

JDBC Overview

Jeff Offutt

<http://www.cs.gmu.edu/~offutt/>

SWE 432

**Design and Implementation of
Software for the Web**

JDBC

- JDBC (Java Database Connectivity) API allows Java programs to connect to databases
- Database access is the same for all database vendors
- The JVM uses a JDBC driver to translate generalized JDBC calls into vendor specific database calls
- There are four general types of JDBC drivers
 - We will look at Type 4 ...

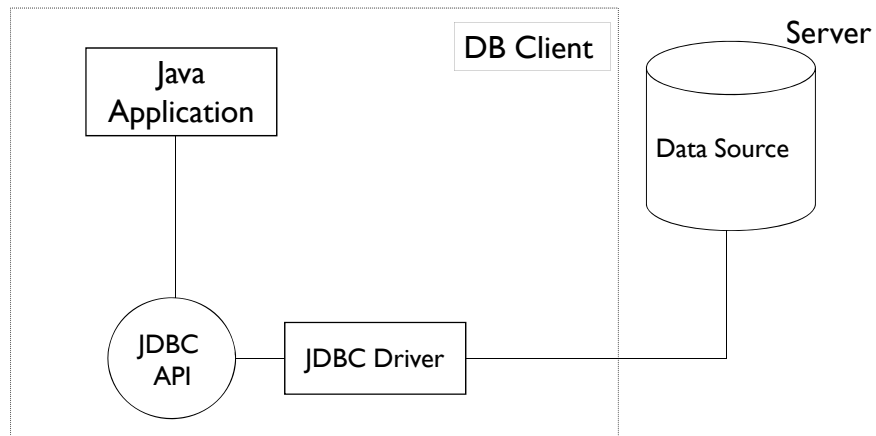
JDBC = Java Data Base Connectivity ?

- Many people believe that JDBC stands for *Java Data Base Connectivity*
- But not quite—here is the link from Sun stating that JDBC is not an acronym:
<http://java.sun.com/docs/books/jdbc/intro.html>
- Excerpt :
 - “JDBC (TM) is a Java (TM) API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for ‘Java Database Connectivity’.) ”

Pure Java Driver (Type 4)

- Converts the JDBC API calls to direct network calls
 - Using vendor-specific networking protocols
 - Making direct socket connections with the database
- It is the most efficient method to access database, both in performance and development time
- It is the simplest to deploy
- All major database vendors provide Type 4 JDBC drivers for their databases
 - Also available from third party vendors
- A list of JDBC drivers:
 - <http://industry.java.sun.com/products/jdbc/drivers>

Pure Java Driver (2)



1 December 2011

© Offutt, 2011

5

Typical JDBC Programming Procedure

1. Load the database driver
2. Obtain a connection
3. Create and execute statements
4. Use result sets to navigate through the results
5. Close the connection

12/1/2011

© Wu, Lee & Offutt

6

Driver Manager

- The `java.sql.DriverManger` class in JDBC provides a common access layer on top of different database drivers
- `DriverManager` requires that each driver used by the application must be registered before use
- Load the database driver using `ClassLoader` :
 - `Class.forName ("oracle.jdbc.driver.OracleDriver");`

12/1/2011

© Wu, Lee & Offutt

7

Connecting to a Database

- **Type I JDBC Driver**
 - `Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");`
 - `Connection con = DriverManager.getConnection ("jdbc:odbc:dsn");`
- **Type 4 JDBC Driver**
 - **Oracle Server**
`Class.forName ("oracle.jdbc.driver.OracleDriver");`
`con = DriverManager.getConnection (`
 `"jdbc:oracle:thin:@apollo.ite.gmu.edu:1521:ite8i",`
 `"swe432", "swe432");`
 - **MySQL Server**
`Class.forName ("org.gjt.mm.mysql.Driver");`
`con = DriverManager.getConnection`
 `("jdbc:mysql://localhost/databasename", uid, passwd);`

12/1/2011

© Wu, Lee & Offutt

8

Creating and Executing SQL Statements

- Statement object
 - Statement statement = con.createStatement();
 - statement.executeUpdate ("CREATE TABLE STUDENT");
 - statement.executeQuery ("SELECT * FROM STUDENT");

12/1/2011

© Wu, Lee & Offutt

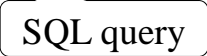
9

Creating Tables Examples

- Creating a Coffee table

```
CREATE TABLE COFFEES (COF_NAME VARCHAR(32), SUP_ID  
INTEGER, PRICE FLOAT, SALES INTEGER, TOTAL INTEGER)
```
- Creating JDBC statements

```
Statement stmt = con.createStatement ();
```


- Execute a statement

```
stmt.executeUpdate ("CREATE TABLE COFFEES " +  
"(COF_NAME VARCHAR(32), SUP_ID INTEGER, PRICE FLOAT," +  
"SALES INTEGER, TOTAL INTEGER)");
```

12/1/2011

© Wu, Lee & Offutt

10

Checking to See if a Table Exists

Use *meta* data about the table

```
DatabaseMetaData dmd = con.getMetaData();
ResultSet rs = dmd.getTables (null, null, "COFFEES", null);
if (rs == null)
{
    // table does not exist, create it.
}
```

12/1/2011

© Wu, Lee & Offutt

11

Execute Statements

- This uses `executeUpdate` because the SQL statement contained in `createTableCoffees` is a data definition language (DDL) statement
- DDL statements are executed with `executeUpdate`
 - Create a table
 - Alter a table
 - Drop a table
- `executeUpdate` is also used to execute SQL statements that update a table

12/1/2011

© Wu, Lee & Offutt

12

Execute Statements

- `executeUpdate` is used more often to update tables than it is to create them
 - We create a table once but updated it many times
- `executeQuery` is used most often to execute SQL statements
- `executeQuery` is used to execute `SELECT` statements
 - `SELECT` statements are the most common SQL statements

12/1/2011

© Wu, Lee & Offutt

13

Entering Data to a Table

```
Statement stmt = con.createStatement();
stmt.executeUpdate ( "INSERT INTO COFFEES" + "VALUES
('Colombian', 101, 7.99, 0, 0)");
stmt.executeUpdate ("INSERT INTO COFFEES" + "VALUES
('French_Roast', 49, 8.99, 0, 0)");
stmt.executeUpdate ("INSERT INTO COFFEES" + "VALUES
('Espresso', 150, 9.99, 0, 0)");
stmt.executeUpdate ("INSERT INTO COFFEES" + "VALUES
('Colombian_Decaf', 101, 8.99, 0, 0)");
stmt.executeUpdate ("INSERT INTO COFFEES" + "VALUES
('French_Roast_Decaf', 49, 9.99, 0, 0)");
```

12/1/2011

© Wu, Lee & Offutt

14

Getting and Using Data From Tables

```

ResultSet rs = stmt.executeQuery ("SELECT COF_NAME, PRICE
FROM COFFEES");
while (rs.next())
{
    String s = rs.getString ("COF_NAME");
    float n = rs.getFloat ("PRICE");
    System.out.println (s + " " + n);
}

```

12/1/2011

© Wu, Lee & Offutt

15

Navigating Result Sets

| JDBC Types Mapped to Java Types | | | | |
|---------------------------------|------------|--|------------|--------------------------|
| JDBC Types | Java Types | | JDBC Types | Java Types |
| CHAR | String | | DATE | Java.sql.Date |
| VARCHAR | String | | TIME | Java.sql.Time |
| LONGVARCHAR | String | | TIMESTAMP | Java.sql.Timestamp |
| TINYINT | short | | | |
| INTEGER | Int | | JAVAOBJECT | Object |
| BIGINT | Long | | BLOB | Java.sql.Blob interface |
| REAL | Float | | CLOB | Java.sql.Clob interface |
| FLOAT | Double | | ARRAY | Java.sql.Array interface |
| DOUBLE | Double | | STRUCT | |
| BIT | boolean | | REF | |

12/1/2011

© Wu, Lee & Offutt

16

Navigating Result Sets

```
ResultSet rs = statement.executeQuery ("select * from student");
while (rs.next())
{
    System.out.print (rs.getString ("name") + ...
}
```

```
ResultSetMetaData metaData = rs.getMetaData();
metaData.getColumnCount();
metaData.getColumnName (i);
metaData.getColumnType (i);
```

12/1/2011

© Wu, Lee & Offutt

17

Navigating Result Sets

- By default, a ResultSet object is read-only (not updateable) and has a cursor that moves forward only (next()).
- Scrollable result sets have more operations
First, last, absolute, relative, next, previous, beforeFirst, afterLast, isFirst, isBeforeFirst, isLast, isAfterLast

12/1/2011

© Wu, Lee & Offutt

18

JDBC Summary

- Most large web applications use databases to make data persistent
- The techniques for accessing databases from Java programs are identical in web applications as in stand-alone Java programs
- This lecture does not teach how to setup or use a database or the details of constructing SQL queries