PreparedStatement interface

The PreparedStatement interface is a subinterface of Statement. It is used to execute parameterized query.

Let's see the example of parameterized query:

1. String sql="insert into emp values(?,?,?)";

As you can see, we are passing parameter (?) for the values. Its value will be set by calling the setter methods of PreparedStatement.

Why use PreparedStatement?

**Improves performance**: The performance of the application will be faster if you use PreparedStatement interface because query is compiled only once.

How to get the instance of PreparedStatement?

The prepareStatement() method of Connection interface is used to return the object of PreparedStatement. Syntax:

1. **public** PreparedStatement prepareStatement(String query)**throws** SQLException{}

Methods of PreparedStatement interface

The important methods of PreparedStatement interface are given below:

|  |  |
| --- | --- |
| **Method** | **Description** |
| public void setInt(int paramIndex, int value) | sets the integer value to the given parameter index. |
| public void setString(int paramIndex, String value) | sets the String value to the given parameter index. |
| public void setFloat(int paramIndex, float value) | sets the float value to the given parameter index. |
| public void setDouble(int paramIndex, double value) | sets the double value to the given parameter index. |
| public int executeUpdate() | executes the query. It is used for create, drop, insert, update, delete etc. |
| public ResultSet executeQuery() | executes the select query. It returns an instance of ResultSet. |

Example of PreparedStatement interface that inserts the record

First of all create table as given below:

1. create table emp(id number(10),name varchar2(50));

Now insert records in this table by the code given below:

1. **import** java.sql.\*;
2. **class** InsertPrepared{
3. **public** **static** **void** main(String args[]){
4. **try**{
5. Class.forName("oracle.jdbc.driver.OracleDriver");
7. Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
9. PreparedStatement stmt=con.prepareStatement("insert into Emp values(?,?)");
10. stmt.setInt(1,101);//1 specifies the first parameter in the query
11. stmt.setString(2,"Ratan");
13. **int** i=stmt.executeUpdate();
14. System.out.println(i+" records inserted");
16. con.close();
18. }**catch**(Exception e){ System.out.println(e);}
20. }
21. }

[download this example](https://www.javatpoint.com/src/jdbc/InsertPrepared.zip)

Example of PreparedStatement interface that updates the record

1. PreparedStatement stmt=con.prepareStatement("update emp set name=? where id=?");
2. stmt.setString(1,"Sonoo");//1 specifies the first parameter in the query i.e. name
3. stmt.setInt(2,101);
5. **int** i=stmt.executeUpdate();
6. System.out.println(i+" records updated");

[download this example](https://www.javatpoint.com/src/jdbc/UpdatePrepared.zip)

Example of PreparedStatement interface that deletes the record

1. PreparedStatement stmt=con.prepareStatement("delete from emp where id=?");
2. stmt.setInt(1,101);
4. **int** i=stmt.executeUpdate();
5. System.out.println(i+" records deleted");

[download this example](https://www.javatpoint.com/src/jdbc/DeletePrepared.zip)

Example of PreparedStatement interface that retrieve the records of a table

1. PreparedStatement stmt=con.prepareStatement("select \* from emp");
2. ResultSet rs=stmt.executeQuery();
3. **while**(rs.next()){
4. System.out.println(rs.getInt(1)+" "+rs.getString(2));
5. }

[download this example](https://www.javatpoint.com/src/jdbc/RetrievePrepared.zip)

Example of PreparedStatement to insert records until user press n

1. **import** java.sql.\*;
2. **import** java.io.\*;
3. **class** RS{
4. **public** **static** **void** main(String args[])**throws** Exception{
5. Class.forName("oracle.jdbc.driver.OracleDriver");
6. Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
8. PreparedStatement ps=con.prepareStatement("insert into emp130 values(?,?,?)");
10. BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.in));
12. **do**{
13. System.out.println("enter id:");
14. **int** id=Integer.parseInt(br.readLine());
15. System.out.println("enter name:");
16. String name=br.readLine();
17. System.out.println("enter salary:");
18. **float** salary=Float.parseFloat(br.readLine());
20. ps.setInt(1,id);
21. ps.setString(2,name);
22. ps.setFloat(3,salary);
23. **int** i=ps.executeUpdate();
24. System.out.println(i+" records affected");
26. System.out.println("Do you want to continue: y/n");
27. String s=br.readLine();
28. **if**(s.startsWith("n")){
29. **break**;
30. }
31. }**while**(**true**);
33. con.close();
34. }}