Payment Funnel Analysis

Business Problem:

The product manager has requested a payment funnel analysis from the analytics team; she/he wants to understand what the furthest point in the payment process users are getting to and where users are falling out of the process. She wants to have full visibility into each possible stage of the payment process from the user’s point of view.

Here’s the payment process a user goes through when signing up for a subscription:

1. The user opens the widget to initiate payment process.
2. The user types in credit card information.
3. The user clicks the submit button to complete their part of the payment process.
4. The product sends the data to the third-party payment processing company.
5. The payment company completes the transaction and reports back with “Complete”.

A close up of a document

Description automatically generatedpaymentstatuslog table

As subscriptions move through the statuses, the movements are logged in the status log table using the statusid. Users can go back and forth and move through statuses multiple times. E.g., A white paper with black text

Description automatically generated

To determine a subscription’s payment funnel stage, we want to consider its max statusid because it will show us the furthest point in the workflow that they successfully reached - regardless of whether they:

1. Completed the process,
2. Hit an error and started the process over,
3. Or hit an error, gave up, and left the workflow.

In addition to the max status reached, we also want to consider if the subscription is currently stuck in an error using the currentstatus column from the subscriptions table.

If a user reaches statusid = 3 but submits an incorrect card number, the transaction will be stopped, and they will see an error message. The user will then need to restart the process and use the correct information. This is considered a user error. If a user reached statusid = 4 and submits correct payment information, the data is sent to a third-party payment provider. If they are unable to process the data and complete the transaction due to an error on their end, it will produce and error message to the user. This is considered a vendor error.

We want to differentiate between these two different user fallout reasons in our funnel analysis, because they point to different business actions to fix them. Here’s a visualization of the logic needed.

A diagram of a payment funnel

Description automatically generated

Task:

Count the number of subscriptions in each payment funnel stage by incorporating the max status reached and current status per subscription. Use the paymentstatuslog table and subscriptions table.

A diagram of a data model

Description automatically generated

A white paper with black text

Description automatically generated

Code:

WITH max\_status\_reached AS (

SELECT

    subscriptionid,

    MAX(statusid) AS maxstatus

FROM paymentstatuslog

GROUP BY subscriptionid

)

,

paymentfunnelstages AS (

SELECT

    subs.subscriptionid,

    CASE

        WHEN maxstatus = 1 THEN 'PaymentWidgetOpened'

        WHEN maxstatus = 2 THEN 'PaymentEntered'

        WHEN maxstatus = 3 AND currentstatus = 0 THEN 'User Error with Payment Submission'

        WHEN maxstatus = 3 AND currentstatus != 0 THEN 'Payment Submitted'

        WHEN maxstatus = 4 AND currentstatus = 0 THEN 'Payment Processing Error with Vendor'

        WHEN maxstatus = 4 AND currentstatus != 0 THEN 'Payment Success'

        WHEN maxstatus = 5 THEN 'Complete'

        WHEN maxstatus IS NULL THEN 'User did not start payment process'

    END AS paymentfunnelstage

    FROM Subscriptions AS subs

    LEFT JOIN max\_status\_reached as m ON subs.subscriptionid = m.subscriptionid

)

SELECT

    paymentfunnelstage,

    COUNT(subscriptionid) AS subscriptions

FROM paymentfunnelstages

GROUP BY paymentfunnelstage;

Credits: Jess Ramos