

More SQL – Insert, Update, and Delete

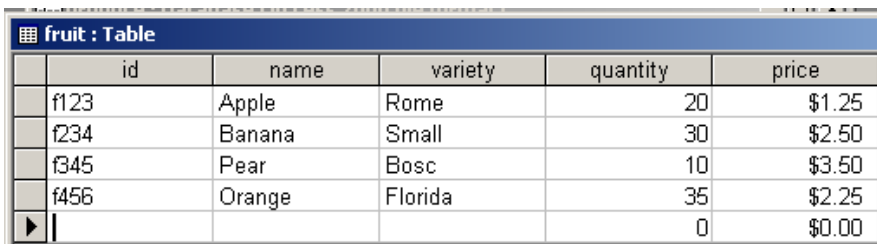
There are many kinds of queries that can be constructed using SQL. The basic ones are Select, Insert, Update, and Delete. We have used Select before. A brief description of the other three follows. Instead of `stmt.executeQuery (query)`, these three use `stmt.executeUpdate (query)`. Also, the method, `executeUpdate` returns an integer value. If the update is successful, the method returns the number of rows affected (the row count). Otherwise it returns 0.

Insert

The insert query is used to add a row to the database. An example that adds a row to a table called fruit is

```
String query = "Insert into Fruit Values ('f456', 'Orange', 'Florida', 35, 2.25)";
```

The fields of the table are: id, name, variety, quantity, and price. The first three are strings, so they are surrounded by quotes. The fields, quantity and price are numeric, so they do not have quotes around them. A picture of the table after the insertion is shown below.



id	name	variety	quantity	price
f123	Apple	Rome	20	\$1.25
f234	Banana	Small	30	\$2.50
f345	Pear	Bosc	10	\$3.50
f456	Orange	Florida	35	\$2.25
f456	Orange	Florida	0	\$0.00

When the data are given by variables, the typical situation, the quotes have to be separately included in the query. This is a common source of errors. For the example above, the resulting query looks like the following:

```
String query = "Insert Into fruit Values ('" + id + "', '" + name + "', '" + variety + "', " + quantity + "', '" + price + "')";
```

Note the placement of single quotes inside of double quotes for the strings, id, name, and variety, and the absence of single quotes for quantity and price.

The insert statement returns an integer value that is 0 if the update failed and is the number of rows inserted if it succeeded. Since we normally insert just one row, it will be 0 if nothing is done and 1 if the data is inserted properly.

A `doGet` method for a servlet that inserts a row into a database called *produce* with a table called *fruit* is shown below. Note that it first gets the data from the webpage using the request parameter and then creates and executes an update query.

```
// Insert a new product into the database.
public void doGet (HttpServletRequest request, HttpServletResponse response)
{
    try
    {
        PrintWriter out = response.getWriter ();
        Page.createHeader (out, "Produce");
```

```

Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con = DriverManager.getConnection ("jdbc:odbc:produce");

// Get parameters from web page.
String id = request.getParameter ("id");
String name = request.getParameter ("name");
String variety = request.getParameter ("variety");
int quantity = Integer.parseInt (request.getParameter ("quantity"));
double price = Double.parseDouble (request.getParameter ("price"));

// Create Insert query statement.
Statement stmt = con.createStatement ();
String query = "Insert Into Fruit Values (" + id + ", " + name + ", "
              + variety + ", "+ quantity + ", " + price + ")";

// Execute update query.
int success = stmt.executeUpdate (query);
stmt.close ();

// Show whether or not the update was successful.
if (success == 0) out.println ("Insert error.");
else out.println ("Data inserted.");
Page.createFooter (out);
} catch (IOException ex) {System.out.println ("IO Exception.");}
catch (ClassNotFoundException e) {System.out.println ("Class Not Found.");}
catch (SQLException es) {System.out.println ("SQL Exception");}
} // doGet

```

Update

The update query is somewhat similar to the Insert query. Update is used to change an entry in the database, such as a price. The query must include a key to indicate which row is to be changed. For example, to change the price of a product in the row with a given id, we can use the following query:

```
"Update fruit Set price = " + newPrice + " Where id = " + id + """;
```

```

// Change the price of a product in the database.
public void doGet (HttpServletRequest request, HttpServletResponse response)
{
    try
    {
        PrintWriter out = response.getWriter ();
        Page.createHeader (out, "Change Price");

        Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection con = DriverManager.getConnection ("jdbc:odbc:produce");

        // Get the parameters from the request.
        String id = request.getParameter ("id");
        double newPrice = Double.parseDouble (request.getParameter ("price"));

        // Create a query and execute the update.
        Statement stmt = con.createStatement ();

```

```

String query = "Update fruit Set price = " + newPrice + " Where id = " + id + """;
int success = stmt.executeUpdate (query);
stmt.close ();

// Check whether or not the updated was carried out.
if (success != 0) out.println ("Price Changed.");
else out.println ("Error, price not changed.");

Page.createFooter (out);
} catch (IOException ex) {System.out.println ("IO Exception.");}
catch (ClassNotFoundException e) {System.out.println ("Class Not Found Exception.");}
catch (SQLException es) {System.out.println ("SQL Exception");}
} // doGet

```

Delete

The delete command is used to delete a row in the database. Again a key field is used to determine the appropriate row. The query here is "Delete From fruit Where id = " + keyId + """. This deletes the row with the id provided by the user.

```

// Delete a row from the database.
public void doGet (HttpServletRequest request, HttpServletResponse response)
{
    try
    {
        PrintWriter out = response.getWriter ();
        Page.createHeader (out, "Delete a Product");

        Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection con = DriverManager.getConnection ("jdbc:odbc:produce");

        // Get the id for the product to be deleted.
        String keyId = request.getParameter ("keyId");

        // Create a query and execute the update.
        Statement stmt = con.createStatement ();
        String query = "Delete From " + table + " Where id = " + keyId + """;
        int success = stmt.executeUpdate (query);
        stmt.close ();

        // Indicate whether or not the deletion was successful.
        if (success == 0) out.println ("ID not found");
        else out.println ("Row deleted.");
        Page.createFooter (out);
    } catch (IOException ex) {System.out.println ("IO Exception.");}
    catch (ClassNotFoundException e) {System.out.println ("Class Not Found Exception.");}
    catch (SQLException es) {System.out.println ("SQL Exception");}
} // doGet

```