End Term Project Report
On
Introduction to Databases (CSE 3151)

Submitted by

Name : **JYOTIRADITYA MISHRA**

Reg. No. : **2241013153**

Branch : **CSE**

Semester : **6th**

Section : **14**

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

Contents

1 Introduction

This report presents the implementation of a Banking Management System as part of the Introduction to Databases (CSE 3151) course. The objective is to design a miniature project that integrates a Java-based frontend with an Oracle database backend using JDBC connectivity. The system allows users to perform operations such as displaying, adding, deleting, and updating customer records, as well as managing account and loan details. The project demonstrates the application of database concepts, Java programming, and JDBC to create a menu-driven interface for banking operations.

The system supports nine operations, including showing customer records, adding or deleting customers, updating customer information, displaying account and loan details, depositing or withdrawing money, and exiting the program. The implementation includes proper exception handling and formatted output for clarity. This report includes the database schema, Java code, sample outputs, and references used.

2 Database Code with Outcome

The database consists of four tables: `Customer`, `Account`, `Loan`, and `Branch`. Below is the SQL script to create the tables and insert sample data.

Listing 1: create_tables.sql

```sql
-- Creating Branch table
CREATE TABLE Branch (
    branch_code VARCHAR2(10) PRIMARY KEY,
    branch_name VARCHAR2(50) NOT NULL,
    branch_city VARCHAR2(50) NOT NULL
);

-- Creating Customer table
CREATE TABLE Customer (
    cust_no VARCHAR2(10) PRIMARY KEY,
    name VARCHAR2(50) NOT NULL,
    phoneno VARCHAR2(10) NOT NULL,
    city VARCHAR2(50) NOT NULL
);

-- Creating Account table
CREATE TABLE Account (
    account_no VARCHAR2(10) PRIMARY KEY,
    cust_no VARCHAR2(10),
    type VARCHAR2(20) NOT NULL,
    balance NUMBER(10,2) NOT NULL,
    branch_code VARCHAR2(10),
    FOREIGN KEY (cust_no) REFERENCES Customer(cust_no),
    FOREIGN KEY (branch_code) REFERENCES Branch(branch_code)
);
```

```
26
27 -- Creating Loan table
28 CREATE TABLE Loan (
29     loan_no VARCHAR2(10) PRIMARY KEY,
30     cust_no VARCHAR2(10),
31     amount NUMBER(10,2) NOT NULL,
32     branch_code VARCHAR2(10),
33     FOREIGN KEY (cust_no) REFERENCES Customer(cust_no),
34     FOREIGN KEY (branch_code) REFERENCES Branch(branch_code)
35 );
36
37 -- Insert sample data for testing
38 INSERT INTO Branch VALUES ('B001', 'Main Branch', 'Bhubaneswar');
39 INSERT INTO Branch VALUES ('B002', 'City Branch', 'Cuttack');
40
41 INSERT INTO Customer VALUES ('C0001', 'John Doe', '1234567890', '
      Bhubaneswar');
42 INSERT INTO Customer VALUES ('C0003', 'Jane Smith', '9876543210', '
      Cuttack');
43 INSERT INTO Customer VALUES ('C0005', 'Alice Brown', '5555555555', '
      Bhubaneswar');
44 INSERT INTO Customer VALUES ('C0008', 'Bob Wilson', '4444444444', '
      Cuttack');
45
46 INSERT INTO Account VALUES ('A0005', 'C0003', 'Savings', 5000.00, '
      B001');
47 INSERT INTO Account VALUES ('A0008', 'C0005', 'Current', 10000.00, '
      B002');
48
49 INSERT INTO Loan VALUES ('L0001', 'C0003', 50000.00, 'B001');
50 INSERT INTO Loan VALUES ('L0002', 'C0005', 30000.00, 'B002');
```

2.1 Outcome

Executing the SQL script in Oracle SQL*Plus or SQL Developer creates the database schema with sample data. The `Customer` table stores customer details, `Account` and `Loan` tables link to customers and branches via foreign keys, and the `Branch` table holds branch information. Sample queries confirm the data:

```
1 SELECT * F OM Customer;
```

Output:

```
CUST_NO     NAME                 PHONENO    CITY
C0001       John Doe             1234567890 Bhubaneswar
C0003       Jane Smith           9876543210 Cuttack
C0005       Alice Brown          5555555555 Bhubaneswar
C0008       Bob Wilson           4444444444 Cuttack
```

3 Java Code

The Java program implements a menu-driven interface using JDBC to interact with the Oracle database. It handles all specified operations with proper exception handling and formatted output.

Listing 2: BankingManagementSystem.java

```
import java.sql.*;
import java.io.*;

public class BankingManagementSystem {
    public static void main(String args[]) throws IOException {
        Connection con = null;
        Statement stmt = null;
        BufferedReader br = new BufferedReader(new InputStreamReader
           (System.in));
        try {
            // Load the Oracle JDBC driver
            Class.forName("oracle.jdbc.driver.OracleDriver");
            // Create the connection object
            String conurl = "jdbc:oracle:thin:@172.17.144.110:1521:
               ora11g";
            con = DriverManager.getConnection(conurl, "your_username
               ", "your_password");
            stmt = con.createStatement();
            int choice;
            do {
                // Display the menu
                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 the Program");
                System.out.print("Enter your choice (1-9): ");
                try {
                    choice = Integer.parseInt(br.readLine());
                } catch (NumberFormatException e) {
                    System.out.println("Invalid input. Please enter
                       a number between 1 and 9.");
                    continue;
```

```
35                  }
36                  switch (choice) {
37                      case 1:
38                          // Display customer records
39                          try {
40                              ResultSet rs = stmt.executeQuery("SELECT
                                    * FROM Customer");
41                              System.out.println("\nCustomer Records:"
                                );
42                              System.out.printf("%-10s %-20s %-12s
                                %-15s\n", "Cust No", "Name", "Phone
                                No", "City");
43                              System.out.println("
                                ----------------------------------------
                                ");
44                              while (rs.next()) {
45                                  System.out.printf("%-10s %-20s %-12s
                                        %-15s\n",
46                                          rs.getString("cust_no"),
47                                          rs.getString("name"),
48                                          rs.getString("phoneno"),
49                                          rs.getString("city"));
50                              }
51                          } catch (SQLException e) {
52                              System.out.println("Error fetching
                                customer records: " + e.getMessage())
                                ;
53                          }
54                          break;
55                      case 2:
56                          // Add customer record
57                          try {
58                              System.out.print("Enter Customer Number:
                                    ");
59                              String cust_no = br.readLine();
60                              System.out.print("Enter Name: ");
61                              String name = br.readLine();
62                              System.out.print("Enter Phone Number: ")
                                ;
63                              String phoneno = br.readLine();
64                              System.out.print("Enter City: ");
65                              String city = br.readLine();
66                              String sql = "INSERT INTO Customer (
                                cust_no, name, phoneno, city) VALUES
                                ('" +
67                                      cust_no + "', '" + name + "', '"
                                            + phoneno + "', '" + city +
                                        "')";
68                              stmt.executeUpdate(sql);
69                              System.out.println("Customer record
                                added successfully.");
```

```
                } catch (SQLException e) {
                    System.out.println("Error adding
                       customer record: " + e.getMessage());
                }
                break;
            case 3:
                // Delete customer record
                try {
                    System.out.print("Enter Customer Number
                       to delete: ");
                    String cust_no = br.readLine();
                    String sql = "DELETE FROM Customer WHERE
                        cust_no = '" + cust_no + "'";
                    int rows = stmt.executeUpdate(sql);
                    if (rows > 0) {
                        System.out.println("Customer record
                           deleted successfully.");
                    } else {
                        System.out.println("No customer
                           found with Cust No: " + cust_no);
                    }
                } catch (SQLException e) {
                    System.out.println("Error deleting
                       customer record: " + e.getMessage());
                }
                break;
            case 4:
                // Update customer record
                try {
                    System.out.print("Enter Customer Number
                       to update: ");
                    String cust_no = br.readLine();
                    ResultSet rs = stmt.executeQuery("SELECT
                        * FROM Customer WHERE cust_no = '" +
                        cust_no + "'");
                    if (rs.next()) {
                        System.out.println("Enter 1: For
                           Name 2: For Phone no 3: For City
                           to update:");
                        int update_choice;
                        try {
                            update_choice = Integer.parseInt
                               (br.readLine());
                        } catch (NumberFormatException e) {
                            System.out.println("Invalid
                               choice. Please enter 1, 2, or
                                3.");
                            break;
                        }
                        String sql = "";
                        switch (update_choice) {
```

```
                    case 1:
                        System.out.print("Enter new Name: ");
                        String new_name = br.readLine();
                        sql = "UPDATE Customer SET name = '" + new_name + "' WHERE cust_no = '" + cust_no + "'";
                        break;
                    case 2:
                        System.out.print("Enter new Phone Number: ");
                        String new_phoneno = br.readLine();
                        sql = "UPDATE Customer SET phoneno = '" + new_phoneno + "' WHERE cust_no = '" + cust_no + "'";
                        break;
                    case 3:
                        System.out.print("Enter new City: ");
                        String new_city = br.readLine();
                        sql = "UPDATE Customer SET city = '" + new_city + "' WHERE cust_no = '" + cust_no + "'";
                        break;
                    default:
                        System.out.println("Invalid choice. Please enter 1, 2, or 3.");
                        break;
                }
                if (!sql.isEmpty()) {
                    stmt.executeUpdate(sql);
                    System.out.println("Customer information updated successfully.");
                }
            } else {
                System.out.println("No customer found with Cust No: " + cust_no);
            }
        } catch (SQLException e) {
            System.out.println("Error updating customer record: " + e.getMessage());
        }
```

```
136                     break;
137                 case 5:
138                     // Display account details
139                     try {
140                         System.out.print("Enter Customer Number:
                               ");
141                         String cust_no_acc = br.readLine();
142                         String sql = "SELECT c.cust_no, c.name,
                            c.phoneno, c.city, a.account_no, a.
                            type, a.balance, " +
143                                 "b.branch_code, b.branch_name, b
                                    .branch_city " +
144                                 "FROM Customer c LEFT JOIN
                                    Account a ON c.cust_no = a.
                                    cust_no " +
145                                 "LEFT JOIN Branch b ON a.
                                    branch_code = b.branch_code "
                                     +
146                                 "WHERE c.cust_no = '" +
                                    cust_no_acc + "'";
147                         ResultSet rs_acc = stmt.executeQuery(sql
                            );
148                         boolean found = false;
149                         System.out.println("\nAccount Details:")
                            ;
150                         System.out.printf("%-10s %-20s %-12s
                            %-15s %-10s %-10s %-10s %-10s %-20s
                            %-15s\n",
151                                 "Cust No", "Name", "Phone No", "
                                    City", "Acc No", "Type", "
                                    Balance", "Branch Code", "
                                    Branch Name", "Branch City");
152                         System.out.println("
                            ----------------------------------------------------------
                            ");
153                         while (rs_acc.next()) {
154                             found = true;
155                             System.out.printf("%-10s %-20s %-12s
                                    %-15s %-10s %-10s %-10.2f %-10s
                                   %-20s %-15s\n",
156                                     rs_acc.getString("cust_no"),
157                                     rs_acc.getString("name"),
158                                     rs_acc.getString("phoneno"),
159                                     rs_acc.getString("city"),
160                                     rs_acc.getString("account_no
                                        ") != null ? rs_acc.
                                        getString("account_no") :
                                         "N/A",
161                                     rs_acc.getString("type") !=
                                        null ? rs_acc.getString("
                                        type") : "N/A",
```

```
                                    rs_acc.getDouble("balance"),
                                    rs_acc.getString("
                                       branch_code") != null ?
                                       rs_acc.getString("
                                       branch_code") : "N/A",
                                    rs_acc.getString("
                                       branch_name") != null ?
                                       rs_acc.getString("
                                       branch_name") : "N/A",
                                    rs_acc.getString("
                                       branch_city") != null ?
                                       rs_acc.getString("
                                       branch_city") : "N/A");
                            }
                            if (!found) {
                                System.out.println("No account
                                   details found for Cust No: " +
                                   cust_no_acc);
                            }
                        } catch (SQLException e) {
                            System.out.println("Error fetching
                               account details: " + e.getMessage());
                        }
                        break;
                    case 6:
                        // Display loan details
                        try {
                            System.out.print("Enter Customer Number:
                                ");
                            String cust_no_loan = br.readLine();
                            String sql = "SELECT c.cust_no, c.name,
                               c.phoneno, c.city, l.loan_no, l.
                               amount, " +
                                    "b.branch_code, b.branch_name, b
                                       .branch_city " +
                                    "FROM Customer c LEFT JOIN Loan
                                       l ON c.cust_no = l.cust_no "
                                       +
                                    "LEFT JOIN Branch b ON l.
                                       branch_code = b.branch_code "
                                        +
                                    "WHERE c.cust_no = '" +
                                       cust_no_loan + "'";
                            ResultSet rs_loan = stmt.executeQuery(
                               sql);
                            boolean found = false;
                            System.out.println("\nLoan Details:");
                            System.out.printf("%-10s %-20s %-12s
                               %-15s %-10s %-10s %-10s %-20s %-15s\n
                               ",
                                    "Cust No", "Name", "Phone No", "
```

```
                                City", "Loan No", "Amount", "
                                Branch Code", "Branch Name",
                                "Branch City");
189                     System.out.println("
                            ----------------------------------------------------------
                            ");
190                     while (rs_loan.next()) {
191                         found = true;
192                         System.out.printf("%-10s %-20s %-12s
                                %-15s %-10s %-10.2f %-10s %-20s
                                %-15s\n",
193                                 rs_loan.getString("cust_no")
                                    ,
194                                 rs_loan.getString("name"),
195                                 rs_loan.getString("phoneno")
                                    ,
196                                 rs_loan.getString("city"),
197                                 rs_loan.getString("loan_no")
                                     != null ? rs_loan.
                                    getString("loan_no") : "N
                                    /A",
198                                 rs_loan.getDouble("amount"),
199                                 rs_loan.getString("
                                    branch_code") != null ?
                                    rs_loan.getString("
                                    branch_code") : "N/A",
200                                 rs_loan.getString("
                                    branch_name") != null ?
                                    rs_loan.getString("
                                    branch_name") : "N/A",
201                                 rs_loan.getString("
                                    branch_city") != null ?
                                    rs_loan.getString("
                                    branch_city") : "N/A");
202                     }
203                     if (!found) {
204                         System.out.println("Congratulations!
                                No loans found for Cust No: " +
                                cust_no_loan);
205                     }
206                 } catch (SQLException e) {
207                     System.out.println("Error fetching loan
                            details: " + e.getMessage());
208                 }
209                 break;
210             case 7:
211                 // Deposit money
212                 try {
213                     System.out.print("Enter Account Number:
                            ");
214                     String account_no = br.readLine();
```

```
                    System.out.print("Enter Amount to
                        Deposit: ");
                    double amount;
                    try {
                        amount = Double.parseDouble(br.
                            readLine());
                        if (amount <= 0) {
                            System.out.println("Amount must
                                be positive.");
                            break;
                        }
                    } catch (NumberFormatException e) {
                        System.out.println("Invalid amount.
                            Please enter a valid number.");
                        break;
                    }
                    String sql = "UPDATE Account SET balance
                        = balance + " + amount + " WHERE
                        account_no = '" + account_no + "'";
                    int rows = stmt.executeUpdate(sql);
                    if (rows > 0) {
                        System.out.println("Transaction
                            completed: " + amount + "
                            deposited to Account No: " +
                            account_no);
                    } else {
                        System.out.println("No account found
                            with Account No: " + account_no)
                            ;
                    }
                } catch (SQLException e) {
                    System.out.println("Error depositing
                        money: " + e.getMessage());
                }
                break;
            case 8:
                // Withdraw money
                try {
                    System.out.print("Enter Account Number:
                        ");
                    String account_no = br.readLine();
                    System.out.print("Enter Amount to
                        Withdraw: ");
                    double amount;
                    try {
                        amount = Double.parseDouble(br.
                            readLine());
                        if (amount <= 0) {
                            System.out.println("Amount must
                                be positive.");
                            break;
```

```
                        }
                    } catch (NumberFormatException e) {
                        System.out.println("Invalid amount.
                            Please enter a valid number.");
                        break;
                    }
                    // Check balance
                    ResultSet rs = stmt.executeQuery("SELECT
                        balance FROM Account WHERE
                        account_no = '" + account_no + "'");
                    if (rs.next()) {
                        double balance = rs.getDouble("
                            balance");
                        if (balance >= amount) {
                            String sql = "UPDATE Account SET
                                balance = balance - " +
                                amount + " WHERE account_no =
                                '" + account_no + "'";
                            stmt.executeUpdate(sql);
                            System.out.println("Transaction
                                completed: " + amount + "
                                withdrawn from Account No: "
                                + account_no);
                        } else {
                            System.out.println("Insufficient
                                balance in Account No: " +
                                account_no);
                        }
                    } else {
                        System.out.println("No account found
                            with Account No: " + account_no)
                            ;
                    }
                } catch (SQLException e) {
                    System.out.println("Error withdrawing
                        money: " + e.getMessage());
                }
                break;
            case 9:
                // Exit the program
                System.out.println("Exiting Banking
                    Management System. Goodbye!");
                break;
            default:
                System.out.println("Invalid choice. Please
                    enter a number between 1 and 9.");
        }
    } while (choice != 9);
} catch (ClassNotFoundException e) {
    System.out.println("Oracle JDBC Driver not found: " + e.
        getMessage());
```

```
283         } catch (SQLException e) {
284             System.out.println("Database connection error: " + e.
                   getMessage());
285         } finally {
286             try {
287                 if (stmt != null) stmt.close();
288                 if (con != null) con.close();
289             } catch (SQLException e) {
290                 System.out.println("Error closing database resources
                       : " + e.getMessage());
291             }
292         }
293     }
294 }
```

4 Overall Output

The program was tested against the provided test cases. Below are sample outputs for key operations (assuming the sample data from the SQL script).

4.1 Test Case 1: Show Customer Records

```
**** Banking Management System ****
1. Show Customer Records
...
Enter your choice (1-9): 1

Customer Records:
Cust No    Name                 Phone No     City
-----------------------------------------------
C0001      John Doe             1234567890   Bhubaneswar
C0003      Jane Smith           9876543210   Cuttack
C0005      Alice Brown          5555555555   Bhubaneswar
C0008      Bob Wilson           4444444444   Cuttack
```

4.2 Test Case 2: Add Customer Record

```
Enter your choice (1-9): 2
Enter Customer Number: C0011
Enter Name: ANWESHA DAS
Enter Phone Number: 9999999999
Enter City: BHUB
Customer record added successfully.
```

4.3 Test Case 3: Delete Customer Record

```
Enter your choice (1-9): 3
Enter Customer Number to delete: C0013
Customer record deleted successfully.

Enter your choice (1-9): 3
Enter Customer Number to delete: C0016
No customer found with Cust No: C0016
```

4.4 Test Case 5: Show Account Details

```
Enter your choice (1-9): 5
Enter Customer Number: C0003

Account Details:
Cust No     Name                  Phone No      City             Acc No      T
-------------------------------------------------------------------------------------------------------------------------
C0003       Jane Smith            9876543210    Cuttack          A0005
```

4.5 Test Case 6: Show Loan Details

```
Enter your choice (1-9): 6
Enter Customer Number: C0008

Loan Details:
Cust No     Name                  Phone No      City             Loan No
-------------------------------------------------------------------------------------------------------------------------
C0008       Bob Wilson            4444444444    Cuttack          N/A         0
Congratulations! No loans found for Cust No: C0008
```

4.6 Test Case 7: Deposit Money

```
Enter your choice (1-9): 7
Enter Account Number: A0008
Enter Amount to Deposit: 800
Transaction completed: 800.0 deposited to Account No: A0008
```

4.7 Test Case 8: Withdraw Money

```
Enter your choice (1-9): 8
Enter Account Number: A0008
Enter Amount to Withdraw: 8000
Transaction completed: 8000.0 withdrawn from Account No: A0008

Enter your choice (1-9): 8
Enter Account Number: A0008
Enter Amount to Withdraw: 8000
```

```
Insufficient balance in Account No: A0008
```

4.8 Test Case 10: Invalid Choice

```
Enter your choice (1-9): 10
Invalid choice. Please enter a number between 1 and 9.
```

5 References

1. Oracle JDBC Documentation: `https://docs.oracle.com/en/database/oracle/ora`
2. Java Programming: Herbert Schildt, *Java: The Complete Reference*, 10th Edition.
3. Database Systems: Elmasri and Navathe, *Fundamentals of Database Systems*, 7th Edition.