MYBATIS - DELETE OPERATION

This chapter describes how to delete records from a table using MyBatis.

We have the following STUDENT table in MySQL −

```
CREATE TABLE details.student(
   ID int(10) NOT NULL AUTO_INCREMENT,
   NAME varchar(100) NOT NULL,
   BRANCH varchar(255) NOT NULL,
   PERCENTAGE int(3) NOT NULL,
   PHONE int(11) NOT NULL,
   EMAIL varchar(255) NOT NULL,
   PRIMARY KEY (`ID`)
);
```

Assume, this table has two records as −

```
mysql> select * from STUDENT;
+----+----------+--------+------------+-----------+----------------------+
| ID |  NAME    | BRANCH | PERCENTAGE |  PHONE    |       EMAIL          |
+----+----------+--------+------------+-----------+----------------------+
| 1  | Mohammad |   It   |     80     | 900000000 | mohamad123@yahoo.com |
| 2  | shyam    |   It   |     75     | 984800000 | shyam@gmail.com      |
+----+----------+--------+------------+-----------+----------------------+
2 rows in set (0.00 sec)
```

STUDENT POJO Class

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

```
public class Student {

   private int id;
   private String name;
   private String branch;
   private int percentage;
   private int phone;
   private String email;

   public Student(int id, String name, String branch, int percentage, int phone, String
email) {
      super();
      this.id = id;
      this.name = name;
      this.setBranch(branch);
      this.setPercentage(percentage);
      this.phone = phone;
      this.email = email;
   }

   public Student() {}

   public int getId() {
      return id;
   }

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

   public String getName() {
```

```
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }

   public int getPhone() {
      return phone;
   }

   public void setPhone(int phone) {
      this.phone = phone;
   }

   public String getEmail() {
      return email;
   }

   public void setEmail(String email) {
      this.email = email;
   }

   public String getBranch() {
      return branch;
   }

   public void setBranch(String branch) {
      this.branch = branch;
   }

   public int getPercentage() {
      return percentage;
   }

   public void setPercentage(int percentage) {
      this.percentage = percentage;
   }

   public String toString(){
      StringBuilder sb = new StringBuilder();

      sb.append("Id = ").append(id).append(" - ");
      sb.append("Name = ").append(name).append(" - ");
      sb.append("Branch = ").append(branch).append(" - ");
      sb.append("Percentage = ").append(percentage).append(" - ");
      sb.append("Phone = ").append(phone).append(" - ");
      sb.append("Email = ").append(email);

      return sb.toString();
   }

}
```

Student.xml File

To define SQL mapping statement using MyBatis, we would use **<delete>** tag in Student.xml and inside this tag definition, we would define an "id" which will be used in mybatisDelete.java file for executing SQL DELETE query on database.

```
<?xml version = "1.0" encoding = "UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace = "Student">
   <resultMap id = "result" type = "Student">
      <result property = "id" column = "ID"/>
```

```
   </resultMap>

   <delete id = "deleteById" parameterType = "int">
      DELETE from STUDENT WHERE ID = #{id};
   </delete>

</mapper>
```

MyBatisDelete.java File

This file has application level logic to delete records from the Student table −

```
import java.io.IOException;
import java.io.Reader;

import org.apache.ibatis.io.Resources;
import org.apache.ibatis.session.SqlSession;
import org.apache.ibatis.session.SqlSessionFactory;
import org.apache.ibatis.session.SqlSessionFactoryBuilder;

public class mybatisDelete {

   public static void main(String args[]) throws IOException{

      Reader reader = Resources.getResourceAsReader("SqlMapConfig.xml");
      SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(reader);
      SqlSession session = sqlSessionFactory.openSession();

      //Delete operation
      session.delete("Student.deleteById", 2);
      session.commit();
      session.close();
      System.out.println("Record deleted successfully");

   }

}
```

Compilation and Run

Here are the steps to compile and run mybatisDelete.java. Make sure, you have set PATH and CLASSPATH appropriately before proceeding for compilation and execution.

- Create Student.xml as shown above.
- Create SqlMapConfig.xml as shown in the MYBATIS - Configuration XML chapter of this tutorial.
- Create Student.java as shown above and compile it.
- Create mybatisDelete.java as shown above and compile it.
- Execute mybatisDelete binary to run the program.

You would get the following result, and a record with ID = 1 would be deleted from the STUDENT.

```
Records Read Successfully
```

If you check the STUDENT table, it should display the following result −

```
mysql> select * from student;
+----+----------+--------+------------+----------+----------------------+
| ID |   NAME   | BRANCH | PERCENTAGE |  PHONE   |        EMAIL         |
+----+----------+--------+------------+----------+----------------------+
| 1  | Mohammad |   It   |    80      | 90000000 | mohamad123@yahoo.com |
+----+----------+--------+------------+----------+----------------------+
```

```
1 row in set (0.00 sec)
```