Ryan Young

CSE 4701

4/28/2020

Project 2 Part 2

I. Screenshots

Main Function Code Snippet:

```
def main():
    # initial Setup and getting choice from user
    while True:
        Format_Output()
        User_Choice = int(input("Enter Your Choice: "))
        # Connection to DB
        connection = Create_Connection()
        # Exit Option
        if User_Choice == 0:
            print("Good Bye")
            break
        # Create Account Option
        elif User_Choice == 1:
            Create_Account(connection)
        # Check Balance Option
        elif User_Choice == 2:
            Check_Balance(connection)
        # Deposit Option
        elif User_Choice == 3:
            Deposit(connection)
        # Withdraw Option
        elif User_Choice == 4:
            Withdraw(connection)
        # Transfer Option
        elif User_Choice == 5:
            Transfer(connection)


main()
```

Deposit Code Snippet:

```python
def Deposit(connection):
    account_no = int(input("Please enter an account number: "))
    # need to now pull account information
    # lock the account with the for update part
    # from the table and print it
    with connection.cursor() as cursor:
        sql = "select * from account where account.account_no = %s for update;" \
              % (account_no)
        cursor.execute(sql)
        result = cursor.fetchone()
        if result is None:
            print("-------------------------")
            print("Invalid Account Number")
            print("-------------------------")
            connection.commit()
        else:
            print("-----Existing Account Balance-----")
            print("Account Number:", result['account_no'])
            print("Account Name:", result['name_on_account'])
            print("Account Balance:", result['balance'])
            print("Account Opening Date:", result['account_open_date'])
            connection.commit()
            # asking teller how much to deposit and doing it
            deposit_amount = int(input("Please enter deposit amount: "))
            new_balance = deposit_amount + result['balance']
            with connection.cursor() as cursor:
                sql = "update account set balance = %d where account_no = %s;" \
                      % (new_balance, account_no)
                cursor.execute(sql)
```

```
        connection.commit()
        with connection.cursor() as cursor:
            sql = "select * from account where account.account_no = %s;" \
                  % (account_no)
            cursor.execute(sql)
            result = cursor.fetchone()
            print("-----New Account Balance-----")
            print("Account Number:", result['account_no'])
            print("Account Name:", result['name_on_account'])
            print("Account Balance:", result['balance'])
            print("Account Opening Date:",
                  result['account_open_date'])
            print("--------------------------")
        connection.commit()
```

Deposit Function Working:

Prior to running the script:

```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         138 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2005 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       60000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       50000 | 2020-04-20 22:30:22 |
+------------+-----------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

Script Run:

```
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice: 3
Please enter an account number: 13
-----Existing Account Balance-----
Account Number: 13
Account Name: Ryan Young
Account Balance: 2005.0
Account Opening Date: 2020-04-20 22:29:55
Please enter deposit amount: 15
-----New Account Balance-----
Account Number: 13
Account Name: Ryan Young
Account Balance: 2020.0
Account Opening Date: 2020-04-20 22:29:55
-------------------------
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice:
```

After running the script:

```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         138 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2020 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       60000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       50000 | 2020-04-20 22:30:22 |
+------------+-----------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

Withdraw Code Snippet:

```python
def Withdraw(connection):
    account_no = int(input("Please enter an account number: "))
    # need to now pull account information
    # lock the account with the for update part
    # from the table and print it
    with connection.cursor() as cursor:
        sql = "select * from account where account.account_no = %s for update;" \
              % (account_no)
        cursor.execute(sql)
        result = cursor.fetchone()
        if result is None:
            print("--------------------------")
            print("Invalid Account Number")
            print("--------------------------")
            connection.commit()
        else:
            print("-----Existing Account Balance-----")
            print("Account Number:", result['account_no'])
            print("Account Name:", result['name_on_account'])
            print("Account Balance:", result['balance'])
            print("Account Opening Date:", result['account_open_date'])
            print("--------------------------")
            connection.commit()
            # asking teller how much to withdraw and doing it
            withdraw_amount = int(input("Please enter withdraw amount: "))
            new_balance = result['balance'] - withdraw_amount
            with connection.cursor() as cursor:
                sql = "update account set balance = %d where account_no = %s;" \
                      % (new_balance, account_no)
                cursor.execute(sql)
```

```
        connection.commit()
        with connection.cursor() as cursor:
            sql = "select * from account where account.account_no = %s;" \
                  % (account_no)
            cursor.execute(sql)
            result = cursor.fetchone()
            print("-----New Account Balance-----")
            print("Account Number:", result['account_no'])
            print("Account Name:", result['name_on_account'])
            print("Account Balance:", result['balance'])
            print("Account Opening Date:",
                  result['account_open_date'])
            print("--------------------------")
        connection.commit()
```

Withdraw Function Working:

Prior to running the script:

```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         138 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2020 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       60000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       50000 | 2020-04-20 22:30:22 |
+------------+-----------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

Script Run:

```
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice: 4
Please enter an account number: 12
-----Existing Account Balance-----
Account Number: 12
Account Name: Person Human
Account Balance: 138.0
Account Opening Date: 2020-04-20 22:24:39
-------------------------
Please enter withdraw amount: 38
-----New Account Balance-----
Account Number: 12
Account Name: Person Human
Account Balance: 100.0
Account Opening Date: 2020-04-20 22:24:39
-------------------------
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice:
```

After running the script:

```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         100 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2020 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       60000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       50000 | 2020-04-20 22:30:22 |
+------------+-----------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

Transfer Code Snippet:

```
def Transfer(connection):
    # GETTING SOURCE ACCOUNT
    source_account_no = int(input("Please enter source account number: "))
    with connection.cursor() as cursor:
        # Locking the account
        sql = "select * from account where account.account_no = %s for update;" \
              % (source_account_no)
        cursor.execute(sql)
        result = cursor.fetchone()
        if result is None:
            print("--------------------------")
            print("Invalid Account Number")
            print("--------------------------")
            connection.commit()
        else:
            print("-----Existing Account Balance-----")
            print("Account Number:", result['account_no'])
            print("Account Name:", result['name_on_account'])
            print("Account Balance:", result['balance'])
            print("Account Opening Date:", result['account_open_date'])
            source_balance = result['balance']
            connection.commit()
    # GETTING TARGET ACCOUNT
    target_account_no = int(input("Please enter target account number: "))
    with connection.cursor() as cursor:
        # Locking the account
        sql = "select * from account where account.account_no = %s for update;" \
              % (target_account_no)
        cursor.execute(sql)
        result = cursor.fetchone()
```

```
        if result is None:
            print("--------------------------")
            print("Invalid Account Number")
            print("--------------------------")
            connection.commit()
        else:
            print("-----Existing Account Balance-----")
            print("Account Number:", result['account_no'])
            print("Account Name:", result['name_on_account'])
            print("Account Balance:", result['balance'])
            print("Account Opening Date:", result['account_open_date'])
            target_balance = result['balance']
            connection.commit()
        # Withdrawing from source account
        transfer_amount = int(input("Please enter transfer amount amount: "))
        Withdraw_balance = source_balance - transfer_amount
        with connection.cursor() as cursor:
            sql = "update account set balance = %d where account_no = %s;" \
                  % (Withdraw_balance, source_account_no)
            cursor.execute(sql)
            sql = "select * from account where account.account_no = %s;" \
                  % (source_account_no)
            cursor.execute(sql)
            result = cursor.fetchone()
            print("-----New Account Balance-----")
            print("Account Number:", result['account_no'])
            print("Account Name:", result['name_on_account'])
            print("Account Balance:", result['balance'])
            print("Account Opening Date:",
                  result['account_open_date'])
```

```
        print("--------------------------")
        #time.sleep(100)
    # Depositing into target account
        Deposit_balance = transfer_amount + target_balance
        sql = "update account set balance = %d where account_no = %s;" \
              % (Deposit_balance, target_account_no)
        cursor.execute(sql)
        sql = "select * from account where account.account_no = %s;" \
              % (target_account_no)
        cursor.execute(sql)
        result = cursor.fetchone()
        print("-----New Account Balance-----")
        print("Account Number:", result['account_no'])
        print("Account Name:", result['name_on_account'])
        print("Account Balance:", result['balance'])
        print("Account Opening Date:",
              result['account_open_date'])
        print("--------------------------")
    connection.commit()
```

Transfer Function Working:

Prior to script:

```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         100 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2020 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       60000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       50000 | 2020-04-20 22:30:22 |
+------------+-----------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

Script Run:

```
C:\Users\ryans\Documents\Academics\Spring_2020_Classes\CSE 4701\Project_2>Python Project#2.py
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice: 5
Please enter source account number: 15
-----Existing Account Balance-----
Account Number: 15
Account Name: Scott Young
Account Balance: 50000.0
Account Opening Date: 2020-04-20 22:30:22
Please enter target account number: 14
-----Existing Account Balance-----
Account Number: 14
Account Name: Susan Young
Account Balance: 60000.0
Account Opening Date: 2020-04-20 22:30:12
Please enter transfer amount amount: 10000
-----New Account Balance-----
Account Number: 15
Account Name: Scott Young
Account Balance: 40000.0
Account Opening Date: 2020-04-20 22:30:22
-------------------------
-----New Account Balance-----
Account Number: 14
Account Name: Susan Young
Account Balance: 70000.0
Account Opening Date: 2020-04-20 22:30:12
-------------------------
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice:
```

After running the script:

```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         100 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2020 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       70000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       40000 | 2020-04-20 22:30:22 |
+------------+-----------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

ii. Proving Atomicity:

```
        print("-----New Account Balance-----")
        print("Account Number:", result['account_no'])
        print("Account Name:", result['name_on_account'])
        print("Account Balance:", result['balance'])
        print("Account Opening Date:",
              result['account_open_date'])
        print("--------------------------")
        time.sleep(100)
    # Depositing into target account
        Deposit_balance = transfer_amount + target_balance
        sql = "update account set balance = %d where account_no = %s;" \
              % (Deposit_balance, target_account_no)
        cursor.execute(sql)
        sql = "select * from account where account.account_no = %s;" \
              % (target_account_no)
        cursor.execute(sql)
        result = cursor.fetchone()
        print("-----New Account Balance-----")
        print("Account Number:", result['account_no'])
        print("Account Name:", result['name_on_account'])
        print("Account Balance:", result['balance'])
        print("Account Opening Date:",
              result['account_open_date'])
        print("--------------------------")
    connection.commit()
```

I added a sleep period of 100 seconds after the withdrawing and before the depositing of the transfer function. This will allow me to close the program after the withdraw is done in the transfer function of the script and be able to check that it did not go through without the deposit.

Before the script:

```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         100 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2020 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       70000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       40000 | 2020-04-20 22:30:22 |
+------------+-----------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

Script Run:

```
C:\Users\ryans\Documents\Academics\Spring_2020_Classes\CSE 4701\Project_2>Python Project#2.py
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice: 5
Please enter source account number: 14
-----Existing Account Balance-----
Account Number: 14
Account Name: Susan Young
Account Balance: 70000.0
Account Opening Date: 2020-04-20 22:30:12
Please enter target account number: 13
-----Existing Account Balance-----
Account Number: 13
Account Name: Ryan Young
Account Balance: 2020.0
Account Opening Date: 2020-04-20 22:29:55
Please enter transfer amount amount: 100
-----New Account Balance-----
Account Number: 14
Account Name: Susan Young
Account Balance: 69900.0
Account Opening Date: 2020-04-20 22:30:12
-------------------------
Traceback (most recent call last):
  File "Project#2.py", line 277, in <module>
    main()
  File "Project#2.py", line 274, in main
    Transfer(connection)
  File "Project#2.py", line 229, in Transfer
    time.sleep(100)
KeyboardInterrupt

C:\Users\ryans\Documents\Academics\Spring_2020_Classes\CSE 4701\Project_2>
```

After the script was run:

```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         100 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2020 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       70000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       40000 | 2020-04-20 22:30:22 |
+------------+-----------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

Before the script was run:

```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         100 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2020 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       70000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       40000 | 2020-04-20 22:30:22 |
+------------+-----------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

Running the script again but this time closing command prompt instead of keyboard interrupt:

```
C:\Users\ryans\Documents\Academics\Spring_2020_Classes\CSE 4701\Project_2>Python Project#2.py
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice: 5
Please enter source account number: 15
-----Existing Account Balance-----
Account Number: 15
Account Name: Scott Young
Account Balance: 40000.0
Account Opening Date: 2020-04-20 22:30:22
Please enter target account number: 10
-----Existing Account Balance-----
Account Number: 10
Account Name: Elon Musk
Account Balance: 8999.0
Account Opening Date: 2020-04-20 22:16:07
Please enter transfer amount amount: 1
-----New Account Balance-----
Account Number: 15
Account Name: Scott Young
Account Balance: 39999.0
Account Opening Date: 2020-04-20 22:30:22
-------------------------
```

After the script was run:

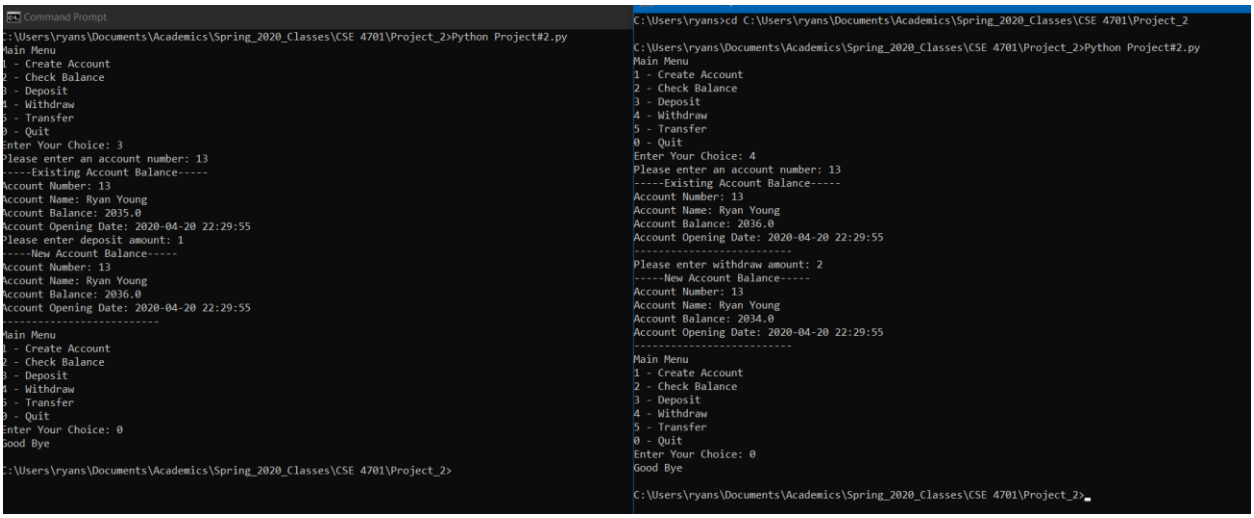
```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         100 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2020 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       70000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       40000 | 2020-04-20 22:30:22 |
+------------+-----------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

iii. Consistency:

Yes, my program absolutely has this property, in the transfer function it will not go through with the withdrawal of money if the deposit is not completed. In the rest of the functions there will also be in no case where a power cut to the machine running the script will impact the database. This is because the only time there could be an issue would be when a SQL query was being remotely run and if this got cut short there would simply be an error saying it was an invalid SQL query!

iv. Isolation:

Now I will show how once an instance of the script has access to an account another instance cannot change the values in it!



In the left command prompt window I started a deposit and then moved over to the right command prompt window where I started a withdraw on the same account. After entering the same account number as the left command prompt window, the script in the right window waited for me to finish in the left window before allowing me to move forward. This can be proved because in the existing account balance portion you can see in the left hand window the existing balance was 2035 and after this script finishes the right hand window has an existing balance of 2036 because I deposited a dollar in the left side window! This is all done by using the row lock feature of SQL! I used “for update” in my script.

After the two scripts were run:

```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         100 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2034 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       70000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       40000 | 2020-04-20 22:30:22 |
+------------+-----------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

v. Durability:

As soon on multiple occasion my program can be run, create some account, have some transactions occur and then when the script is closed the database will accurately reflect the changes made by the program.

Before Script was run:

```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         100 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2034 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       70000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       40000 | 2020-04-20 22:30:22 |
+------------+-----------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

Creation of Accounts:

```
C:\Users\ryans\Documents\Academics\Spring_2020_Classes\CSE 4701\Project_2>Python Project#2.py
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice: 1
Name on account: Dong Shin
Enter Initial Balance: 1
Account Number: 16
Account Name: Dong Shin
Account Balance: 1.0
Account Opening Date: 2020-04-21 12:41:56
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice: 1
Name on account: TA Account
Enter Initial Balance: 50000
Account Number: 17
Account Name: TA Account
Account Balance: 50000.0
Account Opening Date: 2020-04-21 12:42:11
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice: 0
Good Bye

C:\Users\ryans\Documents\Academics\Spring_2020_Classes\CSE 4701\Project_2>
```

Script Closed and Result on Table:

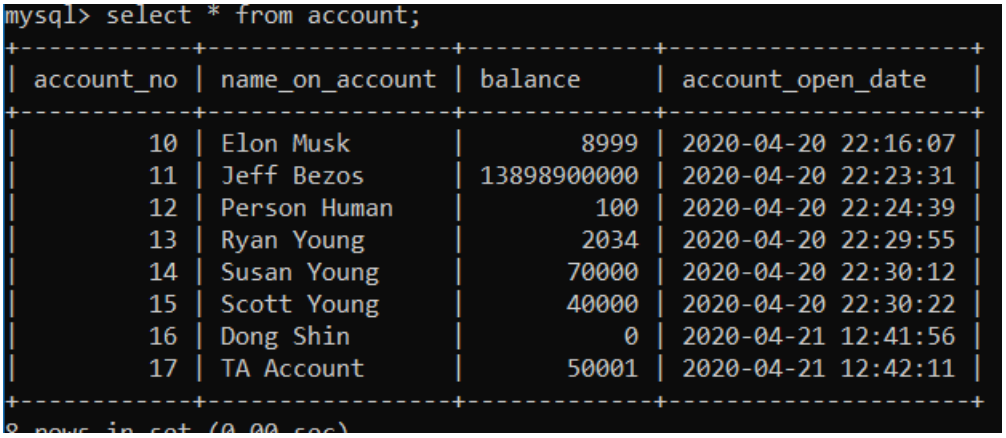
```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         100 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2034 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       70000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       40000 | 2020-04-20 22:30:22 |
|         16 | Dong Shin       |           1 | 2020-04-21 12:41:56 |
|         17 | TA Account      |       50000 | 2020-04-21 12:42:11 |
+------------+-----------------+-------------+---------------------+
8 rows in set (0.00 sec)
```

Reopen Script and run transactions:

```
C:\Users\ryans\Documents\Academics\Spring_2020_Classes\CSE 4701\Project_2>Python Project#2.py
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice: 5
Please enter source account number: 16
-----Existing Account Balance-----
Account Number: 16
Account Name: Dong Shin
Account Balance: 1.0
Account Opening Date: 2020-04-21 12:41:56
Please enter target account number: 17
-----Existing Account Balance-----
Account Number: 17
Account Name: TA Account
Account Balance: 50000.0
Account Opening Date: 2020-04-21 12:42:11
Please enter transfer amount amount: 1
-----New Account Balance-----
Account Number: 16
Account Name: Dong Shin
Account Balance: 0.0
Account Opening Date: 2020-04-21 12:41:56
--------------------------
-----New Account Balance-----
Account Number: 17
Account Name: TA Account
Account Balance: 50001.0
Account Opening Date: 2020-04-21 12:42:11
--------------------------
Main Menu
1 - Create Account
2 - Check Balance
3 - Deposit
4 - Withdraw
5 - Transfer
0 - Quit
Enter Your Choice: 0
Good Bye

C:\Users\ryans\Documents\Academics\Spring_2020_Classes\CSE 4701\Project_2>
```

Result After the Script:



```
mysql> select * from account;
+------------+-----------------+-------------+---------------------+
| account_no | name_on_account | balance     | account_open_date   |
+------------+-----------------+-------------+---------------------+
|         10 | Elon Musk       |        8999 | 2020-04-20 22:16:07 |
|         11 | Jeff Bezos      | 13898900000 | 2020-04-20 22:23:31 |
|         12 | Person Human    |         100 | 2020-04-20 22:24:39 |
|         13 | Ryan Young      |        2034 | 2020-04-20 22:29:55 |
|         14 | Susan Young     |       70000 | 2020-04-20 22:30:12 |
|         15 | Scott Young     |       40000 | 2020-04-20 22:30:22 |
|         16 | Dong Shin       |           0 | 2020-04-21 12:41:56 |
|         17 | TA Account      |       50001 | 2020-04-21 12:42:11 |
+------------+-----------------+-------------+---------------------+
8 rows in set (0.00 sec)
```

Now we can see that the script obviously has durability because not only did I create accounts, close the script and the accounts where in the table in MySQL, but I then reopened the script and preformed a transfer from Dong Shin to the TA Account.