Calculating Churn Rates

LEARN SQL FROM SCRATCH

TINA LUMI
23-07-2018

TABLE OF CONTENTS

GET FAMILIAR WITH THE COMPANY

1. *Get familiar with the company*

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

/ *What segments of users exist?*

First let's take a look at the first 100 rows of the subscriptions table:

```
SELECT *
FROM subscriptions
LIMIT 100;
```

The are four columns in the table: id, subscription_start, subscription_end and segment. This data is enough to perform churn analysis. Let's write additional queries to get more information about company operations

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	Ø	30
14	2016-12-01	2017-03-07	30
15	2016-12-01	2017-02-22	30
16	2016-12-01	Ø	30
17	2016-12-01	Ø	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
...	...	...	...
99	2016-12-06	Ø	30
100	2016-12-06	2017-03-11	30

GET FAMILIAR WITH THE COMPANY

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

```
SELECT
MIN(subscription_start) AS first_signup
,MAX(subscription_end) AS last_cancellation
FROM subscriptions;
```

first_signup	last_cancellation
2016-12-01	2017-03-31

The company started operating on 2016-12-01, last cancellation was on 2017-03-31. According to the table from previous query some subscriptions have not been cancelled but we do not have any data past March, so with absolute certainty we can say that the company has been operating for 4 months: December, January, February and March

Taking in consideration minimum subscription of 31 days there could have been no cancelations in December, hence we can provide churn analysis only for 3 months - January, February and March

crurn = cancelations during a period of time / active subscribers

GET FAMILIAR WITH THE COMPANY

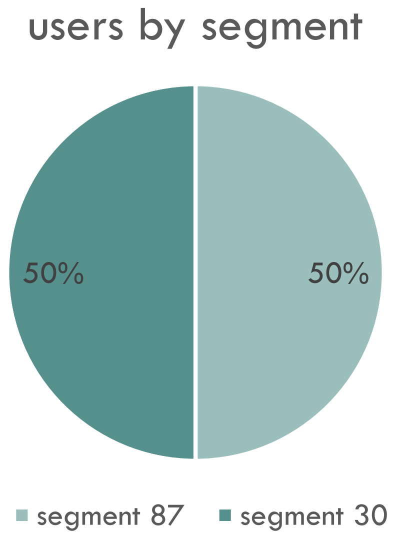
/ *What segments of users exist?*

From the first query we can see two segments – 30 and 87. Let's write an additional query to confirm the segments and the number of users in each segment

```sql
SELECT DISTINCT segment
,COUNT (*) as total
FROM subscriptions
GROUP BY segment;
```

segment	total
87	1000
30	1000

New query confirms that there are two segments of subscribers – 30 and 87



CHURN TREND

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

   Let's perform churn analysis for January, February and March.
   For that let's first create a temporary table ***months*** which will give us three time periods

```
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
FROM subscriptions;
```

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

CHURN TREND

Let's cross join *subscriptions* table with *months*. This will give us a range of data to work with

```
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
  FROM subscriptions
)

  SELECT *
  FROM subscriptions
  CROSS JOIN months;
```

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
..	…	…	…	…	…
2000	2017-03-30	Ø	30	2017-01-01	2017-01-31
2000	2017-03-30	Ø	30	2017-02-01	2017-02-28
2000	2017-03-30	Ø	30	2017-03-01	2017-03-31

CHURN TREND

Let's keep extracting data we need by creating a new table ***status*** from previous table (which we have named ***cross_join)*** and add two new columns: *is_canceled_87* and *is_canceled_30*

is_canceled_87 and *is_canceled_30* equals 1 if the subscription has been cancelled and 0 otherwise

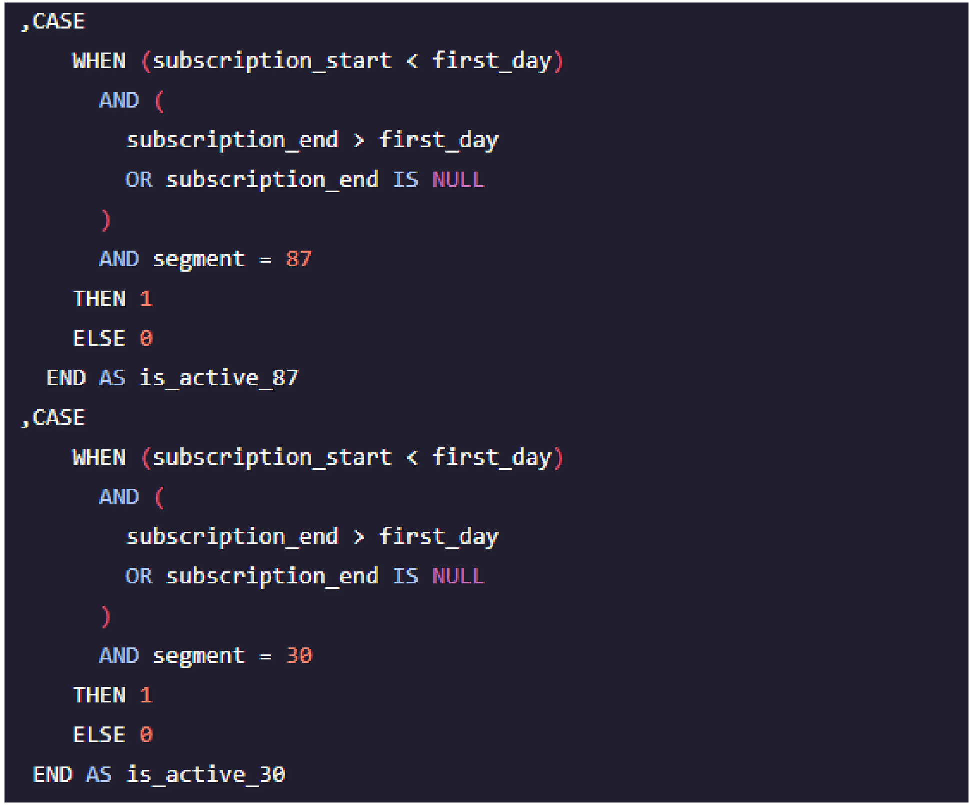
```
SELECT id
,first_day as month
,CASE
    WHEN subscription_end BETWEEN first_day AND last_day
    AND segment = 87
    THEN 1
    ELSE 0
  END AS is_canceled_87
,CASE
    WHEN subscription_end BETWEEN first_day AND last_day
    AND segment = 30
    THEN 1
    ELSE 0
  END AS is_canceled_30
FROM cross_join;
```

id	month	is_canceled_87	is_canceled_30
1	2017-01-01	0	0
1	2017-02-01	1	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	0	0
3	2017-02-01	0	0
3	2017-03-01	1	0
4	2017-01-01	0	0
4	2017-02-01	1	0
4	2017-03-01	0	0
5	2017-01-01	0	0
5	2017-02-01	0	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	0	0
...	...	...	...
1999	2017-01-01	0	0
1999	2017-02-01	0	0
1999	2017-03-01	0	0
2000	2017-01-01	0	0
2000	2017-02-01	0	0
2000	2017-03-01	0	0

CHURN TREND

Similarly to *is_canceled_87* and *is_canceled_30* let's add two more columns: *is_active_87* and *is_active_30*

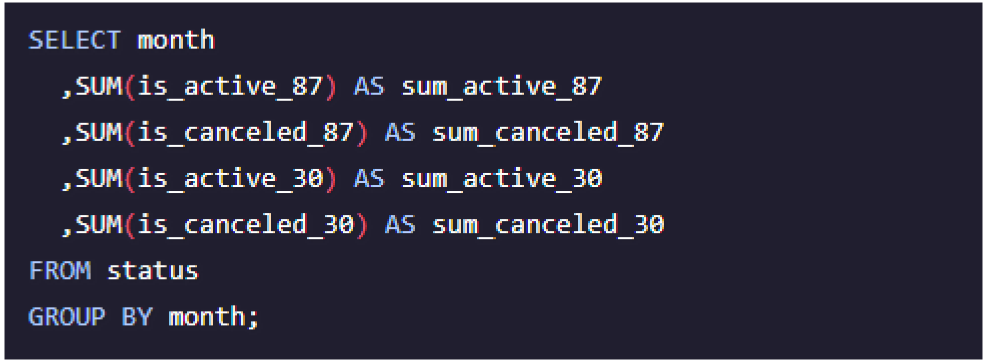
To perform churn analysis we need to compare the number of active subscriptions to the number of cancelled subscriptions in a fixed time period (month in our case)



```
,CASE
    WHEN (subscription_start < first_day)
      AND (
        subscription_end > first_day
        OR subscription_end IS NULL
      )
      AND segment = 87
    THEN 1
    ELSE 0
  END AS is_active_87
,CASE
    WHEN (subscription_start < first_day)
      AND (
        subscription_end > first_day
        OR subscription_end IS NULL
      )
      AND segment = 30
    THEN 1
    ELSE 0
 END AS is_active_30
```

CHURN TREND

To do that let’s create another table using previous table (which we have named *status*)
The table should give us total of cancelled and active subscriptions by segment per each month



```
SELECT month
  ,SUM(is_active_87) AS sum_active_87
  ,SUM(is_canceled_87) AS sum_canceled_87
  ,SUM(is_active_30) AS sum_active_30
  ,SUM(is_canceled_30) AS sum_canceled_30
FROM status
GROUP BY month;
```

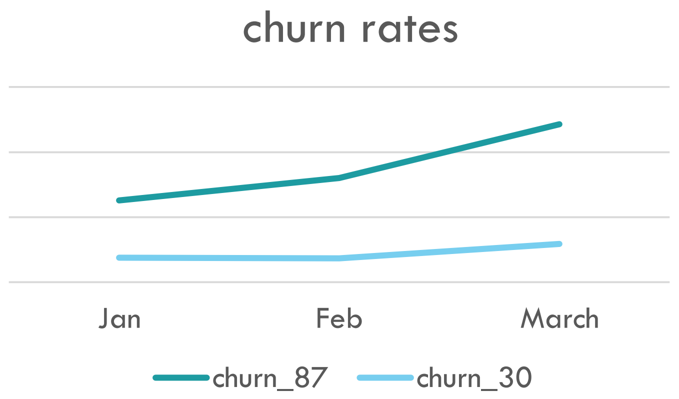
month	sum_active_87	sum_canceled_87	sum_active_30	sum_canceled_30
2017-01-01	278	70	291	22
2017-02-01	462	148	518	38
2017-03-01	531	258	716	84

CHURN TREND

Naming previous table ***status_aggregate*** and using it's data we can now calculate churn rates per month

```
SELECT month
  ,1.0 * sum_canceled_87/sum_active_87 as churn_87
  ,1.0 * sum_canceled_30/sum_active_30 as churn_30
FROM status_aggregate
GROUP BY month;
```

month	churn_87	churn_30
2017-01-01	0.251798561	0.07560137
2017-02-01	0.32034632	0.07335907
2017-03-01	0.485875706	0.11731844



CHURN TREND / CHURN RATES

Churn trend in January, February and March:

month	churn_87	churn_30
2017-01-01	0.251798561	0.07560137
2017-02-01	0.32034632	0.07335907
2017-03-01	0.485875706	0.11731844

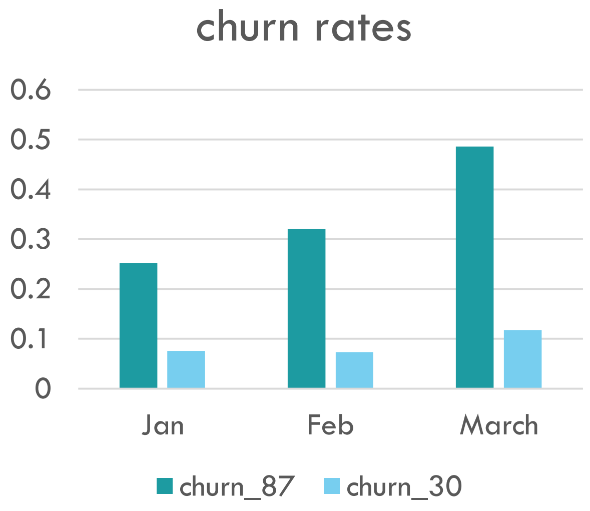
Churn trend in % compared to previous period:

month	churn_87	churn_30
2017-01-01	25.17%	7.56%
2017-02-01	27.20%	2.97%
2017-03-01	51.67%	59.92%

Apart from February when churn rate for ***segment 30*** has dropped by ~3% overall churn rate has been growing for both segments of users

Churn rate for ***segment 87*** has been significantly larger than for ***segment 30***

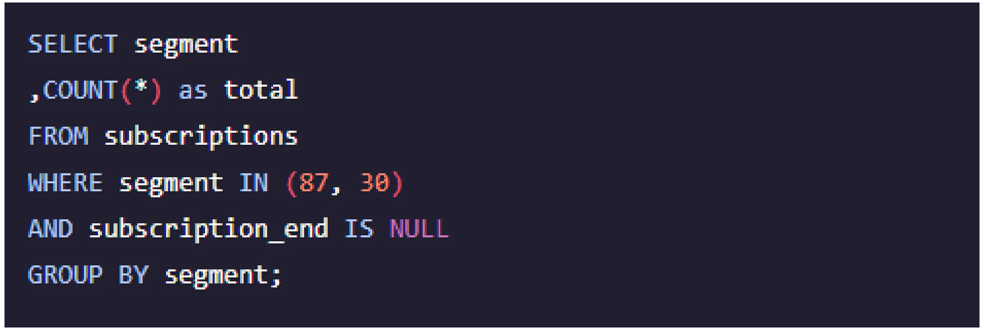
Biggest churn rate jump was for ***segment 30*** in March (almost 60% bigger churn compared to previous period)



CHURN RATES

/ *What segments of users should the company focus on expanding?*

Let’s write another query to find a number of active users per each segment



```
SELECT segment
,COUNT(*) as total
FROM subscriptions
WHERE segment IN (87, 30)
AND subscription_end IS NULL
GROUP BY segment;
```

segment	total
87	524
30	856

Considering that in total there have been 1000 users for each segment the company should focus on expanding ***segment_87*** since churn rates for this segment have been higher and total number of active users is smaller