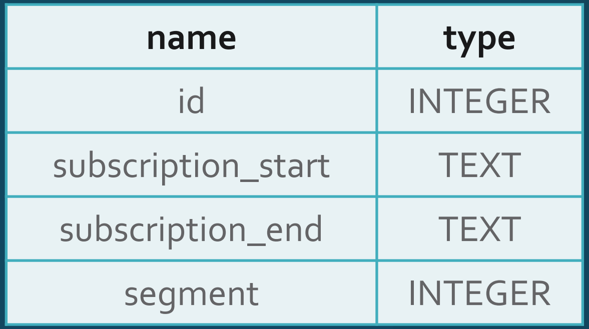
# Codeflix User Churn

Codecademy

Dean Della Loggia

# Table of Contents

- Subscriptions Table Schema:

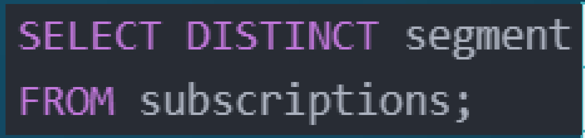


| name | type |
|---|---|
| id | INTEGER |
| subscription_start | TEXT |
| subscription_end | TEXT |
| segment | INTEGER |

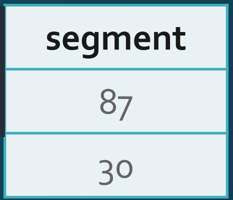
- Data Inspection
  - What user segments exist?
  - How many months has Codeflix been operating?
  - For which months can churn rate be calculated?
- Churn Rate by Segment
  - Which segment has the lower churn rate?
  - Which user segment should Codeflix focus on expanding?

# Data Inspection

- What user segments exist?
  - Two user segments exist: 87 and 30.

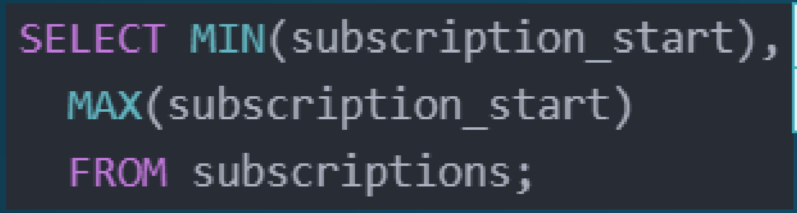


```
SELECT DISTINCT segment
FROM subscriptions;
```

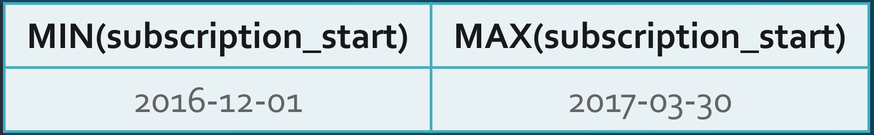


| segment |
|---|
| 87 |
| 30 |

- How many months has Codeflix been operating?
  - Codeflix has been operating for four months, with the first subscription starting on 12-1-2016 and the last starting on 3-30-2017.



```
SELECT MIN(subscription_start),
  MAX(subscription_start)
  FROM subscriptions;
```

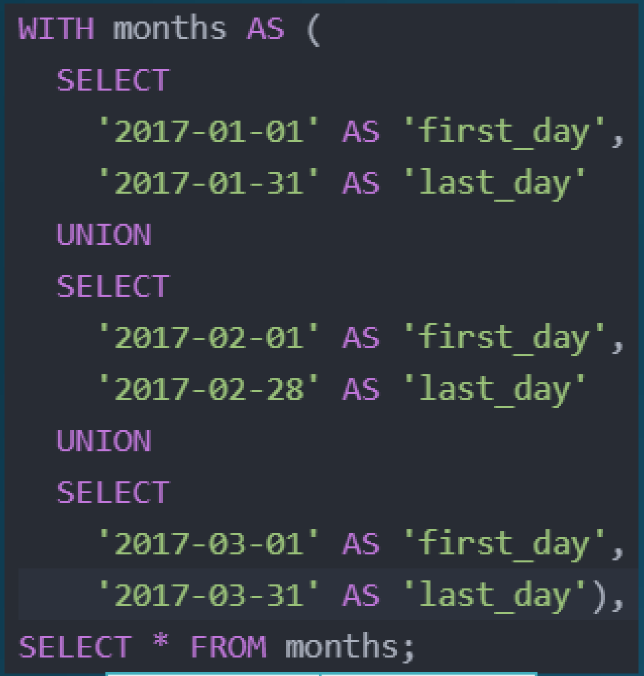


| MIN(subscription_start) | MAX(subscription_start) |
|---|---|
| 2016-12-01 | 2017-03-30 |

- For which months can churn rate be calculated?
  - Since Codeflix has a minimum subscription length of 31 days and a churn rate calculation requires both subscriptions and cancellations, churn rate can be calculated from 1/2017 to 3/2017.

# Churn Rate by Segment

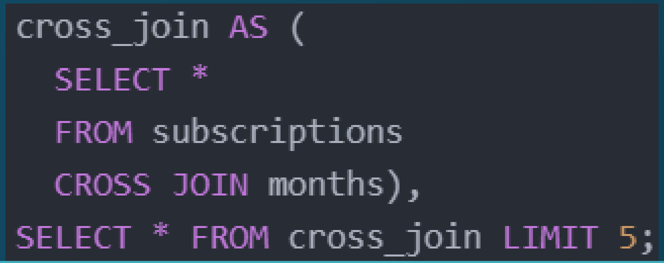
## 1) Create ‘months’ table:



```
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;
```

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

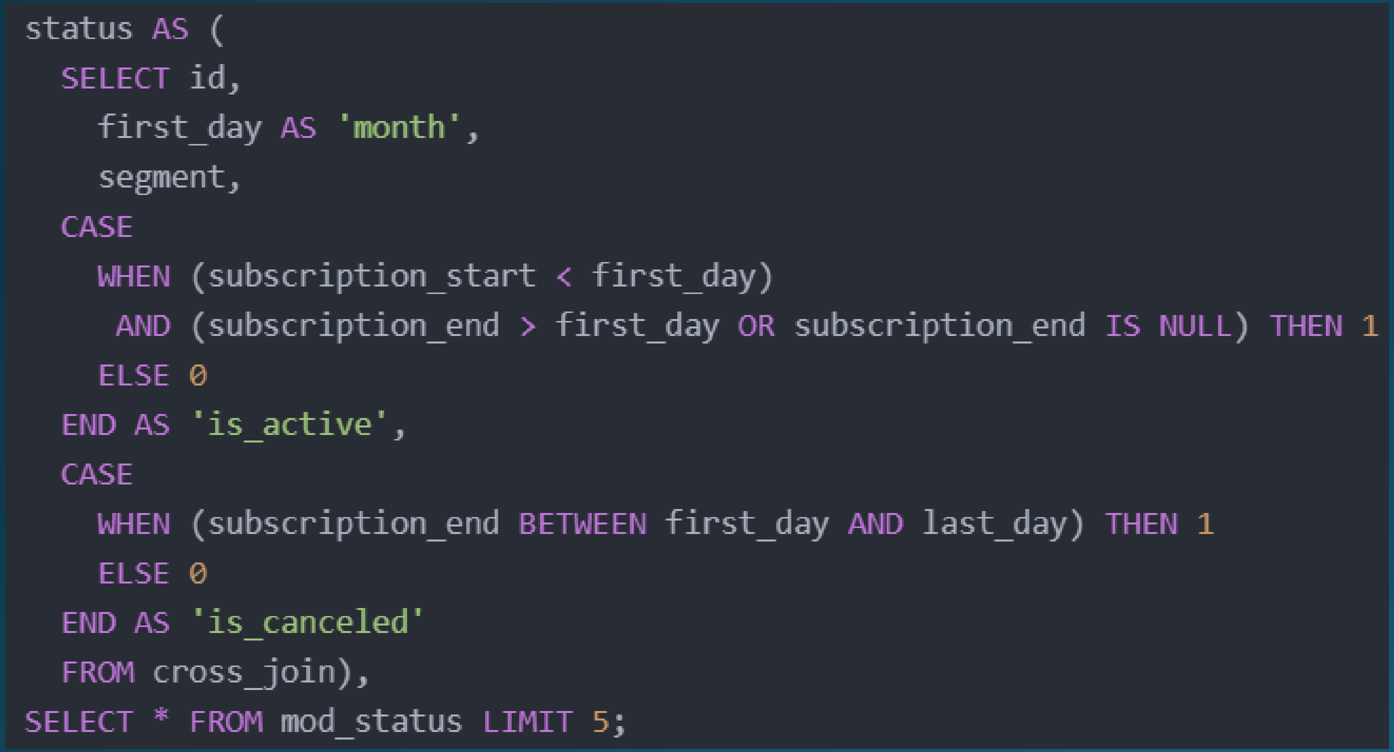
## 2) Join ‘subscriptions’ and ‘months’ tables:



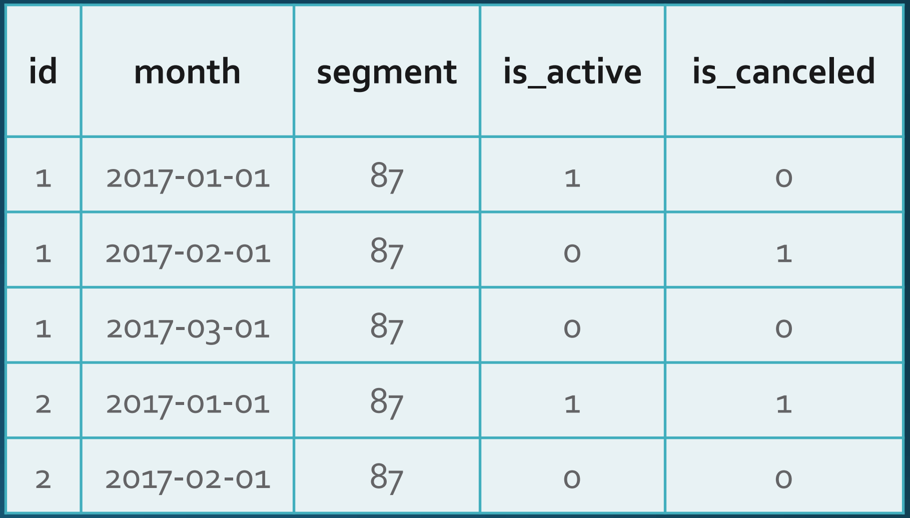
```
cross_join AS (
  SELECT *
  FROM subscriptions
  CROSS JOIN months),
SELECT * FROM cross_join LIMIT 5;
```

| 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 |

# 3) Create 'status' table from ‘cross_join’ that identifies active and canceled users from each segment:

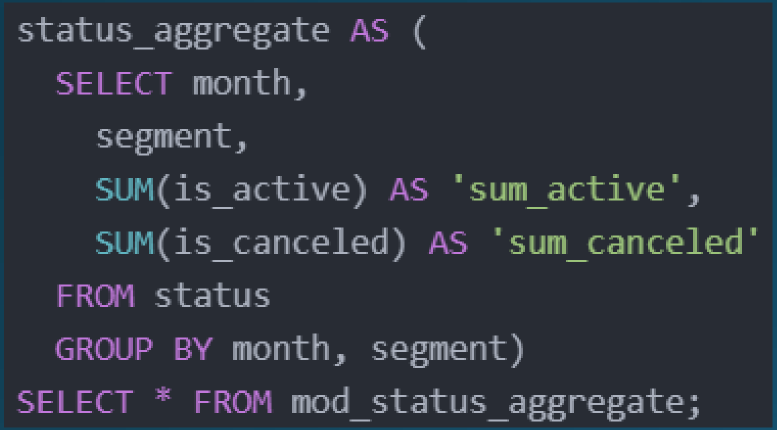


```
status AS (
  SELECT id,
    first_day AS 'month',
    segment,
  CASE
    WHEN (subscription_start < first_day)
     AND (subscription_end > first_day OR subscription_end IS NULL) THEN 1
    ELSE 0
  END AS 'is_active',
  CASE
    WHEN (subscription_end BETWEEN first_day AND last_day) THEN 1
    ELSE 0
  END AS 'is_canceled'
  FROM cross_join),
SELECT * FROM mod_status LIMIT 5;
```



| id | month | segment | is_active | is_canceled |
|---|---|---|---|---|
| 1 | 2017-01-01 | 87 | 1 | 0 |
| 1 | 2017-02-01 | 87 | 0 | 1 |
| 1 | 2017-03-01 | 87 | 0 | 0 |
| 2 | 2017-01-01 | 87 | 1 | 1 |
| 2 | 2017-02-01 | 87 | 0 | 0 |

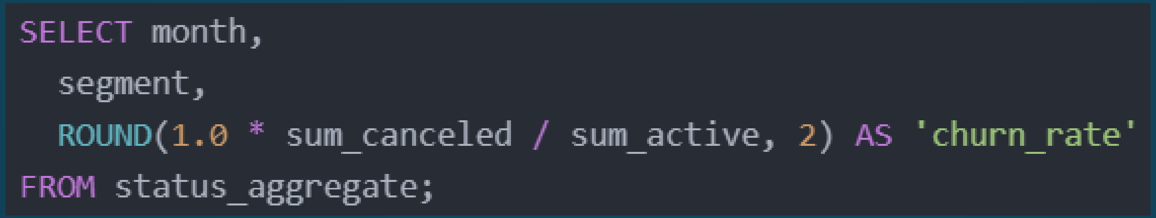
# 4) Create 'status_aggregate’ table that sums active and canceled subscriptions by segment:



```
status_aggregate AS (
  SELECT month,
    segment,
    SUM(is_active) AS 'sum_active',
    SUM(is_canceled) AS 'sum_canceled'
  FROM status
  GROUP BY month, segment)
SELECT * FROM mod_status_aggregate;
```

| month | segment | sum_active | sum_canceled |
|---|---|---|---|
| 2017-01-01 | 30 | 291 | 22 |
| 2017-01-01 | 87 | 278 | 70 |
| 2017-02-01 | 30 | 518 | 38 |
| 2017-02-01 | 87 | 462 | 148 |
| 2017-03-01 | 30 | 716 | 84 |
| 2017-03-01 | 87 | 531 | 258 |

# 5) Calculate churn rate for the two segments over the 3-month period:



```
SELECT month,
  segment,
  ROUND(1.0 * sum_canceled / sum_active, 2) AS 'churn_rate'
FROM status_aggregate;
```

| month | segment | churn_rate |
|---|---|---|
| 2017-01-01 | 30 | 0.08 |
| 2017-01-01 | 87 | 0.25 |
| 2017-02-01 | 30 | 0.07 |
| 2017-02-01 | 87 | 0.32 |
| 2017-03-01 | 30 | 0.12 |
| 2017-03-01 | 87 | 0.49 |

# Churn Rate Conclusions

| month | segment | churn_rate |
|---|---|---|
| 2017-01-01 | 30 | 0.08 |
| 2017-01-01 | 87 | 0.25 |
| 2017-02-01 | 30 | 0.07 |
| 2017-02-01 | 87 | 0.32 |
| 2017-03-01 | 30 | 0.12 |
| 2017-03-01 | 87 | 0.49 |

- Which segment has the lower churn rate?
  - Segment 30 is lower in every month
- Which user segment should Codeflix focus on expanding?
  - Segment 30 is completely outperforming segment 87 in gaining and retaining subscribers.
  - Focus on expanding segment 30.