Section 5 Working with Multiple Inserts and More MySQL Queries

Create the third part of the Inventory Management system

- Create the functionality to implement the Customer portion of the system

- Create a Sale
- Process a Return

Section 5 Working with Multiple Inserts and More MySQL Queries

Processing Sale Data Input Table

<input type="text" name="InvID"> for each inventory entry

How to read the data inputs?
Use: quantity = int(formdata.getfirst(str(InvID)))

Build array of inputs with values and save to database with INSERT statement & cur.executemany(sql, data) where data is a list of tuples

```
sql = 'INSERT INTO sales (Quantity, InvID, CustID) VALUES (%s, %s, %s)'
cur.executemany(sql, data)
```

Section 5 Working with Multiple Inserts and More MySQL Queries

How to update data in a record.

```
UPDATE table SET ColumnName = Value [where ColumnNameX = value]
```

Updates are used to alter values in an existing record:

```
sql = 'UPDATE inventory SET QtyOnHand = %s where InvID = %s'
cur.execute(sql, [quantityonhand, invid])
```

updates in inventory record to set a new value for the quantity of inventory on hand.

Section 5 Working with Multiple Inserts and More MySQL Queries

Examples of complex SELECT statements

```
   sql = 'SELECT COUNT(*) from customer where FirstName = %s and LastName = %s'
   cur.execute(sql, name)
   result=cur.fetchone()    # Quick method to test if customer name is in database

   sql = 'SELECT sales.InvID, sum(sales.Quantity), inventory.ProductName FROM sales, inventory  WHERE sales.custid=%s and sales.InvID = inventory.InvID group by inventory.InvID'
   cur.execute(sql,custid)
```

This is an example of pulling data from two tables (a JOIN). In this case, I'm using it to get the quantity of each inventory item that a customer has purchased over the customer's shopping history.

Section 5 Working with Multiple Inserts and More MySQL Queries

Take some time to write the next portion of the inventory management system – record sales / process returns. Remember to adjust inventory on hand when sales and returns are processed and to adjust quantity purchased when doing a return.

Once you have finished the assignment (or if you need help), the next section will be a code walk-through of our solution for this portion of the system. It is also available in the downloads section for you to study.

Before you begin, let's take a look at the way this section operates in the browser.