code cademy

Calculating Churn Rates: CodeFlix

Learn SQL from Scratch Opeyemi Basiru Amodu 28.11.2018

Table of Contents

- 1. Get familiar with the company.
- 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?

- 2. What is the overall churn trend since the company started?
- 3. Compare the churn rates between user segments.
- Which segment of users should the company focus on expanding?

Take a look at the first 100 rows of data in the subscriptions table. How many different segments do you see?

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
6	2016-12-01	2017-01-19	87

SELECT

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

There are 3 months to calculate churn rates for, CodeFlix has a one-month cancellation policy.

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

Question 3.

You'll be calculating the churn rate for both segments (87 and 30) over the first 3 months of 2017 (you can't calculate it for December, since there are no subscription_end values yet). To get started, create a temporary table of months.

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

```
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 firt day,
      '2017-03-31' as last day
SELECT *
FROM months;
```

Question 4.

Create a temporary table, cross_join, from subscriptions and your months. Be sure to SELECT every column.

id	subscription_ start	subscription_end	segment	first_day	last_day
	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

```
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 firt day,
      '2017-03-31' as last day
cross join AS
(SELECT *
    FROM subscriptions
      CROSS JOIN months
SELECT *
FROM cross join;
```

Create a temporary table, status, from the cross_join table you created. This table should contain: id selected from cross_join month as an alias of first_day is_active_87 created using a CASE WHEN to find any users from segment 87 who existed prior to the beginning of the month. This is 1 if true and 0otherwise. is active 30 created using a CASE WHEN to find any users from segment 30 who

existed prior to the beginning of the month. This is 1 if true and 0otherwise.

				-
id	month	is_active_87	is_active_30	
1	2017-01-01	1	0	
1	2017-02-01	0	0	
1	2017-03-01	0	0	
2	2017-01-01	1	0	
2	2017-02-01	0	0	
2	2017-03-01	0	0	

```
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 firt day,
      '2017-03-31' as last day
  ),
cross join AS
(SELECT *
    FROM subscriptions
     CROSS JOIN months
status AS
  (SELECT id,
     first day AS month,
      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,
     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
SELECT *
FROM status;
```

Add an is_canceled_87 and an is_canceled_30 column to the statustemporary table. This should be 1 if the subscription is canceled during the month and 0 otherwise.

canceled during the month and 0 otherwise.				
month	is_active_8 7	is_active_ 30	is_cancel ed_87	is_cancel ed_30
2017-01-01	1	0	0	0
2017-02-01	0	0	1	0
2017-03-01	0	0	0	0
2017-01-01	1	0	1	0
2017-02-01	0	0	0	0
2017-03-01	0	0	0	0
	month 2017-01-01 2017-02-01 2017-03-01 2017-01-01	month is_active_8 2017-01-01 1 2017-02-01 0 2017-03-01 0 2017-01-01 1 2017-02-01 0	month is_active_8 is_active_3 2017-01-01 1 0 2017-02-01 0 0 2017-03-01 0 0 2017-01-01 1 0 2017-02-01 0 0	month is_active_8 is_active_ded_87 is_cancel ed_87 2017-01-01 1 0 0 2017-02-01 0 0 1 2017-03-01 0 0 0 2017-01-01 1 0 1 2017-02-01 0 0 0

```
WITH months AS
   (SELECT
     '2017-01-01' as first day,
     '2017-01-31' as last day
   SELECT
     '2017-02-01' as first day,
     '2017-02-28' as last day
     '2017-03-01' as firt day,
      '2017-03-31' as last day
  coss join AS
  ELECT *
    FROM subscriptions
     CROSS JOIN months
  atus AS
  (SELECT id,
    first day AS month,
      WHEN (subscription start < first day)
        AND (subscription end > first day OR subscription end IS NULL)
          AND (segment = '87') THEN 1
  ELSE 0
  D 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
  D AS is active 30,
  CASE
    WHEN (subscription end BETWEEN first day AND last day) AND (segment = '87') THEN 1
  ID AS is canceled 87,
  CASE
     WHEN (subscription end BETWEEN first day AND last day) AND (segment = '30') THEN 1
   ELSE 0
  ID AS is canceled 30,
  WHEN (subscription end BETWEEN first day and last day) AND (segment = 87) THEN 1
  ELSE 0
  ID AS is canceled 87,
  WHEN (subscription end BETWEEN first day AND last day) AND (segment = 30) THEN 1
  ELSE 0
  ID AS is canceled 30
  ROM cross join
  LECT *
  ROM status;
```

Create a status_aggregate temporary table that is a SUM of the active and canceled subscriptions for each segment, for each month.

The resulting columns should be:

and canceled subscriptions for each segment, for each month. The resulting columns should be: sum_active_87 sum_active_30 sum_canceled_87 sum_canceled_30), crosss (SELE), statu (SE
month	sum_active_8 7	sum_active _30	sum_cancel ed_87	sum_cancel ed_30	EI END A EI END A CA
2017-01-01	278	291	70	22	EI END A FROM), statu (SELE
2017-02-01	462	518	148	38	S S S FROM GROUF SELEC FROM
2017-03-01	531	716	258	84	

```
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 firt day,
      '2017-03-31' as last day
    s join AS
    ECT *
    FROM subscriptions
     CROSS JOIN months
    us AS
    ELECT 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
    LSE 0
    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
    LSE 0
    AS is active 30,
    WHEN (subscription_end BETWEEN first_day AND last_day) AND (segment = '87') THEN 1
    AS is canceled 87,
     WHEN (subscription end BETWEEN first day AND last day) AND (segment = '30') THEN 1
    AS is canceled 30
    cross join
    us aggregate AS
    ECT 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
    status
    JP BY month)
    CT *
    M status_aggregate;
```

Calculate the churn rates for the two segments over the three month period. Which segment has a lower churn rate?			
month	churn_87	churn_30	
2017-01-01	0.25(25%)	0.08(8%)	
2017-02-01	0.32(32%)	0.07(7%)	
2017-03-01	0.49(49%)	0.12(12%)	1

```
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 firt 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
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
ROUND(1.0 * sum canceled 87 / sum active 87, 2),
ROUND(1.0 * sum_canceled_30 / sum_active_30, 2)
FROM status aggregate;
```