User Engagement Analysis

Analyse user engagement to find the most engaging posts and users based on the given conditions.

Retrieve the comprehensive count of likes, comments, and shares garnered by a specific post identified by its unique post ID

. Calculate the mean number of reactions, encompassing likes, comments, and shares per distinct user within a designated time period

Identify the three most engaging posts, measured by the aggregate sum of reactions, within the preceding week

Posts:

post_id	post_content	post_date
1	Lorem ipsum dolor sit amet...	2023-08-25 10:00:00
2	Exploring the beauty of nature...	2023-08-26 15:30:00
3	Unveiling the latest tech trends...	2023-08-27 12:00:00
4	Journey into the world of literature...	2023-08-28 09:45:00
5	Capturing the essence of city life...	2023-08-29 16:20:00

UserReactions:

reaction_id	user_id	post_id	reaction_type	reaction_date
1	101	1	like	2023-08-25 10:15:00
2	102	1	comment	2023-08-25 11:30:00
3	103	1	share	2023-08-26 12:45:00
4	101	2	like	2023-08-26 15:45:00
5	102	2	comment	2023-08-27 09:20:00
6	104	2	like	2023-08-27 10:00:00
7	105	3	comment	2023-08-27 14:30:00
8	101	3	like	2023-08-28 08:15:00
9	103	4	like	2023-08-28 10:30:00
10	105	4	share	2023-08-29 11:15:00
11	104	5	like	2023-08-29 16:30:00
12	101	5	comment	2023-08-30 09:45:00

```
create table  Posts
(post_id INT,
Post_Content VARCHAR(150),
Post_Date TIMESTAMP);
```

```
INSERT INTO Posts
(Post_ID, Post_Content, Post_Date)
VALUES
(1,'Lorem ipsum dolor sit amet...', '2023-08-25 10:00:00'),
(2,'Exploring the beauty of nature...', '2023-08-26 15:30:00'),
(3,'Unveiling the latest tech trends...', '2023-08-27 12:00:00'),
(4,'Journey into the world of literature...', '2023-08-28 09:45:00'),
(5,'Capturing the essence of city life...', '2023-08-29 16:20:00');

SELECT * from Posts

create table  UserReactions
(reaction_id INT,
user_id INT,
post_id INT,
reaction_type VARCHAR(200),
reaction_date TIMESTAMP);

 user_id INT,
 Post_id int ,
reaction_type VARCHAR(150),
reaction_Date TIMESTAMP);

insert into UserReactions
(reaction_ID, user_id, post_id, reaction_type, reaction_date)
VALUES
(1,101,1,'like','2023-08-25 10:15:00'),
(2,102,1,'comment','2023-08-25 11:30:00'),
(3,103,1,'share','2023-08-26 12:45:00'),
```

```
(4,101,2,'like','2023-08-26 15:45:00'),
(5,102,2,'comment','2023-08-27 09:20:00'),
(6,104,2,'like','2023-08-27 10:00:00'),
(7,105,3,'comment','2023-08-27 14:30:00'),
(8,101,3,'like','2023-08-28 08:15:00'),
(9,103,4,'like','2023-08-28 10:30:00'),
(10,105,4,'share','2023-08-29 11:15:00'),
(11,104,5,'like','2023-08-29 16:30:00'),
(12,101,5,'comment','2023-08-30 09:45:00');
```

1) **Analyse user engagement to find the most engaging posts and users based on the given conditions.**

```
SELECT p.post_id, p.post_content,
COUNT(CASE WHEN u.reaction_type = 'like' THEN 1 end ) as likes,
COUNT(CASE WHEN u.reaction_type = 'comment' THEN 1 END) as comments,
COUNT(CASE WHEN u.reaction_type = 'share' THEN 1 end) as shares from
UserReactions as u
join posts as p on p.post_id = u.post_id
group by  p.post_id, p.post_content
order by p.post_id;
```

2) **Retrieve the comprehensive count of likes, comments, and shares garnered by a specific post identified by its unique post ID**

```
select DATE(U.reaction_date) as reaction_day, avg(count(*)) over (partition BY
DATE(U.reaction_date))AS MEAN_REACTIONS FROM UserReactions AS U
where U.reaction_date BETWEEN '2023-08-25' and '2023-08-30'
group by reaction_day;
```

(or)

```
SELECT user_id, AVG(likes)/6 as AVG_LIKES, avg(comments)/6 AS AVG_COMMENTS,
avg(shares)/6 AS AVG_SHARES
FROM (
   SELECT user_id,
```

```
count(CASE WHEN reaction_type = 'like' THEN 1 end ) as likes,
count(CASE WHEN reaction_type = 'comment' THEN 1 END) as comments,
count(CASE WHEN reaction_type = 'share' THEN 1 end) as shares
   FROM UserReactions
   GROUP BY user_id
) AS user_reaction_counts
GROUP BY user_id;
```

```
select DATE(U.reaction_date) as reaction_day, avg(count(*)) over (partition BY
DATE(U.reaction_date))AS MEAN_REACTIONS FROM UserReactions AS U
where U.reaction_date BETWEEN '2023-08-25' and '2023-08-30'
group by reaction_day;
```

3) **Identify the three most engaging posts, measured by the aggregate sum of reactions, within the preceding week**

```
SELECT
   post_id,
   SUM(CASE WHEN reaction_type = 'like' THEN 1 WHEN reaction_type = 'comment'
THEN 1 WHEN reaction_type = 'share' THEN 1 END) AS total_reactions
FROM UserReactions
WHERE reaction_date <'2023-08-28 08:15:00'
GROUP BY post_id
ORDER BY total_reactions DESC
LIMIT 3;
```