

DTSC660: Data and Database Management with SQL

Module 6

Assignment 5

Purpose

For this assignment, you will utilize the banking database that you created in Assignment 4. You will be performing some advanced queries on the same dataset. To complete this assignment, ensure that you have access to the data and banking database from Assignment 4 before continuing.

Each question is all or nothing. Graders will not attempt to correct or interpret malformed SQL queries.

You will be responsible for testing your code on the provided data set before submission. Each question will be graded based on whether or not it generates the correct output and addresses all requirements specified in the question. Extraneous columns will not count against you as long as correct results are obtained.

Submission

You will submit a total of **1** sql file to CodeGrade. Each file must use the postgres standards taught in the course. Use of other flavors of SQL such as T-SQL will result in an automatic 0 for the assignment. Do not submit files as archives (ZIP) files.

- **File 1:** You must submit a SQL document called <LastName>_Assignment5. This document must include ALL queries requested in the instructions below.
- You will submit the file to the Assignment 5 folder.
- **You will have one attempt to complete this assignment. Additional attempts will not be accepted.** Test carefully. It is expected that if you have questions or difficulties with any portion of this assignment that you utilize the assignment discussion board or email the GAs to gain clarity (dtsc_ga_660@eastern.edu).

PLEASE NOTE: Each of the questions in this assignment should return at least 1 and no more than 4 results (records). Do not use LIMIT to force a result of 4 - if you have more than 4 results, there is a problem with the logic of your query. *Beware of cartesian products.*

Questions

There are seven questions total in this assignment. Please see the rubric on the last page for grading rules.

- **Question 1:** Write a query to find the cust_ID and customer name of each customer at the bank who only has at least one loan at the bank, and no deposit accounts.
- **Question 2 (MUST INCLUDE SUBQUERY):** Write a query to find the cust_ID and customer name of each customer who lives on the same street and in the same city as customer '12345'. Include customer '12345' in your query results.
Tip: Do not hardcode the address for customer 12345. You want to make sure this query continues to return the intended results if customer 12345 moves to a new address or even changes their name.
- **Question 3 (No Joins Allowed):** Write a query to find the name of each branch that has at least one customer who has a deposit account in the branch and who lives in "Harrison". No branch name should be repeated in the results.
- **Question 4 (MUST INCLUDE SUBQUERY):** Write a query to find the name of each customer who has a deposit account at every branch located in "Brooklyn".
Tip: Customers returned in the results may have more than one account at a Brooklyn branch, and they might also have accounts at branches outside Brooklyn. The key is that they must have at least one account at every Brooklyn branch. Do not hardcode the names of the Brooklyn branches - this query should still work if a new branch opens in Brooklyn or if a branch changes its name.
- **Question 5 (MUST INCLUDE SUBQUERY):** Write a query that finds the account number of all customers whose deposit account is in the city where they live. Include the customer's id, branch city, and customer city.
Tip: The "home city" of a deposit account is defined as the city of the branch the account is assigned to. In order to find the correct customers for this query, you will need to compare the customer's city to the branch city for the customer's account(s).
- **Question 6:** Write a query that returns every customer that has a loan at Yonkahs Bankahs. Include the customer name and branch name for verification.
- **Question 7:** Write a query that returns the name and loan number of all customers with loan balances (amounts) higher than \$5,000.00.

*****GRADING RUBRIC ON NEXT PAGE*****

This assignment will be graded on the following rubric. Remember that questions are ALL OR NOTHING. Incorrect syntax, extraneous results, or incorrectly addressing all question requirements will result in loss of points for that question. Graders will NOT attempt to correct malformed sql code. :

Question Number	Points
1	10
2	15
3	15
4	15
5	15
6	15
7	15
Total	100