–First we calculated user churn rate for January of 2017

```
SELECT 1.0 *
(
 SELECT COUNT(*)
 FROM subscriptions
 WHERE subscription_start < '2017-01-01'
 AND (
  subscription_end
  BETWEEN '2017-01-01'
  AND '2017-01-31'
 )
) / (
 SELECT COUNT(*)
 FROM subscriptions
 WHERE subscription_start < '2017-01-01'
 AND (
  (subscription_end >= '2017-01-01')
  OR (subscription_end IS NULL)
 )
)
AS result;
```

–Our results gave us **0.126353790613718**

–Next we union January, February, March using a with cause to create a temporary table

```
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
```

–Now we have our 3 months and their date ranges as first and last in our view

– Next we're going to cross join our temporary Months table with our subscriber table to give us our subscribers first and last of each month

```
–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;
```

-Now i'm going to use a case when statement to derive 2 fields called is_active and is_cancelled using previous first_day and last_day fields and convert them to 1 as active and 0 as canceled

```
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
   ) 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)
SELECT *
FROM status;
```

–Our query returned the user_id, month, is_active as a 0 or 1, and is_cancelled as a 0 or 1 for each user.

–Next we will use a temp table (status_aggregate) and GROUP BY month and sum the values for active and canceled users for each month

```
FROM cross_join),
status_aggregate AS
(SELECT
        Month,
        SUM(is_active) as active,
        SUM(is_canceled) as canceled
FROM status
GROUP BY month)
SELECT *
FROM status_aggregate;
```

–This returned the following table

month	active	canceled
2017-01-01	276	35
2017-02-01	506	63
2017-03-01	667	158

–Now we'll replace the Select command in the previous query to return us the churn rate.

```
SELECT
        Month,
        1.0 * canceled/active AS churn_rate
FROM status_aggregate;
```

–This returned the following churn rate in the table below

month	churn_rate
2017-01-01	0.126811594202899
2017-02-01	0.124505928853755
2017-03-01	0.23688155922039

–From this query we can see that our churn had doubled in March which is something we should look deeper into

–Full SQL Query

```
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
       ) 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 active,
   SUM(is_canceled) AS canceled
 FROM status
 GROUP BY month
)
SELECT
 month,
```

```
 1.0 * canceled / active AS churn_rate
FROM status_aggregate;
```