

# USER RETENTION ANALYSIS

---

FEB 2024

## Union, Join, Concat & Nested CTE

LinkedIn Coding Challenge

Intermediate

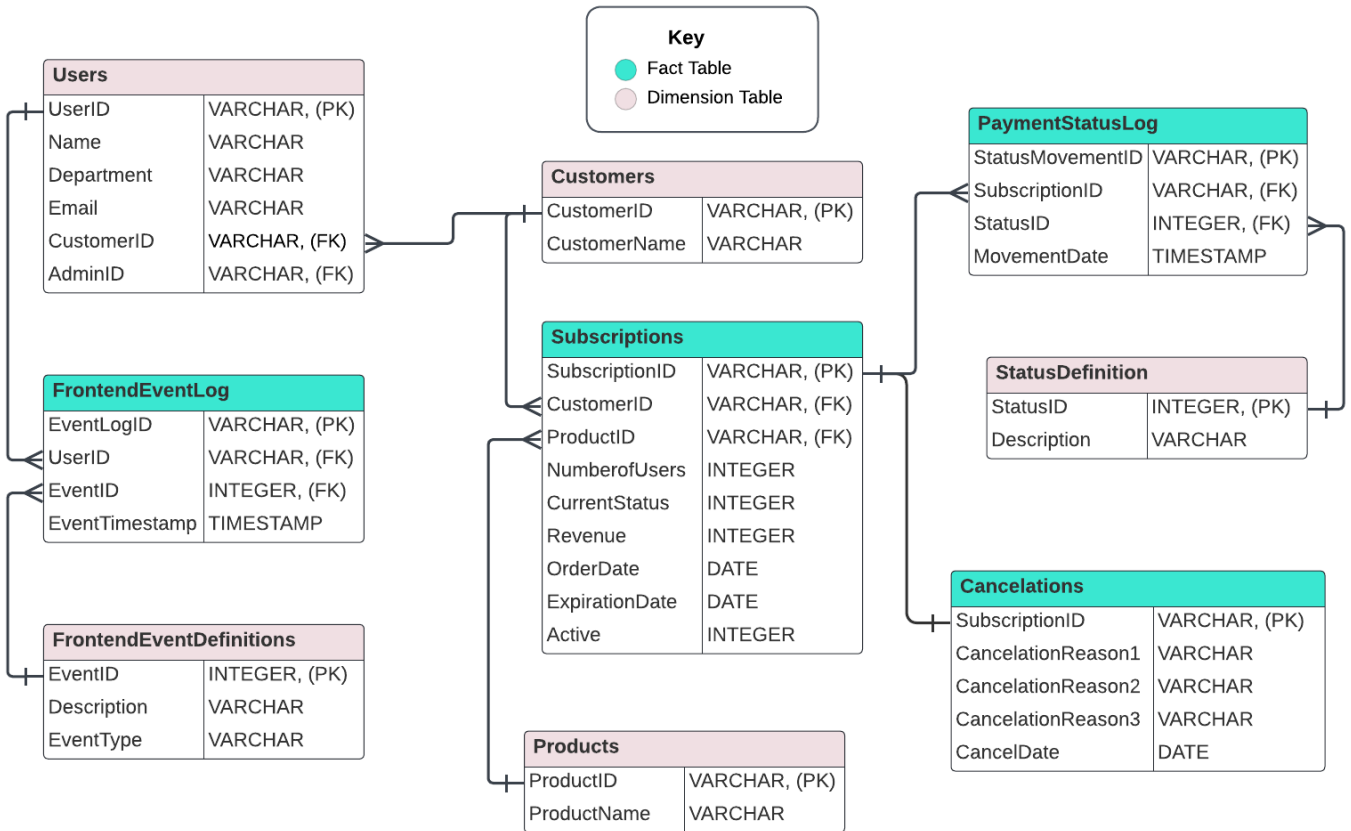
A chief growth officer at a SaaS startup has requested a user retention analysis on clients who do not renew their subscriptions to understand whether a) they are leaving for a competitor OR b) the product has become too expensive to fit into the client's budget. The analysis is important in order to mitigate churn in the new year. The team's initial hypothesis is that the product cost is no longer feasible within a contracting economy.

Test the hypothesis by finding what percentage of users cancelled their subscriptions due to it being too expensive.

# INPUT FORMAT

The main source tables are **CANCELATIONS** and **SUBSCRIPTIONS**.

## Main Data Model



## CODE SOLUTION

---

```
WITH percent AS
(
WITH all_cancelation_reasons AS
(
SELECT
subs.SUBSCRIPTIONID,
canc.CANCELATIONREASON1 AS cancelationreason
FROM cancelations canc
JOIN Subscriptions subs
ON canc.SUBSCRIPTIONID = subs.SUBSCRIPTIONID
UNION
SELECT
subs.SUBSCRIPTIONID,
canc.CANCELATIONREASON2 AS cancelationreason
FROM cancelations canc
JOIN Subscriptions subs
ON canc.SUBSCRIPTIONID = subs.SUBSCRIPTIONID
UNION
SELECT
subs.SUBSCRIPTIONID,
canc.CANCELATIONREASON3 AS cancelationreason
FROM cancelations canc
JOIN Subscriptions subs
ON canc.SUBSCRIPTIONID = subs.SUBSCRIPTIONID
)
SELECT
    CAST(
```

```

        COUNT(CASE WHEN cancelationreason = 'Expensive'
        THEN subscriptionid END) AS FLOAT)
    / COUNT(DISTINCT subscriptionid) AS percent_expensive
FROM all_cancelation_reasons
)

SELECT
    CONCAT('The percentage of users who cancelled their subscriptions because it
was too expensive is ', GROUP_CONCAT(percent_expensive*100)) AS SUMMARY
FROM
    percent
GROUP BY
    percent_expensive

```

---

## SOLUTION PROCESS

---

- Nested CTE function: This initial CTE labeled **ALL\_CANCELATION\_REASONS** combines every user's various cancellation reasons by using the UNION function to un-pivot the columns in the **CANCELATIONS** table. Joining the **SUBSCRIPTIONS** table allows for further analysis in the future to shed light on other factors related to the cancelation such as length of subscription, number of users or revenue.
  - Case, Count and Cast functions: The CASE function identifies each record where **cancelationreason** is 'Expensive' and logs them using a COUNT function. In order to calculate this value as a percentage of all total cancelations, it is converted in a demical format, FLOAT, using the CAST function before dividing it by the distinct of number of subscribers in the **ALL\_CANCELATION\_REASONS** CTE.
  - Outer CTE & CONCAT functions: This CTE labeled **PERCENT** returns a percentage of all total cancelations. The GROUP CONCAT function returns the records of a queried column into a single row or output line. The purpose of the overall CONCAT function is to present a high level **SUMMARY** of the report's findings in layman's terms.
-

# OUTPUT

---

-----  
| SUBSCRIPTIONID | CANCELATIONREASON |

-----  
| 12622 | null |

| 12622 | Not useful |

| 12622 | Went to a competitor |

| 38499 | null |

| 38499 | Expensive |

| 44467 | Bad customer service |

| 44467 | Expensive |

| 44467 | Went to a competitor |

| 72882 | null |

| 72882 | Not useful |

| 72882 | Went to a competitor |

| 93330 | null |

| 93330 | Bad customer service |

| 93330 | Expensive |

| 99332 | null |

| 99332 | Not useful |

| 99332 | Went to a competitor |

-----  
| PERCENT\_EXPENSIVE |

-----  
| 0.5 |

---

| SUMMARY |

---

| The percentage of users who cancelled their subscriptions because it was too expensive is 50.0 |

---

---