Analysis of Subscription Trends for Codeflix Using SQL

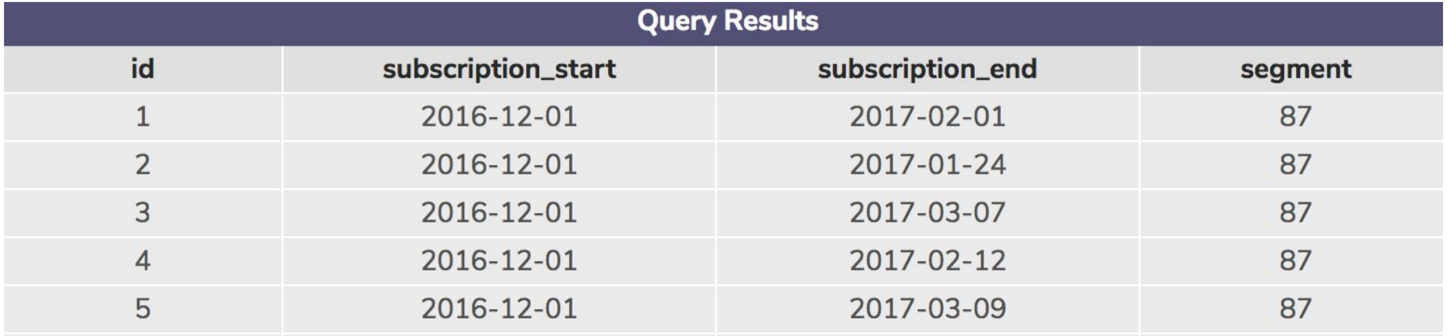
Olayinka Olatunji-Ojo, Ph.D.

What is Codeflix?

- A streaming video start-up that has been in business for 4 months
- Codeflix is interested in the “churn rate” of subscribers
  - Churn rate is the rate at which customers stop subscribing to a service
  - Subscriber information provided in a table called subscriptions
  - Data used for analysis is 3 months; January 2017 - March 2017
- Subscribers are divided into two segments
  - 87
  - 30
- Determine churn rate per month
  - Also compare churn rate between segments

What Basic Information is Provided?

```
SELECT *
FROM subscriptions
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

- id - subscriber identifier
- Subscription start date
- Subscription end date
- Segment: two segments; 87 and 30

Closer Examination of Data

- How much data is provided?

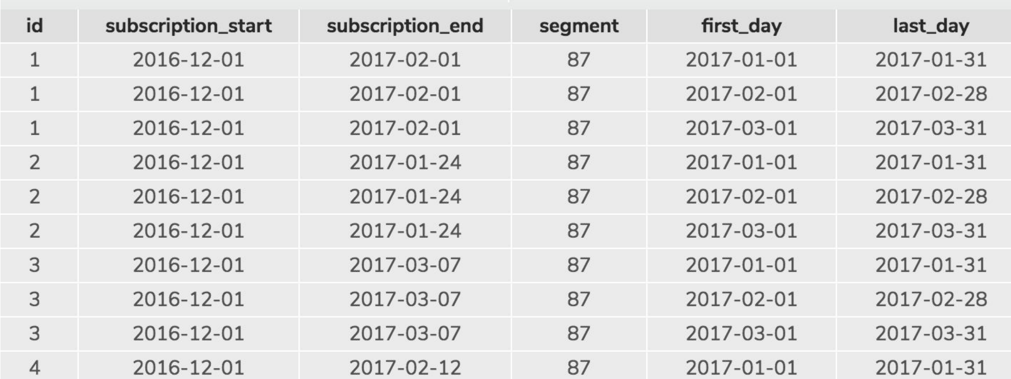
2000 lines of information

- Can we determine the range of months in the subscriptions table?
  - First month - 2016-12-01
  - Last month - 2017-03-30
- Four months of data provided but only 3 can be used for churn rate
  - Subscription end date is only available for 3 months!

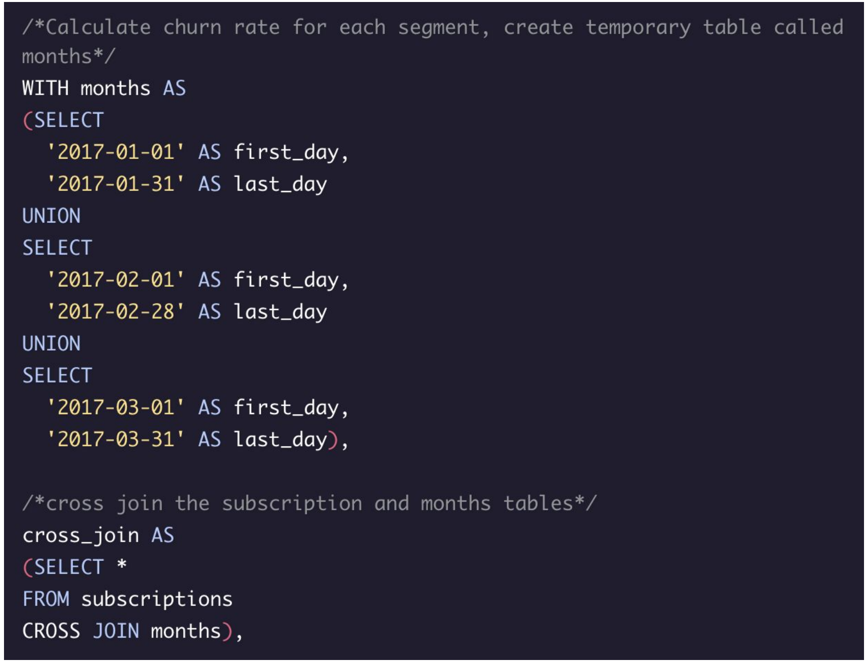
```
/*count total number of entries in the table*/
SELECT COUNT(*)
FROM subscriptions;

 /*Determine range of months of data provided*/
 SELECT MIN(subscription_start) AS first_month,
           MAX(subscription_start) AS last_month
 FROM subscriptions;
```

Pertinent Months

- Extract months needed to its own table
  - Create new table called months
- Need to provide subscriber information for the months selected
- This is achieved by cross joining the subscriptions and months tables



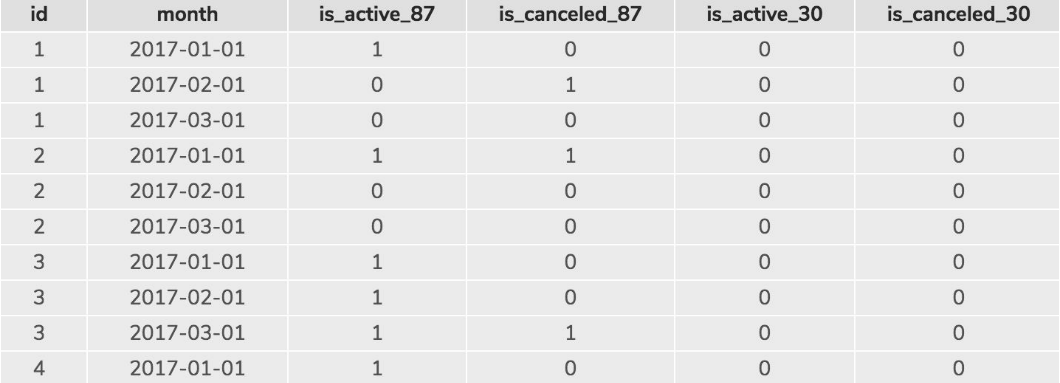
id	subscription_start	subscription_end	segment	first_day	last_day
1	2016-12-01	2017-02-01	87	2017-01-01	2017-01-31
1	2016-12-01	2017-02-01	87	2017-02-01	2017-02-28
1	2016-12-01	2017-02-01	87	2017-03-01	2017-03-31
2	2016-12-01	2017-01-24	87	2017-01-01	2017-01-31
2	2016-12-01	2017-01-24	87	2017-02-01	2017-02-28
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



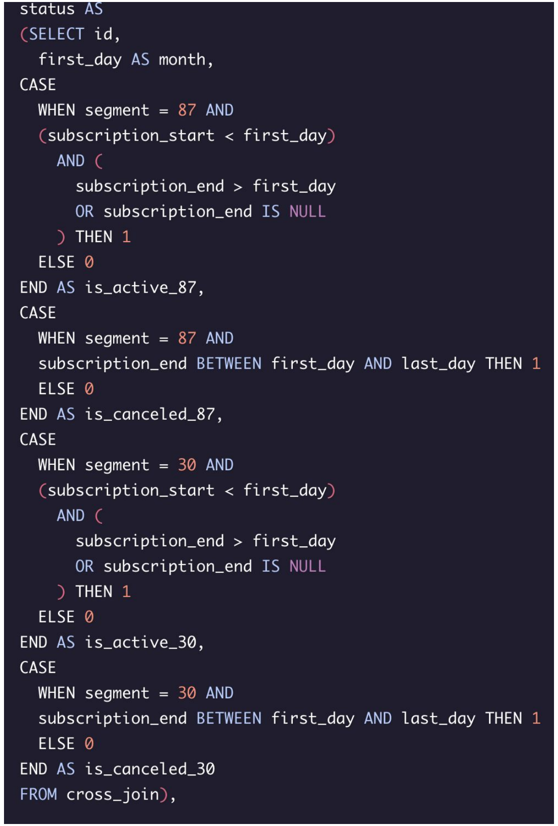
```
/*Calculate churn rate for each segment, create temporary table called
months*/
WITH months AS
(SELECT
  '2017-01-01' AS first_day,
  '2017-01-31' AS last_day
UNION
SELECT
  '2017-02-01' AS first_day,
  '2017-02-28' AS last_day
UNION
SELECT
  '2017-03-01' AS first_day,
  '2017-03-31' AS last_day),

/*cross join the subscription and months tables*/
cross_join AS
(SELECT *
FROM subscriptions
CROSS JOIN months),
```

Determine Active and Canceled Subscribers

- We are interested in monthly churn rate
- Churn rate depends on canceled vs active subscribers
- Determine active and canceled subscribers by segment
  - Group subscribers by segments



id	month	is_active_87	is_canceled_87	is_active_30	is_canceled_30
1	2017-01-01	1	0	0	0
1	2017-02-01	0	1	0	0
1	2017-03-01	0	0	0	0
2	2017-01-01	1	1	0	0
2	2017-02-01	0	0	0	0
2	2017-03-01	0	0	0	0
3	2017-01-01	1	0	0	0
3	2017-02-01	1	0	0	0
3	2017-03-01	1	1	0	0
4	2017-01-01	1	0	0	0



```sql
status AS
(SELECT id,
  first_day AS month,
CASE
  WHEN segment = 87 AND
  (subscription_start < first_day)
    AND (
      subscription_end > first_day
      OR subscription_end IS NULL
    ) THEN 1
  ELSE 0
END AS is_active_87,
CASE
  WHEN segment = 87 AND
  subscription_end BETWEEN first_day AND last_day THEN 1
  ELSE 0
END AS is_canceled_87,
CASE
  WHEN segment = 30 AND
  (subscription_start < first_day)
    AND (
      subscription_end > first_day
      OR subscription_end IS NULL
    ) THEN 1
  ELSE 0
END AS is_active_30,
CASE
  WHEN segment = 30 AND
  subscription_end BETWEEN first_day AND last_day THEN 1
  ELSE 0
END AS is_canceled_30
FROM cross_join),
```

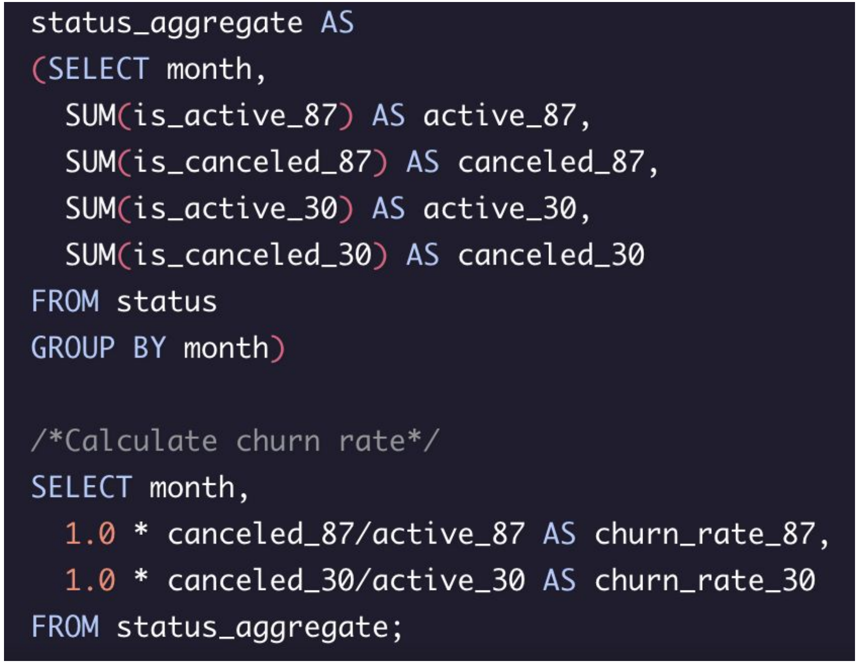
What is the Current Subscriber Breakdown?

Month, 2017	active_87	canceled_87	active_30	canceled_30
January	278	70	291	22
February	462	148	518	38
March	531	258	716	84

- Each segment has roughly the same amount of total subscribers

- Segment 87 has more canceled subscribers monthly than segment 30

- **What is the monthly churn rate?**

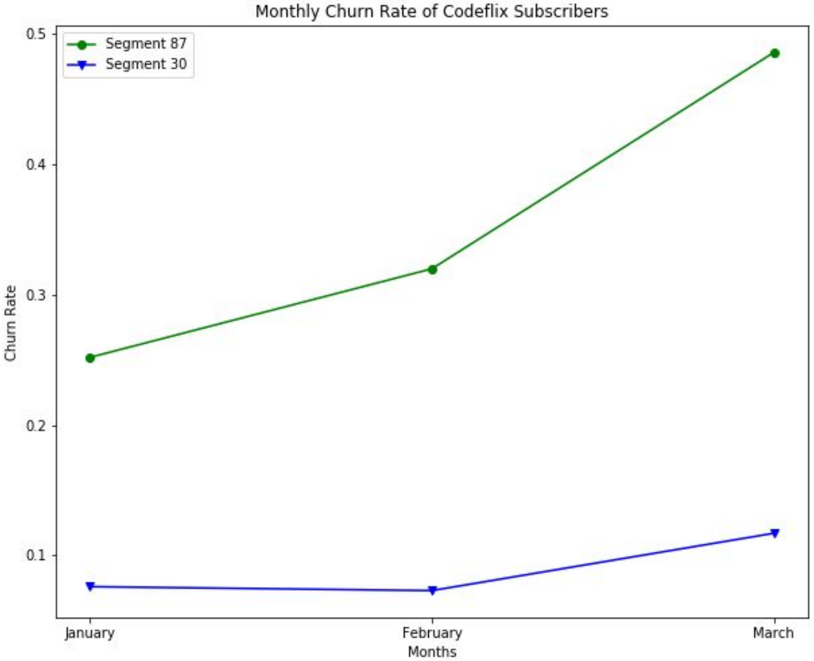


```
status_aggregate AS
(SELECT month,
  SUM(is_active_87) AS active_87,
  SUM(is_canceled_87) AS canceled_87,
  SUM(is_active_30) AS active_30,
  SUM(is_canceled_30) AS canceled_30
FROM status
GROUP BY month)

/*Calculate churn rate*/
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;
```

Monthly Churn Rate

- Churn rate has increased monthly
- Segment 30 has lower churn rates
- Highest loss of subscribers in March
  - Segment 30 churn rate is still lower than segment 87
- What are the major differences between the segments?
- What changes did the company make in the month of March leading to subscriber loss?

Month	Segment 87	Segment 30
January	0.252	0.076
February	0.320	0.073
March	0.486	0.117



Summary and Suggestions

- 3 months of Codeflix data was analyzed to determine churn rate
  - Churn rate has increased monthly
  - Lower churn rate from segment 30 than segment 87
  - Highest churn rate was from March
- February churn rate in segment 30 was lower than January and March
- Subscribers in segment 30 should be increased or reproduce segment 30 marketing efforts in segment 87 to achieve similar churn rate
- Company decisions implemented on or before March should be investigated to determine cause of higher churn rate for that month.