JDBC基础教程之PreparedStatement PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/253/2021_2022_JDBC_E5_9F_BA_E7_A1_80_c97_253051.htm 概述 该 PreparedStatement 接口继承 Statement，并与之在两方面有所不同： PreparedStatement 实例包含已编译的 SQL 语句。这就是使语句“ 准备好 ”。包含于 PreparedStatement 对象中的 SQL 语句可具有一个或多个 IN 参数。IN参数的值在 SQL 语句创建时未被指定。相反的，该语句为每个 IN 参数保留一个问号（“ ？ ”）作为占位符。每个问号的值必须在该语句执行之前，通过适当的setXXX 方法来提供。 由于 PreparedStatement 对象已预编译过，所以其执行速度要快于 Statement 对象。因此，多次执行的 SQL 语句经常创建为 PreparedStatement 对象，以提高效率。 作为 Statement 的子类，PreparedStatement 继承了 Statement 的所有功能。另外它还添加了一整套方法，用于设置发送给数据库以取代 IN 参数占位符的值。同时，三种方法 execute、 executeQuery 和 executeUpdate 已被更改以使之不再需要参数。这些方法的 Statement 形式（接受 SQL 语句参数的形式）不应该用于 PreparedStatement 对象。 1、创建 PreparedStatement 对象 以下的代码段（其中 con 是 Connection 对象）创建包含带两个 IN 参数占位符的 SQL 语句的 PreparedStatement 对象：PreparedStatement pstmt = con.prepareStatement("UPDATE table4 SET m = ? WHERE x = ?"). pstmt 对象包含语句 "UPDATE table4 SET m = ? WHERE x = ?" ，它已发送给DBMS，并为执行作好了准备。 2、传递 IN 参数 在执行 PreparedStatement 对象之前，必须设置每个 ? 参数

的值。这可通过调用 setXXX 方法来完成，其中 XXX 是与该参数相应的类型。例如，如果参数具有Java 类型 long，则使用的方法就是 setLong。setXXX 方法的第一个参数是要设置的参数的序数位置，第二个参数是设置给该参数的值。例如，以下代码将第一个参数设为 123456789，第二个参数设为 100000000：pstmt.setLong(1, 123456789).pstmt.setLong(2, 100000000). 一旦设置了给定语句的参数值，就可用它多次执行该语句，直到调用clearParameters 方法清除它为止。在连接的缺省模式下（启用自动提交），当语句完成时将自动提交或还原该语句。 如果基本数据库和驱动程序在语句提交之后仍保持这些语句的打开状态，则同一个 PreparedStatement 可执行多次。如果这一点不成立，那么试图通过使用PreparedStatement 对象代替 Statement 对象来提高性能是没有意义的。 利用 pstmt（前面创建的 PreparedStatement 对象），以下代码例示了如何设置两个参数占位符的值并执行 pstmt 10 次。如上所述，为做到这一点，数据库不能关闭 pstmt。在该示例中，第一个参数被设置为 "Hi"并保持为常数。在 for 循环中，每次都将第二个参数设置为不同的值：从 0 开始，到 9 结束。pstmt.setString(1, "Hi").for (int i = 0. i pstmt.setInt(2, i). int rowCount = pstmt.executeUpdate().} 3、IN 参数中数据类型的一致性 setXXX 方法中的 XXX 是 Java 类型。它是一种隐含的 JDBC 类型（一般 SQL 类型），因为驱动程序将把 Java 类型映射为相应的 JDBC 类型（遵循该 JDBCGuide中 § 8.6.2 “ 映射 Java 和 JDBC 类型 ” 表中所指定的映射），并将该 JDBC 类型发送给数据库。例如，以下代码段将 PreparedStatement 对象 pstmt 的第二个参数设置为 44，Java 类型为 short

：pstmt.setShort(2, 44). 驱动程序将 44 作为 JDBC SMALLINT 发送给数据库，它是 Java short 类型的标准映射。 程序员的责任是确保将每个 IN 参数的 Java 类型映射为与数据库所需的 JDBC 数据类型兼容的 JDBC 类型。不妨考虑数据库需要 JDBC SMALLINT 的情况。如果使用方法 setByte ，则驱动程序将 JDBC TINYINT 发送给数据库。这是可行的，因为许多数据库可从一种相关的类型转换为另一种类型，并且通常 TINYINT 可用于SMALLINT 适用的任何地方 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com