

Revised Final Project

● Function Search

รูปแบบเดิม ฟังก์ชันในการค้นหาข้อมูลสินค้า function search_product จะสามารถทำการค้นหาสินค้า จากตาราง product ในฐานข้อมูล โดยชื่อสินค้าและราคา ซึ่งมีเงื่อนไขและรูปแบบในการค้นหาสินค้า โดยชื่อสินค้าแบบ non-sensitive case และ การค้นหาด้วยราคาเป็น sensitive case

ตัวอย่างการค้นหาสินค้า ด้วยชื่อสินค้า non-sensitive case

```
=====
TheDrink TheDrank TheDrunk
=====
Main Menu
[1] Show Data
[2] Search Data
[3] Insert Data
[4] Update Data
[5] Delete Data
[6] Exit
Choose Menu : 2

Menu
[1] Search Product
[2] Search Product Type
[3] Exit
Choose Menu : 1

----- Search Product -----
Search Product Name or Price : c
result>> 3
Found | Id: 1 | Name : Coke      | Price : 11.0 | Amount : 11 | Type : Tea
Found | Id: 3 | Name : Chang      | Price : 55.0 | Amount : 24 | Type : Beer
Found | Id: 5 | Name : Nescafe     | Price : 10.0 | Amount : 3  | Type : Coffee

-----

Menu
[1] Search Product
[2] Search Product Type
[3] Exit
Choose Menu : 2
```

ตัวอย่างการค้นหาสินค้า ด้วยราคาสินค้า sensitive case

```
----- Search Product -----
Search Product Name or Price : 20
result>> 0
>> 20 is Not Found !!

-----

Menu
[1] Search Product
[2] Search Product Type
[3] Exit
Choose Menu : 1

----- Search Product -----
Search Product Name or Price : 25
result>> 1
Found | Id: 6 | Name : Osishi   | Price : 25.0 | Amount : 12 | Type : Tea

-----
```

โดยมีชุดคำสั่งต่อไปนี้

```
def search_product(): #สร้างฟังก์ชันชื่อ search_product

    db, conn = connect.connect_db() #เข้าถึงไฟล์ฐานข้อมูลและตารางข้อมูล

    input_search = input("Search Product Name or Price: ") #ประกาศตัวแปรที่รับค่าเข้ามา

    sql = "select * from product,producttype #อ่านค่าข้อมูลทั้งหมดจากตาราง product,producttype

        where product.typeID = producttype.typeID #เงื่อนไขให้ค่ารหัสสินค้าทั้ง2ตารางเท่ากัน

        and(product.productName LIKE %s #เปรียบเทียบตัวแปรที่รับค่าว่ามีในชื่อสินค้า

        OR product.productPrice LIKE %s) " #เปรียบเทียบตัวแปรที่รับค่าว่ามีในราคาสินค้า

    search=['%'+input_search+'%'] %% มีค่าเท่ากับตัวอักษรอะไรก็ได้และจำนวนกี่ตัวก็ได้ หลักwildcard

    keyword = (search,input_search) #จัดเก็บข้อมูลที่ต้องการแทนค่าตามลำดับ %s ลงตัวแปรkeyword

    conn.execute(sql,keyword) #ทำตามคำสั่งsql และอ้างอิงค่าข้อมูลตามตัวแปร keyword

    data_all = conn.fetchall() #อ่านค่าทั้งหมดของข้อมูลตามผลลัพธ์sql

    print("result>>",conn.rowcount) # rowcount นับจำนวนrecordตามผลลัพธ์sql

    for data in data_all: #ลูป for ตัวแปร data อ่านค่าข้อมูลของตัวแปร data_all

        print("Found | Id:", data[0], #print() ค่าของข้อมูลในdata ลำดับที่ 0

            "| Name :", data[1], #print() ค่าของข้อมูลในdata ลำดับที่ 1

            "\t| Price :", data[2], #print() ค่าของข้อมูลในdata ลำดับที่ 2

            "| Amount :", data[3], #print() ค่าของข้อมูลในdata ลำดับที่ 3

            "\t| Type :", data[6]) #print() ค่าของข้อมูลในdata ลำดับที่ 4

    if data_all == (): #เงื่อนไขถ้า data_all ไม่มีค่าของข้อมูลตามผลลัพธ์sql

        print(">>",input_search,"is Not Found !!") #ให้แสดงผลว่าไม่พบข้อมูล

    print("\n") #ขึ้นบรรทัดใหม่

    db.close() #ปิดไฟล์ฐานข้อมูล
```

โจทย์ ฟังก์ชันในการค้นหาข้อมูลจะสามารถทำการค้นหาสินค้าจากช่วงราคา ตามค่าที่กำหนด

รูปแบบใหม่ แยกฟังก์ชันสำหรับการค้นหาสินค้า โดย function search_product จะทำการค้นหาสินค้า จากตาราง product ที่ถูกจัดเก็บไว้ในฐานข้อมูล ด้วยชื่อของสินค้า และยังเป็นแบบ non-sensitive case เพิ่ม function search_price สำหรับ การค้นหาสินค้า จากตาราง product ที่ถูกจัดเก็บไว้ในฐานข้อมูล ตามช่วงราคาที่กำหนด

ตัวอย่างการค้นหาสินค้าตามช่วงราคาที่กำหนด

```
=====
TheDrink TheDrank TheDrunk
=====
Main Menu
[1] Show Data
[2] Search Data
[3] Insert Data
[4] Update Data
[5] Delete Data
[6] Exit
Choose Menu : 2

Menu
[1] Search Product
[2] Search Product Type
[3] Search Product Price
[4] Exit
Choose Menu : 3

----- Search Product Price -----
Input price (Start) : 10
Input price (Stop) : 100
result>> 7
Found | Price : 10.0 | Id: 5 | Name : Nescafe | Amount : 3 | Type : Coffee
Found | Price : 11.0 | Id: 1 | Name : Coke | Amount : 11 | Type : Tea
Found | Price : 12.0 | Id: 14 | Name : Nok | Amount : 8 | Type : Bear
Found | Price : 25.0 | Id: 6 | Name : Osishi | Amount : 12 | Type : Tea
Found | Price : 55.0 | Id: 3 | Name : Chang | Amount : 24 | Type : Beer
Found | Price : 58.0 | Id: 4 | Name : Leo | Amount : 16 | Type : Beer
Found | Price : 85.0 | Id: 7 | Name : Malee | Amount : 2 | Type : Juice
```

ตัวอย่างการแสดงผลเมื่อใส่ข้อมูลไม่ถูกต้อง

```
----- Search Product Price -----
Input price (Start) : -1
Input price (Stop) : -5
input price not correct !
-----

Menu
[1] Search Product
[2] Search Product Type
[3] Search Product Price
[4] Exit
Choose Menu : █
```

โดยมีชุดคำสั่งต่อไปนี้

```
def search_price(): #สร้างฟังก์ชันชื่อ search_product
    db, conn = connect.connect_db() #เข้าถึงไฟล์ฐานข้อมูลและตารางข้อมูล
    input_price1 = input("Input price (Start) : ") #ประกาศตัวแปรเพื่อรับค่าเข้ามา
    input_price2 = input("Input price (Stop) : ") #ประกาศตัวแปรเพื่อรับค่าเข้ามา
    if (input_price1 <= '0') or (input_price2 <= '0'): #ถ้าค่าข้อมูลว่ามีค่า น้อยกว่าหรือเท่ากับ 0
        print("input price not correct !") #ให้แสดงข้อความ
    else: #ถ้าค่าของข้อมูลที่ได้รับเข้ามาไม่เข้าเงื่อนไขให้ทำคำสั่งลำดับถัดไป
        sql = "select * from product,producttype #อ่านค่าข้อมูลทั้งหมดจากตาราง product,producttype
            where (product.productPrice Between%sAND%s) #เงื่อนไขให้ราคาสินค้ามีค่าระหว่าง ตัวแปรที่กำหนด
            AND (product.typeID = producttype.typeID) #เงื่อนไขให้รหัสสินค้าทั้ง2ตารางเท่ากัน
            order by product.productPrice asc " #เรียงตามลำดับ ราคาจากน้อยไปมาก
        input_price =(input_price1,input_price2) #จัดเก็บข้อมูลที่ต้องการแทนค่าตามลำดับ %s
        conn.execute(sql,input_price) #ทำตามคำสั่งsql และอ้างอิงค่าข้อมูลตามตัวแปร keyword
        data_all = conn.fetchall() #อ่านค่าทั้งหมดของข้อมูลตามผลลัพธ์sql
        print("result>>",conn.rowcount) # rowcount นับจำนวนrecordตามผลลัพธ์sql
        for data in data_all: #loop for ตัวแปร data อ่านค่าข้อมูลของตัวแปร data_all
            print("Found | Price :", data[2], #print() ค่าของข้อมูลในdata ลำดับที่ 2
                "| Id:", data[0], " #print() ค่าของข้อมูลในdata ลำดับที่ 0
                | Name :", data[1], #print() ค่าของข้อมูลในdata ลำดับที่ 1
                "\t| Amount :", data[3], #print() ค่าของข้อมูลในdata ลำดับที่ 3
                "\t| Type :", data[6]) #print() ค่าของข้อมูลในdata ลำดับที่ 4
        if data_all == ():
            print(">>",input_search,"is Not Found !!")
            print("\n")
        db.close()
```

- Function Delete

รูปแบบเดิม ฟังก์ชันในการลบข้อมูลประเภทสินค้า function delete_type
จะสามารถทำการลบประเภทสินค้า จากตาราง producttype ในฐานข้อมูล โดยรหัสประเภทสินค้า

```
def delete_type(): #สร้างฟังก์ชันชื่อ delete_type

    db, conn = connect.connect_db() #เข้าถึงไฟล์ฐานข้อมูลและตารางข้อมูล
    select.all_type() #เรียกใช้ไฟล์ select ฟังก์ชัน all_type เพื่อแสดงรายชื่อประเภทสินค้า
    input_typeID = int(input("input typeID: ")) #ประกาศตัวแปรเพื่อรับค่าเข้ามา
    sql2 = "SELECT * FROM producttype WHERE typeID =%s "
    #อ่านค่าข้อมูลทั้งหมดจากตารางproducttype โดยให้แสดงข้อมูลที่มีค่าไอดีเหมือนกัน
    chk = (input_typeID) ) #จัดเก็บข้อมูลที่ต้องการแทนค่าตามลำดับ %s
    conn.execute(sql2,chk) #ทำตามคำสั่งsql และอ้างอิงค่าข้อมูลตามตัวแปร chk
    data_all = conn.fetchall() #อ่านค่าทั้งหมดของข้อมูลตามผลลัพธ์sql
    if data_all ==(): #เงื่อนไขถ้าไอดีที่รับเข้ามาไม่มีในตารางฐานข้อมูล มีค่าว่าง
        print(">>Type ID not correct ! ") #แสดงข้อความ>>Type ID not correct !
        print("Please Try again later") # แสดงข้อความ Please Try again later
    elif data_all != (): #เงื่อนไข ถ้าไอดีที่รับเข้ามามีในตารางฐานข้อมูล ไม่มีค่าว่าง
        cf = input('Comfirm Delete Data ? y/n: ').lower() #ประกาศตัวแปรเพื่อรับค่าเข้ามา
        if(cf == 'y'): #ถ้าค่ารับเข้ามาเท่ากับ y
            sql = "Delete From producttype where typeID = %s" #ลบไอดีจากตารางฐานข้อมูล producttype
            delete_datas = (input_typeID) #จัดเก็บข้อมูลที่ต้องการแทนค่าตามลำดับ %s
            conn.execute(sql, delete_datas) #ทำตามคำสั่งsql และอ้างอิงค่าข้อมูลตามตัวแปร delete_datas
            db.commit() #ยืนยันการเปลี่ยนแปลงข้อมูล
            print(">>>Delete successful<<<") # แสดงข้อความ ">>>Delete successful<<<
            select.all_type() #เรียกใช้ไฟล์ select ฟังก์ชัน all_type เพื่อแสดงรายชื่อประเภทสินค้า
        else: #ถ้าไม่เข้าเงื่อนไข
            print(">>>See you later") # แสดงข้อความ >>>See you later
    db.close()
```

โจทย์ ฟังก์ชันในการลบข้อมูลประเภทสินค้า จะไม่สามารถลบประเภทสินค้าที่มีข้อมูลสินค้าประเภทนั้นๆอยู่

รูปแบบใหม่ ฟังก์ชันในการลบข้อมูลประเภทสินค้า function delete_type จะสามารถทำการลบข้อมูลประเภทสินค้า โดยมีเงื่อนไข หากมีสินค้าประเภทสินค้านี้อยู่จะไม่สามารถทำการลบข้อมูลประเภทสินค้าได้ ให้สามารถลบประเภทสินค้า เฉพาะประเภทสินค้าที่ไม่มีสินค้าชนิดนั้นอยู่ ดังนี้

ตัวอย่างการแสดงผลเมื่อใส่รหัสประเภทสินค้าที่มีสินค้าอยู่

```
----- delete Product Type -----
----- All Type -----
ID : 1 Name : Green tea
ID : 2 Name : Coffee
ID : 3 Name : Alcohol
ID : 4 Name : Beer
ID : 5 Name : Juice
ID : 6 Name : Soda
ID : 7 Name : Milk
ID : 8 Name : Soft drink

input typeID: 6
>>Failed to delete . This Type have product.
Type : 6 Soda Id: 1 Name : Coke          Amount : 10
Type : 6 Soda Id: 2 Name : Pepsi         Amount : 14
```

ตัวอย่างการแสดงผลเมื่อใส่รหัสประเภทสินค้าที่ไม่มีในฐานข้อมูล

```
----- delete Product Type -----
----- All Type -----
ID : 1 Name : Green tea
ID : 2 Name : Coffee
ID : 3 Name : Alcohol
ID : 4 Name : Beer
ID : 5 Name : Juice
ID : 6 Name : Soda
ID : 7 Name : Milk
ID : 8 Name : Soft drink

input typeID: 10
>>Type ID not correct !
Please Try again later
```

ตัวอย่างการแสดงผลเมื่อยกเลิกการลบข้อมูล

```
----- delete Product Type -----
----- All Type -----
ID : 1 Name : Green tea
ID : 2 Name : Coffee
ID : 3 Name : Alcohol
ID : 4 Name : Beer
ID : 5 Name : Juice
ID : 6 Name : Soda
ID : 7 Name : Milk
ID : 8 Name : Soft drink

input typeID: 7
Type : 7 | Name : Milk
Comfirm Delete Data ? y/n: n
>>>See you later
```

ตัวอย่างการแสดงผลเมื่อยืนยันการลบข้อมูล

```
=====
TheDrink TheDrank TheDrunk
=====
Main Menu
[1] Show Data
[2] Search Data
[3] Insert Data
[4] Update Data
[5] Delete Data
[6] Exit
Choose Menu : 5

Menu
[1] delete Product
[2] delete Product Type
[3] Exit
Choose Menu : 2

----- delete Product Type -----
----- All Type -----
ID : 1 Name : Green tea
ID : 2 Name : Coffee
ID : 3 Name : Alcohol
ID : 4 Name : Beer
ID : 5 Name : Juice
ID : 6 Name : Soda
ID : 7 Name : Milk
-----
input typeID: 7
Type : 7 | Name : Milk
Comfirm Delete Data ? y/n: y
>>>Delete successful<<<<
----- All Type -----
ID : 1 Name : Green tea
ID : 2 Name : Coffee
ID : 3 Name : Alcohol
ID : 4 Name : Beer
ID : 5 Name : Juice
ID : 6 Name : Soda
-----

Menu
[1] delete Product
[2] delete Product Type
[3] Exit
Choose Menu : █
```

โดยมีชุดคำสั่งต่อไปนี้

```
def delete_type(): #สร้างฟังก์ชันชื่อ delete_type
    db, conn = connect.connect_db() #เข้าถึงไฟล์ฐานข้อมูลและตารางข้อมูล
    select.all_type() #เรียกใช้ไฟล์ select ฟังก์ชัน all_type เพื่อแสดงรายชื่อประเภทสินค้า
    input_typeID = int(input("input typeID: ")) #ประกาศตัวแปรเพื่อรับค่าเข้ามา
    sql3 = "SELECT * FROM producttype WHERE typeID =%s "
    #อ่านค่าข้อมูลทั้งหมดจากตารางproducttype ที่มีค่าไอดีเท่ากับตัวแปรที่รับค่ามา
    chk = (input_typeID) #จัดเก็บข้อมูลที่ต้องการแทนค่าตามลำดับ %s
    conn.execute(sql3,chk) #ทำตามคำสั่งsql และอ้างอิงค่าข้อมูลตามตัวแปร chk
    data_all = conn.fetchall() #อ่านค่าทั้งหมดของข้อมูลตามผลลัพธ์sql
```

```

if data_all ==(): #เงื่อนไขถ้าไอดีที่รับเข้ามาไม่มีในตารางฐานข้อมูล มีค่าว่าง
    print(">>Type ID not correct ! ") #แสดงข้อความ>>Type ID not correct !
    print("Please Try again later") # แสดงข้อความ Please Try again later
elif data_all !=(): #เงื่อนไข ถ้าไอดีที่รับเข้ามามีในตารางฐานข้อมูล ไม่มีค่าว่าง
    sql2 = "select * from producttype,product #อ่านค่าข้อมูลทั้งหมดจากตาราง producttype,product
            where (product.typeID Like %s) #เงื่อนไขtypeIDตารางproductเท่ากับตัวแปรที่รับค่ามา
            AND producttype.typeID = product.typeID"
            #เงื่อนไขtypeIDตารางproductเท่ากับtypeIDตารางproducttype
    chk2 = (input_typeID) #จัดเก็บข้อมูลที่ต้องการแทนค่าตามลำดับ %s
    conn.execute(sql2,chk2) #ทำตามคำสั่งsql และอ้างอิงค่าข้อมูลตามตัวแปร chk2
    data_all2 = conn.fetchall() #อ่านค่าทั้งหมดของข้อมูลตามผลลัพธ์sql
    i = []
    for data in data_all2 : #ลูป for ตัวแปร data อ่านค่าข้อมูลของตัวแปร data_all2
        if i != data[6]: #ถ้าที่วนอ่านค่าในตำแหน่งที่6 อ่านเจอค่าข้อมูลซึ่งค่าของประเภทสินค้าจากตารางproduct
            print(">>Failed to delete . This Type have product.")
            i = data[6] #กำหนดให้ i มีค่าเท่ากับ ข้อมูลในตำแหน่งที่6
        print("Type :",data[0],data[1], #print() ค่าของข้อมูลไอดีและชื่อจากproducttype
              "Id:", data[2], "Name :", data[3], #print() ค่าของข้อมูลไอดี,ชื่อและจำนวนจากproduct
              "\tAmount :", data[5])
    if data_all2 ==():#ถ้าตารางproduct ไม่มี typeid ตรงกับที่input
        for data in data_all: #ลูป for ตัวแปร data อ่านค่าข้อมูลของตัวแปร data_all
            print("Type :",data[0],"| Name :",data[1]) #print() ค่าของข้อมูลไอดีและชื่อจากproducttype
            cf = input('Confirm Delete Data ? y/n: ').lower() #ประกาศตัวแปรเพื่อรับค่าเข้ามา
            if(cf == 'y'): #ถ้าค่ารับเข้ามาเท่ากับ y
                sql = "Delete From producttype where typeID = %s" #ลบไอดีจากตารางฐานข้อมูล producttype
                delete_datas = (input_typeID) #จัดเก็บข้อมูลที่ต้องการแทนค่าตามลำดับ %s
                conn.execute(sql, delete_datas) #ทำตามคำสั่งsql และอ้างอิงค่าข้อมูลตามตัวแปร delete_datas
                db.commit() #ยืนยันการเปลี่ยนแปลงข้อมูล
                print(">>>Delete successful<<<") # แสดงข้อความ ">>>Delete successful<<<
                select.all_type() #เรียกใช้ไฟล์ select ฟังก์ชัน all_type เพื่อแสดงรายชื่อประเภทสินค้า
            else: #ถ้าไม่เข้าเงื่อนไข
                print(">>>See you later") # แสดงข้อความ >>>See you later
        db.close()

```


- Function Update

รูปแบบเดิม ฟังก์ชันในการอัปเดตข้อมูลสินค้า function update_product จะสามารถอัปเดตข้อมูลจากตาราง product ในฐานข้อมูล โดยการป้อนค่าข้อมูล รหัสสินค้า ชื่อสินค้า ราคาสินค้า จำนวนของสินค้า และ ประเภทสินค้า เพื่อเปลี่ยนแปลงและแก้ไขข้อมูลตามที่กำหนด

```

----- update Product -----
----- All Product -----
Id: 1 | Name : Coke | Price : 25.0 | Amount : 10 | Type : Soda
Id: 2 | Name : Pepsi | Price : 32.0 | Amount : 14 | Type : Soda
Id: 3 | Name : Chang | Price : 55.0 | Amount : 24 | Type : Beer
Id: 4 | Name : Leo | Price : 58.0 | Amount : 16 | Type : Beer
Id: 5 | Name : Nescafe | Price : 10.0 | Amount : 3 | Type : Coffee
Id: 6 | Name : Osishi | Price : 25.0 | Amount : 12 | Type : Green tea
Id: 7 | Name : Malee | Price : 85.0 | Amount : 2 | Type : Juice
-----

please input Name : coke
please input Price : 25
please input Amount : 12
----- All Type -----
ID : 1 Name : Green tea
ID : 2 Name : Coffee
ID : 3 Name : Alcohol
ID : 4 Name : Beer
ID : 5 Name : Juice
ID : 6 Name : Soda
-----

please input typeID : 5
input productID: 1
Comfirm Update Data ? y/n: y
>>>Update successful<<<<
----- All Product -----
Id: 1 | Name : Coke | Price : 25.0 | Amount : 12 | Type : Juice
Id: 2 | Name : Pepsi | Price : 32.0 | Amount : 14 | Type : Soda
Id: 3 | Name : Chang | Price : 55.0 | Amount : 24 | Type : Beer
Id: 4 | Name : Leo | Price : 58.0 | Amount : 16 | Type : Beer
Id: 5 | Name : Nescafe | Price : 10.0 | Amount : 3 | Type : Coffee
Id: 6 | Name : Osishi | Price : 25.0 | Amount : 12 | Type : Green tea
Id: 7 | Name : Malee | Price : 85.0 | Amount : 2 | Type : Juice
-----

```

โดยมีชุดคำสั่งต่อไปนี้

```
def update_product(): #สร้างฟังก์ชันชื่อ update_product
    db, conn = connect.connect_db()
    select.show_product()
    input_productName = input("please input Name : ").capitalize()
    input_productPrice = input("please input Price : ")
    input_productAmount = input("please input Amount : ")
    select.all_type()
    input_typeID = int(input("please input typeID : "))
    input_productID = int(input("input productID: "))
    sql2 = "SELECT * FROM product,producttype #อ่านค่าข้อมูลทั้งหมดจากตารางproducttype,product
           WHERE product.productID =%s AND producttype.typeID =%s "
           #ตัวแปรที่รับค่าว่ามีในรหัสสินค้าตารางproductและรหัสประเภทสินค้าจากตารางproducttypeหรือไม่
    chk = (input_productID,input_typeID) #จัดเก็บข้อมูลที่ต้องการแทนค่าตามลำดับ %s
    conn.execute(sql2,chk) #ทำตามคำสั่งsql และอ้างอิงค่าข้อมูลตามตัวแปร chk
    data_all = conn.fetchall() #อ่านค่าทั้งหมดของข้อมูลตามผลลัพธ์sql
    if data_all ==():#เงื่อนไขถ้าไอดีที่รับเข้ามาไม่มีในตารางฐานข้อมูล มีค่าว่าง
        print(">>Prouduct ID or Type ID not correct ! ")
        print("Please Try again later")
    elif data_all != ():
        cf = input('Comfirm Update Data ? y/n: ').lower()
        if(cf == 'y'):
            sql = "Update product Set productName=%s,productPrice=%s, #อัปเดตข้อมูล product
                   productAmount=%s,typeID=%s where productID = %s "
            update_datas = (input_productName, input_productPrice,
                            input_productAmount,input_typeID,input_productID)
            conn.execute(sql, update_datas) #ทำตามคำสั่งsql และอ้างอิงค่าข้อมูลตามตัวแปร update_datas
            db.commit() #ยืนยันการเปลี่ยนแปลงข้อมูล
            print(">>>Update successful<<<<")
            select.show_product()
        else:
            print(">>>See you later")
    db.close()
```

โจทย์ ฟังก์ชันในการอัปเดตข้อมูลสินค้า ให้อัปเดตเฉพาะ type

รูปแบบใหม่ ฟังก์ชันในการอัปเดตข้อมูลสินค้า function update_product จะสามารถอัปเดตข้อมูลจากตาราง product สามารถเลือกได้ว่าจะทำการอัปเดตแก้ไขข้อมูลใด **ไม่ต้องป้อนข้อมูลสินค้าซ้ำ** เลือกการอัปเดตได้อย่างอิสระมากขึ้น ไม่ว่าจะเป็น รหัสสินค้า ชื่อสินค้า ราคาสินค้า จำนวนของสินค้า หรือประเภทสินค้า ดังตัวอย่างต่อไปนี้

ตัวอย่างการแสดงผลการอัปเดตข้อมูลสินค้า แก้ไขเฉพาะประเภทสินค้า

```
Menu
[1] update Product
[2] update Product Type
[3] update
[4] Exit
Choose Menu : 3
```

update				
All Product				
Id: 1	Name : Coke	Price : 25.0	Amount : 10	Type : Soda
Id: 2	Name : Pepsi	Price : 32.0	Amount : 14	Type : Soda
Id: 3	Name : Chang	Price : 55.0	Amount : 24	Type : Beer
Id: 4	Name : Leo	Price : 58.0	Amount : 16	Type : Beer
Id: 5	Name : Nescafe	Price : 10.0	Amount : 3	Type : Coffee
Id: 6	Name : Osishi	Price : 25.0	Amount : 12	Type : Green tea
Id: 7	Name : Malee	Price : 85.0	Amount : 2	Type : Juice

```
input productID: 7
Do you want to Update Name ? y/n: n
Do you want to Update Price ? y/n: n
Do you want to Update Amount ? y/n: n
```

All Type	
ID : 1	Name : Green tea
ID : 2	Name : Coffee
ID : 3	Name : Alcohol
ID : 4	Name : Beer
ID : 5	Name : Juice
ID : 6	Name : Soda
ID : 7	Name : Milk
ID : 8	Name : Soft drink

```
please input typeID : 8
Id: 7 | Name : Malee | Price : 85.0 | Amount : 2 | Type : 8
Comfirm Update Data ? y/n: y
>>>Update successful<<<<
```

All Product				
Id: 1	Name : Coke	Price : 25.0	Amount : 10	Type : Soda
Id: 2	Name : Pepsi	Price : 32.0	Amount : 14	Type : Soda
Id: 3	Name : Chang	Price : 55.0	Amount : 24	Type : Beer
Id: 4	Name : Leo	Price : 58.0	Amount : 16	Type : Beer
Id: 5	Name : Nescafe	Price : 10.0	Amount : 3	Type : Coffee
Id: 6	Name : Osishi	Price : 25.0	Amount : 12	Type : Green tea
Id: 7	Name : Malee	Price : 85.0	Amount : 2	Type : Soft drink

ตัวอย่างการแสดงผลการอัปเดตข้อมูลสินค้า

Menu

- [1] update Product
- [2] update Product Type
- [3] update
- [4] Exit

Choose Menu : 3

----- update -----				
----- All Product -----				
Id: 1	Name : Kkkk	Price : 10.0	Amount : 10	Type : Green tea
Id: 2	Name : Pepsi	Price : 32.0	Amount : 14	Type : Soda
Id: 3	Name : Chang	Price : 55.0	Amount : 24	Type : Beer
Id: 4	Name : Leo	Price : 58.0	Amount : 16	Type : Beer
Id: 5	Name : Nescafe	Price : 10.0	Amount : 3	Type : Coffee
Id: 6	Name : Osishi	Price : 25.0	Amount : 12	Type : Green tea
Id: 7	Name : Malee	Price : 85.0	Amount : 2	Type : Juice

input productID: 1

Do you want to Update Name ? y/n: y

please input Name : coke

Do you want to Update Price ? y/n: y

please input Price : 25

Do you want to Update Amount ? y/n: n

----- All Type -----

ID : 1 Name : Green tea
 ID : 2 Name : Coffee
 ID : 3 Name : Alcohol
 ID : 4 Name : Beer
 ID : 5 Name : Juice
 ID : 6 Name : Soda
 ID : 7 Name : Milk
 ID : 8 Name : Soft drink

please input typeID : 6

Id: 1 | Name : Coke | Price : 25 | Amount : 10 | Type : 6

Comfirm Update Data ? y/n: y

>>>Update successful<<<<

----- All Product -----				
Id: 1	Name : Coke	Price : 25.0	Amount : 10	Type : Soda
Id: 2	Name : Pepsi	Price : 32.0	Amount : 14	Type : Soda
Id: 3	Name : Chang	Price : 55.0	Amount : 24	Type : Beer
Id: 4	Name : Leo	Price : 58.0	Amount : 16	Type : Beer
Id: 5	Name : Nescafe	Price : 10.0	Amount : 3	Type : Coffee
Id: 6	Name : Osishi	Price : 25.0	Amount : 12	Type : Green tea
Id: 7	Name : Malee	Price : 85.0	Amount : 2	Type : Juice

ตัวอย่างการแสดงผล เมื่อใส่รหัสประเภทสินค้าไม่ถูกต้อง

```
Menu
[1] update Product
[2] update Product Type
[3] update
[4] Exit
Choose Menu : 3

update
All Product
Id: 1 | Name : Coke | Price : 25.0 | Amount : 10 | Type : Soda
Id: 2 | Name : Pepsi | Price : 32.0 | Amount : 14 | Type : Soda
Id: 3 | Name : Chang | Price : 55.0 | Amount : 24 | Type : Beer
Id: 4 | Name : Leo | Price : 58.0 | Amount : 16 | Type : Beer
Id: 5 | Name : Nescafe | Price : 10.0 | Amount : 3 | Type : Coffee
Id: 6 | Name : Osishi | Price : 25.0 | Amount : 12 | Type : Green tea
Id: 7 | Name : Malee | Price : 85.0 | Amount : 2 | Type : Soft drink

input productID: 7
Do you want to Update Name ? y/n: n
Do you want to Update Price ? y/n: n
Do you want to Update Amount ? y/n: n
All Type
ID : 1 Name : Green tea
ID : 2 Name : Coffee
ID : 3 Name : Alcohol
ID : 4 Name : Beer
ID : 5 Name : Juice
ID : 6 Name : Soda
ID : 7 Name : Milk
ID : 8 Name : Soft drink

please input typeID : 10
>>Type ID not correct !
Please Try again later
```

ตัวอย่างการแสดงผล เมื่อใส่รหัสสินค้าไม่ถูกต้อง

```
update
All Product
Id: 1 | Name : Coke | Price : 25.0 | Amount : 10 | Type : Soda
Id: 2 | Name : Pepsi | Price : 32.0 | Amount : 14 | Type : Soda
Id: 3 | Name : Chang | Price : 55.0 | Amount : 24 | Type : Beer
Id: 4 | Name : Leo | Price : 58.0 | Amount : 16 | Type : Beer
Id: 5 | Name : Nescafe | Price : 10.0 | Amount : 3 | Type : Coffee
Id: 6 | Name : Osishi | Price : 25.0 | Amount : 12 | Type : Green tea
Id: 7 | Name : Malee | Price : 85.0 | Amount : 2 | Type : Soft drink

input productID: 10
>>productID not correct!
```

โดยมีชุดคำสั่งดังนี้

```
def update():
    db, conn = connect.connect_db()
    select.show_product() #เรียกใช้ไฟล์ select ฟังก์ชัน show_product เพื่อแสดงรายชื่อประเภทสินค้า
    productID = int(input("input productID: "))#ประกาศตัวแปรเพื่อรับค่าเข้ามา
    sql = "select * from product where productID = %s" #อ่านค่าข้อมูลทั้งหมดจากตารางproduct
    conn.execute(sql,productID)
    data_all = conn.fetchall()
    for data in data_all:
        if data[0] != productID: #เงื่อนไข if ไว้เช็คไอดีที่รับเข้ามาไม่มีในฐานข้อมูล
            print("productID not correct!")
        elif data[0] == productID: #เงื่อนไข if ไว้เช็คไอดีที่รับเข้ามาว่ามีในฐานข้อมูล
            cf1 = input('Do you want to Update Name ? y/n: ').lower()
            if(cf1 == 'y'):
                productName = input("please input Name : ").capitalize()
            elif(cf1 == 'n'):
                productName = data[1]
            cf2 = input('Do you want to Update Price ? y/n: ').lower()
            if(cf2 == 'y'):
                productPrice = input("please input Price : ")
            elif(cf2 == 'n'):
                productPrice = data[2]
            cf3 = input('Do you want to Update Amount ? y/n: ').lower()
            if(cf3 == 'y'):
                productAmount = input("please input Amount : ")
            elif(cf3 == 'n'):
                productAmount = data[3]
            select.all_type() #เรียกใช้ไฟล์ select ฟังก์ชัน all_type เพื่อแสดงรายชื่อประเภทสินค้า
            typeId = int(input("please input typeId : "))
            sql2 = "SELECT * FROM producttype WHERE producttype.typeID =%s "
            conn.execute(sql2,typeID)
            data_all2 = conn.fetchall()
            if data_all2 ==(): #เงื่อนไขถ้าผลลัพธ์ sql ไม่มีในตารางฐานข้อมูล มีค่าว่าง
                print(">>Type ID not correct ! ")
                print("Please Try again later")
            elif data_all2 != (): #เงื่อนไขถ้าผลลัพธ์ sql มีในตารางฐานข้อมูล ไม่มีค่าว่าง
                print("Id:",productID , "| Name :",productName , "| Price :", productPrice,
                    "| Amount :", productAmount,"| Type :", typeId)
                cf = input('Comfirm Update Data ? y/n: ').lower()
                if(cf == 'y'):
                    sql3 = "Update product Set #อัปเดตข้อมูล product
                        productName=%s,productPrice=%s,productAmount=%s,
                        typeId=%s where productID = %s "
                    update_datas = (productName, productPrice,
                        productAmount,typeID,productID)
                    conn.execute(sql3, update_datas) #ทำตามคำสั่งsql และอ้างอิงค่าข้อมูลตามตัวแปร update_datas
                    db.commit() #ยืนยันการเปลี่ยนแปลงข้อมูล
                    print(">>>Update successful<<<")
                    select.show_product()
                else:
                    print(">>>See you later")
    db.close()
```