YIJC/2022/II/Q4

A company StoreSG supplies a wide range of products to the schools in Singapore. The company uses a database to maintain the records of all the orders made by the schools.

The table descriptions for the database are as follows:

- `School (SchoolCode, SchoolName, SchoolAddress)`
- `Product (ProductID, ProductName, UnitCost)`
- `Buy (BuyID, SchoolCode, ProductID, Qty, Status)`

The primary and foreign keys are indicated with underline and dashed underline respectively.

The values for the `BuyID` field in the `Buy` table are auto generated integers.

The field `Status` in the `Buy` table is used to track the status of the delivery and it could be be `'Pending'`, `'Enroute'` or `'Complete'`.

The database `StoreSG.db` provided contains two of the tables `School` and `Product`, which are both populated with data.

Task 4.1

Write Python program code to:

- create a table `Buy` in the database `StoreSG.db`, with the field `Qty` defined as an integer. The foreign keys must also be referenced to the primary keys in their respective tables.
- insert all the information from the data file `Buy.TXT` into the table `Buy`.

Save your program code as `Task4_1_<your name>.py`. [5]

Task 4.2

Hilltop University, school code `7612`, has previously ordered some items from the company.

Write SQL codes to:

a) show the name, quantity and unit cost of all the products ordered;
b) compute the total cost of all the items ordered.

Save the two SQL codes in a text file and name it as

`Task4_2_<your name>.sql` [5]

The company intends to develop a web application to allow the schools to view their products and submit orders online.

Task 4.3

Write a Python program and the necessary files to create a web application that enables the list of items found in the `Product` table to be displayed by `index.html` in a web browser.

For each item, the web page should include the product's:

- ID,
- name, and
- unit cost.

Save your program as

`Task4_3_<your name>_server.py` and
`Task4_3_<your name>_index.html`

Run the web application and save the output of the program as

`Task4_3_<your name>_output.html` [5]

[For Task 4.4 and 4.5, you are not required to include any code for input validation.]

Task 4.4

Modify the code in the `index.html` created in Task 4.3 to display a form for a user to submit an order online. The form should allow the user to order an item from the product list displayed, indicate the product's ID and the quantity to order, and submit with the school code.

Save your program code as

`Task4_4_<your name>_index.html`

Run the web application and save the output of the program as

`Task4_4_<your name>_output.html` [4]

Task 4.5

Modify the Python program code written in Task 4.3 to:

- read the information in the order form submitted by the user.
- insert the new record into the `Buy` table and set the `Status` as `'Pending'`.
- display a confirmation page on the web browser showing the product name, the quantity ordered and the total cost for the order.

Test your web application with the following form inputs:

- School Code: `1292`
- Product ID: `4`
- Order Quantity: `5`

Save your program as

`Task4_5_<your name>_server.py`

Run the web application and save the output of the program as

`Task4_5_<your name>_output.html`

with any additional files / sub-folders as needed in a folder named

`Task4_5_<your name>` [6]