

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows a project named "Task" with a "src" directory containing "Task1", "Task2", "Task3", "Task4", and "Task5".
- Code Editor:** Task1.java contains Java code to connect to an Oracle database. It prints "Name : Mahesh Nikas" and "Batch : IT", loads the Oracle JDBC driver, establishes a connection using DriverManager.getConnection(), and prints "connection Successfully".
- Run Tab:** Task1 is selected for execution.
- Output Tab:** Displays the command used ("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..."), the printed output ("Name : Mahesh Nikas", "Batch : IT", "connection Successfully"), and the exit status ("Process finished with exit code 0").
- Bottom Status Bar:** Shows the current time (10:46), encoding (CRLF), file format (UTF-8), and code style settings (4 spaces).

## 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

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows a project named "Task" with a "src" directory containing "Task1", "Task2", "Task3", "Task4", and "Task5".
- Code Editors:** Task1.java and Task2.java are open. Task2.java contains Java code to read student data from a file. It prints "Name : Mahesh Nikas" and "Batch : IT", loads the Oracle JDBC driver, establishes a connection using DriverManager.getConnection(), and prints "connection Successfully".
- Run Tab:** Task2 is selected for execution.
- Output Tab:** Displays the command used ("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..."), the printed output ("Name : Mahesh Nikas", "Batch : IT", "connection Successfully"), and the exit status ("Process finished with exit code 0").
- Bottom Status Bar:** Shows the current time (9:46), encoding (CRLF), file format (UTF-8), and code style settings (4 spaces). The status bar also indicates the presence of 1 error (red exclamation mark) and 3 warnings (green checkmark).

Task2.java code (extracted):

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

public class Task2 {
    public static void main(String[] args) {
        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 Successfully");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

## **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

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows a project named "Task" with three files: Task1.java, Task2.java, and Task3.java. Task3.java is the active file.
- Code Editor:** Displays the Java code for Task3.java:

```
public class Task3 { new*
    public static void main(String[] args) { 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");
    }
}
```
- Run Tab:** Set to "Task3".
- Output Tab:** Shows the command used to run the application and the standard output:

```
"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.jnu...
Name : Mahesh Nikas
Batch : IT
connection Successfully
Enter Id
6
Enter Name
Mahesh
Enter age
22
Enter Marks
99
1 ows inserted
Process finished with exit code 0
```
- Bottom Status Bar:** Shows the current time (9:46), encoding (CRLF), and file settings (4 spaces).

## **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

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows a tree structure with packages Task1, Task2, Task3, Task4, Task5, and Task6, along with files .gitignore and Task.iml.
- Code Editor:** Task4.java is open, displaying Java code that uses a PreparedStatement to update a database table.
- Run Output:** The terminal shows the execution of the program. It prints the connection URL, user name, batch name, and connection status. It then prompts for an ID (6) and marks (82), and outputs "1 rows updated".
- Status Bar:** Shows the path "Task > src > Task4 > Task4.java", the file encoding "UTF-8", and the line count "1 row".

```
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.net.http.allowRestrictedHeaders=true
Name : Mahesh Nikas
Batch : IT
connection Successfully
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

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows a tree structure with packages Task1, Task2, Task3, Task4, Task5, and Task6, along with files .gitignore and Task.iml.
- Code Editor:** Task5.java is open, displaying Java code that uses a PreparedStatement to delete a record from a database table.
- Run Output:** The terminal shows the execution of the program. It prints the connection URL, user name, batch name, and connection status. It then prompts for an ID (6) and outputs "1 rows deleted".
- Status Bar:** Shows the path "Task > src > Task5 > Task5.java", the file encoding "UTF-8", and the line count "1 row".

```
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.net.http.allowRestrictedHeaders=true
Name : Mahesh Nikas
Batch : IT
connection Successfully
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**

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The project is named "Task". It contains several Java files: Task1.java, Task2.java, Task3.java, Task4.java, Task5.java, Task6.java (which is the active file), and insert.java.
- Task6.java Content:** The code implements a main method that inserts a row into a table, updates it, and then deletes it. It uses a PreparedStatement and prints the count of rows inserted.
- Terminal Output:** The terminal window shows the command run: "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...". The output shows the program's interaction with the database, including connection success, table creation, and record deletion.
- Run Tab:** The "Run" tab is selected, showing the configuration for running Task6.
- Status Bar:** The status bar at the bottom right shows the time as 69:52, encoding as CRLF, and file encoding as UTF-8.

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The project is named "Task". The "src" directory contains files: Task1.java, Task2.java, Task3.java, Task4.java, Task5.java, Task6.java, and insert.java.
- Task6.java Content:**

```
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");
        // statement close
        stmt1.close();
    }
    // UPDATE
    else if (choice == 2) {
```
- Run Tab:** The "Task6" run configuration is selected.
- Output Tab:** The output shows the program's execution:

```
"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.jnu...
Name : Mahesh Nikas
Batch : IT
connection Successfully
Table already exist

Choose Operation:
1 : Insert
2 : Update
3 : Delete
2
Enter Id to update:
1
Enter New City:
Shegaon
1 rows updated
```
- Status Bar:** Shows the time as 69:52, encoding as CRLF, and file encoding as UTF-8.

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The project is named "Task". The "src" directory contains files: Task1.java, Task2.java, Task3.java, Task4.java, Task5.java, Task6.java, and insert.java.
- Task6.java Content:**

```
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");
        // statement close
        stmt1.close();
    }
    // UPDATE
    else if (choice == 2) {
```
- Run Tab:** The "Task6" run configuration is selected.
- Output Tab:** The output shows the program's execution:

```
"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.jnu...
Name : Mahesh Nikas
Batch : IT
connection Successfully
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
```
- Status Bar:** Shows the time as 69:52, encoding as CRLF, and file encoding as UTF-8.

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@gmail.com, City=Mehkar, Contact=9356736650
- Update → Id=1, New City=Shegaon
- Delete → Id=1

## \*\*\*\* Task 7

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows a project named "Task" with a "src" directory containing files: Task1.java, Task2.java, Task3.java, Task4.java, Task5.java, Task6.java, and Task7.java. Task7.java is currently selected.
- Code Editor:** Displays the Java code for Task7.java. The code uses JDBC to update an employee's contact information in a database. It prompts for employee ID and contact, then updates the database and prints the count of rows affected.
- Run Tab:** Shows the run configuration for "Task7".
- Output Tab:** Displays the execution results:
  - For update
  - Enter the employee id
  - 2
  - Enter the employee contact
  - 1234567894
  - 1 row updated
- Status Bar:** Shows the file path as "Task > src > Task7 > Task7.java", encoding as "UTF-8", and line numbers from 54:1 to 54:1.

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows a project named "Task" with a "src" directory containing files: Task1.java, Task2.java, Task3.java, Task4.java, Task5.java, Task6.java, and Task7.java. Task7.java is currently selected.
- Code Editor:** Displays the Java code for Task7.java, which is identical to the one in the first screenshot.
- Run Tab:** Shows the run configuration for "Task7".
- Output Tab:** Displays the execution results:
  - "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.jna.dll=C:\Windows\Sun\Java\Deployment\jna\jna.dll
  - Name : Mahesh Nikas
  - Batch : IT
  - connection Successfully
  - Insert value for inserting..
  - Enter Id:
  - 4
  - Enter Name:
  - Mahesh
  - Enter Email:
  - mahesh@gmail.com
  - Enter City:
  - Mehkher
  - Enter Contact:
  - 9356736650
  - 1Row inserted
- Status Bar:** Shows the file path as "Task > src > Task7 > Task7.java", encoding as "UTF-8", and line numbers from 54:1 to 54:1.

## **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

