A. from the python module inventory\_management/products.py, import product\_list

B. Write a Python program that will connect to the Inventory management database using the connection parameters below:

# Connect to the SQL Server

connection\_string = 'DRIVER={SQL Server};SERVER=DESKTOP-SH1HJJ8\\SQLEXPRESS;DATABASE=InventoryManagement;Trusted\_Connection=yes;'

conn = pyodbc.connect(connection\_string)

cursor = conn.cursor()

1. Display the table Products.   
1b. Create an empty dictionary with integer keys and string values named product\_list.

1c. Create a function update\_product\_list that will append the ProductID and ProductName in the table Products as keys and values of the dictionary product\_list.

2. Then ask user whether they want to add a new ProductName to the table Products OR they want to change the name of an old ProductName OR terminate the program.

3a. if user decides they want to add a new ProductName:

3b. Prompt user to enter the new ProductName.

3c. Append the new ProductName to the dictionary product\_list in the python module inventory\_management/products.py.

3d. assign a new ProductID which is equals to the key of the new ProductName in the dictionary product\_list in the python module inventory\_management/products.py.

3e. Prompt user to enter the StockQuantity(greater than 0) of the new ProductName.

3ei. Check if StockQuantity is an integer AND greater than 0.

3eii. If 3ei. is True, store StockQuantity input in the variable StockQuantity. Then, continue from 3f.

3eiii. While 3ei. is False, tell User that the StockQuantity input is unacceptable, then repeat 3e to 3eiii.

3f. Using the SQL UPDATE statement, update the table Products with the new ProductID, new ProductName, and the corresponding StockQuantity.

4a. Else if user decides they want to change the name of an old ProductName.

4b. Prompt user to enter the name of the old ProductName they want to change.

4c. Check if the old ProductName input is in the dictionary product\_list.

4di. If it is not. Tell user that the old ProductName they want to change is not an existing ProductName in the table Products.

4dii. Prompt user to enter the correct name of the old ProductName they want to change.

4ei. Else if the old ProductName input is in the dictionary product\_list. Prompt user to enter the new ProductName they want to change to.

4eii. Substitute the old ProductName in the product\_list in the python module inventory\_management/products.py. with the new ProductName specified in 4ei.

4f. Repeat 3e. to 3eiii.

4g. Using the SQL UPDATE statement: "UPDATE table\_name SET column\_name = new\_value WHERE condition;" update the table Products with the new ProductID, new ProductName provided in 4b. OR 4dii. and the corresponding StockQuantity entered in 4f.

5a. Else if user decides they want to terminate the program.

5b. Display the updated table Products.

5c. Terminate the program.