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
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

Output:

1.

Post_id	posst_content	numl_likes	num_commetns	num_shares
2	Exploring the beauty of nature	2	1	0

2.

reaction_day	distinct_users	total_reactions	avg_reactions_per_user
2023-08-25	2	2	2.0000
2023-08-26	2	2	2.0000
2023-08-27	3	3	3.0000
2023-08-28	2	2	2.0000
2023-08-30	2	2	2.0000
2023-08-30	1	1	1.0000

3.

post_id	post_content	total_reactions
1	Lorem ipsum dolor sit amet	3
2	Exploring the beauty of nature	3
3	Unveiling the latest tech trends	2

SQL CODE:

```
DROP TABLE Posts


CREATE TABLE Posts (
    post_id INT PRIMARY KEY,
    post_content TEXT,
    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');

CREATE TABLE UserReactions (
    reaction_id INT PRIMARY KEY,
    user_id INT,
    post_id INT,
    reaction_type VARCHAR(10),
    reaction_date TIMESTAMP,
    FOREIGN KEY (post_id) REFERENCES Posts(post_id)
);

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');

select p.post_id,p.post_content,
sum(CASE when u.reaction_type='like' then 1 else 0 end) as num_likes,
sum(CASE when u.reaction_type='comment' then 1 else 0 end) as num_comment,
```

```
sum(case when u.reaction_type='share' then 1 else 0 end) as num_shares
from Posts p
join UserReactions u
on p.post_id=u.post_id
GROUP BY p.post_id,p.post_content

select reaction_date,user_id as distinct_users,
sum(case when reaction_type then 1 end) as total_reactions,
total_reaction/count(user_id) as avg_reaction_per_user
from
UserReactions
group by reaction_date

SELECT reaction_day,
count(DISTINCT user_id) as distinct_users,
count(*) total_reactions,
round(avg(reaction_per_user),4) as avg_reactions_per_user
from(
select reaction_date::date as reaction_day,
user_id,
count(*)reaction_per_user
from UserReactions
GROUP by reaction_date,user_id)
group by reaction_day
ORDER by reaction_day

SELECT p.post_id,p.post_content,count(*) as total_reactions
from
Posts p
```

```
join
UserReactions u
on p.post_id=u.post_id
group by p.post_id,p.post_content
order by total_reactions desc,
post_id
limit 3
```