



Freya Hoppe Capstone

Learn SQL from Scratch

Table of Contents

1. Get familiar with Codeflix
2. What is the overall churn rate by month?
3. Compare the churn rates between segments

Get Familiar with Codeflix

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?

Months in operation, sufficient info to calculate churn

Subscription start data includes values from December 2016 – March 2017. Subscription end data includes Null, and values between January and March 2017.

- The company has been in operation 4 months
- Churn can be calculated for 3 months only (no churn was possible in December due to 31-day minimum subscription length)

```
-- Queries used to explore data
SELECT DISTINCT subscription_start
FROM subscriptions;

SELECT DISTINCT subscription_end
FROM subscriptions
ORDER BY subscription_end;
```

What segments of users exist?

Two segments of users exist: 87 and 30.

segment
87
30

```
-- Queries used to explore data
SELECT DISTINCT segment
FROM subscriptions;
```

What is the overall churn rate by month?

Overall churn rate by month

The overall churn rate is increasing. The company should look into what happened to cause such a big spike in March.

- January: 16%
- February: 19%
- March: 27%

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

```
status AS
(SELECT id, first_day as month,
CASE
  WHEN (subscription_start < first_day)
    AND (
      subscription_end > first_day
      OR subscription_end IS NULL
    ) THEN 1
    ELSE 0
END as is_active,
CASE
  WHEN subscription_end BETWEEN first_day AND last_day
  THEN 1
  ELSE 0
END as is_canceled
FROM cross_join),
status_aggregate AS
(SELECT
  month,
  SUM(is_active) as active,
  SUM(is_canceled) as canceled
FROM status
GROUP BY month)
SELECT month, 1.0 * canceled / active AS churn_rate
FROM status_aggregate;
```

Compare the churn rates between segments

Which segment of users should the company focus on expanding?

Churn rates between segments

Whatever segment “30” is, it’s outstanding and they should keep doing more of that.

- The “87” segment is churning between $\frac{1}{4}$ and $\frac{1}{2}$ of their customers each month, and trending sharply upwards
- After a spike in churn in March, “30” has ended churning only 12% of its customers, much less than the best month for 87
- It’s worth investigating what happened in March that may have contributed to both segments spiking

month	churn_rate_87	churn_rate_30
2017-01-01	0.252	0.076
2017-02-01	0.320	0.073
2017-03-01	0.486	0.117

```
status_aggregate AS
(SELECT
  month,
  SUM(is_active_87) as active_87, SUM(is_active_30) as
active_30,
  SUM(is_canceled_87) as canceled_87,
  SUM(is_canceled_30) as canceled_30
FROM status
GROUP BY month)
```

```
SELECT month, 1.0 * canceled_87 / active_87 AS
churn_rate_87, 1.0 * canceled_30 / active_30 AS
churn_rate_30
FROM status_aggregate;
```

Appendix - queries

Queries used – section 1

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

```
SELECT DISTINCT subscription_start  
FROM subscriptions;
```

```
SELECT DISTINCT subscription_end  
FROM subscriptions  
ORDER BY subscription_end;
```

```
SELECT DISTINCT segment  
FROM subscriptions;
```

Queries used – section 2

```
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 (subscription_start < first_day)
    AND (
      subscription_end > first_day
      OR subscription_end IS NULL
    ) THEN 1
  ELSE 0
END as is_active,
CASE
```

Queries used – section 3

```
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 (subscription_start < first_day)
  AND (
    subscription_end > first_day
    OR subscription_end IS NULL
  ) AND segment = '87'
```

```
THEN 1
```