

JAVA辅导:JDBC基础教程之语句 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_JAVA_E8_BE_85_E5_AF_BC_c97_253053.htm 概述 Statement 对象用于将

SQL 语句发送到数据库中。实际上有三种 Statement 对象，它们都作为在给定连接上执行 SQL 语句的容器：Statement、PreparedStatement（它从 Statement 继承而来）和 CallableStatement（它从 PreparedStatement 继承而来）。它们都专用于发送特定类型的 SQL 语句：Statement 对象用于执行不带参数的简单 SQL 语句；PreparedStatement 对象用于执行带或不带 IN 参数的预编译 SQL 语句；CallableStatement 对象用于执行对数据库已存储过程的调用。Statement 接口提供了执行语句和获取结果的基本方法。PreparedStatement 接口添加了处理 IN 参数的方法；而 CallableStatement 添加了处理 OUT 参数的方法。

1、创建 Statement 对象 建立了到特定数据库的连接之后，就可用该连接发送 SQL 语句。Statement 对象用 Connection 的方法 createStatement 创建，如下列代码段中所示：

```
Connection con = DriverManager.getConnection(url, "sunny", "");
Statement stmt = con.createStatement();
```

为了执行 Statement 对象，被发送到数据库的 SQL 语句将被作为参数提供给 Statement 的方法：

```
ResultSet rs = stmt.executeQuery("SELECT a, b, c FROM Table2");
```

2、使用 Statement 对象执行语句 Statement 接口提供了三种执行 SQL 语句的方法：executeQuery、executeUpdate 和 execute。使用哪一个方法由 SQL 语句所产生的内容决定。方法 executeQuery 用于产生单个结果集的语句，例如 SELECT 语句。方法 executeUpdate 用于执行

INSERT、UPDATE 或 DELETE 语句以及 SQL DDL（数据定义语言）语句，例如 CREATE TABLE 和 DROP TABLE。INSERT、UPDATE 或 DELETE 语句的效果是修改表中零行或多行中的一列或多列。executeUpdate 的返回值是一个整数，指示受影响的行数（即更新计数）。对于 CREATE TABLE 或 DROP TABLE 等不操作行的语句，executeUpdate 的返回值总为零。方法 execute 用于执行返回多个结果集、多个更新计数或二者组合的语句。因为多数程序员不会需要该高级功能，所以本概述后面将在单独一节中对其进行介绍。执行语句的所有方法都将关闭所调用的 Statement 对象的当前打开结果集（如果存在）。这意味着在重新执行 Statement 对象之前，需要完成对当前 ResultSet 对象的处理。应注意，继承了 Statement 接口中所有方法的 PreparedStatement 接口都有自己的 executeQuery、executeUpdate 和 execute 方法。Statement 对象本身不包含 SQL 语句，因而必须给 Statement.execute 方法提供 SQL 语句作为参数。PreparedStatement 对象并不将 SQL 语句作为参数提供给这些方法，因为它们已经包含预编译 SQL 语句。CallableStatement 对象继承这些方法的 PreparedStatement 形式。对于这些方法的 PreparedStatement 或 CallableStatement 版本，使用查询参数将抛出 SQLException。

3、语句完成 当连接处于自动提交模式时，其中所执行的语句在完成时将自动提交或还原。语句在已执行且所有结果返回时，即认为已完成。对于返回一个结果集的 executeQuery 方法，在检索完 ResultSet 对象的所有行时该语句完成。对于方法 executeUpdate，当它执行时语句即完成。但在少数调用方法 execute 的情况中，在检索所有结果集或它生成的更新计

数之后语句才完成。有些 DBMS 将已存储过程中的每条语句视为独立的语句；而另外一些则将整个过程视为一个复合语句。在启用自动提交时，这种差别就变得非常重要，因为它影响什么时候调用 commit 方法。在前一种情况中，每条语句单独提交；在后一种情况中，所有语句同时提交。

4、关闭 Statement 对象

Statement 对象将由 Java 垃圾收集程序自动关闭。而作为一种好的编程风格，应在不需要 Statement 对象时显式地关闭它们。这将立即释放 DBMS 资源，有助于避免潜在的内存问题。

5、Statement 对象中的 SQL 转义语法

Statement 可包含使用 SQL 转义语法的 SQL 语句。转义语法告诉驱动程序其中的代码应该以不同方式处理。驱动程序将扫描任何转义语法，并将它转换成特定数据库可理解的代码。这使得转义语法与 DBMS 无关，并允许程序员使用在没有转义语法时不可用的功能。转义子句由花括号和关键字界定：`{keyword . . . parameters . . .}` 该关键字指示转义子句的类型，如下所示。

`escape` 表示 LIKE 转义字符 字符 “%” 和 “_” 类似于 SQL LIKE 子句中的通配符（“%” 匹配零个或多个字符，而 “_” 则匹配一个字符）。为了正确解释它们，应在其前面加上反斜杠（“\”），它是字符串中的特殊转义字符。在查询末尾包括如下语法即可指定用作转义字符的字符：`{escape escape-character}` 例如，下列查询使用反斜杠字符作为转义字符，查找以下划线开头的标识符名：

```
stmt.executeQuery("SELECT name FROM Identifiers WHERE Id LIKE \"\\_\"% {escape \"\\\"}.
```

`fn` 表示标量函数 几乎所有 DBMS 都具有标量值的数值、字符串、时间、日期、系统和转换函数。要使用这些函数，可使用如下转义语法：关键字 `fn` 后跟所需的

函数名及其参数。例如，下列代码调用函数 `concat` 将两个参数连接在一起：`{fn concat("Hot", "Java")}`。可用下列语法获得当前数据库用户名：`{fn user()}`。标量函数可能由语法稍有不同的 DBMS 支持，而它们可能不被所有驱动程序支持。各种 `DatabaseMetaData` 方法将列出所支持的函数。例如，方法 `getNumericFunctions` 返回用逗号分隔的数值函数列表，而方法 `getStringFunctions` 将返回字符串函数，等等。驱动程序将转义函数调用映射为相应的语法，或直接实现该函数。`d`、`t` 和 `ts` 表示日期和时间文字 DBMS 用于日期、时间和时间标记文字的语法各不相同。JDBC 使用转义子句支持这些文字的语法的 ISO 标准格式。驱动程序必须将转义子句转换成 DBMS 表示。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