

Завдання 1

Створіть тритабличну базу даних Sales (Продажі). У цій базі даних мають бути таблиці: Sales (інформація про конкретні продажі), Salesmen (інформація про продавців), Customers (інформація про покупців). Створіть додаток для відображення даних з таблиць. Меню додатку має містити такий мінімальний набір звітів:

- Відображення усіх угод;
- Відображення угод конкретного продавця;
- Відображення максимальної за сумою угоди;
- Відображення мінімальної за сумою угоди;
- Відображення максимальної суми угоди для конкретного продавця;
- Відображення мінімальної за сумою угоди для конкретного продавця;
- Відображення максимальної за сумою угоди для конкретного покупця;
- Відображення мінімальної за сумою угоди для конкретного покупця;
- Відображення продавця з максимальною сумою продажів за всіма угодами;
- Відображення продавця з мінімальною сумою продажів за всіма угодами;
- Відображення покупця з максимальною сумою покупок за всіма угодами;
- Відображення середньої суми покупки для конкретного покупця;
- Відображення середньої суми покупки для конкретного продавця.

```
import pymysql
import os.path
import json
try:
    connection = pymysql.connect(
        host="localhost",
        port=3306,
        user="root",
        password="IDSharapo_ff220601",
        database="Salesfor",
        cursorclass=pymysql.cursors.DictCursor )
    print("Okay")
    try:
        # Create DATABASE
        # with connection.cursor() as cursor:
        #     create_table = "CREATE DATABASE `Salesfor`"
        #     cursor.execute(create_table)

        #Create table
        # with connection.cursor() as cursor:
        #     create_table = "CREATE TABLE `Sales` (id int AUTO_INCREMENT, nameprod
        varchar(30), namepokup varchar(30),tovar varchar(30),money int, PRIMARY KEY (id));"
        #     cursor.execute(create_table)
        #     print("well done")

        #Insert data
```

```

        # with connection.cursor() as cursor:
        #     insert = "INSERT INTO `Sales` (nameprod,namepokup,tovar,money) VALUES
('Bober','Krean','loberna',1242), ('Ash','Cook','bober',231)"
        #     cursor.execute(insert)
        #     connection.commit()

# Create table
# with connection.cursor() as cursor:
#     create_table = "CREATE TABLE `Salesmen` (id int AUTO_INCREMENT, name
varchar(30), PRIMARY KEY (id));"
#     cursor.execute(create_table)
#     print("well done")

# Insert data
# with connection.cursor() as cursor:
#     insert = "INSERT INTO `Salesmen` (name) VALUES ('Bober'), ('Carl')"
#     cursor.execute(insert)
#     connection.commit()

# Create table
# with connection.cursor() as cursor:
#     create_table = "CREATE TABLE `Customers` (id int AUTO_INCREMENT, name
varchar(30), PRIMARY KEY (id));"
#     cursor.execute(create_table)
#     print("well done")

# Insert data
# with connection.cursor() as cursor:
#     insert = "INSERT INTO `Customers` (name) VALUES ('Robert'), ('Lyda')"
#     cursor.execute(insert)
#     connection.commit()

# Select data
while True:
    v=input(""""
1 Відображення усіх угод;
2 Відображення угод конкретного продавця;
3 Відображення максимальної за сумою угоди;
4 Відображення мінімальної за сумою угоди;
5 Відображення максимальної суми угоди для конкретного
продавця;
6 Відображення мінімальної за сумою угоди для конкретного продавця;
7 Відображення максимальної за сумою угоди для конкретного покупця;
8 Відображення мінімальної за сумою угоди для конкретного покупця;
9 Відображення продавця з максимальною сумою продажів
за всіма угодами
10 Відображення продавця з мінімальною сумою продажів
за всіма угодами;
11 Відображення покупця з максимальною сумою покупок
за всіма угодами;
12 Відображення середньої суми покупки для конкретного
покупця;

```

```

13 Вiдображення середньої суми покупки для конкретного
продавця.;""")
    if v=="1":
        with connection.cursor() as cursor:
            select_all = 'select * from salesfor.sales'
            cursor.execute(select_all)
            result = cursor.fetchall()
            #print(result)
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()

    if v=="2":
        with connection.cursor() as cursor:
            select_all = "select * from salesfor.sales where nameprod='Bober'"
            cursor.execute(select_all)
            result = cursor.fetchall()
            # print(result)
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()

    if v=="3":
        with connection.cursor() as cursor:
            select_all = "select *,max(money) from salesfor.sales "
            cursor.execute(select_all)
            result = cursor.fetchall()
            # print(result)
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()

    if v=="4":
        with connection.cursor() as cursor:
            select_all = "select *,min(money) from salesfor.sales "
            cursor.execute(select_all)
            result = cursor.fetchall()
            # print(result)
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()

    if v=="5":
        with connection.cursor() as cursor:
            select_all = "select *,max(money) from salesfor.sales where
nameprod='Ash'"
            cursor.execute(select_all)
            result = cursor.fetchall()
            # print(result)
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)

```

```

        file.close()
    if v=="6":
        with connection.cursor() as cursor:
            select_all = "select *,min(money) from salesfor.sales where
nameprod='Ash'"
            cursor.execute(select_all)
            result = cursor.fetchall()
            # print(result)
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()
    if v=="7":
        with connection.cursor() as cursor:
            select_all = "select *,max(money) from salesfor.sales where
namepukup='Cook'"
            cursor.execute(select_all)
            result = cursor.fetchall()
            # print(result)
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()
    if v=="8":
        with connection.cursor() as cursor:
            select_all = "select *,min(money) from salesfor.sales where
namepukup='Cook'"
            cursor.execute(select_all)
            result = cursor.fetchall()
            # print(result)
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()
    if v=="9":
        with connection.cursor() as cursor:
            select_all = "select nameprod from salesfor.sales where
money=(select max(money) from salesfor.sales)"
            cursor.execute(select_all)
            result = cursor.fetchall()
            # print(result)
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()
    if v=="10":
        with connection.cursor() as cursor:
            select_all = "select nameprod from salesfor.sales where
money=(select min(money) from salesfor.sales)"
            cursor.execute(select_all)
            result = cursor.fetchall()
            # print(result)
            for row in result:
                print(row)

```

```

        file = open(os.path.join("based", "camber.json"), "a")
        json.dump(row, file)
        file.close()
    if v == "11":
        with connection.cursor() as cursor:
            select_all = "select namepokup from salesfor.sales where
money=(select max(money) from salesfor.sales)"
            cursor.execute(select_all)
            result = cursor.fetchall()
            # print(result)
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()
    if v == "12":
        with connection.cursor() as cursor:
            select_all = "select namepokup,avg(money) from salesfor.sales "
            cursor.execute(select_all)
            result = cursor.fetchall()
            # print(result)
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()
    if v == "13":
        with connection.cursor() as cursor:
            select_all = "select nameprod,avg(money) from salesfor.sales "
            cursor.execute(select_all)
            result = cursor.fetchall()
            # print(result)
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()

```

Завдання 2

Додайте механізми для оновлення, видалення та вставки даних до бази даних за допомогою інтерфейсу меню. Користувач не може ввести запити INSERT, UPDATE, DELETE безпосередньо. Забороніть можливість оновлення та видалення усіх даних для кожної таблиці (UPDATE та DELETE без умов).

```

while True:
    v = input("Enter what you want\n1.(INSERT)\n2.(Update)\n3.(Delete)\n")
    if v=="1":
        nameprod=input("Enter name: ")
        namepokup = input("Enter surname: ")
        tovar = input("Enter city: ")
        money = int(input("Enter country: "))
        with connection.cursor() as cursor:
            insert = f"INSERT INTO `Sales` (nameprod, namepokup,tovar,money) VALUES
('{nameprod}','{namepokup}','{tovar}','{money}')"
            cursor.execute(insert)
            connection.commit()
    if v=="2":
        choose=input("What you want change: ")

```

```

        replace=input("And call what you want enter: ")
        idd=int(input("Enter id peron: "))
        with connection.cursor() as cursor:
            create_table = f"UPDATE `Sales` SET {choose} = '{replace}' WHERE id =
{idd}"
            cursor.execute(create_table)
            connection.commit()
    if v=="3":
        fir=input("Enter named colum: ")
        sec=input("Enter and what you want delete: ")
        with connection.cursor() as cursor:
            create_table = f"DELETE FROM `Sales` WHERE {fir}='{sec}';"
            cursor.execute(create_table)
            connection.commit()

```

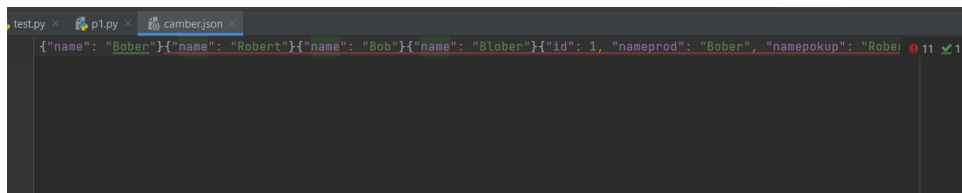
Завдання 3

Додайте до першого завдання можливість збереження результатів фільтрів у файл. Шлях і назву файлу вкажіть у налаштуваннях програми.

```

        file = open(os.path.join("based", "camber.json"), "a")
        json.dump(row, file)
        file.close()

```



```

test.py  p1.py  camber.json
{"name": "Bober"}{"name": "Robert"}{"name": "Bob"}{"name": "Bieber"}{"id": 1, "nameprod": "Bober", "namepoup": "Robe

```