

## SQL Case Study: User Activity & Sessions

**Dataset:** users, logins

**Goal:** Solve business questions on user activity using SQL

---

### ✅ Q1: Users with No Login in Last 5 Months

**Objective:** Find users **inactive for the past 5 months**.

**Approach:**

- Get the **latest login date**.
- Filter users whose **last login** was **before 5 months** from that date.

**Key SQL Concepts:** DATEADD(), MAX(), NOT IN, HAVING MAX()

---

### ✅ Q2: Quarterly Summary – User & Session Count

**Objective:** Show user count and session count **per quarter**.

**Approach:**

- Use DATETRUNC(QUARTER) to group by quarter.
- Count distinct USER\_IDs and SESSION\_IDs.

**Key SQL Concepts:** DATETRUNC, COUNT(DISTINCT), GROUP BY

---

### ✅ Q3: Users Logged in Jan 2024 but Not Nov 2023

**Objective:** Filter users who were **active in Jan 2024** but **not in Nov 2023**.

**Approach:**

- Use MONTH() and YEAR() filters.
- Exclude those present in Nov 2023 using NOT IN.

**Key SQL Concepts:** MONTH(), NOT IN, DISTINCT

---

### ✅ Q4: Add % Change in Sessions from Last Quarter

**Objective:** Extend Q2 with **quarter-over-quarter session growth**.

**Approach:**

- Use LAG() window function to get **previous quarter's session count**.

- Calculate % change using a formula.

**Key SQL Concepts:** LAG(), CTE, ROUND(), OVER(ORDER BY)

---

✅ **Q5: Top User by Score per Day**

**Objective:** Find user with **highest total session score each day**.

**Approach:**

- Group by DATE and USER\_NAME.
- Use ROW\_NUMBER() to rank users by daily score and filter for rn = 1.

**Key SQL Concepts:** ROW\_NUMBER(), PARTITION BY, SUM(), CTE

---

✅ **Q6: Best Users – Logged in Every Day Since First Login**

**Objective:** Identify users who logged in **every single day** since their **first login**.

**Approach:**

- Get first\_login per user.
- Count required days and compare with actual unique login days.

**Key SQL Concepts:** DATEDIFF(), HAVING, COUNT(DISTINCT)

---

✅ **Q7: Dates with No Login Activity**

**Objective:** List dates with **zero logins** from start to '2024-06-28'.

**Approach:**

- Generate a **date calendar** using recursive CTE.
- Exclude dates found in logins.

**Key SQL Concepts:** Recursive CTE, DATEADD(), NOT IN, MAXRECURSION