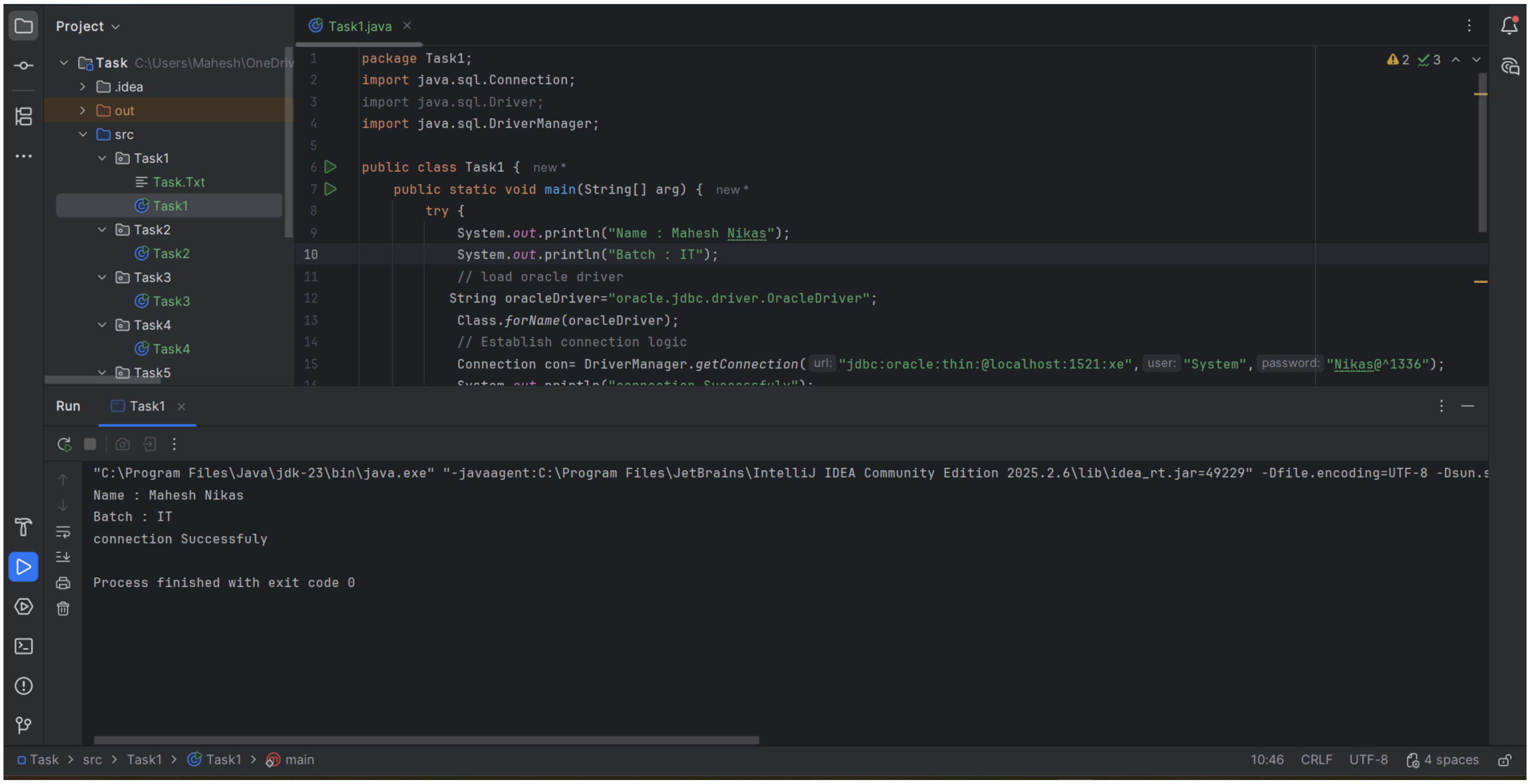


```java
package Task1;
import java.sql.Connection;
import java.sql.Driver;
import java.sql.DriverManager;

public class Task1 {  new *
    public static void main(String[] arg) {  new *
        try {
            System.out.println("Name : Mahesh Nikas");
            System.out.println("Batch : IT");
            // load oracle driver
           String oracleDriver="oracle.jdbc.driver.OracleDriver";
            Class.forName(oracleDriver);
            // Establish connection logic
            Connection con= DriverManager.getConnection( url: "jdbc:oracle:thin:@localhost:1521:xe", user: "System", password: "Nikas@^1336");
            System.out.println("connection Successfuly");
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.6\lib\idea_rt.jar=49229" -Dfile.encoding=UTF-8 -Dsun.s
Name : Mahesh Nikas
Batch : IT
connection Successfuly

Process finished with exit code 0
```

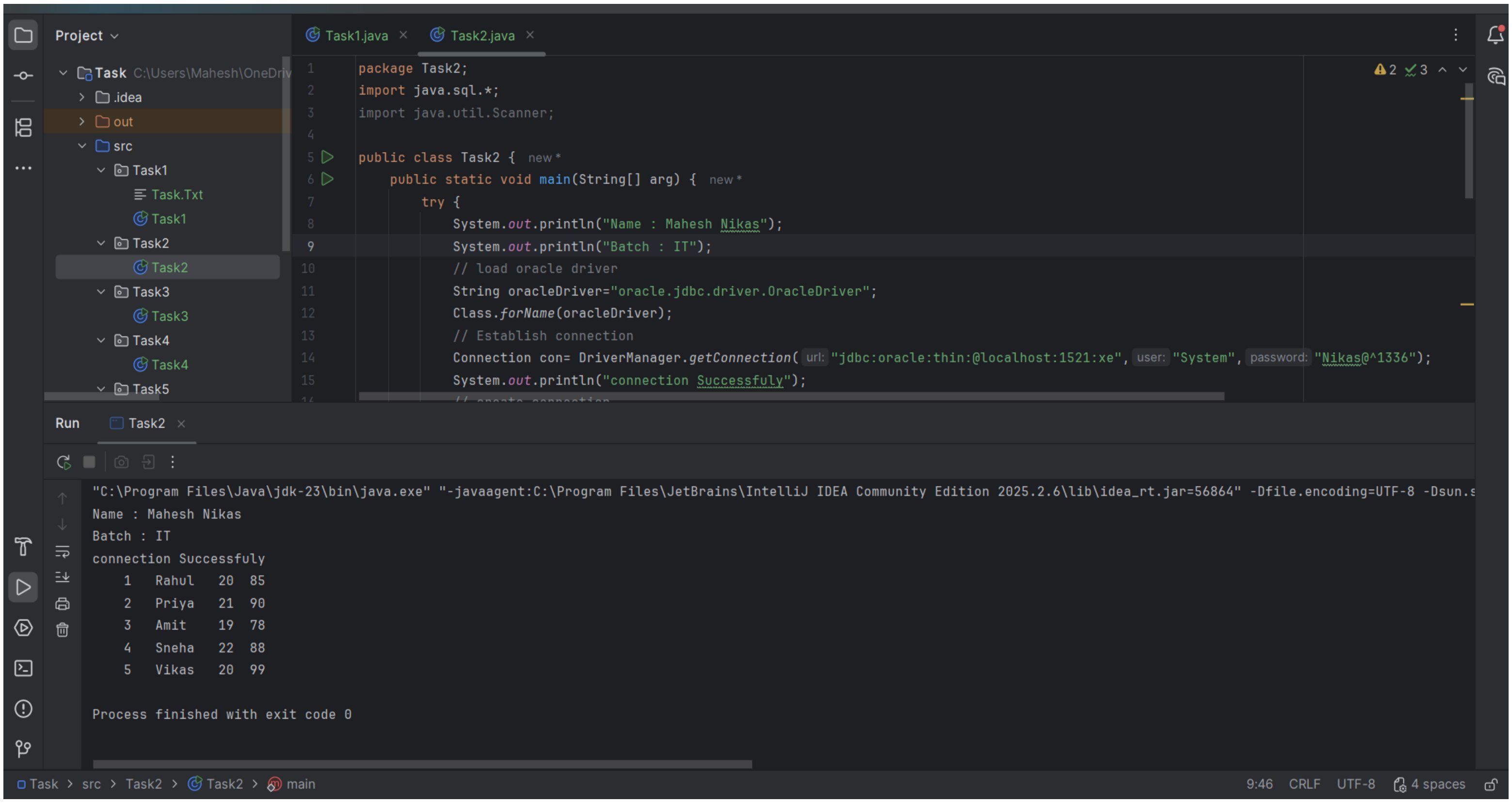
Task 1 Explanation:

- Loads Oracle JDBC Driver using Class.forName().

- Establishes connection with Oracle database using DriverManager.getConnection().

- Verifies database connectivity.

Sample Input Used: No user input



```java
package Task2;
import java.sql.*;
import java.util.Scanner;

public class Task2 {  new *
    public static void main(String[] arg) {  new *
        try {
            System.out.println("Name : Mahesh Nikas");
            System.out.println("Batch : IT");
            // load oracle driver
            String oracleDriver="oracle.jdbc.driver.OracleDriver";
            Class.forName(oracleDriver);
            // Establish connection
            Connection con= DriverManager.getConnection( url: "jdbc:oracle:thin:@localhost:1521:xe", user: "System", password: "Nikas@^1336");
            System.out.println("connection Successfuly");
            // create connection
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.6\lib\idea_rt.jar=56864" -Dfile.encoding=UTF-8 -Dsun.s
Name : Mahesh Nikas
Batch : IT
connection Successfuly
    1   Rahul   20  85
    2   Priya   21  90
    3   Amit    19  78
    4   Sneha   22  88
    5   Vikas   20  99

Process finished with exit code 0
```

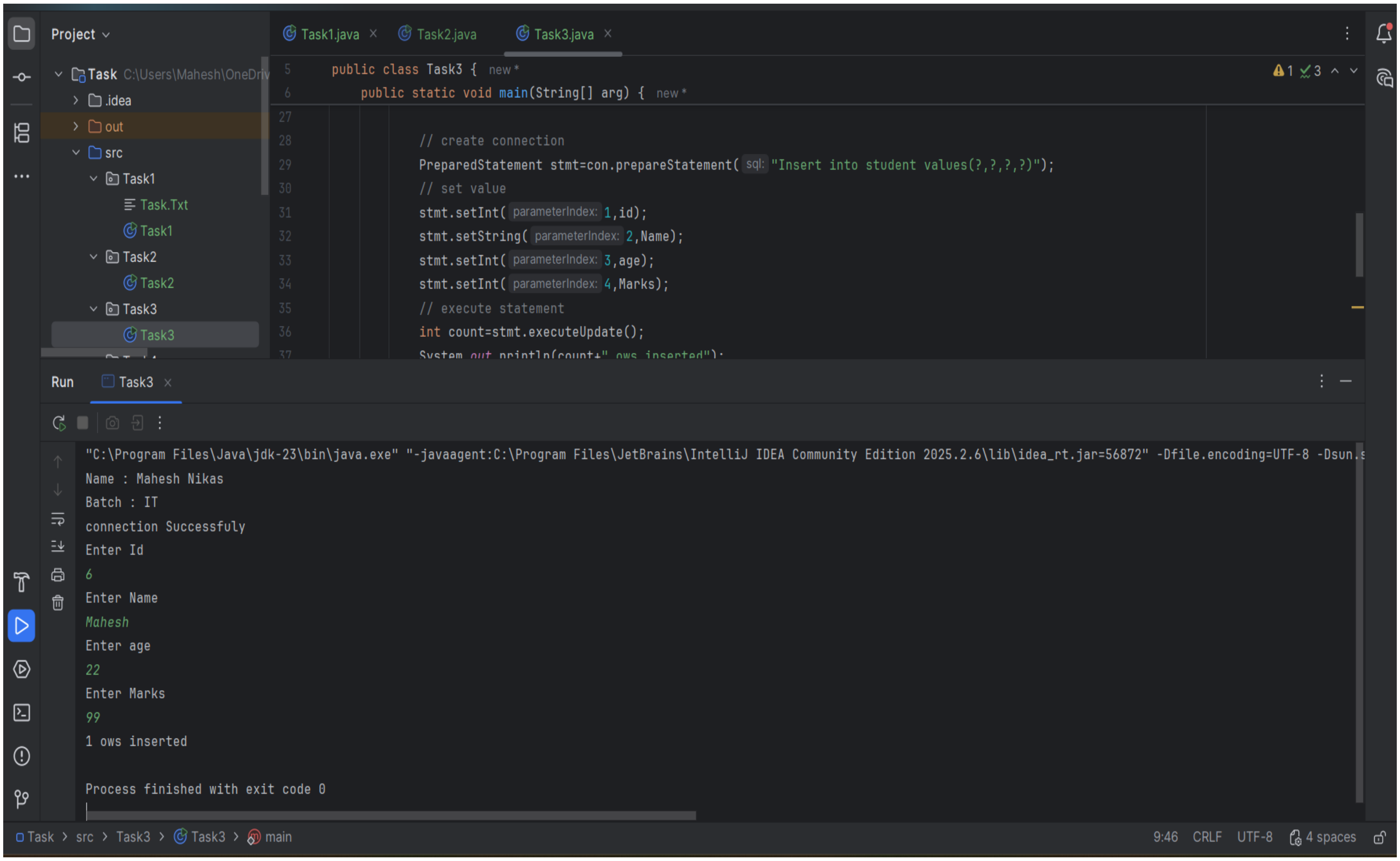
2 Task Explanation:

- Creates JDBC connection.

- Executes SELECT query using Statement.

- Uses ResultSet to fetch and display all student records.

Sample Input Used: No user input



```java
public class Task3 {  new *
    public static void main(String[] arg) {  new *

            // create connection
            PreparedStatement stmt=con.prepareStatement( sql: "Insert into student values(?,?,?,?)");
            // set value
            stmt.setInt( parameterIndex: 1,id);
            stmt.setString( parameterIndex: 2,Name);
            stmt.setInt( parameterIndex: 3,age);
            stmt.setInt( parameterIndex: 4,Marks);
            // execute statement
            int count=stmt.executeUpdate();
            System.out.println(count+" ows inserted");
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.6\lib\idea_rt.jar=56872" -Dfile.encoding=UTF-8 -Dsun.s
Name : Mahesh Nikas
Batch : IT
connection Successfuly
Enter Id
6
Enter Name
Mahesh
Enter age
22
Enter Marks
99
1 ows inserted

Process finished with exit code 0
```

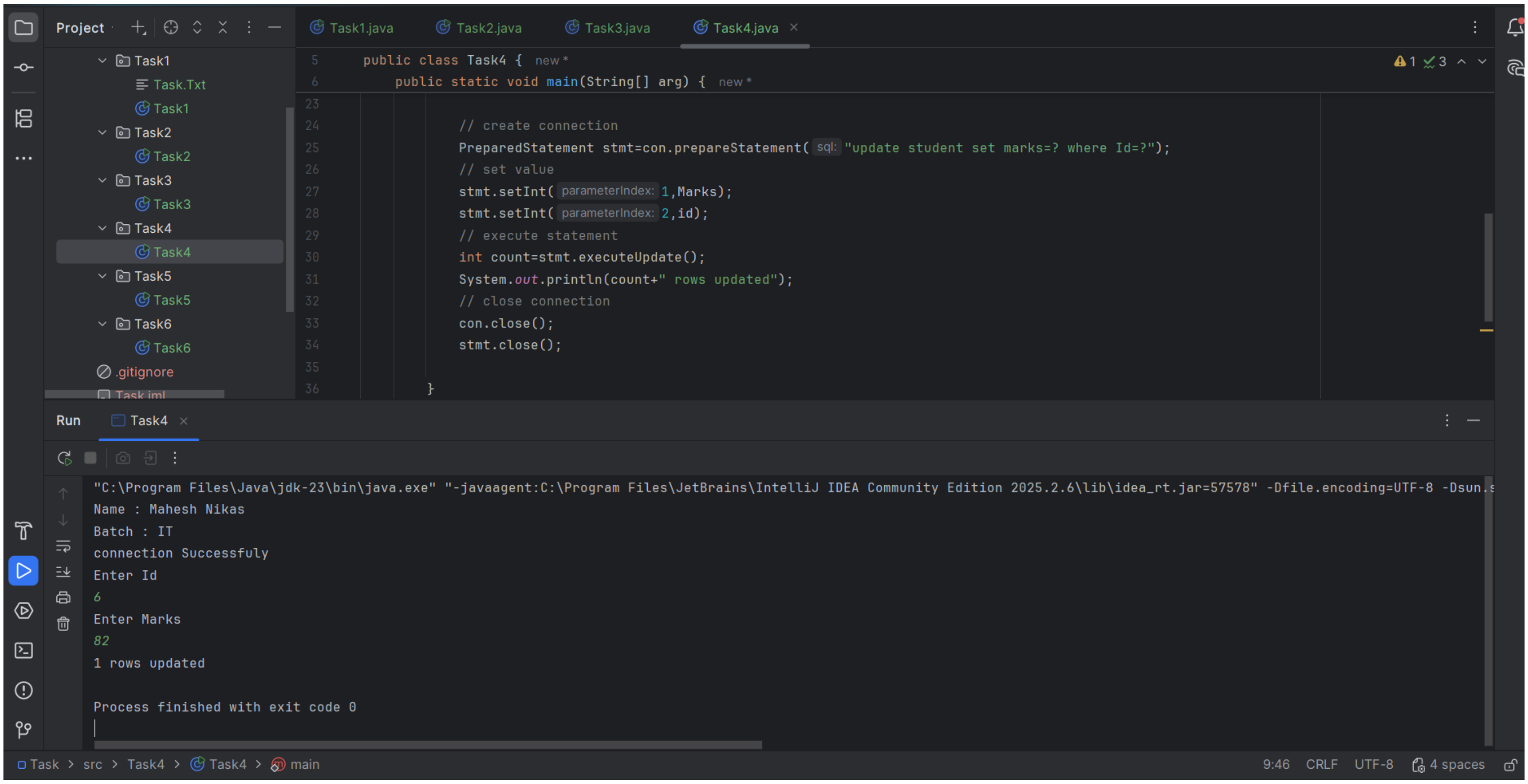
3 Task Explanation:

- Uses PreparedStatement with INSERT query.

- Accepts Id, Name, Age, Marks from user.

- Executes executeUpdate() to insert record.

Sample Input Used: Id=6, Name=Mahesh, Age=22, Marks=99



```
public class Task4 {  new *
    public static void main(String[] arg) {  new *

            // create connection
            PreparedStatement stmt=con.prepareStatement( sql: "update student set marks=? where Id=?");
            // set value
            stmt.setInt( parameterIndex: 1,Marks);
            stmt.setInt( parameterIndex: 2,id);
            // execute statement
            int count=stmt.executeUpdate();
            System.out.println(count+" rows updated");
            // close connection
            con.close();
            stmt.close();

        }
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.6\lib\idea_rt.jar=57578" -Dfile.encoding=UTF-8 -Dsun.s
Name : Mahesh Nikas
Batch : IT
connection Successfuly
Enter Id
6
Enter Marks
82
1 rows updated

Process finished with exit code 0
```

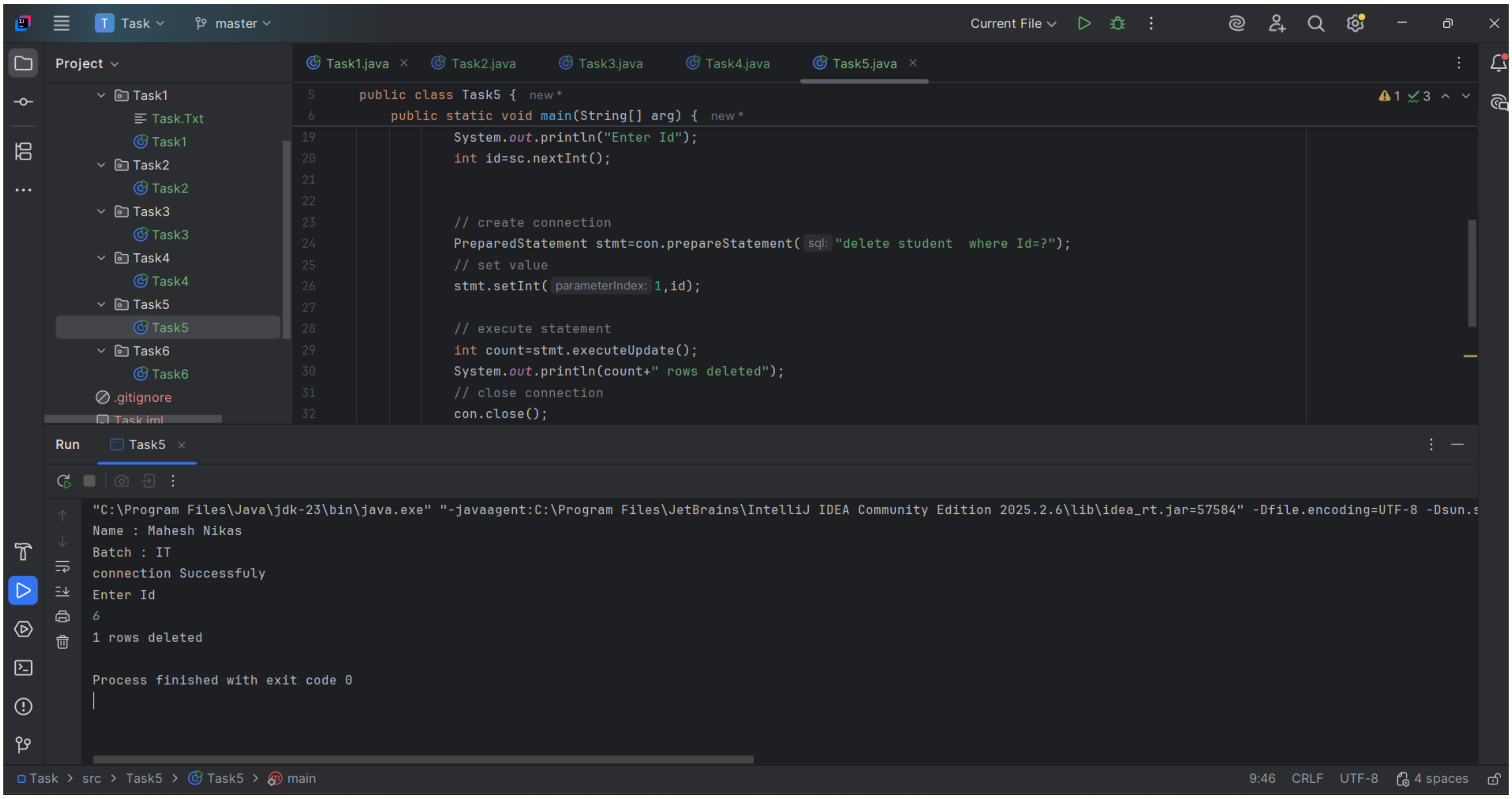
4 Task Explanation:

- Uses PreparedStatement with UPDATE query.

- Accepts Id and new Marks from user.

- Updates record safely using parameters.

Sample Input Used: Id=6, Marks=82



```
public class Task5 {  new *
    public static void main(String[] arg) {  new *
            System.out.println("Enter Id");
            int id=sc.nextInt();


            // create connection
            PreparedStatement stmt=con.prepareStatement( sql: "delete student  where Id=?");
            // set value
            stmt.setInt( parameterIndex: 1,id);

            // execute statement
            int count=stmt.executeUpdate();
            System.out.println(count+" rows deleted");
            // close connection
            con.close();
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.6\lib\idea_rt.jar=57584" -Dfile.encoding=UTF-8 -Dsun.s
Name : Mahesh Nikas
Batch : IT
connection Successfuly
Enter Id
6
1 rows deleted

Process finished with exit code 0
```

5 Task Explanation:

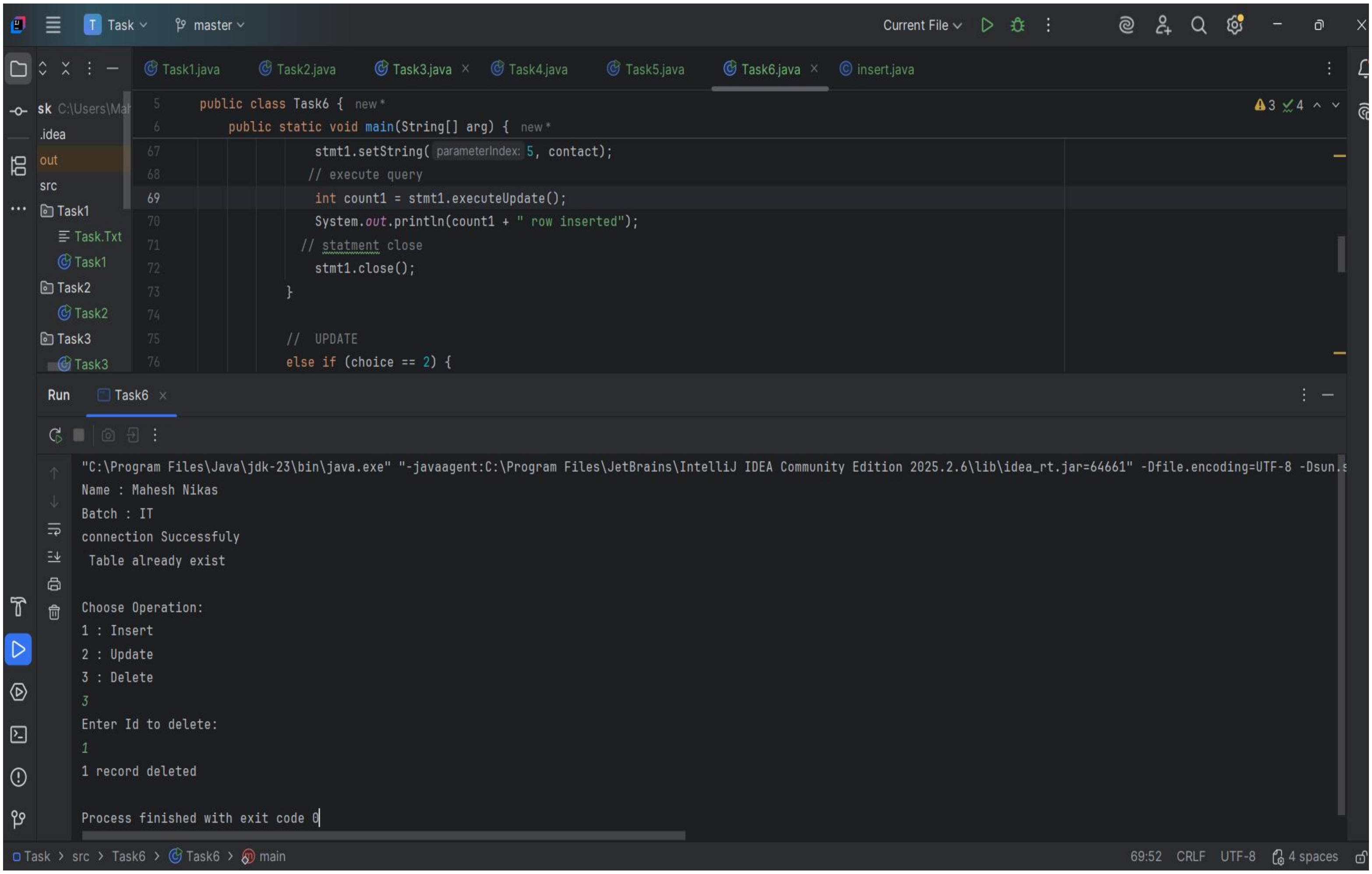
- Uses PreparedStatement with DELETE query.

- Accepts Id from user.

- Deletes student record based on Id.

Sample Input Used: Id=6

Task 6



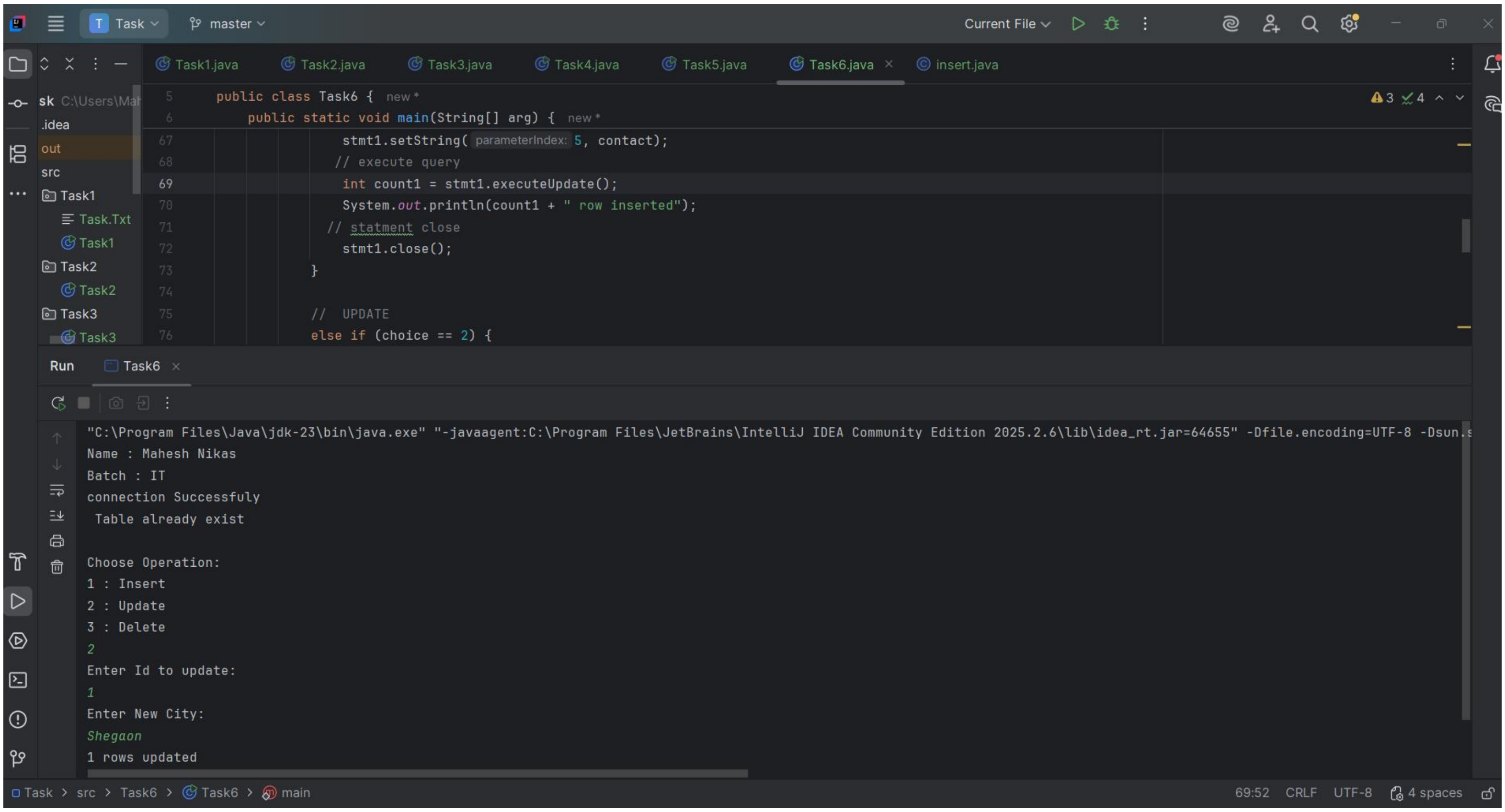
```
public class Task6 {  new *
    public static void main(String[] arg) {  new *
                stmt1.setString( parameterIndex: 5, contact);
               // execute query
                int count1 = stmt1.executeUpdate();
                System.out.println(count1 + " row inserted");
              // statment close
                stmt1.close();
            }

            //  UPDATE
            else if (choice == 2) {
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.6\lib\idea_rt.jar=64661" -Dfile.encoding=UTF-8 -Dsun.s
Name : Mahesh Nikas
Batch : IT
connection Successfuly
 Table already exist

Choose Operation:
1 : Insert
2 : Update
3 : Delete
3
Enter Id to delete:
1
1 record deleted

Process finished with exit code 0
```

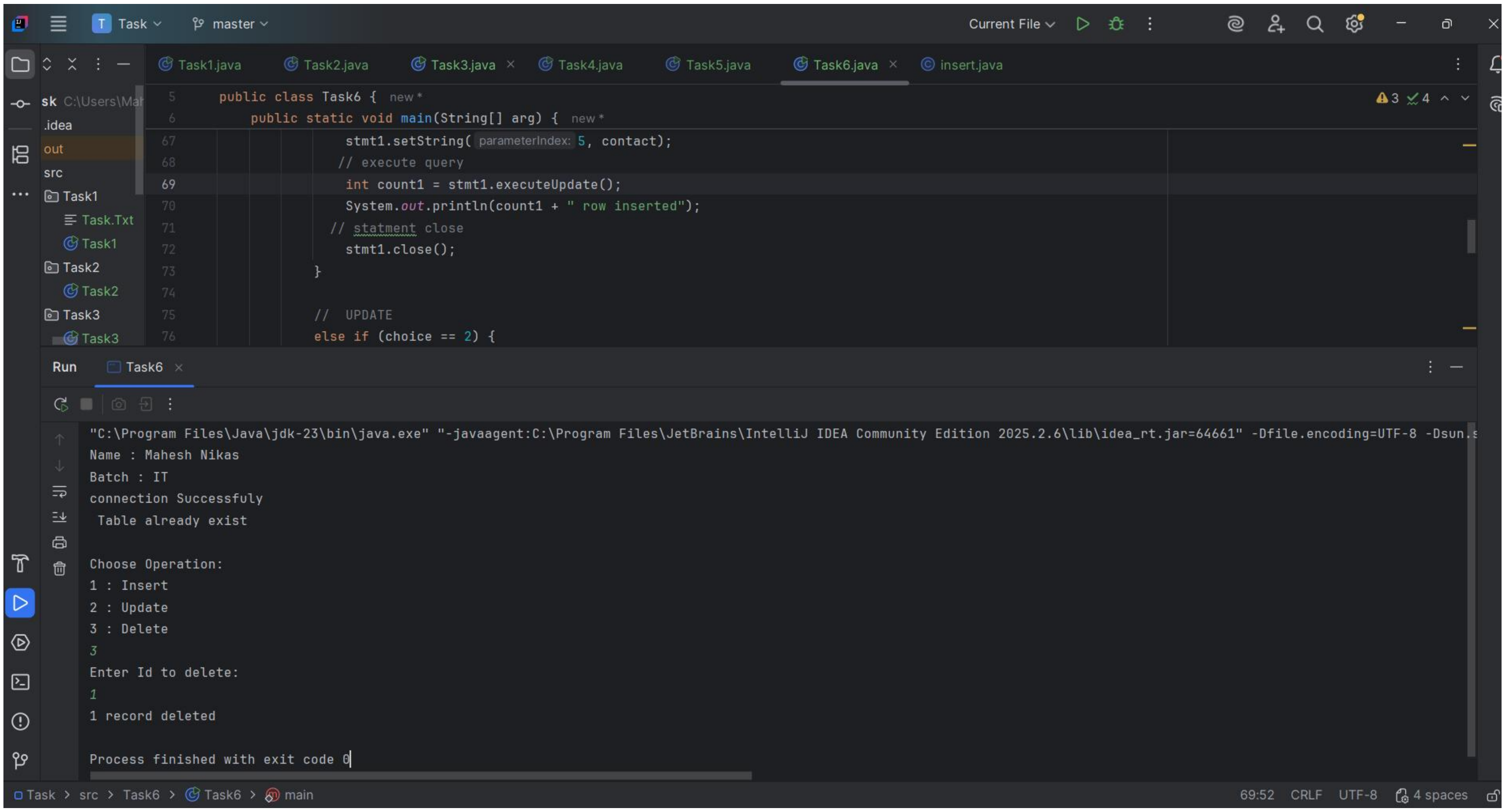


```
public class Task6 {  new *
    public static void main(String[] arg) {  new *
                stmt1.setString( parameterIndex: 5, contact);
               // execute query
                int count1 = stmt1.executeUpdate();
                System.out.println(count1 + " row inserted");
              // statment close
                stmt1.close();
            }

            //  UPDATE
            else if (choice == 2) {
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.6\lib\idea_rt.jar=64655" -Dfile.encoding=UTF-8 -Dsun.s
Name : Mahesh Nikas
Batch : IT
connection Successfuly
 Table already exist

Choose Operation:
1 : Insert
2 : Update
3 : Delete
2
Enter Id to update:
1
Enter New City:
Shegaon
1 rows updated
```



```
public class Task6 {  new *
    public static void main(String[] arg) {  new *
                stmt1.setString( parameterIndex: 5, contact);
               // execute query
                int count1 = stmt1.executeUpdate();
                System.out.println(count1 + " row inserted");
              // statment close
                stmt1.close();
            }

            //  UPDATE
            else if (choice == 2) {
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.6\lib\idea_rt.jar=64661" -Dfile.encoding=UTF-8 -Dsun.s
Name : Mahesh Nikas
Batch : IT
connection Successfuly
 Table already exist

Choose Operation:
1 : Insert
2 : Update
3 : Delete
3
Enter Id to delete:
1
1 record deleted

Process finished with exit code 0
```

Task 6 is a menu JDBC application that performs INSERT, UPDATE, and DELETE operations on the employee table using PreparedStatement. The program also handles table creation logic by checking if the table already exists.

Class Explanation: Task6

The Task6 class contains the main() method. It establishes a database connection, displays a menu to the user, accepts input, and performs database operations based on the selected choice.

Key Methods and Logic

- Class.forName(): Loads the Oracle JDBC driver into memory.

- DriverManager.getConnection(): Establishes connection with Oracle database.

- PreparedStatement: Used for INSERT, UPDATE, and DELETE queries securely.

- executeUpdate(): Executes DML operations and returns affected row count.

- Scanner: Takes user input from console.

- Conditional logic (if-else): Controls menu-driven flow.

Operations Explained

1. Insert Operation

Accepts employee Id, Name, Email, City, and Contact from user. Inserts a new record into employee table using PreparedStatement.

2. Update Operation

Accepts employee Id and new City from user. Updates the city field for the specified employee.

3. Delete Operation

Accepts employee Id from user and deletes the corresponding record from the employee table.

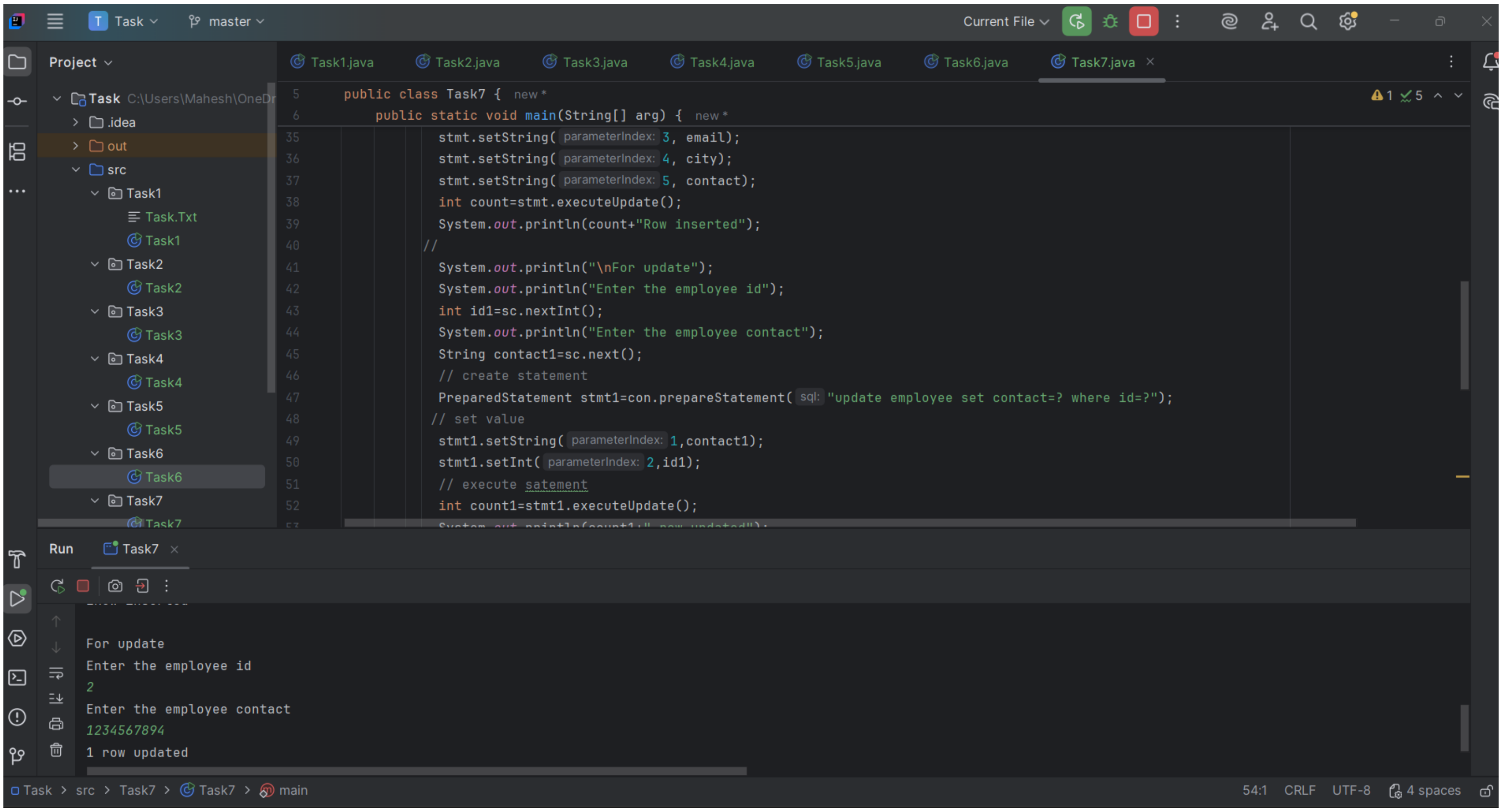
Sample Inputs Used

- Insert → Id=1, Name=Mahesh, Email=mahesh@123gmail.com, City=Mehkar, Contact=9356736650

- Update → Id=1, New City=Shegaon

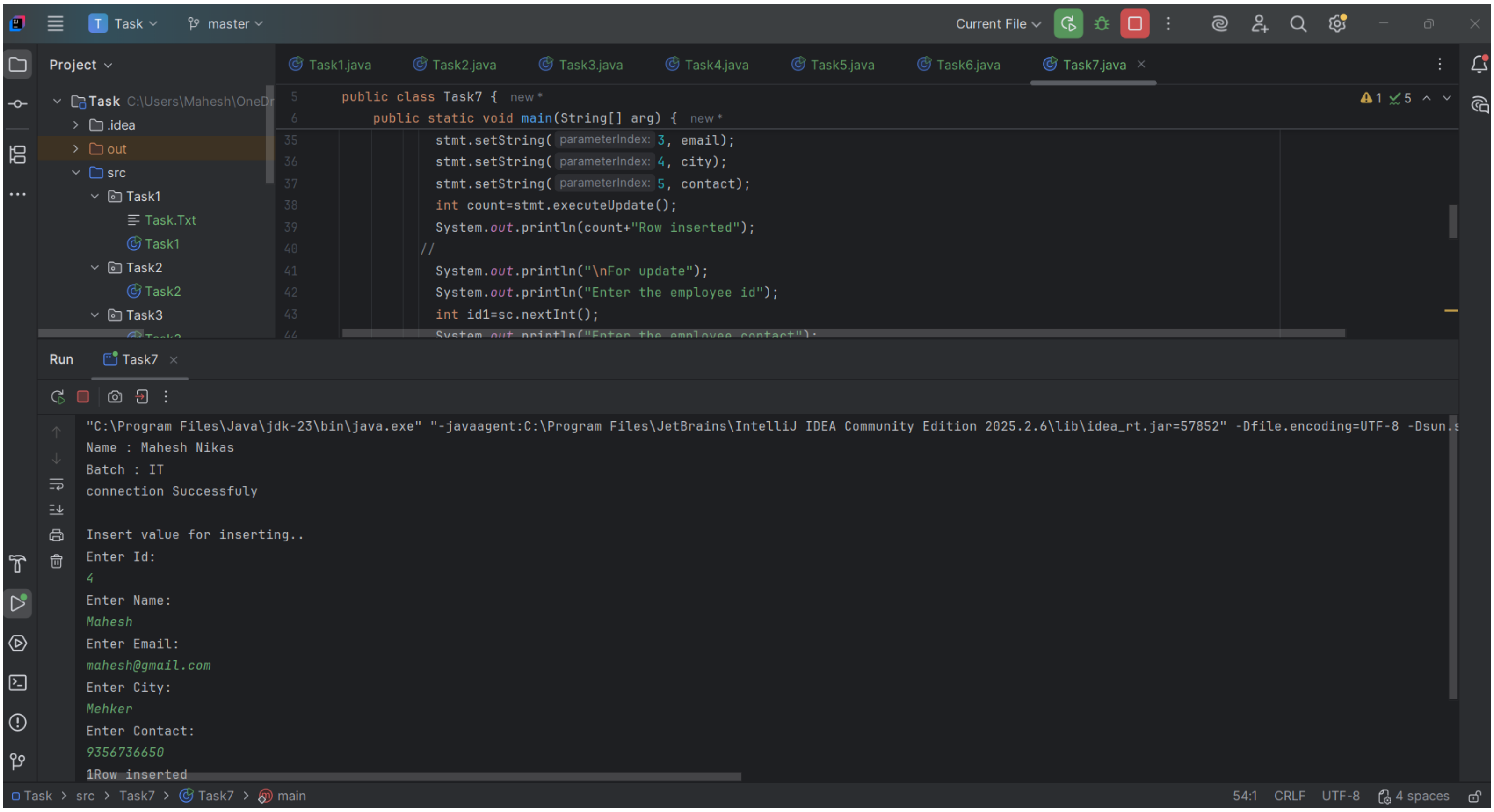
- Delete → Id=1

**** Task 7



```
public class Task7 {  new *
    public static void main(String[] arg) {  new *
            stmt.setString( parameterIndex: 3, email);
            stmt.setString( parameterIndex: 4, city);
            stmt.setString( parameterIndex: 5, contact);
            int count=stmt.executeUpdate();
            System.out.println(count+"Row inserted");
          //
            System.out.println("\nFor update");
            System.out.println("Enter the employee id");
            int id1=sc.nextInt();
            System.out.println("Enter the employee contact");
            String contact1=sc.next();
            // create statement
            PreparedStatement stmt1=con.prepareStatement( sql: "update employee set contact=? where id=?");
           // set value
            stmt1.setString( parameterIndex: 1,contact1);
            stmt1.setInt( parameterIndex: 2,id1);
            // execute satement
            int count1=stmt1.executeUpdate();
```

```
For update
Enter the employee id
2
Enter the employee contact
1234567894
1 row updated
```



```
public class Task7 {  new *
    public static void main(String[] arg) {  new *
            stmt.setString( parameterIndex: 3, email);
            stmt.setString( parameterIndex: 4, city);
            stmt.setString( parameterIndex: 5, contact);
            int count=stmt.executeUpdate();
            System.out.println(count+"Row inserted");
          //
            System.out.println("\nFor update");
            System.out.println("Enter the employee id");
            int id1=sc.nextInt();
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.6\lib\idea_rt.jar=57852" -Dfile.encoding=UTF-8 -Dsun.s
Name : Mahesh Nikas
Batch : IT
connection Successfuly

Insert value for inserting..
Enter Id:
4
Enter Name:
Mahesh
Enter Email:
mahesh@gmail.com
Enter City:
Mehker
Enter Contact:
9356736650
1Row inserted
```

Overview

Task 7 demonstrates JDBC operations for inserting a new employee record and updating an existing employee's contact number using PreparedStatement. This task focuses on parameterized queries and safe update operations.

Class Explanation: Task7

The Task7 class contains the main() method. It establishes a database connection, takes employee details from the user, inserts a new record into the employee table, and then updates the contact number of an existing employee.

Key JDBC Components Used

- Class.forName(): Loads the Oracle JDBC driver.

- DriverManager.getConnection(): Creates a connection to the Oracle database.

- PreparedStatement: Executes INSERT and UPDATE queries securely.

- executeUpdate(): Performs database update and returns number of affected rows.

- Scanner: Collects input values from the user.

Operations Explained

1. Insert Employee Record

The program accepts employee Id, Name, Email, City, and Contact from the user. These values are passed to a PreparedStatement with an INSERT query. The executeUpdate() method inserts the record into the employee table.

2. Update Employee Contact

The program asks for the employee Id and the new contact number. Using a PreparedStatement with an UPDATE query, the contact number of the specified employee is updated.

Sample Input Used

- Insert → Id=4, Name=Mahesh, Email=mahesh@gmail.com, City=Mehkar, Contact=9356736650

- Update → Employee Id=2, New Contact=1234567894

Sample Output

```
1 Row inserted
1 row updated
```