| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ResultSet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/sql/Ref.html)   [**NEXT CLASS**](http://docs.google.com/java/sql/ResultSetMetaData.html) | [**FRAMES**](http://docs.google.com/index.html?java/sql/ResultSet.html)    [**NO FRAMES**](http://docs.google.com/ResultSet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#2jxsxqh) |

## **java.sql**

Interface ResultSet

**All Superinterfaces:** [Wrapper](http://docs.google.com/java/sql/Wrapper.html) **All Known Subinterfaces:** [CachedRowSet](http://docs.google.com/javax/sql/rowset/CachedRowSet.html), [FilteredRowSet](http://docs.google.com/javax/sql/rowset/FilteredRowSet.html), [JdbcRowSet](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html), [JoinRowSet](http://docs.google.com/javax/sql/rowset/JoinRowSet.html), [RowSet](http://docs.google.com/javax/sql/RowSet.html), [SyncResolver](http://docs.google.com/javax/sql/rowset/spi/SyncResolver.html), [WebRowSet](http://docs.google.com/javax/sql/rowset/WebRowSet.html)

public interface **ResultSet**extends [Wrapper](http://docs.google.com/java/sql/Wrapper.html)

A table of data representing a database result set, which is usually generated by executing a statement that queries the database.

A ResultSet object maintains a cursor pointing to its current row of data. Initially the cursor is positioned before the first row. The next method moves the cursor to the next row, and because it returns false when there are no more rows in the ResultSet object, it can be used in a while loop to iterate through the result set.

A default ResultSet object is not updatable and has a cursor that moves forward only. Thus, you can iterate through it only once and only from the first row to the last row. It is possible to produce ResultSet objects that are scrollable and/or updatable. The following code fragment, in which con is a valid Connection object, illustrates how to make a result set that is scrollable and insensitive to updates by others, and that is updatable. See ResultSet fields for other options.

Statement stmt = con.createStatement(  
 ResultSet.TYPE\_SCROLL\_INSENSITIVE,  
 ResultSet.CONCUR\_UPDATABLE);  
 ResultSet rs = stmt.executeQuery("SELECT a, b FROM TABLE2");  
 // rs will be scrollable, will not show changes made by others,  
 // and will be updatable

The ResultSet interface provides *getter* methods (getBoolean, getLong, and so on) for retrieving column values from the current row. Values can be retrieved using either the index number of the column or the name of the column. In general, using the column index will be more efficient. Columns are numbered from 1. For maximum portability, result set columns within each row should be read in left-to-right order, and each column should be read only once.

For the getter methods, a JDBC driver attempts to convert the underlying data to the Java type specified in the getter method and returns a suitable Java value. The JDBC specification has a table showing the allowable mappings from SQL types to Java types that can be used by the ResultSet getter methods.

Column names used as input to getter methods are case insensitive. When a getter method is called with a column name and several columns have the same name, the value of the first matching column will be returned. The column name option is designed to be used when column names are used in the SQL query that generated the result set. For columns that are NOT explicitly named in the query, it is best to use column numbers. If column names are used, the programmer should take care to guarantee that they uniquely refer to the intended columns, which can be assured with the SQL *AS* clause.

A set of updater methods were added to this interface in the JDBC 2.0 API (JavaTM 2 SDK, Standard Edition, version 1.2). The comments regarding parameters to the getter methods also apply to parameters to the updater methods.

The updater methods may be used in two ways:

1. to update a column value in the current row. In a scrollable ResultSet object, the cursor can be moved backwards and forwards, to an absolute position, or to a position relative to the current row. The following code fragment updates the NAME column in the fifth row of the ResultSet object rs and then uses the method updateRow to update the data source table from which rs was derived.  
    rs.absolute(5); // moves the cursor to the fifth row of rs  
    rs.updateString("NAME", "AINSWORTH"); // updates the   
    // NAME column of row 5 to be AINSWORTH  
    rs.updateRow(); // updates the row in the data source
2. to insert column values into the insert row. An updatable ResultSet object has a special row associated with it that serves as a staging area for building a row to be inserted. The following code fragment moves the cursor to the insert row, builds a three-column row, and inserts it into rs and into the data source table using the method insertRow.  
    rs.moveToInsertRow(); // moves cursor to the insert row  
    rs.updateString(1, "AINSWORTH"); // updates the   
    // first column of the insert row to be AINSWORTH  
    rs.updateInt(2,35); // updates the second column to be 35  
    rs.updateBoolean(3, true); // updates the third column to true  
    rs.insertRow();  
    rs.moveToCurrentRow();

A ResultSet object is automatically closed when the Statement object that generated it is closed, re-executed, or used to retrieve the next result from a sequence of multiple results.

The number, types and properties of a ResultSet object's columns are provided by the ResulSetMetaData object returned by the ResultSet.getMetaData method.

**See Also:**[Statement.executeQuery(java.lang.String)](http://docs.google.com/java/sql/Statement.html#executeQuery(java.lang.String)), [Statement.getResultSet()](http://docs.google.com/java/sql/Statement.html#getResultSet()), [ResultSetMetaData](http://docs.google.com/java/sql/ResultSetMetaData.html)

| **Field Summary** | |
| --- | --- |
| static int | [**CLOSE\_CURSORS\_AT\_COMMIT**](http://docs.google.com/java/sql/ResultSet.html#CLOSE_CURSORS_AT_COMMIT)            The constant indicating that open ResultSet objects with this holdability will be closed when the current transaction is commited. |
| static int | [**CONCUR\_READ\_ONLY**](http://docs.google.com/java/sql/ResultSet.html#CONCUR_READ_ONLY)            The constant indicating the concurrency mode for a ResultSet object that may NOT be updated. |
| static int | [**CONCUR\_UPDATABLE**](http://docs.google.com/java/sql/ResultSet.html#CONCUR_UPDATABLE)            The constant indicating the concurrency mode for a ResultSet object that may be updated. |
| static int | [**FETCH\_FORWARD**](http://docs.google.com/java/sql/ResultSet.html#FETCH_FORWARD)            The constant indicating that the rows in a result set will be processed in a forward direction; first-to-last. |
| static int | [**FETCH\_REVERSE**](http://docs.google.com/java/sql/ResultSet.html#FETCH_REVERSE)            The constant indicating that the rows in a result set will be processed in a reverse direction; last-to-first. |
| static int | [**FETCH\_UNKNOWN**](http://docs.google.com/java/sql/ResultSet.html#FETCH_UNKNOWN)            The constant indicating that the order in which rows in a result set will be processed is unknown. |
| static int | [**HOLD\_CURSORS\_OVER\_COMMIT**](http://docs.google.com/java/sql/ResultSet.html#HOLD_CURSORS_OVER_COMMIT)            The constant indicating that open ResultSet objects with this holdability will remain open when the current transaction is commited. |
| static int | [**TYPE\_FORWARD\_ONLY**](http://docs.google.com/java/sql/ResultSet.html#TYPE_FORWARD_ONLY)            The constant indicating the type for a ResultSet object whose cursor may move only forward. |
| static int | [**TYPE\_SCROLL\_INSENSITIVE**](http://docs.google.com/java/sql/ResultSet.html#TYPE_SCROLL_INSENSITIVE)            The constant indicating the type for a ResultSet object that is scrollable but generally not sensitive to changes to the data that underlies the ResultSet. |
| static int | [**TYPE\_SCROLL\_SENSITIVE**](http://docs.google.com/java/sql/ResultSet.html#TYPE_SCROLL_SENSITIVE)            The constant indicating the type for a ResultSet object that is scrollable and generally sensitive to changes to the data that underlies the ResultSet. |

| **Method Summary** | |
| --- | --- |
| boolean | [**absolute**](http://docs.google.com/java/sql/ResultSet.html#absolute(int))(int row)            Moves the cursor to the given row number in this ResultSet object. |
| void | [**afterLast**](http://docs.google.com/java/sql/ResultSet.html#afterLast())()            Moves the cursor to the end of this ResultSet object, just after the last row. |
| void | [**beforeFirst**](http://docs.google.com/java/sql/ResultSet.html#beforeFirst())()            Moves the cursor to the front of this ResultSet object, just before the first row. |
| void | [**cancelRowUpdates**](http://docs.google.com/java/sql/ResultSet.html#cancelRowUpdates())()            Cancels the updates made to the current row in this ResultSet object. |
| void | [**clearWarnings**](http://docs.google.com/java/sql/ResultSet.html#clearWarnings())()            Clears all warnings reported on this ResultSet object. |
| void | [**close**](http://docs.google.com/java/sql/ResultSet.html#close())()            Releases this ResultSet object's database and JDBC resources immediately instead of waiting for this to happen when it is automatically closed. |
| void | [**deleteRow**](http://docs.google.com/java/sql/ResultSet.html#deleteRow())()            Deletes the current row from this ResultSet object and from the underlying database. |
| int | [**findColumn**](http://docs.google.com/java/sql/ResultSet.html#findColumn(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Maps the given ResultSet column label to its ResultSet column index. |
| boolean | [**first**](http://docs.google.com/java/sql/ResultSet.html#first())()            Moves the cursor to the first row in this ResultSet object. |
| [Array](http://docs.google.com/java/sql/Array.html) | [**getArray**](http://docs.google.com/java/sql/ResultSet.html#getArray(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as an Array object in the Java programming language. |
| [Array](http://docs.google.com/java/sql/Array.html) | [**getArray**](http://docs.google.com/java/sql/ResultSet.html#getArray(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as an Array object in the Java programming language. |
| [InputStream](http://docs.google.com/java/io/InputStream.html) | [**getAsciiStream**](http://docs.google.com/java/sql/ResultSet.html#getAsciiStream(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a stream of ASCII characters. |
| [InputStream](http://docs.google.com/java/io/InputStream.html) | [**getAsciiStream**](http://docs.google.com/java/sql/ResultSet.html#getAsciiStream(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a stream of ASCII characters. |
| [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) | [**getBigDecimal**](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.math.BigDecimal with full precision. |
| [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) | [**getBigDecimal**](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(int,%20int))(int columnIndex, int scale)  **Deprecated.** |
| [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) | [**getBigDecimal**](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.math.BigDecimal with full precision. |
| [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) | [**getBigDecimal**](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) columnLabel, int scale)  **Deprecated.** |
| [InputStream](http://docs.google.com/java/io/InputStream.html) | [**getBinaryStream**](http://docs.google.com/java/sql/ResultSet.html#getBinaryStream(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a stream of uninterpreted bytes. |
| [InputStream](http://docs.google.com/java/io/InputStream.html) | [**getBinaryStream**](http://docs.google.com/java/sql/ResultSet.html#getBinaryStream(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a stream of uninterpreted bytes. |
| [Blob](http://docs.google.com/java/sql/Blob.html) | [**getBlob**](http://docs.google.com/java/sql/ResultSet.html#getBlob(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a Blob object in the Java programming language. |
| [Blob](http://docs.google.com/java/sql/Blob.html) | [**getBlob**](http://docs.google.com/java/sql/ResultSet.html#getBlob(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a Blob object in the Java programming language. |
| boolean | [**getBoolean**](http://docs.google.com/java/sql/ResultSet.html#getBoolean(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a boolean in the Java programming language. |
| boolean | [**getBoolean**](http://docs.google.com/java/sql/ResultSet.html#getBoolean(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a boolean in the Java programming language. |
| byte | [**getByte**](http://docs.google.com/java/sql/ResultSet.html#getByte(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a byte in the Java programming language. |
| byte | [**getByte**](http://docs.google.com/java/sql/ResultSet.html#getByte(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a byte in the Java programming language. |
| byte[] | [**getBytes**](http://docs.google.com/java/sql/ResultSet.html#getBytes(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a byte array in the Java programming language. |
| byte[] | [**getBytes**](http://docs.google.com/java/sql/ResultSet.html#getBytes(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a byte array in the Java programming language. |
| [Reader](http://docs.google.com/java/io/Reader.html) | [**getCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#getCharacterStream(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.io.Reader object. |
| [Reader](http://docs.google.com/java/io/Reader.html) | [**getCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#getCharacterStream(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.io.Reader object. |
| [Clob](http://docs.google.com/java/sql/Clob.html) | [**getClob**](http://docs.google.com/java/sql/ResultSet.html#getClob(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a Clob object in the Java programming language. |
| [Clob](http://docs.google.com/java/sql/Clob.html) | [**getClob**](http://docs.google.com/java/sql/ResultSet.html#getClob(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a Clob object in the Java programming language. |
| int | [**getConcurrency**](http://docs.google.com/java/sql/ResultSet.html#getConcurrency())()            Retrieves the concurrency mode of this ResultSet object. |
| [String](http://docs.google.com/java/lang/String.html) | [**getCursorName**](http://docs.google.com/java/sql/ResultSet.html#getCursorName())()            Retrieves the name of the SQL cursor used by this ResultSet object. |
| [Date](http://docs.google.com/java/sql/Date.html) | [**getDate**](http://docs.google.com/java/sql/ResultSet.html#getDate(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Date object in the Java programming language. |
| [Date](http://docs.google.com/java/sql/Date.html) | [**getDate**](http://docs.google.com/java/sql/ResultSet.html#getDate(int,%20java.util.Calendar))(int columnIndex, [Calendar](http://docs.google.com/java/util/Calendar.html) cal)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Date object in the Java programming language. |
| [Date](http://docs.google.com/java/sql/Date.html) | [**getDate**](http://docs.google.com/java/sql/ResultSet.html#getDate(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Date object in the Java programming language. |
| [Date](http://docs.google.com/java/sql/Date.html) | [**getDate**](http://docs.google.com/java/sql/ResultSet.html#getDate(java.lang.String,%20java.util.Calendar))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Calendar](http://docs.google.com/java/util/Calendar.html) cal)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Date object in the Java programming language. |
| double | [**getDouble**](http://docs.google.com/java/sql/ResultSet.html#getDouble(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a double in the Java programming language. |
| double | [**getDouble**](http://docs.google.com/java/sql/ResultSet.html#getDouble(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a double in the Java programming language. |
| int | [**getFetchDirection**](http://docs.google.com/java/sql/ResultSet.html#getFetchDirection())()            Retrieves the fetch direction for this ResultSet object. |
| int | [**getFetchSize**](http://docs.google.com/java/sql/ResultSet.html#getFetchSize())()            Retrieves the fetch size for this ResultSet object. |
| float | [**getFloat**](http://docs.google.com/java/sql/ResultSet.html#getFloat(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a float in the Java programming language. |
| float | [**getFloat**](http://docs.google.com/java/sql/ResultSet.html#getFloat(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a float in the Java programming language. |
| int | [**getHoldability**](http://docs.google.com/java/sql/ResultSet.html#getHoldability())()            Retrieves the holdability of this ResultSet object |
| int | [**getInt**](http://docs.google.com/java/sql/ResultSet.html#getInt(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as an int in the Java programming language. |
| int | [**getInt**](http://docs.google.com/java/sql/ResultSet.html#getInt(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as an int in the Java programming language. |
| long | [**getLong**](http://docs.google.com/java/sql/ResultSet.html#getLong(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a long in the Java programming language. |
| long | [**getLong**](http://docs.google.com/java/sql/ResultSet.html#getLong(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a long in the Java programming language. |
| [ResultSetMetaData](http://docs.google.com/java/sql/ResultSetMetaData.html) | [**getMetaData**](http://docs.google.com/java/sql/ResultSet.html#getMetaData())()            Retrieves the number, types and properties of this ResultSet object's columns. |
| [Reader](http://docs.google.com/java/io/Reader.html) | [**getNCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#getNCharacterStream(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.io.Reader object. |
| [Reader](http://docs.google.com/java/io/Reader.html) | [**getNCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#getNCharacterStream(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.io.Reader object. |
| [NClob](http://docs.google.com/java/sql/NClob.html) | [**getNClob**](http://docs.google.com/java/sql/ResultSet.html#getNClob(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a NClob object in the Java programming language. |
| [NClob](http://docs.google.com/java/sql/NClob.html) | [**getNClob**](http://docs.google.com/java/sql/ResultSet.html#getNClob(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a NClob object in the Java programming language. |
| [String](http://docs.google.com/java/lang/String.html) | [**getNString**](http://docs.google.com/java/sql/ResultSet.html#getNString(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a String in the Java programming language. |
| [String](http://docs.google.com/java/lang/String.html) | [**getNString**](http://docs.google.com/java/sql/ResultSet.html#getNString(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a String in the Java programming language. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getObject**](http://docs.google.com/java/sql/ResultSet.html#getObject(int))(int columnIndex)            Gets the value of the designated column in the current row of this ResultSet object as an Object in the Java programming language. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getObject**](http://docs.google.com/java/sql/ResultSet.html#getObject(int,%20java.util.Map))(int columnIndex, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)            Retrieves the value of the designated column in the current row of this ResultSet object as an Object in the Java programming language. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getObject**](http://docs.google.com/java/sql/ResultSet.html#getObject(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Gets the value of the designated column in the current row of this ResultSet object as an Object in the Java programming language. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getObject**](http://docs.google.com/java/sql/ResultSet.html#getObject(java.lang.String,%20java.util.Map))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)            Retrieves the value of the designated column in the current row of this ResultSet object as an Object in the Java programming language. |
| [Ref](http://docs.google.com/java/sql/Ref.html) | [**getRef**](http://docs.google.com/java/sql/ResultSet.html#getRef(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a Ref object in the Java programming language. |
| [Ref](http://docs.google.com/java/sql/Ref.html) | [**getRef**](http://docs.google.com/java/sql/ResultSet.html#getRef(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a Ref object in the Java programming language. |
| int | [**getRow**](http://docs.google.com/java/sql/ResultSet.html#getRow())()            Retrieves the current row number. |
| [RowId](http://docs.google.com/java/sql/RowId.html) | [**getRowId**](http://docs.google.com/java/sql/ResultSet.html#getRowId(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.RowId object in the Java programming language. |
| [RowId](http://docs.google.com/java/sql/RowId.html) | [**getRowId**](http://docs.google.com/java/sql/ResultSet.html#getRowId(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.RowId object in the Java programming language. |
| short | [**getShort**](http://docs.google.com/java/sql/ResultSet.html#getShort(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a short in the Java programming language. |
| short | [**getShort**](http://docs.google.com/java/sql/ResultSet.html#getShort(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a short in the Java programming language. |
| [SQLXML](http://docs.google.com/java/sql/SQLXML.html) | [**getSQLXML**](http://docs.google.com/java/sql/ResultSet.html#getSQLXML(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet as a java.sql.SQLXML object in the Java programming language. |
| [SQLXML](http://docs.google.com/java/sql/SQLXML.html) | [**getSQLXML**](http://docs.google.com/java/sql/ResultSet.html#getSQLXML(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet as a java.sql.SQLXML object in the Java programming language. |
| [Statement](http://docs.google.com/java/sql/Statement.html) | [**getStatement**](http://docs.google.com/java/sql/ResultSet.html#getStatement())()            Retrieves the Statement object that produced this ResultSet object. |
| [String](http://docs.google.com/java/lang/String.html) | [**getString**](http://docs.google.com/java/sql/ResultSet.html#getString(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a String in the Java programming language. |
| [String](http://docs.google.com/java/lang/String.html) | [**getString**](http://docs.google.com/java/sql/ResultSet.html#getString(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a String in the Java programming language. |
| [Time](http://docs.google.com/java/sql/Time.html) | [**getTime**](http://docs.google.com/java/sql/ResultSet.html#getTime(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Time object in the Java programming language. |
| [Time](http://docs.google.com/java/sql/Time.html) | [**getTime**](http://docs.google.com/java/sql/ResultSet.html#getTime(int,%20java.util.Calendar))(int columnIndex, [Calendar](http://docs.google.com/java/util/Calendar.html) cal)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Time object in the Java programming language. |
| [Time](http://docs.google.com/java/sql/Time.html) | [**getTime**](http://docs.google.com/java/sql/ResultSet.html#getTime(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Time object in the Java programming language. |
| [Time](http://docs.google.com/java/sql/Time.html) | [**getTime**](http://docs.google.com/java/sql/ResultSet.html#getTime(java.lang.String,%20java.util.Calendar))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Calendar](http://docs.google.com/java/util/Calendar.html) cal)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Time object in the Java programming language. |
| [Timestamp](http://docs.google.com/java/sql/Timestamp.html) | [**getTimestamp**](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Timestamp object in the Java programming language. |
| [Timestamp](http://docs.google.com/java/sql/Timestamp.html) | [**getTimestamp**](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(int,%20java.util.Calendar))(int columnIndex, [Calendar](http://docs.google.com/java/util/Calendar.html) cal)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Timestamp object in the Java programming language. |
| [Timestamp](http://docs.google.com/java/sql/Timestamp.html) | [**getTimestamp**](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Timestamp object in the Java programming language. |
| [Timestamp](http://docs.google.com/java/sql/Timestamp.html) | [**getTimestamp**](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(java.lang.String,%20java.util.Calendar))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Calendar](http://docs.google.com/java/util/Calendar.html) cal)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Timestamp object in the Java programming language. |
| int | [**getType**](http://docs.google.com/java/sql/ResultSet.html#getType())()            Retrieves the type of this ResultSet object. |
| [InputStream](http://docs.google.com/java/io/InputStream.html) | [**getUnicodeStream**](http://docs.google.com/java/sql/ResultSet.html#getUnicodeStream(int))(int columnIndex)  **Deprecated.** *use getCharacterStream in place of getUnicodeStream* |
| [InputStream](http://docs.google.com/java/io/InputStream.html) | [**getUnicodeStream**](http://docs.google.com/java/sql/ResultSet.html#getUnicodeStream(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)  **Deprecated.** *use getCharacterStream instead* |
| [URL](http://docs.google.com/java/net/URL.html) | [**getURL**](http://docs.google.com/java/sql/ResultSet.html#getURL(int))(int columnIndex)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.net.URL object in the Java programming language. |
| [URL](http://docs.google.com/java/net/URL.html) | [**getURL**](http://docs.google.com/java/sql/ResultSet.html#getURL(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Retrieves the value of the designated column in the current row of this ResultSet object as a java.net.URL object in the Java programming language. |
| [SQLWarning](http://docs.google.com/java/sql/SQLWarning.html) | [**getWarnings**](http://docs.google.com/java/sql/ResultSet.html#getWarnings())()            Retrieves the first warning reported by calls on this ResultSet object. |
| void | [**insertRow**](http://docs.google.com/java/sql/ResultSet.html#insertRow())()            Inserts the contents of the insert row into this ResultSet object and into the database. |
| boolean | [**isAfterLast**](http://docs.google.com/java/sql/ResultSet.html#isAfterLast())()            Retrieves whether the cursor is after the last row in this ResultSet object. |
| boolean | [**isBeforeFirst**](http://docs.google.com/java/sql/ResultSet.html#isBeforeFirst())()            Retrieves whether the cursor is before the first row in this ResultSet object. |
| boolean | [**isClosed**](http://docs.google.com/java/sql/ResultSet.html#isClosed())()            Retrieves whether this ResultSet object has been closed. |
| boolean | [**isFirst**](http://docs.google.com/java/sql/ResultSet.html#isFirst())()            Retrieves whether the cursor is on the first row of this ResultSet object. |
| boolean | [**isLast**](http://docs.google.com/java/sql/ResultSet.html#isLast())()            Retrieves whether the cursor is on the last row of this ResultSet object. |
| boolean | [**last**](http://docs.google.com/java/sql/ResultSet.html#last())()            Moves the cursor to the last row in this ResultSet object. |
| void | [**moveToCurrentRow**](http://docs.google.com/java/sql/ResultSet.html#moveToCurrentRow())()            Moves the cursor to the remembered cursor position, usually the current row. |
| void | [**moveToInsertRow**](http://docs.google.com/java/sql/ResultSet.html#moveToInsertRow())()            Moves the cursor to the insert row. |
| boolean | [**next**](http://docs.google.com/java/sql/ResultSet.html#next())()            Moves the cursor froward one row from its current position. |
| boolean | [**previous**](http://docs.google.com/java/sql/ResultSet.html#previous())()            Moves the cursor to the previous row in this ResultSet object. |
| void | [**refreshRow**](http://docs.google.com/java/sql/ResultSet.html#refreshRow())()            Refreshes the current row with its most recent value in the database. |
| boolean | [**relative**](http://docs.google.com/java/sql/ResultSet.html#relative(int))(int rows)            Moves the cursor a relative number of rows, either positive or negative. |
| boolean | [**rowDeleted**](http://docs.google.com/java/sql/ResultSet.html#rowDeleted())()            Retrieves whether a row has been deleted. |
| boolean | [**rowInserted**](http://docs.google.com/java/sql/ResultSet.html#rowInserted())()            Retrieves whether the current row has had an insertion. |
| boolean | [**rowUpdated**](http://docs.google.com/java/sql/ResultSet.html#rowUpdated())()            Retrieves whether the current row has been updated. |
| void | [**setFetchDirection**](http://docs.google.com/java/sql/ResultSet.html#setFetchDirection(int))(int direction)            Gives a hint as to the direction in which the rows in this ResultSet object will be processed. |
| void | [**setFetchSize**](http://docs.google.com/java/sql/ResultSet.html#setFetchSize(int))(int rows)            Gives the JDBC driver a hint as to the number of rows that should be fetched from the database when more rows are needed for this ResultSet object. |
| void | [**updateArray**](http://docs.google.com/java/sql/ResultSet.html#updateArray(int,%20java.sql.Array))(int columnIndex, [Array](http://docs.google.com/java/sql/Array.html) x)            Updates the designated column with a java.sql.Array value. |
| void | [**updateArray**](http://docs.google.com/java/sql/ResultSet.html#updateArray(java.lang.String,%20java.sql.Array))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Array](http://docs.google.com/java/sql/Array.html) x)            Updates the designated column with a java.sql.Array value. |
| void | [**updateAsciiStream**](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(int,%20java.io.InputStream))(int columnIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) x)            Updates the designated column with an ascii stream value. |
| void | [**updateAsciiStream**](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(int,%20java.io.InputStream,%20int))(int columnIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) x, int length)            Updates the designated column with an ascii stream value, which will have the specified number of bytes. |
| void | [**updateAsciiStream**](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(int,%20java.io.InputStream,%20long))(int columnIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) x, long length)            Updates the designated column with an ascii stream value, which will have the specified number of bytes. |
| void | [**updateAsciiStream**](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(java.lang.String,%20java.io.InputStream))([String](http://docs.google.com/java/lang/String.html) columnLabel, [InputStream](http://docs.google.com/java/io/InputStream.html) x)            Updates the designated column with an ascii stream value. |
| void | [**updateAsciiStream**](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(java.lang.String,%20java.io.InputStream,%20int))([String](http://docs.google.com/java/lang/String.html) columnLabel, [InputStream](http://docs.google.com/java/io/InputStream.html) x, int length)            Updates the designated column with an ascii stream value, which will have the specified number of bytes. |
| void | [**updateAsciiStream**](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(java.lang.String,%20java.io.InputStream,%20long))([String](http://docs.google.com/java/lang/String.html) columnLabel, [InputStream](http://docs.google.com/java/io/InputStream.html) x, long length)            Updates the designated column with an ascii stream value, which will have the specified number of bytes. |
| void | [**updateBigDecimal**](http://docs.google.com/java/sql/ResultSet.html#updateBigDecimal(int,%20java.math.BigDecimal))(int columnIndex, [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) x)            Updates the designated column with a java.math.BigDecimal value. |
| void | [**updateBigDecimal**](http://docs.google.com/java/sql/ResultSet.html#updateBigDecimal(java.lang.String,%20java.math.BigDecimal))([String](http://docs.google.com/java/lang/String.html) columnLabel, [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) x)            Updates the designated column with a java.sql.BigDecimal value. |
| void | [**updateBinaryStream**](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(int,%20java.io.InputStream))(int columnIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) x)            Updates the designated column with a binary stream value. |
| void | [**updateBinaryStream**](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(int,%20java.io.InputStream,%20int))(int columnIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) x, int length)            Updates the designated column with a binary stream value, which will have the specified number of bytes. |
| void | [**updateBinaryStream**](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(int,%20java.io.InputStream,%20long))(int columnIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) x, long length)            Updates the designated column with a binary stream value, which will have the specified number of bytes. |
| void | [**updateBinaryStream**](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(java.lang.String,%20java.io.InputStream))([String](http://docs.google.com/java/lang/String.html) columnLabel, [InputStream](http://docs.google.com/java/io/InputStream.html) x)            Updates the designated column with a binary stream value. |
| void | [**updateBinaryStream**](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(java.lang.String,%20java.io.InputStream,%20int))([String](http://docs.google.com/java/lang/String.html) columnLabel, [InputStream](http://docs.google.com/java/io/InputStream.html) x, int length)            Updates the designated column with a binary stream value, which will have the specified number of bytes. |
| void | [**updateBinaryStream**](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(java.lang.String,%20java.io.InputStream,%20long))([String](http://docs.google.com/java/lang/String.html) columnLabel, [InputStream](http://docs.google.com/java/io/InputStream.html) x, long length)            Updates the designated column with a binary stream value, which will have the specified number of bytes. |
| void | [**updateBlob**](http://docs.google.com/java/sql/ResultSet.html#updateBlob(int,%20java.sql.Blob))(int columnIndex, [Blob](http://docs.google.com/java/sql/Blob.html) x)            Updates the designated column with a java.sql.Blob value. |
| void | [**updateBlob**](http://docs.google.com/java/sql/ResultSet.html#updateBlob(int,%20java.io.InputStream))(int columnIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream)            Updates the designated column using the given input stream. |
| void | [**updateBlob**](http://docs.google.com/java/sql/ResultSet.html#updateBlob(int,%20java.io.InputStream,%20long))(int columnIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream, long length)            Updates the designated column using the given input stream, which will have the specified number of bytes. |
| void | [**updateBlob**](http://docs.google.com/java/sql/ResultSet.html#updateBlob(java.lang.String,%20java.sql.Blob))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Blob](http://docs.google.com/java/sql/Blob.html) x)            Updates the designated column with a java.sql.Blob value. |
| void | [**updateBlob**](http://docs.google.com/java/sql/ResultSet.html#updateBlob(java.lang.String,%20java.io.InputStream))([String](http://docs.google.com/java/lang/String.html) columnLabel, [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream)            Updates the designated column using the given input stream. |
| void | [**updateBlob**](http://docs.google.com/java/sql/ResultSet.html#updateBlob(java.lang.String,%20java.io.InputStream,%20long))([String](http://docs.google.com/java/lang/String.html) columnLabel, [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream, long length)            Updates the designated column using the given input stream, which will have the specified number of bytes. |
| void | [**updateBoolean**](http://docs.google.com/java/sql/ResultSet.html#updateBoolean(int,%20boolean))(int columnIndex, boolean x)            Updates the designated column with a boolean value. |
| void | [**updateBoolean**](http://docs.google.com/java/sql/ResultSet.html#updateBoolean(java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) columnLabel, boolean x)            Updates the designated column with a boolean value. |
| void | [**updateByte**](http://docs.google.com/java/sql/ResultSet.html#updateByte(int,%20byte))(int columnIndex, byte x)            Updates the designated column with a byte value. |
| void | [**updateByte**](http://docs.google.com/java/sql/ResultSet.html#updateByte(java.lang.String,%20byte))([String](http://docs.google.com/java/lang/String.html) columnLabel, byte x)            Updates the designated column with a byte value. |
| void | [**updateBytes**](http://docs.google.com/java/sql/ResultSet.html#updateBytes(int,%20byte%5B%5D))(int columnIndex, byte[] x)            Updates the designated column with a byte array value. |
| void | [**updateBytes**](http://docs.google.com/java/sql/ResultSet.html#updateBytes(java.lang.String,%20byte%5B%5D))([String](http://docs.google.com/java/lang/String.html) columnLabel, byte[] x)            Updates the designated column with a byte array value. |
| void | [**updateCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(int,%20java.io.Reader))(int columnIndex, [Reader](http://docs.google.com/java/io/Reader.html) x)            Updates the designated column with a character stream value. |
| void | [**updateCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(int,%20java.io.Reader,%20int))(int columnIndex, [Reader](http://docs.google.com/java/io/Reader.html) x, int length)            Updates the designated column with a character stream value, which will have the specified number of bytes. |
| void | [**updateCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(int,%20java.io.Reader,%20long))(int columnIndex, [Reader](http://docs.google.com/java/io/Reader.html) x, long length)            Updates the designated column with a character stream value, which will have the specified number of bytes. |
| void | [**updateCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(java.lang.String,%20java.io.Reader))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Reader](http://docs.google.com/java/io/Reader.html) reader)            Updates the designated column with a character stream value. |
| void | [**updateCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(java.lang.String,%20java.io.Reader,%20int))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Reader](http://docs.google.com/java/io/Reader.html) reader, int length)            Updates the designated column with a character stream value, which will have the specified number of bytes. |
| void | [**updateCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(java.lang.String,%20java.io.Reader,%20long))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Reader](http://docs.google.com/java/io/Reader.html) reader, long length)            Updates the designated column with a character stream value, which will have the specified number of bytes. |
| void | [**updateClob**](http://docs.google.com/java/sql/ResultSet.html#updateClob(int,%20java.sql.Clob))(int columnIndex, [Clob](http://docs.google.com/java/sql/Clob.html) x)            Updates the designated column with a java.sql.Clob value. |
| void | [**updateClob**](http://docs.google.com/java/sql/ResultSet.html#updateClob(int,%20java.io.Reader))(int columnIndex, [Reader](http://docs.google.com/java/io/Reader.html) reader)            Updates the designated column using the given Reader object. |
| void | [**updateClob**](http://docs.google.com/java/sql/ResultSet.html#updateClob(int,%20java.io.Reader,%20long))(int columnIndex, [Reader](http://docs.google.com/java/io/Reader.html) reader, long length)            Updates the designated column using the given Reader object, which is the given number of characters long. |
| void | [**updateClob**](http://docs.google.com/java/sql/ResultSet.html#updateClob(java.lang.String,%20java.sql.Clob))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Clob](http://docs.google.com/java/sql/Clob.html) x)            Updates the designated column with a java.sql.Clob value. |
| void | [**updateClob**](http://docs.google.com/java/sql/ResultSet.html#updateClob(java.lang.String,%20java.io.Reader))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Reader](http://docs.google.com/java/io/Reader.html) reader)            Updates the designated column using the given Reader object. |
| void | [**updateClob**](http://docs.google.com/java/sql/ResultSet.html#updateClob(java.lang.String,%20java.io.Reader,%20long))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Reader](http://docs.google.com/java/io/Reader.html) reader, long length)            Updates the designated column using the given Reader object, which is the given number of characters long. |
| void | [**updateDate**](http://docs.google.com/java/sql/ResultSet.html#updateDate(int,%20java.sql.Date))(int columnIndex, [Date](http://docs.google.com/java/sql/Date.html) x)            Updates the designated column with a java.sql.Date value. |
| void | [**updateDate**](http://docs.google.com/java/sql/ResultSet.html#updateDate(java.lang.String,%20java.sql.Date))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Date](http://docs.google.com/java/sql/Date.html) x)            Updates the designated column with a java.sql.Date value. |
| void | [**updateDouble**](http://docs.google.com/java/sql/ResultSet.html#updateDouble(int,%20double))(int columnIndex, double x)            Updates the designated column with a double value. |
| void | [**updateDouble**](http://docs.google.com/java/sql/ResultSet.html#updateDouble(java.lang.String,%20double))([String](http://docs.google.com/java/lang/String.html) columnLabel, double x)            Updates the designated column with a double value. |
| void | [**updateFloat**](http://docs.google.com/java/sql/ResultSet.html#updateFloat(int,%20float))(int columnIndex, float x)            Updates the designated column with a float value. |
| void | [**updateFloat**](http://docs.google.com/java/sql/ResultSet.html#updateFloat(java.lang.String,%20float))([String](http://docs.google.com/java/lang/String.html) columnLabel, float x)            Updates the designated column with a float value. |
| void | [**updateInt**](http://docs.google.com/java/sql/ResultSet.html#updateInt(int,%20int))(int columnIndex, int x)            Updates the designated column with an int value. |
| void | [**updateInt**](http://docs.google.com/java/sql/ResultSet.html#updateInt(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) columnLabel, int x)            Updates the designated column with an int value. |
| void | [**updateLong**](http://docs.google.com/java/sql/ResultSet.html#updateLong(int,%20long))(int columnIndex, long x)            Updates the designated column with a long value. |
| void | [**updateLong**](http://docs.google.com/java/sql/ResultSet.html#updateLong(java.lang.String,%20long))([String](http://docs.google.com/java/lang/String.html) columnLabel, long x)            Updates the designated column with a long value. |
| void | [**updateNCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#updateNCharacterStream(int,%20java.io.Reader))(int columnIndex, [Reader](http://docs.google.com/java/io/Reader.html) x)            Updates the designated column with a character stream value. |
| void | [**updateNCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#updateNCharacterStream(int,%20java.io.Reader,%20long))(int columnIndex, [Reader](http://docs.google.com/java/io/Reader.html) x, long length)            Updates the designated column with a character stream value, which will have the specified number of bytes. |
| void | [**updateNCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#updateNCharacterStream(java.lang.String,%20java.io.Reader))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Reader](http://docs.google.com/java/io/Reader.html) reader)            Updates the designated column with a character stream value. |
| void | [**updateNCharacterStream**](http://docs.google.com/java/sql/ResultSet.html#updateNCharacterStream(java.lang.String,%20java.io.Reader,%20long))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Reader](http://docs.google.com/java/io/Reader.html) reader, long length)            Updates the designated column with a character stream value, which will have the specified number of bytes. |
| void | [**updateNClob**](http://docs.google.com/java/sql/ResultSet.html#updateNClob(int,%20java.sql.NClob))(int columnIndex, [NClob](http://docs.google.com/java/sql/NClob.html) nClob)            Updates the designated column with a java.sql.NClob value. |
| void | [**updateNClob**](http://docs.google.com/java/sql/ResultSet.html#updateNClob(int,%20java.io.Reader))(int columnIndex, [Reader](http://docs.google.com/java/io/Reader.html) reader)            Updates the designated column using the given Reader The data will be read from the stream as needed until end-of-stream is reached. |
| void | [**updateNClob**](http://docs.google.com/java/sql/ResultSet.html#updateNClob(int,%20java.io.Reader,%20long))(int columnIndex, [Reader](http://docs.google.com/java/io/Reader.html) reader, long length)            Updates the designated column using the given Reader object, which is the given number of characters long. |
| void | [**updateNClob**](http://docs.google.com/java/sql/ResultSet.html#updateNClob(java.lang.String,%20java.sql.NClob))([String](http://docs.google.com/java/lang/String.html) columnLabel, [NClob](http://docs.google.com/java/sql/NClob.html) nClob)            Updates the designated column with a java.sql.NClob value. |
| void | [**updateNClob**](http://docs.google.com/java/sql/ResultSet.html#updateNClob(java.lang.String,%20java.io.Reader))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Reader](http://docs.google.com/java/io/Reader.html) reader)            Updates the designated column using the given Reader object. |
| void | [**updateNClob**](http://docs.google.com/java/sql/ResultSet.html#updateNClob(java.lang.String,%20java.io.Reader,%20long))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Reader](http://docs.google.com/java/io/Reader.html) reader, long length)            Updates the designated column using the given Reader object, which is the given number of characters long. |
| void | [**updateNString**](http://docs.google.com/java/sql/ResultSet.html#updateNString(int,%20java.lang.String))(int columnIndex, [String](http://docs.google.com/java/lang/String.html) nString)            Updates the designated column with a String value. |
| void | [**updateNString**](http://docs.google.com/java/sql/ResultSet.html#updateNString(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel, [String](http://docs.google.com/java/lang/String.html) nString)            Updates the designated column with a String value. |
| void | [**updateNull**](http://docs.google.com/java/sql/ResultSet.html#updateNull(int))(int columnIndex)            Updates the designated column with a null value. |
| void | [**updateNull**](http://docs.google.com/java/sql/ResultSet.html#updateNull(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel)            Updates the designated column with a null value. |
| void | [**updateObject**](http://docs.google.com/java/sql/ResultSet.html#updateObject(int,%20java.lang.Object))(int columnIndex, [Object](http://docs.google.com/java/lang/Object.html) x)            Updates the designated column with an Object value. |
| void | [**updateObject**](http://docs.google.com/java/sql/ResultSet.html#updateObject(int,%20java.lang.Object,%20int))(int columnIndex, [Object](http://docs.google.com/java/lang/Object.html) x, int scaleOrLength)            Updates the designated column with an Object value. |
| void | [**updateObject**](http://docs.google.com/java/sql/ResultSet.html#updateObject(java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Object](http://docs.google.com/java/lang/Object.html) x)            Updates the designated column with an Object value. |
| void | [**updateObject**](http://docs.google.com/java/sql/ResultSet.html#updateObject(java.lang.String,%20java.lang.Object,%20int))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Object](http://docs.google.com/java/lang/Object.html) x, int scaleOrLength)            Updates the designated column with an Object value. |
| void | [**updateRef**](http://docs.google.com/java/sql/ResultSet.html#updateRef(int,%20java.sql.Ref))(int columnIndex, [Ref](http://docs.google.com/java/sql/Ref.html) x)            Updates the designated column with a java.sql.Ref value. |
| void | [**updateRef**](http://docs.google.com/java/sql/ResultSet.html#updateRef(java.lang.String,%20java.sql.Ref))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Ref](http://docs.google.com/java/sql/Ref.html) x)            Updates the designated column with a java.sql.Ref value. |
| void | [**updateRow**](http://docs.google.com/java/sql/ResultSet.html#updateRow())()            Updates the underlying database with the new contents of the current row of this ResultSet object. |
| void | [**updateRowId**](http://docs.google.com/java/sql/ResultSet.html#updateRowId(int,%20java.sql.RowId))(int columnIndex, [RowId](http://docs.google.com/java/sql/RowId.html) x)            Updates the designated column with a RowId value. |
| void | [**updateRowId**](http://docs.google.com/java/sql/ResultSet.html#updateRowId(java.lang.String,%20java.sql.RowId))([String](http://docs.google.com/java/lang/String.html) columnLabel, [RowId](http://docs.google.com/java/sql/RowId.html) x)            Updates the designated column with a RowId value. |
| void | [**updateShort**](http://docs.google.com/java/sql/ResultSet.html#updateShort(int,%20short))(int columnIndex, short x)            Updates the designated column with a short value. |
| void | [**updateShort**](http://docs.google.com/java/sql/ResultSet.html#updateShort(java.lang.String,%20short))([String](http://docs.google.com/java/lang/String.html) columnLabel, short x)            Updates the designated column with a short value. |
| void | [**updateSQLXML**](http://docs.google.com/java/sql/ResultSet.html#updateSQLXML(int,%20java.sql.SQLXML))(int columnIndex, [SQLXML](http://docs.google.com/java/sql/SQLXML.html) xmlObject)            Updates the designated column with a java.sql.SQLXML value. |
| void | [**updateSQLXML**](http://docs.google.com/java/sql/ResultSet.html#updateSQLXML(java.lang.String,%20java.sql.SQLXML))([String](http://docs.google.com/java/lang/String.html) columnLabel, [SQLXML](http://docs.google.com/java/sql/SQLXML.html) xmlObject)            Updates the designated column with a java.sql.SQLXML value. |
| void | [**updateString**](http://docs.google.com/java/sql/ResultSet.html#updateString(int,%20java.lang.String))(int columnIndex, [String](http://docs.google.com/java/lang/String.html) x)            Updates the designated column with a String value. |
| void | [**updateString**](http://docs.google.com/java/sql/ResultSet.html#updateString(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnLabel, [String](http://docs.google.com/java/lang/String.html) x)            Updates the designated column with a String value. |
| void | [**updateTime**](http://docs.google.com/java/sql/ResultSet.html#updateTime(int,%20java.sql.Time))(int columnIndex, [Time](http://docs.google.com/java/sql/Time.html) x)            Updates the designated column with a java.sql.Time value. |
| void | [**updateTime**](http://docs.google.com/java/sql/ResultSet.html#updateTime(java.lang.String,%20java.sql.Time))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Time](http://docs.google.com/java/sql/Time.html) x)            Updates the designated column with a java.sql.Time value. |
| void | [**updateTimestamp**](http://docs.google.com/java/sql/ResultSet.html#updateTimestamp(int,%20java.sql.Timestamp))(int columnIndex, [Timestamp](http://docs.google.com/java/sql/Timestamp.html) x)            Updates the designated column with a java.sql.Timestamp value. |
| void | [**updateTimestamp**](http://docs.google.com/java/sql/ResultSet.html#updateTimestamp(java.lang.String,%20java.sql.Timestamp))([String](http://docs.google.com/java/lang/String.html) columnLabel, [Timestamp](http://docs.google.com/java/sql/Timestamp.html) x)            Updates the designated column with a java.sql.Timestamp value. |
| boolean | [**wasNull**](http://docs.google.com/java/sql/ResultSet.html#wasNull())()            Reports whether the last column read had a value of SQL NULL. |

| **Methods inherited from interface java.sql.**[**Wrapper**](http://docs.google.com/java/sql/Wrapper.html) |
| --- |
| [isWrapperFor](http://docs.google.com/java/sql/Wrapper.html#isWrapperFor(java.lang.Class)), [unwrap](http://docs.google.com/java/sql/Wrapper.html#unwrap(java.lang.Class)) |

| **Field Detail** |
| --- |

### FETCH\_FORWARD

static final int **FETCH\_FORWARD**

The constant indicating that the rows in a result set will be processed in a forward direction; first-to-last. This constant is used by the method setFetchDirection as a hint to the driver, which the driver may ignore.

**Since:** 1.2 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.sql.ResultSet.FETCH_FORWARD)

### FETCH\_REVERSE

static final int **FETCH\_REVERSE**

The constant indicating that the rows in a result set will be processed in a reverse direction; last-to-first. This constant is used by the method setFetchDirection as a hint to the driver, which the driver may ignore.

**Since:** 1.2 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.sql.ResultSet.FETCH_REVERSE)

### FETCH\_UNKNOWN

static final int **FETCH\_UNKNOWN**

The constant indicating that the order in which rows in a result set will be processed is unknown. This constant is used by the method setFetchDirection as a hint to the driver, which the driver may ignore.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.sql.ResultSet.FETCH_UNKNOWN)

### TYPE\_FORWARD\_ONLY

static final int **TYPE\_FORWARD\_ONLY**

The constant indicating the type for a ResultSet object whose cursor may move only forward.

**Since:** 1.2 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.sql.ResultSet.TYPE_FORWARD_ONLY)

### TYPE\_SCROLL\_INSENSITIVE

static final int **TYPE\_SCROLL\_INSENSITIVE**

The constant indicating the type for a ResultSet object that is scrollable but generally not sensitive to changes to the data that underlies the ResultSet.

**Since:** 1.2 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.sql.ResultSet.TYPE_SCROLL_INSENSITIVE)

### TYPE\_SCROLL\_SENSITIVE

static final int **TYPE\_SCROLL\_SENSITIVE**

The constant indicating the type for a ResultSet object that is scrollable and generally sensitive to changes to the data that underlies the ResultSet.

**Since:** 1.2 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.sql.ResultSet.TYPE_SCROLL_SENSITIVE)

### CONCUR\_READ\_ONLY

static final int **CONCUR\_READ\_ONLY**

The constant indicating the concurrency mode for a ResultSet object that may NOT be updated.

**Since:** 1.2 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.sql.ResultSet.CONCUR_READ_ONLY)

### CONCUR\_UPDATABLE

static final int **CONCUR\_UPDATABLE**

The constant indicating the concurrency mode for a ResultSet object that may be updated.

**Since:** 1.2 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.sql.ResultSet.CONCUR_UPDATABLE)

### HOLD\_CURSORS\_OVER\_COMMIT

static final int **HOLD\_CURSORS\_OVER\_COMMIT**

The constant indicating that open ResultSet objects with this holdability will remain open when the current transaction is commited.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.sql.ResultSet.HOLD_CURSORS_OVER_COMMIT)

### CLOSE\_CURSORS\_AT\_COMMIT

static final int **CLOSE\_CURSORS\_AT\_COMMIT**

The constant indicating that open ResultSet objects with this holdability will be closed when the current transaction is commited.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.sql.ResultSet.CLOSE_CURSORS_AT_COMMIT)

| **Method Detail** |
| --- |

### next

boolean **next**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Moves the cursor froward one row from its current position. A ResultSet cursor is initially positioned before the first row; the first call to the method next makes the first row the current row; the second call makes the second row the current row, and so on.

When a call to the next method returns false, the cursor is positioned after the last row. Any invocation of a ResultSet method which requires a current row will result in a SQLException being thrown. If the result set type is TYPE\_FORWARD\_ONLY, it is vendor specified whether their JDBC driver implementation will return false or throw an SQLException on a subsequent call to next.

If an input stream is open for the current row, a call to the method next will implicitly close it. A ResultSet object's warning chain is cleared when a new row is read.

**Returns:**true if the new current row is valid; false if there are no more rows **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set

### close

void **close**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Releases this ResultSet object's database and JDBC resources immediately instead of waiting for this to happen when it is automatically closed.

The closing of a ResultSet object does **not** close the Blob, Clob or NClob objects created by the ResultSet. Blob, Clob or NClob objects remain valid for at least the duration of the transaction in which they are creataed, unless their free method is invoked.

When a ResultSet is closed, any ResultSetMetaData instances that were created by calling the getMetaData method remain accessible.

**Note:** A ResultSet object is automatically closed by the Statement object that generated it when that Statement object is closed, re-executed, or is used to retrieve the next result from a sequence of multiple results.

Calling the method close on a ResultSet object that is already closed is a no-op.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### wasNull

boolean **wasNull**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Reports whether the last column read had a value of SQL NULL. Note that you must first call one of the getter methods on a column to try to read its value and then call the method wasNull to see if the value read was SQL NULL.

**Returns:**true if the last column value read was SQL NULL and false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set

### getString

[String](http://docs.google.com/java/lang/String.html) **getString**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a String in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getBoolean

boolean **getBoolean**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a boolean in the Java programming language.

If the designated column has a datatype of CHAR or VARCHAR and contains a "0" or has a datatype of BIT, TINYINT, SMALLINT, INTEGER or BIGINT and contains a 0, a value of false is returned. If the designated column has a datatype of CHAR or VARCHAR and contains a "1" or has a datatype of BIT, TINYINT, SMALLINT, INTEGER or BIGINT and contains a 1, a value of true is returned.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is false **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getByte

byte **getByte**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a byte in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getShort

short **getShort**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a short in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getInt

int **getInt**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as an int in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getLong

long **getLong**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a long in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getFloat

float **getFloat**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a float in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getDouble

double **getDouble**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a double in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getBigDecimal

[BigDecimal](http://docs.google.com/java/math/BigDecimal.html) **getBigDecimal**(int columnIndex,  
 int scale)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

**Deprecated.**

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.BigDecimal in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...scale - the number of digits to the right of the decimal point **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method

### getBytes

byte[] **getBytes**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a byte array in the Java programming language. The bytes represent the raw values returned by the driver.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getDate

[Date](http://docs.google.com/java/sql/Date.html) **getDate**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Date object in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getTime

[Time](http://docs.google.com/java/sql/Time.html) **getTime**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Time object in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getTimestamp

[Timestamp](http://docs.google.com/java/sql/Timestamp.html) **getTimestamp**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Timestamp object in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getAsciiStream

[InputStream](http://docs.google.com/java/io/InputStream.html) **getAsciiStream**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a stream of ASCII characters. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARCHAR values. The JDBC driver will do any necessary conversion from the database format into ASCII.

**Note:** All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return 0 when the method InputStream.available is called whether there is data available or not.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**a Java input stream that delivers the database column value as a stream of one-byte ASCII characters; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getUnicodeStream

[InputStream](http://docs.google.com/java/io/InputStream.html) **getUnicodeStream**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

**Deprecated.** *use getCharacterStream in place of getUnicodeStream*

Retrieves the value of the designated column in the current row of this ResultSet object as as a stream of two-byte 3 characters. The first byte is the high byte; the second byte is the low byte. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARCHARvalues. The JDBC driver will do any necessary conversion from the database format into Unicode.

**Note:** All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return 0 when the method InputStream.available is called, whether there is data available or not.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**a Java input stream that delivers the database column value as a stream of two-byte Unicode characters; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method

### getBinaryStream

[InputStream](http://docs.google.com/java/io/InputStream.html) **getBinaryStream**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a stream of uninterpreted bytes. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARBINARY values.

**Note:** All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return 0 when the method InputStream.available is called whether there is data available or not.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**a Java input stream that delivers the database column value as a stream of uninterpreted bytes; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getString

[String](http://docs.google.com/java/lang/String.html) **getString**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a String in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getBoolean

boolean **getBoolean**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a boolean in the Java programming language.

If the designated column has a datatype of CHAR or VARCHAR and contains a "0" or has a datatype of BIT, TINYINT, SMALLINT, INTEGER or BIGINT and contains a 0, a value of false is returned. If the designated column has a datatype of CHAR or VARCHAR and contains a "1" or has a datatype of BIT, TINYINT, SMALLINT, INTEGER or BIGINT and contains a 1, a value of true is returned.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is false **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getByte

byte **getByte**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a byte in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getShort

short **getShort**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a short in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getInt

int **getInt**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as an int in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getLong

long **getLong**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a long in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getFloat

float **getFloat**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a float in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getDouble

double **getDouble**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a double in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getBigDecimal

[BigDecimal](http://docs.google.com/java/math/BigDecimal.html) **getBigDecimal**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 int scale)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

**Deprecated.**

Retrieves the value of the designated column in the current row of this ResultSet object as a java.math.BigDecimal in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnscale - the number of digits to the right of the decimal point **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method

### getBytes

byte[] **getBytes**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a byte array in the Java programming language. The bytes represent the raw values returned by the driver.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getDate

[Date](http://docs.google.com/java/sql/Date.html) **getDate**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Date object in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getTime

[Time](http://docs.google.com/java/sql/Time.html) **getTime**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Time object in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getTimestamp

[Timestamp](http://docs.google.com/java/sql/Timestamp.html) **getTimestamp**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Timestamp object in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getAsciiStream

[InputStream](http://docs.google.com/java/io/InputStream.html) **getAsciiStream**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a stream of ASCII characters. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARCHAR values. The JDBC driver will do any necessary conversion from the database format into ASCII.

**Note:** All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return 0 when the method available is called whether there is data available or not.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**a Java input stream that delivers the database column value as a stream of one-byte ASCII characters. If the value is SQL NULL, the value returned is null. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getUnicodeStream

[InputStream](http://docs.google.com/java/io/InputStream.html) **getUnicodeStream**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

**Deprecated.** *use getCharacterStream instead*

Retrieves the value of the designated column in the current row of this ResultSet object as a stream of two-byte Unicode characters. The first byte is the high byte; the second byte is the low byte. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARCHAR values. The JDBC technology-enabled driver will do any necessary conversion from the database format into Unicode.

**Note:** All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return 0 when the method InputStream.available is called, whether there is data available or not.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**a Java input stream that delivers the database column value as a stream of two-byte Unicode characters. If the value is SQL NULL, the value returned is null. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method

### getBinaryStream

[InputStream](http://docs.google.com/java/io/InputStream.html) **getBinaryStream**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a stream of uninterpreted bytes. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARBINARY values.

**Note:** All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return 0 when the method available is called whether there is data available or not.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**a Java input stream that delivers the database column value as a stream of uninterpreted bytes; if the value is SQL NULL, the result is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### getWarnings

[SQLWarning](http://docs.google.com/java/sql/SQLWarning.html) **getWarnings**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the first warning reported by calls on this ResultSet object. Subsequent warnings on this ResultSet object will be chained to the SQLWarning object that this method returns.

The warning chain is automatically cleared each time a new row is read. This method may not be called on a ResultSet object that has been closed; doing so will cause an SQLException to be thrown.

**Note:** This warning chain only covers warnings caused by ResultSet methods. Any warning caused by Statement methods (such as reading OUT parameters) will be chained on the Statement object.

**Returns:**the first SQLWarning object reported or null if there are none **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set

### clearWarnings

void **clearWarnings**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Clears all warnings reported on this ResultSet object. After this method is called, the method getWarnings returns null until a new warning is reported for this ResultSet object.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set

### getCursorName

[String](http://docs.google.com/java/lang/String.html) **getCursorName**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the name of the SQL cursor used by this ResultSet object.

In SQL, a result table is retrieved through a cursor that is named. The current row of a result set can be updated or deleted using a positioned update/delete statement that references the cursor name. To insure that the cursor has the proper isolation level to support update, the cursor's SELECT statement should be of the form SELECT FOR UPDATE. If FOR UPDATE is omitted, the positioned updates may fail.

The JDBC API supports this SQL feature by providing the name of the SQL cursor used by a ResultSet object. The current row of a ResultSet object is also the current row of this SQL cursor.

**Returns:**the SQL name for this ResultSet object's cursor **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method

### getMetaData

[ResultSetMetaData](http://docs.google.com/java/sql/ResultSetMetaData.html) **getMetaData**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the number, types and properties of this ResultSet object's columns.

**Returns:**the description of this ResultSet object's columns **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set

### getObject

[Object](http://docs.google.com/java/lang/Object.html) **getObject**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Gets the value of the designated column in the current row of this ResultSet object as an Object in the Java programming language.

This method will return the value of the given column as a Java object. The type of the Java object will be the default Java object type corresponding to the column's SQL type, following the mapping for built-in types specified in the JDBC specification. If the value is an SQL NULL, the driver returns a Java null.

This method may also be used to read database-specific abstract data types. In the JDBC 2.0 API, the behavior of method getObject is extended to materialize data of SQL user-defined types.

If Connection.getTypeMap does not throw a SQLFeatureNotSupportedException, then when a column contains a structured or distinct value, the behavior of this method is as if it were a call to: getObject(columnIndex, this.getStatement().getConnection().getTypeMap()). If Connection.getTypeMap does throw a SQLFeatureNotSupportedException, then structured values are not supported, and distinct values are mapped to the default Java class as determined by the underlying SQL type of the DISTINCT type.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**a java.lang.Object holding the column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set

### getObject

[Object](http://docs.google.com/java/lang/Object.html) **getObject**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Gets the value of the designated column in the current row of this ResultSet object as an Object in the Java programming language.

This method will return the value of the given column as a Java object. The type of the Java object will be the default Java object type corresponding to the column's SQL type, following the mapping for built-in types specified in the JDBC specification. If the value is an SQL NULL, the driver returns a Java null.

This method may also be used to read database-specific abstract data types.

In the JDBC 2.0 API, the behavior of the method getObject is extended to materialize data of SQL user-defined types. When a column contains a structured or distinct value, the behavior of this method is as if it were a call to: getObject(columnIndex, this.getStatement().getConnection().getTypeMap()).

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**a java.lang.Object holding the column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set

### findColumn

int **findColumn**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Maps the given ResultSet column label to its ResultSet column index.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column index of the given column name **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the ResultSet object does not contain a column labeled columnLabel, a database access error occurs or this method is called on a closed result set

### getCharacterStream

[Reader](http://docs.google.com/java/io/Reader.html) **getCharacterStream**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.io.Reader object.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**a java.io.Reader object that contains the column value; if the value is SQL NULL, the value returned is null in the Java programming language. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### getCharacterStream

[Reader](http://docs.google.com/java/io/Reader.html) **getCharacterStream**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.io.Reader object.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**a java.io.Reader object that contains the column value; if the value is SQL NULL, the value returned is null in the Java programming language **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### getBigDecimal

[BigDecimal](http://docs.google.com/java/math/BigDecimal.html) **getBigDecimal**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.math.BigDecimal with full precision.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value (full precision); if the value is SQL NULL, the value returned is null in the Java programming language. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### getBigDecimal

[BigDecimal](http://docs.google.com/java/math/BigDecimal.html) **getBigDecimal**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.math.BigDecimal with full precision.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value (full precision); if the value is SQL NULL, the value returned is null in the Java programming language. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### isBeforeFirst

boolean **isBeforeFirst**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves whether the cursor is before the first row in this ResultSet object.

**Note:**Support for the isBeforeFirst method is optional for ResultSets with a result set type of TYPE\_FORWARD\_ONLY

**Returns:**true if the cursor is before the first row; false if the cursor is at any other position or the result set contains no rows **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### isAfterLast

boolean **isAfterLast**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves whether the cursor is after the last row in this ResultSet object.

**Note:**Support for the isAfterLast method is optional for ResultSets with a result set type of TYPE\_FORWARD\_ONLY

**Returns:**true if the cursor is after the last row; false if the cursor is at any other position or the result set contains no rows **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### isFirst

boolean **isFirst**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves whether the cursor is on the first row of this ResultSet object.

**Note:**Support for the isFirst method is optional for ResultSets with a result set type of TYPE\_FORWARD\_ONLY

**Returns:**true if the cursor is on the first row; false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### isLast

boolean **isLast**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves whether the cursor is on the last row of this ResultSet object. **Note:** Calling the method isLast may be expensive because the JDBC driver might need to fetch ahead one row in order to determine whether the current row is the last row in the result set.

**Note:** Support for the isLast method is optional for ResultSets with a result set type of TYPE\_FORWARD\_ONLY

**Returns:**true if the cursor is on the last row; false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### beforeFirst

void **beforeFirst**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Moves the cursor to the front of this ResultSet object, just before the first row. This method has no effect if the result set contains no rows.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set or the result set type is TYPE\_FORWARD\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### afterLast

void **afterLast**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Moves the cursor to the end of this ResultSet object, just after the last row. This method has no effect if the result set contains no rows.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set or the result set type is TYPE\_FORWARD\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### first

boolean **first**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Moves the cursor to the first row in this ResultSet object.

**Returns:**true if the cursor is on a valid row; false if there are no rows in the result set **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set or the result set type is TYPE\_FORWARD\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### last

boolean **last**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Moves the cursor to the last row in this ResultSet object.

**Returns:**true if the cursor is on a valid row; false if there are no rows in the result set **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set or the result set type is TYPE\_FORWARD\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getRow

int **getRow**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the current row number. The first row is number 1, the second number 2, and so on.

**Note:**Support for the getRow method is optional for ResultSets with a result set type of TYPE\_FORWARD\_ONLY

**Returns:**the current row number; 0 if there is no current row **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### absolute

boolean **absolute**(int row)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Moves the cursor to the given row number in this ResultSet object.

If the row number is positive, the cursor moves to the given row number with respect to the beginning of the result set. The first row is row 1, the second is row 2, and so on.

If the given row number is negative, the cursor moves to an absolute row position with respect to the end of the result set. For example, calling the method absolute(-1) positions the cursor on the last row; calling the method absolute(-2) moves the cursor to the next-to-last row, and so on.

An attempt to position the cursor beyond the first/last row in the result set leaves the cursor before the first row or after the last row.

**Note:** Calling absolute(1) is the same as calling first(). Calling absolute(-1) is the same as calling last().

**Parameters:**row - the number of the row to which the cursor should move. A positive number indicates the row number counting from the beginning of the result set; a negative number indicates the row number counting from the end of the result set **Returns:**true if the cursor is moved to a position in this ResultSet object; false if the cursor is before the first row or after the last row **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set or the result set type is TYPE\_FORWARD\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### relative

boolean **relative**(int rows)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Moves the cursor a relative number of rows, either positive or negative. Attempting to move beyond the first/last row in the result set positions the cursor before/after the the first/last row. Calling relative(0) is valid, but does not change the cursor position.

Note: Calling the method relative(1) is identical to calling the method next() and calling the method relative(-1) is identical to calling the method previous().

**Parameters:**rows - an int specifying the number of rows to move from the current row; a positive number moves the cursor forward; a negative number moves the cursor backward **Returns:**true if the cursor is on a row; false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set or the result set type is TYPE\_FORWARD\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### previous

boolean **previous**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Moves the cursor to the previous row in this ResultSet object.

When a call to the previous method returns false, the cursor is positioned before the first row. Any invocation of a ResultSet method which requires a current row will result in a SQLException being thrown.

If an input stream is open for the current row, a call to the method previous will implicitly close it. A ResultSet object's warning change is cleared when a new row is read.

**Returns:**true if the cursor is now positioned on a valid row; false if the cursor is positioned before the first row **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set or the result set type is TYPE\_FORWARD\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### setFetchDirection

void **setFetchDirection**(int direction)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Gives a hint as to the direction in which the rows in this ResultSet object will be processed. The initial value is determined by the Statement object that produced this ResultSet object. The fetch direction may be changed at any time.

**Parameters:**direction - an int specifying the suggested fetch direction; one of ResultSet.FETCH\_FORWARD, ResultSet.FETCH\_REVERSE, or ResultSet.FETCH\_UNKNOWN **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set or the result set type is TYPE\_FORWARD\_ONLY and the fetch direction is not FETCH\_FORWARD**Since:** 1.2 **See Also:**[Statement.setFetchDirection(int)](http://docs.google.com/java/sql/Statement.html#setFetchDirection(int)), [getFetchDirection()](http://docs.google.com/java/sql/ResultSet.html#getFetchDirection())

### getFetchDirection

int **getFetchDirection**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the fetch direction for this ResultSet object.

**Returns:**the current fetch direction for this ResultSet object **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set**Since:** 1.2 **See Also:**[setFetchDirection(int)](http://docs.google.com/java/sql/ResultSet.html#setFetchDirection(int))

### setFetchSize

void **setFetchSize**(int rows)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Gives the JDBC driver a hint as to the number of rows that should be fetched from the database when more rows are needed for this ResultSet object. If the fetch size specified is zero, the JDBC driver ignores the value and is free to make its own best guess as to what the fetch size should be. The default value is set by the Statement object that created the result set. The fetch size may be changed at any time.

**Parameters:**rows - the number of rows to fetch **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set or the condition rows >= 0 is not satisfied**Since:** 1.2 **See Also:**[getFetchSize()](http://docs.google.com/java/sql/ResultSet.html#getFetchSize())

### getFetchSize

int **getFetchSize**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the fetch size for this ResultSet object.

**Returns:**the current fetch size for this ResultSet object **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set**Since:** 1.2 **See Also:**[setFetchSize(int)](http://docs.google.com/java/sql/ResultSet.html#setFetchSize(int))

### getType

int **getType**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the type of this ResultSet object. The type is determined by the Statement object that created the result set.

**Returns:**ResultSet.TYPE\_FORWARD\_ONLY, ResultSet.TYPE\_SCROLL\_INSENSITIVE, or ResultSet.TYPE\_SCROLL\_SENSITIVE **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### getConcurrency

int **getConcurrency**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the concurrency mode of this ResultSet object. The concurrency used is determined by the Statement object that created the result set.

**Returns:**the concurrency type, either ResultSet.CONCUR\_READ\_ONLY or ResultSet.CONCUR\_UPDATABLE **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### rowUpdated

boolean **rowUpdated**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves whether the current row has been updated. The value returned depends on whether or not the result set can detect updates.

**Note:** Support for the rowUpdated method is optional with a result set concurrency of CONCUR\_READ\_ONLY

**Returns:**true if the current row is detected to have been visibly updated by the owner or another; false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2 **See Also:**[DatabaseMetaData.updatesAreDetected(int)](http://docs.google.com/java/sql/DatabaseMetaData.html#updatesAreDetected(int))

### rowInserted

boolean **rowInserted**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves whether the current row has had an insertion. The value returned depends on whether or not this ResultSet object can detect visible inserts.

**Note:** Support for the rowInserted method is optional with a result set concurrency of CONCUR\_READ\_ONLY

**Returns:**true if the current row is detected to have been inserted; false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2 **See Also:**[DatabaseMetaData.insertsAreDetected(int)](http://docs.google.com/java/sql/DatabaseMetaData.html#insertsAreDetected(int))

### rowDeleted

boolean **rowDeleted**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves whether a row has been deleted. A deleted row may leave a visible "hole" in a result set. This method can be used to detect holes in a result set. The value returned depends on whether or not this ResultSet object can detect deletions.

**Note:** Support for the rowDeleted method is optional with a result set concurrency of CONCUR\_READ\_ONLY

**Returns:**true if the current row is detected to have been deleted by the owner or another; false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2 **See Also:**[DatabaseMetaData.deletesAreDetected(int)](http://docs.google.com/java/sql/DatabaseMetaData.html#deletesAreDetected(int))

### updateNull

void **updateNull**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a null value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateBoolean

void **updateBoolean**(int columnIndex,  
 boolean x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a boolean value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateByte

void **updateByte**(int columnIndex,  
 byte x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a byte value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateShort

void **updateShort**(int columnIndex,  
 short x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a short value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateInt

void **updateInt**(int columnIndex,  
 int x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with an int value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateLong

void **updateLong**(int columnIndex,  
 long x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a long value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateFloat

void **updateFloat**(int columnIndex,  
 float x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a float value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateDouble

void **updateDouble**(int columnIndex,  
 double x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a double value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateBigDecimal

void **updateBigDecimal**(int columnIndex,  
 [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.math.BigDecimal value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateString

void **updateString**(int columnIndex,  
 [String](http://docs.google.com/java/lang/String.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a String value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateBytes

void **updateBytes**(int columnIndex,  
 byte[] x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a byte array value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateDate

void **updateDate**(int columnIndex,  
 [Date](http://docs.google.com/java/sql/Date.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Date value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateTime

void **updateTime**(int columnIndex,  
 [Time](http://docs.google.com/java/sql/Time.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Time value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateTimestamp

void **updateTimestamp**(int columnIndex,  
 [Timestamp](http://docs.google.com/java/sql/Timestamp.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Timestamp value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateAsciiStream

void **updateAsciiStream**(int columnIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with an ascii stream value, which will have the specified number of bytes. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateBinaryStream

void **updateBinaryStream**(int columnIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a binary stream value, which will have the specified number of bytes. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateCharacterStream

void **updateCharacterStream**(int columnIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) x,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a character stream value, which will have the specified number of bytes. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateObject

void **updateObject**(int columnIndex,  
 [Object](http://docs.google.com/java/lang/Object.html) x,  
 int scaleOrLength)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with an Object value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

If the second argument is an InputStream then the stream must contain the number of bytes specified by scaleOrLength. If the second argument is a Reader then the reader must contain the number of characters specified by scaleOrLength. If these conditions are not true the driver will generate a SQLException when the statement is executed.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column valuescaleOrLength - for an object of java.math.BigDecimal , this is the number of digits after the decimal point. For Java Object types InputStream and Reader, this is the length of the data in the stream or reader. For all other types, this value will be ignored. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateObject

void **updateObject**(int columnIndex,  
 [Object](http://docs.google.com/java/lang/Object.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with an Object value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateNull

void **updateNull**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a null value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateBoolean

void **updateBoolean**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 boolean x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a boolean value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateByte

void **updateByte**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 byte x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a byte value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateShort

void **updateShort**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 short x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a short value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateInt

void **updateInt**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 int x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with an int value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateLong

void **updateLong**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 long x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a long value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateFloat

void **updateFloat**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 float x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a float value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateDouble

void **updateDouble**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 double x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a double value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateBigDecimal

void **updateBigDecimal**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.BigDecimal value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateString

void **updateString**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [String](http://docs.google.com/java/lang/String.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a String value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateBytes

void **updateBytes**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 byte[] x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a byte array value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateDate

void **updateDate**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Date](http://docs.google.com/java/sql/Date.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Date value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateTime

void **updateTime**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Time](http://docs.google.com/java/sql/Time.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Time value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateTimestamp

void **updateTimestamp**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Timestamp](http://docs.google.com/java/sql/Timestamp.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Timestamp value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateAsciiStream

void **updateAsciiStream**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with an ascii stream value, which will have the specified number of bytes. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateBinaryStream

void **updateBinaryStream**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a binary stream value, which will have the specified number of bytes. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateCharacterStream

void **updateCharacterStream**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a character stream value, which will have the specified number of bytes. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnreader - the java.io.Reader object containing the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateObject

void **updateObject**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Object](http://docs.google.com/java/lang/Object.html) x,  
 int scaleOrLength)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with an Object value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

If the second argument is an InputStream then the stream must contain the number of bytes specified by scaleOrLength. If the second argument is a Reader then the reader must contain the number of characters specified by scaleOrLength. If these conditions are not true the driver will generate a SQLException when the statement is executed.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column valuescaleOrLength - for an object of java.math.BigDecimal , this is the number of digits after the decimal point. For Java Object types InputStream and Reader, this is the length of the data in the stream or reader. For all other types, this value will be ignored. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateObject

void **updateObject**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Object](http://docs.google.com/java/lang/Object.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with an Object value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### insertRow

void **insertRow**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Inserts the contents of the insert row into this ResultSet object and into the database. The cursor must be on the insert row when this method is called.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY, this method is called on a closed result set, if this method is called when the cursor is not on the insert row, or if not all of non-nullable columns in the insert row have been given a non-null value [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### updateRow

void **updateRow**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the underlying database with the new contents of the current row of this ResultSet object. This method cannot be called when the cursor is on the insert row.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY; this method is called on a closed result set or if this method is called when the cursor is on the insert row [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### deleteRow

void **deleteRow**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Deletes the current row from this ResultSet object and from the underlying database. This method cannot be called when the cursor is on the insert row.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY; this method is called on a closed result set or if this method is called when the cursor is on the insert row [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### refreshRow

void **refreshRow**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Refreshes the current row with its most recent value in the database. This method cannot be called when the cursor is on the insert row.

The refreshRow method provides a way for an application to explicitly tell the JDBC driver to refetch a row(s) from the database. An application may want to call refreshRow when caching or prefetching is being done by the JDBC driver to fetch the latest value of a row from the database. The JDBC driver may actually refresh multiple rows at once if the fetch size is greater than one.

All values are refetched subject to the transaction isolation level and cursor sensitivity. If refreshRow is called after calling an updater method, but before calling the method updateRow, then the updates made to the row are lost. Calling the method refreshRow frequently will likely slow performance.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set; the result set type is TYPE\_FORWARD\_ONLY or if this method is called when the cursor is on the insert row [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method or this method is not supported for the specified result set type and result set concurrency.**Since:** 1.2

### cancelRowUpdates

void **cancelRowUpdates**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Cancels the updates made to the current row in this ResultSet object. This method may be called after calling an updater method(s) and before calling the method updateRow to roll back the updates made to a row. If no updates have been made or updateRow has already been called, this method has no effect.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set; the result set concurrency is CONCUR\_READ\_ONLY or if this method is called when the cursor is on the insert row [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### moveToInsertRow

void **moveToInsertRow**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Moves the cursor to the insert row. The current cursor position is remembered while the cursor is positioned on the insert row. The insert row is a special row associated with an updatable result set. It is essentially a buffer where a new row may be constructed by calling the updater methods prior to inserting the row into the result set. Only the updater, getter, and insertRow methods may be called when the cursor is on the insert row. All of the columns in a result set must be given a value each time this method is called before calling insertRow. An updater method must be called before a getter method can be called on a column value.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set or the result set concurrency is CONCUR\_READ\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### moveToCurrentRow

void **moveToCurrentRow**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Moves the cursor to the remembered cursor position, usually the current row. This method has no effect if the cursor is not on the insert row.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs; this method is called on a closed result set or the result set concurrency is CONCUR\_READ\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getStatement

[Statement](http://docs.google.com/java/sql/Statement.html) **getStatement**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the Statement object that produced this ResultSet object. If the result set was generated some other way, such as by a DatabaseMetaData method, this method may return null.

**Returns:**the Statment object that produced this ResultSet object or null if the result set was produced some other way **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### getObject

[Object](http://docs.google.com/java/lang/Object.html) **getObject**(int columnIndex,  
 [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as an Object in the Java programming language. If the value is an SQL NULL, the driver returns a Java null. This method uses the given Map object for the custom mapping of the SQL structured or distinct type that is being retrieved.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...map - a java.util.Map object that contains the mapping from SQL type names to classes in the Java programming language **Returns:**an Object in the Java programming language representing the SQL value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getRef

[Ref](http://docs.google.com/java/sql/Ref.html) **getRef**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a Ref object in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**a Ref object representing an SQL REF value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getBlob

[Blob](http://docs.google.com/java/sql/Blob.html) **getBlob**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a Blob object in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**a Blob object representing the SQL BLOB value in the specified column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getClob

[Clob](http://docs.google.com/java/sql/Clob.html) **getClob**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a Clob object in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**a Clob object representing the SQL CLOB value in the specified column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getArray

[Array](http://docs.google.com/java/sql/Array.html) **getArray**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as an Array object in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**an Array object representing the SQL ARRAY value in the specified column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getObject

[Object](http://docs.google.com/java/lang/Object.html) **getObject**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as an Object in the Java programming language. If the value is an SQL NULL, the driver returns a Java null. This method uses the specified Map object for custom mapping if appropriate.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnmap - a java.util.Map object that contains the mapping from SQL type names to classes in the Java programming language **Returns:**an Object representing the SQL value in the specified column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getRef

[Ref](http://docs.google.com/java/sql/Ref.html) **getRef**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a Ref object in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**a Ref object representing the SQL REF value in the specified column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getBlob

[Blob](http://docs.google.com/java/sql/Blob.html) **getBlob**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a Blob object in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**a Blob object representing the SQL BLOB value in the specified column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getClob

[Clob](http://docs.google.com/java/sql/Clob.html) **getClob**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a Clob object in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**a Clob object representing the SQL CLOB value in the specified column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getArray

[Array](http://docs.google.com/java/sql/Array.html) **getArray**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as an Array object in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**an Array object representing the SQL ARRAY value in the specified column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getDate

[Date](http://docs.google.com/java/sql/Date.html) **getDate**(int columnIndex,  
 [Calendar](http://docs.google.com/java/util/Calendar.html) cal)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Date object in the Java programming language. This method uses the given calendar to construct an appropriate millisecond value for the date if the underlying database does not store timezone information.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...cal - the java.util.Calendar object to use in constructing the date **Returns:**the column value as a java.sql.Date object; if the value is SQL NULL, the value returned is null in the Java programming language **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### getDate

[Date](http://docs.google.com/java/sql/Date.html) **getDate**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Calendar](http://docs.google.com/java/util/Calendar.html) cal)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Date object in the Java programming language. This method uses the given calendar to construct an appropriate millisecond value for the date if the underlying database does not store timezone information.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columncal - the java.util.Calendar object to use in constructing the date **Returns:**the column value as a java.sql.Date object; if the value is SQL NULL, the value returned is null in the Java programming language **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### getTime

[Time](http://docs.google.com/java/sql/Time.html) **getTime**(int columnIndex,  
 [Calendar](http://docs.google.com/java/util/Calendar.html) cal)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Time object in the Java programming language. This method uses the given calendar to construct an appropriate millisecond value for the time if the underlying database does not store timezone information.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...cal - the java.util.Calendar object to use in constructing the time **Returns:**the column value as a java.sql.Time object; if the value is SQL NULL, the value returned is null in the Java programming language **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### getTime

[Time](http://docs.google.com/java/sql/Time.html) **getTime**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Calendar](http://docs.google.com/java/util/Calendar.html) cal)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Time object in the Java programming language. This method uses the given calendar to construct an appropriate millisecond value for the time if the underlying database does not store timezone information.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columncal - the java.util.Calendar object to use in constructing the time **Returns:**the column value as a java.sql.Time object; if the value is SQL NULL, the value returned is null in the Java programming language **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### getTimestamp

[Timestamp](http://docs.google.com/java/sql/Timestamp.html) **getTimestamp**(int columnIndex,  
 [Calendar](http://docs.google.com/java/util/Calendar.html) cal)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Timestamp object in the Java programming language. This method uses the given calendar to construct an appropriate millisecond value for the timestamp if the underlying database does not store timezone information.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...cal - the java.util.Calendar object to use in constructing the timestamp **Returns:**the column value as a java.sql.Timestamp object; if the value is SQL NULL, the value returned is null in the Java programming language **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### getTimestamp

[Timestamp](http://docs.google.com/java/sql/Timestamp.html) **getTimestamp**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Calendar](http://docs.google.com/java/util/Calendar.html) cal)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.Timestamp object in the Java programming language. This method uses the given calendar to construct an appropriate millisecond value for the timestamp if the underlying database does not store timezone information.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columncal - the java.util.Calendar object to use in constructing the date **Returns:**the column value as a java.sql.Timestamp object; if the value is SQL NULL, the value returned is null in the Java programming language **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid or if a database access error occurs or this method is called on a closed result set**Since:** 1.2

### getURL

[URL](http://docs.google.com/java/net/URL.html) **getURL**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.net.URL object in the Java programming language.

**Parameters:**columnIndex - the index of the column 1 is the first, 2 is the second,... **Returns:**the column value as a java.net.URL object; if the value is SQL NULL, the value returned is null in the Java programming language **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; this method is called on a closed result set or if a URL is malformed [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### getURL

[URL](http://docs.google.com/java/net/URL.html) **getURL**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.net.URL object in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value as a java.net.URL object; if the value is SQL NULL, the value returned is null in the Java programming language **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; this method is called on a closed result set or if a URL is malformed [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### updateRef

void **updateRef**(int columnIndex,  
 [Ref](http://docs.google.com/java/sql/Ref.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Ref value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### updateRef

void **updateRef**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Ref](http://docs.google.com/java/sql/Ref.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Ref value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### updateBlob

void **updateBlob**(int columnIndex,  
 [Blob](http://docs.google.com/java/sql/Blob.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Blob value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### updateBlob

void **updateBlob**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Blob](http://docs.google.com/java/sql/Blob.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Blob value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### updateClob

void **updateClob**(int columnIndex,  
 [Clob](http://docs.google.com/java/sql/Clob.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Clob value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### updateClob

void **updateClob**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Clob](http://docs.google.com/java/sql/Clob.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Clob value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### updateArray

void **updateArray**(int columnIndex,  
 [Array](http://docs.google.com/java/sql/Array.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Array value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### updateArray

void **updateArray**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Array](http://docs.google.com/java/sql/Array.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.Array value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### getRowId

[RowId](http://docs.google.com/java/sql/RowId.html) **getRowId**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.RowId object in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second 2, ... **Returns:**the column value; if the value is a SQL NULL the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### getRowId

[RowId](http://docs.google.com/java/sql/RowId.html) **getRowId**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.sql.RowId object in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value ; if the value is a SQL NULL the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateRowId

void **updateRowId**(int columnIndex,  
 [RowId](http://docs.google.com/java/sql/RowId.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a RowId value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second 2, ...x - the column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateRowId

void **updateRowId**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [RowId](http://docs.google.com/java/sql/RowId.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a RowId value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### getHoldability

int **getHoldability**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the holdability of this ResultSet object

**Returns:**either ResultSet.HOLD\_CURSORS\_OVER\_COMMIT or ResultSet.CLOSE\_CURSORS\_AT\_COMMIT **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed result set**Since:** 1.6

### isClosed

boolean **isClosed**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves whether this ResultSet object has been closed. A ResultSet is closed if the method close has been called on it, or if it is automatically closed.

**Returns:**true if this ResultSet object is closed; false if it is still open **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**Since:** 1.6

### updateNString

void **updateNString**(int columnIndex,  
 [String](http://docs.google.com/java/lang/String.html) nString)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a String value. It is intended for use when updating NCHAR,NVARCHAR and LONGNVARCHAR columns. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second 2, ...nString - the value for the column to be updated **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; this method is called on a closed result set; the result set concurrency is CONCUR\_READ\_ONLY or if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateNString

void **updateNString**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [String](http://docs.google.com/java/lang/String.html) nString)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a String value. It is intended for use when updating NCHAR,NVARCHAR and LONGNVARCHAR columns. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnnString - the value for the column to be updated **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; this method is called on a closed result set; the result set concurrency is CONCUR\_READ\_ONLY or if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateNClob

void **updateNClob**(int columnIndex,  
 [NClob](http://docs.google.com/java/sql/NClob.html) nClob)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.NClob value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second 2, ...nClob - the value for the column to be updated **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; this method is called on a closed result set; if a database access error occurs or the result set concurrency is CONCUR\_READ\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateNClob

void **updateNClob**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [NClob](http://docs.google.com/java/sql/NClob.html) nClob)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.NClob value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnnClob - the value for the column to be updated **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; this method is called on a closed result set; if a database access error occurs or the result set concurrency is CONCUR\_READ\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### getNClob

[NClob](http://docs.google.com/java/sql/NClob.html) **getNClob**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a NClob object in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**a NClob object representing the SQL NCLOB value in the specified column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; this method is called on a closed result set or if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### getNClob

[NClob](http://docs.google.com/java/sql/NClob.html) **getNClob**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a NClob object in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**a NClob object representing the SQL NCLOB value in the specified column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; this method is called on a closed result set or if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### getSQLXML

[SQLXML](http://docs.google.com/java/sql/SQLXML.html) **getSQLXML**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet as a java.sql.SQLXML object in the Java programming language.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**a SQLXML object that maps an SQL XML value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### getSQLXML

[SQLXML](http://docs.google.com/java/sql/SQLXML.html) **getSQLXML**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet as a java.sql.SQLXML object in the Java programming language.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**a SQLXML object that maps an SQL XML value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateSQLXML

void **updateSQLXML**(int columnIndex,  
 [SQLXML](http://docs.google.com/java/sql/SQLXML.html) xmlObject)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.SQLXML value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second 2, ...xmlObject - the value for the column to be updated **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; this method is called on a closed result set; the java.xml.transform.Result, Writer or OutputStream has not been closed for the SQLXML object; if there is an error processing the XML value or the result set concurrency is CONCUR\_READ\_ONLY. The getCause method of the exception may provide a more detailed exception, for example, if the stream does not contain valid XML. [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateSQLXML

void **updateSQLXML**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [SQLXML](http://docs.google.com/java/sql/SQLXML.html) xmlObject)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a java.sql.SQLXML value. The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnxmlObject - the column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; this method is called on a closed result set; the java.xml.transform.Result, Writer or OutputStream has not been closed for the SQLXML object; if there is an error processing the XML value or the result set concurrency is CONCUR\_READ\_ONLY. The getCause method of the exception may provide a more detailed exception, for example, if the stream does not contain valid XML. [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### getNString

[String](http://docs.google.com/java/lang/String.html) **getNString**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a String in the Java programming language. It is intended for use when accessing NCHAR,NVARCHAR and LONGNVARCHAR columns.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### getNString

[String](http://docs.google.com/java/lang/String.html) **getNString**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a String in the Java programming language. It is intended for use when accessing NCHAR,NVARCHAR and LONGNVARCHAR columns.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**the column value; if the value is SQL NULL, the value returned is null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### getNCharacterStream

[Reader](http://docs.google.com/java/io/Reader.html) **getNCharacterStream**(int columnIndex)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.io.Reader object. It is intended for use when accessing NCHAR,NVARCHAR and LONGNVARCHAR columns.

**Parameters:**columnIndex - the first column is 1, the second is 2, ... **Returns:**a java.io.Reader object that contains the column value; if the value is SQL NULL, the value returned is null in the Java programming language. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### getNCharacterStream

[Reader](http://docs.google.com/java/io/Reader.html) **getNCharacterStream**([String](http://docs.google.com/java/lang/String.html) columnLabel)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the value of the designated column in the current row of this ResultSet object as a java.io.Reader object. It is intended for use when accessing NCHAR,NVARCHAR and LONGNVARCHAR columns.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column **Returns:**a java.io.Reader object that contains the column value; if the value is SQL NULL, the value returned is null in the Java programming language **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateNCharacterStream

void **updateNCharacterStream**(int columnIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) x,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a character stream value, which will have the specified number of bytes. The driver does the necessary conversion from Java character format to the national character set in the database. It is intended for use when updating NCHAR,NVARCHAR and LONGNVARCHAR columns.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateNCharacterStream

void **updateNCharacterStream**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a character stream value, which will have the specified number of bytes. The driver does the necessary conversion from Java character format to the national character set in the database. It is intended for use when updating NCHAR,NVARCHAR and LONGNVARCHAR columns.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnreader - the java.io.Reader object containing the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateAsciiStream

void **updateAsciiStream**(int columnIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with an ascii stream value, which will have the specified number of bytes.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateBinaryStream

void **updateBinaryStream**(int columnIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a binary stream value, which will have the specified number of bytes.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateCharacterStream

void **updateCharacterStream**(int columnIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) x,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a character stream value, which will have the specified number of bytes.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateAsciiStream

void **updateAsciiStream**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with an ascii stream value, which will have the specified number of bytes.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateBinaryStream

void **updateBinaryStream**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a binary stream value, which will have the specified number of bytes.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateCharacterStream

void **updateCharacterStream**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a character stream value, which will have the specified number of bytes.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnreader - the java.io.Reader object containing the new column valuelength - the length of the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateBlob

void **updateBlob**(int columnIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column using the given input stream, which will have the specified number of bytes.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...inputStream - An object that contains the data to set the parameter value to.length - the number of bytes in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateBlob

void **updateBlob**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column using the given input stream, which will have the specified number of bytes.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columninputStream - An object that contains the data to set the parameter value to.length - the number of bytes in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateClob

void **updateClob**(int columnIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column using the given Reader object, which is the given number of characters long. When a very large UNICODE value is input to a LONGVARCHAR parameter, it may be more practical to send it via a java.io.Reader object. The JDBC driver will do any necessary conversion from UNICODE to the database char format.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...reader - An object that contains the data to set the parameter value to.length - the number of characters in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateClob

void **updateClob**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column using the given Reader object, which is the given number of characters long. When a very large UNICODE value is input to a LONGVARCHAR parameter, it may be more practical to send it via a java.io.Reader object. The JDBC driver will do any necessary conversion from UNICODE to the database char format.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnreader - An object that contains the data to set the parameter value to.length - the number of characters in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateNClob

void **updateNClob**(int columnIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column using the given Reader object, which is the given number of characters long. When a very large UNICODE value is input to a LONGVARCHAR parameter, it may be more practical to send it via a java.io.Reader object. The JDBC driver will do any necessary conversion from UNICODE to the database char format.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnIndex - the first column is 1, the second 2, ...reader - An object that contains the data to set the parameter value to.length - the number of characters in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; this method is called on a closed result set, if a database access error occurs or the result set concurrency is CONCUR\_READ\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateNClob

void **updateNClob**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column using the given Reader object, which is the given number of characters long. When a very large UNICODE value is input to a LONGVARCHAR parameter, it may be more practical to send it via a java.io.Reader object. The JDBC driver will do any necessary conversion from UNICODE to the database char format.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnreader - An object that contains the data to set the parameter value to.length - the number of characters in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; this method is called on a closed result set; if a database access error occurs or the result set concurrency is CONCUR\_READ\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateNCharacterStream

void **updateNCharacterStream**(int columnIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a character stream value. The data will be read from the stream as needed until end-of-stream is reached. The driver does the necessary conversion from Java character format to the national character set in the database. It is intended for use when updating NCHAR,NVARCHAR and LONGNVARCHAR columns.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateNCharacterStream which takes a length parameter.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateNCharacterStream

void **updateNCharacterStream**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a character stream value. The data will be read from the stream as needed until end-of-stream is reached. The driver does the necessary conversion from Java character format to the national character set in the database. It is intended for use when updating NCHAR,NVARCHAR and LONGNVARCHAR columns.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateNCharacterStream which takes a length parameter.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnreader - the java.io.Reader object containing the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateAsciiStream

void **updateAsciiStream**(int columnIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with an ascii stream value. The data will be read from the stream as needed until end-of-stream is reached.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateAsciiStream which takes a length parameter.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateBinaryStream

void **updateBinaryStream**(int columnIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a binary stream value. The data will be read from the stream as needed until end-of-stream is reached.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateBinaryStream which takes a length parameter.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateCharacterStream

void **updateCharacterStream**(int columnIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a character stream value. The data will be read from the stream as needed until end-of-stream is reached.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateCharacterStream which takes a length parameter.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...x - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateAsciiStream

void **updateAsciiStream**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with an ascii stream value. The data will be read from the stream as needed until end-of-stream is reached.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateAsciiStream which takes a length parameter.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateBinaryStream

void **updateBinaryStream**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a binary stream value. The data will be read from the stream as needed until end-of-stream is reached.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateBinaryStream which takes a length parameter.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnx - the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateCharacterStream

void **updateCharacterStream**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column with a character stream value. The data will be read from the stream as needed until end-of-stream is reached.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateCharacterStream which takes a length parameter.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnreader - the java.io.Reader object containing the new column value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateBlob

void **updateBlob**(int columnIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column using the given input stream. The data will be read from the stream as needed until end-of-stream is reached.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateBlob which takes a length parameter.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...inputStream - An object that contains the data to set the parameter value to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateBlob

void **updateBlob**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column using the given input stream. The data will be read from the stream as needed until end-of-stream is reached.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateBlob which takes a length parameter.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columninputStream - An object that contains the data to set the parameter value to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateClob

void **updateClob**(int columnIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column using the given Reader object. The data will be read from the stream as needed until end-of-stream is reached. The JDBC driver will do any necessary conversion from UNICODE to the database char format.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateClob which takes a length parameter.

**Parameters:**columnIndex - the first column is 1, the second is 2, ...reader - An object that contains the data to set the parameter value to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateClob

void **updateClob**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column using the given Reader object. The data will be read from the stream as needed until end-of-stream is reached. The JDBC driver will do any necessary conversion from UNICODE to the database char format.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateClob which takes a length parameter.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnreader - An object that contains the data to set the parameter value to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if a database access error occurs; the result set concurrency is CONCUR\_READ\_ONLY or this method is called on a closed result set [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateNClob

void **updateNClob**(int columnIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column using the given Reader The data will be read from the stream as needed until end-of-stream is reached. The JDBC driver will do any necessary conversion from UNICODE to the database char format.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateNClob which takes a length parameter.

**Parameters:**columnIndex - the first column is 1, the second 2, ...reader - An object that contains the data to set the parameter value to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnIndex is not valid; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; this method is called on a closed result set, if a database access error occurs or the result set concurrency is CONCUR\_READ\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### updateNClob

void **updateNClob**([String](http://docs.google.com/java/lang/String.html) columnLabel,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Updates the designated column using the given Reader object. The data will be read from the stream as needed until end-of-stream is reached. The JDBC driver will do any necessary conversion from UNICODE to the database char format.

The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of updateNClob which takes a length parameter.

**Parameters:**columnLabel - the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the columnreader - An object that contains the data to set the parameter value to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the columnLabel is not valid; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; this method is called on a closed result set; if a database access error occurs or the result set concurrency is CONCUR\_READ\_ONLY [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ResultSet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/sql/Ref.html)   [**NEXT CLASS**](http://docs.google.com/java/sql/ResultSetMetaData.html) | [**FRAMES**](http://docs.google.com/index.html?java/sql/ResultSet.html)    [**NO FRAMES**](http://docs.google.com/ResultSet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#2jxsxqh) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).