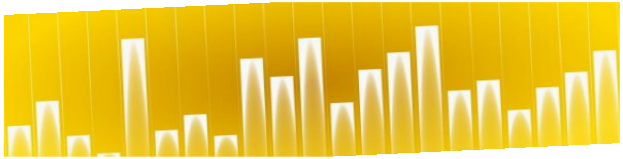
OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

Submitted By:- Shubham Saxena

AGENDA

- Project Description
- Steps Used
- Tech Stack Used
- Insights
- Result
- Drive Link

Project Description



In this project, Operational Analytics is investigating metric spikes, which 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, we'll need to answer these questions daily, making it crucial to understand how to investigate these metric spikes. In this project, we'll take on the role of a Lead Data Analyst at a company like Microsoft where we'll be provided with various datasets and tables, and our task will be to derive insights from this data to answer questions posed by different departments within the company.

Steps Used

1.

Create Data Sets:-

Create data sets linked with the project using MySQLite.

2.

Perform Analysis:-

Perform various tasks given in the project using tools MySQLite

3.

Create Presentation:-

Create presentation of the project by pasting screenshots of the tasks performed..

4.

Submission:-

Finally last step is to submit the project in PDF pr DOC format.

Tech Stack Used

MYSQL

Google Slides

Google Sheets

DATABASE CREATION
Project 3

Create Table

Job_data

Create Columns

Ds, job_id, actor_id, Event, Language, Time_spent, Org

Insert Values

As given in data sets

CREATION OF DATABASE PROJECT 3

Server Tools Scripting Help

QL File 12*

Limit to 1000 rows

```
Create Database Project3;
show databases;
use project3;
Create table job_data
(ds DATE,
job_id INT NOT NULL,
actor_id INT NOT NULL,
Event varchar(25) NOT NULL,
Language varchar(25) NOT NULL,
Time_spent INT NOT NULL,
Org varchar(2) NOT NULL
);
select * from job_data;
```

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

ds	job_id	actor_id	Event	Language	Time_spent	Org
2020-11-30	21	1001	skip	English	15	A
2020-11-30	22	1005	transfer	Arabic	25	B
2020-11-29	23	1003	decision	Persian	20	C
2020-11-28	23	1005	transfer	Persian	22	D
2020-11-28	25	1002	decision	Hindi	11	B
2020-11-27	11	1007	decision	French	104	D
2020-11-26	23	1004	skip	Persian	56	A
2020-11-25	20	1003	transfer	Italian	45	C

ob_data 6

Output

Create Table

Job_data

Create Columns

Ds, job_id, actor_id, Event, Language, Time_spent, Org

Insert Values

As given in data sets

Insights

➔ CASE STUDY 1 : Job Data Analysis

A. Jobs Reviewed Over Time:
- Objective: Calculate the number of jobs reviewed per hour for each day in November 2020.
- OurTask: Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.

B. Throughput Analysis:
- Objective: Calculate the 7-day rolling average of throughput (number of events per second).
- Our Task: Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

C. Language Share Analysis:
- Objective: Calculate the percentage share of each language in the last 30 days.
- Our Task: Write an SQL query to calculate the percentage share of each language over the last 30 days.

D. Duplicate Rows Detection:
- Objective: Identify duplicate rows in the data.
- Our Task: Write an SQL query to display duplicate rows from the job_data table.

CASE STUDY 1 : Job Data Analysis

Task

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

Task

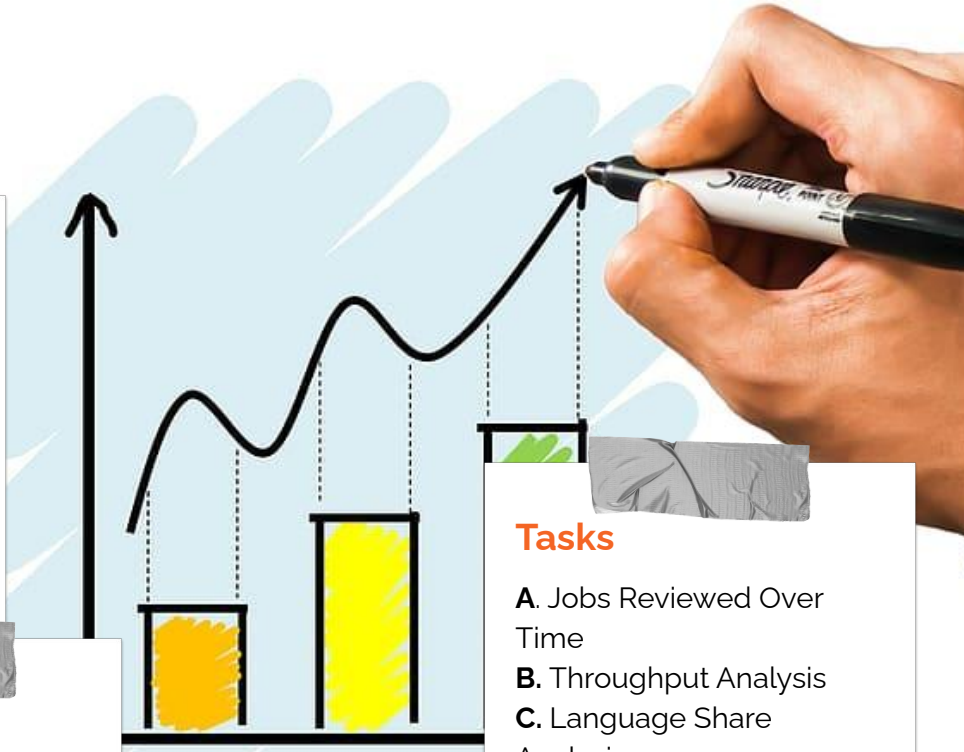
B. Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

Task

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

Task

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



Tasks

A. Jobs Reviewed Over Time
B. Throughput Analysis
C. Language Share Analysis
D. Duplicate Rows Detection:

Server Tools Scripting Help

roject 3 Job-Data* Task 1 - calculate the number of...

Limit to 1000 rows

```
/*TASK 1 - Write an SQL query to calculate the number of jobs reviewed per hour for each d
Use project3;
select avg(t) as 'Average jobs reviewed per day per hour',
avg(p) as 'Average 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;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

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

esult 2

utput

JOBS REVIEWED OVER TIME

Tasks

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

ANS# Average jobs reviewed per day per hour, Average jobs reviewed per day per second

'126.18048333', '0.03505000'

THROUGHPUT ANALYSIS

Task

B. Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

ANS: **# Daily Throughput**

'0.02','0.02','0.01','0.06','0.05','0.05'

Regarding the question about whether to prefer the daily metric or the 7-day rolling average for throughput, it depends on the specific use case and the nature of the data. Here are some considerations:

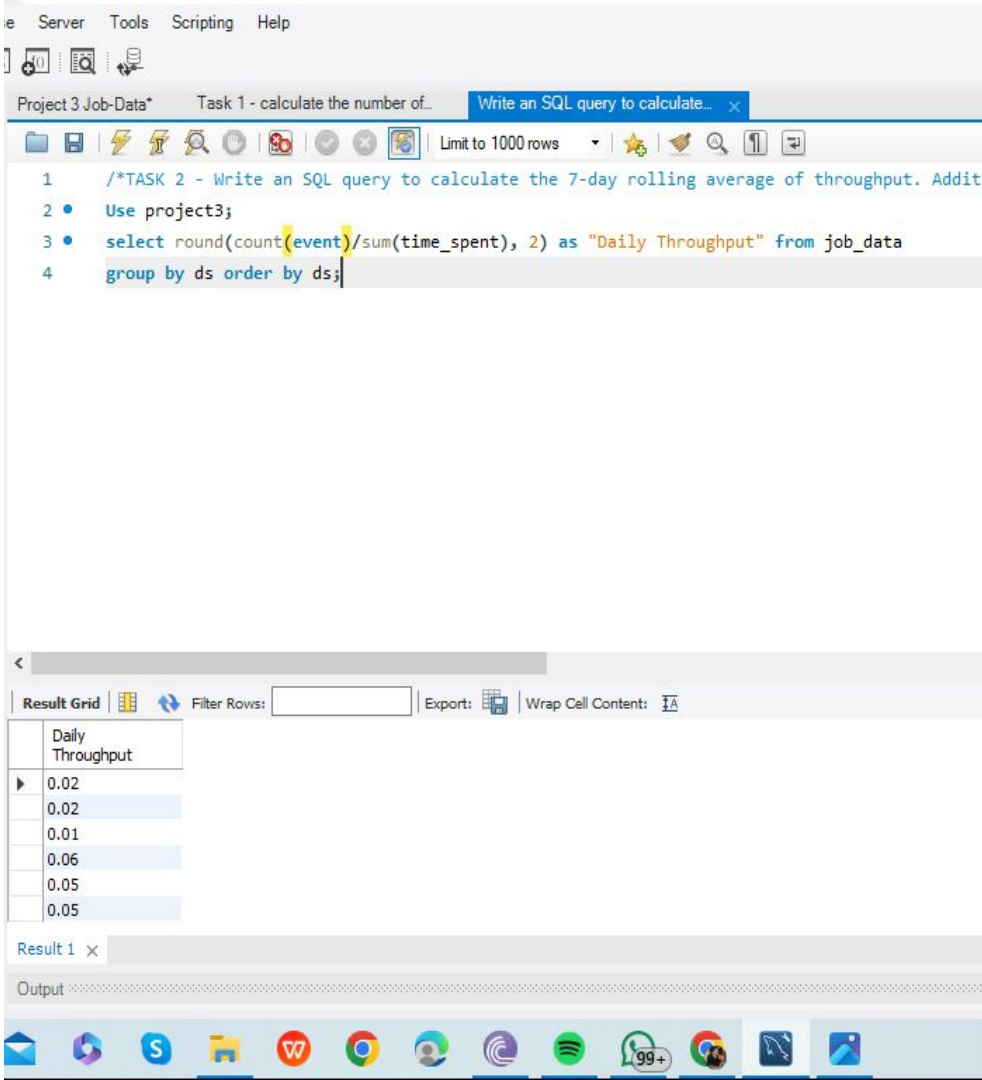
Daily Metric:

- Provides a snapshot of throughput for each individual day.
- Can be sensitive to daily fluctuations or anomalies.
- Useful for analyzing short-term trends.

7-Day Rolling Average:

- Smooths out short-term fluctuations, offering a more stable and representative trend.
- Helps identify long-term patterns and trends.
- Particularly useful for detecting underlying patterns while minimizing the impact of daily variations.

Hence, mine choice would be 7-Day Rolling Average as it is more stable and smooth and helps in identifying long term patterns and trends.



Server Tools Scripting Help

oject 3 Job-Data* | Task 1 - calculate the number of... | Write an SQL query to calculate... | Write an SQL query to c

Limit to 1000 rows

```sql
    language AS Languages,
    ROUND(100 * COUNT(*) / sub.total, 2) AS Percentage,
    sub.total AS TotalEvents
  FROM
    job_data
  CROSS JOIN (
    SELECT COUNT(*) AS total
    FROM job_data
  ) AS sub
  GROUP BY
    language, sub.total;
```

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

Languages	Percentage	TotalEvents
English	12.50	8
Arabic	12.50	8
Persian	37.50	8
Hindi	12.50	8
French	12.50	8

sult 2

tput

Action Output

#	Time	Action	Message
36	00:52:06	Use project3	0 row(s) affected
37	00:52:06	SELECT language AS Languages, ROUND(100 * COUNT(*) / sub.total, 2) AS Percentag...	6 row(s) returned

LANGUAGE SHARE ANALYSIS

Tasks

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

ANS# Languages, Percentage, TotalEvents
'English', '12.50', '8'
'Arabic', '12.50', '8'
'Persian', '37.50', '8'
'Hindi', '12.50', '8'
'French', '12.50', '8'

DUPLICATE ROW DETECTION

Task

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

ANS: **# actor_id, Duplicates**

'1005', '2'

'1003', '2'

Server Tools Scripting Help

Task 1 - calculate the number of... | Write an SQL query to calculate... | Write an SQL query to calculate...

Limit to 1000 rows

```
/*TASK 2 -  Write an SQL query to calculate the percentage share of each language over
Use project3;
SELECT
actor_id, count(*) as Duplicates from job_data
group by actor_id having count(*)>1;
```

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

actor_id	Duplicates
1005	2
1003	2

Result 1

Output

Action Output

#	Time	Action	Message
38	00:57:46	Use project3	0 row(s) affected
39	00:57:46	SELECT actor_id, count(*) as Duplicates from job_data group by actor_id having count(*)>1 ...	2 row(s) returned

CASE STUDY 2
Uploading of tables

1. Users
2. Events
3. Email_Events

Tip

Here we uploaded the table one by one manually as per the instructions.

Creation of User Table

Server Tools Scripting Help

lculate the number of... | Write an SQL query to calculate... | Write an SQL query to calculate... | Write an SQL query to displ

Limit to 1000 rows

```
13 • SHOW VARIABLES LIKE 'secure_file_priv';
14
15   -- Load data from CSV file into users table
16 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users.csv'
17   INTO TABLE users
```

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

user_id	company_id	language	activated_at	state	created_at
0	5737	english	01-01-2013 21:01	active	NULL
3	2800	german	01-01-2013 18:42	active	2013-01-01 18:40:00
4	5110	indian	01-01-2013 14:39	active	2013-01-01 14:37:00
6	11699	english	01-01-2013 18:38	active	2013-01-01 18:37:00
7	4765	french	01-01-2013 16:20	active	2013-01-01 16:19:00
8	2698	french	01-01-2013 04:40	active	2013-01-01 04:38:00
11	3745	english	01-01-2013 08:09	active	2013-01-01 08:07:00
13	4025	english	02-01-2013 12:29	active	2013-01-02 12:27:00
15	4259	english	02-01-2013 15:41	active	2013-01-02 15:39:00
17	5025	japanese	02-01-2013 10:57	active	2013-01-02 10:56:00
19	326	english	02-01-2013 09:55	active	2013-01-02 09:54:00
20	7	italian	02-01-2013 09:43	active	2013-01-02 09:41:00
21	2606	english	02-01-2013 09:30	active	2013-01-02 09:29:00
22	545	german	02-01-2013 17:38	active	2013-01-02 17:36:00
27	6	japanese	03-01-2013 16:15	active	2013-01-03 16:14:00
30	4148	english	03-01-2013 08:29	active	2013-01-03 08:28:00
31	39	arabic	03-01-2013 15:46	active	2013-01-03 15:45:00
33	10768	english	03-01-2013 12:18	active	2013-01-03 12:16:00
35	1891	english	03-01-2013 16:07	active	2013-01-03 16:06:00
36	2	english	03-01-2013 11:53	active	2013-01-03 11:51:00

sers 16

Output

*Untitled - Notepad

File Edit Format View Help

```
USE Project3;

CREATE TABLE users (
    user_id INT,
    created_at VARCHAR(100),
    company_id INT,
    language VARCHAR(65),
    activated_at VARCHAR(150),
    state VARCHAR(50)
);

-- Check secure_file_priv variable
SHOW VARIABLES LIKE 'secure_file_priv';

-- Load data from CSV file into users table
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;

-- Display the data in the users table
SELECT * FROM users;

-- Alter table to add temp_created_at column with datetime datatype
ALTER TABLE users ADD COLUMN temp_created_at DATETIME;

-- as it is shwoing error 1175 of using safe update mode for that use syntax
Set SQL_safe_updates=0;

-- Update temp_created_at column with correct datetime values
UPDATE users SET temp_created_at = STR_TO_DATE(created_at, '%d-%m-%Y %H:%i')WHERE
alter table users drop column created_at;
ALTER TABLE users CHANGE COLUMN temp_created_at created_at DATETIME;
```

Ln 26, Col 45 | 100% | Windows (CRLF) | UTF-8

Creation of Event Table

Server Tools Scripting Help

Write an SQL query to calculate... | Write an SQL query to calculate... | Write an SQL query to display d... | create user t

Limit to 1000 rows

```
6    event_type varchar(55),
7    event_name VARCHAR(65),
8    location VARCHAR(150),
9    device VARCHAR(75),
10   user_type int
```

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

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
10736	09-05-2014 17:53	engagement	like_message	Austria	iphone 4s	2

Result 1 | events 2

Output

*Untitled - Notepad

File Edit Format View Help

```
USE Project3;

CREATE TABLE events (
    user_id INT,
    occurred_at VARCHAR(100),
    event_type varchar(55),
    event_name VARCHAR(65),
    location VARCHAR(150),
    device VARCHAR(75),
    user_type int
);

-- Check secure_file_priv variable
SHOW VARIABLES LIKE 'secure_file_priv';

-- Load data from CSV file into event table
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;

-- Display the data in the event table
SELECT * FROM events;

-- Alter table to add temp_created_at column with datetime datatype
ALTER TABLE events ADD COLUMN temp_occurred_at DATETIME;

-- as it is shwoing error 1175 of using safe update mode for that use syntax
Set SQL_safe_updates=0;

-- Update temp_created_at column with correct datetime values
UPDATE events SET temp_occurred_at = STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i')WHERE user_id > 0;
alter table events drop column occurred_at;
ALTER TABLE events CHANGE COLUMN temp_occurred_at occurred_at DATETIME;
```

Creation of Email_Event Table

Server Tools Scripting Help

Write an SQL query to calculate... | Write an SQL query to display d... | create user table* | Create Event Table*

Limit to 1000 rows

```
11 • SHOW VARIABLES LIKE 'secure_file_priv';
12
13   -- Load data from CSV file into email_events table
14 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/email_events.csv'
15   INTO TABLE events
```

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

user_id	action	user_type	occurred_at
4	sent_weekly_digest	3	2014-06-0...
4	sent_weekly_digest	3	2014-06-1...
4	sent_weekly_digest	3	2014-06-1...
4	sent_weekly_digest	3	2014-06-2...
4	email_open	3	2014-06-2...
4	email_clickthrough	3	2014-06-2...
4	sent_weekly_digest	3	2014-07-0...
4	email_open	3	2014-07-0...
4	email_clickthrough	3	2014-07-0...
4	sent_weekly_digest	3	2014-07-0...
4	sent_weekly_digest	3	2014-07-1...
4	sent_weekly_digest	3	2014-07-2...
4	sent_weekly_digest	3	2014-07-2...
4	sent_weekly_digest	3	2014-08-0...
4	sent_weekly_digest	3	2014-08-1...
4	email_open	3	2014-08-1...
4	sent_weekly_digest	3	2014-08-1...
4	sent_weekly_digest	3	2014-08-2...
8	sent_weekly_digest	3	2014-05-0...
8	sent_weekly_digest	3	2014-05-1...

mail_events 7

utput

titled - Notepad

lit Format View Help

```
oject3;

 TABLE email_events (
er_id INT,
curred_at VARCHAR(100),
tion varchar(55),
er_type int


ck secure_file_priv variable
ARIABLES LIKE 'secure_file_priv';

d data from CSV file into email_events table
ATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/email_events.csv'
ABLE events
 TERMINATED BY ','
ED BY '"'
TERMINATED BY '\n'
 1 ROWS;

play the data in the email_events table
 * FROM email_events;

er table to add temp_occurred_at column with datetime datatype
TABLE email_events ADD COLUMN temp_occurred_at DATETIME;

it is shwoing error 1175 of using safe update mode for that use syntax
L_safe_updates=0;

late temp_occurred_at column with correct datetime values
 email_events SET temp_occurred_at = STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i')WHERE user_id >
table email_events drop column occurred_at;
TABLE email_events CHANGE COLUMN temp_occurred_at occurred_at DATETIME;
```

Ln 33, Col 78 | 100% | Windows (CRLF) | UTF-8

Insights

CASE STUDY 2 : Investigating Metric Spike

A. Weekly User Engagement::
- Objective: Measure the activeness of users on a weekly basis.
- Your Task: Write an SQL query to calculate the weekly user engagement.

B. User Growth Analysis:
- Objective: Analyze the growth of users over time for a product.
- Your Task: Write an SQL query to calculate the user growth for the product.

C. Weekly Retention Analysis:
- Objective: Analyze the retention of users on a weekly basis after signing up for a product.
- Your Task: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

D. Weekly Engagement Per Device:
- Objective: Measure the activeness of users on a weekly basis per device.
- Your Task: Write an SQL query to calculate the weekly engagement per device.

E. Email Engagement Analysis:
- Objective: Analyze how users are engaging with the email service.
- Your Task: Write an SQL query to calculate the email engagement metrics.

Case Study 2
Investigating Metric Spike

TASK 1

Write an SQL query to calculate the weekly user engagement

TASK 2

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

Task 3

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

Case Study 2
Investigating Metric Spike

TASK 4

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

Task 5

Write an SQL query to calculate the email engagement metrics.

WEEKLY USER ENGAGEMENT

Server Tools Scripting Help

calculate... | Write an SQL query to display d... | create user table* | Create Event Table* | Create Email_event Table*

Limit to 1000 rows

```sql
Use Project3;
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;
```

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

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

Result 1

Output

Task

A. Write an SQL query to calculate the weekly user engagement.

ANS: # 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'

***NOTE: HIGHEST USER WEEK 30 WITH 1467 ACTIVE USERS**
MINIMUM USER WEEK 35 WITH 104 ACTIVE USERS

USER GROWTH ANALYSIS

Task

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

ANS: HIGHEST USER GROWTH WAS 261 IN 2014 IN 33RD WEEK

LOWEST USER GROWTH WAS 18 IN 2014 IN 35TH WEEK

Server Tools Scripting Help

to display d... | create user table* | Create Event Table* | Create Email_event Table* | Weekly User Engagment*

Limit to 1000 rows

```sql
USE Project3;
SELECT year, week_num, num_users,
    SUM(num_users) OVER (PARTITION BY year ORDER BY year, week_num) as cum_users
FROM(
    SELECT EXTRACT(YEAR FROM created_at) AS year, EXTRACT(WEEK FROM created_at) AS week
    FROM users
    WHERE state = 'active'
    GROUP BY year, week_num
) AS subquery;
```

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

year	week_num	num_users	cum_users
2013	11	31	434
2013	12	33	467
2013	13	39	506
2013	14	35	541
2013	15	43	584
2013	16	46	630
2013	17	49	679
2013	18	44	723
2013	19	57	780
2013	20	39	819
2013	21	49	868
2013	22	54	922
2013	23	50	972

ult 1

put

Action Output

WEEKLY RETENTION ANALYSIS

Server Tools Scripting Help

Create Event Table* | Create Email_event Table* | Weekly User Engagment* | user GROWTH ANALYSIS* | WEEKLY RE

Limit to 1000 rows

```
USE Project3;

WITH cte1 AS (
    SELECT DISTINCT
        user_id,
        EXTRACT(WEEK FROM occurred_at) AS signup_week
    FROM
        events
    WHERE
        event_type = 'signup-flow'  -- Corrected column name from events_type to event_ty
        AND event_name = 'complete_signup'
        AND EXTRACT(WEEK FROM occurred_at) = 10
),
cte2 AS (
    SELECT DISTINCT
        user_id,
        EXTRACT(WEEK FROM occurred_at) AS engagement_week
```

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

total_engaged_users	retained_users
317	236

Result 1

Output

Task

A. Write an SQL query to calculate the weekly user engagement.
ANS:# total_engaged_users, retained_users
'317', 236

WEEKLY ENGAGEMENT PER DEVICES

Task

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

ANS: HIGHEST WEEKLY ENGAGEMENT WAS 1753 BY MACBOOK PRO

LOWEST WEEKLY ENGAGEMENT WAS 80 BY AMAZON FIRE PHONE

 # year, week, device, weekly_engagement
NULL, NULL, 'amazon fire phone', '80'
NULL, NULL, 'samsumg galaxy tablet', '90'
NULL, NULL, 'samsung galaxy note', '103'
NULL, NULL, 'mac mini', '137'
NULL, NULL, 'windows surface', '154'
NULL, NULL, 'htc one', '168'
NULL, NULL, 'acer aspire desktop', '176'
NULL, NULL, 'kindle fire', '179'
NULL, NULL, 'nokia lumia 635', '190'
NULL, NULL, 'ipad mini', '243'
NULL, NULL, 'nexus 10', '249'
NULL, NULL, 'hp pavilion desktop', '298'
NULL, NULL, 'nexus 7', '305'
NULL, NULL, 'acer aspire notebook', '310'
NULL, NULL, 'asus chromebook', '315'
NULL, NULL, 'dell inspiron desktop', '326'
NULL, NULL, 'iphone 4s', '353'
NULL, NULL, 'ipad air', '422'
NULL, NULL, 'nexus 5', '546'
NULL, NULL, 'iphone 5s', '553'
NULL, NULL, 'dell inspiron notebook', '603'
NULL, NULL, 'samsung galaxy s4', '690'
NULL, NULL, 'macbook air', '869'
NULL, NULL, 'iphone 5', '907'
NULL, NULL, 'lenovo thinkpad', '1163'
NULL, NULL, 'macbook pro', '1753'

Server Tools Scripting Help

Weekly User Engagment* user GROWTH ANALYSIS* WEEKLY RETENTION ANALYS... Weekly Engageme

Limit to 1000 rows

```sql
USE Project3;
SELECT
    EXTRACT(YEAR FROM occurred_at) AS year,
    EXTRACT(WEEK FROM occurred_at) AS week,
    device,
    COUNT(DISTINCT user_id) AS weekly_engagement
FROM
    events
WHERE
    event_type = 'engagement'
GROUP BY
    year, week, device
ORDER BY
    year, week, device;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

year	week	device	weekly_engagement
NULL	NULL	macbook pro	1753
NULL	NULL	lenovo thinkpad	1163
NULL	NULL	iphone 5	907
NULL	NULL	macbook air	869
NULL	NULL	samsung galaxy s4	690
NULL	NULL	dell inspiron notebook	603
NULL	NULL	iphone 5s	553
NULL	NULL	nexus 5	546
NULL	NULL	ipad air	422

Result 5

Output

EMAIL ENGAGEMENT ANALYSIS

Server Tools Scripting Help

Weekly User Engagment* | user GROWTH ANALYSIS* | WEEKLY RETENTION ANALYS... | Weekly Engagement

Limit to 1000 rows

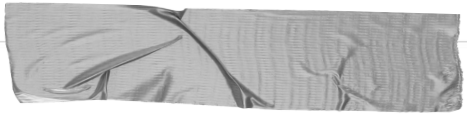
```sql
USE Project3;
SELECT * FROM email_events;
SELECT ACTION FROM email_events GROUP BY ACTION;
SELECT DISTINCT ACTION FROM email_events;
SELECT COUNT(*) AS COUNT, ACTION FROM email_events GROUP BY ACTION;
SELECT
    100.0 * SUM(CASE WHEN email_category = 'email_open' THEN 1 ELSE 0 END) / SUM(CASE WH
    100.0 * SUM(CASE WHEN email_category = 'email_click' THEN 1 ELSE 0 END) / SUM(CASE W
FROM (
    SELECT *,
        CASE
            WHEN action IN ('sent_weekly_digest', 'sent_reengagement_email') THEN 'email
            WHEN action IN ('email_open') THEN 'email_open'
            WHEN action IN ('email_clickthrough') THEN 'email_click'
        END AS email_category
    FROM email_events
) AS a;
```

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

open_rate	click_rate
33.58339	14.78989

email_events 10 | email_events 11 | email_events 12 | Result 13 | Result 14

Output



Task

E. Write an SQL query to calculate the email engagement metrics.

ANS: # open_rate, click_rate

'33.58339', '14.78989'

Thank You!