department.py

```
from re import compile
import mysql.connector

# input validation
pos_int = compile(r'^[1-9]\d*$')

# db connection
group_number="10"
mydb = mysql.connector.connect(
    host="127.0.0.1",
    user="ht22_2_group_" + group_number,
    passwd="pwd_" + group_number,
    database="ht22_2_project_group_" + group_number
)
cur = mydb.cursor()

# prompt with viable id ranges
cur.execute("SELECT dept_id FROM department")
rs = [r[0] for r in cur.fetchall()]
prompt_id = f"Please choose a department ID from range ({min(rs)}, {max(rs)}) or 'exit': "


# ask for id until 'exit' is entered
while True:

    print()
    department_id = input(prompt_id)

    if department_id == 'exit':
        print('Goodbye')
        break

    if not pos_int.match(department_id):
        print('Invalid ID input')
        continue

    # check if entered id exists
    cur.execute("SELECT dept_id FROM department WHERE dept_id = " + department_id)
    department_id = cur.fetchone()

    if not department_id:
        print('Department ID not found')
        continue
    else:
        department_id = str(department_id[0])

    # fetch child departments
    cur.execute("SELECT dept_id FROM parent_department WHERE super_department_id = " + department_id)
    child_departments = tuple(r[0] for r in cur.fetchall())

    if child_departments:

        # fetch and print info on child departments
        cur.execute("SELECT dept_id, dept_title FROM department WHERE dept_id IN " + str(child_departments))
        child_departments = cur.fetchall()

        print('id          dept_title\n--------------------------')
        for cd in child_departments:
            print('{: >2d}{: >20s}'.format(*cd))

    else:

        # if no child departments exist, fetch and print info on products of leaf department
        cur.execute("SELECT product_id, prod_title, price_without_VAT * (1 + VAT_percentage / 100) * (1 - discount / 100) FROM products WHERE dept_id = " + department_id)
        products = cur.fetchall()

        print('id                                  prod_title     retail price\n-----------------------------------------------------------------')
        for p in products:
            print('{: >2d}{: >45s}{: >17s}'.format(p[0], p[1], str(round(p[2], 2))))


mydb.close()
```

product.py

```
from re import compile
import mysql.connector

# input validation
pos_int = compile(r'^[1-9]\d*$')
pos_double = compile(r'^\d+$|^\d+\.\d+$|^\.\d+$')

# db connection
group_number="10"
mydb = mysql.connector.connect(
    host="127.0.0.1",
    user="ht22_2_group_" + group_number,
    passwd="pwd_" + group_number,
    database="ht22_2_project_group_" + group_number
)
cur = mydb.cursor()

# prompt with viable id ranges
cur.execute("SELECT product_id FROM products")
rs = [r[0] for r in cur.fetchall()]
prompt_id = f"Please choose a product ID from range ({min(rs)}, {max(rs)}) or 'exit': "


# ask for id until 'exit' is entered
while True:

    print()
    product_id = input(prompt_id)

    if product_id == 'exit':
        print('Goodbye')
        break

    if not pos_int.match(product_id):
        print('Invalid ID input')
        continue

    # check if entered id exists
    cur.execute("SELECT discount FROM products WHERE product_id = " + product_id)
    product_discount = cur.fetchone()

    if not product_discount:
        print('Product ID not found')
        continue
    else:
        product_discount = product_discount[0] or '0'

    # display info on selected product
    print(f'Current discount on product {product_id} is {product_discount}')

    # check if user wants to change discount
    change = ''
    valid_response = False
    while not valid_response:
        change = input('Would you like to change the discount (y/n)? ').lower()
        if change not in ['y', 'n']:
            print('Invalid response - requires y or n')
        else:
            valid_response = True

    if change == 'n':
        continue

    # get and set new discount
    new_discount = ''
    valid_response = False
    while not valid_response:
        new_discount = input('What should the new discount be? ')
        if not pos_double.match(new_discount):
            print('Invalid reponse - requires positive double')
        else:
            valid_response = True

    cur.execute(f"UPDATE products SET discount = {new_discount} WHERE product_id = {product_id}")


mydb.close()
```

requirements.txt

```
mysql-connector-python==8.0.31
protobuf==3.20.1
```