**User Activity Analysis Using SQL**

**Script:**

*CREATE TABLE users (*

*USER\_ID INT PRIMARY KEY,*

*USER\_NAME VARCHAR(20) NOT NULL,*

*USER\_STATUS VARCHAR(20) NOT NULL*

*);*

*CREATE TABLE logins (*

*USER\_ID INT,*

*LOGIN\_TIMESTAMP DATETIME NOT NULL,*

*SESSION\_ID INT PRIMARY KEY,*

*SESSION\_SCORE INT,*

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

*);*

*-- Users Table*

*INSERT INTO USERS VALUES (1, 'Alice', 'Active');*

*INSERT INTO USERS VALUES (2, 'Bob', 'Inactive');*

*INSERT INTO USERS VALUES (3, 'Charlie', 'Active');*

*INSERT INTO USERS VALUES (4, 'David', 'Active');*

*INSERT INTO USERS VALUES (5, 'Eve', 'Inactive');*

*INSERT INTO USERS VALUES (6, 'Frank', 'Active');*

*INSERT INTO USERS VALUES (7, 'Grace', 'Inactive');*

*INSERT INTO USERS VALUES (8, 'Heidi', 'Active');*

*INSERT INTO USERS VALUES (9, 'Ivan', 'Inactive');*

*INSERT INTO USERS VALUES (10, 'Judy', 'Active');*

*-- Logins Table*

*INSERT INTO LOGINS VALUES (1, '2023-07-15 09:30:00', 1001, 85);*

*INSERT INTO LOGINS VALUES (2, '2023-07-22 10:00:00', 1002, 90);*

*INSERT INTO LOGINS VALUES (3, '2023-08-10 11:15:00', 1003, 75);*

*INSERT INTO LOGINS VALUES (4, '2023-08-20 14:00:00', 1004, 88);*

*INSERT INTO LOGINS VALUES (5, '2023-09-05 16:45:00', 1005, 82);*

*INSERT INTO LOGINS VALUES (6, '2023-10-12 08:30:00', 1006, 77);*

*INSERT INTO LOGINS VALUES (7, '2023-11-18 09:00:00', 1007, 81);*

*INSERT INTO LOGINS VALUES (8, '2023-12-01 10:30:00', 1008, 84);*

*INSERT INTO LOGINS VALUES (9, '2023-12-15 13:15:00', 1009, 79);*

*-- 2024 Q1*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (1, '2024-01-10 07:45:00', 1011, 86);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (2, '2024-01-25 09:30:00', 1012, 89);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (3, '2024-02-05 11:00:00', 1013, 78);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (4, '2024-03-01 14:30:00', 1014, 91);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (5, '2024-03-15 16:00:00', 1015, 83);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (6, '2024-04-12 08:00:00', 1016, 80);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (7, '2024-05-18 09:15:00', 1017, 82);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (8, '2024-05-28 10:45:00', 1018, 87);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (9, '2024-06-15 13:30:00', 1019, 76);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (10, '2024-06-25 15:00:00', 1010, 92);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (10, '2024-06-26 15:45:00', 1020, 93);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (10, '2024-06-27 15:00:00', 1021, 92);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (10, '2024-06-28 15:45:00', 1022, 93);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (1, '2024-01-10 07:45:00', 1101, 86);*

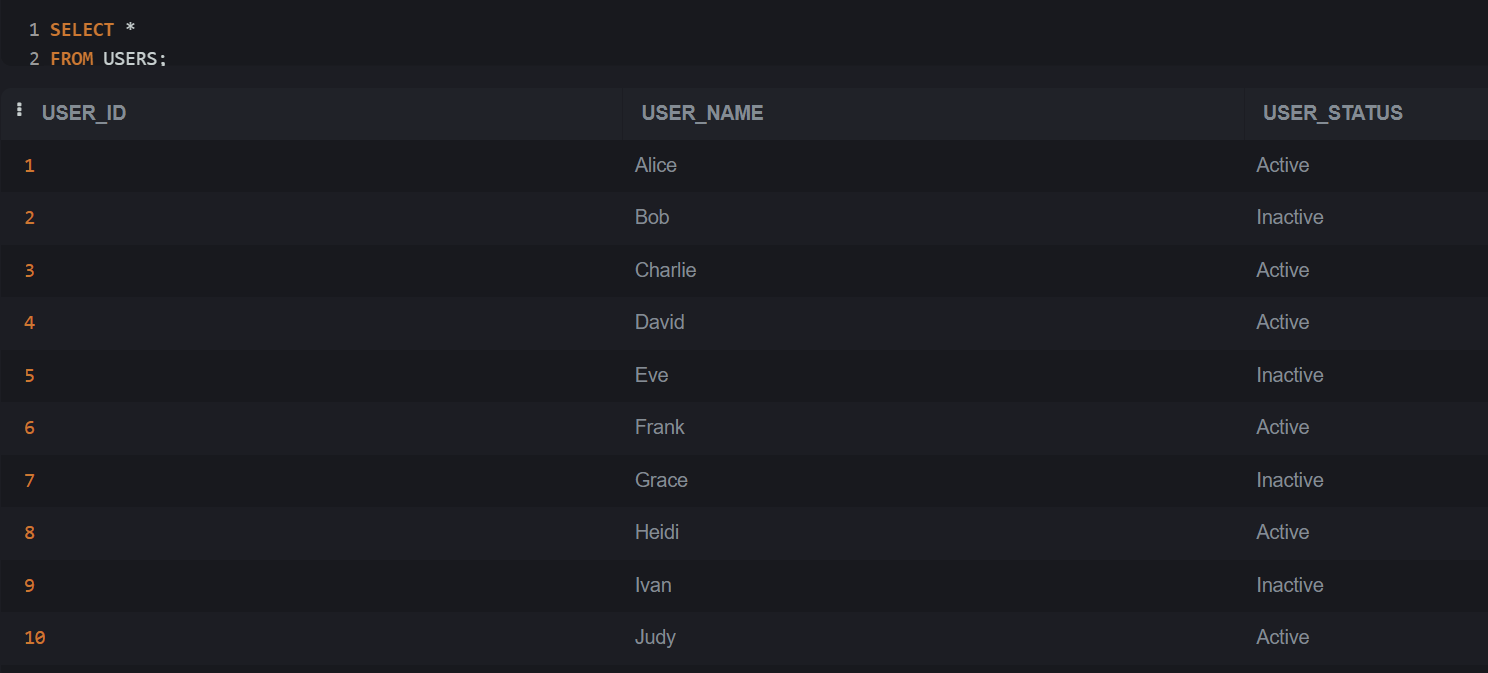
*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (3, '2024-01-25 09:30:00', 1102, 89);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (5, '2024-01-15 11:00:00', 1103, 78);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (2, '2023-11-10 07:45:00', 1201, 82);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (4, '2023-11-25 09:30:00', 1202, 84);*

*INSERT INTO LOGINS (USER\_ID, LOGIN\_TIMESTAMP, SESSION\_ID, SESSION\_SCORE) VALUES (6, '2023-11-15 11:00:00', 1203, 80);*





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

Today’s date – 2024-AUGUST-04

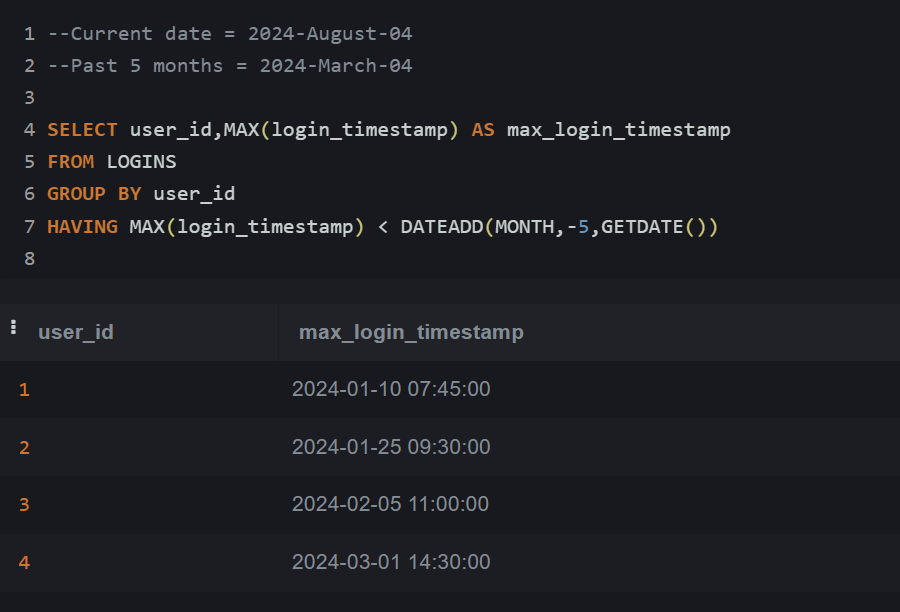
Past 5 months date – 2024-MAR-04

*SELECT user\_id,MAX(login\_timestamp)*

*FROM LOGINS*

*GROUP BY user\_id*

*HAVING MAX(login\_timestamp) < DATEADD(MONTH,-5,GETDATE());*



2ND Method:

*SELECT DISTINCT user\_id, login\_timestamp*

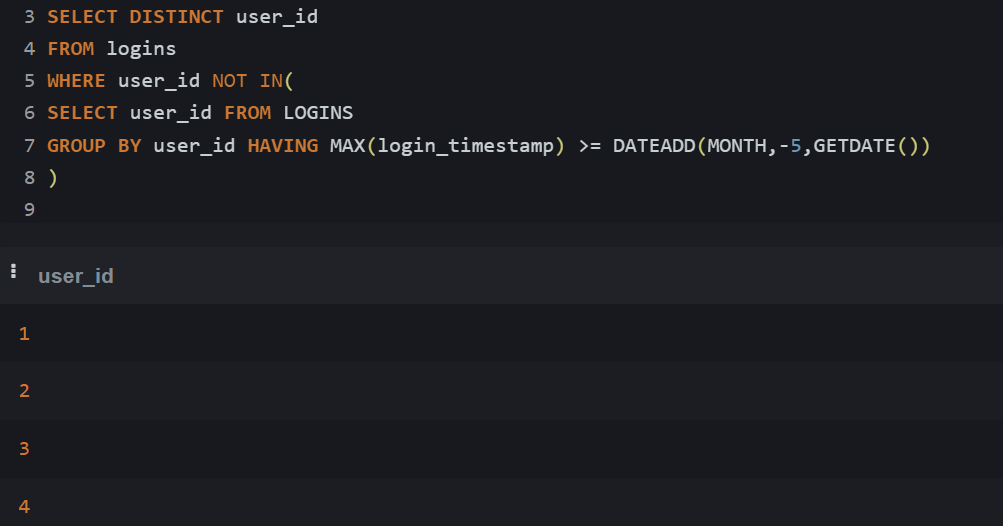
*FROM logins*

*WHERE user\_id NOT IN(*

*SELECT user\_id FROM LOGINS*

*GROUP BY user\_id HAVING MAX(login\_timestamp) >= DATEADD(MONTH,-5,GETDATE())*

*)*



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

Return first day of the quarter, user count, session count

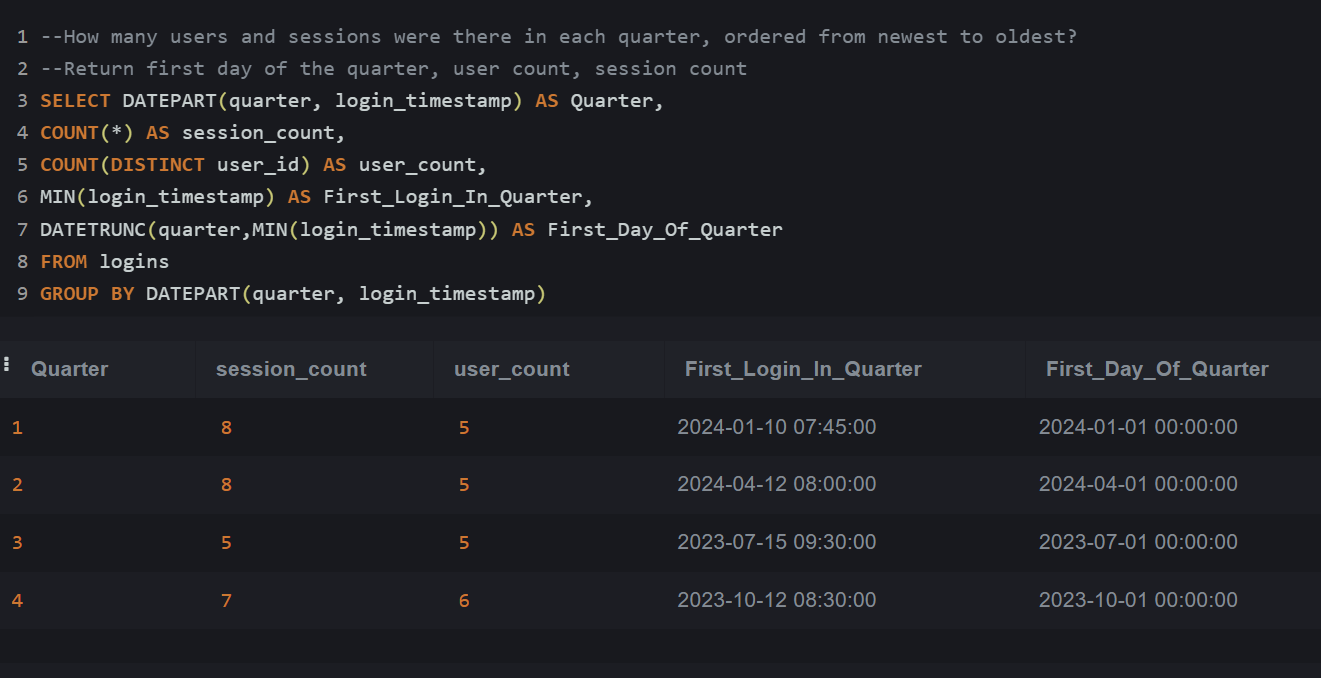
*SELECT*

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

*DATETRUNC(quarter,MIN(login\_timestamp)) AS First\_Day\_Of\_Quarter*

*FROM logins*

*GROUP BY DATEPART(quarter, login\_timestamp);*



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

*SELECT DISTINCT user\_id FROM logins*

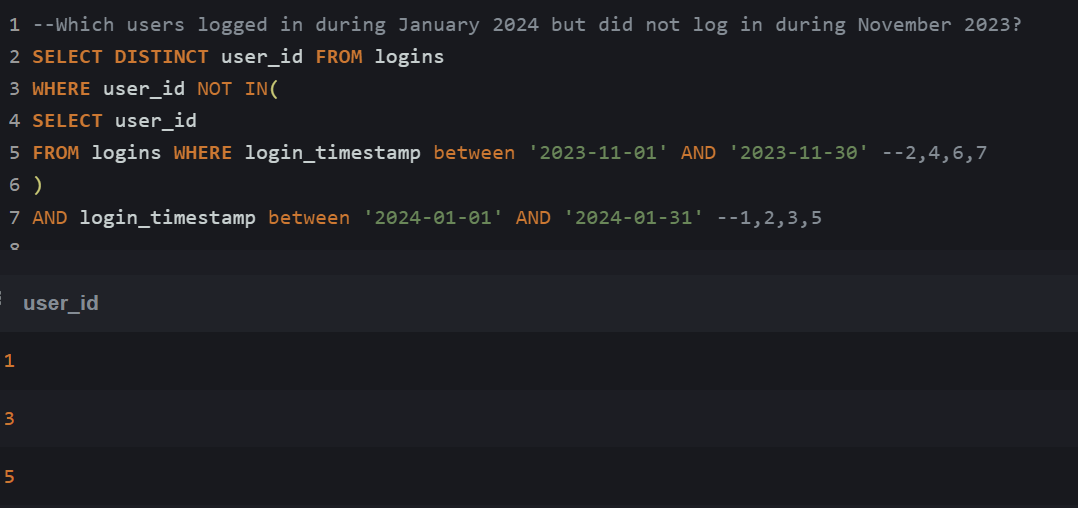
*WHERE user\_id NOT IN(*

*SELECT user\_id*

*from logins WHERE login\_timestamp between '2023-11-01' AND '2023-11-30' --2,4,6,7*

*)*

*AND login\_timestamp between '2024-01-01' AND '2024-01-31' --1,2,3,5;*



***Que4) What is the percentage change in sessions from the last quarter? Add to question 2 above***

--Return first day of the quarter, user count, session count, previous session count, percentage change

*with first\_cte AS(*

*SELECT DATETRUNC(quarter,MIN(login\_timestamp)) AS First\_Day\_Of\_Quarter,*

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

*COUNT(\*) AS session\_count*

*FROM logins*

*GROUP BY DATEPART(quarter, login\_timestamp) )*

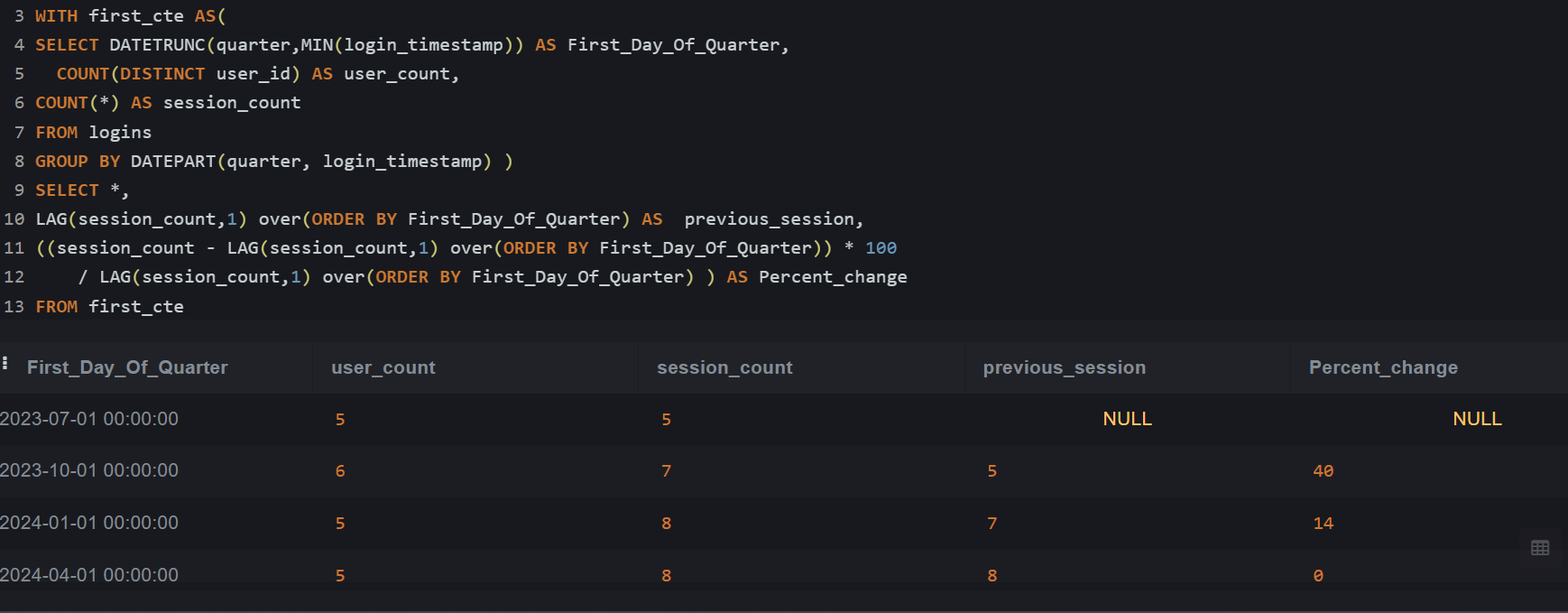
*SELECT \*,*

*LAG(session\_count,1) over(order by First\_Day\_Of\_Quarter) AS previous\_session,*

*((session\_count - LAG(session\_count,1) over(order by First\_Day\_Of\_Quarter)) \* 100*

*/ LAG(session\_count,1) over(order by First\_Day\_Of\_Quarter) ) AS Percent\_change*

*FROM first\_cte*



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

*WITH temp AS(*

*SELECT user\_id, login\_timestamp,SUM(session\_score) AS score*

*FROM logins*

*GROUP BY user\_id,login\_timestamp*

*)*

*,temp1 AS (*

*SELECT \*,*

*RANK() over( PARTITION by login\_timestamp order BY score DESC ) AS rn*

*FROM temp*

*)*

*SELECT user\_id,login\_timestamp FROM temp1 WHERE* rn=1



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

*WITH temp AS(*

*SELECT user\_id, min(login\_timestamp) AS First\_Login, MAX(login\_timestamp) AS Last\_Login, COUNT(session\_id) as total\_logins*

*FROM logins*

*GROUP BY user\_id*

*),*

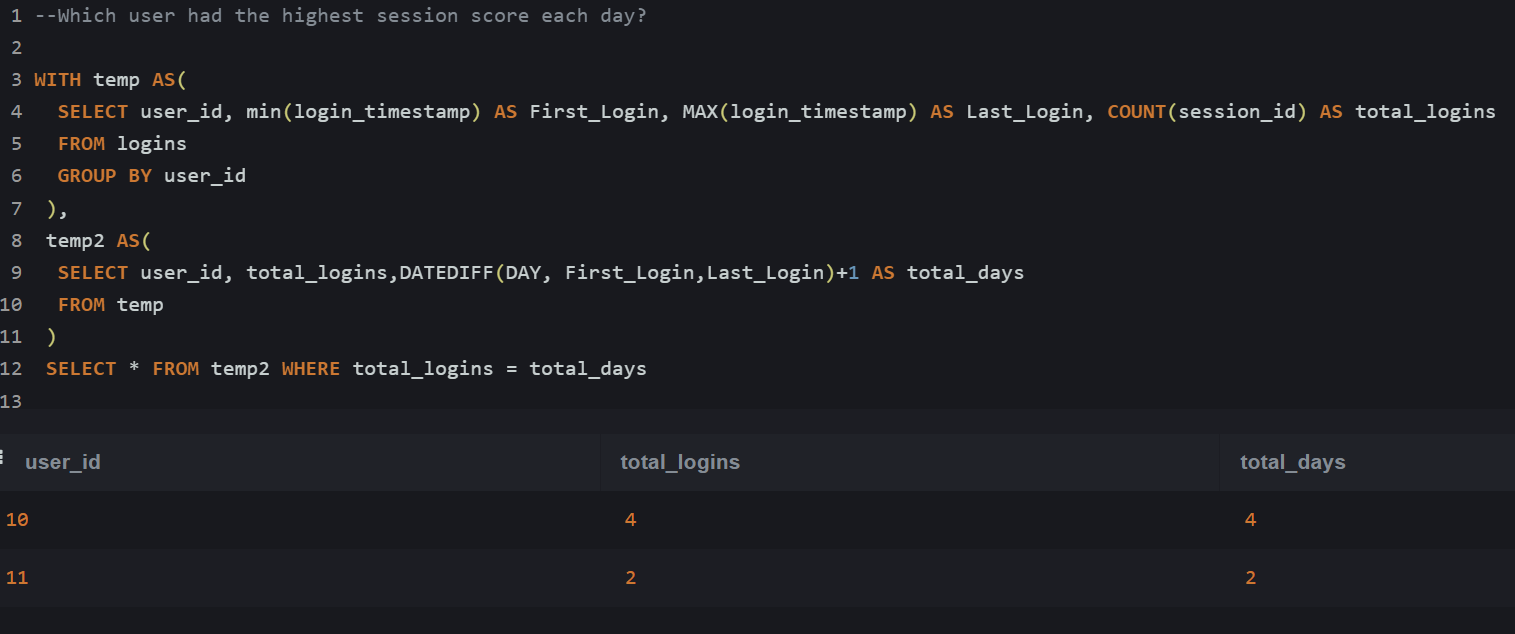
*temp2 AS(*

*SELECT user\_id, total\_logins,DATEDIFF(DAY, First\_Login,Last\_Login)+1 AS total\_days*

*FROM temp*

*)*

*SELECT \* FROM temp2 WHERE total\_logins = total\_days*



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

*WITH temp AS*

*(*

*select MIN(login\_timestamp) AS first\_login\_date, MAX(login\_timestamp) AS last\_login\_date*

*from logins*

*UNION ALL*

*select DATEADD(DAY,1, first\_login\_date) AS first\_login\_date, last\_login\_date*

*from temp*

*WHERE first\_login\_date < last\_login\_date*

*)*

*SELECT \**

*FROM temp WHERE first\_login\_date NOT IN (SELECT DISTINCT login\_timestamp from logins)*

*option(maxrecursion 400)*

