codecademy

Learn SQL from Scratch

Capstone project by Roman Khromotov

Example Table of Contents

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

Introduction. The Codeflix

- The company has been operating for four month already. Since 2016-12-01 till 2017-03-30. We have data for only three months available, inasmuch as there is no ”subscription_end” values for December 2016
- There are two segments of users: 30 and 87

id	Subscription_start	Subscription_end	Segment
1	2016-12-01	2017-02-01	87
30	2016-12-02	2017-01-20	30
98	2016-12-06	2017-03-05	87

```
-- You can put your query here
SELECT *
FROM subscriptions
LIMIT 100;

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

Preliminary work

- First of all, we had to do some preliminary work creating tables months and cross_join
- They will help us to calculate churn rates

id	Subscription_start	Subscription_end	Segment
1	2016-12-01	2017-02-01	87
30	2016-12-02	2017-01-20	30
98	2016-12-06	2017-03-05	87

```
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
```

Overall Churn Rates

- Using the code on the right, I was able to calculate overall churn rate for the period
- I grouped churn rates by months
- Further we will group them by segment too
- The resulting churn rates are in the table below

month	churn_rate
2017-01-01	0.16168
2017-02-01	0.18979
2017-03-01	0.27425

```
  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 sum_active,
 SUM(is_canceled) AS sum_canceled
 FROM status
  GROUP BY month
  ) SELECT month,
1.0 * sum_canceled/sum_active AS churn_rate FROM
status_aggregate;
```

Segment churn rates

- Using the code on the right, I was able to calculate segment churn rates
- The code is pretty similar to the previous one
- It is easy to see that churn rates in segment 87 are much, much higher than in segment 30. It may be a warning signal for the Codeflix to work more on balancing in segment 87
- On the other hand, it is worth working on segment 30 since it is very popular, and only a small portion of users are unsubscribing

month	churn_rate_87	churn_rate_30
2017-01-01	0.25179	0.07560
2017-02-01	0.32034	0.07335
2017-03-01	0.48587	0.11731

```
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;
```