|  |  |
| --- | --- |
| INTERFACE NAME | WHAT FOR USED |
| 1.STATEMENT | Used for executing simple SQL statements like CRUD (create, retrieve, update and delete) |
| 2.PREPAREDSTATEMENT | Used for executing dynamic SQL statements |
| 3.CALLABLESTATEMENT | Used for executing stored procedures |

Example:

**Calling a Stored Procedure Returning a Result Set from Java**

A stored procedure can returns a result set. Consider the following procedure:

|  |  |
| --- | --- |
|  | CREATE PROCEDURE `get\_books`(IN rate INT)  BEGIN      SELECT \* FROM book WHERE rating >= rate;  END |

Let’s see how to retrieve this result set in Java. The following code snippet shows you how to retrieve and process a result set returned from a stored procedure using JDBC code

|  |  |
| --- | --- |
|  | CallableStatement statement = conn.prepareCall("{call get\_books(?)}");    statement.setInt(1, 5);    boolean hadResults = statement.execute();    while (hadResults) {      ResultSet resultSet = statement.getResultSet();          // process result set      while (resultSet.next()) {          // retrieve values of fields  String title = resultSet.getString("title");        }        hadResults = statement.getMoreResults();  } |

Following example shows how to use Collable Statement in java:

Goto: <https://www.codejava.net/java-se/jdbc/jdbc-examples-for-calling-stored-procedures-mysql>