ASSIGNMENT

WEEK 8

Q1) Write a simple application program to establish JDBC connection.

CODE:

```
import java.sql.*;

public class Main {
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection connection = DriverManager.getConnection(=
"jdbc:mysql://localhost:3306/your_database_name",”root”,”root”);

            // Check if the connection is successful
            if (connection != null) {
                System.out.println("Connected to the database!");
                connection.close(); // Close the connection when done
            } else {
                System.out.println("Failed to make connection!");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
```

```
    }
}
```

Q2) Implementation of airline reservation system using JDBC.

CODE:

```
import java.sql.*;
import java.util.Scanner;

public class AirlineLibrarySystem {
    static final String JDBC_URL = "jdbc:mysql://localhost:3306/airline_library";
    static final String DB_USER = "your_username";
    static final String DB_PASSWORD = "your_password";

    public static void main(String[] args) {
        try {
            Connection connection = DriverManager.getConnection(JDBC_URL,
DB_USER, DB_PASSWORD);

            if (connection != null) {
                System.out.println("Connected to the database!");

                Scanner scanner = new Scanner(System.in);

                while (true) {
                    System.out.println("\n1. Add Airline");
                    System.out.println("2. Update Airline");
                    System.out.println("3. Delete Airline");
```

```
            System.out.println("4. List All Airlines");
            System.out.println("5. Quit");
            System.out.print("Enter your choice: ");

            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    addAirline(connection, scanner);
                    break;
                case 2:
                    updateAirline(connection, scanner);
                    break;
                case 3:
                    deleteAirline(connection, scanner);
                    break;
                case 4:
                    listAirlines(connection);
                    break;
                case 5:
                    System.out.println("Exiting...");
                    connection.close();
                    return;
                default:
                    System.out.println("Invalid choice! Try again.");
            }
        }
    }
```

```
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

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

        System.out.print("Enter country: ");
        String country = scanner.nextLine();

        System.out.print("Enter established year: ");
        int establishedYear = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        String sql = "INSERT INTO airlines (name, country, established_year)
VALUES (?, ?, ?)";

        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setString(1, name);
            statement.setString(2, country);
            statement.setInt(3, establishedYear);

            int rowsAffected = statement.executeUpdate();

            if (rowsAffected > 0) {
                System.out.println("Airline added successfully!");
            } else {
```

```
            System.out.println("Failed to add airline.");
        }
    }
}

private static void updateAirline(Connection connection, Scanner scanner)
throws SQLException {
    System.out.print("Enter airline ID to update: ");
    int id = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    System.out.print("Enter new airline name: ");
    String name = scanner.nextLine();

    System.out.print("Enter new country: ");
    String country = scanner.nextLine();

    System.out.print("Enter new established year: ");
    int establishedYear = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    String sql = "UPDATE airlines SET name = ?, country = ?, established_year
= ? WHERE id = ?";

    try (PreparedStatement statement = connection.prepareStatement(sql)) {
        statement.setString(1, name);
        statement.setString(2, country);
        statement.setInt(3, establishedYear);
        statement.setInt(4, id);
```

```
            int rowsAffected = statement.executeUpdate();

            if (rowsAffected > 0) {
                System.out.println("Airline updated successfully!");
            } else {
                System.out.println("Failed to update airline.");
            }
        }
    }

    private static void deleteAirline(Connection connection, Scanner scanner)
throws SQLException {
        System.out.print("Enter airline ID to delete: ");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        String sql = "DELETE FROM airlines WHERE id = ?";

        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setInt(1, id);

            int rowsAffected = statement.executeUpdate();

            if (rowsAffected > 0) {
                System.out.println("Airline deleted successfully!");
            } else {
                System.out.println("Failed to delete airline. Check the ID.");
            }
        }
```

```
    }

    private static void listAirlines(Connection connection) throws SQLException
{
        String sql = "SELECT * FROM airlines";

        try (Statement statement = connection.createStatement();
             ResultSet resultSet = statement.executeQuery(sql)) {

            while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String name = resultSet.getString("name");
                String country = resultSet.getString("country");
                int establishedYear = resultSet.getInt("established_year");

                System.out.println("ID: " + id + ", Name: " + name + ", Country: " +
country + ", Established Year: " + establishedYear);
            }
        }
    }
}
```


Q3) Write a JDBC program to retrieve the student details from the database.

CODE:

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

```
public class StudentDetailsRetriever {
    static final String JDBC_URL =
"jdbc:mysql://localhost:3306/student_database";
    static final String DB_USER = "your_username";
    static final String DB_PASSWORD = "your_password";

    public static void main(String[] args) {
        try {
            Connection connection = DriverManager.getConnection(JDBC_URL,
DB_USER, DB_PASSWORD);

            if (connection != null) {
                System.out.println("Connected to the database!");

                // Retrieve student details
                retrieveStudentDetails(connection);

                connection.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private static void retrieveStudentDetails(Connection connection) throws
SQLException {
        String sql = "SELECT * FROM students";

        try (Statement statement = connection.createStatement();
             ResultSet resultSet = statement.executeQuery(sql)) {
```

```
            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("ID: " + id + ", Name: " + name + ", Age: " + age +
", Grade: " + grade);
            }
        }
    }
}
```


Q4) Implement java program to retrieve contents of a table using JDBC connection.

CODE:

```
import java.sql.*;

public class TableContentRetriever {
    static final String JDBC_URL = "jdbc:mysql://localhost:3306/your_database_name";
    static final String DB_USER = "your_username";
    static final String DB_PASSWORD = "your_password";

    public static void main(String[] args) {
        try {
            Connection connection = DriverManager.getConnection(JDBC_URL, DB_USER,
DB_PASSWORD);

            if (connection != null) {
```

```
            System.out.println("Connected to the database!");

            // Retrieve table contents
            retrieveTableContents(connection);

            connection.close();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

private static void retrieveTableContents(Connection connection) throws SQLException {
    String tableName = "your_table_name"; // Replace with the actual table name
    String sql = "SELECT * FROM " + tableName;

    try (Statement statement = connection.createStatement();
         ResultSet resultSet = statement.executeQuery(sql)) {

        ResultSetMetaData metaData = resultSet.getMetaData();
        int columnCount = metaData.getColumnCount();

        // Print column names
        for (int i = 1; i <= columnCount; i++) {
            System.out.print(metaData.getColumnName(i) + "\t");
        }
        System.out.println();

        // Print table contents
        while (resultSet.next()) {
            for (int i = 1; i <= columnCount; i++) {
                System.out.print(resultSet.getString(i) + "\t");
```

```
            }
            System.out.println();
        }
    }
  }
}
```

Q5) Write JDBC program to insert records to a table using JDBC connection.

CODE

```
import java.sql.*;

public class InsertRecord {
    static final String JDBC_URL = "jdbc:mysql://localhost:3306/your_database_name";
    static final String DB_USER = "your_username";
    static final String DB_PASSWORD = "your_password";

    public static void main(String[] args) {
        try {
            Connection connection = DriverManager.getConnection(JDBC_URL,
DB_USER, DB_PASSWORD);

            if (connection != null) {
                System.out.println("Connected to the database!");

                // Insert records
                insertRecord(connection, "John Doe", 25, "john@example.com");
                insertRecord(connection, "Jane Doe", 23, "jane@example.com");

                connection.close();
            }
```

```
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private static void insertRecord(Connection connection, String name, int age, String
email) throws SQLException {
        String sql = "INSERT INTO your_table_name (name, age, email) VALUES (?, ?,
?)";

        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setString(1, name);
            statement.setInt(2, age);
            statement.setString(3, email);

            int rowsAffected = statement.executeUpdate();

            if (rowsAffected > 0) {
                System.out.println("Record inserted successfully!");
            } else {
                System.out.println("Failed to insert record.");
            }
        }
    }
}
```


Q6) Write JDBC program to update contents of a library management system using JDBC connection.

CODE

```
import java.sql.*;

public class PreparedStatementExample {
```

```
    static final String JDBC_URL = "jdbc:mysql://localhost:3306/your_database_name";
    static final String DB_USER = "your_username";
    static final String DB_PASSWORD = "your_password";

    public static void main(String[] args) {
        try {
            Connection connection = DriverManager.getConnection(JDBC_URL,
DB_USER, DB_PASSWORD);

            if (connection != null) {
                System.out.println("Connected to the database!");

                // Define the query with a parameter
                String query = "SELECT * FROM your_table_name WHERE id = ?";

                // Prepare the statement
                PreparedStatement preparedStatement =
connection.prepareStatement(query);

                // Set the parameter value
                int targetId = 1; // Example: You want to retrieve the record with ID 1
                preparedStatement.setInt(1, targetId);

                // Execute the query
                ResultSet resultSet = preparedStatement.executeQuery();

                // Process the result set
                while (resultSet.next()) {
                    int id = resultSet.getInt("id");
                    String name = resultSet.getString("name");

                    System.out.println("ID: " + id + ", Name: " + name);
                }
```

```
            // Close resources
            resultSet.close();
            preparedStatement.close();
            connection.close();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
  }
}
```

Q7) Write a simple application program to establish JDBC query execution using Prepared Statement.

CODE:

```
import java.sql.*;

public class PreparedStatementExample {
   static final String JDBC_URL =
"jdbc:mysql://localhost:3306/your_database_name";
   static final String DB_USER = "your_username";
   static final String DB_PASSWORD = "your_password";

   public static void main(String[] args) {
      try {
         Connection connection =
DriverManager.getConnection(JDBC_URL, DB_USER,
DB_PASSWORD);

         if (connection != null) {
            System.out.println("Connected to the database!");

            // Define the query with a parameter
```

```
            String query = "SELECT * FROM your_table_name WHERE id
= ?";

            // Prepare the statement
            PreparedStatement preparedStatement =
connection.prepareStatement(query);

            // Set the parameter value
            int targetId = 1; // Example: You want to retrieve the record with
ID 1
            preparedStatement.setInt(1, targetId);

            // Execute the query
            ResultSet resultSet = preparedStatement.executeQuery();

            // Process the result set
            while (resultSet.next()) {
               int id = resultSet.getInt("id");
               String name = resultSet.getString("name");

               System.out.println("ID: " + id + ", Name: " + name);
            }

            // Close resources
            resultSet.close();
            preparedStatement.close();
            connection.close();
         }
      } catch (SQLException e) {
         e.printStackTrace();
      }
```

```
    }
}
```

Q8) Write a simple application program to establish JDBC query execution using ResultSet executeQurey.

CODE:

```
import java.sql.*;

public class ResultSetExample {
    static final String JDBC_URL =
"jdbc:mysql://localhost:3306/your_database_name";
    static final String DB_USER = "your_username";
    static final String DB_PASSWORD = "your_password";

    public static void main(String[] args) {
        try {
            Connection connection =
DriverManager.getConnection(JDBC_URL, DB_USER,
DB_PASSWORD);

            if (connection != null) {
                System.out.println("Connected to the database!");

                // Define the query
                String query = "SELECT * FROM your_table_name";

                // Create a Statement object
                Statement statement = connection.createStatement();

                // Execute the query and get the ResultSet
```

```
            ResultSet resultSet = statement.executeQuery(query);

            // Process the result set
            while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String name = resultSet.getString("name");

                System.out.println("ID: " + id + ", Name: " + name);
            }

            // Close resources
            resultSet.close();
            statement.close();
            connection.close();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
  }
}
```


Q9) Implement java program Query data from MYSQL using JDBC with simple SQL statement.

CODE:

```
import java.sql.*;

public class SimpleQueryExample {
    static final String JDBC_URL =
"jdbc:mysql://localhost:3306/your_database_name";
```

```
    static final String DB_USER = "your_username";
    static final String DB_PASSWORD = "your_password";

    public static void main(String[] args) {
        try {
            Connection connection =
DriverManager.getConnection(JDBC_URL, DB_USER,
DB_PASSWORD);

            if (connection != null) {
                System.out.println("Connected to the database!");

                // Define the query
                String query = "SELECT * FROM your_table_name";

                // Create a Statement object
                Statement statement = connection.createStatement();

                // Execute the query and get the ResultSet
                ResultSet resultSet = statement.executeQuery(query);

                // Process the result set
                while (resultSet.next()) {
                    int id = resultSet.getInt("id");
                    String name = resultSet.getString("name");

                    System.out.println("ID: " + id + ", Name: " + name);
                }

                // Close resources
                resultSet.close();
```

```
            statement.close();
            connection.close();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
  }
}
```

Q10) Implementation of airline Library maintenance system using JDBC.

CODE:

```
import java.sql.*;
import java.util.Scanner;

public class AirlineLibrarySystem {
  static final String JDBC_URL =
"jdbc:mysql://localhost:3306/airline_library";
  static final String DB_USER = "your_username";
  static final String DB_PASSWORD = "your_password";

  public static void main(String[] args) {
    try {
      Connection connection =
DriverManager.getConnection(JDBC_URL, DB_USER,
DB_PASSWORD);

      if (connection != null) {
        System.out.println("Connected to the database!");
```

```
Scanner scanner = new Scanner(System.in);

while (true) {
    System.out.println("\n1. Add Airline");
    System.out.println("2. Update Airline");
    System.out.println("3. Delete Airline");
    System.out.println("4. List All Airlines");
    System.out.println("5. Quit");
    System.out.print("Enter your choice: ");

    int choice = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    switch (choice) {
        case 1:
            addAirline(connection, scanner);
            break;
        case 2:
            updateAirline(connection, scanner);
            break;
        case 3:
            deleteAirline(connection, scanner);
            break;
        case 4:
            listAirlines(connection);
            break;
        case 5:
            System.out.println("Exiting...");
            connection.close();
            return;
```

```
                default:
                    System.out.println("Invalid choice! Try again.");
            }
        }
    }
} catch (SQLException e) {
    e.printStackTrace();
}
}

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

    System.out.print("Enter country: ");
    String country = scanner.nextLine();

    System.out.print("Enter established year: ");
    int establishedYear = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    String sql = "INSERT INTO airlines (name, country,
established_year) VALUES (?, ?, ?)";

    try (PreparedStatement statement =
connection.prepareStatement(sql)) {
        statement.setString(1, name);
        statement.setString(2, country);
        statement.setInt(3, establishedYear);
```

```
            int rowsAffected = statement.executeUpdate();

            if (rowsAffected > 0) {
                System.out.println("Airline added successfully!");
            } else {
                System.out.println("Failed to add airline.");
            }
        }
    }

    private static void updateAirline(Connection connection, Scanner
scanner) throws SQLException {
        System.out.print("Enter airline ID to update: ");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        System.out.print("Enter new airline name: ");
        String name = scanner.nextLine();

        System.out.print("Enter new country: ");
        String country = scanner.nextLine();

        System.out.print("Enter new established year: ");
        int establishedYear = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        String sql = "UPDATE airlines SET name = ?, country = ?,
established_year = ? WHERE id = ?";

        try (PreparedStatement statement =
connection.prepareStatement(sql)) {
```

```
            statement.setString(1, name);
            statement.setString(2, country);
            statement.setInt(3, establishedYear);
            statement.setInt(4, id);

            int rowsAffected = statement.executeUpdate();

            if (rowsAffected > 0) {
                System.out.println("Airline updated successfully!");
            } else {
                System.out.println("Failed to update airline.");
            }
        }
    }

    private static void deleteAirline(Connection connection, Scanner
scanner) throws SQLException {
        System.out.print("Enter airline ID to delete: ");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        String sql = "DELETE FROM airlines WHERE id = ?";

        try (PreparedStatement statement =
connection.prepareStatement(sql)) {
            statement.setInt(1, id);

            int rowsAffected = statement.executeUpdate();

            if (rowsAffected > 0) {
                System.out.println("Airline deleted successfully!");
```

```
            } else {
                System.out.println("Failed to delete airline. Check the ID.");
            }
        }
    }

    private static void listAirlines(Connection connection) throws
SQLException {
        String sql = "SELECT * FROM airlines";

        try (Statement statement = connection.createStatement();
             ResultSet resultSet = statement.executeQuery(sql)) {

            while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String name = resultSet.getString("name");
                String country = resultSet.getString("country");
                int establishedYear = resultSet.getInt("established_year");

                System.out.println("ID: " + id + ", Name: " + name + ", Country:
" + country + ", Established Year: " + establishedYear);
            }
        }
    }
}
```