

Learn SQL from Scratch

Capstone Project

Danielle Furlani
April 2019



TABLE OF CONTENTS

OPTION 2: Churn Rates with Codeflix

Analyze churn for two groups of Codeflix users and recommend which segment represents a better long-term user base.

Codeflix, a streaming video startup, is interested in measuring their user churn rate

The marketing department is particularly interested in how the churn compares between two segments of users.

1. Get familiar with the company.
 - What segments of users exist?
 - How many months has the company been operating? Which months do you have enough information to calculate a churn rate?
2. What is the overall churn trend since the company started?
3. Compare the churn rates between user segments.
 - Which segment of users should the company focus on expanding?

1) Get familiar with the company, Codeflix

What segments of users exist?

1)

Take a look at the first 100 rows of data in the subscriptions table

- `SELECT *` allows me to view data for all the columns in the subscriptions table
- `LIMIT` is used to look at the first 100 rows of data

```
1 SELECT *
2 FROM subscriptions
3 LIMIT 100;
```

| Query Results | | | |
|---------------|--------------------|------------------|---------|
| id | subscription_start | subscription_end | segment |
| 1 | 2016-12-01 | 2017-02-01 | 87 |
| 2 | 2016-12-01 | 2017-01-24 | 87 |
| 3 | 2016-12-01 | 2017-03-07 | 87 |
| 4 | 2016-12-01 | 2017-02-12 | 87 |
| 5 | 2016-12-01 | 2017-03-09 | 87 |
| 6 | 2016-12-01 | 2017-01-19 | 87 |
| 7 | 2016-12-01 | 2017-02-03 | 87 |
| 8 | 2016-12-01 | 2017-03-02 | 87 |
| 9 | 2016-12-01 | 2017-02-17 | 87 |
| 10 | 2016-12-01 | 2017-01-01 | 87 |
| 11 | 2016-12-01 | 2017-01-17 | 87 |
| 12 | 2016-12-01 | 2017-02-07 | 87 |
| 13 | 2016-12-01 | ∅ | 30 |

How many different segments do you see?

- From my first query results I saw two different segments, 87 and 30
- `SELECT DISTINCT segments` helped me determine that 87 and 30 were the only segments in the table

```
1 SELECT DISTINCT segment
2 FROM subscriptions;
```

| Query Results | |
|---------------|--|
| segment | |
| 87 | |
| 30 | |

1) Get familiar with the company, Codeflix (cont.)

How many months has the company been operating?

2)

Determine the range of months of data provided

- `MIN()` allows me to see the earliest subscription start date
- `MAX()` allows me to see the latest subscription start date
- `AS` allows me to rename each column

```
1 SELECT MIN(subscription_start) AS min,  
2    MAX(subscription_start) AS max  
3 FROM subscriptions;
```

From this query we can determine that the company has been operating for 4 months

The range of months for our date is between December 1, 2016 and March 30, 2017

Which months will you be able to calculate churn for?

I can calculate the churn rate for the first three months of 2017

We can't calculate for Dec 2016, because there are no subscription_end values

| Query Results | |
|-------------------------|------------|
| min | max |
| 2016-12-01 | 2017-03-30 |
| Database Schema | |
| subscriptions 2000 rows | |
| id | INTEGER |
| subscription_start | TEXT |
| subscription_end | TEXT |
| segment | INTEGER |

2) What is the overall churn trend since the company started?

3)

You'll be calculating the churn rate for both segments over the first 3 months of 2017. To get started, create a temporary table of *months*

- **WITH** allows me to create a temporary table, *months*, to be used in the churn calculation
- **UNION** allows me to stack one dataset on top of another

Calculating churn rate

- **SELECT** the first day of each month to be used as a cutoff for subscribers
- **SELECT** the last day of each month to be used as the cutoff for cancellations

```
1  WITH months AS
2  (SELECT '2017-01-01' AS first_day,
3       '2017-01-31' AS last_day
4  UNION
5   SELECT '2017-02-01' AS first_day,
6        '2017-02-28' AS last_day
7  UNION
8   SELECT '2017-03-01' AS first_day,
9        '2017-03-31' AS last_day)
10 SELECT *
11 FROM months;
```

Query Results

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

2) What is the overall churn trend since the company started? (cont.)

4)

Create a temporary table, *cross_join*, from *subscriptions* and your *months*

AS allows me to name this new joined table, *cross_join*. This query was placed directly after the query that created the *months* table

I joined all columns of the *subscriptions* table with all columns of the temporary *months* table using **CROSS JOIN**

CROSS JOIN simply combines columns from multiple tables together, it does not join data

This step is necessary to calculate churn because we can now view which segment of users correspond with each subscriptions first and last days. This will help to determine which segment of users represent a better long-term user base.

```
1 WITH months AS
2 (SELECT '2017-01-01' AS first_day,
3  '2017-01-31' AS last_day
4  UNION
5  SELECT '2017-02-01' AS first_day,
6  '2017-02-28' AS last_day
7  UNION
8  SELECT '2017-03-01' AS first_day,
9  '2017-03-31' AS last_day),
10 cross_join AS
11 (SELECT *
12  FROM subscriptions
13  CROSS JOIN months)
14 SELECT *
15 FROM cross_join
16 LIMIT 50;
```

| Query Results | | | | | |
|---------------|--------------------|------------------|---------|------------|------------|
| 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 |
| 2 | 2016-12-01 | 2017-01-24 | 87 | 2017-01-01 | 2017-01-31 |
| 2 | 2016-12-01 | 2017-01-24 | 87 | 2017-02-01 | 2017-02-28 |
| 2 | 2016-12-01 | 2017-01-24 | 87 | 2017-03-01 | 2017-03-31 |
| 3 | 2016-12-01 | 2017-03-07 | 87 | 2017-01-01 | 2017-01-31 |
| 3 | 2016-12-01 | 2017-03-07 | 87 | 2017-02-01 | 2017-02-28 |
| 3 | 2016-12-01 | 2017-03-07 | 87 | 2017-03-01 | 2017-03-31 |
| 4 | 2016-12-01 | 2017-02-12 | 87 | 2017-01-01 | 2017-01-31 |
| 4 | 2016-12-01 | 2017-02-12 | 87 | 2017-02-01 | 2017-02-28 |
| 4 | 2016-12-01 | 2017-02-12 | 87 | 2017-03-01 | 2017-03-31 |
| 5 | 2016-12-01 | 2017-03-09 | 87 | 2017-01-01 | 2017-01-31 |
| 5 | 2016-12-01 | 2017-03-09 | 87 | 2017-02-01 | 2017-02-28 |
| 5 | 2016-12-01 | 2017-03-09 | 87 | 2017-03-01 | 2017-03-31 |
| 6 | 2016-12-01 | 2017-01-19 | 87 | 2017-01-01 | 2017-01-31 |
| 6 | 2016-12-01 | 2017-01-19 | 87 | 2017-02-01 | 2017-02-28 |
| 6 | 2016-12-01 | 2017-01-19 | 87 | 2017-03-01 | 2017-03-31 |
| 7 | 2016-12-01 | 2017-02-03 | 87 | 2017-01-01 | 2017-01-31 |
| 7 | 2016-12-01 | 2017-02-03 | 87 | 2017-02-01 | 2017-02-28 |
| 7 | 2016-12-01 | 2017-02-03 | 87 | 2017-03-01 | 2017-03-31 |
| 8 | 2016-12-01 | 2017-03-02 | 87 | 2017-01-01 | 2017-01-31 |
| 8 | 2016-12-01 | 2017-03-02 | 87 | 2017-02-01 | 2017-02-28 |
| 8 | 2016-12-01 | 2017-03-02 | 87 | 2017-03-01 | 2017-03-31 |
| 9 | 2016-12-01 | 2017-02-17 | 87 | 2017-01-01 | 2017-01-31 |
| 9 | 2016-12-01 | 2017-02-17 | 87 | 2017-02-01 | 2017-02-28 |

2) What is the overall churn trend since the company started? (cont.)

5) Create a temporary table, *status*, from *cross_join*. This table should contain:

- id selected from *cross_join*
- month as an alias of *first_day*
- *is_active_87* and *is_active_30* created using a CASE WHEN to find any users from each segment who existed prior to the beginning of the month. This is 1 if true and 0 otherwise.

The results show us the months a user had an active subscription and which segment the user is from

1 represents a true statement in a CASE WHEN *subscription_start* is prior to *first_day* AND *subscription_end* is after *first_day* OR *subscription_end* IS NULL

0 represents if the CASE WHEN statement is false

This data will be used as the denominator in our churn calculation for each segment

```
12 FROM subscriptions
13 CROSS JOIN months),
14 status AS
15 (SELECT id, first_day AS month,
16 CASE
17 WHEN (segment = 87)
18 AND (subscription_start < first_day)
19 AND (subscription_end > first_day
20 OR subscription_end IS NULL) THEN 1
21 ELSE 0
22 END AS is_active_87,
23 CASE
24 WHEN (segment = 30)
25 AND (subscription_start < first_day)
26 AND (subscription_end > first_day
27 OR subscription_end IS NULL) THEN 1
28 ELSE 0
29 END AS is_active_30
30 FROM cross_join)
31 SELECT *
32 FROM status
33 LIMIT 50;
```

| Query Results | | | |
|---------------|------------|--------------|--------------|
| id | month | is_active_87 | is_active_30 |
| 1 | 2017-01-01 | 1 | 0 |
| 1 | 2017-02-01 | 0 | 0 |
| 1 | 2017-03-01 | 0 | 0 |
| 2 | 2017-01-01 | 1 | 0 |
| 2 | 2017-02-01 | 0 | 0 |
| 2 | 2017-03-01 | 0 | 0 |
| 3 | 2017-01-01 | 1 | 0 |
| 3 | 2017-02-01 | 1 | 0 |
| 3 | 2017-03-01 | 1 | 0 |
| 4 | 2017-01-01 | 1 | 0 |
| 4 | 2017-02-01 | 1 | 0 |
| 4 | 2017-03-01 | 0 | 0 |
| 5 | 2017-01-01 | 1 | 0 |
| 5 | 2017-02-01 | 1 | 0 |
| 5 | 2017-03-01 | 1 | 0 |
| 6 | 2017-01-01 | 1 | 0 |
| 6 | 2017-02-01 | 0 | 0 |
| 6 | 2017-03-01 | 0 | 0 |
| 7 | 2017-01-01 | 1 | 0 |
| 7 | 2017-02-01 | 1 | 0 |
| 7 | 2017-03-01 | 0 | 0 |

2) What is the overall churn trend since the company started? (cont.)

6)

Add an `is_canceled_87` and an `is_canceled_30` column to the status temporary table. This should be 1 if the subscription is canceled during the month and 0 otherwise.

A `CASE WHEN` statement can be used to determine whether a subscription was cancelled and in what month for each segment

The results now show us the months a user had an active subscription, if they cancelled their subscription and when, as well as which segment the user is from

For the new `is_canceled` columns, 1 represents a true statement in a `CASE WHEN` `subscription_end` is `BETWEEN first_day` `AND last_day`
0 represents if the `CASE WHEN` statement is false

This data will be used as the numerator in our churn calculation for each segment

```
28     OR subscription_end IS NULL)
29   THEN 1
30   ELSE 0
31 END AS is_active_30,
32 CASE
33   WHEN segment = 87
34   AND (subscription_end BETWEEN first_day
35       AND last_day) THEN 1
36   ELSE 0 END AS is_canceled_87,
37 CASE
38   WHEN segment = 30
39   AND (subscription_end BETWEEN first_day
40       AND last_day) THEN 1
41   ELSE 0
42 END as is_canceled_30
43 FROM cross_join)
44 SELECT *
45 FROM status
46 limit 50;
```

| Query Results | | | | | |
|---------------|------------|--------------|--------------|----------------|----------------|
| 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 |

2) What is the overall churn trend since the company started? (cont.)

7)

Create a status_aggregate temporary table that is a SUM of the active and canceled subscriptions for each segment, for each month.

The resulting columns should be:

sum_active_87

sum_active_30

sum_canceled_87

sum_canceled_30

SUM()

- SUM () is an aggregate function, meaning that it can perform calculations on multiple rows
- With each calculation we have found the sum of the values in each column of our status table

Calculating churn rate

- GROUP BY month allows me to find the SUM() for each month
- We now have our numerators and denominators to calculate the churn for each segment

```
31 END AS is_active_30,  
32 CASE  
33   WHEN segment = 87  
34     AND (subscription_end BETWEEN first_day  
35         AND last_day) THEN 1  
36     ELSE 0 END AS is_canceled_87,  
37 CASE  
38   WHEN segment = 30  
39     AND (subscription_end BETWEEN first_day  
40         AND last_day) THEN 1  
41     ELSE 0  
42 END as is_canceled_30  
43 FROM cross_join),  
44 status_aggregate AS  
45 (SELECT month,  
46   SUM(is_active_87) AS sum_active_87,  
47   SUM(is_active_30) AS sum_active_30,  
48   SUM(is_canceled_87) AS sum_canceled_87,  
49   SUM(is_canceled_30) AS sum_canceled_30  
50 FROM status  
51 GROUP BY month)  
52 SELECT *  
53 FROM status_aggregate  
54 limit 100;
```

Query Results

| month | sum_active_87 | sum_active_30 | sum_canceled_87 | sum_canceled_30 |
|------------|---------------|---------------|-----------------|-----------------|
| 2017-01-01 | 278 | 291 | 70 | 22 |
| 2017-02-01 | 462 | 518 | 148 | 38 |
| 2017-03-01 | 531 | 716 | 258 | 84 |

3) Compare the churn rates between user segments

8)

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

To calculate churn we must divide the number of cancellations for each month by the number of active subscribers at the beginning of each month and multiply by 1 to force a float result instead of an integer.

We must do this for each segment

Segment 87

$1 \times 70/278$ = January 2017 churn

$1 \times 148/462$ = February 2017 churn

$1 \times 258/531$ = March 2017 churn

Segment 30

$1 \times 22/291$ = January 2017 churn

$1 \times 38/518$ = February 2017 churn

$1 \times 84/716$ = March 2017 churn

```
35         AND last_day) THEN 1
36     ELSE 0 END AS is_canceled_87,
37 CASE
38     WHEN segment = 30
39     AND (subscription_end BETWEEN first_day
40         AND last_day) THEN 1
41     ELSE 0
42     END as is_canceled_30
43 FROM cross_join),
44 status_aggregate AS
45 (SELECT month,
46     SUM(is_active_87) AS sum_active_87,
47     SUM(is_active_30) AS sum_active_30,
48     SUM(is_canceled_87) AS sum_canceled_87,
49     SUM(is_canceled_30) AS sum_canceled_30
50 FROM status
51 GROUP BY month)
52 SELECT month,
53     1.0 * sum_canceled_87/sum_active_87 AS churn_87,
54     1.0 * sum_canceled_30/sum_active_30 AS churn_30
55 FROM status_aggregate;
```

3) Compare the churn rates between user segments (cont.)

Which segment of users should the company focus on expanding?

8)

Which segment has a lower churn rate?

We've found our churn rates for each segment over the course of 3 months.

January 2017: 25% for segment 87 & 7% for segment 30

February 2017: 32% for segment 87 & 7% for segment 30

March 2017: 48% for segment 87 & 11% for segment 30

By comparing segment 87 and 30 for each month, we see that the churn is significantly lower for segment 30 for each month

After analyzing the churn for the two groups of Codeflix users, I would highly recommend that segment 30 represents a better long-term user base

| Query Results | | |
|---------------|-------------------|--------------------|
| month | churn_87 | churn_30 |
| 2017-01-01 | 0.251798561151079 | 0.0756013745704467 |
| 2017-02-01 | 0.32034632034632 | 0.0733590733590734 |
| 2017-03-01 | 0.485875706214689 | 0.11731843575419 |