codecademy

CODEfliX

Calculate Churn Rate with SQL

Judy Ping McCormick

31st DEC 2020

T1. Sample of Subscription table

id	subscription_start	subscription_end	segment
1	2016-12-01	2017-02-01	87
14	2016-12-01	2017-03-07	30

```
SELECT *
FROM subscriptions
LIMIT 100;
```

T2. Range

Can not calculate churn rate for December, no ‘subscription_end’ value, as Codeflix requires a minimum subscription length of 31 days, but can do for Jan, Feb and March.

MIN(subscription_start)	MIN(subscription_end)	MAX(subscription_start)	MAX(subscription_end)
2016-12-01	2017-01-01	2017-03-30	2017-03-31

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

T3. Create “Months” Table

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 first_day,
'2017-03-31' AS last_day)
SELECT *
FROM months;
```

T4. ‘Cross_join’ table = ‘subscription’ table join ‘months’ table

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

```
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)
SELECT * FROM cross_join;
```

T5. Create ‘status’ table from ‘cross_join’ table

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

```
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 segment = 87 AND (subscription_end > first_day OR subscription_end IS NULL)
THEN 1 ELSE 0 END AS is_active_87,

CASE WHEN (subscription_start < first_day) AND segment = 30
AND (subscription_end > first_day OR subscription_end IS NULL)
THEN 1 ELSE 0 END AS is_active_30  FROM cross_join )
SELECT * FROM status;
```

T6. add ‘is_canceled’ column to ‘status’ table

id	month	is_active_87	is_active_30	is_canceled_87	is_canceled_30
1	2017-01-01	1	0	0	0
1	2017-02-01	0	0	1	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 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 segment = 87
AND (subscription_end > first_day OR subscription_end IS NULL) THEN 1 ELSE 0 END AS is_active_87,
CASE WHEN (subscription_start < first_day)  AND segment = 30
AND (subscription_end > first_day OR subscription_end IS NULL)
THEN 1 ELSE 0 END AS is_active_30,

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_end BETWEEN first_day AND last_day)
THEN 1 ELSE 0 END AS is_canceled_30
FROM cross_join)  SELECT *  FROM status;
```

T7. ‘status_aggregate’ table

month	sum_active_87	sum_active_30	sum_canceled_87	sum_canceled_30
2017-01-01	278	291	70	22
2017-02-01	462	518	148	38
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 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 segment = 87
AND (subscription_end > first_day OR subscription_end IS N
ULL)
THEN 1 ELSE 0 END AS is_active_87,
CASE WHEN (subscription_start < first_day)
AND segment = 30 AND (subscription_end > first_day
OR subscription_end IS NULL)
THEN 1 ELSE 0 END AS is_active_30,
```

```
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_end BETWEEN first_day AND last_day)
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 * FROM status_aggregate;
```

T8. churn rate

month	churn_rate%_87	churn_rate%_30
2017-01-01	25	7
2017-02-01	32	7
2017-03-01	48	11

```
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 segment = 87
AND (subscription_end > first_day OR subscription_end IS N
ULL)
THEN 1 ELSE 0 END AS is_active_87,
CASE WHEN (subscription_start < first_day)
AND segment = 30 AND (subscription_end > first_day
OR subscription_end IS NULL)
THEN 1 ELSE 0 END AS is_active_30,
```

```
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_end BETWEEN first_day AND last_day)
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,
100 * sum_canceled_87/ sum_active_87 AS churn_rate_87,
100 * sum_canceled_30/ sum_active_30 AS churn_rate_30
FROM status_aggregate;
```

Conclusions:

1. Overall churn rates for segment 87 is much higher than segment 30, should we drop segment 87?
2. Churn rates for both segments 87 & 30 increase from January to March, has the momentum been lost as the year progresses?

month	churn_rate%_87	churn_rate%_30
2017-01-01	25	7
2017-02-01	32	7
2017-03-01	48	11