Section 4 Working with Inserting, Updating & Selecting Data

Create the second part of the Inventory Management system

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

- Add Inventory to Database
- Edit Existing Inventory Record
- Delete Existing Inventory Record

Section 4 Working with Inserting, Updating & Selecting Data

MySQL Statements and Python functions you will need to know:

Retrieve data from database: Done with MySQL SELECT statement and Python .execute() function

Basic MySQL syntax: SELECT ColumnNames from tablename [where CONDITIONALS]
(Note: This is a very basic select statement. They can get quite sophisticated. We will demonstrate some of that later in the course.)

Sample Code:

```
record = int(formdata.getfirst('record', ''))
sql = 'SELECT * from inventory where InvID = %s'
cur.execute(sql, record)
```

Section 4 Working with Inserting, Updating & Selecting Data

MySQL Statements and Python functions you will need to know:

Insert data into database: Done with MySQL INSERT statement and Python .execute() function

Basic MySQL syntax: INSERT INTO tablename (ColumnNames) VALUES (ColumnValues)

Sample Code:

```
sql = '''INSERT INTO inventory (SKU, ProductName, QtyOnHand, Cost, RetailPrice) VALUES (%s, %s, %s, %s, %s)'''
record =  (sku, productname, qtyonhand, cost, retailprice)
cur.execute(sql, record)
```

Section 4 Working with Inserting, Updating & Selecting Data

MySQL Statements and Python functions you will need to know:

Update data in database: Done with MySQL UPDATE statement and Python .execute() function

Basic MySQL syntax: UPDATE tablename SET column1=val1, column2=val2 WHERE conditionals

Sample Code:

```
sql = '''UPDATE inventory SET SKU=%s, ProductName=%s, QtyOnHand=%s, Cost=%s, RetailPrice=%s WHERE InvID=%s'''
record =  (sku, productname, qtyonhand, cost, retailprice, invid)
cur.execute(sql, record)
```

Section 4 Working with Inserting, Updating & Selecting Data

MySQL Statements and Python functions you will need to know:

Accessing data one you have SELECTed it

Continuing with code from last page:

cur.fetchone() - fetches the next row in the query result – or the only row if only one was returned
cur.fetchmany(size) -fetches the next "size" rows of the query result. Data returned is a list of tuples.
cur. fetchall() - fetches all the rows from the query. Data returned is a list of tuples.

Sample Code:

```
#      for row in cur:    --- Both methods work but this one is Database Adapter implementation dependent
      for row in cur.fetchall():
        print ('<input type="radio" name="record" value="', row[0], '"> SKU: ', row[1], ' - ', row[2], '<br>', sep='')
```

Section 4 Working with Inserting, Updating & Selecting Data

Take some time to write the next portion of the inventory management system – add/edit/delete inventory

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.