Calculating Churn Rates

Analyze Data with SQL
Julia Macha
12.05.2023

This project involves conducting a data analysis using MySQL to measure the user churn rate for Codeflix, a streaming video startup. The data covers a four-month period from December 2016 to March 2017. The analysis will focus on two user segments and provide insights to help improve customer retention strategies.

SQL table ‘subscriptions’

id	the subscription id
subscription_start	the start date of the subscription
subscription_end	the end date of the subscription
segment	this identifies which segment the subscription owner belongs to

Project objective:

- The project's objective is to conduct an **analysis of subscription churn rates for a company** with the aim of assessing its customer retention performance and identifying opportunities for improvement. Another goal is to **compare churn rates between two segments of users** to gain insights into potential differences in their behavior and preferences. The project aims to provide actionable insights to improve the company's marketing strategy.

Creating ‘months’ CTE table

To **divide provided subscription dates into monthly buckets**, a CTE table called "months" is created based on the subscription data.

first_day	last_day
2017-01-01	2017-01-31
2017-02-01	2017-02-28
2017-03-01	2017-03-31

Cross joining months and subscriptions table

In order to perform a detailed analysis of churn rates, the **"months" table and the “subscription” data are cross-joined** to create a single table that includes all relevant data.

id	subscription_start	subscription_end	segment	first_day	last_day
1	2016-12-01	2017-02-01	87	2017-01-01	2017-01-31
1	2016-12-01	2017-02-01	87	2017-02-01	2017-02-28
1	2016-12-01	2017-02-01	87	2017-03-01	2017-03-31

Adding ‘status’ CTE to see subscription status

"Status" CTE table is created to **classify users by segment and subscription status (active or cancelled).** This enables a more detailed analysis of churn rates between the two user segments and can inform marketing strategies to improve customer retention.

id	month	is_active_87	is_active_30	is_canceled_87	is_canceled_30
1	2017-01-01	1	0	0	0
1	2017-02-01	0	0	1	0
1	2017-03-01	0	0	0	0
2	2017-01-01	1	0	1	0
2	2017-02-01	0	0	0	0
2	2017-03-01	0	0	0	0
3	2017-01-01	1	0	0	0
3	2017-02-01	1	0	0	0
3	2017-03-01	1	0	1	0
4	2017-01-01	1	0	0	0
4	2017-02-01	1	0	1	0
4	2017-03-01	0	0	0	0
5	2017-01-01	1	0	0	0
5	2017-02-01	1	0	0	0
5	2017-03-01	1	0	1	0
6	2017-01-01	1	0	1	0
6	2017-02-01	0	0	0	0
6	2017-03-01	0	0	0	0
7	2017-01-01	1	0	0	0
7	2017-02-01	1	0	1	0
7	2017-03-01	0	0	0	0
8	2017-01-01	1	0	0	0
8	2017-02-01	1	0	0	0
8	2017-03-01	1	0	1	0

Adding status_aggregate CTE table

To calculate the total number of active and cancelled subscriptions for each month, a "status_aggregate" CTE table is created by aggregating the data from the "status" CTE table. This allows for a more streamlined and efficient analysis of churn rates on a monthly basis.

sum_active_87	sum_active_30	sum_canceled_87	sum_canceled_30
278	291	70	22
462	518	148	38
531	716	258	84

Calculating the churn rates for the two segments over the three month period

By comparing the churn rates for the two segments, we can gain insights into which group may require further attention to improve retention.

month	churn_rate_87	churn_rate_30
2017-01-01	0.251798561151079	0.0756013745704467
2017-02-01	0.32034632034632	0.0733590733590734
2017-03-01	0.485875706214689	0.11731843575419

To see full SQL code

Click to see SQL code

Conclusions:

- The analysis of churn rates revealed that the percentage of subscribers who cancelled their subscriptions in January and February was significantly higher for segment 30 (76%, 73%) than for segment 87 (25%, 32%). However, this trend started to change in March, with the churn rate for segment 87 increasing to 49% while the churn rate for segment 30 decreased to only 12%. These findings highlight the importance of regularly analyzing churn rates to identify patterns and opportunities for improving customer retention strategies.