



Calculate Churn Rates with SQL

Learn SQL from Scratch

Anna Rolicka 10.09.2018

Table of Contents

1. Get familiar with Codeflix:

- 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 Codeflix

➤ What segments of users exist?

From the data obtained, it is clearly visible that there are two segments of users ('30' and '87').

```
1 -- Take a look at the first 100 rows of data and check
2 -- how many different segments there are.
3 SELECT * FROM subscriptions
4 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	0	30
14	2016-12-01	2017-03-07	30
15	2016-12-01	2017-02-22	30
16	2016-12-01	0	30
17	2016-12-01	0	30
18	2016-12-02	2017-01-29	87
19	2016-12-02	2017-01-13	87
20	2016-12-02	2017-01-15	87
21	2016-12-02	2017-01-15	87
22	2016-12-02	2017-01-24	87
23	2016-12-02	2017-01-14	87
24	2016-12-02	2017-01-18	87
25	2016-12-02	2017-02-24	87
26	2016-12-02	2017-01-18	87
27	2016-12-02	2017-01-11	87
28	2016-12-02	2017-03-30	30
29	2016-12-02	2017-02-11	30

1. Get familiar with Codeflix

How many months has the company been operating? Which months do you have enough information to calculate a churn rate?

The company was operating for 4 months, since 2016-12-01 till 2017-03-30, that is the coverage we have the data for. We can calculate a churn rate for January, February and March of 2017, but it's not possible to calculate it for December 2016 since there are no subscription_end values yet for that month.

```
1  /* 1.How many months has the company been operating?
2  Determine the subscription start date*/
3  SELECT MIN(subscription_start)
4  FROM subscriptions;
5
6  -- Determine the subscription end date
7  SELECT MAX(subscription_start)
8  FROM subscriptions;
9
10
```

MIN(subscription_start)
2016-12-01

MAX(subscription_start)
2017-03-30

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

```
1  -- creating a temporary table of months
2  WITH months AS
3      (SELECT
4          '2017-01-01' AS first_day,
5          '2017-01-31' AS last_day
6      UNION
7      SELECT
8          '2017-02-01' AS first_day,
9          '2017-02-28' AS last_day
10     UNION
11     SELECT
12         '2017-03-01' AS first_day,
13         '2017-03-31' AS last_day),
14
15  -- a temporary table, cross_join of subscriptions and months
16  cross_join AS
17      (SELECT *
18       FROM subscriptions
19       CROSS JOIN months),
20
```

Churn rate is a measure of the number of customers who leave a company during a given period.

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

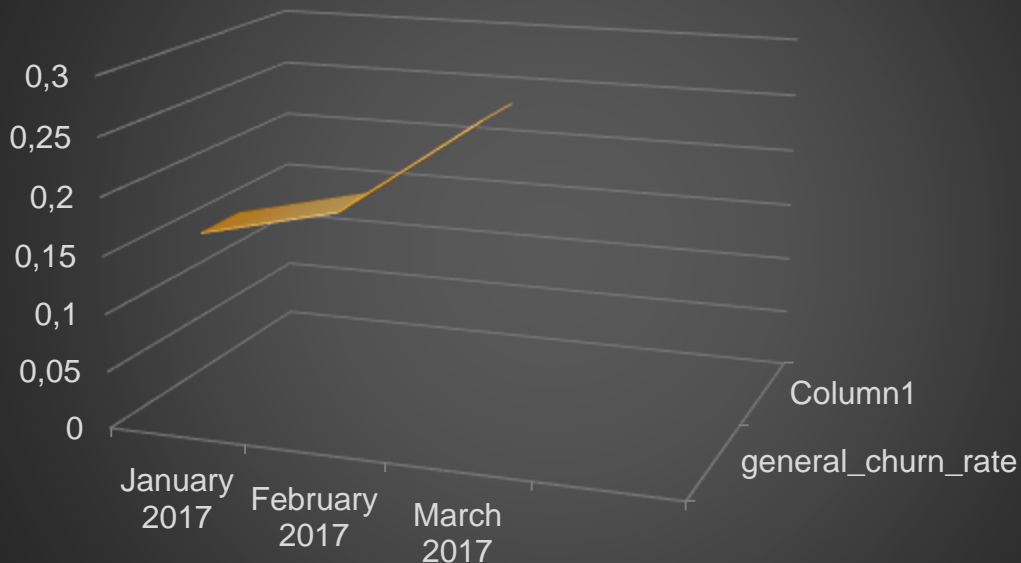
```
21 -- create a temporary table status from cross_join
22 -- where active and canceled customers are chosen
23 status AS
24     (SELECT id, first_day AS month,
25      CASE
26        WHEN (subscription_start < first_day)
27         AND (subscription_end > first_day
28              OR subscription_end IS NULL)
29        THEN 1
30        ELSE 0
31      END AS is_active,
32      CASE
33        WHEN subscription_end BETWEEN first_day AND last_day
34        THEN 1
35        ELSE 0
36      END AS is_canceled
37     FROM cross_join),
38 -- create a temporary table with sums of active and canceled users
39 -- are calculated and grouped by month.
40 status_aggregate AS
41     (SELECT
42      month,
43      SUM(is_active) as sum_active,
44      SUM(is_canceled) as sum_canceled
45     FROM status
46     GROUP BY month)
47
48 -- calculation of churn rate overall trend
49 SELECT month,
50        (1.0* status_aggregate.sum_canceled/
51         status_aggregate.sum_active) AS general_churn_rate
52 FROM status_aggregate;
```

month	general_churn_rate
2017-01-01	0.161687170474517
2017-02-01	0.189795918367347
2017-03-01	0.274258219727346

The overall churn rate has been increasing, from 0.16 in January to as high as 0.27 in March.

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

Overall churn rate



month	general_churn_rate
2017-01-01	0.161687170474517
2017-02-01	0.189795918367347
2017-03-01	0.274258219727346

Clearly, the overall churn rate has been increasing, especially in the month of March it has risen from 18 % to 27 % which is bad news for the company.

3. Compare the churn rates between user segments.

There are two segments of users that we need to compare: 30 and 87.

```
1  -- create a temporary table of months for January, February and March
2  WITH months AS
3      (SELECT
4          '2017-01-01' AS first_day,
5          '2017-01-31' AS last_day
6      UNION
7      SELECT
8          '2017-02-01' AS first_day,
9          '2017-02-28' AS last_day
10     UNION
11     SELECT
12         '2017-03-01' AS first_day,
13         '2017-03-31' AS last_day),
14
15  -- a temporary table, cross_join of subscriptions and months
16  cross_join AS
17      (SELECT *
18         FROM subscriptions
19         CROSS JOIN months),
```

```
21  /* create a temporary table status from cross_join
22     where active and canceled customers are chosen
23     for segments 30 and 87 separately
24     */
25
26  status AS
27      (SELECT id, first_day AS month,
28         CASE
29             WHEN (subscription_start < first_day)
30                  AND (subscription_end > first_day
31                      AND (segment == '87'))
32                  OR subscription_end IS NULL)
33                  AND (segment == '87')
34             THEN 1
35             ELSE 0
36         END AS is_active_87,
37         CASE
38             WHEN (subscription_start < first_day)
39                  AND (subscription_end > first_day
40                      AND (segment == '30')
41                      OR subscription_end IS NULL)
42                  AND (segment == '30')
43             THEN 1
44             ELSE 0
45         END AS is_active_30,
46         CASE
47             WHEN subscription_end BETWEEN first_day AND last_day
48                  AND (segment == '87')
49             THEN 1
50             ELSE 0
51         END AS is_canceled_87,
52         CASE
53             WHEN subscription_end BETWEEN first_day AND last_day
54                  AND (segment == '30')
55             THEN 1
56             ELSE 0
57         END AS is_canceled_30
58     FROM cross_join),
```


3. Compare the churn rates between user segments.

```
60 -- SELECT a temporary table status_aggregate calculating sums of
61 -- customers from both segments who were active and who cancelled
62 -- in a given month
63 status_aggregate AS
64 (SELECT
65     month,
66     SUM(is_active_87) as sum_active_87,
67     SUM(is_active_30) as sum_active_30,
68     SUM(is_canceled_87) as sum_canceled_87,
69     SUM(is_canceled_30) as sum_canceled_30
70 FROM status
71 GROUP BY month)
72
73 -- calculate the final churn rates for both segments of customers
74 -- for every month.
75 SELECT month,
76     (1.0* status_aggregate.sum_canceled_87/
77     status_aggregate.sum_active_87) AS churn_rate_87,
78     (1.0* status_aggregate.sum_canceled_30/
79     status_aggregate.sum_active_30) AS churn_rate_30
80 FROM status_aggregate;
```

Results:

month	churn_rate_87	churn_rate_30
2017-01-01	0.251798561151 079	0.0756013745704 467
2017-02-01	0.320346320346 32	0.0733590733590 734
2017-03-01	0.485875706214 689	0.1173184357541 9

3. Compare the churn rates between user segments. Which segment of users should the company focus on expanding?

Results:

The company should focus on expanding the 30 segment, which has a much lower churn rate overall and its churn rate has been increasing very slightly, from 7 % in January to 11% in March.

