Section 3 Setting Up Your Data Tables

Create the first part of the Inventory Management system

- Create two modules – one for HTML functions and one for MySQL functions
- Create a setup program to build the data tables
- Create a main program that has the stubs for Inventory, Sales, Returns, Reports and exiting the system.
- Be sure to include appropriate exception logic (i.e. use try... except) where appropriate

- Important: Code is executed in browser, not in Python IDE

Section 3 Setting Up Your Data Tables

Mysql Statements & Python functions you will need to know:

Connect to a Database & Get a Table Cursor

```
conn = pymysql. connect(host='localhost', user='root', passwd='???',
database='DBNAME', autocommit=True)

cur = conn.cursor()
```

Section 3 Setting Up Your Data Tables

To autocommit or not to autocommit

Autocommit forces a database write operation to happen at the execute command and makes code simple.

But if there is a chance of an error occurring, you don't want to use autocommit but rather do an explicit commit or rollback:

```
try:
   cursor.execute(...)
except DatabaseError:
   cursor.rollback()
   raise    # throws the exception code up the stack for further processing – allows for graceful error recovery
else:
   cursor.commit()
finally:
   cursor.close()
```

Section 3 Setting Up Your Data Tables

Mysql Statements & Python functions you will need to know:

Create a table:

Syntax: (Full definition: http://dev.mysql.com/doc/refman/5.1/en/create-table.html)
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name (create_definition,...) [table_options] [partition_options]

Code example:

```
cur.execute('''CREATE TABLE IF NOT EXISTS sales (SalesID int(11) NOT NULL AUTO_INCREMENT,
    Quantity int(11),
    ProductName varchar(50),
    InvID int(11),
    CustID int(11),
    PRIMARY KEY (SalesID))''')
```

Section 3 Setting Up Your Data Tables

Mysql Statements & Python functions you will need to know:

Close database functions

```
cur.commit()  # commits were handled with setting autocommit=True on DB open.
cur.close()
conn.commit()
conn.close()
```

Section 3 Setting Up Your Data Tables

Mysql Statements & Python functions you will need to know:

cgi module functions you need to know.

```
formdata = cgi.FieldStorage()
xyz = formdata.getfirst('variablename', '')  #2nd parameter is the default value if no match is found
```

How to do HTML code in python...

The print statement:

```
print('hello world')
```

Section 3 Setting Up Your Data Tables

Getting details on code errors

Python IDE gives traceback errors, etc. in the development environmnet

When developing under a web based environment, those errors are invisible.

They are located at:

C:\xampp\apache\logs\error.log

Section 3 Setting Up Your Data Tables

Mysql Statements & Python functions you will need to know:

Additional note with respect to HTML: You must generate properly formatted pages to get them to process correctly. So you need a header and footer.

Header:

```
print('Content-type: text/html\n')
print('<html>')
print('<head><title>', headertitle, '</title></head>')
print('<body>')
```

Footer:

```
print('</body>')
print('</html>')
```

Section 3 Setting Up Your Data Tables

Take some time to write the first iteration of the inventory management system – basic setup of the modules, creation of the database tables and a main menu.

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.