**-національний університет БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ україни**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра комп’ютерних наук

**ЛАБОРАТОРНА РОБОТА №7**

*Варіант 0*

Виконав:

студент групи ІПЗ-21007б

Щур В. М.

Київ – 2024

**Хід виконання**

***1.*** *На платформі Docker, за допомогою файлу docker-compose.yml, створити контейнер з СУБД PostgreSQL або MySQL. Зробити прокидання портів та папок для зберігання БД.*

*Зображення, що містить текст, знімок екрана, програмне забезпечення, монітор

Автоматично згенерований опис*

*Зображення, що містить текст, знімок екрана, Шрифт, програмне забезпечення

Автоматично згенерований опис*

***2.*** *В цьому контейнері, використовуючи мову Python, створити базу даних. Створити в ній необхідні таблиці з відповідними полями (предметна область та дані наведені нижче).*

***3.*** *Визначить типи даних (лічильник, текстовий, числовий тощо) та опис, якщо потрібно.*

***4.*** *Встановіть необхідні властивості полів (розмір поля, маску вводу, значення за замовченням, обмеження та повідомлення про помилку) створених таблиць.*

***5.*** *Визначить первинні ключі в створених таблицях.*

***6.*** *Визначить необхідні зв’язки між таблицями, задайте необхідні параметри забезпечення цілісності даних.*

***7.*** *Заповнить створені таблиці даними (13 ліків, 5 постачальників, 11 поставок)*

**Предметна область:** Магазин одягу (складається з 4 класів)

**Сутності та дані:**

* *Склади*[Номер складу, адреса, завідувач складом, телефон (маска вводу)]
* *Товари*[Код товару, тип (жіночий, чоловічий, дитячий), назва одягу, виробник, номер складу, на якому знаходиться, кількість на складі, ціна]
* *Клієнти*[Код клієнта, назва клієнта, адреса клієнта, телефон клієнта (маска вводу), контактна особа]
* *Продаж*[Код продажу, дата продажу, код клієнта, код товару, кількість купленого одягу, знижка].

**Код:**

import psycopg2  
  
# Підключення до бази даних  
conn = psycopg2.connect(  
 dbname="clothing\_store\_db", user="user", password="password", host="localhost", port="5432"  
)  
  
# Створення таблиць  
with conn:  
 cursor = conn.cursor()  
  
 cursor.execute("""  
 CREATE TABLE IF NOT EXISTS Warehouses (  
 warehouse\_id SERIAL PRIMARY KEY,  
 address VARCHAR(255),  
 manager\_name VARCHAR(100),  
 phone VARCHAR(15)  
 );  
 """)  
  
 cursor.execute("""  
 CREATE TABLE IF NOT EXISTS Products (  
 product\_id SERIAL PRIMARY KEY,  
 product\_type VARCHAR(50) CHECK (product\_type IN ('Жіночий', 'Чоловічий', 'Дитячий')),  
 product\_name VARCHAR(100),  
 manufacturer VARCHAR(100),  
 warehouse\_id INTEGER REFERENCES Warehouses(warehouse\_id),  
 quantity\_on\_hand INTEGER,  
 price NUMERIC(10, 2)  
 );  
 """)  
  
 cursor.execute("""  
 CREATE TABLE IF NOT EXISTS Clients (  
 client\_id SERIAL PRIMARY KEY,  
 client\_name VARCHAR(100),  
 client\_address VARCHAR(255),  
 phone VARCHAR(15),  
 contact\_person VARCHAR(100)  
 );  
 """)  
  
 cursor.execute("""  
 CREATE TABLE IF NOT EXISTS Sales (  
 sale\_id SERIAL PRIMARY KEY,  
 sale\_date DATE,  
 client\_id INTEGER REFERENCES Clients(client\_id),  
 product\_id INTEGER REFERENCES Products(product\_id),  
 quantity\_sold INTEGER,  
 discount NUMERIC(5, 2)  
 );  
 """)  
  
 print("Таблиці успішно створено!")  
  
# Заповнення таблиць даними  
with conn:  
 cursor = conn.cursor()  
  
 cursor.execute("""  
 INSERT INTO Warehouses (address, manager\_name, phone)  
 VALUES  
 ('Вул. Хрещатик, 1, Київ', 'Іванов Іван', '+380661234567'),  
 ('Вул. Пушкіна, 10, Львів', 'Петренко Олександр', '+380661234568'),  
 ('Вул. Грушевського, 20, Одеса', 'Сидоренко Марія', '+380661234569');  
 """)  
  
 cursor.execute("""  
 INSERT INTO Products (product\_type, product\_name, manufacturer, warehouse\_id, quantity\_on\_hand, price)  
 VALUES  
 ('Жіночий', 'Сукня', 'Fashion Ltd', 1, 50, 1200.50),  
 ('Чоловічий', 'Футболка', 'Trend Wear', 1, 150, 350.75),  
 ('Дитячий', 'Штани', 'Kids Fashion', 2, 75, 500.25),  
 ('Жіночий', 'Туфлі', 'Luxury Shoes', 3, 30, 1500.99),  
 ('Чоловічий', 'Сорочка', 'Classic Wear', 1, 100, 700.45),  
 ('Жіночий', 'Спідниця', 'Fashion Ltd', 2, 200, 800.30),  
 ('Дитячий', 'Плаття', 'Cute Kids', 2, 120, 400.00),  
 ('Чоловічий', 'Куртка', 'Winter Gear', 3, 45, 2500.99),  
 ('Жіночий', 'Светр', 'Style Co.', 1, 70, 1100.60),  
 ('Дитячий', 'Костюм', 'Kidswear Co', 1, 80, 750.50),  
 ('Жіночий', 'Шорти', 'Summer Style', 2, 90, 650.20),  
 ('Чоловічий', 'Джинси', 'Denim Works', 3, 55, 1300.40),  
 ('Жіночий', 'Блуза', 'Chic Wear', 1, 120, 950.60),  
 ('Чоловічий', 'Спортивний костюм', 'Sport Fashion', 2, 60, 1500.99),  
 ('Дитячий', 'Шапка', 'Cute Kids', 3, 150, 300.00);  
 """)  
  
 cursor.execute("""  
 INSERT INTO Clients (client\_name, client\_address, phone, contact\_person)  
 VALUES  
 ('Петро Петров', 'вул. Лесі Українки, 5, Київ', '+380661234570', 'Петро Петров'),  
 ('Марія Іванова', 'вул. Сумська, 15, Харків', '+380661234571', 'Марія Іванова'),  
 ('Олександр Семенов', 'вул. Б. Хмельницького, 12, Львів', '+380661234572', 'Олександр Семенов'),  
 ('Наталія Дорошенко', 'вул. Академіка Сахарова, 22, Одеса', '+380661234573', 'Наталія Дорошенко'),  
 ('Іван Коваленко', 'вул. Петра Кучеренка, 8, Дніпро', '+380661234574', 'Іван Коваленко');  
 """)  
  
 cursor.execute("""  
 INSERT INTO Sales (sale\_date, client\_id, product\_id, quantity\_sold, discount)  
 VALUES  
 ('2024-11-01', 1, 1, 2, 5.00),  
 ('2024-11-02', 2, 3, 1, 10.00),  
 ('2024-11-03', 3, 2, 5, 0.00),  
 ('2024-11-04', 4, 5, 3, 7.50),  
 ('2024-11-05', 5, 4, 1, 0.00),  
 ('2024-11-06', 1, 6, 2, 5.00),  
 ('2024-11-07', 2, 8, 4, 15.00),  
 ('2024-11-08', 3, 10, 1, 5.00),  
 ('2024-11-09', 4, 9, 3, 10.00),  
 ('2024-11-10', 5, 12, 2, 0.00);  
 """)  
  
 print("Таблиці успішно заповнено даними!")  
  
# Закриття з'єднання  
conn.close()

***8.*** *Створіть наступні запити:*

* Відобразити інформацію по покупкам, яка містить дату покупки, назву товару, назву клієнта, кількість купленого товару, ціну товару. Відсортувати назви клієнтів за алфавітом;
* Відобразити увесь одяг за вибраним типом (запит з параметром); Порахувати кількість покупок, яку зробив кожний клієнт (підсумковий запит);
* Порахувати вартість кожної покупки без урахування та з урахуванням знижки (запит з обчислювальним полем);
* Порахувати загальну суму грошей, яку витратив кожен клієнт на купівлю одягу (підсумковий запит)
* Відобразити кількість кожного виду одягу на кожному складі (перехресний запит).

**Код:**

import psycopg2  
from datetime import date  
from decimal import Decimal  
  
# Підключення до бази даних  
conn = psycopg2.connect(  
 dbname="clothing\_store\_db", user="user", password="password", host="localhost", port="5432"  
)  
  
with conn:  
 cursor = conn.cursor()  
  
 # 1. Відобразити інформацію по покупкам, відсортовано за назвою клієнта  
 cursor.execute("""  
 SELECT s.sale\_date, p.product\_name, c.client\_name, s.quantity\_sold, p.price  
 FROM Sales s  
 JOIN Products p ON s.product\_id = p.product\_id  
 JOIN Clients c ON s.client\_id = c.client\_id  
 ORDER BY c.client\_name;  
 """)  
 purchases\_info = cursor.fetchall()  
 print("1) Інформація по покупкам (відсортовано за назвою клієнта):")  
 for row in purchases\_info:  
 sale\_date, product\_name, client\_name, quantity\_sold, price = row  
 print(", ".join(  
 map(str, [sale\_date.strftime("%Y-%m-%d"), product\_name, client\_name, quantity\_sold, float(price)])))  
  
 # 2. Відобразити увесь одяг за вибраним типом  
 product\_type = 'Жіночий'  
 cursor.execute("""  
 SELECT product\_name, manufacturer, quantity\_on\_hand, price  
 FROM Products  
 WHERE product\_type = %s;  
 """, (product\_type,))  
 clothing\_by\_type = cursor.fetchall()  
 print("\n2) Увесь одяг за вибраним типом (тип:", product\_type, "):")  
 for row in clothing\_by\_type:  
 product\_name, manufacturer, quantity\_on\_hand, price = row  
 print(", ".join(map(str, [product\_name, manufacturer, quantity\_on\_hand, float(price)])))  
  
 # 3. Порахувати кількість покупок кожного клієнта  
 cursor.execute("""  
 SELECT c.client\_name, COUNT(s.sale\_id) AS purchases\_count  
 FROM Sales s  
 JOIN Clients c ON s.client\_id = c.client\_id  
 GROUP BY c.client\_name;  
 """)  
 client\_purchase\_counts = cursor.fetchall()  
 print("\n3) Кількість покупок кожного клієнта:")  
 for row in client\_purchase\_counts:  
 client\_name, purchases\_count = row  
 print(", ".join(map(str, [client\_name, purchases\_count])))  
  
 # 4. Порахувати вартість кожної покупки (без знижки та зі знижкою)  
 cursor.execute("""  
 SELECT s.sale\_id, p.product\_name, s.quantity\_sold,   
 (p.price \* s.quantity\_sold) AS total\_price,  
 (p.price \* s.quantity\_sold \* (1 - s.discount / 100)) AS discounted\_price  
 FROM Sales s  
 JOIN Products p ON s.product\_id = p.product\_id;  
 """)  
 purchase\_costs = cursor.fetchall()  
 print("\n4) Вартість кожної покупки (без знижки та зі знижкою):")  
 for row in purchase\_costs:  
 sale\_id, product\_name, quantity\_sold, total\_price, discounted\_price = row  
 print(", ".join(map(str, [sale\_id, product\_name, quantity\_sold, float(total\_price), float(discounted\_price)])))  
  
 # 5. Порахувати загальну суму витрат кожного клієнта  
 cursor.execute("""  
 SELECT c.client\_name,  
 SUM(p.price \* s.quantity\_sold \* (1 - s.discount / 100)) AS total\_spent  
 FROM Sales s  
 JOIN Clients c ON s.client\_id = c.client\_id  
 JOIN Products p ON s.product\_id = p.product\_id  
 GROUP BY c.client\_name;  
 """)  
 total\_spent\_by\_clients = cursor.fetchall()  
 print("\n5) Загальна сума витрат кожного клієнта на купівлю одягу:")  
 for row in total\_spent\_by\_clients:  
 client\_name, total\_spent = row  
 print(", ".join(map(str, [client\_name, float(total\_spent)])))  
  
 # 6. Відобразити кількість кожного виду одягу на кожному складі  
 cursor.execute("""  
 SELECT w.address, p.product\_name, SUM(p.quantity\_on\_hand) AS total\_quantity  
 FROM Products p  
 JOIN Warehouses w ON p.warehouse\_id = w.warehouse\_id  
 GROUP BY w.address, p.product\_name;  
 """)  
 stock\_quantities = cursor.fetchall()  
 print("\n6) Кількість кожного виду одягу на кожному складі:")  
 for row in stock\_quantities:  
 address, product\_name, total\_quantity = row  
 print(", ".join(map(str, [address, product\_name, total\_quantity])))  
  
# Закриття з'єднання  
conn.close()

**Результат:**

*Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис*

*Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис*

*Зображення, що містить текст, знімок екрана, Шрифт, дизайн

Автоматично згенерований опис*

*Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис*

***9.*** *На мові Python написати програму, що підключається до створеної БД, виводить всі таблиці (структура + дані, які в ній зберігаються) та результати виконання запитів в консоль в форматованому вигляді (заголовки стовпців + всі стовпці рівні).*

**Код:**

import psycopg2  
from prettytable import PrettyTable  
  
  
def print\_table\_data(cursor, table\_name):  
 cursor.execute(f"SELECT \* FROM {table\_name};")  
 rows = cursor.fetchall()  
 colnames = [desc[0] for desc in cursor.description]  
  
 table = PrettyTable()  
 table.field\_names = colnames  
 for row in rows:  
 table.add\_row(row)  
  
 print(f"\nTable: {table\_name}")  
 print(table)  
  
  
# Функція для виведення структури всіх таблиць та їхніх даних  
def show\_all\_tables\_data(cursor, tables):  
 for table in tables:  
 print\_table\_data(cursor, table)  
  
  
# Підключення до бази даних  
conn = psycopg2.connect(  
 dbname="clothing\_store\_db", user="user", password="password", host="localhost", port="5432"  
)  
cursor = conn.cursor()  
  
# Список таблиць  
tables = ['Warehouses', 'Products', 'Clients', 'Sales']  
  
# Виведення даних з усіх таблиць  
show\_all\_tables\_data(cursor, tables)  
  
# Закриття з'єднання  
cursor.close()  
conn.close()

**Результат:**

*Зображення, що містить текст, знімок екрана

Автоматично згенерований опис*

***10.*** *На платформі Docker створити контейнер з графічним клієнтом (адмінка) для управління БД. Запустити його і підключитись до створеної БД. Переконатись, що всі таблиці і запити створені вірно.*

*Зображення, що містить текст, знімок екрана, Шрифт, програмне забезпечення

Автоматично згенерований опис*

*Зображення, що містить текст, знімок екрана, число, Шрифт

Автоматично згенерований опис*

***11.*** *Завантажити проект на GitHub, попередньо додавши до файлу .gitignore всі технічні папки та файли.*