**Updating and Deleting Records**

So far, you've seen how to insert new records into a database using HTML forms. If you want to allow your store manager to update existing products, it's a little different.

Not only do you have to create the HTML form, but you must be able to populate the form with the existing data in the database. When the store manager is done making changes, you must provide a button that allows him or her to send the changes back to the database.

If you remember from earlier, the HTML form used to input data into the products table uses the text input type. To set a default value for a text input type, all you need is to define it using the *value* attribute:

echo "<input type=\"text\" name=\"price\" value=\"$price\">\n";

When the form appears, the browser shows the value data inside the textbox. If the store manager clicks the button to send the data, the default data is sent back as part of the form data.

Notice that even though the price data field is a numerical value, the HTML form still treats it as a text value. The value the form returns is also a text value, which works fine, since you'll use the text version of the price to create the SQL string needed to update the value in the table.

The editproducts.inc.php program passed the URL admin.php?content=updateproduct to the showproducts() function. When the showproducts() function lists the products, it creates a link for the product description that points to the updateproduct.inc.php Web page. This page paints the HTML form for our store manager.

Let's create this code now.

1. Create a file called *updateproduct.inc.php* in the *admin* folder in the store folder.
2. Open the file in a text editor, and enter the following code:

<?php

$prodid = $\_GET['id'];

$query = "SELECT prodid, catid, description, price, quantity, onsale FROM products where prodid = $prodid";

$result = mysql\_query($query);

$row = mysql\_fetch\_array($result, MYSQL\_ASSOC);

$catid = $row['catid'];

$description = $row['description'];

$price = $row['price'];

$quantity = $row['quantity'];

$onsale = $row['onsale'];

$query = "SELECT name FROM categories WHERE catid = $catid";

$result=mysql\_query($query);

$row = mysql\_fetch\_array($result, MYSQL\_ASSOC);

$catname = $row['name'];

echo "<h2>Update Product Information</h2>\n";

echo "<form enctype=\"multipart/form-data\" action=\"admin.php\" method=\"post\">\n";

echo "<input type=\"hidden\" name=\"content\" value=\"changeproduct\">\n";

echo "<input type=\"hidden\" name=\"prodid\" value=\"$prodid\">\n";

echo "<table width=\"100%\" cellpadding=\"1\" border=\"1\">\n";

echo "<tr><td><h3>Product ID</h3></td><td>$prodid</td></tr>\n";

echo "<tr><td><h3>Category</h3></td>\n";

echo "<td><select name=\"catid\">\n";

$query="SELECT catid,name from categories";

$result=mysql\_query($query);

while($row=mysql\_fetch\_array($result,MYSQL\_ASSOC))

{

$tempcatid = $row['catid'];

$name = $row['name'];

if ($tempcatid == $catid)

echo "<option value=\"$tempcatid\" selected=\"selected\">$name</option>\n";

else

echo "<option value=\"$tempcatid\">$name</option>";

}

echo "</select></td></tr>\n";

echo "<tr><td><h3>Description</h3></td><td><input type=\"text\" name=\"description\" value=\"$description\"></td></tr>\n";

echo "<tr><td><h3>Price</h3></td><td><input type=\"text\" name=\"price\" value=\"$price\"></td></tr>\n";

echo "<tr><td><h3>Quantity</h3></td><td><input type=\"text\" name=\"quantity\" value=\"$quantity\"></td></tr>\n";

if ($onsale)

echo "<tr><td><h3>On Sale</h3></td><td><input type=\"checkbox\" name=\"onsale\" value=\"1\" checked></td></tr>\n";

else

echo "<tr><td><h3>On Sale</h3></td><td><input type=\"checkbox\" name=\"onsale\" value=\"1\"></td></tr>\n";

echo "<tr><td><h3>Image</h3></td><td><img src=\"showimage.php?id=$prodid\" width=\"80\" height=\"60\"></td></tr>\n";

echo "<tr><td><h3>Update image</h3></td><td><input type=\"file\" name=\"picture\"></td></tr>\n";

echo "</table>\n";

echo "<input type=\"submit\" name=\"button\" value=\"Update\">\n";

echo "<input type=\"submit\" name=\"button\" value=\"Delete Product\">\n";

echo "</form>\n";

?>

That's a lot of code! You should recognize most of it, though. The first thing the code does is extract the product ID value passed by the link. It then queries the products table to retrieve the existing information for the product.

The code uses that information to populate the HTML form, one entry for each data field. There are a couple of cool things to see here:

* + It uses a select input type form to select a new category for the product. When it creates the drop-down box, it marks the current data value for the catid field as *selected*. That makes it the default value in the drop-down box.
  + The *onsale* data field is created as a check box. If the onsale value is true, the check box is marked as *checked*. If the value is false, the check box isn't checked. This is a great way to indicate Boolean data type values on a form. The Web site visitor can easily see if the value is set or not.
  + The current image is displayed in the form, and another field to enter a new image name is added. If the manager leaves this entry blank, it'll keep the existing image.
  + There are two submit buttons for the form. One indicates we want to update the information, and the other indicates we want to remove the product from the database.

Okay, now that the form is there, we need something to retrieve the data from the form and update the database information accordingly. That's the job of the changeproduct.inc.php file:

* + Create a file called *changeproduct.inc.php* in the *admin* folder, under the store area.
  + Enter the following code into the file:

<?php

$delete = $\_POST['button'];

if ($delete == "Delete Product")

{

$prodid = $\_POST['prodid'];

$query = "DELETE from products WHERE prodid = $prodid";

$result = mysql\_query($query);

if ($result)

{

echo "<h2>Product: $prodid deleted</h2>\n";

exit;

} else

{

echo "<h2>Problem deleting $prodid</h2>\n";

exit;

}

} else

{

$prodid = $\_POST['prodid'];

$catid = $\_POST['catid'];

$description = $\_POST['description'];

$price = $\_POST['price'];

$quantity = $\_POST['quantity'];

if (get\_magic\_quotes\_gpc())

{

$description = stripslashes($desription);

}

$description = mysql\_real\_escape\_string($description);

if (isset($\_POST['onsale']))

$onsale = 1;

else

$onsale = 0;

$PictName = $\_FILES['picture']['name'];

if ($PictName)

{

$thumbnail = getThumb($\_FILES['picture']);

$thumbnail = mysql\_real\_escape\_string($thumbnail);

$query = "UPDATE products SET catid='$catid', description = '$description', " .

"price = $price, quantity = $quantity, onsale = $onsale, picture = '$thumbnail' " .

"WHERE prodid = $prodid";

}

else

{

$query = "UPDATE products SET catid='$catid', description = '$description', " .

"price = $price, quantity = $quantity, onsale = $onsale " .

"WHERE prodid = $prodid";

}

$result = mysql\_query($query) or die(mysql\_error());

if ($result)

{

echo "<h2>Product information changed.</h2>\n";

}

else

{

echo "<h2>Sorry, I could not change the product information.</h2>\n";

}

}

?>

The first thing the code needs to detect is which submit button the manager selected in the form. In the form, you assigned both buttons the same name but gave them different values. All you need to do is retrieve the data for the button name and determine which value appears.

If the manager selects the button to remove the record, the code creates a *DELETE* SQL statement and submits it to the server. You must be careful when working with DELETE statements. By default, a DELETE statement that has no WHERE clause will delete every record in your database! That little tidbit has bitten quite a few database programmers. The WHERE clause specifies which records you want deleted. In your case, you must specify the product ID you want to delete.

If the manager selects the Update button, you must retrieve all of the posted data from the HTML form. Again, there are a couple of interesting things happening here as well:

* + - The onsale check box is only set if there's a check in it. Use the isset() function to determine if that variable is set or not. If so, assign a true value (1) to the variable.
    - If a new image is set in the image field, the $\_FILES[] value will be set. You assign the ['name'] element to a variable and check if it exists. If it does exist, you need to create a new thumbnail using your friend getThumb(), then use that value along with the other data values to update the record. If no new image file was specified, you can leave the picture BLOB alone and just replace the other data.

The UPDATE SQL statement updates data in records. Using the set clause, you can set individual data field values:

UPDATE products SET catid=1, description = 'test', price =1.25, quantity =10, onsale = 1 WHERE prodid = 1;

Again, just like the DELETE statement, the WHERE clause is important. It defines which records get the updated data. Without it, the UPDATE statement updates all of the records with the new data, which would not be a very good thing.

The UPDATE statement can also use equations to update the data fields:

UPDATE products SET quantity = quantity - 5 WHERE prodid = 10;

This statement reduces the quantity data field by five for the product with prodid of 10.

Once you have your code created, give it a test run. You should be able to update any data field in the products table using the HTML form, including specifying a new image for the product.

Updating product information

That's quite enough for one day! Let's go on to Chapter 5 and finish things up.