

## Scrollable ResultSets

```
Statement createStatement(  
    int resultSetType,  
    int resultSetConcurrency)
```

resultSetType:

`ResultSet.TYPE_FORWARD_ONLY`

- default; same as in JDBC 1.0
- allows only forward movement of the cursor
- when `rset.next()` returns false, the data is no longer available and the result set is closed.

`ResultSet.TYPE_SCROLL_INSENSITIVE`

- backwards, forwards, random cursor movement.
- changes made in the database are not seen in the result set object in Java memory.

`ResultSet TYPE_SCROLL_SENSITIVE`

- backwards, forwards, random cursor movement.
- changes made in the database are seen in the result set object in Java memory.

resultSetConcurrency:

```
ResultSet.CONCUR_READ_ONLY
```

This is the default (and same as in JDBC 1.0) and allows only data to be read from the database.

```
ResultSet.CONCUR_UPDATABLE
```

This option allows for the Java program to make changes to the database based on new methods and positioning ability of the cursor.

### **Example:**

```
Statement stmt = conn.createStatement(  
    ResultSet.TYPE_SCROLL_INSENSITIVE,  
    ResultSet.CONCUR_READ_ONLY);  
ResultSet rset = stmt.executeQuery(  
    "select * from cat");
```

`public boolean absolute(int row) throws SQLException`

- If the given row number is positive, this method moves the cursor to the given row number (with the first row numbered 1).
- If the row number is negative, the cursor moves to a relative position from the last row.
- If the row number is 0, an SQLException will be raised.

`public boolean relative(int row) throws SQLException`

- This method call moves the cursor a relative number of rows, either positive or negative. An attempt to move beyond the last row (or before the first row) in the result set positions the cursor after the last row (or before the first row).

`public boolean first() throws SQLException`

`public boolean last() throws SQLException`

`public boolean previous() throws SQLException`

`public boolean next() throws SQLException`

```
public boolean beforeFirst() throws SQLException
```

```
public boolean afterLast() throws SQLException
```

The following methods return True/False

```
public boolean isFirst() throws SQLException
```

```
public boolean isLast() throws SQLException
```

```
public boolean isAfterLast() throws SQLException
```

```
public boolean isBeforeFirst() throws SQLException
```

The following method retrieves the current row number:

```
public int getRow() throws SQLException
```

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