IBATIS - UPDATE OPERATION

We discussed, in the last chapter, how to perform READ operation on a table using iBATIS. This chapter explains how you can update records in a table using iBATIS.

We have the following EMPLOYEE table in MySQL −

```
CREATE TABLE EMPLOYEE (
   id INT NOT NULL auto_increment,
   first_name VARCHAR(20) default NULL,
   last_name  VARCHAR(20) default NULL,
   salary     INT  default NULL,
   PRIMARY KEY (id)
);
```

This table has only one record as follows −

```
mysql> select * from EMPLOYEE;
+----+------------+-----------+--------+
| id | first_name | last_name | salary |
+----+------------+-----------+--------+
|  1 | Zara       | Ali       |   5000 |
+----+------------+-----------+--------+
1 row in set (0.00 sec)
```

Employee POJO Class

To perform udpate operation, you would need to modify Employee.java file as follows −

```
public class Employee {
   private int id;
   private String first_name;
   private String last_name;
   private int salary;

   /* Define constructors for the Employee class. */
   public Employee() {}

   public Employee(String fname, String lname, int salary) {
      this.first_name = fname;
      this.last_name = lname;
      this.salary = salary;
   }

   /* Here are the required method definitions */
   public int getId() {
      return id;
   }

   public void setId(int id) {
      this.id = id;
   }

   public String getFirstName() {
      return first_name;
   }

   public void setFirstName(String fname) {
      this.first_name = fname;
   }

   public String getLastName() {
      return last_name;
```

```
   }
   public void setlastName(String lname) {
      this.last_name = lname;
   }

   public int getSalary() {
      return salary;
   }

   public void setSalary(int salary) {
      this.salary = salary;
   }

} /* End of Employee */
```

Employee.xml File

To define SQL mapping statement using iBATIS, we would add <update> tag in Employee.xml and inside this tag definition, we would define an "id" which will be used in IbatisUpdate.java file for executing SQL UPDATE query on database.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sqlMap PUBLIC "-//ibatis.apache.org//DTD SQL Map 2.0//EN"
"http://ibatis.apache.org/dtd/sql-map-2.dtd">

<sqlMap namespace="Employee">

   <insert >
      INSERT INTO EMPLOYEE(first_name, last_name, salary)
      values (#first_name#, #last_name#, #salary#)

      <selectKey resultClass="int" keyProperty="id">
         select last_insert_id() as id
      </selectKey>
   </insert>

   <select >
      SELECT * FROM EMPLOYEE
   </select>

   <update >
      UPDATE EMPLOYEE
      SET    first_name = #first_name#
      WHERE  id = #id#
   </update>

</sqlMap>
```

IbatisUpdate.java File

This file has application level logic to update records into the Employee table −

```
import com.ibatis.common.resources.Resources;
import com.ibatis.sqlmap.client.SqlMapClient;
import com.ibatis.sqlmap.client.SqlMapClientBuilder;

import java.io.*;
import java.sql.SQLException;
import java.util.*;

public class IbatisUpdate{
   public static void main(String[] args)
   throws IOException,SQLException{
      Reader rd = Resources.getResourceAsReader("SqlMapConfig.xml");
      SqlMapClient smc = SqlMapClientBuilder.buildSqlMapClient(rd);

      /* This would update one record in Employee table. */
```

```
      System.out.println("Going to update record.....");
      Employee rec = new Employee();
      rec.setId(1);
      rec.setFirstName( "Roma");
      smc.update("Employee.update", rec );
      System.out.println("Record updated Successfully ");

      System.out.println("Going to read records.....");
      List <Employee> ems = (List<Employee>)
         smc.queryForList("Employee.getAll", null);
      Employee em = null;

      for (Employee e : ems) {
         System.out.print("  " + e.getId());
         System.out.print("  " + e.getFirstName());
         System.out.print("  " + e.getLastName());
         System.out.print("  " + e.getSalary());
         em = e;
         System.out.println("");
      }

      System.out.println("Records Read Successfully ");
   }
}
```

Compilation and Run

Here are the steps to compile and run the above-mentioned software. Make sure you have set PATH and CLASSPATH appropriately before proceeding for compilation and execution.

- Create Employee.xml as shown above.
- Create Employee.java as shown above and compile it.
- Create IbatisUpdate.java as shown above and compile it.
- Execute IbatisUpdate binary to run the program.

You would get following result, and a record would be updated in EMPLOYEE table and later, the same record would be read from the EMPLOYEE table.

```
Going to update record.....
Record updated Successfully
Going to read records.....
   1  Roma  Ali  5000
Records Read Successfully
```