

JDBC基础教程之ResultSet对象 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\\_253052.htm](https://www.100test.com/kao_ti2020/253/2021_2022_JDBC_E5_9F_BA_E7_A1_80_c97_253052.htm) 概述 ResultSet 包含符合 SQL 语

句中条件的所有行，并且它通过一套 get 方法（这些 get 方法可以访问当前行中的不同列）提供了对这些行中数据的访问。ResultSet.next 方法用于移动到 ResultSet 中的下一行，使下一行成为当前行。结果集一般是一个表，其中有查询所返回的列标题及相应的值。例如，如果查询为 SELECT a, b, c FROM Table1，则结果集将具有如下形式：a b c-----

-----12345 Cupertino CA83472 Redmond WA83492  
Boston MA 下面的代码段是执行 SQL 语句的示例。该 SQL 语句将返回行集合，其中列 1 为 int，列 2 为 String，而列 3 则为字节数组：Java.sql.Statement stmt =

```
conn.createStatement().ResultSet r = stmt.executeQuery("SELECT a, b, c FROM Table1").while (r.next()){ // 打印当前行的值。 int i = r.getInt("a"). String s = r.getString("b"). float f = r.getFloat("c").
```

```
System.out.println("ROW = " i " " s " " f).} 1、行和光标 ResultSet 维护指向其当前数据行的光标。每调用一次 next 方法，光标向下移动一行。最初它位于第一行之前，因此第一次调用 next 将把光标置于第一行上，使它成为当前行。随着每次调用 next 导致光标向下移动一行，按照从上至下的次序获取ResultSet 行。在 ResultSet 对象或其父辈 Statement 对象关闭之前，光标一直保持有效。在 SQL 中，结果表的光标是有名字的。如果数据库允许定位更新或定位删除，则需要将光标的名字作为参数提供给更新或删除命令。可通过调用方
```

法 `getCursorName` 获得光标名。注意：不是所有的 DBMS 都支持定位更新和删除。可使用 `DatabaseMetaData.supportsPositionedDelete` 和 `supportsPositionedUpdate` 方法来检查特定连接是否支持这些操作。当支持这些操作时，DBMS/驱动程序必须确保适当锁定选定行，以使定位更新不会导致更新异常或其它并发问题。

2、列方法 `getXXX` 提供了获取当前行中某列值的途径。在每一行内，可按任何次序获取列值。但为了保证可移植性，应该从左至右获取列值，并且一次性地读取列值。列名或列号可用于标识要从中获取数据的列。例如，如果 `ResultSet` 对象 `rs` 的第二列名为“title”，并将值存储为字符串，则下列任一代码将获取存储在该列中的值：`String s = rs.getString("title");` `String s = rs.getString(2);` 注意列是从左至右编号的，并且从列 1 开始。同时，用作 `getXXX` 方法的输入的列名不区分大小写。提供使用列名这个选项的目的是为了让在查询中指定列名的用户可使用相同的名字作为 `getXXX` 方法的参数。另一方面，如果 `select` 语句未指定列名（例如在“`select * from table1`”中或列是导出的时），则应该使用列号。这些情况下，用户将无法确切知道列名。有些情况下，SQL 查询返回的结果集中可能有多个列具有相同的名字。如果列名用作 `getXXX` 方法的参数，则 `getXXX` 将返回第一个匹配列名的值。因而，如果多个列具有相同的名字，则需要使用列索引来确保检索了正确的列值。这时，使用列号效率要稍微高一些。关于 `ResultSet` 中列的信息，可通过调用方法 `ResultSet.getMetaData` 得到。返回的 `ResultSetMetaData` 对象将给出其 `ResultSet` 对象各列的编号、类型和属性。如果列名已

知，但不知其索引，则可用方法 findColumn 得到其列号。 3

、数据类型和转换 对于 getXXX 方法，JDBC 驱动程序试图将基本数据转换成指定 Java 类型，然后返回适合的 Java 值。例如，如果 getXXX 方法为 getString，而基本数据库中数据类型为 VARCHAR，则 JDBC 驱动程序将把 VARCHAR 转换成 Java String。getString 的返回值将为 Java String 对象。下表显示了允许用 getXXX 获取的 JDBC 类型及推荐用它获取的 JDBC 类型（通用 SQL 类型）。小写的 x 表示允许 getXXX 方法获取该数据类型；大写的 X 表示对该数据类型推荐使用 getXXX 方法。例如，除了 getBytes 和 getBinaryStream 之外的任何 getXXX 方法都可用来获取 LONGVARCHAR 值，但是推荐根据返回的数据类型使用 getAsciiStream 或 getUnicodeStream 方法。方法 getObject 将任何数据类型返回为 Java Object。当基本数据类型是特定于数据库的抽象类型或当通用应用程序需要接受任何数据类型时，它是非常有用的。可使用 ResultSet.getXXX 方法获取常见的 JDBC 数据类型。“x”表示该 getXXX 方法可合法地用于获取给定 JDBC 类型。“X”表示推荐使用该 getXXX 方法来获取给定 JDBC 类型。

getXXX 方法	通用 SQL 类型
getBytes	X
getBinaryStream	X
getBoolean	x
getByte	x
getCharacterStream	x
getDouble	x
getFloat	x
getInteger	x
getLong	x
getShort	x
getString	x
getTime	x
getTimestamp	x
getAsciiStream	x
getUnicodeStream	x
getObject	x
getBigDecimal	x
getBinaryStream	x
getBlob	x
getClob	x
getNClob	x
getNCharacterStream	x
getNCharacterStream	x
getNString	x
getRef	x
getSQLXML	x
getURL	x
getXML	x

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](http://www.100test.com)