Talking to database (JDBC)

**Establishing a Connection**
The first thing you need to do is establish a connection with the DBMS you want to use. This involves two steps: (1) **loading the driver** and (2) **making the connection**.

**Loading Drivers**
Loading the driver or drivers you want to use is very simple and involves just one line of code. If, for example, you want to use the JDBC-ODBC Bridge driver, the following code will load it:

```java
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

**Making the Connection**
The second step in establishing a connection is to have the appropriate driver connect to the DBMS. The following line of code illustrates the general idea:

```java
Connection con = DriverManager.getConnection(url, "myLogin", "myPassword");
```

```java
String url = "jdbc:odbc:Fred";
Connection con = DriverManager.getConnection(url, "Fernanda", "J8");
```

**Creating a Table**

<table>
<thead>
<tr>
<th>COFFEES</th>
<th>COF_NAME</th>
<th>SUP_ID</th>
<th>PRICE</th>
<th>SALES</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Colombian</td>
<td>101</td>
<td>7.99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>French_Roast</td>
<td>49</td>
<td>8.99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Espresso</td>
<td>150</td>
<td>9.99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Colombian_Decaf</td>
<td>101</td>
<td>8.99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>French_Roast_Decaf</td>
<td>49</td>
<td>9.99</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
SUPPLIERS

<table>
<thead>
<tr>
<th>SUP_ID</th>
<th>SUP_NAME</th>
<th>STREET</th>
<th>CITY</th>
<th>STATE</th>
<th>ZIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Acme, Inc.</td>
<td>99 Market Street</td>
<td>Groundsville</td>
<td>CA</td>
<td>95199</td>
</tr>
<tr>
<td>49</td>
<td>Superior Coffee</td>
<td>1 Party Place</td>
<td>Mendocino</td>
<td>CA</td>
<td>95460</td>
</tr>
<tr>
<td>150</td>
<td>The High Ground</td>
<td>100 Coffee Lane</td>
<td>Meadows</td>
<td>CA</td>
<td>93966</td>
</tr>
</tbody>
</table>

CREATE TABLE COFFEES
(   COF_NAME VARCHAR(32),
   SUP_ID INTEGER,
   PRICE FLOAT,
   SALES INTEGER,
   TOTAL INTEGER)

String createTableCoffees = "CREATE TABLE COFFEES " +
   "(COF_NAME VARCHAR(32), SUP_ID INTEGER, PRICE FLOAT, " +
   "SALES INTEGER, TOTAL INTEGER)";

Creating JDBC Statements
A Statement object is what sends your SQL statement to the DBMS

Statement stmt = con.createStatement();

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

stmt.executeUpdate(createTableCoffees);
**Entering Data into a Table**

```java
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)");
```

**Getting Data from a Table**

```sql
SELECT * FROM COFFEES
```

The result, which is the entire table, will look similar to the following:

<table>
<thead>
<tr>
<th>COF_NAME</th>
<th>SUP_ID</th>
<th>PRICE</th>
<th>SALES</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colombian</td>
<td>101</td>
<td>7.99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>French_Roast</td>
<td>49</td>
<td>8.99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Espresso</td>
<td>150</td>
<td>9.99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Colombian_Decaf</td>
<td>101</td>
<td>8.99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>French_Roast_Decaf</td>
<td>49</td>
<td>9.99</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

```sql
SELECT COF_NAME, PRICE FROM COFFEES
```

The results of this query will look something like this:

<table>
<thead>
<tr>
<th>COF_NAME</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colombian</td>
<td>7.99</td>
</tr>
<tr>
<td>French_Roast</td>
<td>8.99</td>
</tr>
<tr>
<td>Espresso</td>
<td>9.99</td>
</tr>
<tr>
<td>Colombian_Decaf</td>
<td>8.99</td>
</tr>
<tr>
<td>French_Roast_Decaf</td>
<td>9.99</td>
</tr>
</tbody>
</table>
SELECT COF_NAME, PRICE
FROM COFFEES
WHERE PRICE < 9.00

The results would look similar to this:

COF_NAME       PRICE
---------------------
Colombian      7.99
French_Roast   8.99
Colombian Decaf 8.99

Retrieving Values from Result Sets

ResultSet rs = stmt.executeQuery(
    "SELECT COF_NAME, PRICE FROM COFFEES");

Using the get<data_type> Methods

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

The output will look something like this:

Colombian  7.99
French_Roast  8.99
Espresso  9.99
Colombian Decaf  8.99
French_Roast Decaf  9.99
String s = rs.getString("COF_NAME");  
String s = rs.getString(1);  
float n = rs.getFloat(2);  

String updateString = "UPDATE COFFEES " +  
"SET SALES = 75 " +  
"WHERE COF_NAME LIKE 'Colombian'";

The table COFFEES will now look like this:

<table>
<thead>
<tr>
<th>COF_NAME</th>
<th>SUP_ID</th>
<th>PRICE</th>
<th>SALES</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>Colombian</td>
<td>101</td>
<td>7.99</td>
<td>75</td>
<td>0</td>
</tr>
<tr>
<td>French_Roast</td>
<td>49</td>
<td>8.99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Espresso</td>
<td>150</td>
<td>9.99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Colombian_Decaf</td>
<td>101</td>
<td>8.99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>French_Roast_Decaf</td>
<td>49</td>
<td>9.99</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

String query = "SELECT COF_NAME, SALES FROM COFFEES " +  
"WHERE COF_NAME LIKE 'Colombian'";
ResultSet rs = stmt.executeQuery(query);
while (rs.next()) {
    String s = rs.getString("COF_NAME");
    int n = rs.getInt("SALES");
    System.out.println(n + " pounds of " + s +  
                        " sold this week.");
}

Now let's update the TOTAL column by adding the weekly amount  
sold to the existing total, and then let's print out the number of  
pounds sold to date:

String updateString = "UPDATE COFFEES " +  
"SET TOTAL = TOTAL + 75 " +  
"WHERE COF_NAME LIKE 'Colombian'";
stmt.executeUpdate(updateString);
String query = "SELECT COF_NAME, TOTAL FROM COFFEES " +  
"WHERE COF_NAME LIKE 'Colombian'";
ResultSet rs = stmt.executeQuery(query);
while (rs.next()) {
    String s = rs.getString(1);
    int n = rs.getInt(2);
    System.out.println(n + " pounds of " + s + " sold to date.");
}