**національний університет БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ україни**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**ЛАБОРАТОРНА РОБОТА №6**

**З Програмування Python**

Виконала:

студентка групи КН-20001б

Герасімова Д.І.

Київ – 2023

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.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

**Результат:**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

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 (4 постачальника, 22 поставки, матеріали: деревина, лак, сталеві деталі) Предметна область: Відділ поставок (складається з 3 класів). Сутності та дані: Постачальники[Код постачальника, назва компанії постачальника, контактна особа, телефон(маска вводу), розрахунковий рахунок], Матеріали, що поставляються[Код матеріалу, назва матеріалу, ціна], Поставки[номер поставки, дата поставки, код постачальника, код матеріалу, кількість днів, протягом яких здійснюється поставка(1-7; обмеження, повідомлення про помилку), кількість матеріалів, які поставляються].

**db\_setup.py**

*import* sqlite3  
  
*# Підключення до бази даних*conn = sqlite3.connect("gerda\_database.db")  
cur = conn.cursor()  
  
*# Створення таблиці "Постачальники"*cur.execute("""  
 CREATE TABLE IF NOT EXISTS suppliers (  
 supplier\_code INTEGER PRIMARY KEY,  
 company\_name TEXT NOT NULL,  
 contact\_person TEXT,  
 phone TEXT,  
 bank\_account TEXT  
 );  
""")  
  
*# Створення таблиці "Матеріали"*cur.execute("""  
 CREATE TABLE IF NOT EXISTS materials (  
 material\_code INTEGER PRIMARY KEY,  
 material\_name TEXT NOT NULL,  
 price REAL NOT NULL  
 );  
""")  
  
*# Створення таблиці "Поставки"*cur.execute("""  
 CREATE TABLE IF NOT EXISTS supplies (  
 supply\_number INTEGER PRIMARY KEY,  
 supply\_date DATE,  
 supplier\_code INTEGER REFERENCES suppliers(supplier\_code),  
 material\_code INTEGER REFERENCES materials(material\_code),  
 delivery\_days INTEGER CHECK (delivery\_days >= 1 AND delivery\_days <= 7),  
 quantity INTEGER,  
 CONSTRAINT fk\_supplier FOREIGN KEY (supplier\_code) REFERENCES suppliers(supplier\_code),  
 CONSTRAINT fk\_material FOREIGN KEY (material\_code) REFERENCES materials(material\_code)  
 );  
""")  
  
*# Додавання даних у таблицю "Постачальники"*cur.executemany("""  
 INSERT INTO suppliers (company\_name, contact\_person, phone, bank\_account)  
 VALUES (?, ?, ?, ?);  
""", [  
 ('Supplier1', 'John Doe', '+123456789', '1234567890123456'),  
 ('Supplier2', 'Jane Smith', '+987654321', '6543210987654321'),  
 ('Supplier3', 'Bob Johnson', '+111222333', '1111222233334444'),  
 ('Supplier4', 'Alice Brown', '+444555666', '4444555566667777')  
])  
  
*# Додавання даних у таблицю "Матеріали"*cur.executemany("""  
 INSERT INTO materials (material\_name, price)  
 VALUES (?, ?);  
""", [  
 ('Wood', 50.00),  
 ('Lacquer', 20.00),  
 ('Steel Parts', 100.00)  
])  
  
*# Додавання даних у таблицю "Поставки"*cur.executemany("""  
 INSERT INTO supplies (supply\_date, supplier\_code, material\_code, delivery\_days, quantity)  
 VALUES (?, ?, ?, ?, ?);  
""", [  
 ('2023-11-16', 1, 1, 2, 100),  
 ('2023-11-17', 2, 2, 3, 50),  
 ('2023-11-18', 3, 3, 1, 200),  
 ('2023-11-19', 4, 1, 5, 150)  
])  
  
*# Виведення заголовків стовпців для таблиці "Постачальники"*cur.execute("PRAGMA table\_info(suppliers);")  
columns = cur.fetchall()  
print("\nСтруктура таблиці suppliers:")  
*for* column *in* columns:  
 print(column[1], end='\t')  
print()  
  
*# Виведення даних для таблиці "Постачальники"*cur.execute("SELECT \* FROM suppliers;")  
data = cur.fetchall()  
print("\nДані таблиці suppliers:")  
*for* row *in* data:  
 print(row)  
  
*# Виведення заголовків стовпців для таблиці "Матеріали"*cur.execute("PRAGMA table\_info(materials);")  
columns = cur.fetchall()  
print("\nСтруктура таблиці materials:")  
*for* column *in* columns:  
 print(column[1], end='\t')  
print()  
  
*# Виведення даних для таблиці "Матеріали"*cur.execute("SELECT \* FROM materials;")  
data = cur.fetchall()  
print("\nДані таблиці materials:")  
*for* row *in* data:  
 print(row)  
  
*# Виведення заголовків стовпців для таблиці "Поставки"*cur.execute("PRAGMA table\_info(supplies);")  
columns = cur.fetchall()  
print("\nСтруктура таблиці supplies:")  
*for* column *in* columns:  
 print(column[1], end='\t')  
print()  
  
*# Виведення даних для таблиці "Поставки"*cur.execute("SELECT \* FROM supplies;")  
data = cur.fetchall()  
print("\nДані таблиці supplies:")  
*for* row *in* data:  
 print(row)

**Результат:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание**

**8.** Створіть наступні запити: Відобразити всі поставки, які здійснюються за 3 або менше днів. Відсортувати назви постачальників за алфавітом; Порахувати суму, яку треба сплатити за кожну поставку (запит з обчислювальним полем); Відобразити всі поставки обраного матеріалу (запит з параметром); Порахувати кількість кожного матеріалу, що поставляється кожним постачальником (перехресний запит); Порахувати загальну кількість кожного матеріалу (підсумковий запит); Порахувати кількість поставок від кожного постачальника (підсумковий запит).

*# Виведення результатів запитів  
  
# Запит 1: Відобразити всі поставки, які здійснюються за 3 або менше днів. Відсортувати назви постачальників за алфавітом*query1 = """  
 SELECT s.supply\_number, s.supply\_date, sp.company\_name  
 FROM supplies s  
 JOIN suppliers sp ON s.supplier\_code = sp.supplier\_code  
 WHERE s.delivery\_days <= 3  
 ORDER BY sp.company\_name;  
"""  
cur.execute(query1)  
result1 = cur.fetchall()  
print("\nРезультат запиту 1:")  
*for* row *in* result1:  
 print(row)  
  
*# Запит 2: Порахувати суму, яку треба сплатити за кожну поставку (запит з обчислювальним полем)*query2 = """  
 SELECT supply\_number, SUM(quantity \* m.price) AS total\_cost  
 FROM supplies s  
 JOIN materials m ON s.material\_code = m.material\_code  
 GROUP BY supply\_number;  
"""  
cur.execute(query2)  
result2 = cur.fetchall()  
print("\nРезультат запиту 2:")  
*for* row *in* result2:  
 print(row)  
  
*# Запит 3: Відобразити всі поставки обраного матеріалу (запит з параметром)*selected\_material\_code = 1 *# Приклад значення*query3 = f"""  
 SELECT \*  
 FROM supplies  
 WHERE material\_code = {selected\_material\_code};  
"""  
cur.execute(query3)  
result3 = cur.fetchall()  
print("\nРезультат запиту 3:")  
*for* row *in* result3:  
 print(row)  
  
*# Запит 4: Порахувати кількість кожного матеріалу, що поставляється кожним постачальником (перехресний запит)*query4 = """  
 SELECT sp.company\_name, m.material\_name, SUM(s.quantity) AS total\_quantity  
 FROM supplies s  
 JOIN suppliers sp ON s.supplier\_code = sp.supplier\_code  
 JOIN materials m ON s.material\_code = m.material\_code  
 GROUP BY sp.company\_name, m.material\_name;  
"""  
cur.execute(query4)  
result4 = cur.fetchall()  
print("\nРезультат запиту 4:")  
*for* row *in* result4:  
 print(row)  
  
*# Запит 5: Порахувати загальну кількість кожного матеріалу (підсумковий запит)*query5 = """  
 SELECT m.material\_name, SUM(s.quantity) AS total\_quantity  
 FROM supplies s  
 JOIN materials m ON s.material\_code = m.material\_code  
 GROUP BY m.material\_name;  
"""  
cur.execute(query5)  
result5 = cur.fetchall()  
print("\nРезультат запиту 5:")  
*for* row *in* result5:  
 print(row)  
  
*# Запит 6: Порахувати кількість поставок від кожного постачальника (підсумковий запит)*query6 = """  
 SELECT sp.company\_name, COUNT(\*) AS total\_shipments  
 FROM supplies s  
 JOIN suppliers sp ON s.supplier\_code = sp.supplier\_code  
 GROUP BY sp.company\_name;  
"""  
cur.execute(query6)  
result6 = cur.fetchall()  
print("\nРезультат запиту 6:")  
*for* row *in* result6:  
 print(row)  
  
*# Закриття підключення та збереження змін*conn.commit()  
conn.close()

**Результат:**

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

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).

**display\_tables\_data.py**

*import* sqlite3  
*from* prettytable *import* PrettyTable  
*# Створення PrettyTable для виводу результатів  
def* print\_pretty\_table(title, description, data):  
 table = PrettyTable()  
 table.field\_names = [desc[0] *for* desc *in* description]  
 table.add\_rows(data)  
 print(f"\n{title}:")  
 print(table)  
  
*# Підключення до бази даних*conn = sqlite3.connect("gerda\_database.db")  
cur = conn.cursor()  
  
*# Отримання списку всіх таблиць у базі даних*cur.execute("SELECT name FROM sqlite\_master WHERE type='table';")  
tables = cur.fetchall()  
  
*# Для кожної таблиці виводимо її структуру та дані  
for* table *in* tables:  
 table\_name = table[0]  
  
 *# Отримання структури таблиці* cur.execute(f"PRAGMA table\_info({table\_name});")  
 columns = cur.fetchall()  
 print(f"\nСтруктура таблиці {table\_name}:")  
 *for* column *in* columns:  
 print(column[1], column[2])  
  
 *# Отримання даних таблиці* cur.execute(f"SELECT \* FROM {table\_name};")  
 data = cur.fetchall()  
 print(f"\nДані таблиці {table\_name}:")  
 *for* row *in* data:  
 print(row)  
  
*# Запит 1*query1 = """  
 SELECT s.supply\_number, s.supply\_date, sp.company\_name  
 FROM supplies s  
 JOIN suppliers sp ON s.supplier\_code = sp.supplier\_code  
 WHERE s.delivery\_days <= 3  
 ORDER BY sp.company\_name;  
"""  
cur.execute(query1)  
result1 = cur.fetchall()  
print\_pretty\_table("Результат запиту 1", cur.description, result1)  
  
*# Запит 2*query2 = """  
 SELECT supply\_number, SUM(quantity \* m.price) AS total\_cost  
 FROM supplies s  
 JOIN materials m ON s.material\_code = m.material\_code  
 GROUP BY supply\_number;  
"""  
cur.execute(query2)  
result2 = cur.fetchall()  
print\_pretty\_table("Результат запиту 2", cur.description, result2)  
  
*# Запит 3*selected\_material\_code = 1 query3 = f"""  
 SELECT \*  
 FROM supplies  
 WHERE material\_code = {selected\_material\_code};  
"""  
cur.execute(query3)  
result3 = cur.fetchall()  
print\_pretty\_table("Результат запиту 3", cur.description, result3)  
  
*# Запит 4*query4 = """  
 SELECT sp.company\_name, m.material\_name, SUM(s.quantity) AS total\_quantity  
 FROM supplies s  
 JOIN suppliers sp ON s.supplier\_code = sp.supplier\_code  
 JOIN materials m ON s.material\_code = m.material\_code  
 GROUP BY sp.company\_name, m.material\_name;  
"""  
cur.execute(query4)  
result4 = cur.fetchall()  
print\_pretty\_table("Результат запиту 4", cur.description, result4)  
  
*# Запит 5*query5 = """  
 SELECT m.material\_name, SUM(s.quantity) AS total\_quantity  
 FROM supplies s  
 JOIN materials m ON s.material\_code = m.material\_code  
 GROUP BY m.material\_name;  
"""  
cur.execute(query5)  
result5 = cur.fetchall()  
print\_pretty\_table("Результат запиту 5", cur.description, result5)  
  
*# Запит 6*query6 = """  
 SELECT sp.company\_name, COUNT(\*) AS total\_shipments  
 FROM supplies s  
 JOIN suppliers sp ON s.supplier\_code = sp.supplier\_code  
 GROUP BY sp.company\_name;  
"""  
cur.execute(query6)  
result6 = cur.fetchall()  
print\_pretty\_table("Результат запиту 6", cur.description, result6)  
  
*# Закриття підключення*conn.close()

**Результат:**

**Изображение выглядит как текст, снимок экрана, меню, дизайн

Автоматически созданное описание**

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Запит 1:

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

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

Изображение выглядит как текст, снимок экрана, Шрифт, Цвет электрик

Автоматически созданное описание