Batch Update

Is the technique in JDBC used to combine multiple queries and send to the database at once

Rowset

Advanced version of resultset

* Introduced in JDBC 2.0 (supports from JDK 1.4 onwards)
* Rowset is the child interface of Resultset.
* Rowset is defined in javax.sql package
* Rowset is the advanced version of Resultset used to manage the list of records received from the table

What is the difference between Resultset and Rowset?

|  |  |
| --- | --- |
| ResultSet | Rowset |
| Always moves in forward direction by default | Rowset is the scrollable by default |
| Result set allows read only by default | Row set allows update operation by default (updatable) |
| Result set is not serializable. So not possible to send the object across the network | Rowset is serializable by default |
| Result set is always connected   * that means DB connection is mandatory to retrieve the data from the resultset. * If DB connection is closed, then we can’t retrieve the data from resultset | Rowset is either connected or disconnected.  JDBCRowset 🡪 connected rowset  CachedRowset/Webrowset/Filtered rowset are disconnected rowset |

Types of Rowsets

|  |  |
| --- | --- |
| ConnectedRowset | DisconnectedRowset |
| * Similar to resultset * We can’t serialize the rowset object because we are not allowed to retrieve the data without DB connection | * Serializable – we can send this object across the network because disconnected rowset won’t require DB connection to retrieve the data |
| * JDBCRowset | * + - CachedRowset     - WebRowset     - JoinRowset     - FilterRowset |

Hierarchy of Rowset

Steps to create Rowset objects:

Step 1:

RowsetFactory rsf = RowSetProvider.newFactory();

Step 2: using rsf reference, we can create any type of rowset object

JdbcRowSet jrs = rsf.createJdbcRowSet();

CachedRowSet crs = rsf.createCachedRowSet();

WebRowSet wrs = rsf.createWebRowSet();

JoinRowSet jrs = rsf.createJoinRowSet();

FilteredRowSet frs = rsf.createFilteredRowset();

Step 3: provide the URL, username, password and select query using setter methods.

jrs.setUrl("jdbc:oracle:thin:@localhost:1521:orcl");

jrs.setUsername("system");

jrs.setPassword("manager");

jrs.setCommand("select htno,sname from student");

Step 4: call the execute method to get the rowset from database

jrs.execute()

Step 5: process the obtained rowset using reference variable of rowset.

while(jrs.next())

{

System.out.println(jrs.getInt(1) +"::" +

jrs.getString(2));

}

Note:

we can use the methods like absolute(), first(), last() and so on to navigate the control

Step 6: close the rowset using close()