Project 3

Submitted By

Swathy Mugundan

Project 3

Table Of Contents

Problem Statement

- Operational Analytics is a crucial process that involves analyzing a company's end-to-end operations. This analysis helps identify areas for improvement within the company. As a Data Analyst, you'll work closely with various teams, such as operations, support, and marketing, helping them derive valuable insights from the data they collect.

- One of the key aspects of Operational Analytics is investigating metric spikes. This involves understanding and explaining sudden changes in key metrics, such as a dip in daily user engagement or a drop in sales. As a Data Analyst, you'll need to answer these questions daily, making it crucial to understand how to investigate these metric spikes.

- In this project, you'll take on the role of a Lead Data Analyst at a company like Microsoft. You'll be provided with various datasets and tables, and your task will be to derive insights from this data to answer questions posed by different departments within the company. Your goal is to use your advanced SQL skills to analyze the data and provide valuable insights that can help improve the company's operations and understand sudden changes in key metrics.

Project Approach

DB Data

Download the dataset (CSV) files and save them in a folder. Using the **Table Data Import Wizard** form the database and tables.

Problem Def'n

Analyse the tables and its attributes and understand the references between tables. Read the question properly.

Query Formation

Figure out the related tables and write the suitable SQL Statement for fetching the results

Result Analysis

Analyse and understand the result set and capture the inference of the problem statement

Tech-Stack Used

My SQL Server

User for the Relational Database Management System. To create database, To create Tables.

Teampaper Snap

To take screenshots of the result-grid of the SQL Statements from SQL Workbench

MySQL Workbench

It is a GUI Based tool for Querying the Database. It is a visual-based DB tool to observe the databases , its tables, its attributes and how they are being stored in the system.

MS-EXCEL

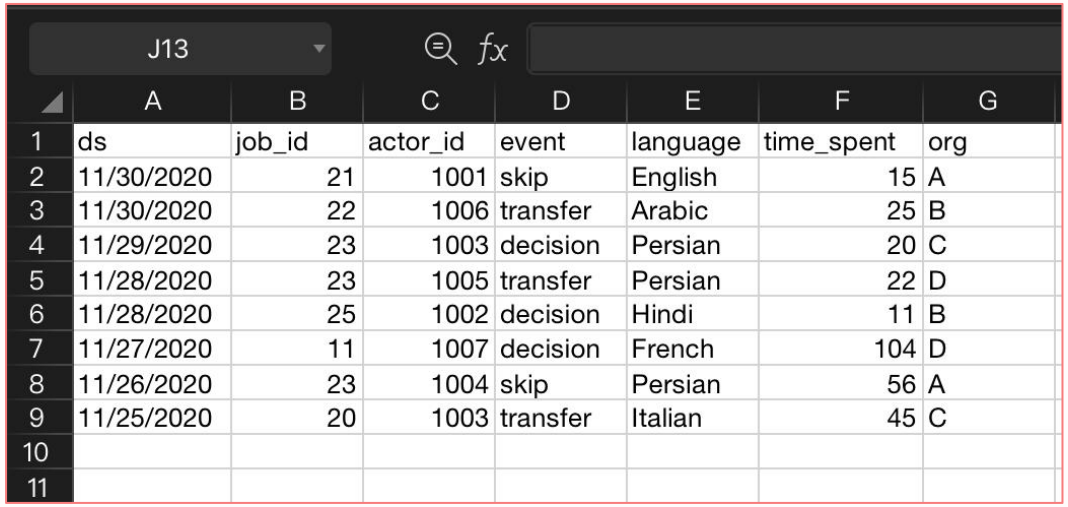
To open the CSV Files, view the attributes and records. Identify the NULL values and remove them. Check the data values.

Case Study 1

Job Data Analysis

Database Description

- Table named **job_data** will be used for the analysis.
- The attributes are explained as follows :
  - **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).

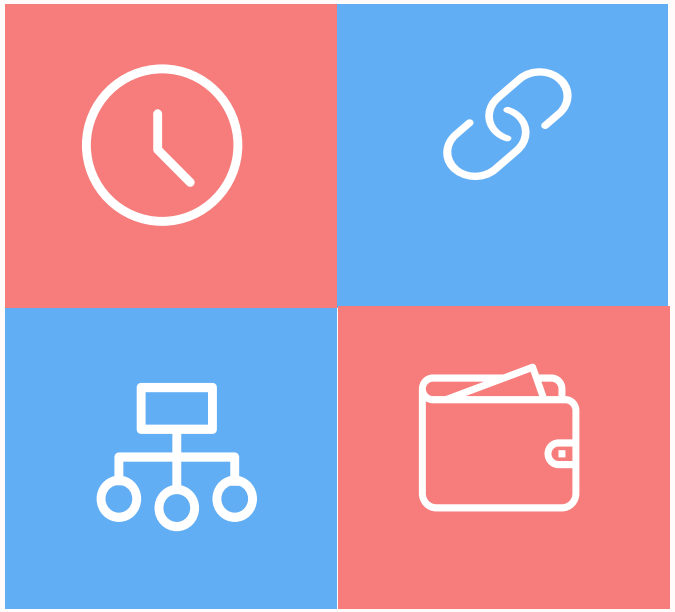


J13

	A	B	C	D	E	F	G
1	ds	job_id	actor_id	event	language	time_spent	org
2	11/30/2020	21	1001	skip	English	15	A
3	11/30/2020	22	1006	transfer	Arabic	25	B
4	11/29/2020	23	1003	decision	Persian	20	C
5	11/28/2020	23	1005	transfer	Persian	22	D
6	11/28/2020	25	1002	decision	Hindi	11	B
7	11/27/2020	11	1007	decision	French	104	D
8	11/26/2020	23	1004	skip	Persian	56	A
9	11/25/2020	20	1003	transfer	Italian	45	C
10							
11							

Job_Data CSV Screenshot

Project Approach



Jobs Reviewed Over Time

To calculate the number of jobs reviewed per hour for each day in November 2020.

Throughput Analysis

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

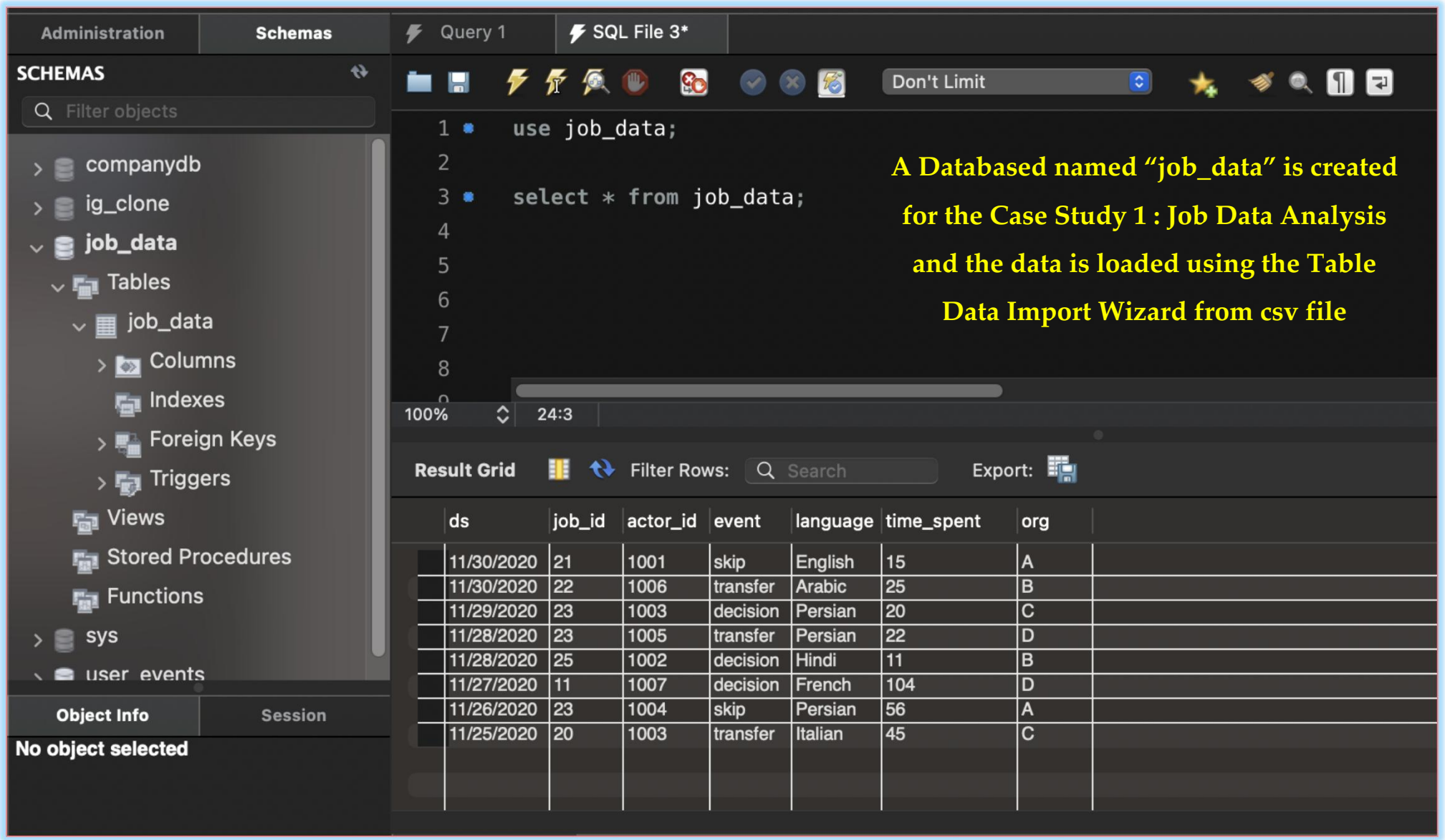
Language Share Analysis

To calculate the percentage share of each language in the last 30 days.

Duplicate Rows Detection

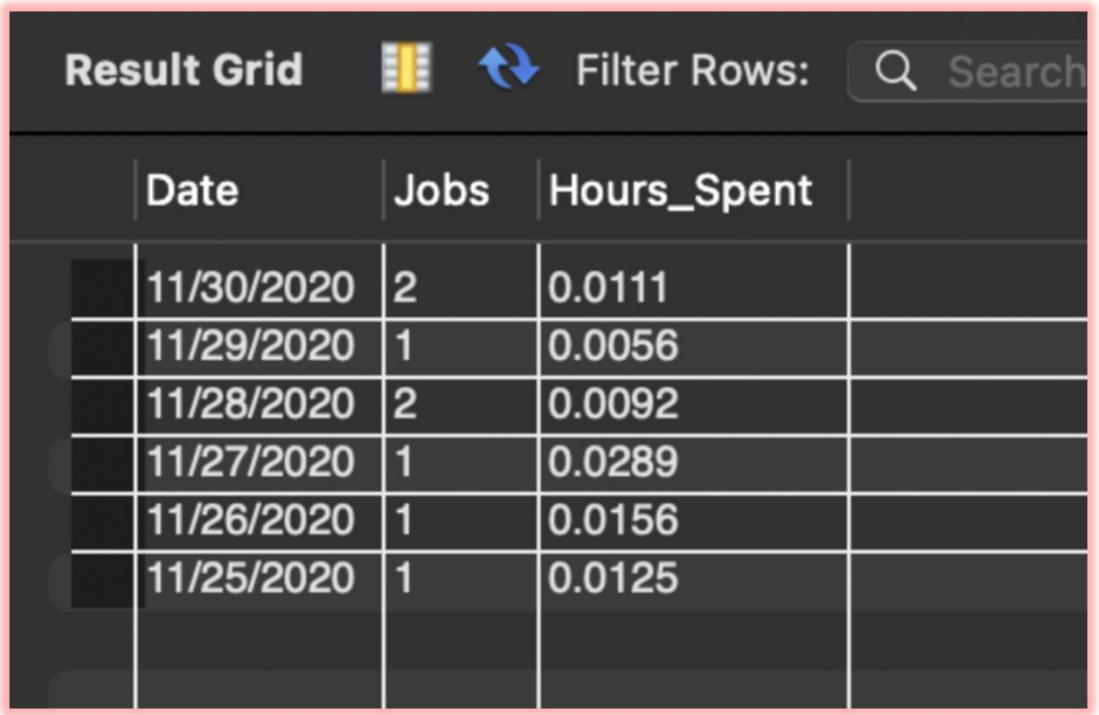
Identify duplicate rows in the data.

Job Data Analysis : Database & Table Creation Screenshot



Job Data Analysis : Job Reviewed Over Time

```
select ds as Date,
        count(ds) as Jobs,
        sum(time_spent)/3600 as Hours_Spent
from job_data
group by ds ;
```



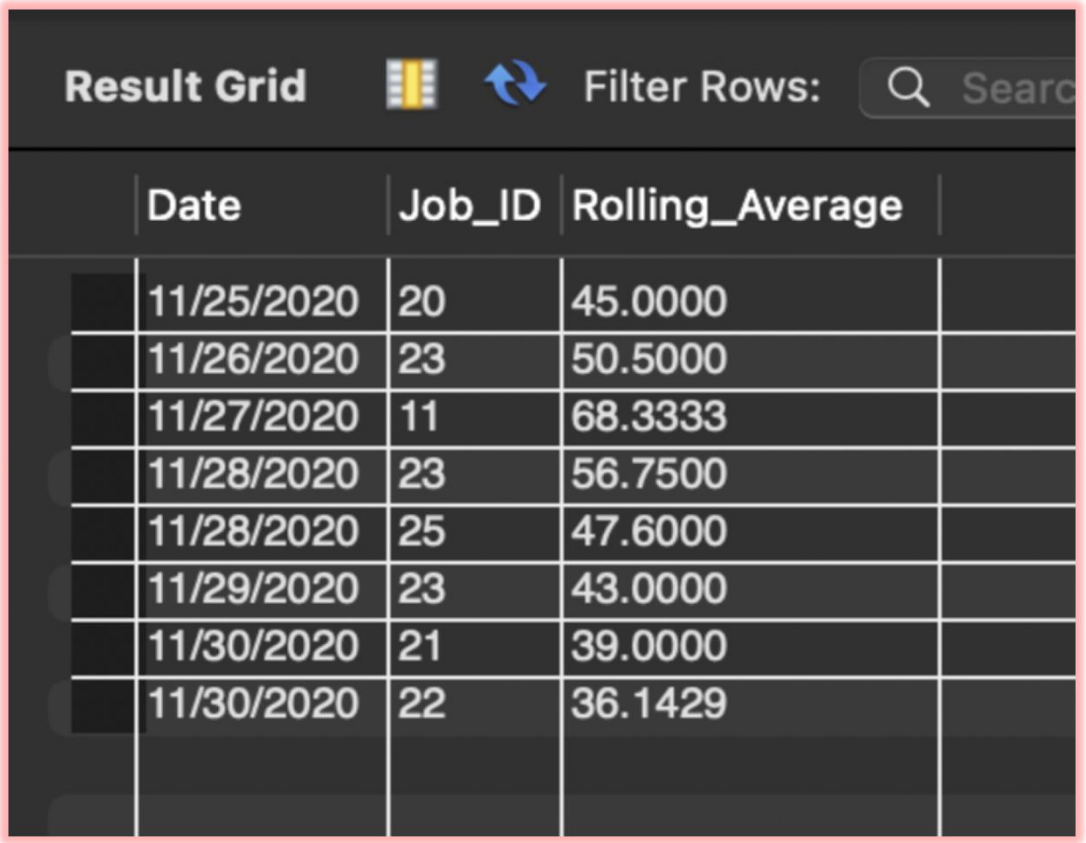
Date	Jobs	Hours_Spent
11/30/2020	2	0.0111
11/29/2020	1	0.0056
11/28/2020	2	0.0092
11/27/2020	1	0.0289
11/26/2020	1	0.0156
11/25/2020	1	0.0125

NOTE : The given time was in seconds. Hence, converted that to hours by dividing it by 3600. Also, all the data are in month of November. So where clause between the date is omitted

Job Data Analysis : Throughput Analysis (Rolling Average)

```
SELECT   ds as Date,    job_id as Job_ID,
   AVG(time_spent)
        OVER  (ORDER BY ds
              ROWS BETWEEN 6 PRECEDING AND CURRENT
ROW) AS Rolling_Average
FROM job_data;
```

NOTE : In order to track the trends in analysis, Rolling Averages are preferred as the daily metric can have unwanted noise in the data. Also, here to track the 7-day pattern (specific time series) as mentioned rolling average is best to be used to spot the trend.



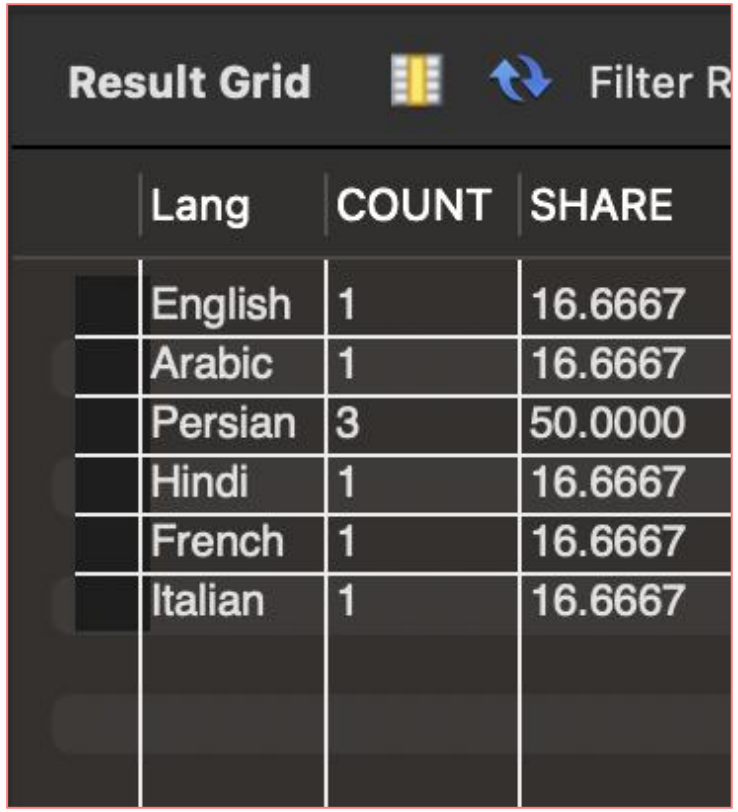
Date	Job_ID	Rolling_Average
11/25/2020	20	45.0000
11/26/2020	23	50.5000
11/27/2020	11	68.3333
11/28/2020	23	56.7500
11/28/2020	25	47.6000
11/29/2020	23	43.0000
11/30/2020	21	39.0000
11/30/2020	22	36.1429

Job Data Analysis : Language Share Analysis

```
SELECT
        language as Lang,
        count(language) as COUNT,
        (count(language) * 100)/6 as SHARE
from job_data
group by language;
```

NOTE : The total number of languages is 6.

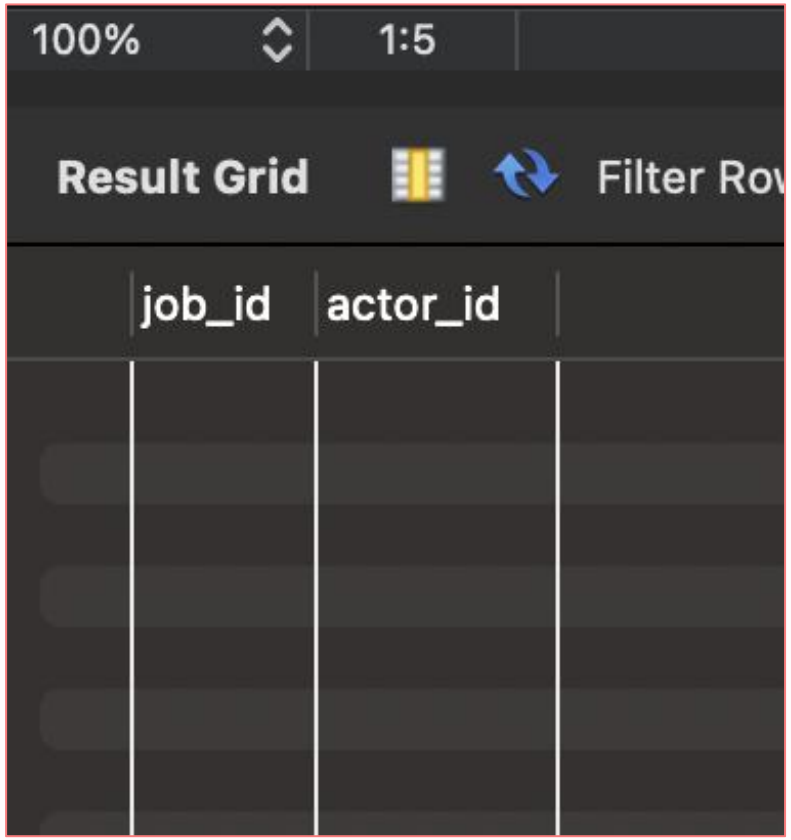
Result Inference : Persian Language has a maximum share in the data.



Lang	COUNT	SHARE
English	1	16.6667
Arabic	1	16.6667
Persian	3	50.0000
Hindi	1	16.6667
French	1	16.6667
Italian	1	16.6667

Job Data Analysis : Duplicate Data Detection

```
SELECT
        job_id, actor_id
FROM job_data
GROUP BY job_id, actor_id
HAVING
    (COUNT(job_id)>1);
```



NOTE : Returns an empty result-set depicting NULL values. From this, we can conclude that there are no duplicate rows in the job_data table.

Case Study 2

Investigating Metric Spike

Database Description

- For this project, we will be working with **Three Tables**.
- **users:** Contains one row per user, with descriptive information about that user's account.
- **events:** Contains one row per event, where an event is an action that a user has taken (e.g., login, messaging, search).
- **email_events:** Contains events specific to the sending of emails.

Sample CSV Data

Screenshots are available in the Next Slide

Database Description - CSV Files

A	B	C	D	E	F
user_id	created_at	company_	language	activated_at	state
0	01/01/2013 20:59	5737	english	01/01/2013 21:01	active
3	01/01/2013 18:40	2800	german	01/01/2013 18:42	active
4	01/01/2013 14:37	5110	indian	01/01/2013 14:39	active
6	01/01/2013 18:37	11699	english	01/01/2013 18:38	active
7	01/01/2013 16:19	4765	french	01/01/2013 16:20	active
8	01/01/2013 04:38	2698	french	01/01/2013 04:40	active
11	01/01/2013 08:07	3745	english	01/01/2013 08:09	active
13	02/01/2013 12:27	4025	english	02/01/2013 12:29	active
15	02/01/2013 15:39	4259	english	02/01/2013 15:41	active
17	02/01/2013 10:56	5025	japanese	02/01/2013 10:57	active
19	02/01/2013 09:54	326	english	02/01/2013 09:55	active
20	02/01/2013 09:41	7	italian	02/01/2013 09:43	active
21	02/01/2013 09:29	2606	english	02/01/2013 09:30	active
22	02/01/2013 17:36	545	german	02/01/2013 17:38	active

users table

events table

A	B	C	D	E
user_id	occurred_at	action	user_type	
0	06/05/2014 09:30	sent_weekly_digest	1	
0	13/05/2014 09:30	sent_weekly_digest	1	
0	20/05/2014 09:30	sent_weekly_digest	1	
0	27/05/2014 09:30	sent_weekly_digest	1	
0	03/06/2014 09:30	sent_weekly_digest	1	
0	03/06/2014 09:30	email_open	1	
0	10/06/2014 09:30	sent_weekly_digest	1	
0	10/06/2014 09:30	email_open	1	
0	17/06/2014 09:30	sent_weekly_digest	1	
0	17/06/2014 09:30	email_open	1	
0	24/06/2014 09:30	sent_weekly_digest	1	
0	01/07/2014 09:30	sent_weekly_digest	1	
0	08/07/2014 09:30	sent_weekly_digest	1	
0	15/07/2014 09:30	sent_weekly_digest	1	
0	22/07/2014 09:30	sent_weekly_digest	1	

email_events table

A	B	C	D	E	F	G	H
user_id	occurred_at	event_type	event_name	location	device	user_type	
10522	02/05/2014 11:02	engagement	login	Japan	dell inspiron notebook	3	
10522	02/05/2014 11:02	engagement	home_page	Japan	dell inspiron notebook	3	
10522	02/05/2014 11:03	engagement	like_message	Japan	dell inspiron notebook	3	
10522	02/05/2014 11:04	engagement	view_inbox	Japan	dell inspiron notebook	3	
10522	02/05/2014 11:03	engagement	search_run	Japan	dell inspiron notebook	3	
10522	02/05/2014 11:03	engagement	search_run	Japan	dell inspiron notebook	3	
10612	01/05/2014 09:59	engagement	login	Netherlands	iphone 5	1	
10612	01/05/2014 10:00	engagement	like_message	Netherlands	iphone 5	1	
10612	01/05/2014 10:00	engagement	send_message	Netherlands	iphone 5	1	
10612	01/05/2014 10:01	engagement	home_page	Netherlands	iphone 5	1	
10612	01/05/2014 10:01	engagement	like_message	Netherlands	iphone 5	1	
10612	01/05/2014 10:02	engagement	home_page	Netherlands	iphone 5	1	
10612	01/05/2014 10:02	engagement	view_inbox	Netherlands	iphone 5	1	
10612	01/05/2014 10:03	engagement	like_message	Netherlands	iphone 5	1	
10612	01/05/2014 10:03	engagement	home_page	Netherlands	iphone 5	1	
10612	01/05/2014 10:04	engagement	send_message	Netherlands	iphone 5	1	
10612	01/05/2014 10:04	engagement	like_message	Netherlands	iphone 5	1	
10612	01/05/2014 10:05	engagement	send_message	Netherlands	iphone 5	1	
10736	09/05/2014 17:52	engagement	login	Austria	iphone 4s	2	

Case Study 2 : Project Approach

Weekly User Engagment

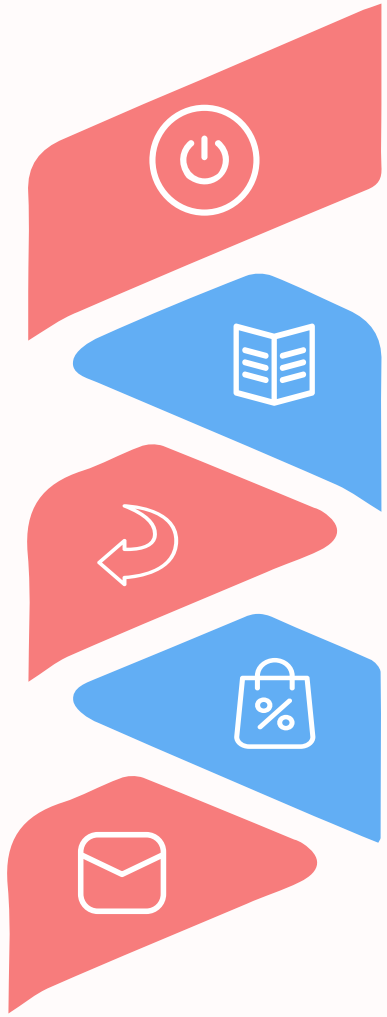
Measure the activeness of users on a weekly basis.

Weekly Retention Analysis

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

Email Engagement Analysis

Analyze how users are engaging with the email service.



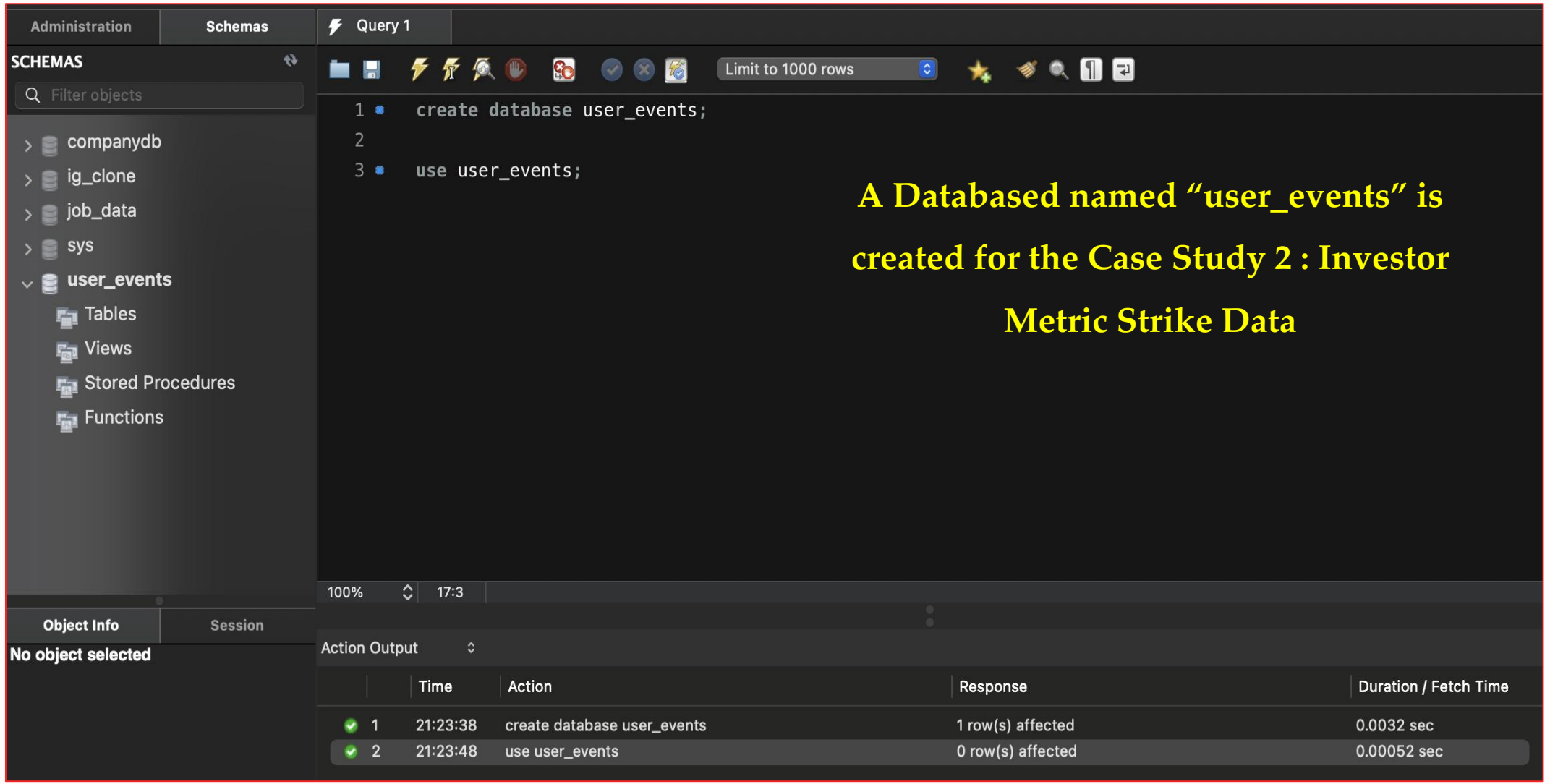
User Growth Analysis

Analyze the growth of users over time for a product on weekly basis

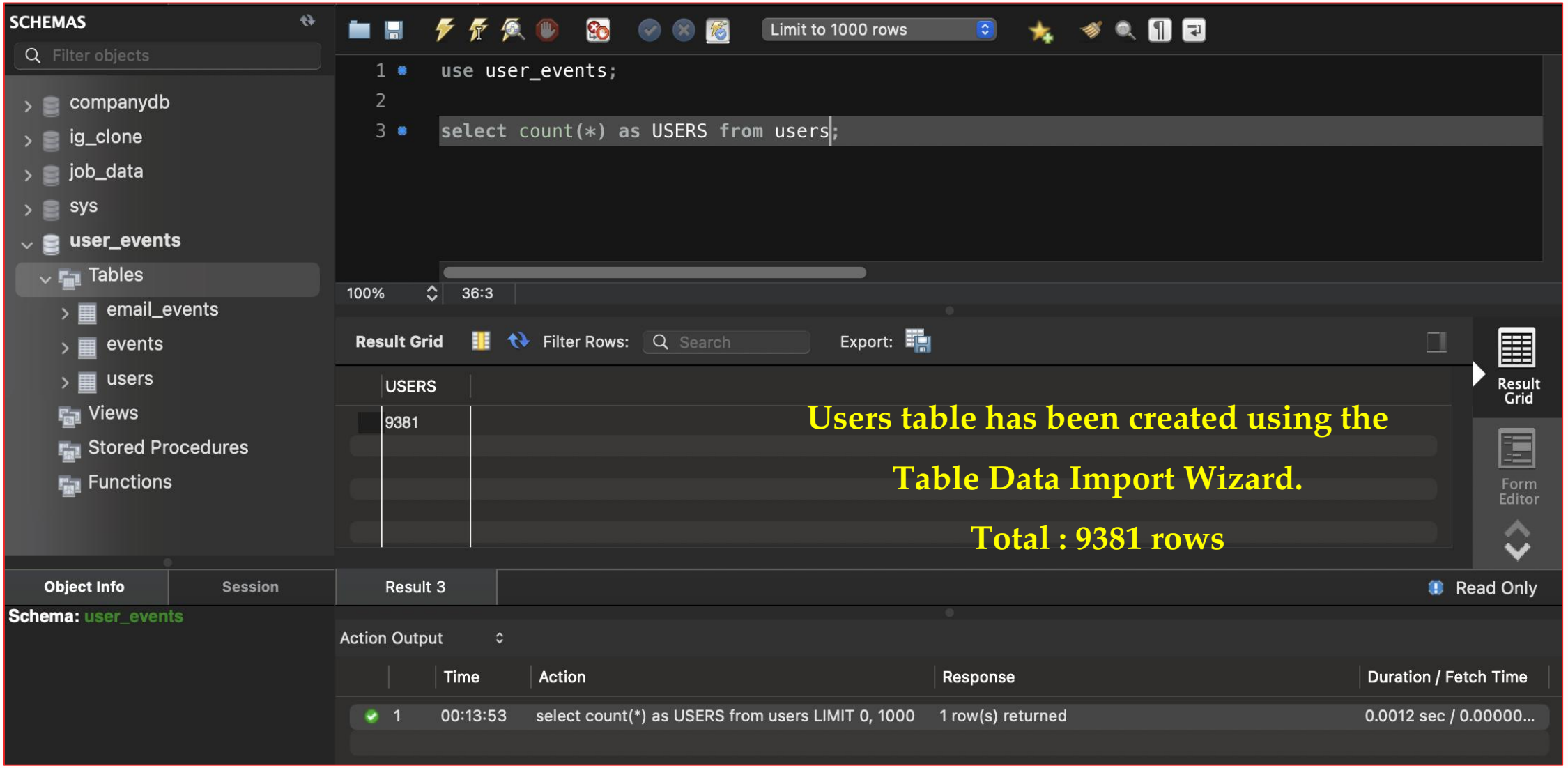
Weekly Engagement (per Device)

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

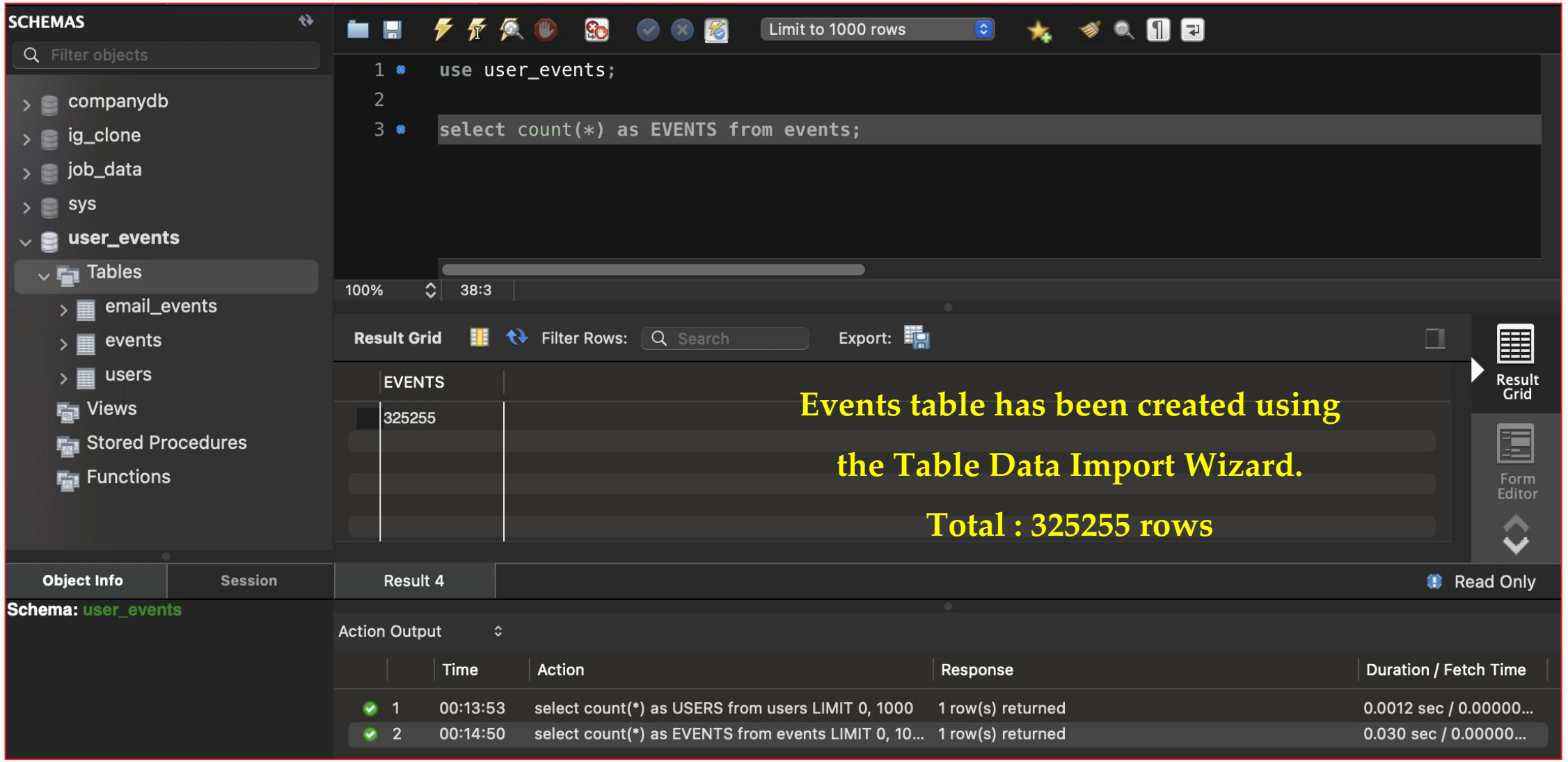
Case Study 2 : Database Creation Screenshot



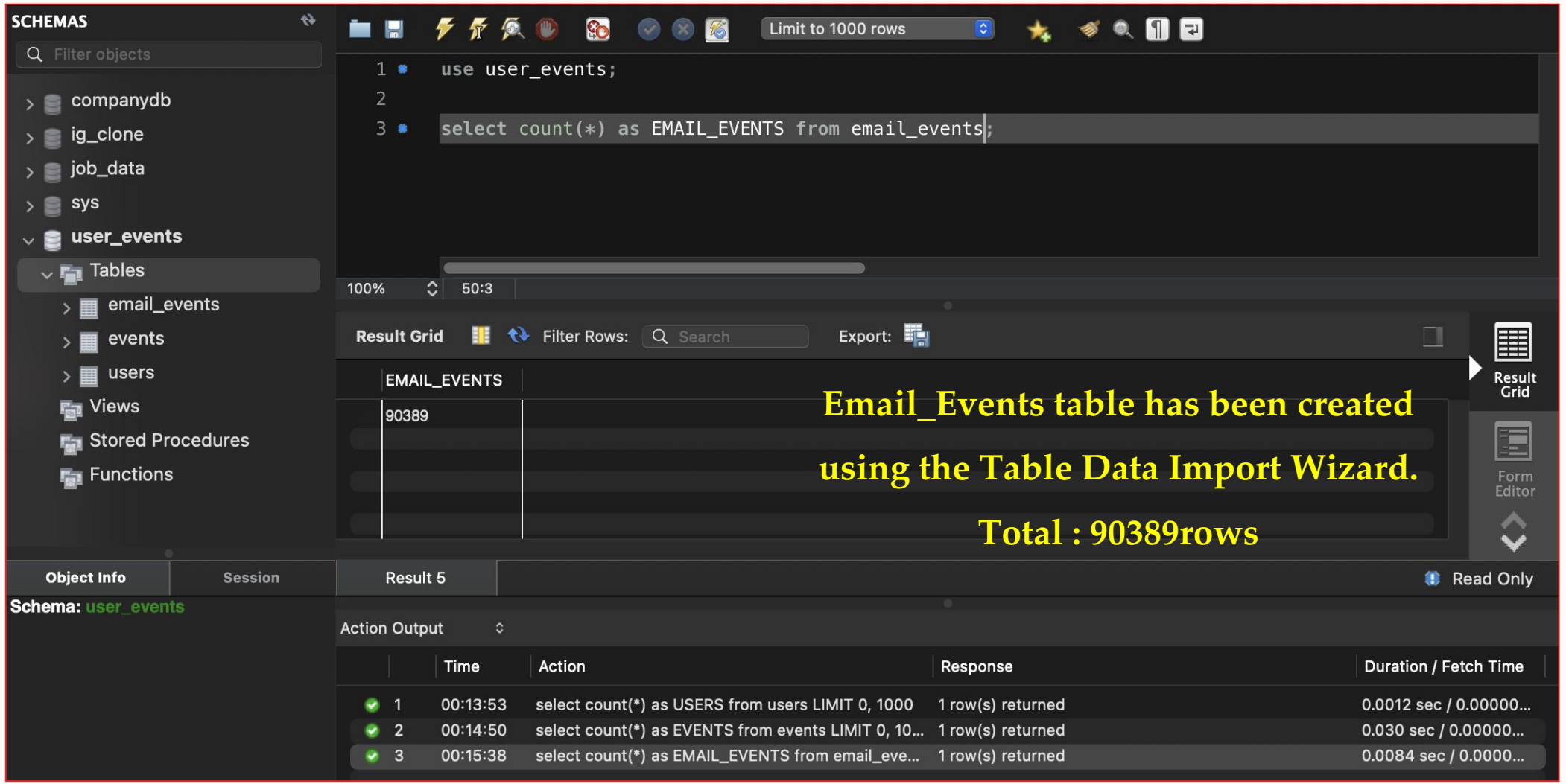
Case Study 2 : “User” table Creation Screenshot



Case Study 2 : “Events” table Creation Screenshot

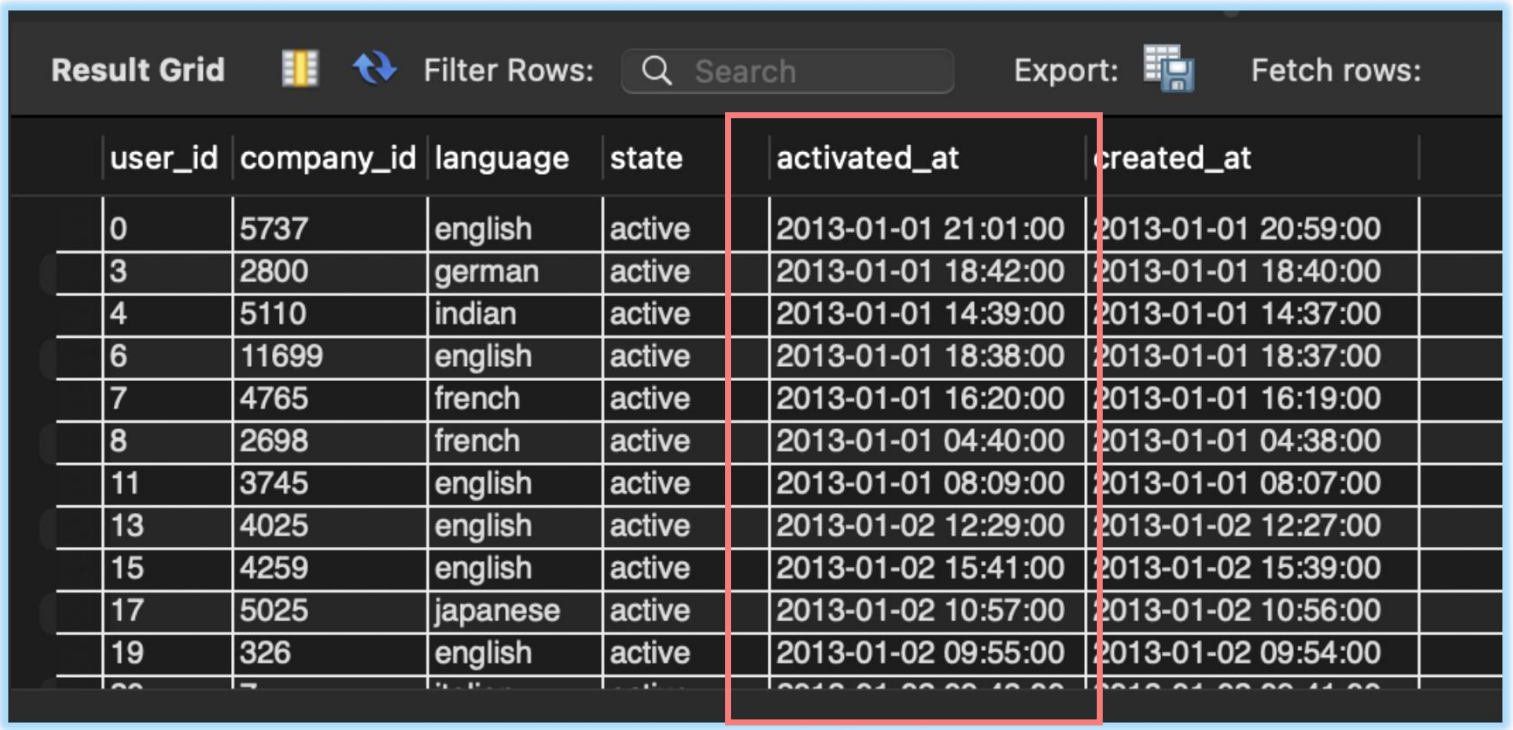


Case Study 2 : “Email_Events” table Creation Screenshot



Case Study 2 : alter users table datatype text to datetime

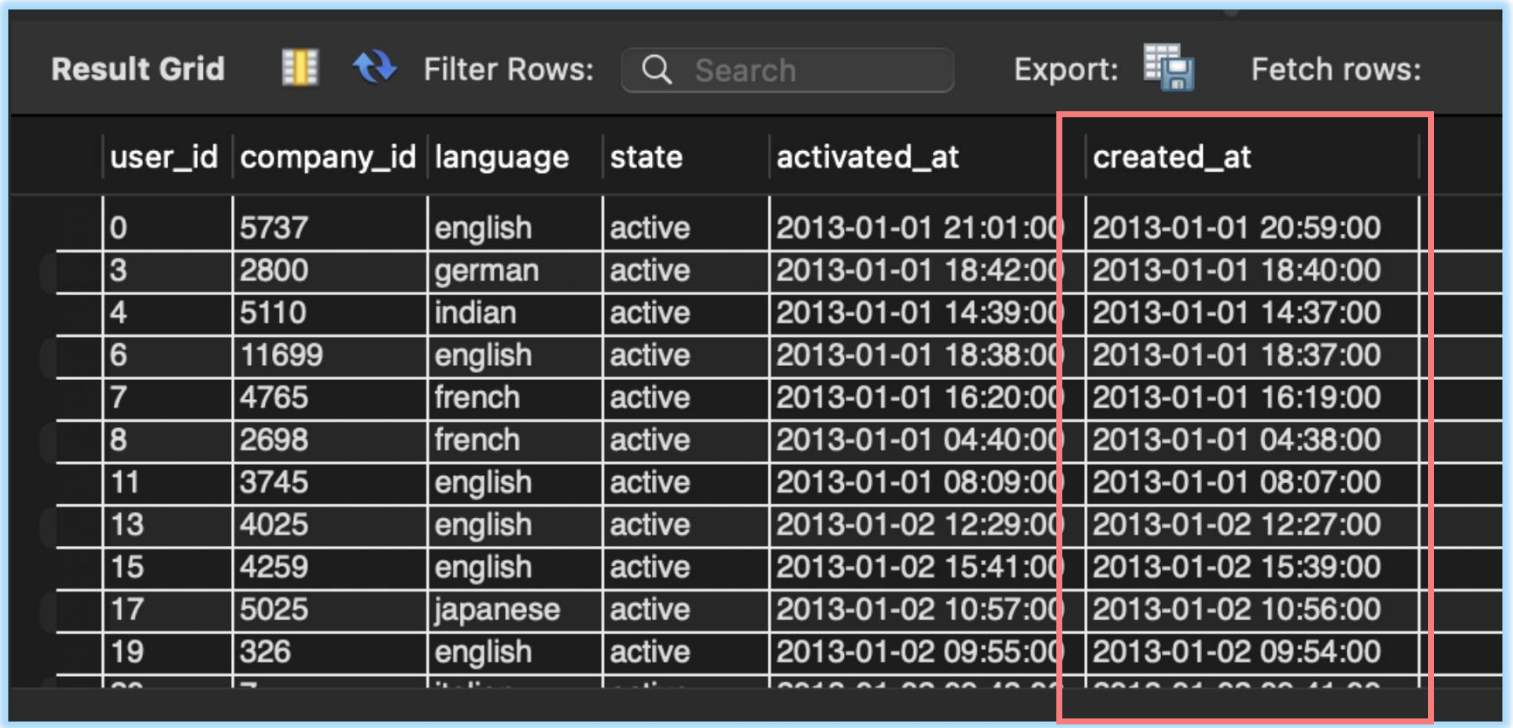
```
SET SQL_SAFE_UPDATES = 0;

alter table users add column temp
DATETIME;

update users set temp =
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 DATETIME;
```



user_id	company_id	language	state	activated_at	created_at
0	5737	english	active	2013-01-01 21:01:00	2013-01-01 20:59:00
3	2800	german	active	2013-01-01 18:42:00	2013-01-01 18:40:00
4	5110	indian	active	2013-01-01 14:39:00	2013-01-01 14:37:00
6	11699	english	active	2013-01-01 18:38:00	2013-01-01 18:37:00
7	4765	french	active	2013-01-01 16:20:00	2013-01-01 16:19:00
8	2698	french	active	2013-01-01 04:40:00	2013-01-01 04:38:00
11	3745	english	active	2013-01-01 08:09:00	2013-01-01 08:07:00
13	4025	english	active	2013-01-02 12:29:00	2013-01-02 12:27:00
15	4259	english	active	2013-01-02 15:41:00	2013-01-02 15:39:00
17	5025	japanese	active	2013-01-02 10:57:00	2013-01-02 10:56:00
19	326	english	active	2013-01-02 09:55:00	2013-01-02 09:54:00

Case Study 2 : alter users table datatype text to datetime

```
SET SQL_SAFE_UPDATES = 0;

alter table users add column temp
DATETIME;

update users set temp =
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 DATETIME;
```



user_id	company_id	language	state	activated_at	created_at
0	5737	english	active	2013-01-01 21:01:00	2013-01-01 20:59:00
3	2800	german	active	2013-01-01 18:42:00	2013-01-01 18:40:00
4	5110	indian	active	2013-01-01 14:39:00	2013-01-01 14:37:00
6	11699	english	active	2013-01-01 18:38:00	2013-01-01 18:37:00
7	4765	french	active	2013-01-01 16:20:00	2013-01-01 16:19:00
8	2698	french	active	2013-01-01 04:40:00	2013-01-01 04:38:00
11	3745	english	active	2013-01-01 08:09:00	2013-01-01 08:07:00
13	4025	english	active	2013-01-02 12:29:00	2013-01-02 12:27:00
15	4259	english	active	2013-01-02 15:41:00	2013-01-02 15:39:00
17	5025	japanese	active	2013-01-02 10:57:00	2013-01-02 10:56:00
19	326	english	active	2013-01-02 09:55:00	2013-01-02 09:54:00

Case Study 2 : alter events table datatype text to datetime

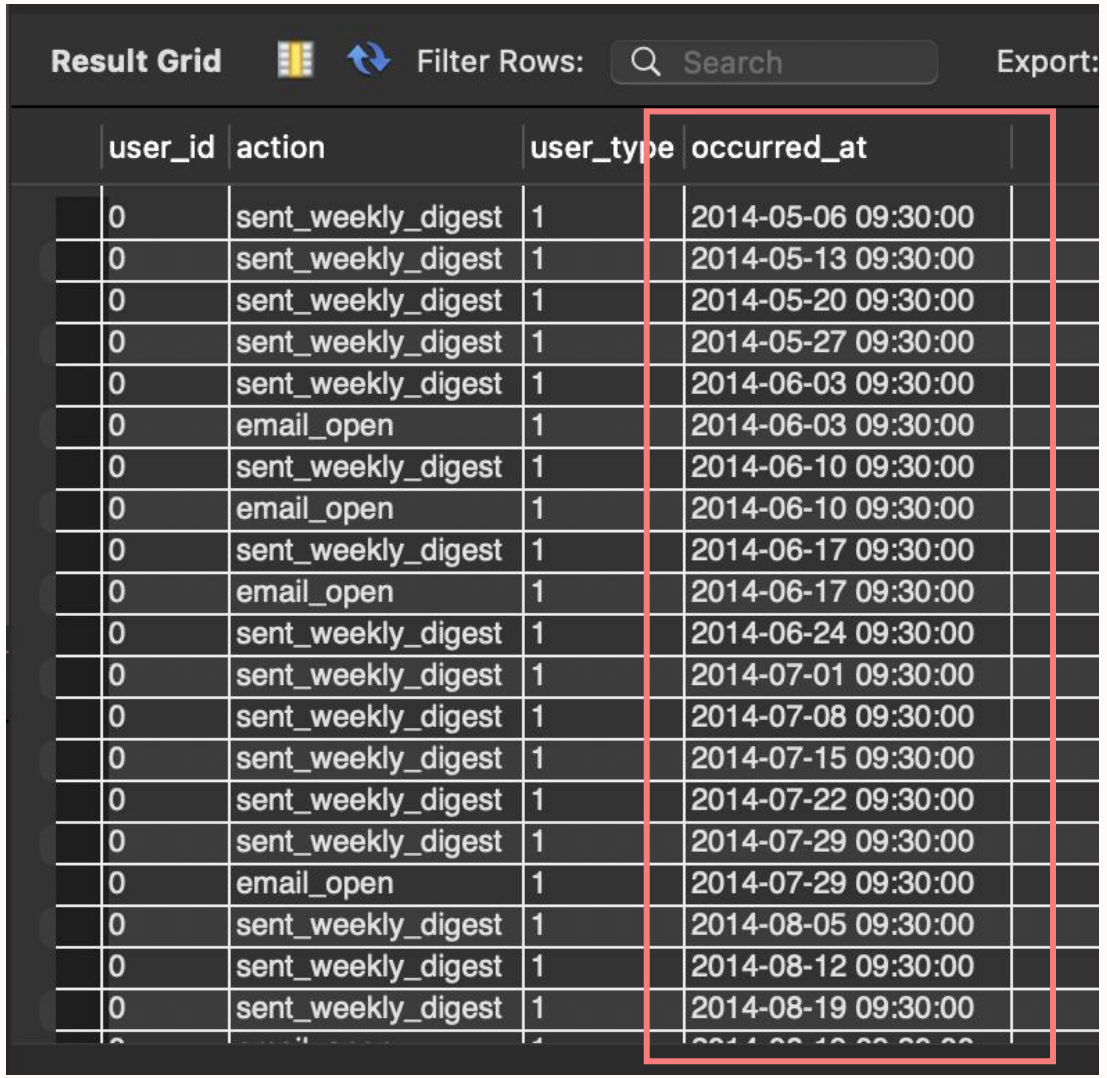
```
SET SQL_SAFE_UPDATES = 0;

alter table events add column temp
DATETIME;

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

alter table events drop column
occurred_at;

alter table events change column temp
occurred_at DATETIME;

select * from events;
```

Result Grid | Filter Rows: Search | Export: | Fetch rows:

user_id	event_type	event_name	location	device	user_type	occurred_at
10522	engagement	login	Japan	dell inspiron notebook	3	2014-05-02 11:02:00
10522	engagement	home_page	Japan	dell inspiron notebook	3	2014-05-02 11:02:00
10522	engagement	like_message	Japan	dell inspiron notebook	3	2014-05-02 11:03:00
10522	engagement	view_inbox	Japan	dell inspiron notebook	3	2014-05-02 11:04:00
10522	engagement	search_run	Japan	dell inspiron notebook	3	2014-05-02 11:03:00
10522	engagement	search_run	Japan	dell inspiron notebook	3	2014-05-02 11:03:00
10612	engagement	login	Netherlands	iphone 5	1	2014-05-01 09:59:00
10612	engagement	like_message	Netherlands	iphone 5	1	2014-05-01 10:00:00
10612	engagement	send_message	Netherlands	iphone 5	1	2014-05-01 10:00:00
10612	engagement	home_page	Netherlands	iphone 5	1	2014-05-01 10:01:00
10612	engagement	like_message	Netherlands	iphone 5	1	2014-05-01 10:01:00
10612	engagement	home_page	Netherlands	iphone 5	1	2014-05-01 10:02:00
10612	engagement	view_inbox	Netherlands	iphone 5	1	2014-05-01 10:02:00
10612	engagement	like_message	Netherlands	iphone 5	1	2014-05-01 10:03:00
10612	engagement	home_page	Netherlands	iphone 5	1	2014-05-01 10:03:00
10612	engagement	send_message	Netherlands	iphone 5	1	2014-05-01 10:04:00
10612	engagement	like_message	Netherlands	iphone 5	1	2014-05-01 10:04:00
10612	engagement	send_message	Netherlands	iphone 5	1	2014-05-01 10:05:00

Case Study 2 : alter email_ events table datatype text to datetime

```
SET SQL_SAFE_UPDATES = 0;

alter table email_events add column temp
DATETIME;

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

alter table email_events drop column occurred_at;

alter table email_events change column temp
occurred_at DATETIME;

select * from email_events;
```



user_id	action	user_type	occurred_at
0	sent_weekly_digest	1	2014-05-06 09:30:00
0	sent_weekly_digest	1	2014-05-13 09:30:00
0	sent_weekly_digest	1	2014-05-20 09:30:00
0	sent_weekly_digest	1	2014-05-27 09:30:00
0	sent_weekly_digest	1	2014-06-03 09:30:00
0	email_open	1	2014-06-03 09:30:00
0	sent_weekly_digest	1	2014-06-10 09:30:00
0	email_open	1	2014-06-10 09:30:00
0	sent_weekly_digest	1	2014-06-17 09:30:00
0	email_open	1	2014-06-17 09:30:00
0	sent_weekly_digest	1	2014-06-24 09:30:00
0	sent_weekly_digest	1	2014-07-01 09:30:00
0	sent_weekly_digest	1	2014-07-08 09:30:00
0	sent_weekly_digest	1	2014-07-15 09:30:00
0	sent_weekly_digest	1	2014-07-22 09:30:00
0	sent_weekly_digest	1	2014-07-29 09:30:00
0	email_open	1	2014-07-29 09:30:00
0	sent_weekly_digest	1	2014-08-05 09:30:00
0	sent_weekly_digest	1	2014-08-12 09:30:00
0	sent_weekly_digest	1	2014-08-19 09:30:00

Case Study 2 : Weekly User Engagement

```
select extract(week from occurred_at) as Week_Num,
count(user_id) as Users_Count
from events
where
        event_type = 'engagement'
        AND event_name = 'login'
group by Week_Num;
```

NOTE : The result-set describes the number of users who have logged_in on weekly basis.

The maximum is week 30 and minimum is week 35

Result Grid Filter Rows:

Week_Num	Users_Count
17	887
18	1985
19	2030
20	2093
21	1986
23	2188
22	2157
24	2265
25	2244
29	2433
26	2266
30	2583
28	2493
27	2397
31	2278
32	2098
33	2071
34	2052
35	104

Case Study 2 : User Growth Analysis

```
select week_num as Week, year_num as Year, active_users as Users,
sum(active_users) over (order by year_num, week_num rows between unbounded
preceding and current row) as Cumulative
from (select count(distinct user_id) as active_users,
extract(week from activated_at) as week_num,
extract(year from activated_at) as year_num
from users where state='active'
group by year_num, week_num , order by year_num, week_num)t;
```

NOTE : Result set contains the number of active users on weekly basis. The cumulative column gives the relative count of the users based on previous weeks.

THE RESULT-SET OF THE QUERY IS TOO LARGE. ATTACHED SCREENSHOTS IN THE FOLLOWING SLIDES PLEASE REFER NEXT 2 SLIDES

Case Study 2 : User Growth Analysis - Result Set(1)

Result Grid Filter Rows:

Week	Year	Users	Cumulative
0	2013	23	23
1	2013	30	53
2	2013	48	101
3	2013	36	137
4	2013	30	167
5	2013	48	215
6	2013	38	253
7	2013	42	295
8	2013	34	329
9	2013	43	372
10	2013	32	404
11	2013	31	435
12	2013	33	468
13	2013	39	507
14	2013	35	542
15	2013	43	585
16	2013	46	631
17	2013	49	680
18	2013	44	724
19	2013	57	781

Result Grid Filter Rows:

Week	Year	Users	Cumulative
20	2013	39	820
21	2013	49	869
22	2013	54	923
23	2013	50	973
24	2013	45	1018
25	2013	57	1075
26	2013	56	1131
27	2013	52	1183
28	2013	72	1255
29	2013	67	1322
30	2013	67	1389
31	2013	67	1456
32	2013	71	1527
33	2013	73	1600
34	2013	78	1678
35	2013	63	1741
36	2013	72	1813
37	2013	85	1898
38	2013	90	1988
39	2013	84	2072
40	2013	87	2159

Week	Year	Users	Cumulative
40	2013	87	2159
41	2013	73	2232
42	2013	99	2331
43	2013	89	2420
44	2013	96	2516
45	2013	91	2607
46	2013	88	2695
47	2013	102	2797
48	2013	97	2894
49	2013	116	3010
50	2013	124	3134
51	2013	102	3236
52	2013	47	3283

RESULTS FOR THE YEAR 2013

Case Study 2 : User Growth Analysis - Result Set(2)

Week	Year	Users	Cumulative
0	2014	83	3366
1	2014	126	3492
2	2014	109	3601
3	2014	113	3714
4	2014	130	3844
5	2014	133	3977
6	2014	135	4112
7	2014	125	4237
8	2014	129	4366
9	2014	133	4499
10	2014	154	4653
11	2014	130	4783
12	2014	148	4931
13	2014	167	5098
14	2014	162	5260
15	2014	164	5424
16	2014	179	5603
17	2014	170	5773
18	2014	163	5936
19	2014	185	6121

RESULTS FOR THE YEAR 2014

Result Grid Filter Rows:

Week	Year	Users	Cumulative
16	2014	179	5603
17	2014	170	5773
18	2014	163	5936
19	2014	185	6121
20	2014	176	6297
21	2014	183	6480
22	2014	196	6676
23	2014	196	6872
24	2014	229	7101
25	2014	207	7308
26	2014	201	7509
27	2014	222	7731
28	2014	215	7946
29	2014	221	8167
30	2014	238	8405
31	2014	193	8598
32	2014	245	8843
33	2014	261	9104
34	2014	259	9363
35	2014	18	9381

Case Study 2 : Weekly Retention Analysis

```
select t.LOGIN_WEEK,
SUM(CASE WHEN t.RETENTION = 0 THEN 1 ELSE 0 END) AS week_0,
SUM(CASE WHEN t.RETENTION = 1 THEN 1 ELSE 0 END) AS week_1,
SUM(CASE WHEN t.RETENTION = 2 THEN 1 ELSE 0 END) AS week_2,
SUM(CASE WHEN t.RETENTION = 3 THEN 1 ELSE 0 END) AS week_3,
SUM(CASE WHEN t.RETENTION = 4 THEN 1 ELSE 0 END) AS week_4,
SUM(CASE WHEN t.RETENTION = 5 THEN 1 ELSE 0 END) AS week_5,
SUM(CASE WHEN t.RETENTION = 6 THEN 1 ELSE 0 END) AS week_6,
SUM(CASE WHEN t.RETENTION = 7 THEN 1 ELSE 0 END) AS week_7,
SUM(CASE WHEN t.RETENTION = 8 THEN 1 ELSE 0 END) AS week_8,
SUM(CASE WHEN t.RETENTION = 9 THEN 1 ELSE 0 END) AS week_9
from
(select s.user_id AS User_ID, s.Signup_Week AS SIGNUP_WEEK, l.Login_Week AS
LOGIN_WEEK,
s.Signup_Week - l.Login_Week as RETENTION from
```

QUERY BREAKDOWN

COHORT FORMATION WITH THE RETENTION USERS

(Refer Next Slide for the Sub-Query)

Case Study 2 : Weekly Retention Analysis

```
select s.user_id AS User_ID, s.Signup_Week AS SIGNUP_WEEK,
l.Login_Week AS LOGIN_WEEK, s.Signup_Week - l.Login_Week as RETENTION
from (SELECT user_id,
week(occurred_at) AS Signup_Week
FROM events where event_name='complete_signup'
group by user_id, Signup_Week)s,
(SELECT user_id, min(week(occurred_at)) AS Login_Week
FROM events where event_name='login'
group by user_id
order by user_id)l
where s.user_id = l.user_id;
```

Users who have completed the sign_up process (week)

Users who have logged in after sign_up (First Login Week)

Case Study 2 : Weekly Retention Analysis - Result Set

Result Grid | Filter Rows: Search | Export:

LOGIN_WEEK	week_0	week_1	week_2	week_3	week_4	week_5	week_6	week_7	week_8	week_9
17	72	0	0	0	0	0	0	0	0	0
18	163	0	0	0	0	0	0	0	0	0
19	185	0	0	0	0	0	0	0	0	0
20	176	0	0	0	0	0	0	0	0	0
21	183	0	0	0	0	0	0	0	0	0
22	196	0	0	0	0	0	0	0	0	0
23	196	0	0	0	0	0	0	0	0	0
24	229	0	0	0	0	0	0	0	0	0
25	207	0	0	0	0	0	0	0	0	0
26	201	0	0	0	0	0	0	0	0	0
27	222	0	0	0	0	0	0	0	0	0
28	215	0	0	0	0	0	0	0	0	0
29	221	0	0	0	0	0	0	0	0	0
30	238	0	0	0	0	0	0	0	0	0
31	193	0	0	0	0	0	0	0	0	0
32	245	0	0	0	0	0	0	0	0	0
33	261	0	0	0	0	0	0	0	0	0
34	259	0	0	0	0	0	0	0	0	0
35	18	0	0	0	0	0	0	0	0	0

From the result-set. we can observe that, Users who have completed the sign_up process have Logged into the platform in the same week. There is no retention among the users who have completed both signup and login.

Note : There are users who have completed the signup process, but have not logged_in to the platform.

Case Study 2 : Weekly Engagement Per Device

```
select   Month(occurred_at) as Month,
         Year(occurred_at) as Year,
         device as Device_Type,
         count(distinct user_id) as Users
from events
group by device, Month(occurred_at),
Year(occurred_at);
```

NOTE : The result-set contains the device type and the list of active users respectively. This is sorted based on the month and year of the dataset.

THE RESULT-SET OF THE QUERY IS TOO LARGE.

ATTACHED SCREENSHOTS IN THE FOLLOWING SLIDES

PLEASE REFER NEXT 2 SLIDES

Case Study 2 : Weekly Engagement Per Device - Result Set(1)

Result Grid Filter Rows: Search

Month	Year	Device_Type	Users
5	2014	acer aspire desktop	61
6	2014	acer aspire desktop	69
7	2014	acer aspire desktop	100
8	2014	acer aspire desktop	87
5	2014	acer aspire notebook	108
6	2014	acer aspire notebook	118
7	2014	acer aspire notebook	137
8	2014	acer aspire notebook	160
5	2014	amazon fire phone	21
6	2014	amazon fire phone	31
7	2014	amazon fire phone	33
8	2014	amazon fire phone	38
5	2014	asus chromebook	107
6	2014	asus chromebook	127
7	2014	asus chromebook	153
8	2014	asus chromebook	150
5	2014	dell inspiron desktop	122
6	2014	dell inspiron desktop	138
7	2014	dell inspiron desktop	145
8	2014	dell inspiron desktop	145

Result Grid Filter Rows: Search

Month	Year	Device_Type	Users
5	2014	dell inspiron notebook	225
6	2014	dell inspiron notebook	263
7	2014	dell inspiron notebook	285
8	2014	dell inspiron notebook	290
5	2014	hp pavilion desktop	108
6	2014	hp pavilion desktop	132
7	2014	hp pavilion desktop	148
8	2014	hp pavilion desktop	131
5	2014	htc one	75
6	2014	htc one	67
7	2014	htc one	88
8	2014	htc one	50
5	2014	ipad air	171
6	2014	ipad air	164
7	2014	ipad air	187
8	2014	ipad air	148
5	2014	ipad mini	94
6	2014	ipad mini	97
7	2014	ipad mini	121
8	2014	ipad mini	91

Result Grid Filter Rows: Search

Month	Year	Device_Type	Users
5	2014	iphone 4s	142
6	2014	iphone 4s	143
7	2014	iphone 4s	187
8	2014	iphone 4s	143
5	2014	iphone 5	358
6	2014	iphone 5	393
7	2014	iphone 5	460
8	2014	iphone 5	336
5	2014	iphone 5s	221
6	2014	iphone 5s	210
7	2014	iphone 5s	278
8	2014	iphone 5s	204
5	2014	kindle fire	68
6	2014	kindle fire	70
7	2014	kindle fire	92
8	2014	kindle fire	48
5	2014	lenovo thinkpad	461
6	2014	lenovo thinkpad	480
7	2014	lenovo thinkpad	576
8	2014	lenovo thinkpad	562

Case Study 2 : Weekly Engagement Per Device - Result Set(2)

Result Grid | Filter Rows: Search

Month	Year	Device_Type	Users
5	2014	mac mini	54
6	2014	mac mini	59
7	2014	mac mini	63
8	2014	mac mini	76
5	2014	macbook air	321
6	2014	macbook air	365
7	2014	macbook air	428
8	2014	macbook air	375
5	2014	macbook pro	688
6	2014	macbook pro	700
7	2014	macbook pro	839
8	2014	macbook pro	837
5	2014	nexus 10	89
6	2014	nexus 10	99
7	2014	nexus 10	110
8	2014	nexus 10	86
5	2014	nexus 5	245
6	2014	nexus 5	233
7	2014	nexus 5	245
8	2014	nexus 5	202

Result Grid | Filter Rows: Search

Month	Year	Device_Type	Users
5	2014	nexus 7	101
6	2014	nexus 7	135
7	2014	nexus 7	149
8	2014	nexus 7	108
5	2014	nokia lumia 635	79
6	2014	nokia lumia 635	88
7	2014	nokia lumia 635	101
8	2014	nokia lumia 635	65
5	2014	samsumg galaxy ta…	31
6	2014	samsumg galaxy ta…	37
7	2014	samsumg galaxy ta…	43
8	2014	samsumg galaxy ta…	33
5	2014	samsung galaxy note	45
6	2014	samsung galaxy note	42
7	2014	samsung galaxy note	46
8	2014	samsung galaxy note	38
5	2014	samsung galaxy s4	257
6	2014	samsung galaxy s4	290
7	2014	samsung galaxy s4	346

100% 24:10

Result Grid | Filter Rows: Search

Month	Year	Device_Type	Users
5	2014	nokia lumia 635	79
6	2014	nokia lumia 635	88
7	2014	nokia lumia 635	101
8	2014	nokia lumia 635	65
5	2014	samsumg galaxy ta…	31
6	2014	samsumg galaxy ta…	37
7	2014	samsumg galaxy ta…	43
8	2014	samsumg galaxy ta…	33
5	2014	samsung galaxy note	45
6	2014	samsung galaxy note	42
7	2014	samsung galaxy note	46
8	2014	samsung galaxy note	38
5	2014	samsung galaxy s4	257
6	2014	samsung galaxy s4	290
7	2014	samsung galaxy s4	346
8	2014	samsung galaxy s4	265
5	2014	windows surface	56
6	2014	windows surface	55
7	2014	windows surface	85
8	2014	windows surface	53

Case Study 2 : Email Engagement Analysis

```
select  Week(occurred_at) as Week,
        action as Action,
        count(distinct user_id) as Users
from email_events
group by action, Week;
```

THE RESULT-SET OF THE QUERY IS LARGE. (Total 75 rows)

ATTACHED SCREENSHOTS IN THE FOLLOWING SLIDES

PLEASE REFER NEXT 2 SLIDES

NOTE : The result-set gives the total count of email events based on the type of event (action) triggered.
This is grouped based on the week number. (Weekly Analysis)

Case Study 2 : Email Engagement Metrics - Result Set (1)

Result Grid | Filter Rows: | Search

Week	Action	Users
17	email_clickthrough	166
18	email_clickthrough	425
19	email_clickthrough	476
20	email_clickthrough	501
21	email_clickthrough	436
22	email_clickthrough	478
23	email_clickthrough	529
24	email_clickthrough	549
25	email_clickthrough	524
26	email_clickthrough	550
27	email_clickthrough	613
28	email_clickthrough	594
29	email_clickthrough	583
30	email_clickthrough	625
31	email_clickthrough	444
32	email_clickthrough	416
33	email_clickthrough	490
34	email_clickthrough	481
35	email_clickthrough	38

Weekly Analysis report of Email Engagement Metrics for the Actions clickthrough and open

Result Grid | Filter Rows: | Search

Week	Action	Users
17	email_open	310
18	email_open	900
19	email_open	961
20	email_open	989
21	email_open	996
22	email_open	965
23	email_open	1057
24	email_open	1136
25	email_open	1084
26	email_open	1149
27	email_open	1207
28	email_open	1228
29	email_open	1201
30	email_open	1363
31	email_open	1338
32	email_open	1318
33	email_open	1417
34	email_open	1502
35	email_open	41

Case Study 2 : Email Engagement Metrics - Result Set (2)

Result Grid | Filter Rows: | Search

Week	Action	Users
17	sent_reengagement_email	73
18	sent_reengagement_email	157
19	sent_reengagement_email	173
20	sent_reengagement_email	191
21	sent_reengagement_email	164
22	sent_reengagement_email	192
23	sent_reengagement_email	197
24	sent_reengagement_email	226
25	sent_reengagement_email	196
26	sent_reengagement_email	219
27	sent_reengagement_email	213
28	sent_reengagement_email	213
29	sent_reengagement_email	213
30	sent_reengagement_email	231
31	sent_reengagement_email	222
32	sent_reengagement_email	200
33	sent_reengagement_email	264
34	sent_reengagement_email	261
35	sent_reengagement_email	48

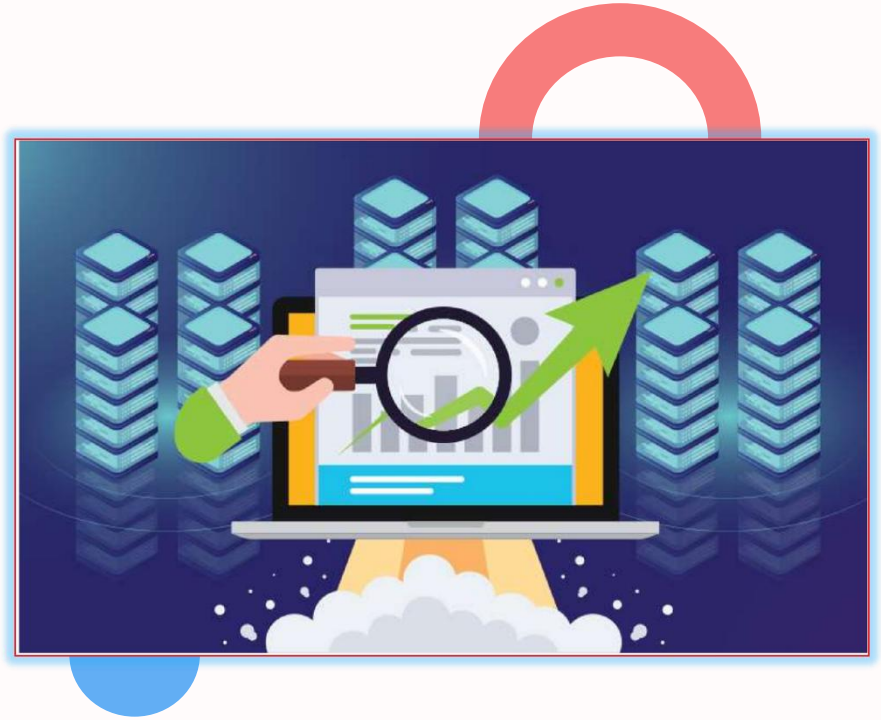
Weekly Analysis report of Email Engagement Metrics for the Actions

sent_reengagement

sent_weekly_digest

Result Grid | Filter Rows: | Search

Week	Action	Users
34	sent_reengagement_email	261
35	sent_reengagement_email	48
17	sent_weekly_digest	908
18	sent_weekly_digest	2602
19	sent_weekly_digest	2665
20	sent_weekly_digest	2733
21	sent_weekly_digest	2822
22	sent_weekly_digest	2911
23	sent_weekly_digest	3003
24	sent_weekly_digest	3105
25	sent_weekly_digest	3207
26	sent_weekly_digest	3302
27	sent_weekly_digest	3399
28	sent_weekly_digest	3499
29	sent_weekly_digest	3592
30	sent_weekly_digest	3706
31	sent_weekly_digest	3793
32	sent_weekly_digest	3897
33	sent_weekly_digest	4012
34	sent_weekly_digest	4111




thank you