

Clayton Cohn
23 April 2021

Part2: SQL Queries

I am going to do the best I can to explain my logic in the problems below (I haven't used SQL since I was an undergraduate, but I have no doubt that I will pick it right back up if given the opportunity).

1. How many users completed an exercise in their first month per monthly cohort?

```
SELECT COUNT(user_id), MONTH(created_date)
FROM users
WHERE user_id IN
    (SELECT user_id,
        FROM exercises
        WHERE MONTH(exercise_completion_date) = MONTH(users.creation_date)
        GROUP BY user_id)
GROUP BY MONTH(created_date);
```

My logic was to first group the exercises by user, then take the MIN date (first exercise) of each user.

I then wanted to return a list of users whose creation_dates were in the same month as their MIN exercise_completion_dates.

Grouping by creation_date, I would return the counts of users in each creation month who also had the same MIN exercise_completion_dates months.

I am not sure if I can compare two columns in different tables without a JOIN, so I would have to look more into this as well.

2. How many users completed a given amount of exercises?

```
SELECT COUNT(exercises.exercise_completion_date), COUNT(users.user_id)
FROM users
INNER JOIN exercises
ON users.user_id = exercises.user_id
GROUP BY user_id;
```

For this one, my logic was to first join the table on user_id, then group by user_id and count the number of exercises for each user.

After that, I wanted to return a list of the number of exercise completions with the corresponding number of users who completed that many exercises.

3. Which organizations have the most severe patient population?

```
SELECT TOP 5 Providers.organization_id
FROM Providers
INNER JOIN Phq9
ON Providers.provider_id = Phq9.provider_id
WHERE COUNT(Phq9.scores) >= 20
GROUP BY organization_id
ORDER BY COUNT(Phq9.scores) >= 20 DESCENDING;
```

For this one, I first wanted to join the tables on provider_id.

Next, I wanted to group by organization_id.

After they were grouped, I wanted each organization's count of how many scores greater than or equal to 20.

I then ordered the results by the number of scores greater than 20 in descending order, and returned the top 5 organization_ids.