## Project Summary
Our project allows users to find hikes/trails in Canada based on their preferences. Users can also read and write reviews and give ratings to the hikes they go on. Additionally, users can view other users who have gone on a specific hike, see the average difficulty of all hikes for a specific season, find hikes that are deemed safe (ie. less than 1 safety hazard), find hikes in good condition, and users who have hiked every hike.

## Schema Changes
We added relations for Needs, View, Saves, and Has. We also added proper foreign keys to Hike1, Hike2, and Location2. We also added the missing foreign key handling to many relations. We did all of this based on the feedback we received from our TA. No other changes were made.

## SQL Queries
All of the below queries are found in the appService.js file
1. Insert line 208
2. Update line 248
3. Delete line 278
4. Selection line 331
5. Projection line 360
6. Join 385
7. Aggregation with GROUP BY line 404
```
SELECT h2.Season, AVG(h1.Difficulty) AS AvgDifficulty
        FROM Hike2 h2
        JOIN Hike1 h1
          ON h1.Kind = h2.Kind
         AND h1.Distance = h2.Distance
         AND h1.Elevation = h2.Elevation
         AND h1.Duration = h2.Duration
        GROUP BY h2.Season
        ORDER BY h2.Season
```

8. Aggregation with HAVING line 435
```
SELECT h2.Name
        FROM Hike2 h2
        LEFT JOIN Has h ON h2.HikeID = h.HikeID
        GROUP BY h2.HikeID, h2.Name
        HAVING COUNT(h.SafetyHazardID) < 1
        ORDER BY h2.Name
```

9. Nested Aggregation with GROUP BY line line 461
```
SELECT h2.Name, h2.Season, h2.TrailCondition
        FROM Hike2 h2
        GROUP BY h2.HikeID, h2.Name, h2.Season, h2.TrailCondition
        HAVING h2.TrailCondition > (
```

```
            SELECT AVG(h2b.TrailCondition)
            FROM Hike2 h2b
            WHERE h2bSeason = h2.season
        )
```

## 10. Division line 480

```
SELECT u.Name
        FROM AppUser u
        WHERE NOT EXISTS (
            SELECT h.HikeID
            FROM Hike2 h
            MINUS (
                SELECT s.HikeId
                FROM Saves s
                WHERE s.UserID = u.UserID
            )
        )
```