codecademy

Ryan Leatham Capstone

Learn SQL from Scratch

Table of Contents

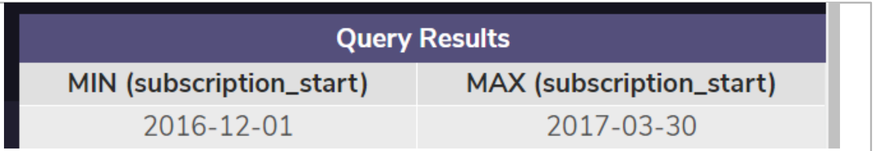
1. Get familiar with Codeflix
2. What is the overall churn rate by month?
3. Compare the churn rates between segments

1. Get Familiar with Codeflix

Get Familiar Answers

How many months has the company been operating?

- Answer: 4 months thus far



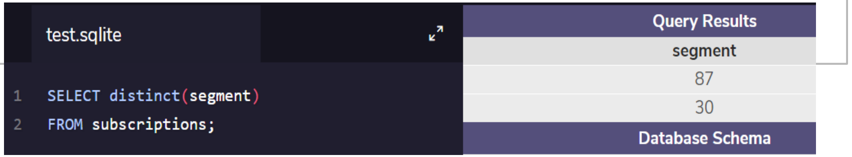
MIN (subscription_start)	MAX (subscription_start)
2016-12-01	2017-03-30

Which months do you have enough information to calculate a churn rate?

- All but December, since minimum subscription time is 31 days (meaning there won’t be cancellations during December, a necessary component for measuring churn)

What segments of users exist?

- 87
- 30



```sql
SELECT distinct(segment)
FROM subscriptions;
```

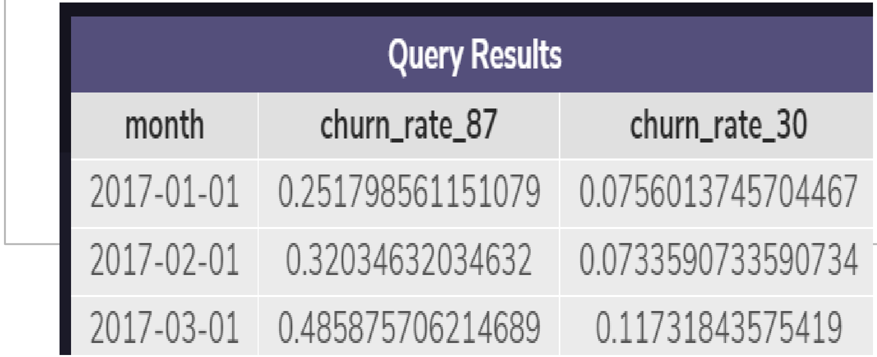
segment
87
30

2. What is the overall churn rate by month?

Overall Churn Rate Answers

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

- The overall churn rate is increasing since the company started



Query Results

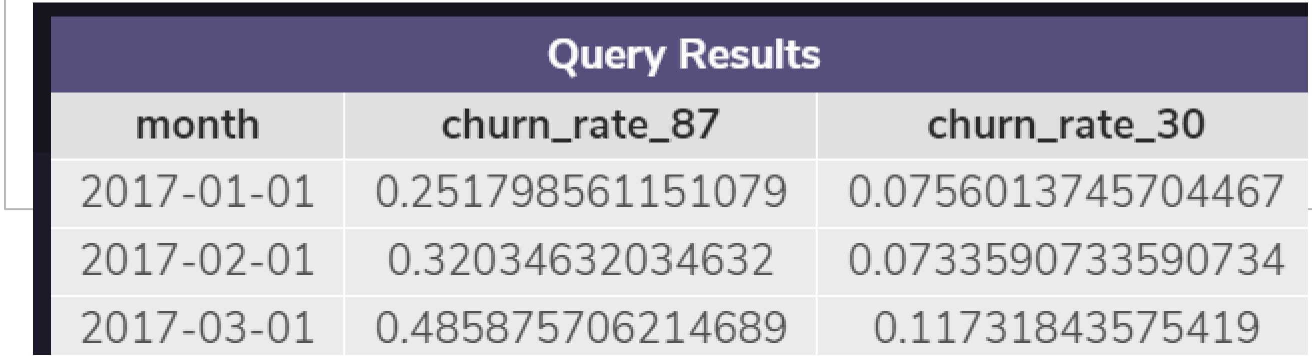
month	churn_rate_87	churn_rate_30
2017-01-01	0.251798561151079	0.0756013745704467
2017-02-01	0.32034632034632	0.0733590733590734
2017-03-01	0.485875706214689	0.11731843575419

3. Compare the churn rates between segments

Compare the Churn Rates Answers

3. Compare the churn rates between user segments. Which segment of users should the company focus on expanding?

- The company should focus on expanding segment 30, as it appears more effective (lower churn)



Query Results

month	churn_rate_87	churn_rate_30
2017-01-01	0.251798561151079	0.0756013745704467
2017-02-01	0.32034632034632	0.0733590733590734
2017-03-01	0.485875706214689	0.11731843575419

Final Code

Final Code

```
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
    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,
   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
   ),
   status_aggregate AS
   (SELECT
    month,
   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
   GROUP BY month
   )
   SELECT month, 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;
```

Thank You