# JDBC - ResultSet

manipalglobal
education services

# INTRODUCTION

*ResultSet*

manipalglobal
education services

## LEARNING OBJECTIVES

At the end of this lesson, you will be able to:

- Understand attributes of ResultSet
- Illustrate the methods of ResultSet Interface
- Update a ResultSet
- Understand RowSet and its types

manipalglobal
education services

Refer package **com.mgait.jdbc** in the provided code base for demo programs on the topics covered in this presentation

The demo programs use the 'hr' schema of Oracle Express Edition

manipalglobal
education services

## ResultSet and its attributes

- ResultSet object contains the records/rows returned by query execution
- Each record in a ResultSet contains same number of columns
- ResultSet object maintains a cursor that points to a record/row

| | |
|---|---|
| IN | India |
| AU | Australia |

**RESULTSET**

- ResultSet has the following attributes
  - Type
  - Concurrency
  - Holdability

manipalglobal
education services

## ResultSet Types

- Type determines characteristic and abilities of the ResultSet as described below

| | |
|---|---|
| TYPE_FORWARD_ONLY | • ResultSet can only be navigated forward<br>• Cursor cannot be moved backward<br>• **Default** type of ResultSet |
| TYPE_SCROLL_INSENSITIVE | • ResultSet can be navigated both forward and backward (Scrollable)<br>• ResultSet is insensitive to changes in the underlying data source while it is open<br>• You can jump to a absolute position or a position relative to the current position |
| TYPE_SCROLL_SENSITIVE | • ResultSet can be navigated both forward and backward (Scrollable)<br>• ResultSet is sensitive to changes in the underlying data source while it is open<br>• You can jump to a absolute position or a position relative to the current position |

manipalglobal
education services

## ResultSet Concurrency

- Concurrency determines whether the ResultSet can be updated, or only read

| | |
|---|---|
| CONCUR_READ_ONLY | • ResultSet can only be Read<br>• **Default** concurrency of ResultSet |
| CONCUR_UPDATABLE | • ResultSet can be both read and updated |

manipalglobal
education services

## ResultSet Holdability

- Holdability determines if a ResultSet is closed when the commit() method of the underlying connection is called

| | |
|---|---|
| CLOSE_CURSORS_AT_COMMIT | • ResultSet instances are closed when commit() is called on the connection that created the ResultSet |
| HOLD_CURSORS_OVER_COMMIT | • ResultSet is kept open when commit() is called on the connection that created the ResultSet. |

manipalglobal
education services

## Setting ResultSet attributes

- The ResultSet attribute's are set while creating the Statement objects
- Connection Interface methods for Statement creation are used to set ResultSet attributes

```
createStatement() // defaults are set
createStatement(int type, int concurrency)
createStatement(int type, int concurrency, int holdability)
```

- To make a ResultSet read only, scrollable and insensitive, create the

```
Statement stmt = conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
```

- commit, create statement as

```
Statement stmt = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE, ResultSet.CLOSE_CURSORS_AT_COMMIT);
```

manipalglobal
education services

## Setting ResultSet attributes

- ResultSet attributes can also be set while creating the PreparedStatement and Callable Statement objects
- Connection Interface methods to set ResultSet attributes
  - While creating PreparedStatement

```
prepareStatement(String sql)
prepareStatement(String sql, int type, int concurrency)
prepareStatement(String sql, int type, int concurrency, int holdability)
```

  - While creating CallableStatement

```
prepareCall(String sql)
prepareCall(String sql, int type, int concurrency)
prepareCall(String sql, int type, int concurrency, int holdability)
```

manipalglobal
education services

## Methods in ResultSet Interface

- Methods of ResultSet Interface can be divided into following three categories :
  - Navigational methods - Used to move the cursor to a row in the Resultset
  - Retrieval methods - Used to retrieve the data from the current row
  - Update methods of ResultSet. - Used to insert/update/delete the data in current row

Updates done in the ResultSet can be transferred to the underlying database

manipalglobal
education services

## ResultSet – Navigational Methods

```
next() : boolean        - moves the cursor forward by one row
previous() : boolean    - moves the cursor backwards by one row
first() : boolean       - positions the cursor to the first row in the ResultSet
last() : boolean        - positions the cursor to the last row in the ResultSet
beforeFirst():void      - positions the cursor before the first row of ResultSet
afterLast() :void       - positions the cursor after the last row of ResultSet

Note: All the above methods except next() throw SQLException, if called on a
ResultSet of type TYPE_FORWARD_ONLY
```

```
getRow() : int          - Returns an int with the current position of the cursor
isFirst() : boolean     - Returns true if cursor is on the first row
isLast() : boolean      - Returns true if cursor is on the last row
isBeforeFirst() : boolean - Returns true if cursor is before the first row
isAfterLast() : boolean   - Returns true if cursor is after the last row
```

manipalglobal
education services

## ResultSet – Navigational Methods

```
relative(int rows) : moves the cursor relative to its current position

Example: Consider a ResultSet object "rs"
        rs.relative(4) : moves the cursor 4 rows ahead of the current position
        rs.relative(-2) - moves the cursor 2 rows previous to the current position
```

```
absolute(int rows)     : positions the cursor to the given row number

Example: Consider a ResultSet object "rs"
        rs.absolute(30) - moves the cursor to the 3oth row
        rs.absolute(-5) - move the cursor to the 5th row from the end of the Resultset
                          In a resultset of 50 rows, cursor will be moved to 46th row
```

**Note: relative(..) and absolute(..) methods throw SQLException, if called on a resultset of type TYPE_FORWARD_ONLY**

manipalglobal
education services

## ResultSet – Retrieval Methods

- To retrieve data from a ResultSet, the cursor has to be first positioned on the row
- After positioning, the column data can be retrieved using getXXX methods
- Every column datatype in a table has a corresponding get method
- getXXX methods take column name(String) or column index(int) as parameter

```
Consider ResultSet 'rs' with 1st column COUNTRY_ID and 2nd column COUNTRY_NAME
After cursor positioning, value in COUNTRY_NAME column can be retrieved as below

String countryName = rs.getString(COUNTRY_NAME);

              or

String countryName = rs.getString(2);
```

rs

| | |
|---|---|
| IN | India |
| AU | Australia |

manipalglobal
education services

## ResultSet – Retrieval Methods

- More retrieval methods in ResultSet

```
getShort(..)    : short
getInt(…)       : int
getFloat(..)    : float
getDouble(..)   : double
getLong(..)     : long
getDate(..)     : java.sql.Date
getTime(..)     : java.sql.Time
getTimestamp(..) : java.sql.Timestamp
getBlob(..)     : java.sql.Blob
getClob(..)     : java.sql.Clob
..

Refer Java API documentation for the complete list of retrieval methods
```

manipalglobal
education services

## ResultSet – Update Methods

- Updates to the table can be also made by updating values in the ResultSet
- For updates, ResultSet should be defined with concurrency CONCUR_UPDATABLE
- Updating rows in a table through ResultSet is a two step process
  - Move the cursor to the row and update column values using updateXXX methods
    - (updateString(..), updateFloat(..), updateDate(..) and so on)
  - Update the changes made in ResultSet row to the Table row us
    `rs` updateRow() method

| | |
|---|---|
| IN | India |
| AU | **Astrali a** |

rs.absolute(2);
rs.updateString("NAME","Australia")

| | |
|---|---|
| IN | India |
| AU | **Austral ia** |

rs.updateRow()

| ID *char* | NAME *varchar* | REGION_ ID *Number* |
|---|---|---|
| BE | Belgium | 1 |
| IN | India | 3 |
| AU | Australia | 3 |

manipalglobal
education services

## ResultSet – Update Methods

Demo
Class :
ResultSetUpdateDem

```
updateRow()
Updates the database with the new contents of the current row of ResultSet

cancelRowUpdates()
Cancels the updates made to the current row in this ResultSet

deleteRow()
Deletes the current row from ResultSet and from the database

moveToInsertRow()
Moves the cursor to the insert row.

insertRow()
inserts the contents of the insert row into ResultSet and into the database
```

17



refreshRow() : Refreshes the current row with its most recent value in the database.

manipalglobal
education services

## RowSet

- RowSet objects holds tabular data like ResultSet
- RowSet Interface is derived from the ResultSet interface and therefore share its capabilities
- RowSet adds following capabilities to ResultSet
  - Functions as java bean component with standard set of properties and an event notification mechanism
  - Add Scrollability and Updatability
- Advantages
  - It is easy and flexible to use
  - It is Scrollable and Updatable by default

manipalglobal
education services

## Types of RowSet

- Connected RowSet
  - Makes a connection to DBMS and maintains that connection throughout its life span
    - JdbcRowSet
- Disconnected RowSet
  - Makes a connection to a DBMS only to read in data or to write data back to the data source
  - After reading data from or writing data to its data source, the RowSet object disconnects from it
    - CachedRowSet
    - WebRowSet
    - FilteredRowSet
    - JoinRowSet

19



**Connected RowSet Objects**

Only one of the standard RowSet implementations is a connected RowSet object: JdbcRowSet. Always being connected to a database, a JdbcRowSet object is most similar to a ResultSet object and is often used as a wrapper to make an otherwise non-scrollable and read-only ResultSet object scrollable and updatable.

As a JavaBeans component, a JdbcRowSet object can be used, for example, in a GUI tool to select a JDBC driver. A JdbcRowSet object can be used this way because it is effectively a wrapper for the driver that obtained its connection to the database.

**Disconnected RowSet Objects**

The other four implementations are disconnected RowSet implementations. Disconnected RowSet objects have all the capabilities of connected RowSet objects plus they have the additional capabilities available only to disconnected RowSet objects. For example, not having to maintain a connection to a data source makes disconnected RowSet objects far more lightweight than a JdbcRowSet object or a ResultSet object. Disconnected RowSet objects are also serializable, and the combination of being both serializable and lightweight makes them ideal for sending data over a network. They can even be used for sending data to thin clients such as PDAs and mobile phones.

The CachedRowSet interface defines the basic capabilities available to all

disconnected RowSet objects. The other three are extensions of the CachedRowSet interface, which provide more specialized capabilities. The following information shows how they are related:

A CachedRowSet object has all the capabilities of a JdbcRowSet object plus it can also do the following:

Obtain a connection to a data source and execute a query

Read the data from the resulting ResultSet object and populate itself with that data

Manipulate data and make changes to data while it is disconnected

Reconnect to the data source to write changes back to it

Check for conflicts with the data source and resolve those conflicts

A WebRowSet object has all the capabilities of a CachedRowSet object plus it can also do the following:

Write itself as an XML document

Read an XML document that describes a WebRowSet object

A JoinRowSet object has all the capabilities of a WebRowSet object (and therefore also those of a CachedRowSet object) plus it can also do the following:

Form the equivalent of a SQL JOIN without having to connect to a data source

A FilteredRowSet object likewise has all the capabilities of a WebRowSet object (and therefore also a CachedRowSet object) plus it can also do the following:

Apply filtering criteria so that only selected data is visible. This is equivalent to executing a query on a RowSet object without having to use a query language or connect to a data source.

manipalglobal
education services

## JdbcRowSet Demo

Demo
Class :

```
JdbcRowSet rs = RowSetProvider.newFactory().createJdbcRowSet();
rs.setUrl("jdbc:oracle:thin:@localhost:1521:xe");
rs.setUsername("HR");
rs.setPassword("HR");

rs.setCommand("select * from countries");
rs.execute();

while(rs.next()){
    String id = rs.getString("COUNTRY_ID");
    String name = rs.getString("COUNTRY_NAME");

    System.out.println(id + "  "+ name);
}
```

manipalglobal
education services

What is the meaning of ResultSet.TYPE_SCROLL_INSENSITIVE

1. This means that the ResultSet is insensitive to scrolling
2. This means that the ResultSet is sensitive to scrolling, but insensitive to updates, i.e. not updateable
3. This means that the ResultSet is sensitive to scrolling, but insensitive to changes made by others
4. The meaning depends on the type of data source, and the type and version of the driver you use with this data source

manipalglobal
education services

## References

- Refer following demo videos on EduNxt
  - Resultset

manipalglobal
education services

## SUMMARY

*JDBC - ResultSet*

manipalglobal
education services

## SUMMARY

In this lesson, you've learned to:

- Create ResultSet with different attributes
- Use methods of ResultSet Interface
- Update a table through ResultSet
- Understand RowSet and its types