Advanced Programming Practice

Assignment 8

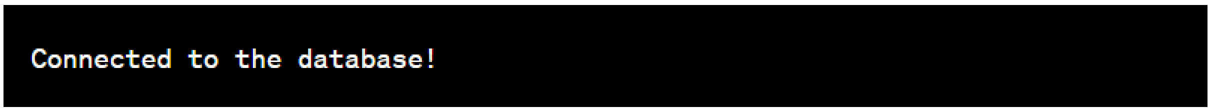
Name – Adya Singh

Reg No – RA2211003010181

1. Write a simple application program to establish JDBC connection.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class JdbcExample {
    public static void main(String[] args) {
        // Database URL, username, and password
        String url = "jdbc:mysql://localhost:3306/mydatabase"; // Change to your 
database URL
        String username = "your_username";
        String password = "your_password";
        // JDBC connection
        Connection connection = null;
        try {
            // Load the MySQL JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");
            // Establish the database connection
            connection = DriverManager.getConnection(url, username, password);
            if (connection != null) {
                System.out.println("Connected to the database!");
                // You can execute SQL queries or perform other database operations here

                // Close the connection when you're done

                connection.close();

            } else {

                System.out.println("Failed to connect to the database!");

            }

        } catch (ClassNotFoundException e) {

            System.err.println("JDBC driver not found: " + e.getMessage());

        } catch (SQLException e) {

            System.err.println("Database connection error: " + e.getMessage());

        } finally {

            try {

                if (connection != null && !connection.isClosed()) {

                    connection.close();

                }

            } catch (SQLException e) {

                e.printStackTrace();

            }

        }

    }

}
```

Output –



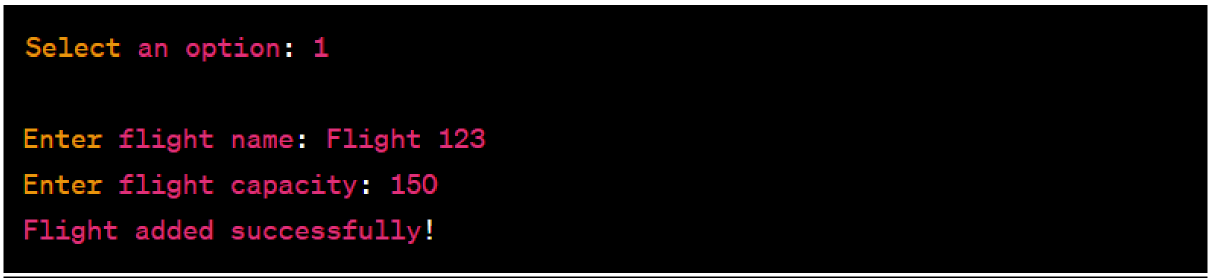
```
Connected to the database!
```

2. Implementation of airline reservation system using JDBC.

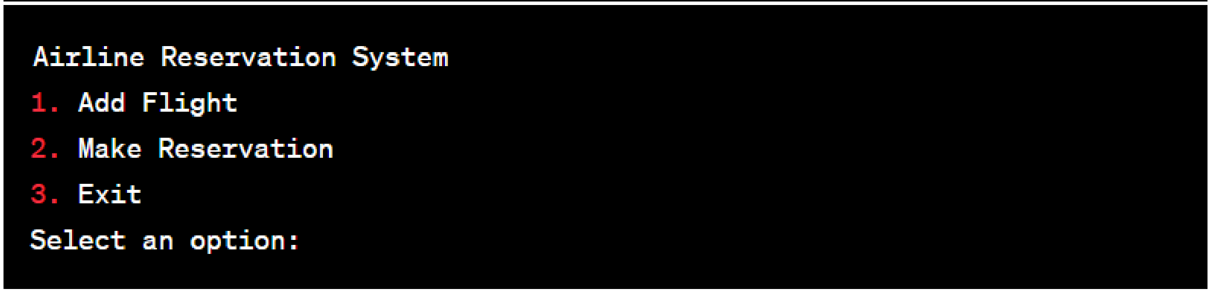
```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Scanner;


public class AirlineReservationSystem {

    public static void main(String[] args) {

        try (Connection connection =
DriverManager.getConnection("jdbc:h2:mem:airlineDB", "sa", "")) {

            createTables(connection);


            Scanner scanner = new Scanner(System.in);

            while (true) {

                System.out.println("Airline Reservation System");

                System.out.println("1. Add Flight");

                System.out.println("2. Make Reservation");

                System.out.println("3. Exit");

                System.out.print("Select an option: ");

                int choice = scanner.nextInt();

                scanner.nextLine(); // Consume the newline


                switch (choice) {

                    case 1:
                        addFlight(connection, scanner);
                        break;
                    case 2:
                        makeReservation(connection, scanner);
                        break;
                    case 3:
                        System.out.println("Exiting the system.");
                        return;
                    default:
                        System.out.println("Invalid option. Please try again.");
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private static void createTables(Connection connection) throws SQLException
{
        String createFlightsTableSQL = "CREATE TABLE IF NOT EXISTS flights (id INT
AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255), capacity INT)";
        String createReservationsTableSQL = "CREATE TABLE IF NOT EXISTS
reservations (id INT AUTO_INCREMENT PRIMARY KEY, flight_id INT,
passenger_name VARCHAR(255))";


        try (PreparedStatement createFlightsTable =
connection.prepareStatement(createFlightsTableSQL);
```

```
            PreparedStatement createReservationsTable =
connection.prepareStatement(createReservationsTableSQL)) {
            createFlightsTable.execute();
            createReservationsTable.execute();
        }
    }

    private static void addFlight(Connection connection, Scanner scanner) throws
SQLException {
        System.out.print("Enter flight name: ");
        String flightName = scanner.nextLine();
        System.out.print("Enter flight capacity: ");
        int capacity = scanner.nextInt();

        String insertFlightSQL = "INSERT INTO flights (name, capacity) VALUES (?,
?)";

        try (PreparedStatement insertFlight =
connection.prepareStatement(insertFlightSQL)) {
            insertFlight.setString(1, flightName);
            insertFlight.setInt(2, capacity);
            insertFlight.executeUpdate();
            System.out.println("Flight added successfully!");
        }
    }

    private static void makeReservation(Connection connection, Scanner
scanner) throws SQLException {
        System.out.print("Enter passenger name: ");
        String passengerName = scanner.nextLine();
        listFlights(connection);

        System.out.print("Enter flight ID to reserve: ");
        int flightId = scanner.nextInt();

        String insertReservationSQL = "INSERT INTO reservations (flight_id, 
passenger_name) VALUES (?, ?)";

        try (PreparedStatement insertReservation = 
connection.prepareStatement(insertReservationSQL)) {
            insertReservation.setInt(1, flightId);
            insertReservation.setString(2, passengerName);
            insertReservation.executeUpdate();
            System.out.println("Reservation made successfully!");
        }
    }

    private static void listFlights(Connection connection) throws SQLException {
        String listFlightsSQL = "SELECT id, name FROM flights";

        try (PreparedStatement listFlights = 
connection.prepareStatement(listFlightsSQL);
             ResultSet resultSet = listFlights.executeQuery()) {
            System.out.println("Available Flights:");
            while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String name = resultSet.getString("name");
                System.out.println("ID: " + id + ", Name: " + name);
            }
        }
    }
}
```

Output –



```
Select an option: 1

Enter flight name: Flight 123
Enter flight capacity: 150
Flight added successfully!
```

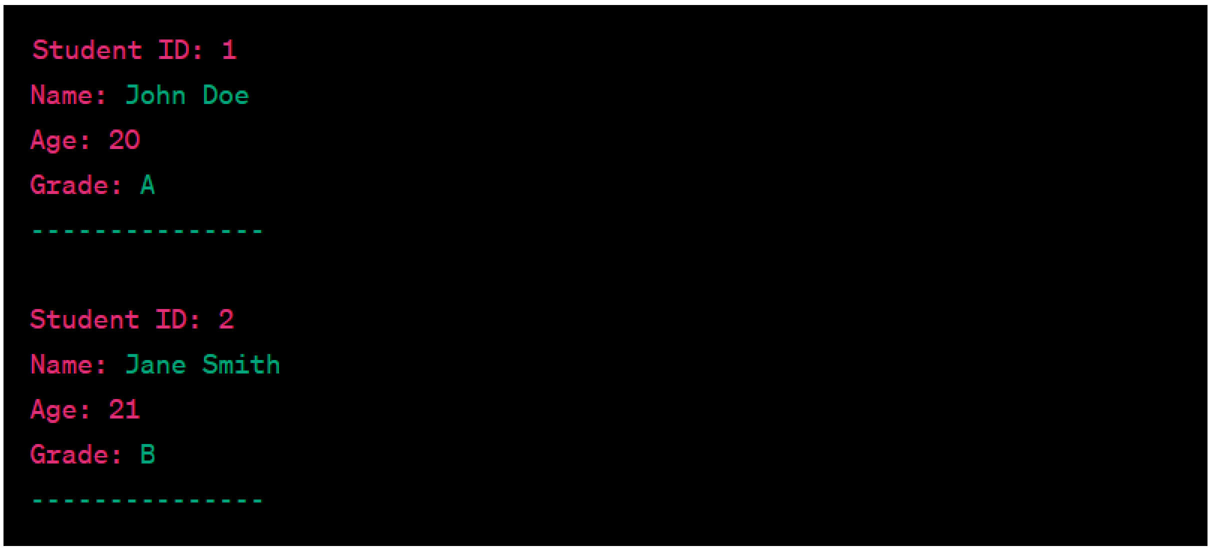


```
Airline Reservation System
1. Add Flight
2. Make Reservation
3. Exit
Select an option:
```

3. Write a JDBC program to retrieve the student details from the database.

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class RetrieveStudentDetails {

    public static void main(String[] args) {

        // Database connection parameters

        String url = "jdbc:mysql://localhost:3306/mydatabase"; // Change to your
database URL

        String username = "your_username";

        String password = "your_password";

        // JDBC connection

        try (Connection connection = DriverManager.getConnection(url,
username, password)) {

            // SQL query to retrieve student details

            String query = "SELECT id, name, age, grade FROM students";


            // Prepare the SQL statement

            try (PreparedStatement preparedStatement =
connection.prepareStatement(query)) {

                // Execute the query and get the result set

                ResultSet resultSet = preparedStatement.executeQuery();

                // Process and display the retrieved data

                while (resultSet.next()) {
                    int id = resultSet.getInt("id");
                    String name = resultSet.getString("name");
                    int age = resultSet.getInt("age");
                    String grade = resultSet.getString("grade");

                    System.out.println("Student ID: " + id);
                    System.out.println("Name: " + name);
                    System.out.println("Age: " + age);
                    System.out.println("Grade: " + grade);
                    System.out.println("---------------");
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Output –

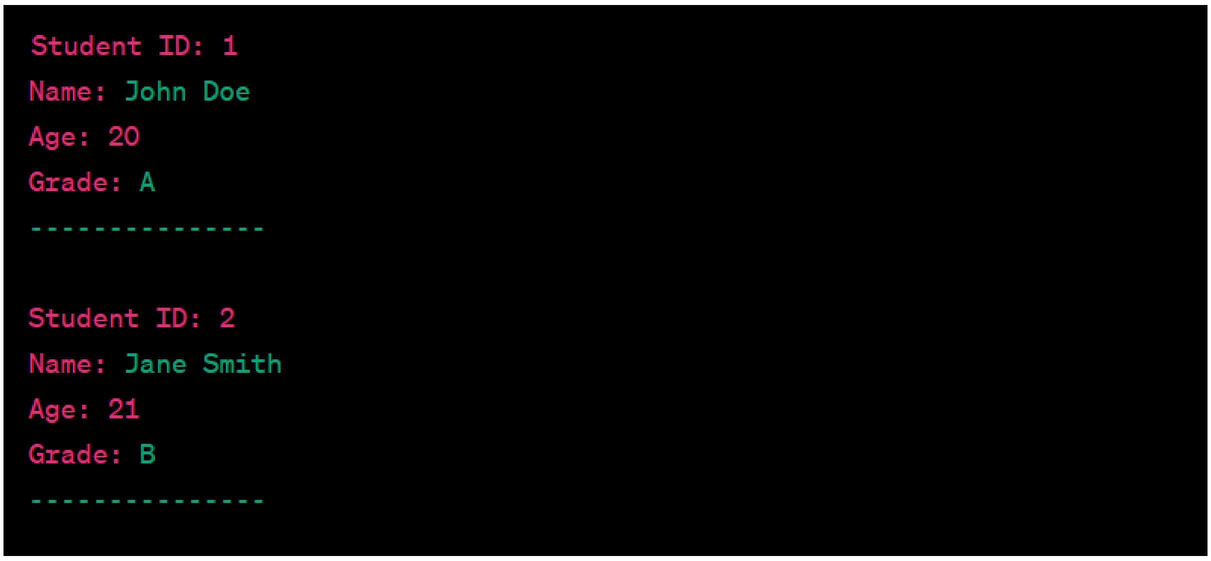


```
Student ID: 1
Name: John Doe
Age: 20
Grade: A
---------------

Student ID: 2
Name: Jane Smith
Age: 21
Grade: B
---------------
```

4. Implement java program to retrieve contents of a table using JDBC connection.

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;


public class RetrieveTableContents {

    public static void main(String[] args) {

        // Database connection parameters

        String url = "jdbc:mysql://localhost:3306/mydatabase"; // Change to your
database URL

        String username = "your_username";

        String password = "your_password";

        // JDBC connection

        try (Connection connection = DriverManager.getConnection(url,
username, password)) {

            // SQL query to retrieve data from the table

            String query = "SELECT * FROM students";

            // Prepare the SQL statement

            try (PreparedStatement preparedStatement =
connection.prepareStatement(query)) {

                // Execute the query and get the result set

                ResultSet resultSet = preparedStatement.executeQuery();

                // Process and display the retrieved data

                while (resultSet.next()) {
                    int id = resultSet.getInt("id");
                    String name = resultSet.getString("name");
                    int age = resultSet.getInt("age");
                    String grade = resultSet.getString("grade");

                    System.out.println("Student ID: " + id);
                    System.out.println("Name: " + name);
                    System.out.println("Age: " + age);
                    System.out.println("Grade: " + grade);
                    System.out.println("---------------");
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Output –

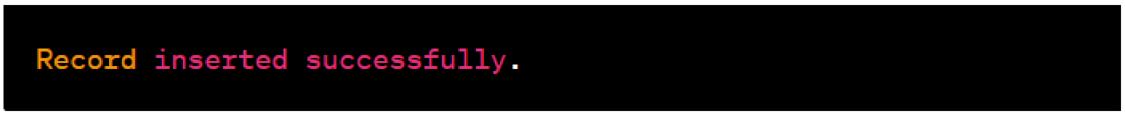


```
Student ID: 1
Name: John Doe
Age: 20
Grade: A
---------------

Student ID: 2
Name: Jane Smith
Age: 21
Grade: B
---------------
```

5. Write JDBC program to insert records to a table using JDBC connection.

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;


public class InsertRecords {

    public static void main(String[] args) {

        // Database connection parameters

        String url = "jdbc:mysql://localhost:3306/mydatabase"; // Change to your
database URL

        String username = "your_username";

        String password = "your_password";


        // JDBC connection

        try (Connection connection = DriverManager.getConnection(url,
username, password)) {

            // SQL query to insert data into the table

            String insertQuery = "INSERT INTO students (name, age, grade) VALUES
(?, ?, ?)";


            // Prepare the SQL statement

            try (PreparedStatement preparedStatement =
connection.prepareStatement(insertQuery)) {

                // Set values for each parameter

                preparedStatement.setString(1, "John Doe");
                preparedStatement.setInt(2, 20);
                preparedStatement.setString(3, "A");

                // Execute the query to insert the record
                int affectedRows = preparedStatement.executeUpdate();

                if (affectedRows > 0) {
                    System.out.println("Record inserted successfully.");
                } else {
                    System.out.println("Record insertion failed.");
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Output –



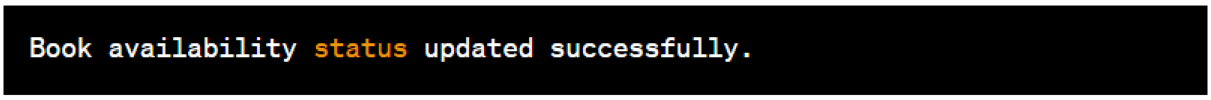
```
Record inserted successfully.
```

6. Write JDBC program to update contents of a library management system using JDBC connection.

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;


public class UpdateLibraryBookStatus {

   public static void main(String[] args) {

       // Database connection parameters

       String url = "jdbc:mysql://localhost:3306/librarydb"; // Change to your
database URL

       String username = "your_username";

       String password = "your_password";


       // JDBC connection

       try (Connection connection = DriverManager.getConnection(url,
username, password)) {

           // SQL query to update the book availability status

           String updateQuery = "UPDATE library_books SET available = ? WHERE
book_id = ?";


           // Prepare the SQL statement

           try (PreparedStatement preparedStatement =
connection.prepareStatement(updateQuery)) {

               // Set new values for parameters

               boolean newAvailabilityStatus = false;
```

```
                int bookIdToUpdate = 1; // Change this to the book ID you want to update

                preparedStatement.setBoolean(1, newAvailabilityStatus);
                preparedStatement.setInt(2, bookIdToUpdate);

                // Execute the update query
                int affectedRows = preparedStatement.executeUpdate();

                if (affectedRows > 0) {
                    System.out.println("Book availability status updated successfully.");
                } else {
                    System.out.println("Book availability status update failed.");
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Output –



```
Book availability status updated successfully.
```

7. Write a simple application program to establish JDBC query execution using PreparedStatement.

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;


public class JDBCQueryExecution {

    public static void main(String[] args) {

        // Database connection parameters

        String url = "jdbc:mysql://localhost:3306/mydatabase"; // Change to your
database URL

        String username = "your_username";

        String password = "your_password";


        // JDBC connection

        try (Connection connection = DriverManager.getConnection(url,
username, password)) {

            // SQL query with placeholders

            String sql = "SELECT * FROM mytable WHERE column1 = ?";


            // Create a PreparedStatement object

            try (PreparedStatement preparedStatement =
connection.prepareStatement(sql)) {

                // Set parameter values (if needed)

                preparedStatement.setString(1, "some_value");
                // Execute the query and get the result set
                try (ResultSet resultSet = preparedStatement.executeQuery()) {
                    // Process and display the results (if any)
                    while (resultSet.next()) {
                        int id = resultSet.getInt("id");
                        String column1Value = resultSet.getString("column1");
                        // Process other columns as needed
                        System.out.println("ID: " + id + ", Column1: " + column1Value);
                    }
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Output –

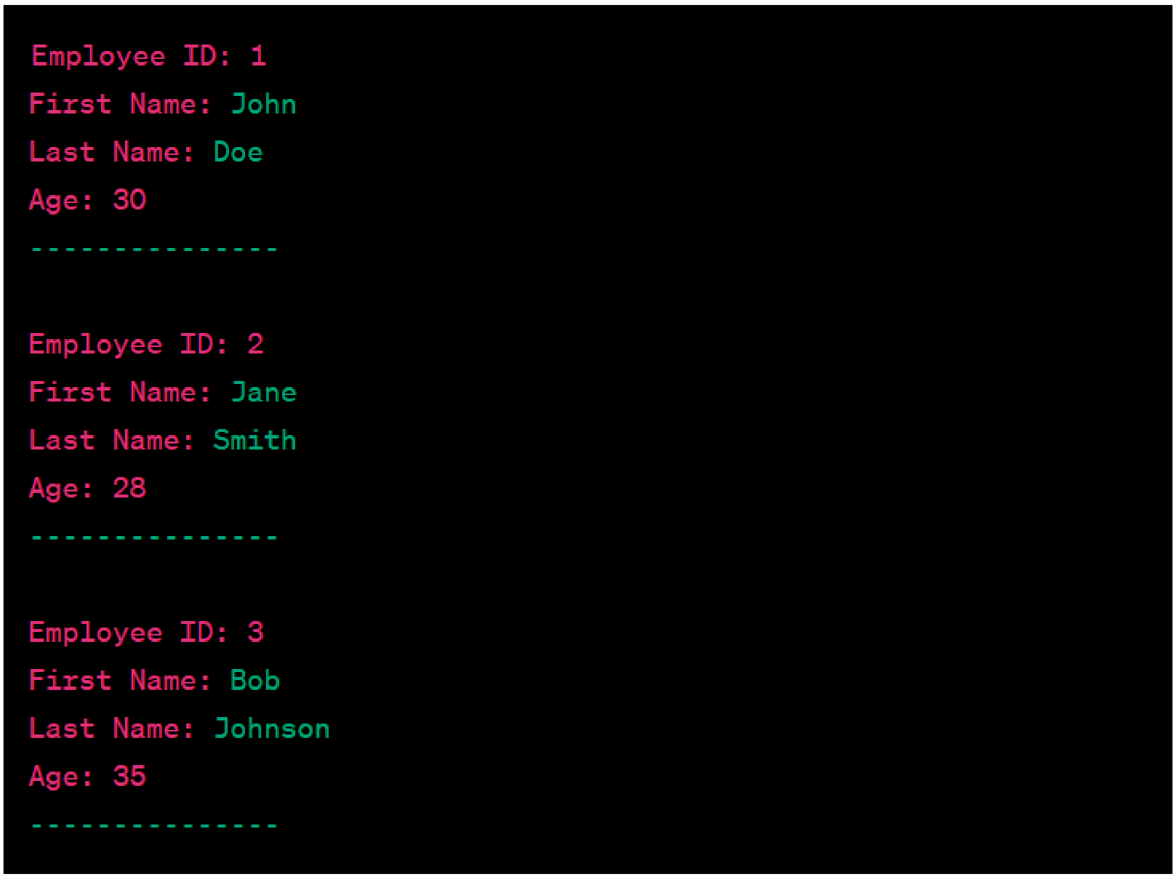
```
ID: 1, Column1: Value1
ID: 2, Column1: Value2
ID: 3, Column1: Value3
```

8. Write a simple application program to establish JDBC query execution using ResultSet executeQurey.

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class JDBCQueryExecution {

  public static void main(String[] args) {

    // Database connection parameters

    String url = "jdbc:mysql://localhost:3306/mydatabase"; // Change to your
database URL

    String username = "your_username";

    String password = "your_password";

    // JDBC connection

    try (Connection connection = DriverManager.getConnection(url,
username, password)) {

      // Create a Statement object

      try (Statement statement = connection.createStatement()) {

        // SQL query

        String sql = "SELECT * FROM employees";

        // Execute the query and get the result set

        try (ResultSet resultSet = statement.executeQuery(sql)) {

          // Process and display the results

          while (resultSet.next()) {

            int id = resultSet.getInt("id");

            String firstName = resultSet.getString("first_name");
```

```
                String lastName = resultSet.getString("last_name");
                int age = resultSet.getInt("age");
                System.out.println("Employee ID: " + id);
                System.out.println("First Name: " + firstName);
                System.out.println("Last Name: " + lastName);
                System.out.println("Age: " + age);
                System.out.println("---------------");    }
              }   }
        } catch (SQLException e) {
            e.printStackTrace();  }
        }
    }
```

Output –

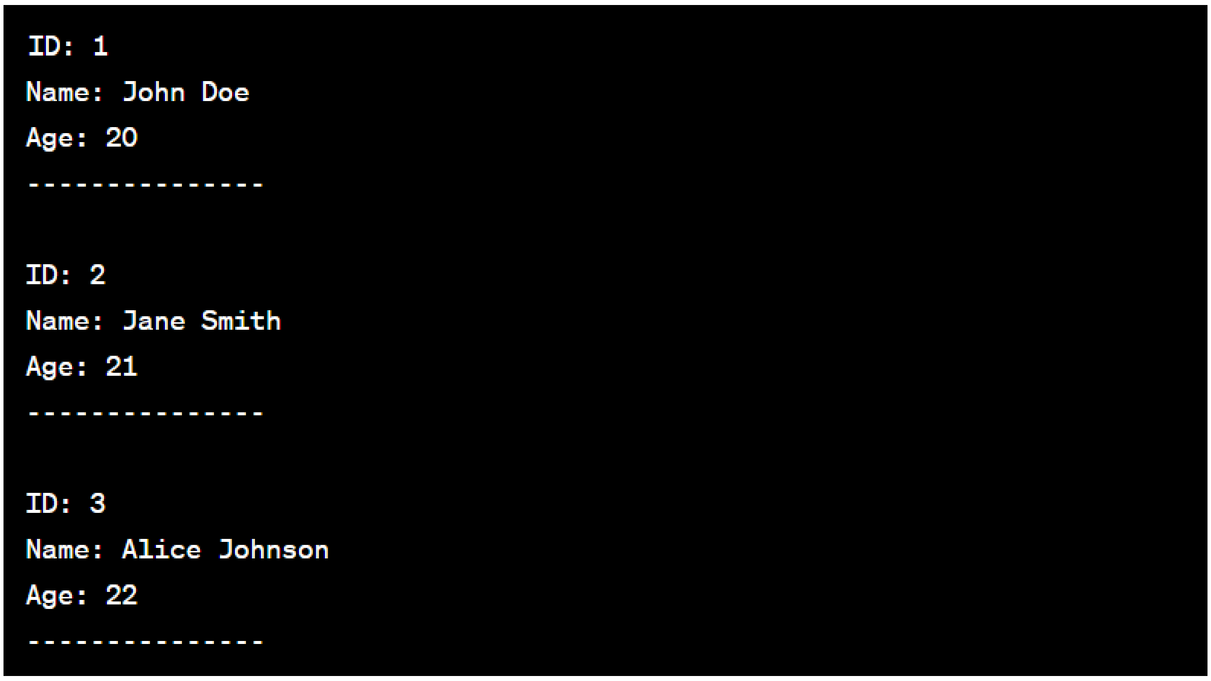


```
Employee ID: 1
First Name: John
Last Name: Doe
Age: 30
---------------

Employee ID: 2
First Name: Jane
Last Name: Smith
Age: 28
---------------

Employee ID: 3
First Name: Bob
Last Name: Johnson
Age: 35
---------------
```

9. Implement java program Query data from MYSQL using JDBC with simple SQL statement.

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;


public class JDBCQueryExample {

    public static void main(String[] args) {

        // Database connection parameters

        String url = "jdbc:mysql://localhost:3306/mydatabase"; // Change to your
database URL

        String username = "your_username";

        String password = "your_password";

        // JDBC connection

        try (Connection connection = DriverManager.getConnection(url,
username, password)) {

            // Create a Statement object

            try (Statement statement = connection.createStatement()) {

                // SQL query

                String sql = "SELECT * FROM students";


                // Execute the query and get the result set

                try (ResultSet resultSet = statement.executeQuery(sql)) {

                    // Process and display the results

                    while (resultSet.next()) {
                        int id = resultSet.getInt("id");
                        String name = resultSet.getString("name");
                        int age = resultSet.getInt("age");
                        System.out.println("ID: " + id);
                        System.out.println("Name: " + name);
                        System.out.println("Age: " + age);
                        System.out.println("---------------");
                    }
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Output –



```
ID: 1
Name: John Doe
Age: 20
---------------

ID: 2
Name: Jane Smith
Age: 21
---------------

ID: 3
Name: Alice Johnson
Age: 22
---------------
```

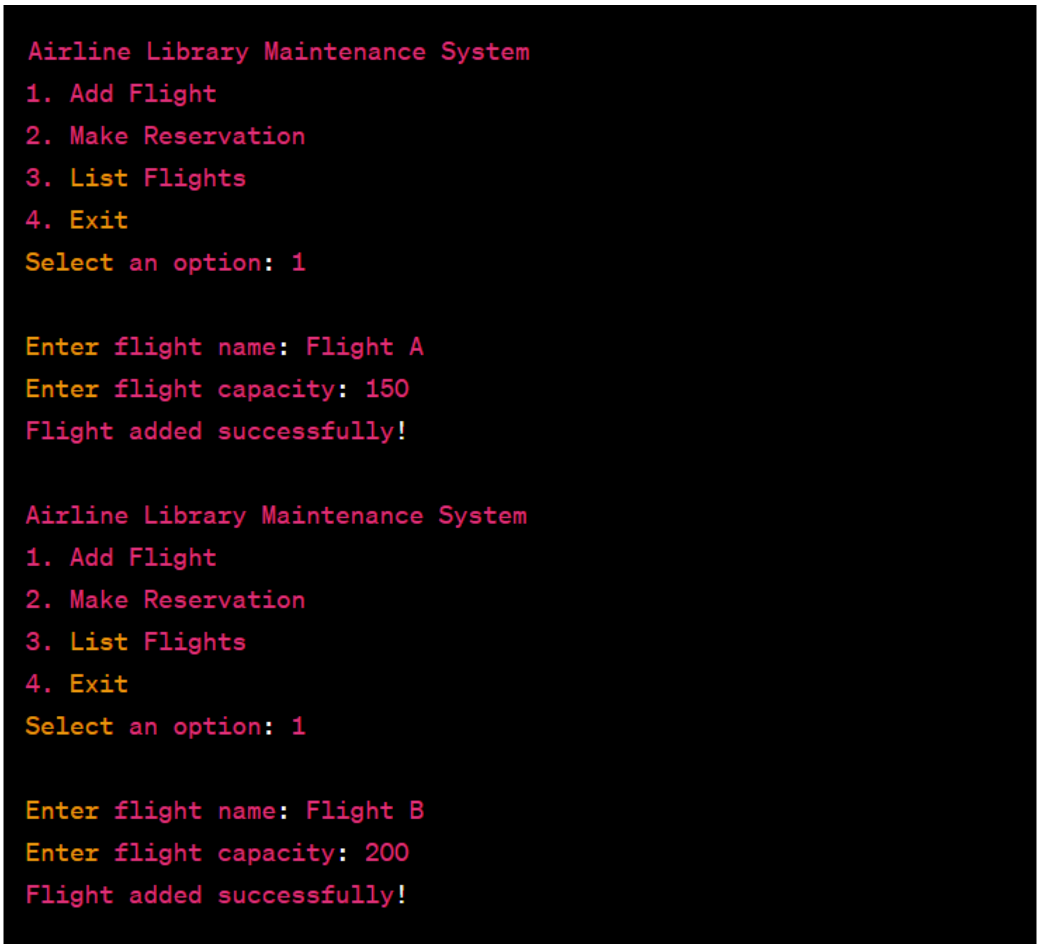
10. Implementation of airline Library maintenance system using JDBC.

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Scanner;


public class AirlineLibraryMaintenanceSystem {

   public static void main(String[] args) {

       try (Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/airlines_library",
"your_username", "your_password")) {

           createTables(connection);


           Scanner scanner = new Scanner(System.in);

           while (true) {

               System.out.println("Airline Library Maintenance System");

               System.out.println("1. Add Flight");

               System.out.println("2. Make Reservation");

               System.out.println("3. List Flights");

               System.out.println("4. Exit");

               System.out.print("Select an option: ");

               int choice = scanner.nextInt();

               scanner.nextLine();
```

```
                switch (choice) {
                    case 1:
                        addFlight(connection, scanner);
                        break;
                    case 2:
                        makeReservation(connection, scanner);
                        break;
                    case 3:
                        listFlights(connection);
                        break;
                    case 4:
                        System.out.println("Exiting the system.");
                        return;
                    default:
                        System.out.println("Invalid option. Please try again.");
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private static void createTables(Connection connection) throws SQLException
{
        String createFlightsTableSQL = "CREATE TABLE IF NOT EXISTS flights (id INT
AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255), capacity INT)";
```

```
        String createReservationsTableSQL = "CREATE TABLE IF NOT EXISTS
reservations (id INT AUTO_INCREMENT PRIMARY KEY, flight_id INT,
passenger_name VARCHAR(255))";

        try (PreparedStatement createFlightsTable =
connection.prepareStatement(createFlightsTableSQL);
             PreparedStatement createReservationsTable =
connection.prepareStatement(createReservationsTableSQL)) {
            createFlightsTable.execute();
            createReservationsTable.execute();
        }
    }

    private static void addFlight(Connection connection, Scanner scanner) throws
SQLException {
        System.out.print("Enter flight name: ");
        String flightName = scanner.nextLine();
        System.out.print("Enter flight capacity: ");
        int capacity = scanner.nextInt();

        String insertFlightSQL = "INSERT INTO flights (name, capacity) VALUES (?,
?)";

        try (PreparedStatement insertFlight =
connection.prepareStatement(insertFlightSQL)) {
            insertFlight.setString(1, flightName);
            insertFlight.setInt(2, capacity);
            insertFlight.executeUpdate();
            System.out.println("Flight added successfully!");
        }
    }

    private static void makeReservation(Connection connection, Scanner
scanner) throws SQLException {
        System.out.print("Enter passenger name: ");
        String passengerName = scanner.nextLine();
        listFlights(connection);
        System.out.print("Enter flight ID to reserve: ");
        int flightId = scanner.nextInt();

        String insertReservationSQL = "INSERT INTO reservations (flight_id,
passenger_name) VALUES (?, ?)";

        try (PreparedStatement insertReservation =
connection.prepareStatement(insertReservationSQL)) {
            insertReservation.setInt(1, flightId);
            insertReservation.setString(2, passengerName);
            insertReservation.executeUpdate();
            System.out.println("Reservation made successfully!");
        }
    }

    private static void listFlights(Connection connection) throws SQLException {
        String listFlightsSQL = "SELECT id, name FROM flights";
        try (PreparedStatement listFlights =
connection.prepareStatement(listFlightsSQL);
             ResultSet resultSet = listFlights.executeQuery()) {
            System.out.println("Available Flights:");
            while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String name = resultSet.getString("name");
                System.out.println("ID: " + id + ", Name: " + name);
            }
        }
    }
}
```

Output –



```
Airline Library Maintenance System
1. Add Flight
2. Make Reservation
3. List Flights
4. Exit
Select an option: 1

Enter flight name: Flight A
Enter flight capacity: 150
Flight added successfully!

Airline Library Maintenance System
1. Add Flight
2. Make Reservation
3. List Flights
4. Exit
Select an option: 1

Enter flight name: Flight B
Enter flight capacity: 200
Flight added successfully!
```