



Calculating Churn Rate

Learn SQL from Scratch – Codeflix Project

Table of Contents

1. Get familiar with Codeflix
 - a. How many different segments are there?
 - b. Which months will you be able to calculate churn rate?
2. What is the overall churn rate by month?
 - a. Compare the churn rates between segments
 - b. Which segment has a lower churn rate?
3. How would you modify this code to support a large number of segments?

1. Get familiar with Codeflix

1.1 Codeflix

Codeflix is a software as a service company specializing in streaming video content.

- Dataset provided is a table named “subscriptions” and the schema for this table is below with a couple examples for each column.
- ‘id’ is a unique identifier assigned to a customer
- ‘subscription_start’ is the date the customer subscribed
- ‘subscription_end’ is the date the customer canceled
- ‘segment’ is the channel used to acquire the customer
- The dataset range is from 2016-12-01 to 2017-03-30 for subscriber start dates
- Churn rate can be calculated for January, February, and March of 2017 but will not be calculated for December 2016 due to minimum 31 day subscription from start date.
- There are 2 segments in this data set, 87 and 30.

```
1  /*Select everything from subscriptions table,
2  limit display to 100*/
3  SELECT *
4  FROM subscriptions
5  LIMIT 100;
6
7  --Get the range of months of data provided
8  SELECT MIN(subscription_start),
9  MAX(subscription_start)
10 FROM subscriptions;
11
12 --How many different segments are there?
13 SELECT DISTINCT(segment)
14 FROM subscriptions;
```

id	subscription_start	subscription_end	segment
1	2016-12-01	2017-02-01	87
13	2016-12-01	Ø	30

MIN(subscription_start)	MAX(subscription_start)
2016-12-01	2017-03-30

segment
87
30

2. What is the overall churn rate by month?

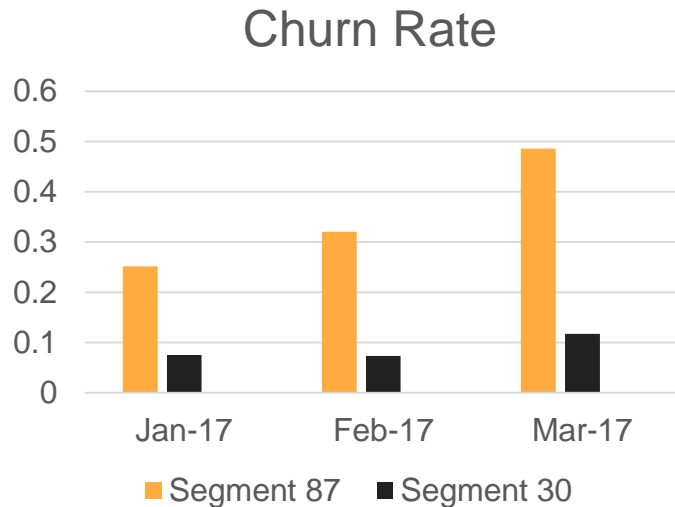
2.1 The overall churn rate by month

Churn rate for each month is listed below in the table.

- Segment 30 has a lower churn rate than segment 87
- An event in March that caused the churn rate to increase 4% for segment 30 and 16% for segment 87 and should be looked at in more depth

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

```
89  /*Calculate churn rate using SUM values from
90  temporary
91  status_aggregate table*/
92  SELECT
93  month,
94  1.0 * sum_canceled_87/sum_active_87 AS
churn_rate_87,
95  1.0 * sum_canceled_30/sum_active_30 AS
churn_rate_30
96  FROM status_aggregate
97  GROUP BY month;
```



3. How would you modify this code to support a large number of segments

3.1 Modified code for large number of segments.

- The temporary table labeled 'status' has the segments hardcoded. Add a column called segment to this table.
- Remove the hardcoded segments in the CASE statements of the 'status' table and relabel to just is_active and is_canceled.
- Update the status_aggregate table to include the segment column and the SUM to reference the is_active and is_canceled data, group by month and segment
- Update the query for churn rate to add the segment column and to reference the new column names from the status_aggregate table. Group data by segment and month or by month and segment, depending on how data would like to be viewed.

GROUP BY segment, month

```
75  /*Calculate churn rate using SUM values from temporary
76  status_aggregate table*/
77  SELECT
78  month, |
79  segment,
80  1.0 * sum_canceled/sum_active AS churn_rate
81  FROM status_aggregate
82  GROUP BY segment, month;
83
```

Output display

month	segment	churn_rate
2017-01-01	30	0.0756013745704467
2017-02-01	30	0.0733590733590734
2017-03-01	30	0.11731843575419
2017-01-01	87	0.251798561151079
2017-02-01	87	0.32034632034632
2017-03-01	87	0.485875706214689

GROUP BY month, segment

```
75  /*Calculate churn rate using SUM values from temporary
76  status_aggregate table*/
77  SELECT
78  month,
79  segment,
80  1.0 * sum_canceled/sum_active AS churn_rate
81  FROM status_aggregate
82  GROUP BY month, segment;
```

Output display

month	segment	churn_rate
2017-01-01	30	0.0756013745704467
2017-01-01	87	0.251798561151079
2017-02-01	30	0.0733590733590734
2017-02-01	87	0.32034632034632
2017-03-01	30	0.11731843575419
2017-03-01	87	0.485875706214689