End Term Project Report

On

Introduction to Databases (CSE 3151)

Submitted by

Name	: **Mukti Singh**
Reg. No.	: **2241019496**
Branch	: **CSE (AI/ML)**
Semester	: **6**
Section	: **08**
Session	: **2024-2025**
Admission Batch	: **2022**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

FACULTY OF ENGINEERING & TECHNOLOGY (ITER)

SIKSHA 'O' ANUSANDHAN DEEMED TO BE UNIVERSITY

BHUBANESWAR, ODISHA – 751030

SOLUTION

1)Database Schema (SQL):

```sql
CREATE TABLE customers (

    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,

    name VARCHAR2(100), email VARCHAR2(100),

    phone VARCHAR2(20)

    );

CREATE TABLE accounts (

     id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,

    customer_id INT REFERENCES customers(id),

    balance DECIMAL(10, 2)

    );

CREATE TABLE loans (

    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,

    customer_id INT REFERENCES customers(id),

    amount DECIMAL(10, 2), status VARCHAR2(20)

    );
```

2) Code:

```java
import java.sql.*;
import java.io.*;
public class BankingSystem {
  public static void main(String[] args) throws IOException {
    Connection con = null;
    Statement stmt = null;
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    try {
      Class.forName("oracle.jdbc.driver.OracleDriver");
      String conurl = "jdbc:oracle:thin:@172.17.144.110:1521:ora11g";
      con = DriverManager.getConnection(conurl, "system", "mukti123");
      stmt = con.createStatement();
      int choice;
      do {
        System.out.println("\n***** Banking Management System *****");
        System.out.println("1. Show Customers");
        System.out.println("2. Add Customer");
        System.out.println("3. Delete Customer");
        System.out.println("4. Update Customer");
        System.out.println("5. Show Account Details");
```

```java
System.out.println("6. Show Loan Details");

System.out.println("7. Deposit");

System.out.println("8. Withdraw");

System.out.println("9. Exit");

System.out.print("Enter your choice: ");

choice = Integer.parseInt(br.readLine());

switch (choice) {

    case 1:

        showCustomers(stmt);

        break;

    case 2:

        addCustomer(stmt, br);

        break;

    case 3:

        deleteCustomer(stmt, br);

        break;

    case 4:

        updateCustomer(stmt, br);

        break;

    case 5:

        showAccountDetails(stmt, br);

        break;
```

```java
                case 6:
                    showLoanDetails(stmt, br);
                    break;
                case 7:
                    deposit(stmt, br);
                    break;
                case 8:
                    withdraw(stmt, br);
                    break;
                case 9:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice!");
            }
        } while (choice != 9);
    } catch (Exception e) {
        System.out.println("Error: " + e);
    } finally {
        try {
            if (stmt != null) stmt.close();
            if (con != null) con.close();
```

```java
        } catch (Exception e) {
            System.out.println("Closing error: " + e);
        }
    }
}
public static void showCustomers(Statement stmt) throws
SQLException {
    ResultSet rs = stmt.executeQuery("SELECT * FROM Customer");
    System.out.println("\nCust_No\tName\tPhone\tCity");
    while (rs.next()) {
        System.out.printf("%s\t%s\t%s\t%s\n",
            rs.getString("cust_no"),
            rs.getString("name"),
            rs.getString("phone_no"),
            rs.getString("city"));
    }
}
public static void addCustomer(Statement stmt, BufferedReader br)
throws IOException, SQLException {
    System.out.print("Enter Customer No: ");
    String custNo = br.readLine();
    System.out.print("Enter Name: ");
    String name = br.readLine();
```

```java
        System.out.print("Enter Phone No: ");

        String phone = br.readLine();

        System.out.print("Enter City: ");

        String city = br.readLine();

        String query = String.format("INSERT INTO Customer VALUES ('%s',
'%s', '%s', '%s')",

                custNo, name, phone, city);

        int res = stmt.executeUpdate(query);

        if (res > 0) {

            System.out.println("Customer added successfully!");

            showCustomers(stmt);

        }

    }

    public static void deleteCustomer(Statement stmt, BufferedReader br)
throws IOException, SQLException {

        System.out.print("Enter Customer No to delete: ");

        String custNo = br.readLine();

        String query = "DELETE FROM Customer WHERE cust_no = '" +
custNo + "'";

        int res = stmt.executeUpdate(query);

        if (res > 0) {

            System.out.println("Customer deleted successfully.");

            showCustomers(stmt);
```

```java
        } else {

            System.out.println("Customer not found.");

        }

    }

    public static void updateCustomer(Statement stmt, BufferedReader
br) throws IOException, SQLException {

        System.out.print("Enter Customer No to update: ");

        String custNo = br.readLine();

        System.out.println("1 - Name, 2 - Phone, 3 - City");

        int option = Integer.parseInt(br.readLine());

        String column = "", newValue = "";

        switch (option) {

            case 1:

                column = "name";

                System.out.print("Enter new name: ");

                break;

            case 2:

                column = "phone_no";

                System.out.print("Enter new phone: ");

                break;

            case 3:

                column = "city";
```

```java
                System.out.print("Enter new city: ");

                break;

            default:

                System.out.println("Invalid option.");

                return;

        }

        newValue = br.readLine();

        String query = String.format("UPDATE Customer SET %s = '%s' WHERE cust_no = '%s'",

                column, newValue, custNo);

        int res = stmt.executeUpdate(query);

        if (res > 0) {

            System.out.println("Customer updated successfully.");

            showCustomers(stmt);

        } else {

            System.out.println("Update failed.");

        }

    }

    public static void showAccountDetails(Statement stmt, BufferedReader br) throws IOException, SQLException {

        System.out.print("Enter Customer No: ");

        String custNo = br.readLine();
```
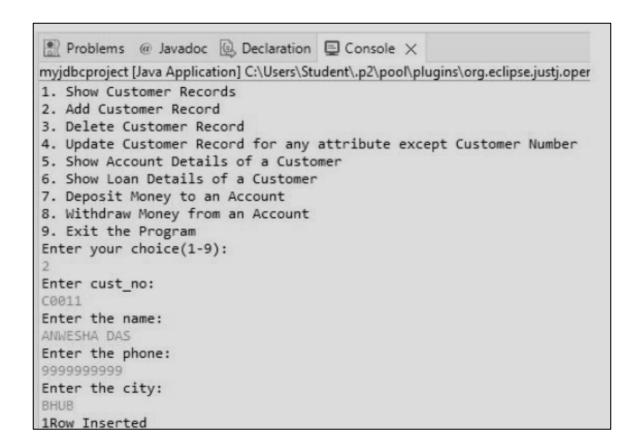
```java
        String query = "SELECT c.cust_no, c.name, a.account_no, a.type,
a.balance, b.branch_code, b.branch_name, b.branch_city " +
            "FROM Customer c JOIN Account a ON c.cust_no = a.cust_no " +
            "JOIN Branch b ON a.branch_code = b.branch_code " +
            "WHERE c.cust_no = '" + custNo + "'";
        ResultSet rs = stmt.executeQuery(query);
        boolean found = false;
        while (rs.next()) {
            found = true;
            System.out.printf("Cust: %s, Name: %s, Acc No: %s, Type: %s,
Balance: %.2f, Branch: %s, %s, %s\n",
                rs.getString("cust_no"),
                rs.getString("name"),
                rs.getString("account_no"),
                rs.getString("type"),
                rs.getDouble("balance"),
                rs.getString("branch_code"),
                rs.getString("branch_name"),
                rs.getString("branch_city"));
        }
        if (!found) {
            System.out.println("No account details found.");
        }
```
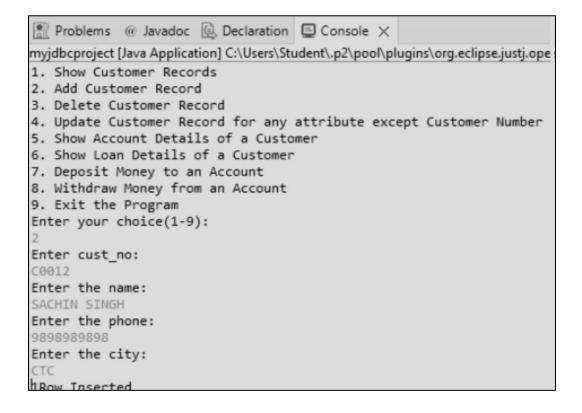
```java
    }
    public static void showLoanDetails(Statement stmt, BufferedReader
br) throws IOException, SQLException {
        System.out.print("Enter Customer No: ");
        String custNo = br.readLine();
        String query = "SELECT c.cust_no, c.name, l.loan_no, l.amount,
b.branch_code, b.branch_name, b.branch_city " +
            "FROM Customer c JOIN Loan l ON c.cust_no = l.cust_no " +
            "JOIN Branch b ON l.branch_code = b.branch_code " +
            "WHERE c.cust_no = '" + custNo + "'";
        ResultSet rs = stmt.executeQuery(query);
        boolean found = false;
        while (rs.next()) {
            found = true;
            System.out.printf("Cust: %s, Name: %s, Loan No: %s, Amount:
%.2f, Branch: %s, %s, %s\n",
                rs.getString("cust_no"),
                rs.getString("name"),
                rs.getString("loan_no"),
                rs.getDouble("amount"),
                rs.getString("branch_code"),
                rs.getString("branch_name"),
                rs.getString("branch_city"));
```
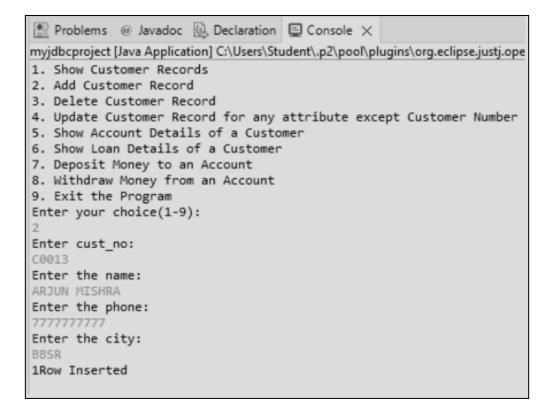
```java
        }
        if (!found) {
            System.out.println("Congratulations! No loans found.");
        }
    }
    public static void deposit(Statement stmt, BufferedReader br) throws
IOException, SQLException {
        System.out.print("Enter Account No: ");
        String accNo = br.readLine();
        System.out.print("Enter Amount to Deposit: ");
        double amount = Double.parseDouble(br.readLine());
        String update = "UPDATE Account SET balance = balance + " +
amount + " WHERE account_no = '" + accNo + "'";
        int res = stmt.executeUpdate(update);
        if (res > 0) {
            System.out.println("Amount deposited successfully.");
        } else {
            System.out.println("Account not found.");
        }
    }
    public static void withdraw(Statement stmt, BufferedReader br)
throws IOException, SQLException {
        System.out.print("Enter Account No: ");
```

```java
        String accNo = br.readLine();

        System.out.print("Enter Amount to Withdraw: ");

        double amount = Double.parseDouble(br.readLine());

        ResultSet rs = stmt.executeQuery("SELECT balance FROM Account
WHERE account_no = '" + accNo + "'");

        if (rs.next()) {

            double balance = rs.getDouble("balance");

            if (balance >= amount) {

                String update = "UPDATE Account SET balance = balance - " +
amount + " WHERE account_no = '" + accNo + "'";

                stmt.executeUpdate(update);

                System.out.println("Amount withdrawn successfully.");

            } else {

                System.out.println("Insufficient balance.");

            }

        } else {

            System.out.println("Account not found.");

        }

    }

}
```

3) OUTPUT:

```
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
1

C0001          RAJ ANAND SINGH 9861258466      DELHI
C0002          ANKITA SINGH    9879958651      BANGALORE
C0003          SOUMYA JHA      9885623344      MUMBAI
C0004          ABHIJIT MISHRA  9455845425      MUMBAI
C0005          YASH SARAF      9665854585      KOLKATA
C0006          SWAROOP RAY     9437855466      CHENNAI
C0007          SURYA NARAYAN   9937955212      GURGAON
C0008          PRANAV PRAVEEN  9336652441      PUNE
C0009          STUTI MISRA     7870266534      DELHI
C0010          ASLESHA TIWARI  null    MUMBAI
The number of rows selected is 10
```
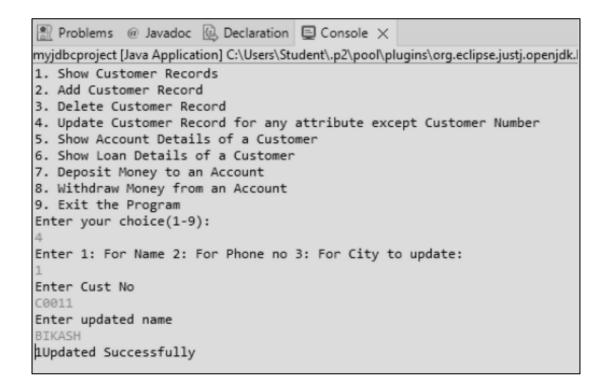
```
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
2
Enter cust_no:
C0011
Enter the name:
ANWESHA DAS
Enter the phone:
9999999999
Enter the city:
BHUB
1Row Inserted
```

```
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
2
Enter cust_no:
C0012
Enter the name:
SACHIN SINGH
Enter the phone:
9898989898
Enter the city:
CTC
1Row Inserted
```

```
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
2
Enter cust_no:
C0013
Enter the name:
ARJUN MISHRA
Enter the phone:
7777777777
Enter the city:
BBSR
1Row Inserted
```

```
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
1

C0001          RAJ ANAND SINGH 9861258466      DELHI
C0002          ANKITA SINGH    9879958651      BANGALORE
C0003          SOUMYA JHA      9885623344      MUMBAI
C0004          ABHIJIT MISHRA  9455845425      MUMBAI
C0005          YASH SARAF      9665854585      KOLKATA
C0006          SWAROOP RAY     9437855466      CHENNAI
C0007          SURYA NARAYAN   9937955212      GURGAON
C0008          PRANAV PRAVEEN  9336652441      PUNE
C0009          STUTI MISRA     7870266534      DELHI
C0010          ASLESHA TIWARI  null    MUMBAI
C0011          ANWESHA DAS     9999999999      BHUB
C0012          SACHIN SINGH    9898989898      CTC
C0013          ARJUN MISHRA    7777777777      BBSR
The number of rows selected is 13
```

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
3
Enter a cust_no for deletion:
C0013
1Row deleted
```

```
Problems   @ Javadoc   Declaration   Console  ×
myjdbcproject [Java Application] C:\Users\Student\.p2\pool\plugins\org.eclipse.justj.openjdk.ha

***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
3
Enter a cust_no for deletion:
C0016
0Row deleted
```
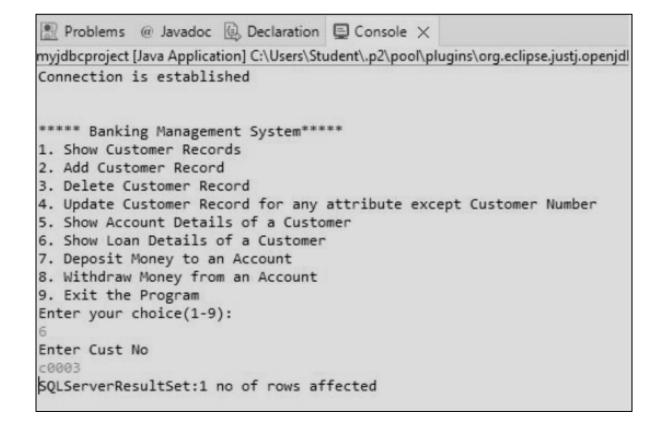
```
Problems   @ Javadoc   Declaration   Console  ×
myjdbcproject [Java Application] C:\Users\Student\.p2\pool\plugins\org.eclipse.justj.openjdk.l

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
4
Enter 1: For Name 2: For Phone no 3: For City to update:
1
Enter Cust No
C0011
Enter updated name
BIKASH
1Updated Successfully
```

```
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Num
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
4
Enter 1: For Name 2: For Phone no 3: For City to update:
2
Enter Cust No
C0011
Enter updated Phone
6370105068
1Updated Successfully
```

```
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
4
Enter 1: For Name 2: For Phone no 3: For City to update:
3
Enter Cust No
C0011
Enter updated City
ROURKELA
1Updated Successfully
```
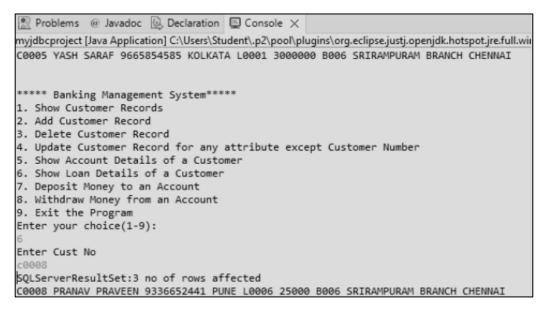
```
Problems  @ Javadoc  Declaration  Console ×
myjdbcproject [Java Application] C:\Users\Student\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_18
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
5
Enter Cust No
c0003
Acc. No.:      Type:   Balance:       Branch Code:    Branch Name:       Branch City:
A0001          SB      200000          B003            JUHU BRANCH      MUMBAI
```

```
Problems  @ Javadoc  Declaration  Console ×
myjdbcproject [Java Application] C:\Users\Student\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_18.

***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
5
Enter Cust No
c0005
Acc. No.:      Type:   Balance:       Branch Code:    Branch Name:       Branch City:
```

```
Problems  @ Javadoc  Declaration  Console  X
myjdbcproject [Java Application] C:\Users\Student\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_18:

***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
5
Enter Cust No
c0016
Acc. No.:      Type:   Balance:        Branch Code:   Branch Name:        Branch City:
```

```
Problems  @ Javadoc  Declaration  Console  X
myjdbcproject [Java Application] C:\Users\Student\.p2\pool\plugins\org.eclipse.justj.openjdl

Connection is established


***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
6
Enter Cust No
c0003
SQLServerResultSet:1 no of rows affected
```

```
SQLServerResultSet:1 no of rows affected


***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
6
Enter Cust No
c0005
SQLServerResultSet:2 no of rows affected
C0005 YASH SARAF 9665854585 KOLKATA L0001 3000000 B006 SRIRAMPURAM BRANCH CHENNAI
```

```
C0005 YASH SARAF 9665854585 KOLKATA L0001 3000000 B006 SRIRAMPURAM BRANCH CHENNAI


***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
6
Enter Cust No
c0008
SQLServerResultSet:3 no of rows affected
C0008 PRANAV PRAVEEN 9336652441 PUNE L0006 25000 B006 SRIRAMPURAM BRANCH CHENNAI
```

```
Problems   @ Javadoc  Declaration  Console ✕
myjdbcproject [Java Application] C:\Users\Student\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.\
SQLServerResultSet:3 no of rows affected
C0008 PRANAV PRAVEEN 9336652441 PUNE L0006 25000 B006 SRIRAMPURAM BRANCH CHENNAI


***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
6
Enter Cust No
c0016
SQLServerResultSet:4 no of rows affected
```

myjdbcproject [Java Application] C:\Users\Student\.p2\pool\plugins\org.eclipse.justj.openjdk

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
7
Enter Account No
a0005
Enter the amount to deposite
10000
1Balance Updated
```

myjdbcproject [Java Application] C:\Users\Student\.p2\pool\plugins\org.eclipse.justj.openjdl

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
7
Enter Account No
a0008
Enter the amount to deposite
800
1Balance Updated
```

<terminated> myjdbcproject [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (26-Jul-2022, 4:02:3

```
Connection is established


***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
8
Enter Account No
a0005
Enter the amount to deposite
10000
1 Balance Updated
com.microsoft.sqlserver.jdbc.SQLServerException: The result set is closed.
```

myjdbcproject [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (26-Jul-2022, 4:00:26 pm) [pid: 10016]

```
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
8
Enter Account No
a0008
Enter the amount to deposite
8000
Insuffient balance
```

```
Connection is established


***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
9
Visit us next time
```

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
10
Enter from 1 to 9 nothing else please again


***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
```