

## Introduction to Databases (CSE 3151)

### MINOR ASSIGNMENT-6: MINI PROJECT USING JDBC CONNECTIVITY

**Objective of this Assignment:** To design a miniature Project for a Banking Management System using Java, Oracle and JDBC.

**Requisite:**

- Completion of IDB Laboratory Assignment-4
- Basic Java Programming knowledge

**Overview of the Project:** A Banking Management System is to be designed, putting together the concepts learnt in theory and practised in laboratory. The Project will integrate a Java frontend menu driven program to the backend Banking Database designed in Oracle through JDBC connectivity.

**Project Description:** The Java program provides an interface to the user to access, insert, delete and update the database. The program handles user input, output to and from the database for the said operations. User should be able to do the following operations:

1. Show Customer Records:
  - a) Using this option the details of all the customers should be displayed in particular format.
2. Add Customer Record:
  - a) Using this option the user needs to provide the information such as cust\_no, name, phoneno and city through user input, which will be saved in database. After that using option 1, details of all the customers will be displayed in particular format.
3. Delete Customer Record:
  - a) Using this option the user needs to provide the cust\_no of a customer through user input and all the information related to that customer will be deleted from the database. After that using option 1, details of all the customers will be displayed in particular format.
4. Update Customer Information:
  - a) Using this option the user needs to provide the cust\_no of a customer through user input and based on the following choice the information related to the customer will be updated.
    - 4.1: Update name
    - 4.2: Update Phoneno.
    - 4.3: Update city
  - b) After that using option 1, details of all the customers will be displayed in particular format.
5. Show Account Details of a Customer:
  - a) Using this option the user needs to provide the cust\_no of a customer through user input and all the information of that customer along with his account\_no, type, balance, branch\_code, branch\_name and branch\_city will be displayed in proper format.
6. Show Loan Details of a Customer:
  - a) Using this option the user needs to provide the cust\_no of a customer through user input and all the information of that customer along with his loan\_no, loan amount, branch\_code, branch\_name and branch\_city will be displayed in proper format.
7. Deposit Money to an Account:
  - a) Using this option the user needs to provide the account\_no of a customer and the amount to be deposited through user input. According to the deposited amount the updated balance will be verified in proper format using option 5.
8. Withdraw Money from an Account:
  - a) Using this option the user needs to provide the account\_no of a customer and the amount to be withdraw through user input. According to the withdraw amount the updated balance will be verified in proper format using option 5.
9. Exit the Program

The operations are choice based. Appropriate option has to be chosen from a switch case based menu driven program and the operation on the database is performed accordingly. The output is displayed in the terminal screen with appropriate messages from the database as displayed by Oracle during direct access. Exceptions

should be handled properly by the Java program. The output should be displayed in a formatted way for clarity of understanding and visual.

Code:

```
import java.io.*;
import java.sql.*;
import java.util.Properties;
import java.util.Scanner;
public class BankingManagementSystem {
    public static void main(String[] args) throws IOException {
        Connection con = null;
        Statement stmt = null;
        Scanner sc = new Scanner(System.in);

        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            String conurl = "jdbc:oracle:thin:@localhost:1521:orcl19";
            Properties props = new Properties();
            props.put("user", "sys");
            props.put("password", "Oracle@+777");
            props.put("internal_logon", "sysdba");
            con = DriverManager.getConnection(conurl, props);
            stmt = con.createStatement();

            int choice;
            do {
                System.out.println("\n***** Banking Management System *****");
                System.out.println("1. Show Customer Records");
                System.out.println("2. Add Customer Record");
                System.out.println("3. Delete Customer Record");
                System.out.println("4. Update Customer Information");
                System.out.println("5. Show Account Details of a Customer");
                System.out.println("6. Show Loan Details of a Customer");
                System.out.println("7. Deposit Money to an Account");
                System.out.println("8. Withdraw Money from an Account");
                System.out.println("9. Exit");
                System.out.print("Enter your choice: ");
                choice = sc.nextInt();
                sc.nextLine();

                switch (choice) {
                    case 1:
                        ResultSet rs1 = stmt.executeQuery("SELECT CUST_NO, NAME,
                            PHONE_NO, CITY FROM CUSTOMER");
                        System.out.println("Customer No | Name | Phone | City");
                        System.out.println("-----|-----|-----|-----");
                        System.out.println("-----|-----");
                        while (rs1.next()) {
                            System.out.printf("%-11s | %-23s | %-10s | %-10s\n",
                                rs1.getString("CUST_NO"), rs1.getString("NAME"),
                                rs1.getString("PHONE_NO"), rs1.getString("CITY"));
                        }
                        rs1.close();
                        break;

                    case 2:
                        System.out.print("Enter Customer No: ");
                        String cust_no = sc.nextLine();
                        System.out.print("Enter Name: ");
                        String name = sc.nextLine();
                        System.out.print("Enter Phone Number: ");
                        String phone = sc.nextLine();
                        System.out.print("Enter City: ");
```

```

String city = sc.nextLine();
stmt.executeUpdate("INSERT INTO CUSTOMER (CUST_NO, NAME,
    PHONE_NO, CITY) VALUES ('" + cust_no + "', '" + name + "', '"
    + phone + "', '" + city + "')");
System.out.println("Customer added successfully.");
break;

case 3:
System.out.print("Enter Customer No to delete: ");
String del_cust = sc.nextLine();
stmt.executeUpdate("DELETE FROM CUSTOMER WHERE CUST_NO = '" +
    del_cust + "'");
System.out.println("Customer deleted (if existed).");
break;

case 4:
System.out.print("Enter Customer No to update: ");
String up_cust = sc.nextLine();
System.out.println("1. Update Name\n2. Update Phone No\n3.
    Update City");
int up_choice = sc.nextInt();
sc.nextLine();
switch (up_choice) {
    case 1:
        System.out.print("Enter new Name: ");
        String newName = sc.nextLine();
        stmt.executeUpdate("UPDATE CUSTOMER SET NAME = '" + newName
            + "' WHERE CUST_NO = '" + up_cust + "'");
        break;
    case 2:
        System.out.print("Enter new Phone No: ");
        String newPhone = sc.nextLine();
        stmt.executeUpdate("UPDATE CUSTOMER SET PHONE_NO = '" +
            newPhone + "' WHERE CUST_NO = '" + up_cust + "'");
        break;
    case 3:
        System.out.print("Enter new City: ");
        String newCity = sc.nextLine();
        stmt.executeUpdate("UPDATE CUSTOMER SET CITY = '" + newCity
            + "' WHERE CUST_NO = '" + up_cust + "'");
        break;
    default:
        System.out.println("Invalid update option.");
}
System.out.println("Customer info updated (if existed).");
break;

case 5:
System.out.print("Enter Customer No: ");
String acc_cust = sc.nextLine();
ResultSet rs5 = stmt.executeQuery(
    "SELECT a.ACCOUNT_NO, a.TYPE, a.BALANCE, b.BRANCH_CODE,
    b.BRANCH_NAME, b.BRANCH_CITY " + "FROM ACCOUNT a " + "JOIN
    BRANCH b ON a.BRANCH_CODE = b.BRANCH_CODE " + "JOIN DEPOSITOR
    d ON a.ACCOUNT_NO = d.ACCOUNT_NO " + "WHERE d.CUST_NO = '" +
    acc_cust + "'");
System.out.println("Account No | Type | Balance | Branch Code |
    Branch Name | Branch City");
System.out.println("-----|-----|-----|-----
    -|-----|-----");
while (rs5.next()) {
    System.out.printf("%-10s | %-4s | %-10.2f | %-11s | %-16s | %-
        12s\n",

```

```
        rs5.getString("ACCOUNT_NO"), rs5.getString("TYPE"),
        rs5.getDouble("BALANCE"), rs5.getString("BRANCH_CODE"),
        rs5.getString("BRANCH_NAME"), rs5.getString("BRANCH_CITY"));
    }
    rs5.close();
    break;

case 6:
    System.out.print("Enter Customer No: ");
    String loan_cust = sc.nextLine();
    ResultSet rs6 = stmt.executeQuery(
        "SELECT l.LOAN_NO, l.AMOUNT, b.BRANCH_CODE, b.BRANCH_NAME,
        b.BRANCH_CITY " + "FROM LOAN l JOIN BRANCH b ON
        l.BRANCH_CODE = b.BRANCH_CODE " + "WHERE l.CUST_NO = '" +
        loan_cust + "'");
    boolean hasLoan = false;
    while (rs6.next()) {
        hasLoan = true; System.out.printf("Loan No: %s | Amount: %.2f
        | Branch: %s - %s (%s)\n",
        rs6.getString("LOAN_NO"), rs6.getDouble("AMOUNT"),
        rs6.getString("BRANCH_CODE"), rs6.getString("BRANCH_NAME"),
        rs6.getString("BRANCH_CITY"));
    }
    if (!hasLoan) {
        System.out.println("Congratulations! The customer has no
        loans.");
    }
    rs6.close();
    break;

case 7:
    System.out.print("Enter Account No: ");
    String depAcc = sc.nextLine();
    System.out.print("Enter Amount to Deposit: ");
    double depAmt = sc.nextDouble();
    sc.nextLine();
    ResultSet rsAccCheck = stmt.executeQuery("SELECT ACCOUNT_NO FROM
    ACCOUNT WHERE ACCOUNT_NO = '" + depAcc + "'");
    if (rsAccCheck.next()) {
        stmt.executeUpdate("UPDATE ACCOUNT SET BALANCE = BALANCE + " +
        depAmt + " WHERE ACCOUNT_NO = '" + depAcc + "'");
        System.out.println("Amount deposited successfully.");
    } else {
        System.out.println("Account not found.");
    }
    rsAccCheck.close();
    break;

case 8:
    System.out.print("Enter Account No: ");
    String wdAcc = sc.nextLine();
    System.out.print("Enter Amount to Withdraw: ");
    double wdAmt = sc.nextDouble();
    sc.nextLine();
    ResultSet rs8 = stmt.executeQuery("SELECT BALANCE FROM ACCOUNT
    WHERE ACCOUNT_NO = '" + wdAcc + "'");
    if (rs8.next()) {
        double currBal = rs8.getDouble("BALANCE");
        if (currBal >= wdAmt) {
            stmt.executeUpdate("UPDATE ACCOUNT SET BALANCE = BALANCE - "
            + wdAmt + " WHERE ACCOUNT_NO = '" + wdAcc + "'");
            System.out.println("Amount withdrawn successfully.");
        }
    }
}
```

```

        } else {
            System.out.println("Insufficient balance!");
        }
    } else {
        System.out.println("Account not found.");
    }
    rs8.close();
    break;

case 9:
    System.out.println("Exiting program.");
    break;

default:
    System.out.println("Invalid choice. Try again.");
}
} while (choice != 9);

sc.close();
stmt.close();
con.close();
} catch (Exception e) {
    System.out.println("Error: " + e.getMessage());
}
}
}

```

Output:

```

PS D:\IDProject> javac -cp ".;ojdbc8.jar" BankingManagementSystem.java
PS D:\IDProject> java -cp ".;ojdbc17.jar" BankingManagementSystem

```

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice: 1

Customer No	Name	Phone	City
-----	-----	-----	-----
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 PRADHAN	9937955212	GURGAON
C0008	PRANAV PRAVEEN	9336652441	PUNE
C0009	STUTI MISRA	7870266534	DELHI
C0010	Hardik	null	MUMBAI

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice: 2

Enter Customer No: C0011

Enter Name: Varun

Enter Phone Number: 0000000000

Enter City: Bhubaneswar

Customer added successfully.

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice: 3

Enter Customer No to delete: C0011

Customer deleted (if existed).

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice: 4

Enter Customer No to update: C0005

1. Update Name
2. Update Phone No
3. Update City

2

Enter new Phone No: 9090909090

Customer info updated (if existed).



\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice: 5

Enter Customer No: C0010

Account No	Type	Balance	Branch Code	Branch Name	Branch City
A0006	FD	550000.00	B005	SALT LAKE BRANCH	KOLKATA

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice: 6

Enter Customer No: C0010

Loan No: L0004 | Amount: 100000.00 | Branch: B004 - ANDHERI BRANCH (MUMBAI)

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice: 7

Enter Account No: A0001

Enter Amount to Deposit: 5000

Amount deposited successfully.

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice: 8

Enter Account No: A0001

Enter Amount to Withdraw: 10000

Amount withdrawn successfully.

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice: 9

Exiting program.

.PS D:\IDProject> █