

## Session (02.02) Interfacing MS Access Database using Java (2)

Dr. Ahmed M. ElShafee

Dr. Ahmed ElShafee, ACU Fall 2013, Practical Applications in CS1

- (3) Do you want to move forward AND backward through the records, but are not bothered about any changes made to the records
- Type number 1 on the list above is called a **TYPE\_FORWARD\_ONLY ResultSet**.
  - Number 2 on the list is a **TYPE\_SCROLL\_SENSITIVE ResultSet**.
  - The third ResultSet option is called **TYPE\_SCROLL\_INSENSITIVE**.
  - The ResultSet type goes between the round brackets of createStatement:

```
Statement s = conn.createStatement();
```

Dr. Ahmed ElShafee, ACU Fall 2013, Practical Applications in CS1

### ResultSets

- A **ResultSet** is a way to store and manipulate the records returned from a SQL query.
- **ResultSets** come in three different types.
- The type you use depends on what you want to do with the data:

```
ResultSet rs = s.getResultSet();
```

- 1) Do you just want to move forward through the records, from beginning to end?
- 2) Do you want to move forward AND backward through the records, as well as detecting any changes made to the

**records?**  
Dr. Ahmed ElShafee, ACU Fall 2013, Practical Applications in CS1

- Because we've left the round brackets empty, we'll get the default ResultSet, which is **TYPE\_FORWARD\_ONLY**.
- we'll use one of the other types.
- But you use them like this:

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

- So you first type the word ResultSet. After a dot, you add the ResultSet type you want to use.

Dr. Ahmed ElShafee, ACU Fall 2013, Practical Applications in CS1

- you also need to specify whether the **ResultSet** is Read Only or whether it is Updatable.
- You do this with two built-in constants: **CONCUR\_READ\_ONLY** and **CONCUR\_UPDATABLE**.
- Again, these come after the word RecordSet:

```
ResultSet.CONCUR_READ_ONLY
ResultSet.CONCUR_UPDATABLE
```

- The final code will be,...

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

- One more thing to get used to with **ResultSets** is something called a **Cursor**.
- A **Cursor** is really just a pointer to a table row.
- When you first load the records into a **ResultSet**, the **Cursor** is pointing to just before the first row in the table.
- You then use methods to manipulate the **Cursor**.
- But the idea is to identify a particular row in your table.

### Using a ResultSet

- Once you have all the records in a Results set, there are methods you can use to manipulate your records.
- Here are the methods you'll use most often:

<b>Next</b>	Moves the Cursor to the next row in your table. If there are no more rows in the table, a value of False will be returned.
<b>Previous</b>	Moves the Cursor back one row in your table. If there are no more rows in the table, a value of False will be returned.

<b>first</b>	Moves the Cursor to the first row in your table
<b>last</b>	Moves the Cursor to the last row in your table
<b>Absolute</b>	Moves the Cursor to a particular row in the table. So absolute( 5 ) will move the Cursor to row number 5 in the table

## javadbexample02

---

- Check lab manual

## update

---

- Using updatable result set,...

```
rs.updateInt("fieldname", intValue);  
rs.updateString( fieldnumber, stringValue);  
rs.updateRow();
```

## Javadbexample3

---

- Check lab manual

## Insert

---

- To insert new record

```
rs.moveToInsertRow();  
rs.updateInt(field name,number);  
rs.updateString(field name, string);  
rs.insertRow();
```

# Javadbexample4

---

- Check lab manual

13

Dr. Ahmed ElShafee, ACU Fall 2013, Practical Applications  
in CS I

# Assignment

---

- Re-write "JavaDBExample 02,03,04" programs again to work with Example02.accdb

14

Dr. Ahmed ElShafee, ACU Fall 2013, Practical Applications  
in CS I

---

Thanks,..  
See you next week (ISA),...

15

Dr. Ahmed ElShafee, ACU Fall 2013, Practical Applications  
in CS I