CSCI 5408

DATA MANAGEMENT AND WAREHOUSING

LAB ASSIGNMENT - 3

Banner ID: B00948977

Git Assignment Link :
https://git.cs.dal.ca/sukumaran/csci5408_f23_b00948977_balaji_sukumaran

Table of contents

Problem Statement 1: Design a banking application database with the customer's details (minimal attributes - name, mailing address, permanent address, primary email, primary phone number), account details (minimal attributes – account number, account balance) and account transfer details (minimal attributes - account number, date of transfer, recipient name, status(varchar value waiting/accepted/declined)).

Created the banking database tables. To simulate transaction environment

Line #1: create table command for customer_details

Line #2: create table command for account_details

Line #3: create table command for transfer_details

Tables created successfully.

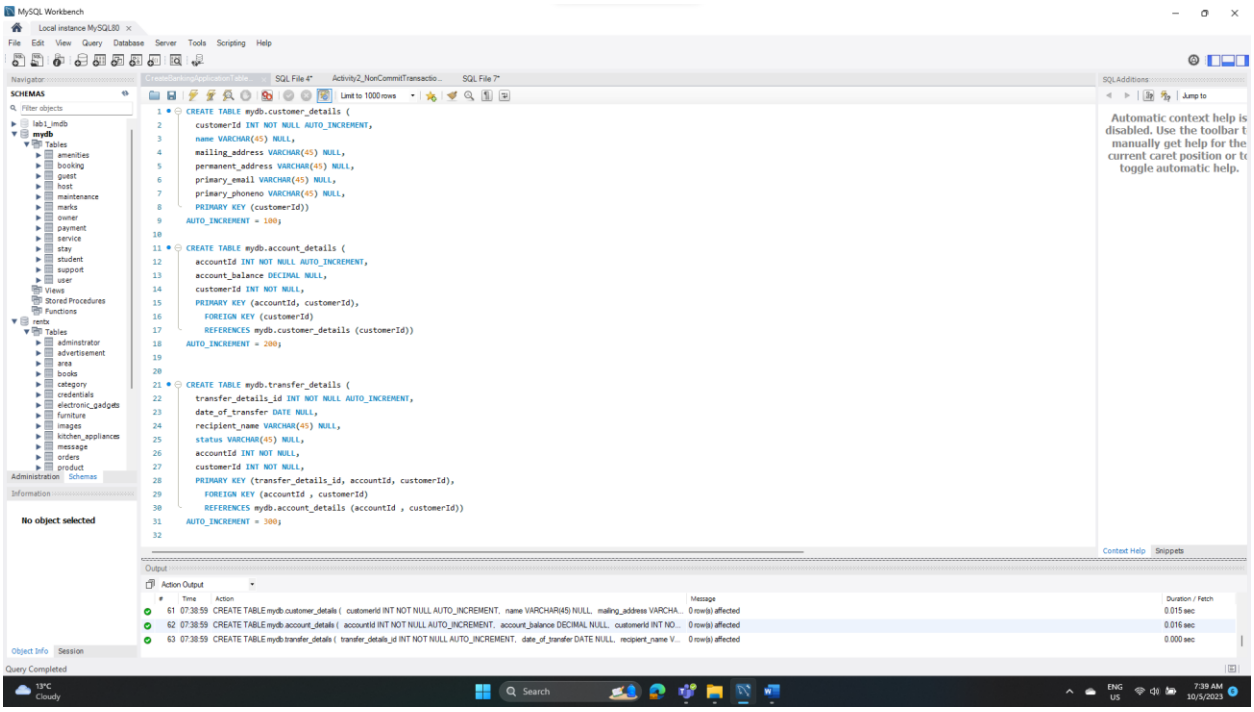


Figure 1: create statement for customer_details, account_details, transfer_details

Inserted sample data for simulating transactions

Line #3 till #11: Insert statements for customer_details

Line #13 till #20: Insert statements for account_details

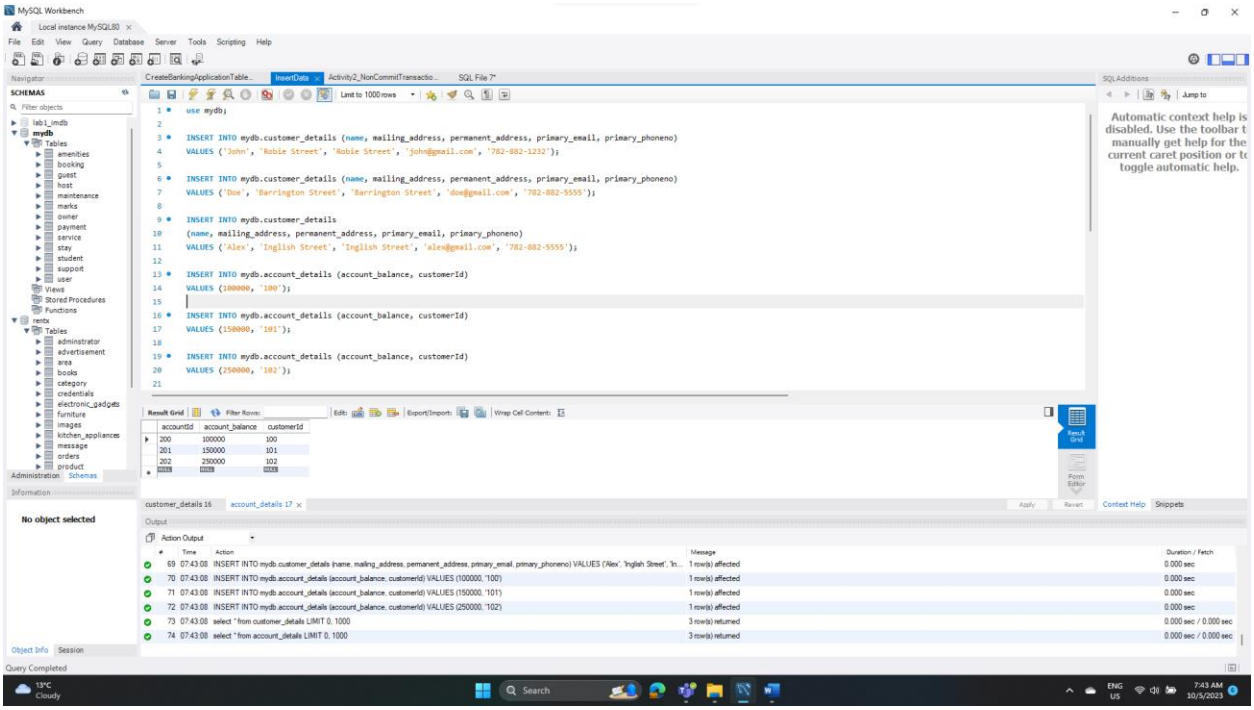


Figure 2: Insert Statement for customer_details and account_details

Records in customer_details table

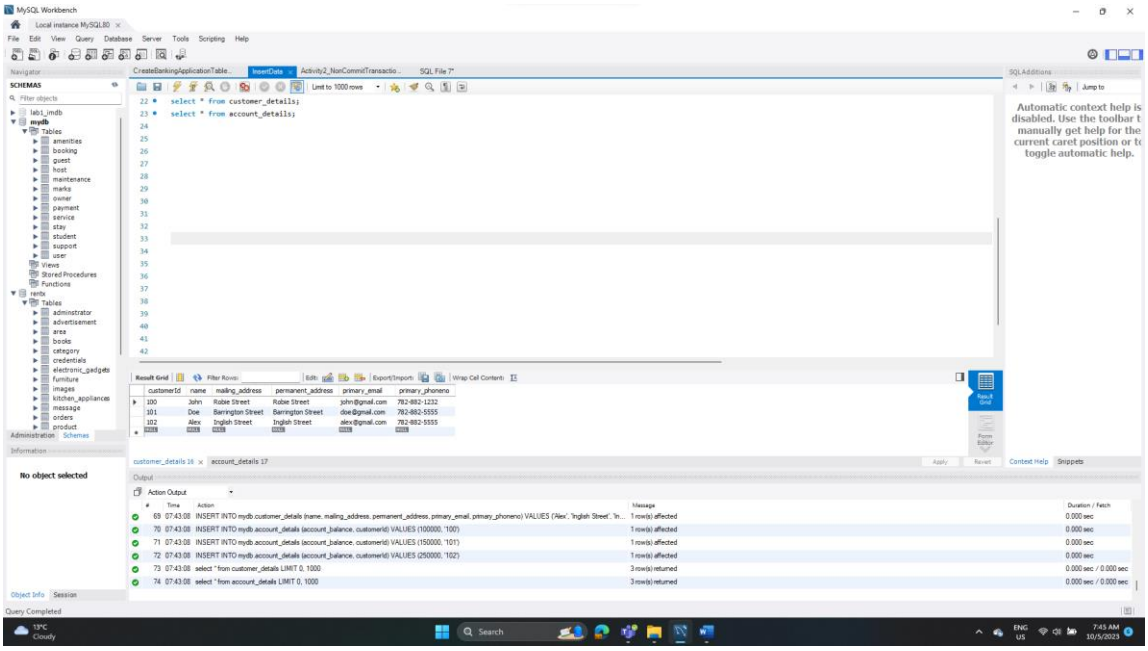


Figure 3: Contents of customer_details table

Records in account_details table

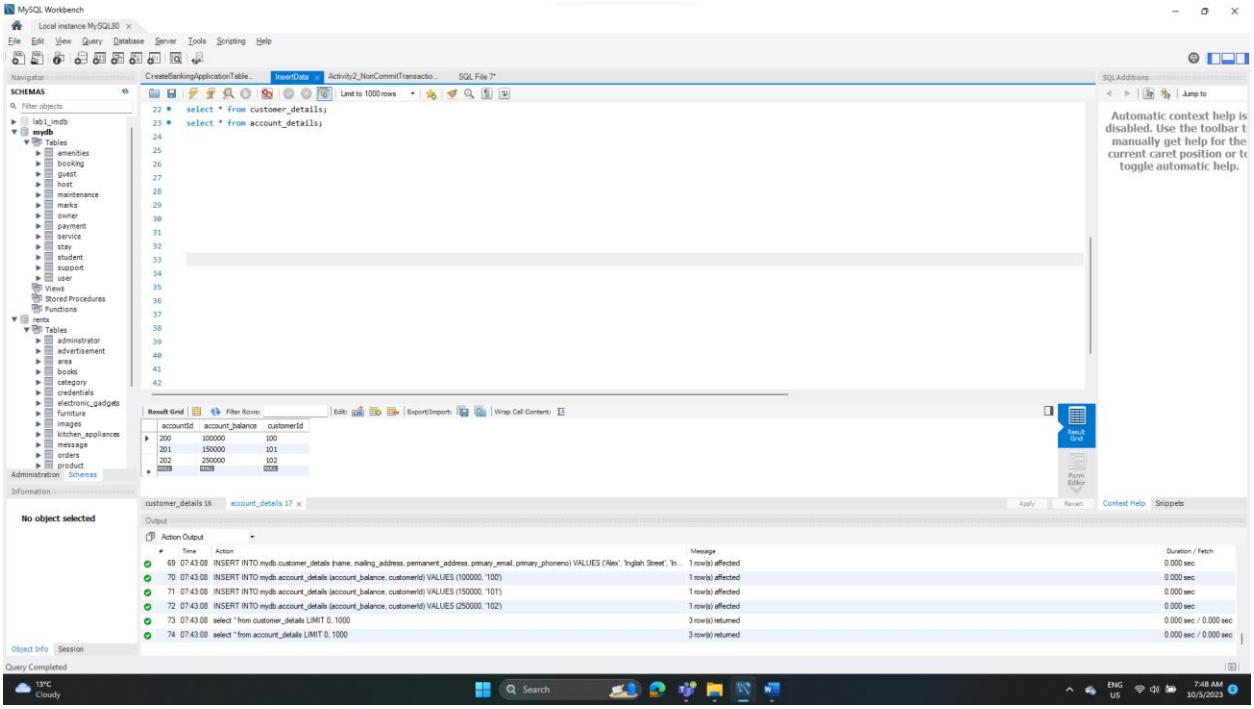


Figure 4: Contents of account_details table

Problem Statement 2: Create a transaction environment where on every account debit, the record in the account details table is updated but not committed. Upon this update, insert a record in the account transfer details table with waiting state.

Line #4: Setting Autocommit to OFF, to keep the transaction in a limbo state when this query page is executed

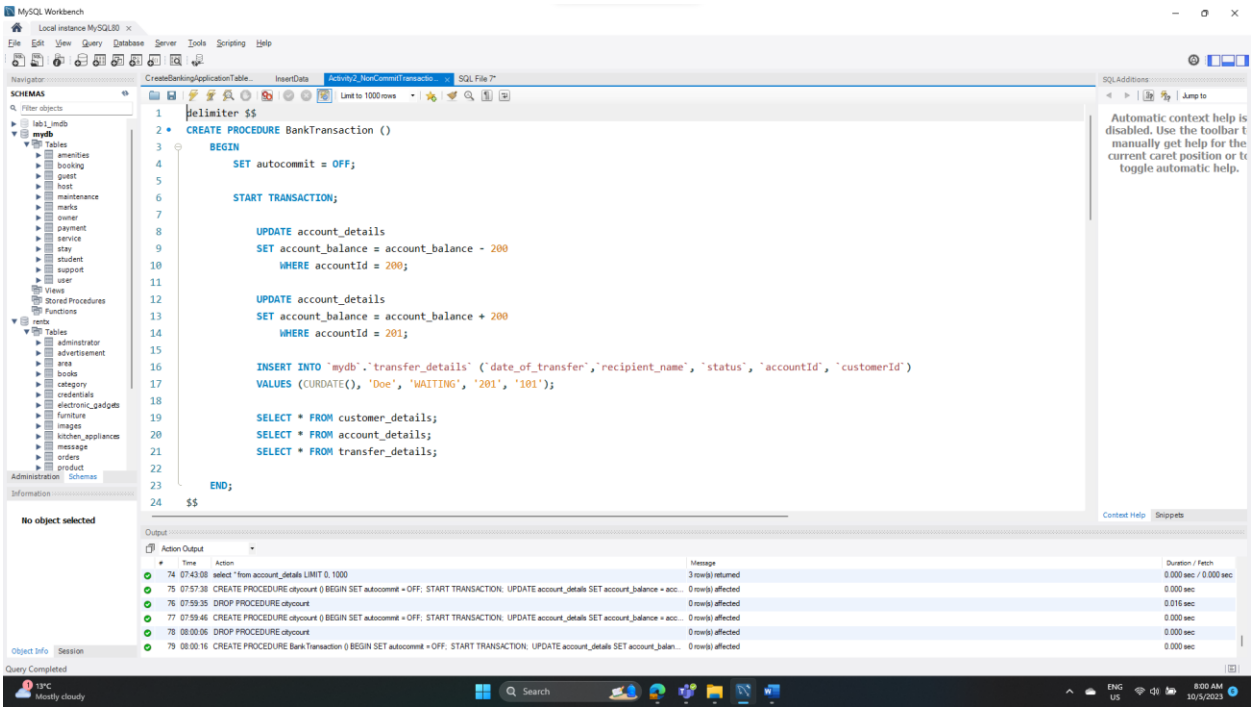
Line #6: Start of transaction

Line #8: Debited $200 from John's account

Line #12: Transferred that $200 to Doe's account

Line #16: INSERTED into transfer_details table with a WAITING state indicating this state is not committed

Line #19,20,21: to see the current state of the customer_details, account_details and transfer_details table.

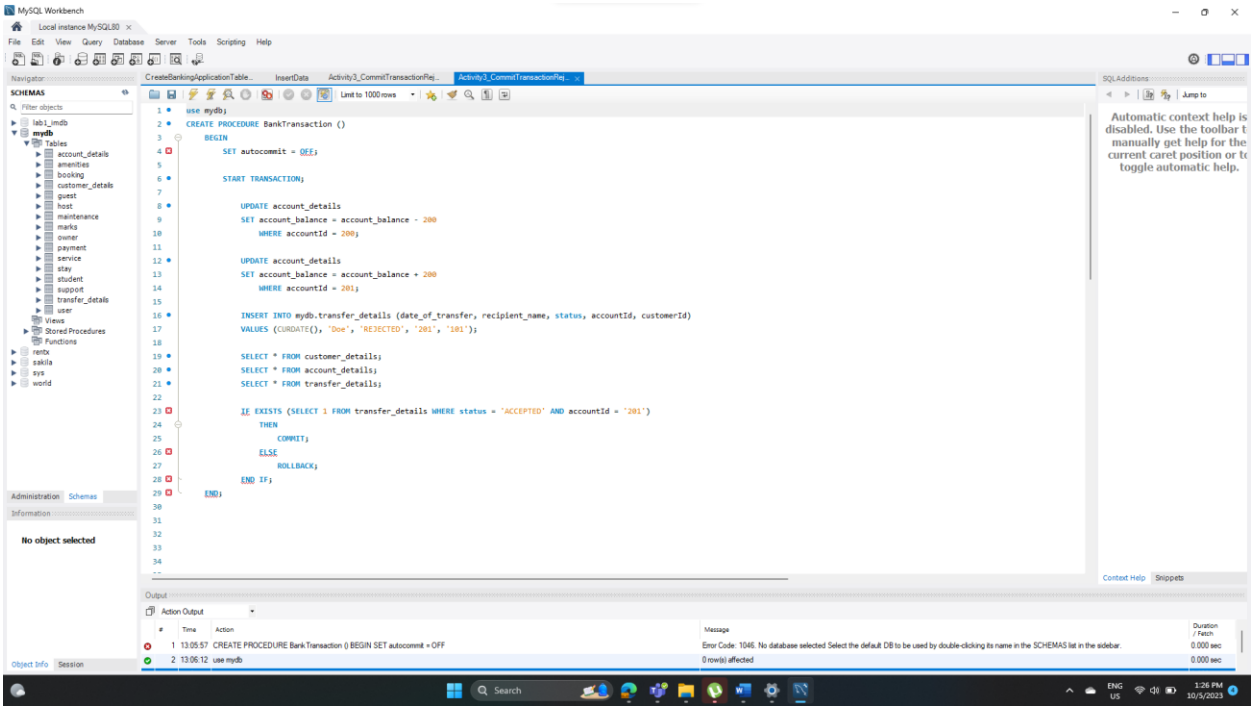


Problem Statement 3: Assume that the transaction status gets verified by some X business logic (pure assumption here, no need to implement any logic), the transaction status is changed to either accepted or declined. Based on this status, handle the update balance query.

(Hint: Based on the transaction status, either commit or rollback the account details table SQL statements. Try exploring SAVEPOINT).

Scenario #1: If the backend logic changed it to ACCEPTED then the transaction should COMMIT and if it is REJECTED it should rollback everything that is done in this transaction

Line #23: checks for the state of the transfer_details record. If it is ACCEPTED then commit, If REJECTED then ROLLBACK



Scenario #2: If the backend logic changed it to ACCEPTED then the transaction should COMMIT and if it is REJECTED it should rollback up till a SAVEPOINT.

Here in case of in case of ACCEPTED it will commit if not It will rollback to JOHN_ACC_UPDATE savepoint. Which means everything from Line #12 rolled back.