

# IBATIS - DELETE OPERATION

[http://www.tutorialspoint.com/ibatis/ibatis\\_delete\\_operation.htm](http://www.tutorialspoint.com/ibatis/ibatis_delete_operation.htm)

Copyright © tutorialspoint.com

This chapter describes how to delete records from 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)  
);
```

Assume this table has two records as follows –

```
mysql> select * from EMPLOYEE;  
+----+-----+-----+-----+  
| id | first_name | last_name | salary |  
+----+-----+-----+-----+  
|  1 | Zara      | Ali      |  5000 |  
|  2 | Roma     | Ali      |  3000 |  
+----+-----+-----+-----+  
2 row in set (0.00 sec)
```

## Employee POJO Class

To perform delete operation, you do not need to modify Employee.java file. Let us keep it as it was in the last chapter.

```
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 <delete> tag in Employee.xml and inside this tag definition, we would define an "id" which will be used in IbatisDelete.java file for executing SQL DELETE 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>

    <delete >
        DELETE FROM EMPLOYEE
        WHERE id = #id#
    </delete>

</sqlMap>

```

## IbatisDelete.java File

This file has application level logic to delete records from 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 IbatisDelete{
    public static void main(String[] args)
    throws IOException, SQLException{
        Reader rd = Resources.getResourceAsReader("SqlMapConfig.xml");
        SqlMapClient smc = SqlMapClientBuilder.buildSqlMapClient(rd);

        /* This would delete one record in Employee table. */
        System.out.println("Going to delete record.....");
        int id = 1;

        smc.delete("Employee.delete", id );
        System.out.println("Record deleted 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 IbatisDelete.java as shown above and compile it.
- Execute IbatisDelete binary to run the program.

You would get the following result, and a record with ID = 1 would be deleted from the EMPLOYEE table and the rest of the records would be read.

```

Going to delete record.....
Record deleted Successfully
Going to read records.....
 2 Roma Ali 3000
Records Read Successfully

```