Hands-on Lab: Committing and Rolling b Transaction using a Stored Procedure

Estimated time needed: 10 minutes

A transaction is simply a sequence of operations performed using one or more SQL statements as a single log must be ACID (Atomic, Consistent, Isolated and Durable). The effects of all the SQL statements in a transactior the COMMIT command or undone from the database using the ROLLBACK command.

In this lab, you will learn some commonly used TCL (Transaction Control Language) commands of SQL througl You will learn about COMMIT, which is used to permanently save the changes done in the transactions in a tal undo the transactions that have not been saved in a table. ROLLBACK can only be used to undo the changes i

Software Used in this Lab

In this lab, you will use an IBM Db2 Database. Db2 is a Relational Database Management System (RDBMS) froı retrieve data efficiently.

To complete this lab you will utilize a Db2 database service on IBM Cloud. If you did not already complete this yet have access to Db2 on IBM Cloud, and you will need to follow the lab below first:

- Hands-on Lab : Sign up for IBM Cloud, Create Db2 service instance and Get started with the Db2 console

Data Used in this Lab

The data used in this lab is internal data. You will be working on the **BankAccounts** and **ShoeShop** tables.

ACCOUNTNUMBER	ACCOUNTNAME
B001	Rose
B002	James
B003	Shoe Shop
B004	Corner Shop

- Permanently save the changes done in a transaction
- Undo the transaction that has not been saved

Instructions

When you approach the exercises in this lab, follow the instructions to run the queries on Db2:

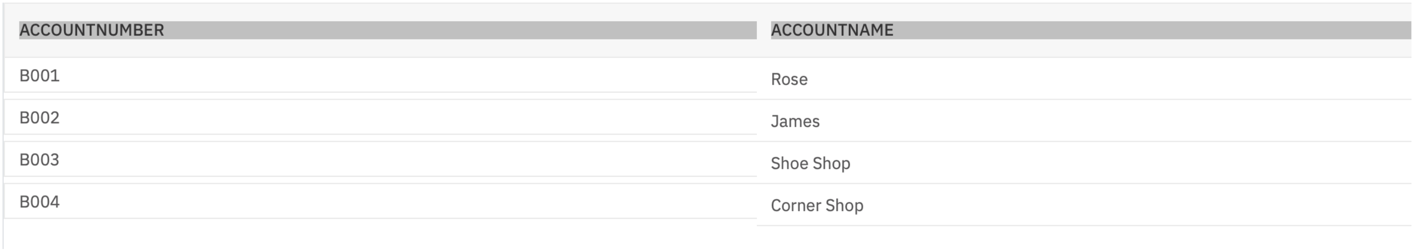
- Go to the Resource List of IBM Cloud by logging in where you can find the Db2 service instance that you under **Services** section. Click on the **Db2-xx service**. Next, open the Db2 Console by clicking on **Open Co** in the top left corner and go to the **Run SQL** page. The Run SQL tool enables you to run SQL statements.
    - If needed, follow Hands-on Lab : Sign up for IBM Cloud, Create Db2 service instance and Get started

Exercise

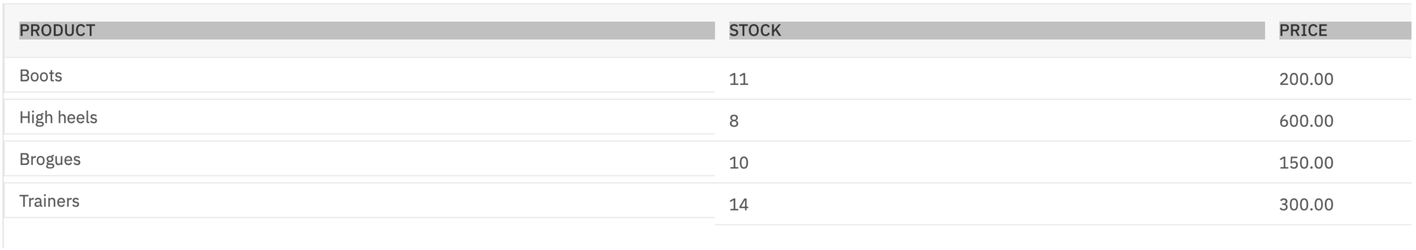
Task A: Example exercise

Let us go through an example on committing and rolling back a transaction

1. Make sure you have created and populated the **BankAccounts** and **ShoeShop** tables by following the **“D**



ACCOUNTNUMBER	ACCOUNTNAME
B001	Rose
B002	James
B003	Shoe Shop
B004	Corner Shop



PRODUCT	STOCK	PRICE
Boots	11	200.00
High heels	8	600.00
Brogues	10	150.00
Trainers	14	300.00

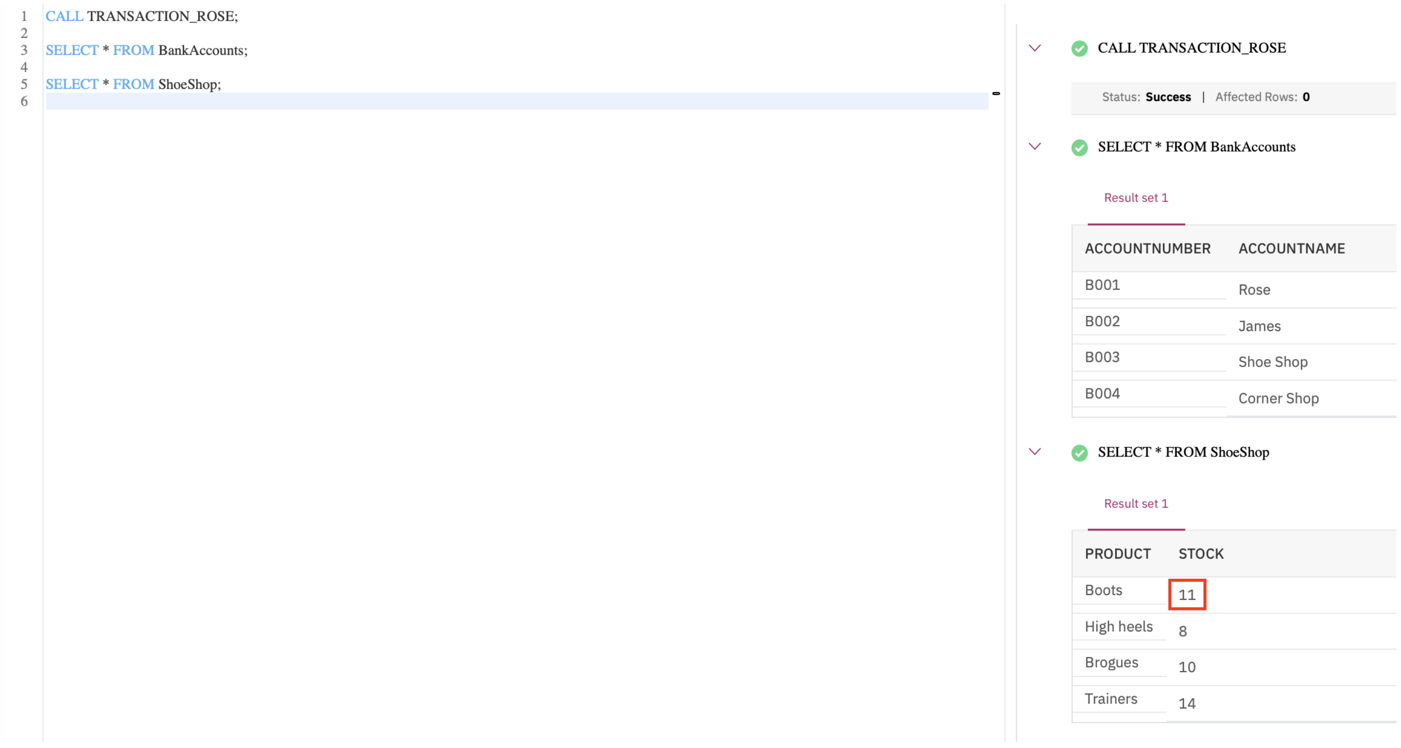
2.
    - You will create a stored procedure routine named **TRANSACTION_ROSE** which will include TCL comm
    - Now develop the routine based on the given scenario to execute a transaction.
    - **Scenario:** Let’ s buy Rose a pair of Boots from ShoeShop. So we have to update the Rose balance as BankAccounts table. Then we also have to update Boots stock in the ShoeShop table. After Boots, let’ Trainers.
    - To create the stored procedure routine on Db2, copy the code below and paste it to the textbox of the

```sql
--#SET TERMINATOR @
CREATE PROCEDURE TRANSACTION_ROSE                               -- Name of this stored

LANGUAGE SQL                                                    -- Language used in thi
MODIFIES SQL DATA                                               -- This routine will or

BEGIN

        DECLARE SQLCODE INTEGER DEFAULT 0;                      -- Host variable SQLCOD
        DECLARE retcode INTEGER DEFAULT 0;                      -- Local variable retco
        DECLARE CONTINUE HANDLER FOR SQLEXCEPTION               -- Handler tell the rou
warning occurs
        SET retcode = SQLCODE;                                  -- Value of SQLCODE ass

        UPDATE BankAccounts
        SET Balance = Balance-200
        WHERE AccountName = 'Rose';

        UPDATE BankAccounts
        SET Balance = Balance+200
        WHERE AccountName = 'Shoe Shop';

        UPDATE ShoeShop
        SET Stock = Stock-1
        WHERE Product = 'Boots';

        UPDATE BankAccounts
        SET Balance = Balance-300
        WHERE AccountName = 'Rose';


        IF retcode < 0 THEN                                      --  SQLCODE returns ne
success, positive value for warning
            ROLLBACK WORK;

        ELSE
            COMMIT WORK;

        END IF;

```

```sql
CALL TRANSACTION_ROSE;  -- Caller query

SELECT * FROM BankAccounts;

SELECT * FROM ShoeShop;

```

4. We can observe that the transaction has been executed. But when we observe the tables, no changes hav
   COMMIT. All the possible changes happened might have been undone through ROLLBACK since the whol
   statement or more. Let’ s go through the possible reason behind the failure of the transaction and how C
   procedure:

   - The first three UPDATEs should run successfully. Both the balance of Rose and ShoeShop should have
     The current balance of Rose should stand at 300 - 200 (price of a pair of Boots) = 100. The current bal
     200 = 124400. The stock of Boots should also be updated in the ShoeShop table after the successful
   - The last UPDATE statement tries to buy Rose a pair of Trainers, but her balance becomes insufficient (
     Trainers: 300) after buying a pair of Boots. So, the last UPDATE statement fails. Since the whole transa
     the transaction won’ t be committed.
   - The **SQLCODE** which is a stand-alone host variable contains success/failure/warning information of ea
     since **SQLCODE** variable gets reset back as the next SQL statement runs, **retcode** is our local variable
     this **SQLCODE**. **SQLCODE** returns negative value for each SQL statement if not executed successfully.
     are rolled back. Commit only takes place after the transaction gets executed successfully without any



```sql
CALL TRANSACTION_ROSE;

SELECT * FROM BankAccounts;

SELECT * FROM ShoeShop;
```

CALL TRANSACTION_ROSE

Status: **Success** | Affected Rows: **0**

SELECT * FROM BankAccounts

Result set 1

ACCOUNTNUMBER	ACCOUNTNAME
B001	Rose
B002	James
B003	Shoe Shop
B004	Corner Shop

SELECT * FROM ShoeShop

Result set 1

PRODUCT	STOCK
Boots	11
High heels	8
Brogues	10
Trainers	14

Task B: Practice exercise

Now let’ s practice an exercise on committing and rolling back a transaction.