Bank's Portal Report

Link to Source Code: https://github.com/rafmsq/banks_portal.git

Project Description

We are given a starter code in Python with a platform that allows us to run on a Python HTTP Server. Our task is to create a MySQL Server and to complete the Database, so that the platform would be functionable. The Idea of the Project is to have a web platform for bank tellers to be able to manage bank accounts.

Creating and Using Database

First, I had to create a MySQL connection with a hostname, port, username, and a password. I also had to create a connection name for the MySQL connection. After that was done, I had to create a database for the bank's portal. I first started with the code, (CREATE DATABASE "banks_portal";). After I created the database, I then used the code, (USE "banks_portal";) to be able to create the database and to use the database.

Creating a Table with Attributes

I used the codes to create a table with the attributes since the table is what holds all the attributes. As the requirements were to create a table named after accounts and transactions. They both have different sets of attributes with different string types. As the accounts table needs (accountID, ownerName, owner_ssn, balance, and account_status) attributes to create that table. The transactions table needs (transactionID, accountID, transactionType, transactionAmount) attributes to create that table.

INSERTING VALUES TO THE TABLE

In this portion, to insert any information to the tables that I created in the previous step, I must use the code that I provided in the Script file. The values that were provided followed the same steps as how the attributes were written in the tables. So, for the accounts table, the values matched with (accountID, ownerName, owner_ssn, balance, and account_status). For the Transactions table, the values matched with (transactionID, accountID, transactionType, and transactionAmount). Doing this in the incorrect order wouldn't allow you to input values due to the string types that were placed for the attributes.

REVIEWING INSERTED VALUES IN TABLE

To review and see the values that I inserted into both tables, I used (SELECT * FROM banks_portal.accounts;) and (SELECT * FROM banks_portal.transactions;). Both codes show both tables that were created with the values that I had to input. I added this step since I wanted to double check if every single value was placed correctly and not missing any information before I proceed with the database.

CREATING STORED PROCEDURES WITH PARAMETERS

Three Stored Procedures were created with different Parameters and different named procedures. The first Stored Procedure was to create procedure amountTransactions with parameters for in accountID and selecting from the transactions table. The second stored procedure was to create procedure deposit with parameters for in accountID and transactionsAmount. But this procedure had to be inserted into transactions table and to update the accounts table accordingly. The third stored procedure was to create procedure withdraw with parameters for in accountID and transactionsAmount. It also had to be inserted into transactions table and to update the accounts table accordingly.

CONNECTING/IMPORTING MYSQL TO PYTHON

The first thing that had to be done was installing python to my computer. I had to put python in command prompt and then download. After that was done, I opened the portalDatabase python file script since that is where we input all our information of the database. The information we needed was the first step which was creating a connection to the database where it shows the host name, port, username, and password. That information we input it into portalDatabase python script, so it will be able to read that database that I created. After running portalDatabase python script, then we can head into command prompt to write out the codes to import MySQL and to connect to our database. In the command prompt we must write these certain commands: "import mysql.connector", "from mysql.connector import Error", and "connection = mysql.connector.connect(host=127.0.0.1, port=3306, database= "banks_portal", user=root, password= "RafiWalton1")". All of this allows you to have a connection.

RUNNING SERVER FOR WEB BROWSER

After creating a connection with MySQL and python. I had to open the portalServer python script and run the module after completing connection with MySQL and python. Running the portalServer python script allows you to have the database server ready on your web browser as a HTTP server. After running the script, I went to my local browser and typed "localhost:8000/" and that's where I got my database and saw all the accounts that I created from the database; including all the values I inserted to the table. To also verify, I clicked on add account and it opened allowing for bank tellers to add new accounts.

INSERTING ALL ACCOUNTS TABLE THROUGH PYTHON SOURCE CODE

First, I opened portalDatabase Python source code, and went under where it says “addAccount”. This is where if I add an account under this code, it will also show up under All Accounts without update the table values in MySQL. I gave it two codes that were like the ones in the step where we had to create attributes and insert values. After finishing portalDatabase source code, I ran the module. I went into command prompt, and having it already connected, I just had to execute the code that I inserted into portalDatabase source code. After executing the code on command prompt, I went into my local browser and inserted “localhost:8000/. There I saw the new account that I added through the portalDatabase source code.