

Querying the Database

TABLE:USERS Fields and constraints : USER_ID INT PRIMARY KEY, USER_NAME VARCHAR(20) NOT NULL, USER_STATUS VARCHAR(20) NOT NULL

TABLE:LOGINS Fields and constraints :USER_ID INT, LOGIN_TIMESTAMP DATETIME NOT NULL, SESSION_ID INT PRIMARY KEY, SESSION_SCORE INT, FOREIGN KEY (USER_ID) REFERENCES USERS(USER_ID)

OVERVIEW OF RELATIONAL TABLES

- `select * from logins;`
- `select * from users;`
- Logins:

USER_ID	LOGIN_TIMESTAMP	SESSION_ID	SESSION_SCORE
1	2023-07-15 09:30:00.000	1001	85
2	2023-07-22 10:00:00.000	1002	90
3	2023-08-10 11:15:00.000	1003	75
4	2023-08-20 14:00:00.000	1004	88
5	2023-09-05 16:45:00.000	1005	82
6	2023-10-12 08:30:00.000	1006	77
7	2023-11-18 09:00:00.000	1007	81
8	2023-12-01 10:30:00.000	1008	84
9	2023-12-15 13:15:00.000	1009	79
10	2024-06-25 15:00:00.000	1010	92
1	2024-01-10 07:45:00.000	1011	86
2	2024-01-25 09:30:00.000	1012	89
3	2024-02-05 11:00:00.000	1013	78
4	2024-03-01 14:30:00.000	1014	91
5	2024-03-15 16:00:00.000	1015	83
6	2024-04-12 08:00:00.000	1016	80

Users :

USER_ID	USER_NAME	USER_STATUS
1	Alice	Active
2	Bob	Inactive
3	Charlie	Active
4	David	Active
5	Eve	Inactive
6	Frank	Active
7	Grace	Inactive
8	Heidi	Active
9	Ivan	Inactive
10	Judy	Active

1. Management wants to know who did not login for last 5 months

- Query : `select user_id, max(login_timestamp) as last_login
from logins group by user_id
having max(login_timestamp) < DATEADD(month,-5,getdate())`

Output:

user_id	last_login
1	2024-01-10 07:45:00.000
2	2024-01-25 09:30:00.000
3	2024-02-05 11:00:00.000
4	2024-03-01 14:30:00.000

2.How many sessions , users at each quarter (return , quarter , first day of quarter , no of sessions , no of users in quarter)

Query : `select DATEPART(QUARTER,login_timestamp) as quarter ,
Dateadd(Quarter,DATEDIFF(quarter,0,min(LOGIN_TIMESTAMP)),0) as First_day,
count(user_id) as no_of_sessions,
count(distinct user_id) as no_of_users,
min(login_timestamp) as first_login_on
from logins
group by DATEPART(QUARTER,login_timestamp);`

OUTPUT2:

quarter	First_day	no_of_sessions	no_of_users	first_login_on
1	2024-01-01 00:00:00.000	8	5	2024-01-10 07:45:00.000
2	2024-04-01 00:00:00.000	8	5	2024-04-12 08:00:00.000
3	2023-07-01 00:00:00.000	5	5	2023-07-15 09:30:00.000
4	2023-10-01 00:00:00.000	7	6	2023-10-12 08:30:00.000

3. Select a user_id where he logged in january 2024 and not logged in november 2023

Query: `select distinct user_id from logins
where login_timestamp between '2024-01-01' and '2024-01-31'
except
select user_id from logins where LOGIN_TIMESTAMP
between '2023-11-01' and '2023-11-30'`

- OUTPUT:

Results		Messages
	user_id	
1	1	
2	3	
3	5	

4.From Query NO.2 add percentage change in sessions from last column
-- Return : first_day_of_Quarter ,session_cnt , prev_session ,session_change_
percentage_change

Query:

```
with cte as (select DATEPART(QUARTER,login_timestamp) as quarter ,
Dateadd(Quarter,DATEDIFF(quarter,0,min(LOGIN_TIMESTAMP)),0) as First_day,
count(user_id) as session_cnt,
lag(count(user_id)) over(order by DATEPART(QUARTER,login_timestamp)) as lagi,
min(login_timestamp) as first_login_on
from logins
group by DATEPART(QUARTER,login_timestamp))
select First_day,session_cnt,lagi as prev_session,round(cast((session_cnt-lagi)*100.0/session_cnt as float) ,2) as
percentage_change_of_session from cte
```

OUTPUT:

First_day	session_cnt	prev_session	percentage_change_of_session
2024-01-01 00:00:00.000	8	NULL	NULL
2024-04-01 00:00:00.000	8	8	0
2023-07-01 00:00:00.000	5	8	-60
2023-10-01 00:00:00.000	7	5	28.57

5.Display the User that has highest session_score for each day

--Return : Date , user_Name,score

Query:

```
select cast(LOGIN_TIMESTAMP as date) as Date,  
s.USER_NAME ,max(SESSION_SCORE) as Highest_session_score  
from Users s join logins l  
on s.user_id = l.user_id  
group by cast(LOGIN_TIMESTAMP as date),  
s.USER_NAME order by Date
```

OUTPUT:

Date	USER_NAME	Highest_session_score
2023-07-15	Alice	85
2023-07-22	Bob	90
2023-08-10	Charlie	75
2023-08-20	David	88
2023-09-05	Eve	82
2023-10-12	Frank	77
2023-11-10	Bob	82
2023-11-15	Frank	80
2023-11-18	Grace	81
2023-11-25	David	84
2023-12-01	Heidi	84
2023-12-15	Ivan	79
2024-01-10	Alice	86
2024-01-15	Eve	78
2024-01-25	Bob	89
2024-01-25	Charlie	89
2024-02-05	Charlie	78
2024-03-01	David	91
2024-03-15	Eve	83
2024-04-12	Frank	80
2024-05-18	Grace	82
2024-05-28	Heidi	87
2024-06-15	Ivan	76
2024-06-25	Judy	92
2024-06-26	Judy	93
2024-06-27	Judy	92
2024-06-28	Judy	93

6. To Identify best users

-- Return the Users that had a session in every single possible date

Query:

```
select USER_ID,min(cast(login_timestamp as date))as First_login,  
max(cast(login_timestamp as date )) as Last_Login,  
datediff(day,min(cast(login_timestamp as date )),GETDATE()+1 as date,  
count(distinct cast(login_timestamp as date )) as no_of_logins  
from logins  
group by USER_ID  
order by no_of_logins desc;
```

OUTPUT:

USER_ID	First_login	Last_Login	date	no_of_logins
10	2024-06-25	2024-06-28	38	4
2	2023-07-22	2024-01-25	377	3
3	2023-08-10	2024-02-05	358	3
4	2023-08-20	2024-03-01	348	3
5	2023-09-05	2024-03-15	332	3
6	2023-10-12	2024-04-12	295	3
7	2023-11-18	2024-05-18	258	2
8	2023-12-01	2024-05-28	245	2
9	2023-12-15	2024-06-15	231	2
1	2023-07-15	2024-01-10	384	2

7. Find all login dates where there are no logins
 -- Return login dates

Query: with cte as (
 select cast('2023-07-15 09:30:00.000' as date) as date1
 union all
 select DATEADD(day,1,date1) from cte where date1 < '2024-06-28'
)
 select date1 from cte
 where date1 not in (select cast(LOGIN_TIMESTAMP as date)
 from logins)option(maxrecursion 500)

OUTPUT:

date1	date1	date1
2023-07-16	2023-08-23	2023-09-27
2023-07-17	2023-08-24	2023-09-28
2023-07-18	2023-08-25	2023-09-29
2023-07-19	2023-08-26	2023-09-30
2023-07-20	2023-08-27	2023-10-01
2023-07-21	2023-08-28	2023-10-02
2023-07-23	2023-08-29	2023-10-03
2023-07-24	2023-08-30	2023-10-04
2023-07-25	2023-08-31	2023-10-05
2023-07-26	2023-09-01	2023-10-06
2023-07-27	2023-09-02	2023-10-07
2023-07-28	2023-09-03	2023-10-08
2023-07-29	2023-09-04	2023-10-09
2023-07-30	2023-09-06	2023-10-10
2023-07-31	2023-09-07	2023-10-11
2023-08-01	2023-09-08	2023-10-12
2023-08-02	2023-09-09	2023-10-13
2023-08-03	2023-09-10	2023-10-14
2023-08-04	2023-09-11	2023-10-15
2023-08-05	2023-09-12	2023-10-16
2023-08-06	2023-09-13	2023-10-17
2023-08-07	2023-09-14	2023-10-18
2023-08-08	2023-09-15	2023-10-19
2023-08-09	2023-09-16	2023-10-20
2023-08-11	2023-09-17	2023-10-21
2023-08-12	2023-09-18	2023-10-22
2023-08-13	2023-09-19	2023-10-23
2023-08-14	2023-09-20	2023-10-24
2023-08-15	2023-09-21	2023-10-25
2023-08-16	2023-09-22	2023-10-26
2023-08-17	2023-09-23	2023-10-27
2023-08-18	2023-09-24	2023-10-28
2023-08-19	2023-09-25	2023-10-29
2023-08-21	2023-09-26	2023-10-30