JDBC 接口



| **接口** | **应用场景** |
| --- | --- |
| Statement | 当在运行时使用静态 SQL 语句时（Statement接口不能接收参数） |
| CallableStatement | 当要访问数据库中的存储过程时（CallableStatement对象的接口还可以接收运行时输入参数） |
| PreparedStatement | 当计划多次使用 SQL 语句时（PreparedStatement 接口接收在运行时输入参数） |

**1.1 Statement接口**

**1.1.1 获得接口: 使用 Connection 对象的 createStatement() 方法进行创建。**

**1.1.2 常用方法**

| **方法** | **说明** |
| --- | --- |
| boolean execute(String SQL) | 如果 ResultSet 对象可以被检索返回布尔值 true，否则返回 false。使用这个方法来执行 SQL DDL 语句，或当需要使用真正的动态 SQL |
| int executeUpdate(String SQL) | 用于执行 INSERT、UPDATE 或 DELETE 语句以及 SQLDDL（数据定义语言）语句。返回值是一个整数，指示受影响的行数（即更新计数） |
| ResultSet executeQuery(String SQL) | 返回 ResultSet 对象。用于产生单个结果集的语句，例如 SELECT 语句 |

**1.1.3 关闭接口资源： 正如关闭一个 Connection 对象来释放数据库连接资源，出于同样的原因，也应该关闭 Statement 对象。如果关闭了 Connection 对象首先它会关闭 Statement 对象，然而应该始终明确关闭 Statement 对象，以确保正确的清除。**

**1.2 PreparedStatement 接口**

**PreparedStatement 接口扩展了 Statement 接口，有利于高效地执行多次使用的 SQL 语句。我们先来创建一个 PreparedStatement 对象。 Statement 为一条 SQL 语句生成执行计划。如果要执行两条 SQL 语句，会生成两个执行计划。一万个查询就生成一万个执行计划！**

**1.2.1 获得接口： 使用 Connection 对象的 prepareStatement(String) 方法进行创建。**

String sql = "UPDATE student set age=? WHERE id=?";

PreparedStatement stmt = conn.prepareStatement(sql);

注： PreparedStatement **的中文翻译是：预备声明，即预先准备的声明**

他表示，**这个接口是特定SQL格式的语句声明，**如上面是更新数据的声明，还可以是增加数据等的声明，

和Statement接口的区别是：Statement接口是在获得之后在不绑定特定的SQL语句，通过在通过在方法调用时接收字符串作为绑定的SQL语句，只要保证字符串正确可以了。而PreparedStatement接口是在建立接口的时候就绑定了特定类型的SQL语句，这相当于，他与为你写了一半可重用的语句框架，你只需要为该？号赋值就可以实现不同数据的语句操作。例如

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

如果要执行两次不同数据的同一类型SQL语句：

updata student set age = 21 where id = 1;

update student set age = 22 where id = 2;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

如果使用：Statement接口，就要声明两次字符串

statement = connection.createStatement();  
String update = “updata student set age = 21 where id = 1”;  
statement.executeUpdate(update);  
update = “updata student set age = 21 where id = 2”;  
statement.executeUpdate(update);

如果使用PerpareStatement接口，则只需如下：

String update = “update student set age = ? where id = ?”;  
perpareStatement = connection.perpareStatement(update);  
perpareStatement.setInt(1, 22); //第1个？值为22  
perpareStatement.setInt(2, 1); //第2个？值为1

perpareStatement.executeUpdate(); //执行SQL语句

perpareStatement.setInt(1, 23); //第1个？值为23  
perpareStatement.setInt(2, 2); //第2个？值为2

perpareStatement.executeUpdate(); //执行SQL语句