CSE 4020 – Database Systems

Spring 2023

Activity 3: Advanced SQL

Total Points: 25

Date Assigned: Friday, Mar 24 , 2023 **Due Date:** Saturday, Apr 1, 2023

Instructions: Please submit your work on Canvas as a Jupyter Notebook `ipynb` file named `cse4020_yourname_activity3.ipynb`. Make sure to use Markdown to include headings and the question numbers in your notebook. Also, ensure that the corresponding output for each question is displayed.

Key Concepts Demonstrated

- Executing Advanced SQL Statements on a database
  - Coding in a Jupyter notebook
  - Creating and verifying a trigger on a database table
  - Issuing SQL statements on a cloud-based database (AWS RDS)

For this activity, you will work with the `bankdb` schema that was created on Activity 1 and populated with data on Activity 2. The schema has been included below for your reference:

```
branch(branch-name, branch-city, assets)
customer (customer-name, customer-street, customer-city)
account (account-number, branch-name, balance)
loan (loan-number, branch-name, amount)
depositor (customer-name, account-number)
borrower (customer-name, loan-number)
```

1. Imagine that the bank has received a portion of a government bailout totaling 10 million dollars. One requirement for banks receiving this bailout money is that they remain their current size (i.e., large branches are branches with assets >= 3 million dollars and small branches have assets < 3 million dollars).The CEO of the bank, Nate Richmond, would like to divide this bailout money equally among the small branches at the bank.

   In a Jupyter Notebook, write code in Python, Java or C (or a language that works in Jupyter) to help DBA Christopher Hemsworth complete the following:

   (a) (5 points) Execute a single query to return and print out the total assets owned by the bank (i.e., a cumulative total of the assets at all branches)

   (b) (10 points) Create a trigger named **`maintain_branch_size`** on your branch table that ensures that a small branch does not become a large branch (i.e., that its total assets never gets to 3 million dollars). Whenever an update to the assets of a small branch is attempted that would cause it to become a large branch, your trigger should store this excess money in a temporary table (e.g. bailout_cache) to be later divided equally among the existing large branches, capping the small branch at 2,999,999.99 in total assets.

   (c) (5 points) Execute a query that attempts to divide the bailout money ($10M) among the small branches as a test of your trigger.

   (d) (3 points) Execute a query that divides any excess money from the bailout (money in your bailout_cache table) among the large branches

   (e) (2 points) Execute a single query to return and print out the total assets owned by the bank (i.e., a cumulative total of the assets at all branches) after all changes above have been made.