

Ben Milligan
Database Program1

I did not make any changes to the database schema given to me. The reason why I chose to do this was because I found that by taking the inverse of the brother sister relations you could extract much more information from the BasicRelationship table than if left be. If I had to change it I would have probably added the extended BasicRelationships and their inverses to the BasicRelationship table in order to make it easier to query that table when doing the 2nd, 3rd, and 4th algorithms.

My program file firstly connects to the database and then establishes the tables in the database if they already are not established. I then ask the user which option they would like to choose. If they choose 1 I let them enter in a person, if they choose two then I let them enter two people and if those two people exist then I allow them to enter a relationship between them. I then enter that relationship and the inverse into the BasicRelationship table. If the user enters 3 then I let them define a relationship and enter the corresponding information into the Relationship and RSteps table. If they enter 4 I run the 1st algorithm. If they enter 5 I run the 2nd algorithm. If they enter 6 I run the 3rd algorithm. If they enter 7 I run the final algorithm. If they enter 8 then the program terminates.

For algorithm 1 I prompt the user to input their first and last name and then I validate that by using a query to find matching people in the Person table. I then did a query on the BasicRelationship table to find all the names of the brothers and sisters and their BasicRelationship to the person the user entered. If it's a brother then I output that, if it's a sister I output that, otherwise I output that the person entered has no siblings. Next, I query the BasicRelationship table for the entered persons father and mother. If I find a father then I output that relationship, and if I find a mother I output that relationship. If I don't find either, then I output that the person does not have a father or mother. The last step I perform I query the Person table to find the gender of the person entered. Based on that gender I find if it's a husband or wife. If the gender is Male, I then perform a query on the BasicRelationship table to find all tuples where the lastName1, firstName1 is equal to the person entered by the user and the relationship is set to H for husband, and return the second person in that relationship (the wife). If that returns something then I output the husband relationship. Likewise, if the gender is female then I query for tuples where the lastName2, firstName2 is equal to the person entered and if that returns something I output the husband wife relationship. Otherwise, that person has no husband.

For algorithm 2 I let the user enter in the first and last name of the person and I then validated that. I then let the user input the name of the relationship they want. I then queried the Relationship table to get the RelationCount and RelationshipName. Based on the RelationCount I did a loop to loop that many times and count down. I then used that count of the loop as a key for the RSteps table and queried for the tuples where the Steps was equal to the number and the RelationshipName was the same as queried from the Relationship table. I then, based on the BasicRelationship queried for the siblings. I was not able to complete algorithm 2. But if I had I would have used recursion for the steps. I would have stuck the results of querying the RSteps table into a list and performed a Breadth First Search to get each step and the relations related to it.

I was not able to complete algorithm 3. I would have done a breadth first search on that person and found all people related to him and listed all the steps.

I was not able to complete algorithm 4. But for algorithm 4 I would have made a loop that only goes to 3. Inside the loop I would have queried for all people related to the person entered, and then set the values to query for next to those results and kept querying. Essentially I would get those related to person 1 and get the people related to those people and then the people related to them and if in that set the second person didn't appear then there is not a relationship between them that is 3 basic relationships away.