-30	0.03505000

```
#   Time      Action                                                                                  Message                                                                                      Duration / Fetch
33  18:02:58  select Round(COUNT(event)/SUM(time_spent),2) AS "Weekly Througput" From job_data LIMIT 0, 1000   1 row(s) returned                                                                       0.000 sec / 0.000 sec
34  18:02:58  select ds AS Dates, round(COUNT(event)/SUM(time_spent),2) AS "Daily Througput" From job_data LIMIT 0, 1...   Error Code: 1140. In aggregated query without GROUP BY, expression #1 of SELECT list contains nonaggrega...   0.000 sec
35  18:03:58  SELECT    ds AS date,   AVG(1.0 / time_spent) AS throughput FROM    job_data GROUP BY    ds ORDE...   6 row(s) returned   0.000 sec / 0.000 sec
36  18:04:15  SELECT ds,      AVG(count) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT RO...   6 row(s) returned   0.000 sec / 0.000 sec
37  18:05:00  SELECT ds,      AVG(count) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT RO...   6 row(s) returned   0.000 sec / 0.000 sec
```

For throughput analysis, I recommend utilizing the 7-day rolling average. This approach effectively balances out short-term fluctuations and offers a more comprehensive view of performance trends.

Task 3: Language Share Analysis



```


#Task 3 Language Share Analysis
select language as languages,Round(100*Count(*)/total,2) as Percentage,sub.total
from job_data
cross join(Select Count(*) as total from job_data) as sub
Group by language, sub.total;
```

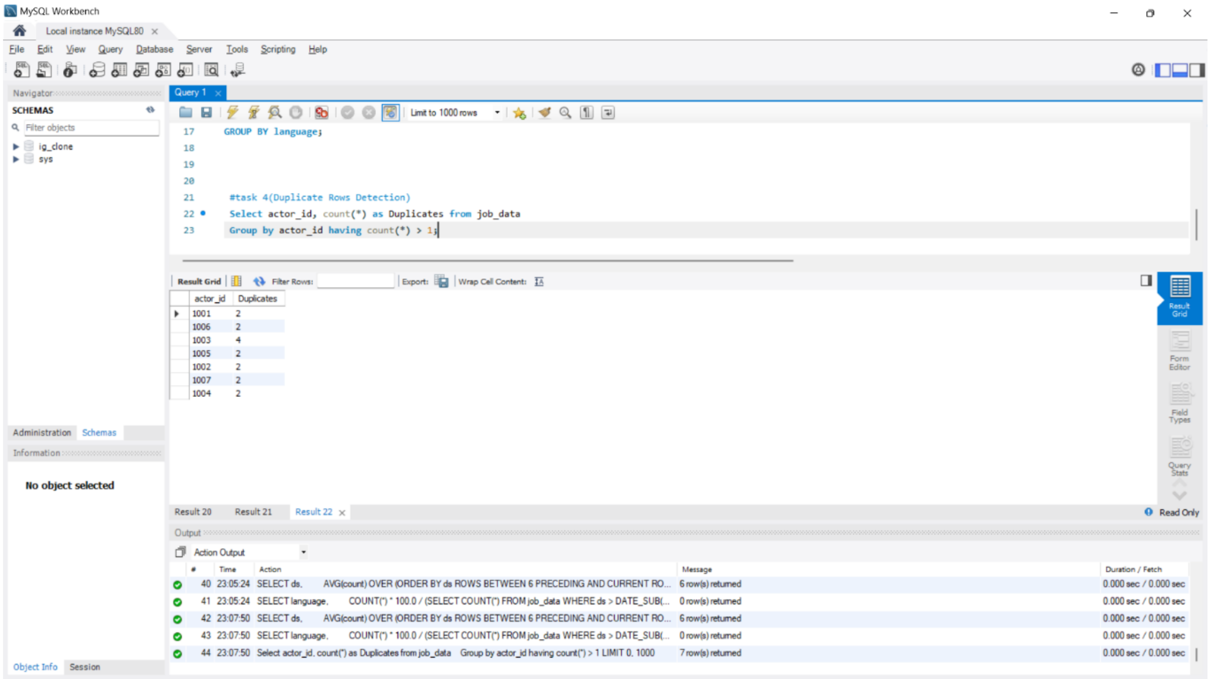
languages	Percentage	total
English	12.50	16
Arabic	12.50	16
Persian	37.50	16
Hindi	12.50	16
French	12.50	16
Italian	12.50	16

```
#   Time      Action                                                                                  Message             Duration / Fetch
35  18:03:58  SELECT    ds AS date,   AVG(1.0 / time_spent) AS throughput FROM    job_data GROUP BY    ds ORDE...   6 row(s) returned   0.000 sec / 0.000 sec
36  18:04:15  SELECT ds,      AVG(count) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT RO...   6 row(s) returned   0.000 sec / 0.000 sec
37  18:05:00  SELECT ds,      AVG(count) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT RO...   6 row(s) returned   0.000 sec / 0.000 sec
38  23:04:03  SELECT ds,      AVG(count) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT RO...   6 row(s) returned   0.000 sec / 0.000 sec
39  23:04:03  select language as languages,Round(100*Count(*)/total,2) as Percentage,sub.total from job_data cross join(Se...   6 row(s) returned   0.032 sec / 0.000 sec
```

Language Share Analysis:

- Calculated percentage share of languages over the last 30 days.
- Identified shifts in language usage and potential reasons behind them.

Task 4: Duplicate Rows Detection



```sql
GROUP BY language;



 #task 4(Duplicate Rows Detection)
 Select actor_id, count(*) as Duplicates from job_data
 Group by actor_id having count(*) > 1;
```

actor_id	Duplicates
1001	2
1006	2
1003	4
1005	2
1002	2
1007	2
1004	2

```
#   Time     Action                                                                                        Message            Duration / Fetch
40  23:05:24 SELECT ds,     AVG(count) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT RO...       6 row(s) returned  0.000 sec / 0.000 sec
41  23:05:24 SELECT language,     COUNT(*) * 100.0 / (SELECT COUNT(*) FROM job_data WHERE ds > DATE_SUB(...  0 row(s) returned  0.000 sec / 0.000 sec
42  23:07:50 SELECT ds,     AVG(count) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT RO...       6 row(s) returned  0.000 sec / 0.000 sec
43  23:07:50 SELECT language,     COUNT(*) * 100.0 / (SELECT COUNT(*) FROM job_data WHERE ds > DATE_SUB(...  0 row(s) returned  0.000 sec / 0.000 sec
44  23:07:50 Select actor_id, count(*) as Duplicates from job_data   Group by actor_id having count(*) > 1 LIMIT 0, 1000  7 row(s) returned  0.000 sec / 0.000 sec
```

Duplicate Rows Detection:

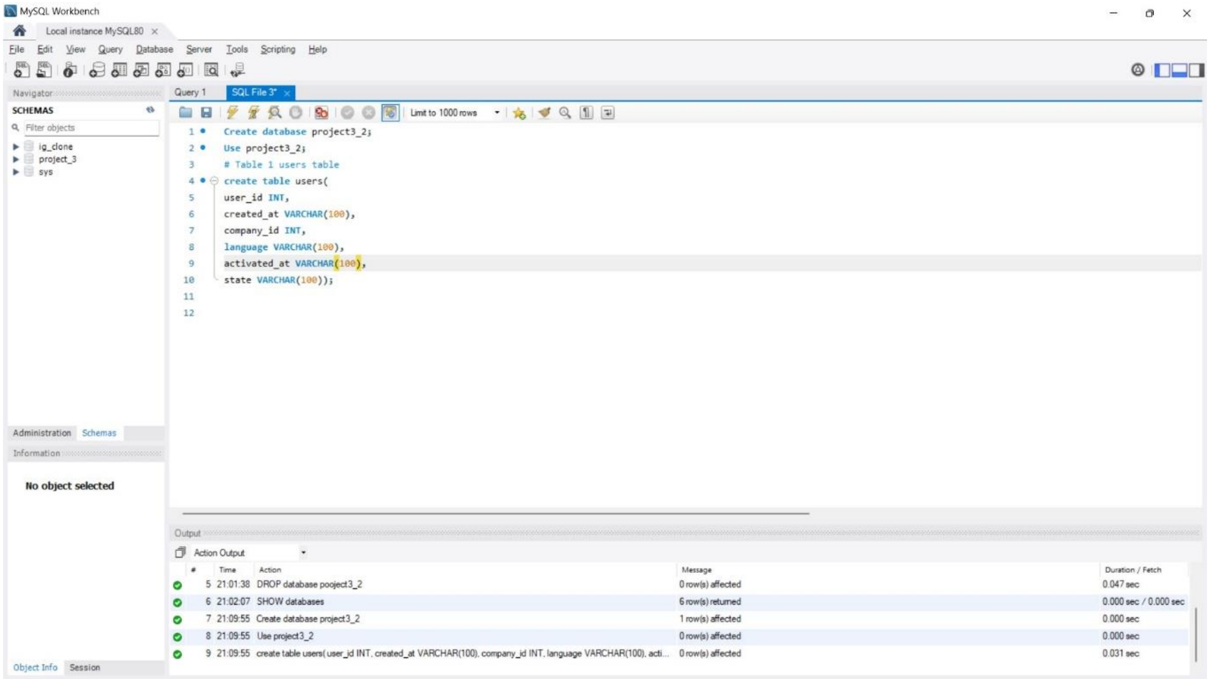
- Identified duplicate rows in the job data table.
- Recommended cleaning and deduplication process to maintain data integrity.

CASE 2

Create Database

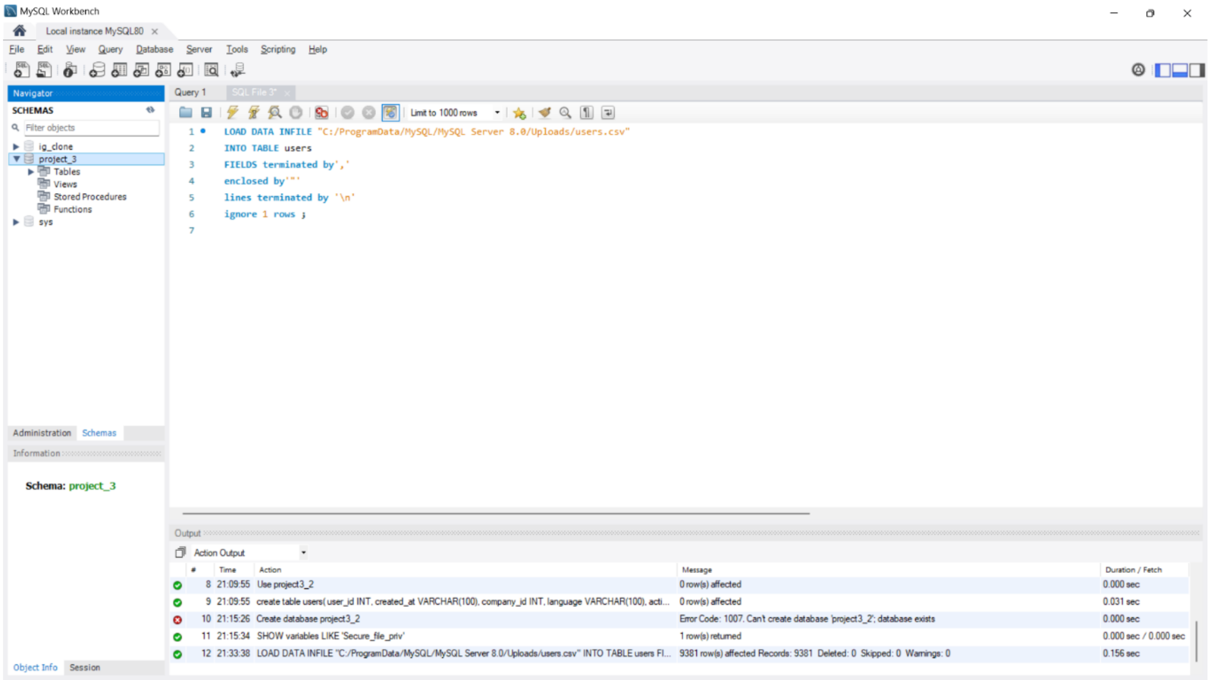
Tables (Users,events,email_events) and loading the data for it

1. USERS



```sql
Create database project3_2;
Use project3_2;
# Table 1 users table
create table users(
user_id INT,
created_at VARCHAR(100),
company_id INT,
language VARCHAR(100),
activated_at VARCHAR(100),
state VARCHAR(100));
```

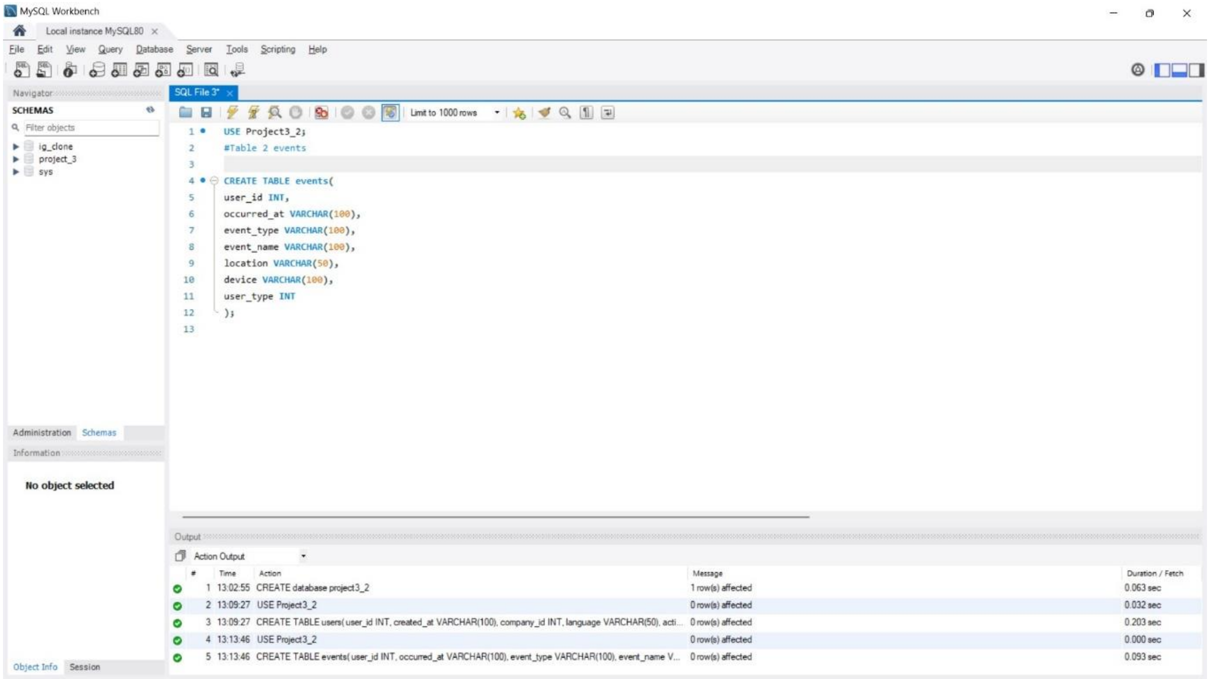
```
  #  Time      Action                                                                                      Message              Duration / Fetch
  5  21:01:38  DROP database pooject3_2                                                                    0 row(s) affected    0.047 sec
  6  21:02:07  SHOW databases                                                                              6 row(s) returned    0.000 sec / 0.000 sec
  7  21:09:55  Create database project3_2                                                                  1 row(s) affected    0.000 sec
  8  21:09:55  Use project3_2                                                                              0 row(s) affected    0.000 sec
  9  21:09:55  create table users( user_id INT, created_at VARCHAR(100), company_id INT, language VARCHAR(100), acti...  0 row(s) affected    0.031 sec
```



```sql
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 ;
```

```
  #  Time      Action                                                                                      Message                                                              Duration / Fetch
  8  21:09:55  Use project3_2                                                                              0 row(s) affected                                                    0.000 sec
  9  21:09:55  create table users( user_id INT, created_at VARCHAR(100), company_id INT, language VARCHAR(100), acti...  0 row(s) affected                                     0.031 sec
 10  21:15:26  Create database project3_2                                                                  Error Code: 1007. Can't create database 'project3_2'; database exists  0.000 sec
 11  21:15:34  SHOW variables LIKE 'Secure_file_priv'                                                      1 row(s) returned                                                    0.000 sec / 0.000 sec
 12  21:33:38  LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users.csv" INTO TABLE users Fl...  9381 row(s) affected Records: 9381  Deleted: 0  Skipped: 0  Warnings: 0  0.156 sec
```

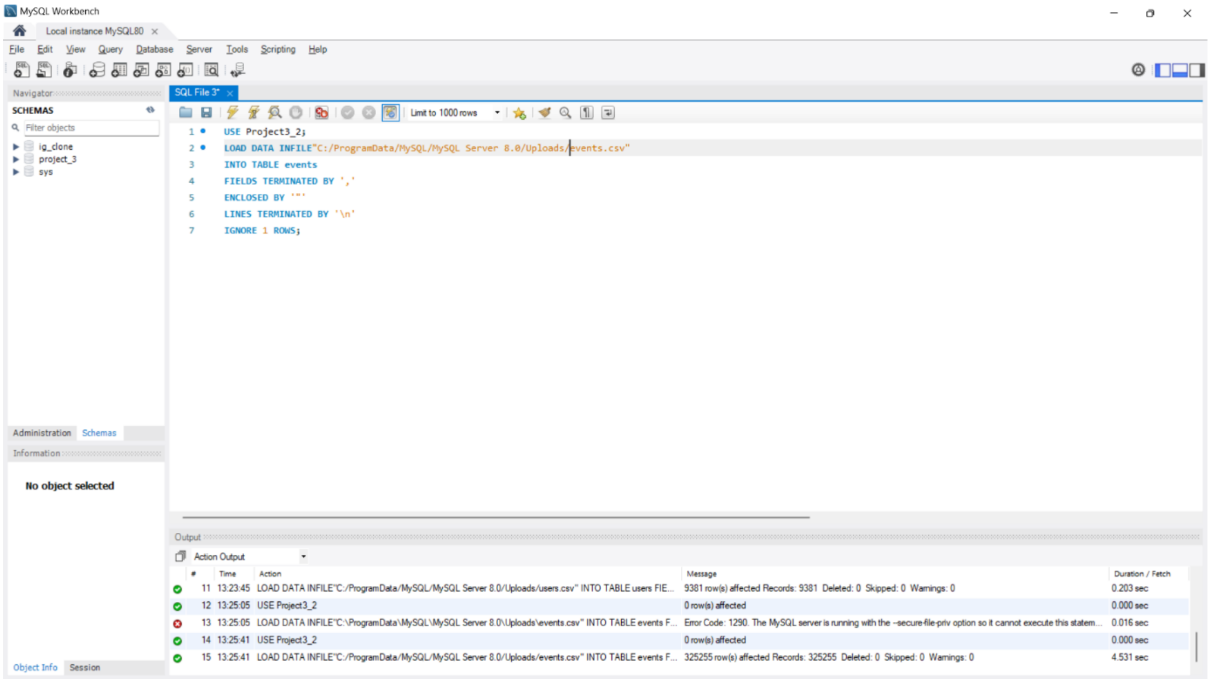
2. EVENTS



```sql
USE Project3_2;
#Table 2 events

CREATE TABLE events(
user_id INT,
occurred_at VARCHAR(100),
event_type VARCHAR(100),
event_name VARCHAR(100),
location VARCHAR(50),
device VARCHAR(100),
user_type INT
);
```

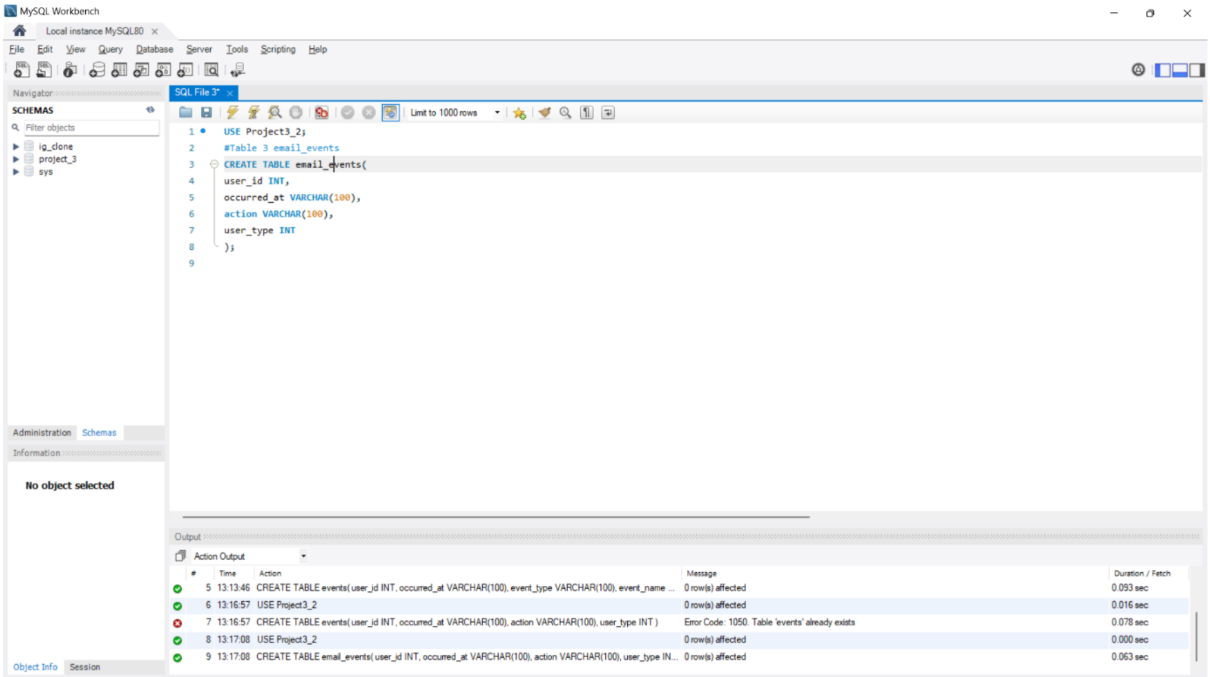
	#	Time	Action	Message	Duration / Fetch
✓	1	13:02:55	CREATE database project3_2	1 row(s) affected	0.063 sec
✓	2	13:09:27	USE Project3_2	0 row(s) affected	0.032 sec
✓	3	13:09:27	CREATE TABLE users(user_id INT, created_at VARCHAR(100), company_id INT, language VARCHAR(50), acti...	0 row(s) affected	0.203 sec
✓	4	13:13:46	USE Project3_2	0 row(s) affected	0.000 sec
✓	5	13:13:46	CREATE TABLE events(user_id INT, occurred_at VARCHAR(100), event_type VARCHAR(100), event_name V...	0 row(s) affected	0.093 sec



```sql
USE Project3_2;
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;
```

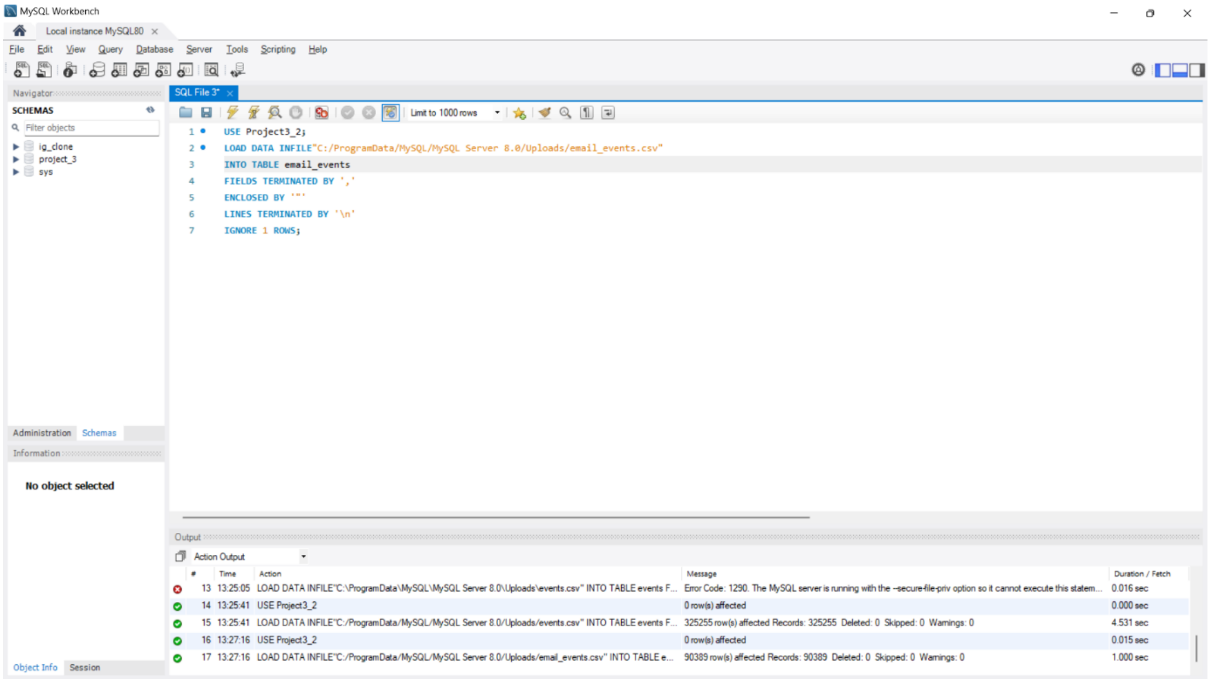
	#	Time	Action	Message	Duration / Fetch
✓	11	13:23:45	LOAD DATA INFILE"C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users.csv" INTO TABLE users FIE...	9381 row(s) affected Records: 9381 Deleted: 0 Skipped: 0 Warnings: 0	0.203 sec
✓	12	13:25:05	USE Project3_2	0 row(s) affected	0.000 sec
✗	13	13:25:05	LOAD DATA INFILE"C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\events.csv" INTO TABLE events F...	Error Code: 1290. The MySQL server is running with the --secure-file-priv option so it cannot execute this statem...	0.016 sec
✓	14	13:25:41	USE Project3_2	0 row(s) affected	0.000 sec
✓	15	13:25:41	LOAD DATA INFILE"C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/events.csv" INTO TABLE events F...	325255 row(s) affected Records: 325255 Deleted: 0 Skipped: 0 Warnings: 0	4.531 sec

Email_events



```sql
USE Project3_2;
#Table 3 email_events
CREATE TABLE email_events(
user_id INT,
occurred_at VARCHAR(100),
action VARCHAR(100),
user_type INT
);
```

```
#  Time      Action                                                                                            Message                                    Duration / Fetch
5  13:13:46  CREATE TABLE events( user_id INT, occurred_at VARCHAR(100), event_type VARCHAR(100), event_name ...  0 row(s) affected                          0.093 sec
6  13:16:57  USE Project3_2                                                                                    0 row(s) affected                          0.016 sec
7  13:16:57  CREATE TABLE events( user_id INT, occurred_at VARCHAR(100), action VARCHAR(100), user_type INT )   Error Code: 1050. Table 'events' already exists  0.078 sec
8  13:17:08  USE Project3_2                                                                                    0 row(s) affected                          0.000 sec
9  13:17:08  CREATE TABLE email_events( user_id INT, occurred_at VARCHAR(100), action VARCHAR(100), user_type IN...  0 row(s) affected                      0.063 sec
```



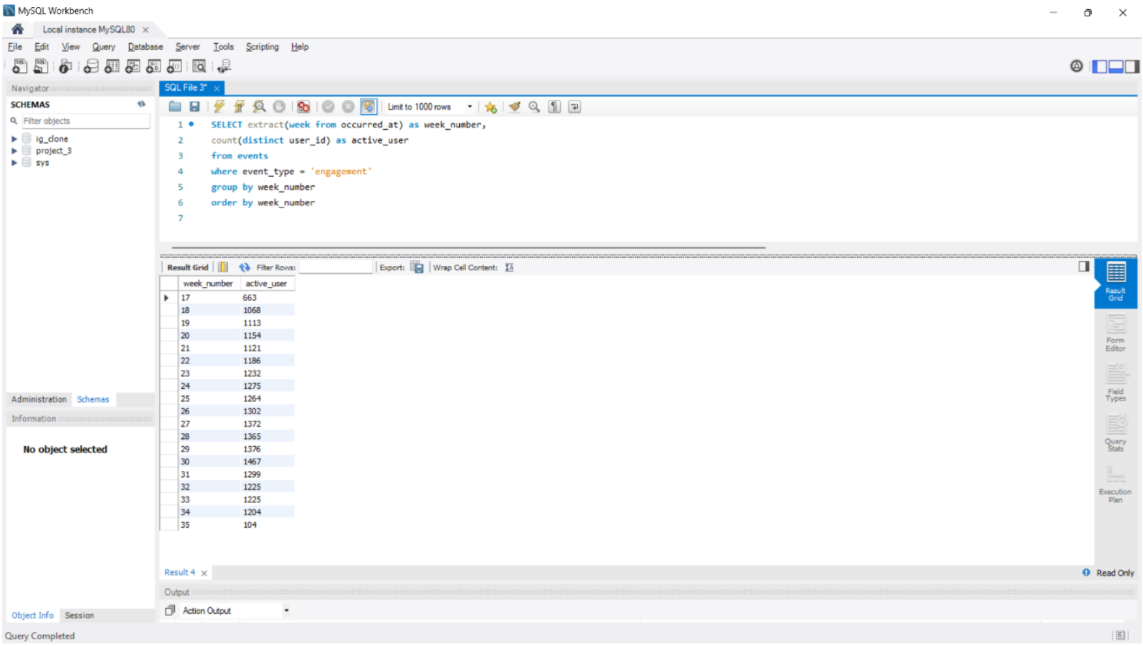
```sql
USE Project3_2;
LOAD DATA INFILE"C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/email_events.csv"
INTO TABLE email_events
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

```
#   Time      Action                                                                                         Message                                                                                      Duration / Fetch
13  13:25:05  LOAD DATA INFILE"C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\events.csv" INTO TABLE events F...  Error Code: 1290. The MySQL server is running with the --secure-file-priv option so it cannot execute this statem...  0.016 sec
14  13:25:41  USE Project3_2                                                                                 0 row(s) affected                                                                            0.000 sec
15  13:25:41  LOAD DATA INFILE"C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/events.csv" INTO TABLE events F...  325255 row(s) affected Records: 325255  Deleted: 0  Skipped: 0  Warnings: 0                 4.531 sec
16  13:27:16  USE Project3_2                                                                                 0 row(s) affected                                                                            0.015 sec
17  13:27:16  LOAD DATA INFILE"C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/email_events.csv" INTO TABLE e...  90389 row(s) affected Records: 90389  Deleted: 0  Skipped: 0  Warnings: 0               1.000 sec
```

TASK 1

Weekly User Engagement:

Objective: Measure the activeness of users on a weekly basis.



```sql
SELECT extract(week from occurred_at) as week_number,
count(distinct user_id) as active_user
from events
where event_type = 'engagement'
group by week_number
order by week_number
```

week_number	active_user
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

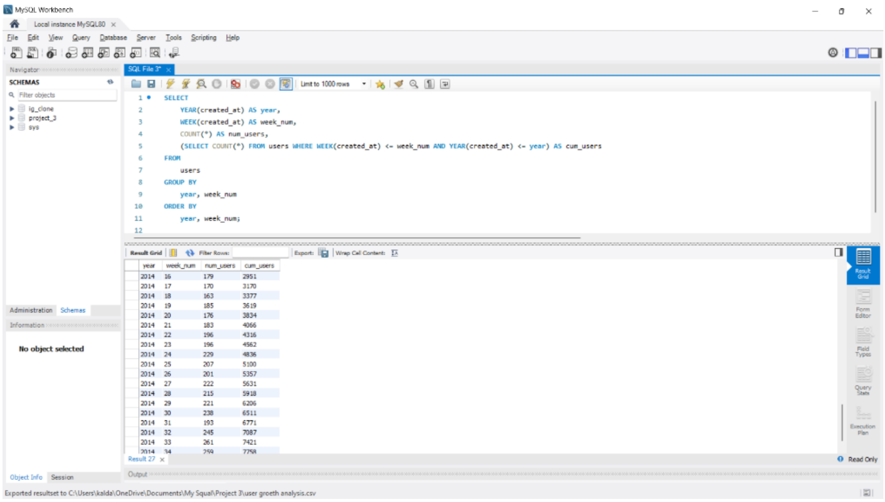
Query Completed

- Calculated weekly user engagement metrics.
- Identified trends in user engagement over time and possible factors influencing them.

TASK 2

User Growth Analysis:

Objective: Analyze the growth of users over time for a product



```sql
SELECT
    YEAR(created_at) AS year,
    WEEK(created_at) AS week_num,
    COUNT(*) AS num_users,
    (SELECT COUNT(*) FROM users WHERE WEEK(created_at) <= week_num AND YEAR(created_at) <= year) AS cum_users
FROM
    users
GROUP BY
    year, week_num
ORDER BY
    year, week_num;
```

year	week_num	num_users	cum_users
2014	16	179	2951
2014	17	170	3170
2014	18	163	3377
2014	19	185	3619
2014	20	176	3834
2014	21	183	4066
2014	22	196	4316
2014	23	196	4562
2014	24	229	4836
2014	25	207	5100
2014	26	201	5357
2014	27	222	5631
2014	28	215	5918
2014	29	221	6206
2014	30	238	6511
2014	31	193	6771
2014	32	245	7087
2014	33	261	7421
2014	34	259	7758

Exported resultset to C:\Users\kalda\OneDrive\Documents\My Squal\Project 3\user groeth analysis.csv

Full output

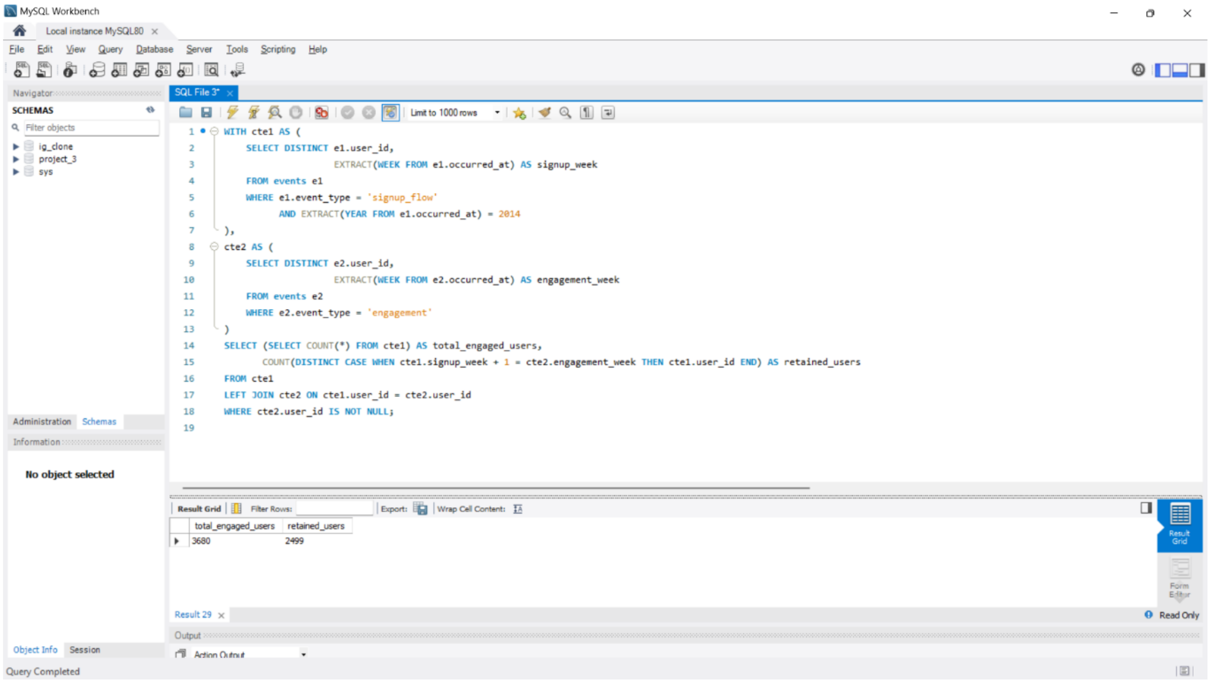
https://drive.google.com/drive/folders/1PxI2aI70V_RuCm3L7lbce0hNv7eCbqbJ?usp=sharing

- Analyzed user growth trends over time.
- Identified periods of accelerated growth and potential drivers behind them.

TASK4

Weekly Retention Analysis:

Objective: Analyze the retention of users on a weekly basis after signing up for a product.



```sql
WITH cte1 AS (
    SELECT DISTINCT e1.user_id,
                    EXTRACT(WEEK FROM e1.occurred_at) AS signup_week
    FROM events e1
    WHERE e1.event_type = 'signup_flow'
         AND EXTRACT(YEAR FROM e1.occurred_at) = 2014
),
cte2 AS (
    SELECT DISTINCT e2.user_id,
                    EXTRACT(WEEK FROM e2.occurred_at) AS engagement_week
    FROM events e2
    WHERE e2.event_type = 'engagement'
)
SELECT (SELECT COUNT(*) FROM cte1) AS total_engaged_users,
       COUNT(DISTINCT CASE WHEN cte1.signup_week + 1 = cte2.engagement_week THEN cte1.user_id END) AS retained_users
FROM cte1
LEFT JOIN cte2 ON cte1.user_id = cte2.user_id
WHERE cte2.user_id IS NOT NULL;
```

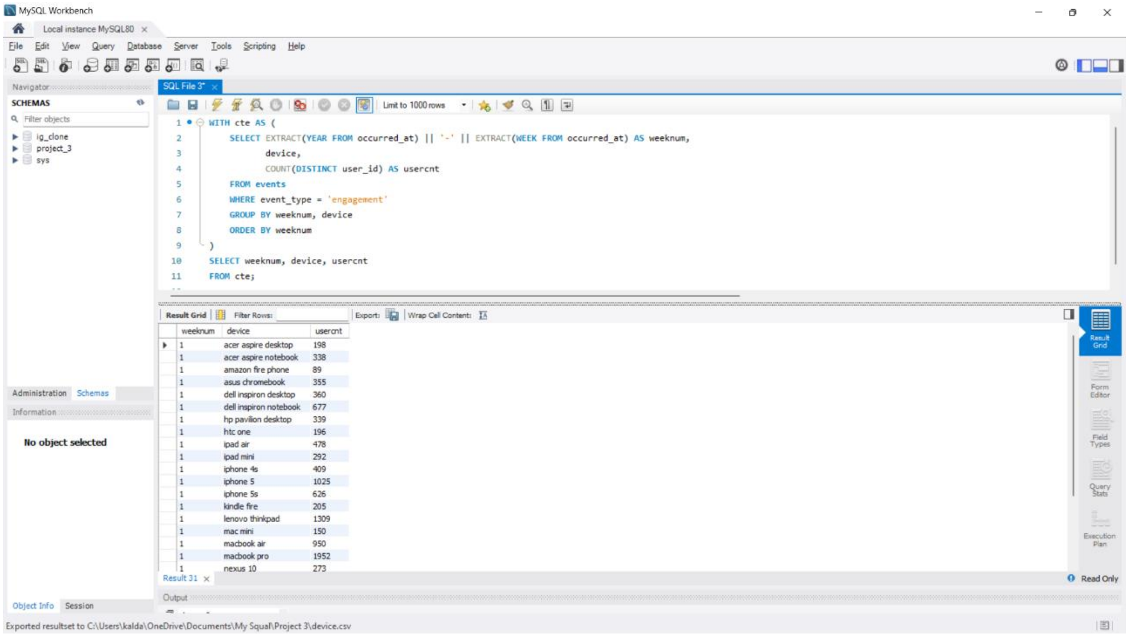
total_engaged_users	retained_users
3680	2499

- Calculated weekly retention rates based on user sign-up cohorts.
- Identified patterns in user retention and proposed strategies for improvement.

TASK 5

Weekly Engagement Per Device:

Objective: Measure the activeness of users on a weekly basis per device.



```sql
WITH cte AS (
    SELECT EXTRACT(YEAR FROM occurred_at) || '-' || EXTRACT(WEEK FROM occurred_at) AS weeknum,
           device,
           COUNT(DISTINCT user_id) AS usercnt
    FROM events
    WHERE event_type = 'engagement'
    GROUP BY weeknum, device
    ORDER BY weeknum
)
SELECT weeknum, device, usercnt
FROM cte;
```

weeknum	device	usercnt
1	acer aspire desktop	198
1	acer aspire notebook	338
1	amazon fire phone	89
1	asus chromebook	355
1	dell inspiron desktop	360
1	dell inspiron notebook	677
1	hp pavilion desktop	339
1	htc one	196
1	ipad air	478
1	ipad mini	292
1	iphone 4s	409
1	iphone 5	1025
1	iphone 5s	626
1	kindle fire	205
1	lenovo thinkpad	1309
1	mac mini	150
1	macbook air	950
1	macbook pro	1952
1	nexus 10	273

Exported resultset to C:\Users\kalda\OneDrive\Documents\My Squal\Project 3\device.csv

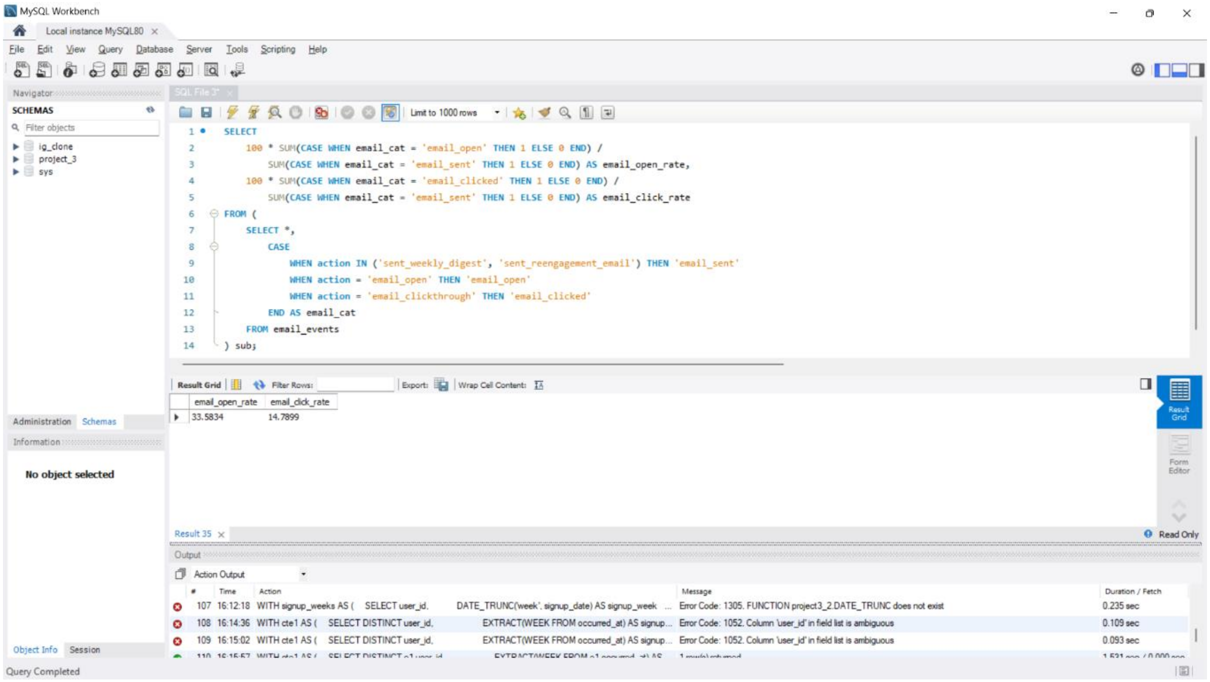
Output: https://drive.google.com/file/d/1Y1ICybnbxZRMccec4jeOLHz5Ka2kRM1T/view?usp=sharing

- Analyzed weekly engagement metrics per device.
- Identified variations in engagement levels across different devices and potential reasons for them

TASK 6

Email Engagement Analysis:

Objective: Analyze how users are engaging with the email service.



```sql
SELECT
    100 * SUM(CASE WHEN email_cat = 'email_open' THEN 1 ELSE 0 END) /
        SUM(CASE WHEN email_cat = 'email_sent' THEN 1 ELSE 0 END) AS email_open_rate,
    100 * SUM(CASE WHEN email_cat = 'email_clicked' THEN 1 ELSE 0 END) /
        SUM(CASE WHEN email_cat = 'email_sent' THEN 1 ELSE 0 END) AS email_click_rate
FROM (
    SELECT *,
        CASE
            WHEN action IN ('sent_weekly_digest', 'sent_reengagement_email') THEN 'email_sent'
            WHEN action = 'email_open' THEN 'email_open'
            WHEN action = 'email_clickthrough' THEN 'email_clicked'
        END AS email_cat
    FROM email_events
) sub;
```

email_open_rate	email_click_rate
33.5834	14.7899

```
#    Time      Action                                                                                  Message                                                               Duration / Fetch
107  16:12:18  WITH signup_weeks AS (   SELECT user_id,   DATE_TRUNC('week', signup_date) AS signup_week  ...  Error Code: 1305. FUNCTION project3_2.DATE_TRUNC does not exist  0.235 sec
108  16:14:36  WITH cte1 AS (   SELECT DISTINCT user_id,   EXTRACT(WEEK FROM occurred_at) AS signup...  Error Code: 1052. Column 'user_id' in field list is ambiguous       0.109 sec
109  16:15:02  WITH cte1 AS (   SELECT DISTINCT user_id,   EXTRACT(WEEK FROM occurred_at) AS signup...  Error Code: 1052. Column 'user_id' in field list is ambiguous       0.093 sec
110  16:15:57  WITH cte1 AS (   SELECT DISTINCT e1.user_id,   EXTRACT(WEEK FROM e1.occurred_at) AS  1 row(s) returned                                                    1.531 sec / 0.000 sec
```

Query Completed

- Calculated email engagement metrics such as open rates and click-through rates.
- Identified areas for optimizing email engagement strategies.

Results

The project's key results included the identification of reviewed jobs distribution across languages calculation of retention Rates

SQL is one of the most crucial skills for anyone in a data driven position. Analysts can effectively contribute to improving daily operations, optimizing user engagement, and boosting sales. By delivering actionable insights, the Lead Data Analyst plays a pivotal role in driving the company's success and ensuring it remains agile and responsive in a dynamic business landscape.

Additionally,

- this project helped me to gain insight of various factors which are
- crucially important for the business to run for a long period and grow to a new height