User Activity Analysis Using SQL

**Table of Contents**

1. [Project Overview](#project-overview)
2. [Data Schema](#data-schema)
3. [Technologies Used](#technologies-used)
4. [Data Analysis Techniques](#data-analysis-techniques)
5. [Business Questions and Answers](#business-questions-and-answers)
6. [Usage Instructions](#usage-instructions)
7. Examples
8. [Acknowledgements](#acknowledgements)
9. [Contact Information](#contact-information)

**Project Overview**

This project focuses on analysing user activity data from two tables, users and logins. The goal is to provide valuable insights into user engagement, activity patterns, and overall usage trends over time. Understanding user behaviour helps in improving user experience and guiding data-driven decisions.

**Data Schema**

Users Table

USER\_ID INT PRIMARY KEY

USER\_NAME VARCHAR(20) NOT NULL

USER\_STATUS VARCHAR(20) NOT NULL

Logins Table

USER\_ID INT

LOGIN\_TIMESTAMP DATETIME NOT NULL

SESSION\_ID INT PRIMARY KEY

SESSION\_SCORE INT

FOREIGN KEY (USER\_ID) REFERENCES USERS(USER\_ID)

**Technologies Used**

* SQL
* Database Management System (MySQL)

**Data Analysis Techniques**

The project employs SQL queries to address various business questions, using techniques such as:

* Subqueries for filtering and comparison.
* Window Functions for trend analysis.
* Common Table Expressions (CTEs) for recursive queries and generating date ranges.

**Business Questions and Answers**

1. **Which users did not log in during the past 5 months?**

select user\_id,max(login\_timestamp) as last\_login

from logins

group by user\_id

having max(login\_timestamp) < date\_sub(curdate(), interval 5 month)

2. **How many users and sessions were there in each quarter, ordered from newest to oldest?**

select quarter(login\_timestamp) as quarter\_no,

date\_format(min(login\_timestamp), '%Y-%m-01') as first\_date\_of\_quarter,

min(date(login\_timestamp)) as first\_date,

count(\*) as session\_count,count(distinct user\_id) as user\_count

from logins

group by quarter(login\_timestamp)

3. **Which users logged in during January 2024 but did not log in during November 2023?**

select distinct user\_id from logins

where date(login\_timestamp) between '2024-01-01' and '2024-01-31'

except

select distinct user\_id from logins

where date(login\_timestamp) between '2023-11-01' and '2023-11-30'

4. **What is the percentage change in sessions from the last quarter?**

with previous\_session as

(

select quarter(login\_timestamp) as quarter\_num,

count(\*) as session\_count ,

lag(count(\*)) over (order by quarter(login\_timestamp)) as previous\_session\_count

from logins

group by quarter(login\_timestamp)

)

select \*,round(((session\_count- previous\_session\_count)/(session\_count)\*100),2) as percentage\_change

from previous\_session

5. **Which user had the highest session score each day?**

select date,user\_name,highest\_session\_score from

(select u.user\_id,user\_name,session\_score as highest\_session\_score,date(login\_timestamp) as date,

row\_number() over (partition by date(login\_timestamp) order by session\_score desc) as row\_num

from users u inner join logins l on u.user\_id = l. user\_id)T

where row\_num=1

6. **Which users have had a session every single day since their first login?**

select user\_id, min(date(login\_timestamp)) as first\_login,count(distinct date(login\_timestamp)) as no\_of\_logins\_made,

datediff(curdate(),min(date(login\_timestamp))) as no\_of\_logins\_required

from logins

group by user\_id

having count(distinct date(login\_timestamp)) = datediff(curdate(),min(date(login\_timestamp)))

7. **On what dates were there no logins at all?**

with recursive cte as

(

select min(date(login\_timestamp)) as first\_date, max(date(login\_timestamp)) as last\_date

from logins

union all

select date\_add(first\_date, interval 1 day), last\_date from cte

where first\_date < last\_date

)

select \* from cte

where first\_date not in (select distinct date(login\_timestamp) from logins)

**Usage Instructions**

* Clone the repository from GitHub.
* Set up your database and run the queries in the setup file to import the provided schema.
* Execute the SQL queries provided in the queries.sql file to obtain the analysis results.

**Examples**

Refer to the example file for sample outputs that illustrate the results of the analysis.

**Acknowledgements**

I would like to extend my sincere gratitude to Ankit Bansal for their invaluable contributions to this project. Their guidance, support, and insights have been instrumental in achieving the project objectives.

**Contact Information**

For questions or feedback, contact Harini at <harinijaya2000@gmail.com>