CSE 4020/5260 – Database Systems

Spring 2023

Activity 2: SQL

Total Points: 25

Date Assigned: Monday, Feb 27, 2023 **Due Date:** Friday, Mar 10, 2023

Instructions: Please submit your work on Canvas as a Jupyter Notebook `ipynb` file named `cse4020_yourname_activity2.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.

Use only packages approved by the instructor when completing this activity. For example, do not use advanced Python packages such as pandas or SQLAlchemy.

Key Concepts Demonstrated

- Executing SQL Statements on a database
  - Inserting data into a database
  - Coding in a Jupyter notebook
  - Writing program code that reads CSV file(s) and writes the data to a database
  - Issuing SQL statements on a cloud-based database (AWS RDS)

Figure 1 below shows a snapshot of a set of CSV files provided on Canvas (data/banking_csvs.zip) for the banking schema. Download the files to your computer.

Figure 1: Snapshot of CSV files containing sample data for the banking schema

customer

customer_name	customer_street	customer_city
Jones	Main	Harrison
Smith	Main	Rye
Hayes	Main	Harrison
Curry	North	Rye
Lindsay	Park	Pittsfield
Turner	Putnam	Stamford
Williams	Nassau	Princeton
Adams	Spring	Pittsfield
Johnson	Alma	Palo Alto
Glenn	Sand Hill	Woodside
Brooks	Senator	Brooklyn
Green	Walnut	Stamford
Jackson	University	Salt Lake
Majeris	First	Rye
McBride	Safety	Rye

branch

branch_name	branch_city	assets
Downtown	Brooklyn	900000
Redwood	Palo Alto	2100000
Perryridge	Horseneck	1700000
Mianus	Horseneck	400200
Round Hill	Horseneck	8000000
Pownal	Bennington	400000
North Town	Rye	3700000
Brighton	Brooklyn	7000000
Central	Rye	400280

account

account_number	branch_name	balance
A-101	Downtown	500
A-215	Mianus	700
A-102	Perryridge	400
A-305	Round Hill	350
A-201	Perryridge	900
A-222	Redwood	700
A-217	Brighton	750
A-333	Central	850
A-444	North Town	625

depositor

customer_name	account_number
Johnson	A-101
Smith	A-215
Hayes	A-102
Hayes	A-101
Turner	A-305
Johnson	A-201
Jones	A-217
Lindsay	A-222
Majeris	A-333
Smith	A-444

loan

loan_number	branch_name	amount
L-17	Downtown	1000
L-23	Redwood	2000
L-15	Perryridge	1500
L-14	Downtown	1500
L-93	Mianus	500
L-11	Round Hill	900
L-16	Perryridge	1300
L-20	North Town	7500
L-21	Central	570

borrower

customer_name	loan_number
Jones	L-17
Smith	L-23
Hayes	L-15
Jackson	L-14
Curry	L-93
Smith	L-11
Williams	L-17
Adams	L-16
McBride	L-20
Smith	L-21

1. (10 points) In a Jupyter Notebook, write code either in Python, Java or C to connect to your AWS RDS instance and insert the data from each of the CSV files into the RDS database that you created on AWS on Activity 1. Display the number of rows affected for each table (1 cumulative count for all inserts per table - example: 15 rows inserted for table customer). (Hint: If your data types do not match the given data, you may use the following SQL statement to update your data types: `ALTER TABLE table_name MODIFY column_name NEW_DATA_TYPE;`)

2. Write code to issue queries on your database and display your results for the following scenarios. **Make sure to print out a tabulation of your branch table ordered by assets in descending order to show the data before and after the changes.**

   A branch of the bank is considered a large branch if it has assets >= 3 million dollars. Imagine that COVID-19 affected the bank terribly and the branches with assets >= 3 million dollars had to liquidate 12.5% of their assets to help the smaller branches (branches with assets < 3 million dollars).

   (a) (3 points) Issue a single query that returns the branch_name, branch_city, and assets of the branches that will be liquidated based on the liquidation formula above.

   (b) (3 points) Issue an update query that reduces the assets for the large branches by the liquidation formula above.

   (c) (4 points) Issue an update query that divides equally the total amount liquidated among the small branches.

   (d) (5 points) Use a single query to fetch and display a tabulation showing the customer_name, customer_city, and account_number of all customers who have accounts at a small branch. Include the branch_name and assets in your results and order your results by assets in descending order

   Expected Format of Output:

```
customer_name  customer_city  account_number branch_name    assets
Xxxxxxx        Xxxxxxx        x-xxx          Xxxxxxx        x,xxx,xxx.xx
```

