CSE 4/560 PA 3: SQL Query - Recursion

Due 23:59 10/28/2020 EST

October 21, 2020

This is an individual programming assignment for writing SQL queries. There are 7 problems with 15 points in total. **Please note that academic integrity is strictly implemented and any violation will lead to a F grade in this course.**

1 No Specific Database

1.1 Problem 1, 1 point

Use recursion to find out the factors of 999 and the quotient. The result of query is similar to following table:

```
n  |d  |q       |
---|---|--------|
999|  1|999.0000|
999|  3|333.0000|
999|  9|111.0000|
999| 27| 37.0000|
999| 37| 27.0000|
999|111|  9.0000|
999|333|  3.0000|
999|999|  1.0000|
```

1.2 Problem 2, 2 points

Given the definition of a sequence:

- if $n \leq 1$, then $f(n) = n$
- if $n > 1$ and $f(n-1)$ is an odd number, then $f(n) = f(n-1) + f(n-2)$
- if $n > 1$ and $f(n-1)$ is an even number, then $f(n) = f(n-1) + f(n-2) + 1$

Write a query to generate the sequence until 100. The result of query is similar to following table:

```
n |f(n)|
--|----|
 0|   0|
 1|   1|
 2|   1|
 3|   2|
 4|   4|
 5|   7|
 6|  11|
 7|  18|
 8|  30|
 9|  49|
10|  79|
```

1.3 Problem 3, 2 points

Given m, n in the range of 1 to 9 as integer, write a query to generate a table to show the result of multiplication $m * n$. The result of query is similar to following table:

```
m|n|m*n|
-|-|---|
1|1|  1|
1|2|  2|
1|3|  3|
...
1|9|  9|
2|1|  2|
2|2|  4|
...
9|7| 63|
9|8| 72|
9|9| 81|
```

2 Database: Employees

From problem 4 use the employees database, which is imported from PA0, to answer the query. Note that we can treat the date 9999-01-01 as ”current” in the value of to_date columns in this database.

2.1 Problem 4, 3 point

Assuming a title is a node and a promotion is an edge between nodes. e.g. A promotion from Engineer to Senior Engineer means their is a path from Node ’Engineer’ to Node ’Senior Engineer’. Also, we treat the years from beginning of a title until promotion as the distance between nodes. e.g. An employee started as an Assistant Engineer from 1950-01-01 to 1955-12-31 then be promoted to Engineer on 1955-12-31. Then there is an edge between node ”Assistant Engineer” to ”Engineer” with distance 6.

Calculate the average distance of all possible pair of titles and ordered by source node. To simplify the problem, there is no need to consider months and date when calculating the distance. Only year is required for calculating the distance. Besides, we can assume the distances of any given pair is less than 100.

Sort the result by src then dst. The expected result is shown as follow:

```
src                    dst            years
Assistant Engineer     Engineer       7.7926
Assistant Engineer     Manager        20.5266
...
Engineer               Manager        12.7340
...
```

As the table shows, the average distance between node ”Assistant Engineer” and node ”Engineer” is 7.7926. We add it with the distance between ”Engineer” to ”Manager”, which is 12.7340, to find out the distance between ”Assistant Engineer” to ”Manager” is 20.5266.

3 Database: salika

From problem 5 import the database from salika-db.zip and answer following questions.

3.1 Problem 5, 3 point

By looking into address table, we know there is a column ’location’ for each address. Given following code to calculate the distance between two addresses in miles:

```
ST_Distance_Sphere(a1.location, a2.location) * .000621371192
```

We define any two customers lives within 20 miles to each other as neighbor. e.g. Customer 1 lives 10 miles aways from Customer 2, and Customer 2 lives 15 miles away from Customer 3. We will treat customer 1, 2, and 3 as a cluster as represent the cluster with the lowest customer id. The expected result is shown as follow:

```
c1 |c2 |distance          |
---|---|------------------|
  2| 51| 8.386184629235382|
...
 53|147|14.417206471432088|
 53|194| 33.35065469262307|
 53|396|13.867384588845791|
 53|404|15.975982268727984|
 53|574| 33.42354390047083|
```

We can see the there is a cluster which is represent by customer with id 53. Although customer 194 is 33 miles away from customer 53, since there is a neighbor path from customer 53 to customer 147 in 14 miles then customer 147 to customer 194 in 18 miles, we treat them as the same cluster.

Few things to consider when writing this query:

1. Use $-180 < ST\_X(location) \leq 180$ to filter out the bogus instances.
2. Use $ST\_Y(location) \leq 90$ to filter out the bogus instances.
3. If there are difference distances between two customers, take the shortest one.

4 Database: Flight

From problem 6 and 7 import the flight database from flight-db.zip and answer following questions.

4.1 Problem 6, 1 point

Find all the possible source and destination with any number of flights. Sort the result by src and dst. The result of query is similar to following table:

```
src|dst|
---|---|
BOS|ATL|
BOS|BUF|
...
BUF|ATL|
BUF|BUF|
...
BUF|SFO|
DAL|ATL|
...
```

4.2 Problem 7, 3 points

For each src, find the destinations can be reached within 2 intermediate stops and list out the minimum cost. The result of query is similar to following table:

```
src|dst|cost|
---|---|----|
BOS|DAL| 139|
BOS|JFK| 265|
...
BUF|EWR| 197|
BUF|FLL| 239|
```

5 Offline Grader and Expected Output

To be released on Oct. 24.

6 Submission

Failure to comply with the submission specifications will incur penalties for EACH violation.

- What to submit: A zip file has to be submitted through the 'submit_cse460' (if you are CSE460 student) or 'submit_cse560' (if you are CSE560 student) submit script by 23:59 10/28/2020 EST. Only zip extension will be accepted, please **don't** use any other compression methods such as tar or 7zip. You can submit multiple times, note that **only the last submission** will be kept on the server. **Even one minute late is still a late submission. No late submission will be accepted**.
- Zip file naming: Use *ubit_pa3* (**NO SPACE!**) for the filename, for example: *jsmith_pa3.zip*, where *jsmith* is the ubit of submitter. The project is an **INDIVIDUAL** work, so everyone needs to submit ONE zip file.
- Structure of zip file: On unzipping the zip file, there should be a folder named with your ubit *ubit_pa3*, under the folder *ubit_pa3*, there should be 7 SQL files, starting from *q1.sql*, *q2.sql ...* ,*q7.sql* which correspond to SQL query for each problem.
- Follow steps below to submit your work:
  1. copy your file to server, note that there is a dot at the end of the command:
     `scp jsmith_pa3.zip jsmith@timberlake.cse.buffalo.edu:.`
  2. login to server:
     `ssh jsmith@timberlake.cse.buffalo.edu`
  3. submit your file (if you miss this step, we won't be able to see your work and you will NOT receive any score):
     - For CSE 460 students:
       `submit_cse460 jsmith_pa3.zip`
     - For CSE 560 students:
       `submit_cse560 jsmith_pa3.zip`