



User Churn Analysis with SQL

By Mark Bellmann

Table of Contents

- 1. Introduction
- 2. Analysis
 - 2.1. Range of data
 - 2.2. Churn trends
- 3. Conclusion

Introduction

Codeflix, a streaming video startup, is interested in measuring their user churn rate. Our task is to compare churn rates between different user segments. Questions are:

- For which months we can calculate a churn rate?
- What segments exist?
- What is the overall churn rate trend since the start of the data?
- How do the different segments compare to each other?

2.1 Range of data

2.1 Range of data

The dataset provided consists of one SQL table, subscriptions. Within the table are 4 columns, id, subscription_start, subscription_end and segment respectively. In “1.” we check for the MIN and MAX values for the dates, to get to know the range we can use. In “2.” we check for the different segments.

MIN(subscripti on_start)	MAX(subscription_s tart)	MIN(subscriptio n_end)	MAX(subscriptio n_end)
2016-12-01	2017-03-30	2017-01-01	2017-03-31

Take aways:

- The data ranges from 2016-12-01 to 2017-03-31.
- here are two segments, 87 and 30.

```
-- 1. Range of data --  
SELECT MIN(subscription_start),  
       MAX(subscription_start),  
       MIN(subscription_end),  
       MAX(subscription_end)  
FROM subscriptions;
```

```
-- 2. Segments --
```

```
SELECT DISTINCT segment  
FROM subscriptions;
```

2.2 Churn trend

2.2 Churn trend

This query shows the churn rate for each segment over the 3 months. We can see that segment 87 is constantly performing significantly lower than segment 30.

month	segment	sum_active	sum_canceled	churn_rate
2017-01-01	30	291	22	7.56
2017-01-01	87	278	70	25.18
2017-02-01	30	518	38	7.34
2017-02-01	87	462	148	32.03
2017-03-01	30	716	84	11.73
2017-03-01	87	531	258	48.59

```
WITH months AS
(
  SELECT
    '2017-01-01' as first_day,
    '2017-01-31' as last_day
  UNION
  SELECT
    '2017-02-01' as first_day,
    '2017-02-28' as last_day
  UNION
  SELECT
    '2017-03-01' as first_day,
    '2017-03-31' as last_day
),
cross_join AS
(
  SELECT *
  FROM subscriptions
  CROSS JOIN months
),
status AS
(
  SELECT id, segment,
    first_day as month,
    CASE WHEN (subscription_start < first_day)
      AND (subscription_end > first_day
        OR subscription_end IS NULL)
      THEN 1
      ELSE 0
    END AS is_active,
    CASE
      WHEN (subscription_end BETWEEN first_day AND last_day)
      THEN 1
      ELSE 0
    END as is_canceled
  FROM cross_join
)
SELECT month, segment,
  SUM(is_active) AS sum_active,
  SUM(is_canceled) AS sum_canceled
  ROUND((100.0 * SUM(is_canceled) / SUM(is_active)), 2) as churn_rate
FROM status
GROUP BY month, segment;
```

3. Conclusion

- Segment 30 is performing significantly better, indicating a potential future focus.
- At the same time, we can investigate into potential causes for the higher churn rate for segment 87.
- March was the worst month for both segments - what might be the causes here?