L. Elaine Dazzio BSE Chapter 10: Project #2 Subscription Cancelation Analysis

⚡ **Project 2: Subscription Cancelation Analysis**

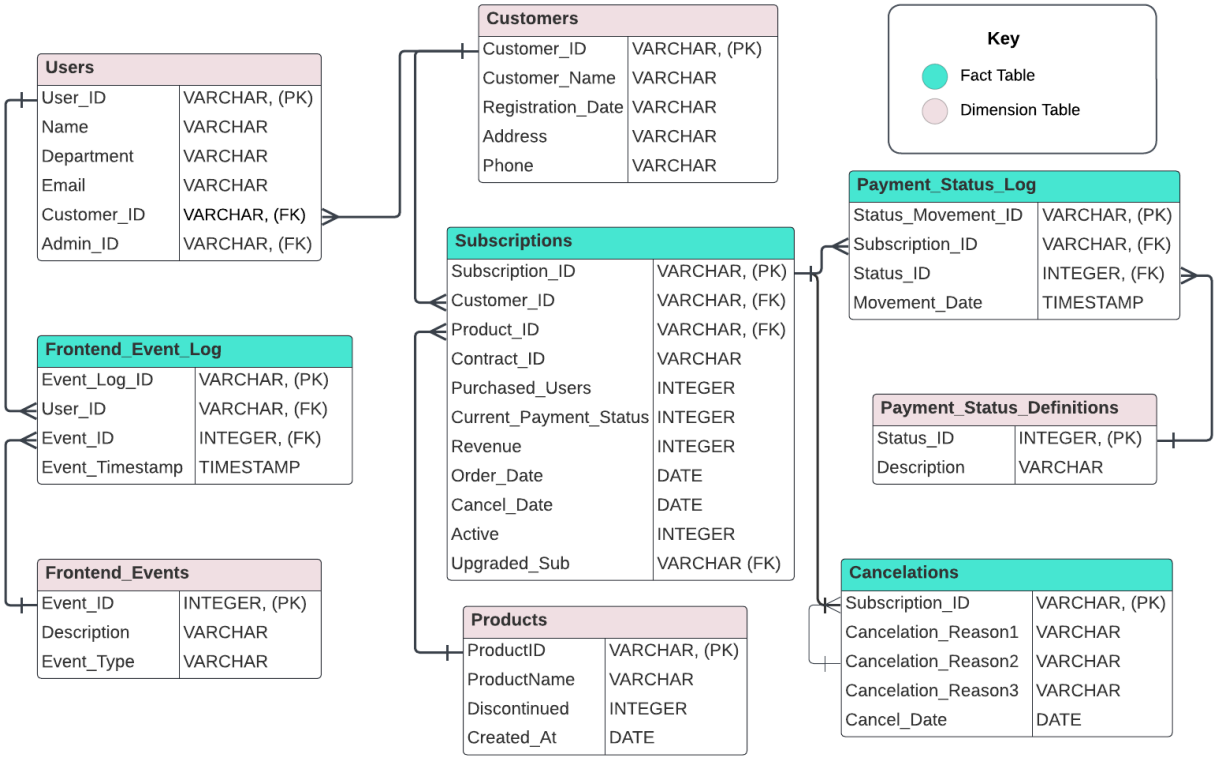
Case Study: Subscriptions Cancelation Analysis

You're having a lovely Monday morning getting a slow, quiet start to your day. It's 8am, and you're working from home today. You take your first sip of coffee, and it's heavenly... UNTIL! Your CEO loops you into an email with the leadership team in a panic. She says that churn has been much higher than expected this year which means that many customers have been leaving and canceling their subscriptions instead of renewing. The company is having a retention crisis and needs to figure out how to improve retention before the next board meeting— or the board is not going to be happy! You brainstorm a few quick ideas on how you can make the product stickier and keep customers around longer, but you know you can't just trust your gut! You have to look into the data first to see what insights you find.

You reach out to your customer support team first and ask them to pull qualitative data and testimonies from customers who have left. This will be useful to understand on a deeper level why customers are leaving and what their overall tone is when they leave. Then you work with the product manager to understand what the cancelation workflow looks like. Users first have to go to their subscription settings, click cancel, and then enter the cancelation flow. The cancelation flow requires them to select one reason why they are canceling, and then there are 2 additional selections that are optional. Each selection is a dropdown with the same preset values— so you're already thinking that this categorical data will be much easier to work with than text boxes that allow any text feedback to be typed into them. Qualitative text data can be the wild west due to the inconsistencies and inability to easily summarize insights.

You know you already want to look into these selections in the cancelations table, but you're also curious how many users actually select more than just the first required selection. Is it possible that this workflow could be really annoying and frustrating for users and they click random things to get through as quickly as possible? Or do we think users are providing accurate info and we can depend on most of these selections to accurately reflect the user's reason for canceling? You have a lot to dig into and can't wait to report your findings back to leadership!

Big SQL Energy Main Data Model



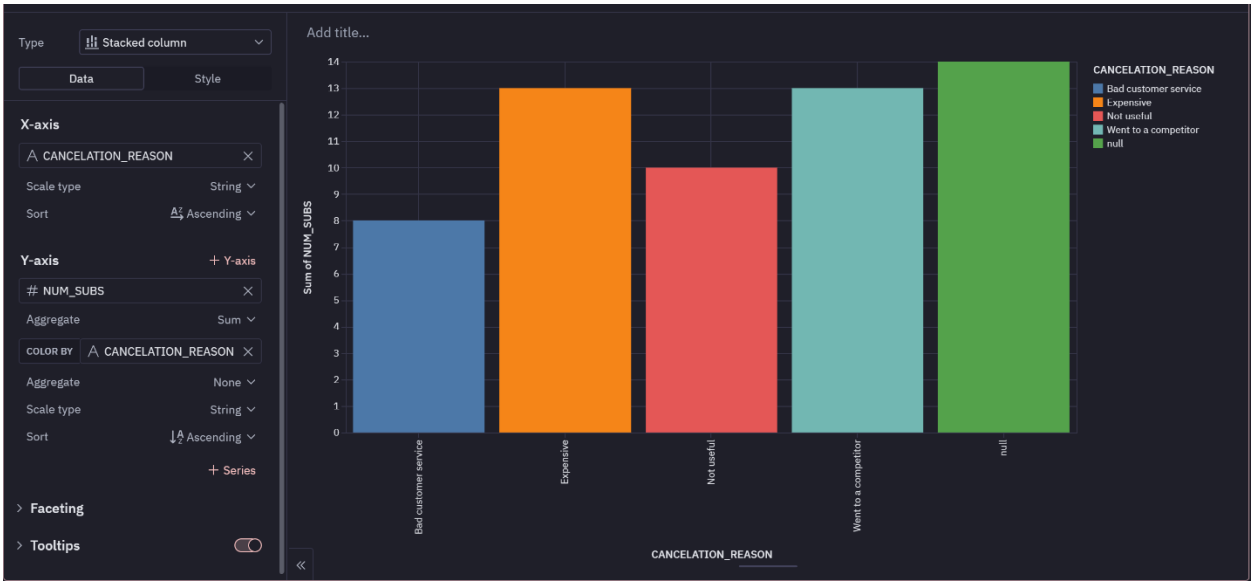
```
SELECT
    cancelation_reason1,
    COUNT(*) AS num_instances,
    COUNT(DISTINCT subscription_id) AS num_subs
FROM
    public.cancelations
GROUP BY
    1
;
```

	A CANCELATION_REASON3	# NUM_INSTANCES	# NUM_SUBS	+
0	None	14	14	
1	Bad customer service	5	5	
2	Not useful	2	2	
3	Expensive	1	1	

```
WITH cancels AS(
SELECT
    subscription_id,
    CASE WHEN cancelation_reason1 IS NOT NULL THEN 1 ELSE 0 END AS has_reason1,
    CASE WHEN cancelation_reason2 IS NOT NULL THEN 1 ELSE 0 END AS has_reason2,
    CASE WHEN cancelation_reason3 IS NOT NULL THEN 1 ELSE 0 END AS has_reason3,
    has_reason1 + has_reason2 + has_reason3 AS total_reasons,
    has_reason2 + has_reason3 AS additional_reasons
FROM
    public.cancelations
)

SELECT
    AVG(total_reasons) AS avg_total_per_sub,
    AVG(additional_reasons) AS avg_additional_per_sub
FROM
    cancels
;
```

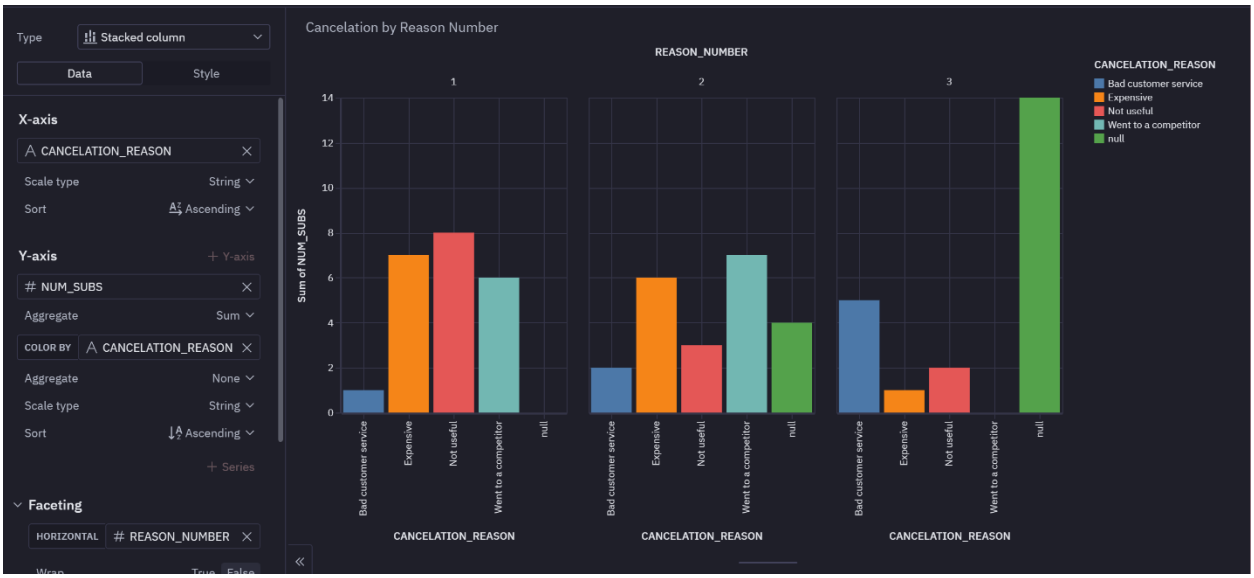
	# AVG_TOTAL_PER_SUB	# AVG_ADDITIONAL_PER_SUB	+
0	2.181818	1.181818	

```
WITH cancel_subs AS(
SELECT
    subscription_id,
    cancelation_reason1 as cancelation_reason
FROM
    public.cancelations
UNION
SELECT
    subscription_id,
    cancelation_reason2 as cancelation_reason
FROM
    public.cancelations
UNION
SELECT
    subscription_id,
    cancelation_reason3 as cancelation_reason
FROM
    public.cancelations
)
SELECT
   cancelation_reason,
   COUNT(subscription_id) AS num_subs
FROM
    cancel_subs
GROUP BY
    1
;
```



```
CREATE OR REPLACE VIEW junk.all_cancelation_reasons_ElaineD AS
SELECT
    subscription_id,
    cancel_date,
    cancelation_reason1 as cancelation_reason,
    1 AS reason_number
FROM
    public.cancelations

UNION

SELECT
    subscription_id,
    cancel_date,
    cancelation_reason2 as cancelation_reason,
    2 AS reason_number
FROM
    public.cancelations

UNION

SELECT
    subscription_id,
    cancel_date,
    cancelation_reason3 as cancelation_reason,
    3 AS reason_number
FROM
    public.cancelations
;
```

```
SELECT
   cancelation_reason,
   reason_number,
   COUNT(subscription_id) AS num_subs
FROM
    junk.all_cancelation_reasons_ElaineD
GROUP BY
    1, 2
;
```



```
WITH yearly AS(
SELECT
    DATE_TRUNC('year', cancel_date::date) AS cancel_year,
    cancelation_reason,
    COUNT(*) AS num_reason
FROM
    junk.all_cancelation_reasons_ElaineD
GROUP BY
    1, 2
)

SELECT
    cancel_year,
    cancelation_reason,
    num_reason,
    SUM(num_reason) OVER(PARTITION BY cancel_year) AS year_total,
    num_reason * 100 / year_total AS perc_reason_annual
FROM
    yearly
;
```