

# CodeFlix Churn Rate

# Table of Contents

1. Codeflix Data
2. Setting up the Data to Calculate the Churn Rate
3. What is the overall churn trend?

# CodeFlix Data

# The Subscriptions Table

The subscriptions table contains the following data: id, subscription start date, subscription end date and a segment identifier.

- The company has been operating since December 2016 and there is enough data to calculate the churn rate for January – March 2017, since no subscriptions ended in December 2016.
- There are 2 segments in this table, 87 and 30.

To view a subset of the data I used the query:

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

To find the range of data available I used the query:

```
SELECT MIN(subscription_start) AS 'Start', MAX(subscription_end) AS 'End'  
FROM subscriptions;
```

id	subscription_start	subscription_end	segment	Start	End
1	2016-12-01	2017-02-01	87	2016-12-01	2017-03-31
2	2016-12-01	2017-01-24	87		
14	2016-12-01	2017-03-07	30		
15	2016-12-01	2017-02-22	30		

# Setting Up the Data to Calculate Churn Rate

# Setting up the Months Table

- The first step is to setup a temporary table with the first and last day of each month to use to determine subscription length.

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

```
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
)
SELECT *
FROM months;
```

# Cross Join Months and Subscriptions

- Next we need to create another temporary table called `cross_join` to be used to compare the subscription start and end dates with the first and last days of the months we are analyzing.

id	subscription_start	subscription_end	segment	first_day	last_day
17	2016-12-01		30	2017-01-01	2017-01-31
17	2016-12-01		30	2017-02-01	2017-02-28
17	2016-12-01		30	2017-03-01	2017-03-31
18	2016-12-02	2017-01-29	87	2017-01-01	2017-01-31
18	2016-12-02	2017-01-29	87	2017-02-01	2017-02-28
18	2016-12-02	2017-01-29	87	2017-03-01	2017-03-31

```
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)

SELECT *
FROM cross_join;
LIMIT 500;
```

# Find Subscription Status

- For each segment, compare subscription start date to the first day of the month to determine if the subscription was active for that month.

id	Month	is_active_87	is_active_30
12	2017-01-01	1	0
12	2017-02-01	1	0
12	2017-03-01	1	0
13	2017-01-01	0	1
13	2017-02-01	0	1
13	2017-03-01	0	1

```
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, first_day AS 'Month',
CASE WHEN (segment = 87 AND first_day >
subscription_start) THEN 1 ELSE 0 END AS 'is_active_87',
CASE WHEN (segment = 30 AND first_day >
subscription_start) THEN 1 ELSE 0 END AS 'is_active_30',
FROM cross_join
)
SELECT *
FROM status
LIMIT 500;
```



# Find Subscription Status

- For each segment, compare subscription start date to the first day of the month to determine if the subscription was active for that month and compare the last day of the month to the subscription end date to determine if the subscription was canceled during that month.

id	Month	is_active_87	is_active_30	is_canceled_87	is_canceled_30
12	2017-01-01	1	0	0	0
12	2017-02-01	1	0	1	0
12	2017-03-01	1	0	1	0
13	2017-01-01	0	1	0	0
13	2017-02-01	0	1	0	0
13	2017-03-01	0	1	0	0

```
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, first_day AS 'Month',
CASE WHEN (segment = 87 AND first_day > subscription_start)
THEN 1 ELSE 0 END AS 'is_active_87',
CASE WHEN (segment = 30 AND first_day > subscription_start)
THEN 1 ELSE 0 END AS 'is_active_30',
CASE WHEN (segment = 87 AND last_day > subscription_end)
THEN 1 ELSE 0 END AS 'is_canceled_87',
CASE WHEN (segment = 30 AND last_day > subscription_end)
THEN 1 ELSE 0 END AS 'is_canceled_30'
FROM cross_join
)
SELECT *
FROM status
LIMIT 500;
```

# Overall Churn Trend

# Determine the Total Number of Active and Canceled Subscriptions

- We need to determine the total number of active and canceled subscriptions over the time period we are analyzing so that we can then determine the churn rate for each segment.

sum_active_87	sum_active_30	sum_canceled_87	sum_canceled_30
1575	1609	743	221

```
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, first_day AS 'Month',
CASE WHEN (segment = 87 AND first_day > subscription_start) THEN 1 ELSE 0
END as 'is_active_87',
CASE WHEN (segment = 30 AND first_day > subscription_start) THEN 1 ELSE 0
END as 'is_active_30',
CASE WHEN (segment = 87 AND last_day > subscription_end) THEN 1 ELSE 0
END as 'is_canceled_87',
CASE WHEN (segment = 30 AND last_day > subscription_end) THEN 1 ELSE 0
END as 'is_canceled_30'
FROM cross_join
),
status_aggregate as (
select sum(is_active_87) as'sum_active_87', sum(is_active_30) as
'sum_active_30', sum(is_canceled_87) as 'sum_canceled_87',
sum(is_canceled_30) as 'sum_canceled_30'
from status
)
Select *
from status_aggregate;
```

# Calculate the Churn Rate for Each Segment

- Last we will determine the churn rate for each segment. As you can see the churn rate for segment 30 is considerably lower than the churn rate for segment 87.

churn_rate_87	churn_rate_30
0.471746031746032	0.137352392790553

```
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, first_day AS 'Month',
CASE WHEN (segment = 87 AND first_day > subscription_start) THEN 1 ELSE 0
END as 'is_active_87',
CASE WHEN (segment = 30 AND first_day > subscription_start) THEN 1 ELSE 0
END as 'is_active_30',
CASE WHEN (segment = 87 AND last_day > subscription_end) THEN 1 ELSE 0
END as 'is_canceled_87',
CASE WHEN (segment = 30 AND last_day > subscription_end) THEN 1 ELSE 0
END as 'is_canceled_30'
FROM cross_join
),
status_aggregate as (
select sum(is_active_87) as'sum_active_87', sum(is_active_30) as
'sum_active_30', sum(is_canceled_87) as 'sum_canceled_87',
sum(is_canceled_30) as 'sum_canceled_30'
from status
)
select 1.0 * sum_canceled_87 / sum_active_87 as 'churn_rate_87', 1.0 *
sum_canceled_30 / sum_active_30 as'churn_rate_30'
from status_aggregate;
```