CSCI 3357: Database System Implementation
Homework Assignment 11
Due Thursday, December 7

THIS ASSIGNMENT IS FOR EXTRA CREDIT: Its score will replace the lowest score from your previous assignments. Also, NO LATE SUBMISSIONS ACCEPTED.

1. In HW7 you added the methods `isNull`, `previous`, and `afterLast` to `TableScan`. I have added these methods to the `Scan` interface, which you should download. Your job is to implement these methods in the classes `SelectScan`, `ProjectScan`, and `ProductScan`.

[Technically, you should implement these methods in all of the classes in SimpleDB that implement `Scan`, but doing so is beyond the scope of this assignment. My revised version of `Scan` contains a default implementation of these methods, which will keep the compiler happy.]

2. The methods `previous` and `afterLast` are part of the standard JDBC `ResultSet` interface, but are not implemented by SimpleDB. You should fix this problem by adding implementations of these methods to the class `EmbeddedResultSet` in the package simpledb.jdbc.embedded. Be aware that these methods should `throw SQLException`, just like every other JDBC method.

3. The method `isNull` is not part of the JDBC `ResultSet` interface. Instead, JDBC has the method `wasNull`. This method takes no arguments, and returns `true` if the most recently retrieved value was a null. For example, the following JDBC code prints the name and major id of all students, printing "null" when a major id value is null.

```
ResultSet rs = stmt.executeQuery("select sname, majorid
                                  from student");
while (rs.next()) {
   String s = rs.getString("sname");
   int major = rs.getInt("majorid");
   if (rs.wasNull())
       System.out.println(s + " null");
   else
       System.out.println(s + " " + major);
}
```

Your task is to add the method `wasNull` to the class `EmbeddedResultSet`. To do this, modify the `getInt` and `getString` methods so that they call `isNull` to check if the retrieved value is null, and set a flag to indicate whether the value is null or not. Then your implementation of `wasNull` can simply check the flag.

When you are done, you should try out your enhanced version of JDBC. As an example, I wrote two clients that you can download. These clients use the student database produced by the CreateStudentDB client. The file *HW11Client.java* modifies the records in the STUDENT table so that all students having majorid=30 have a null graduation year, and then prints the records in reverse order. The file *HW11StudentMajor.java* prints the names of students and their majors, just like the original StudentMajor client but in reverse order.

Create a zip file containing the four files `SelectScan.java`, `ProductScan.java`, `ProjectScan.java`, and `EmbeddedResultSet.java`, and submit it to Canvas.

By the way, these modifications to JDBC only work for embedded connections. If you are interested, feel free to extend your modifications to network connections by modifying the classes `NetworkResultSet`, `RemoteResultSet`, and `RemoteResultSetImplementation` in the package simpledb.jdbc.network. Read sections 11.3-11.5 of the text for details.