

User Activity Project: SQL Data Exploration

Column	Type	Description
user_id	STRING	Unique user identifier
region	STRING	Region code (e.g., NA, EU, APAC)
activity_type	STRING	Type of activity (e.g., login, view, click)
activity_date	DATE	Date of the activity
engagement_score	FLOAT	Numeric score assigned per session

Question
1. Write a SQL query that returns the number of distinct users per region in the last 30 days.
2. Write a SQL query that shows, for each region, the average engagement score per activity_type.
3. Write a query that returns the top 5 users with the highest total engagement score in the last 60 days.
4. How would you define and create a calculated field to measure user adoption rate?
5. Please suggest a formula you would use, and describe what clarifying questions you might ask the business team to ensure the metric aligns with their expectations.

User Activity Project: SQL Data Exploration

SQL Syntax	Answer / Comment
<pre>SELECT COUNT(DISTINCT user_id) AS "Distinct_Users", region AS "Region" FROM user_activity WHERE activity_date >= CURRENT_DATE -30 GROUP BY region ORDER BY "Distinct_Users" DESC;</pre>	I assume that any type of activity qualifies a user as active, so I check whether the user had any activity within the past 30 days.
<pre>SELECT region AS "Region", activity_type AS "Activity_Type", ROUND(AVG(engagement_score:: numeric), 2) AS "Avg_Engagement_Score" FROM user_activity GROUP BY region, activity_type ORDER BY region ASC, activity_type ASC;</pre>	
<pre>WITH ranked_users AS (SELECT user_id AS "User_ID", ROUND(SUM(engagement_score::numeric), 2) AS "Total_Engagement_Score", RANK() OVER (ORDER BY SUM(engagement_score) DESC) AS "Rank" FROM user_activity WHERE activity_date >= CURRENT_DATE -60 GROUP BY user_id) SELECT * FROM ranked_users WHERE "Rank" <= 5 ORDER BY "Rank";</pre>	
<pre>WITH user_activity_period AS (SELECT user_id FROM user_activity WHERE activity_date >= CURRENT_DATE - 14), user_activity_counts AS (SELECT user_id, COUNT(*) AS activity_count FROM user_activity_period GROUP BY user_id), adoption_grouped AS (SELECT CASE WHEN activity_count >= 2 THEN 'Adopted' ELSE 'Not Adopted' END AS "Status", COUNT(*) AS "User_Count" FROM user_activity_counts GROUP BY "Status"), totals AS (SELECT SUM("User_Count") AS total_users FROM adoption_grouped) SELECT adoption_grouped."Status", adoption_grouped."User_Count", ROUND(100.0 * adoption_grouped."User_Count" / NULLIF(totals.total_users, 0), 2) AS "Adoption_Rate_%" FROM adoption_grouped, totals ORDER BY adoption_grouped."Status";</pre>	<p>Adoption depends on which adoption behavior we want to measure, but generally I would measure the percentage of users who performed a specific action (e.g. were active during a certain time period, used a specific feature or interacted with a part of the product) out of the total users. Before defining, I would first check with the business KPIs and confirm the data support the definition.</p> <p>In this case, and after reviewing the generated data, I decided to define adoption with at least 2 activity events in the past 14 days.</p>

1. Deviation from Engagement KPI

```
WITH weekly_data AS (  
  SELECT (activity_date - EXTRACT(DOW FROM activity_date)::int)::date AS week_start, COUNT  
  (*) AS "Actual_Total_Activities",  
  4 * 7 AS "KPI", COUNT(*) - 4 * 7 AS "Deviation",  
  ROUND(100.0 * (COUNT(*) - 4 * 7) / (4 * 7), 2) AS "Deviation_%"  
  FROM user_activity  
  GROUP BY week_start)  
SELECT CONCAT("Week ", ROW_NUMBER() OVER (ORDER BY week_start)) AS "Week_#",  
"Actual_Total_Activities", "KPI", "Deviation", "Deviation_%"  
FROM weekly_data  
ORDER BY week_start ASC;
```

2. Trend Using Simple Linear Regression

```
WITH daily_activity AS (  
  SELECT (activity_date::date) AS "Date", COUNT(*) AS "Total_Activities"  
  FROM user_activity  
  GROUP BY activity_date::date),  
numbered_days AS (  
  SELECT "Date", "Total_Activities", ROW_NUMBER() OVER (ORDER BY "Date") AS  
  "Day_Number"  
  FROM daily_activity), stats AS (  
  SELECT COUNT(*) AS "Total_Days", SUM("Day_Number") AS "Sum_Day_Numbers", SUM  
  ("Total_Activities") AS "Sum_Total_Activities", SUM("Day_Number" * "Total_Activities") AS  
  "Sum_Day_Mult_Activities", SUM("Day_Number" * "Day_Number") AS "Sum_Day_Squared"  
  FROM numbered_days)  
SELECT "Total_Days", "Sum_Day_Numbers", "Sum_Total_Activities",  
"Sum_Day_Mult_Activities", "Sum_Day_Squared", ROUND(("Total_Days" *  
"Sum_Day_Mult_Activities" - "Sum_Day_Numbers" * "Sum_Total_Activities")::numeric / NULLIF  
(("Total_Days" * "Sum_Day_Squared" - "Sum_Day_Numbers" * "Sum_Day_Numbers"), 0), 4) AS  
"Slope_Per_Day"  
FROM stats;
```

Before suggesting a formula, I would start by clarifying a few key points to ensure the goal is clear:

- * **Validate the source of the data:** is it automated or manually tagged? can additional fields or sources be integrated to improve the context for me?
- * **Review a sample of the raw data:** check for missing values, duplicates, test/internal accounts, outliers or other anomalies.
- * **Identify known or potential data quality issues:** bot activity, non-production/test environments or inconsistent input sources.
- * **Check whether data from previous time periods is available for comparison.**

Then, I would align with the business team on the following:

- * **Confirm the definition of engagement:** does it refer to any interaction or a specific type or volume of activity?
- * **Clarify which user behaviors are the most important:** for example, what is expected to happen after a click? what action defines meaningful engagement?
- * **Determine what are the existing KPIs we should measure against** and what benchmarks we aim to reach.
- * **Understand how the engagement score is calculated:** is it based on clicks, session duration, feature usage or something else?
- * **Define the goal of tracking engagement:** are we trying to increase conversions, to boost retention, validate product changes or improve overall usage?
- * **Confirm the reporting breakdown:** should data be segmented by platform, product, region or user type (executives, developers, etc.)?

Once these are clarified, here are two example formulas that could support the business:

1. Deviation from Engagement KPI

Assuming a KPI of 3 activities per user per day, I calculated the actual daily average and compared it to the target. In this case, all results show a deviation from the expected KPI, indicating lower-than-target engagement levels.

2. Trend Using Simple Linear Regression

To assess whether user engagement is increasing or decreasing over time, I applied a simple linear regression to the total daily activity. The slope of the regression line represents the average change in activity per day. A positive slope indicates rising engagement, while a negative slope indicates a decline. This helps identify overall patterns that may not be visible in daily fluctuations. In this dataset, the trend shows a little increase in engagement over time.

User Activity Project: SQL Data Exploration

#1

	Distinct_Users	Region
1	21	NA
2	18	LATAM
3	16	APAC
4	14	MEA
5	10	EU

#3

	User_ID	Total_Engagement_Score	Rank
1	153	9.35	1
2	12	9.16	2
3	133	7.92	3
4	51	7.86	4
5	80	7.58	5

#2

	Region	Activity_Type	Avg_Engagement_Score
1	APAC	click	2.84
2	APAC	login	2.51
3	APAC	view	1.94
4	EU	click	2.83
5	EU	login	3.64
6	EU	view	3.10
7	LATAM	click	2.94
8	LATAM	login	2.28
9	LATAM	view	2.79
10	MEA	click	2.67
11	MEA	login	2.86
12	MEA	view	2.32
13	NA	click	2.46
14	NA	login	2.40
15	NA	view	2.10

#5.1

	Week_#	Actual_Total_Activities	KPI	Deviation	Deviation_%
1	Week 1	14	28	-14	-50.00
2	Week 2	19	28	-9	-32.14
3	Week 3	17	28	-11	-39.29
4	Week 4	12	28	-16	-57.14
5	Week 5	22	28	-6	-21.43
6	Week 6	19	28	-9	-32.14
7	Week 7	22	28	-6	-21.43
8	Week 8	24	28	-4	-14.29
9	Week 9	11	28	-17	-60.71
10	Week 10	23	28	-5	-17.86
11	Week 11	22	28	-6	-21.43
12	Week 12	17	28	-11	-39.29
13	Week 13	22	28	-6	-21.43
14	Week 14	6	28	-22	-78.57

#4

#5.2

	Status	User_Count	Adoption_Rate_%
1	Adopted	1	2.70
2	Not Adopted	36	97.30

	Total_Days	Sum_Day_Numbers	Sum_Total_Activities	Sum_Day_Mult_Activities	Sum_Day_Squared	Slope_Per_Day
1	86	3741	250	11065	215731	0.0036

