|  |  |
| --- | --- |
| JoinRowSet | * It is the child interface of WebRowSet. * If we want to join rows from different rowsets into a single rowset based on matched column(common column) then we should go for JoinRowSet. * We can add RowSets to the JoinRowSet by using addRowSet() method.   addRowSet(RowSet rs,int commonColumnIndex); |

package com.sssit.product.controller;

import java.sql.SQLException;

import javax.sql.rowset.CachedRowSet;

import javax.sql.rowset.JoinRowSet;

import javax.sql.rowset.RowSetFactory;

import javax.sql.rowset.RowSetProvider;

public class JoinRowSetDemo {

public static void main(String[] args) {

try

{

RowSetFactory rsf = RowSetProvider.newFactory();

CachedRowSet empJdbc = rsf.createCachedRowSet();

CachedRowSet deptJdbc = rsf.createCachedRowSet();

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

empJdbc.setUsername("advdec23");

empJdbc.setPassword("advdec23");

empJdbc.setCommand("select empno,ename,job,deptno from employee");

empJdbc.execute();

/\*

\* while(empJdbc.next()) { System.out.printf("%10d%20s%10s%10d\n",

\* empJdbc.getInt(1),empJdbc.getString(2),

\* empJdbc.getString(3),empJdbc.getInt(4)); }

\*/

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

deptJdbc.setUsername("advdec23");

deptJdbc.setPassword("advdec23");

deptJdbc.setCommand("select departmentid,departmentname from department");

deptJdbc.execute();

JoinRowSet jrs = rsf.createJoinRowSet();

jrs.addRowSet(empJdbc, "deptno");

jrs.addRowSet(deptJdbc, "departmentid");

while(jrs.next()) {

System.out.printf("%10d%20s%10s%10s\n",

jrs.getInt(1),jrs.getString(2),

jrs.getString(3),jrs.getString("departmentname"));

}

}catch(SQLException e) {

System.out.println("SQL Exception.....");

}

}

}

|  |  |
| --- | --- |
| FilteredRowSet | * It is the child interface of WebRowSet. * If we want to filter rows based on some condition then we should go for FilteredRowSet. * We can define the filter by implementing Predicate interface. |

**package** com.sssit.product.controller;

**import** java.sql.SQLException;

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**import** javax.sql.RowSet;

**import** javax.sql.rowset.Predicate;

**class** SearchFilter **implements** Predicate {

**private** Pattern pattern;

**public** SearchFilter(String searchRegex) {

**if** (searchRegex != **null** && !searchRegex.isEmpty()) {

pattern = Pattern.*compile*(searchRegex);

}

}

**public** **boolean** evaluate(RowSet rs) {

System.***out***.println("SearchFilter.evaluate called ");

**try** {

**if** (!rs.isAfterLast()) {

String name = rs.getString("pname");

System.***out***.println(String.*format*(

"Searching for pattern '%s' in %s", pattern.toString(),

name));

Matcher matcher = pattern.matcher(name);

**return** matcher.matches();

} **else**

**return** **false**;

} **catch** (Exception e) {

e.printStackTrace();

**return** **false**;

}

}

@Override

**public** **boolean** evaluate(Object value, **int** column) **throws** SQLException {

// **TODO** Auto-generated method stub

**return** **false**;

}

@Override

**public** **boolean** evaluate(Object value, String columnName) **throws** SQLException {

// **TODO** Auto-generated method stub

**return** **false**;

}

}

package com.sssit.product.controller;

import java.sql.Connection;

import javax.sql.rowset.FilteredRowSet;

import javax.sql.rowset.RowSetFactory;

import javax.sql.rowset.RowSetProvider;

public class FilterRowSetDemo {

public static void main(String[] args) throws Exception {

RowSetFactory rsf = RowSetProvider.newFactory();

FilteredRowSet usersRS = rsf.createFilteredRowSet();

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

usersRS.setUsername("advdec23");

usersRS.setPassword("advdec23");

usersRS.setCommand("select \* from product");

usersRS.execute();

usersRS.setFilter(new SearchFilter("^[A-L].\*"));

while(usersRS.next()) {

System.out.printf("%10d%20s%10.2f\n",

usersRS.getInt(1),usersRS.getString(2),usersRS.getDouble(3));

}

}

}

Transaction Management

**package** com.sssit.product.controller;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.util.Scanner;

**import** com.sssit.product.util.JDBCConnection;

**public** **class** TransactionMgmt {

**public** **static** **void** main(String[] args) {

**try**

{

Connection con = JDBCConnection.*getConnection*();

con.setAutoCommit(**false**);

**final** String QUERY = "insert into login values(?,?,?)";

PreparedStatement pstmt = con.prepareStatement(QUERY);

Scanner ip = **new** Scanner(System.***in***);

System.***out***.println("Enter User Id?");

**int** uid = ip.nextInt();

System.***out***.println("Enter User Name?");

String uname = ip.next();

System.***out***.println("Enter Password?");

String passwd = ip.next();

pstmt.setInt(1, uid);

pstmt.setString(2, uname);

pstmt.setString(3, passwd);

pstmt.executeUpdate();

System.***out***.println("Do you want save?");

**char** ch = ip.next().charAt(0);

**if**(ch=='y')

con.commit();

**else**

con.rollback();

}

**catch**(Exception e) {

System.***out***.println(e);

}

}

}