

The Leaky Bucket

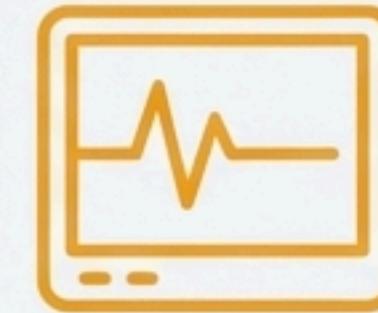
A SQL Case Study in Solving Customer Churn



A Tale of Two Truths at FitLife Pro



The Marketing team is celebrating record numbers of new users acquired in January.



The Finance team is worried because total revenue barely increased.

“The CEO suspects a ‘Leaky Bucket’ problem: We are pouring new users in, but old users are leaking out just as fast.”

The Investigation Begins: Define the Metric

Before writing a single line of SQL, you must define what you are measuring. To prove to the CEO that we have a retention problem, our objective is to calculate one specific, critical metric:



****Monthly Churn Rate for
January 2024****

Examining the Evidence: The 'Subscriptions' Table

Column	Type	Description
User_ID	INT	Unique ID for the user
Signup_Date	DATE	When they started paying
Cancel_Date	DATE	When they stopped paying (NULL if still active)
Plan_Price	DECIMAL	e.g., 9.99

The Formula for Finding the Truth

****Users who cancelled in Jan****

****Users Active on Jan 1st****

Our entire SQL query will be built to calculate these two numbers.

The Blueprint, Part 1: Identifying “The Leakers” (Numerator)

A user is counted as a 'leaker' if their `Cancel_Date` falls within the month of January.

Cancel_Date is between '2024-01-01'
AND '2024-01-31'.



The Blueprint, Part 2: Defining “The Starting Base” (Denominator)

This is trickier. A user was ‘Active on Jan 1st’ only if they met two conditions:



They signed up before Jan 1st.

```
Signup_Date < '2024-01-01'
```



They hadn't cancelled yet.

(Meaning their Cancel_Date is either NULL or *after* Jan 1st)

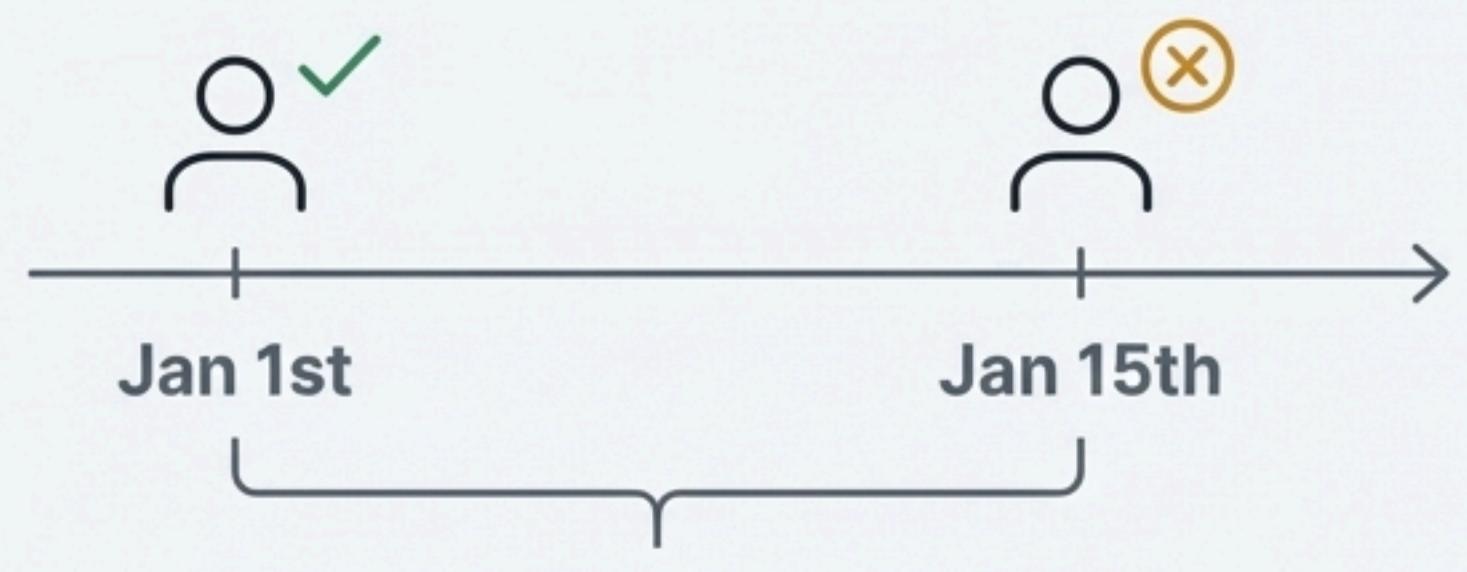
Expert Insight: The Denominator Logic Trap

The Mistake:

Many people make the mistake of just checking `Cancel_Date IS NULL`. This is wrong.

The Correct Logic:

Someone who cancels on **Jan 15th** was still active on **Jan 1st**. They must be counted as part of the starting base, even if they left later that month.



The Takeaway: That is why we must check
`'Cancel_Date IS NULL OR
Cancel_Date >= '2024-01-01''`

Assembling the Full Logical Blueprint

Numerator (The Leakers)

```
Cancel_Date >= '2024-01-01'
```

AND

```
Cancel_Date <= '2024-01-31'
```

Denominator (The Starting Base)

```
Signup_Date < '2024-01-01'
```

```
AND (Cancel_Date IS NULL OR  
Cancel_Date >= '2024-01-01')
```

With this logic defined, we can now build the SQL query.

The Tool: Building the SQL Query

```
SELECT
```

```
-- 1. Count the Leakers (Numerator)
```

```
COUNT(CASE WHEN Cancel_Date >= '2024-01-01' AND Cancel_Date <= '2024-01-31'  
THEN 1 END) AS Churned_Users,
```

→ Numerator Logic

```
-- 2. Count the Starting Base (Denominator)
```

```
COUNT(CASE WHEN Signup_Date < '2024-01-01' AND (Cancel_Date IS NULL OR  
Cancel_Date >= '2024-01-01') THEN 1 END) AS Active_Users_Jan1,
```

→ Denominator Logic

```
-- 3. The Math (Churn Rate %)
```

```
100.0 * COUNT(CASE WHEN Cancel_Date >= '2024-01-01' AND Cancel_Date <=  
'2024-01-31' THEN 1 END)
```

```
/
```

```
NULLIF(COUNT(CASE WHEN Signup_Date < '2024-01-01' AND (Cancel_Date IS NULL  
OR Cancel_Date >= '2024-01-01') THEN 1 END), 0) AS Churn_Rate_Percentage
```

→ The Math

```
FROM Subscriptions;
```

Expert Insight: Making the Query Bulletproof



Avoiding Integer Division

$$5 / 10 = \boxed{0} \xrightarrow{\hspace{1cm}} 5.0 / 10 = \boxed{0.5}$$

Multiplying by `100.0` forces the database to use float division, ensuring a precise decimal result.

The Division-by-Zero Safety Net

$$\dots / 0 \xrightarrow{\hspace{1cm}} \text{ERROR} \triangle \xrightarrow{\hspace{1cm}} \text{NULLIF}(\dots, 0) \xrightarrow{\hspace{1cm}} \text{NULL}$$

`NULLIF(..., 0)` is a safety net. It turns a 0 in the denominator into a `NULL`, which makes the final result `NULL` instead of an error.

The Reveal: The Numbers Don't Lie

Churned Users

500

Active Users on Jan 1st

2,000

January Churn Rate

25.0%

The Verdict: 25% Monthly Churn is Catastrophic

A 25% monthly churn rate means you lose your entire customer base in 4 months.



From Analysis to Action

The SQL analysis provides a clear, data-driven mandate for business strategy.

“

It doesn't matter that we acquired 600 new users this month. We lost 25% of our existing base. We need to **pause Marketing spend** and **fix the Product immediately**.

”

The Analyst as a Strategic Partner

A single, well-crafted query did more than answer a question.
It settled a debate, revealed a critical business threat,
and dictated immediate strategic action.



This is the power of SQL.