JOIN

Бутенко Сергій

План лекції

- Формулювання задачі
- Практика
- Синтаксис JOIN

ЗАДАЧА

Orders

id	customer_id	product_id	order_date
1	1	1	12:22:23

ЗАДАЧА

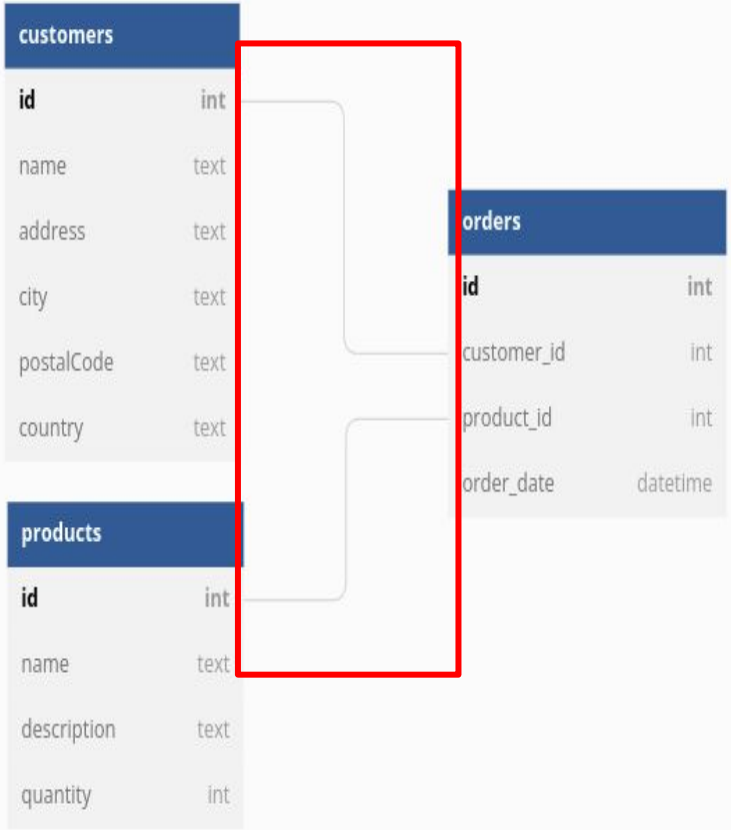
Задача

Поєднати дані з таблиць, які містять інформацію про замовлення, покупців і продукти

Orders

id	customer_id	product_id	order_date
1	1	1	12:22:23

Визначення



JOIN — команда поєднання двох, або більше таблиць, які мають між собою зв’язок.

Визначення


customers
id int
name text
address text
city text
postalCode text
country text
products
id int
name text
description text
quantity int
orders
id int
customer_id int
product_id int
order_date datetime

Огляд таблиці

id	customer_id	product_id	order_date
1	1	1	12:22:23

id	name	name	description	order_date
1	Sergii	солодка вода	з цукром	12:22:23

Огляд таблиці

id	name	name	description	order_date
1	Sergii	солодка вода	з цукром	12:22:23

JOIN. Синтаксис

```
SELECT orders.id, customers.name, products.name, products.description,
orders.order_date
FROM orders
JOIN customers ON orders.customer_id = customers.id
JOIN products ON orders.product_id = products.id
```

JOIN. Синтаксис

```
SELECT orders.id, customers.name, products.name, products.description,
orders.order_date
FROM orders
JOIN customers ON orders.customer_id = customers.id
JOIN products ON orders.product_id = products.id
```

JOIN. Синтаксис

```
SELECT orders.id, customers.name, products.name, products.description,
orders.order_date
FROM orders
JOIN customers ON orders.customer_id = customers.id
JOIN products ON orders.product_id = products.id
```

JOIN. Синтаксис

```
SELECT orders.id, customers.name, products.name, products.description,
orders.order_date
FROM orders
JOIN customers ON orders.customer_id = customers.id
JOIN products ON orders.product_id = products.id
```

JOIN. Синтаксис

```
SELECT orders.id, customers.name, products.name, products.description,
orders.order_date
FROM orders
JOIN customers ON orders.customer_id = customers.id
JOIN products ON orders.product_id = products.id
```

JOIN. Синтаксис

```
SELECT orders.id, customers.name, products.name, products.description,
orders.order_date
FROM orders
JOIN customers ON orders.customer_id = customers.id
JOIN products ON orders.product_id = products.id
```

JOIN. Синтаксис

```
SELECT orders.id, customers.name, products.name, products.description,
orders.order_date
FROM orders
JOIN customers ON orders.customer_id = customers.id
JOIN products ON orders.product_id = products.id
```

JOIN. Синтаксис

```
SELECT orders.id, customers.name, products.name, products.description,
orders.order_date
FROM orders
JOIN customers ON orders.customer_id = customers.id
JOIN products ON orders.product_id = products.id
```

Практика

Задача

Створити тест, що перевіряє правильність даних після виконання команди **JOIN** в таблиці **`orders`**.

А саме порядок даних і дані, які виводяться

◎ Практика

database.py

modules > common > database.py > ...

```python
import sqlite3


class Database():

    def __init__(self):
        self.connection = sqlite3.connect(r'/home/sbutenko/rep
        self.cursor = self.connection.cursor()

    def test_connection(self):
        sqlite_select_Query = "SELECT sqlite_version();"
        self.cursor.execute(sqlite_select_Query)
        record = self.cursor.fetchall()
        print(f"Connected successfully. SQLite Database Versio

    def get_all_users(self):
        query = "SELECT name, address, city FROM customers"
        self.cursor.execute(query)
        record = self.cursor.fetchall()
        return record
```

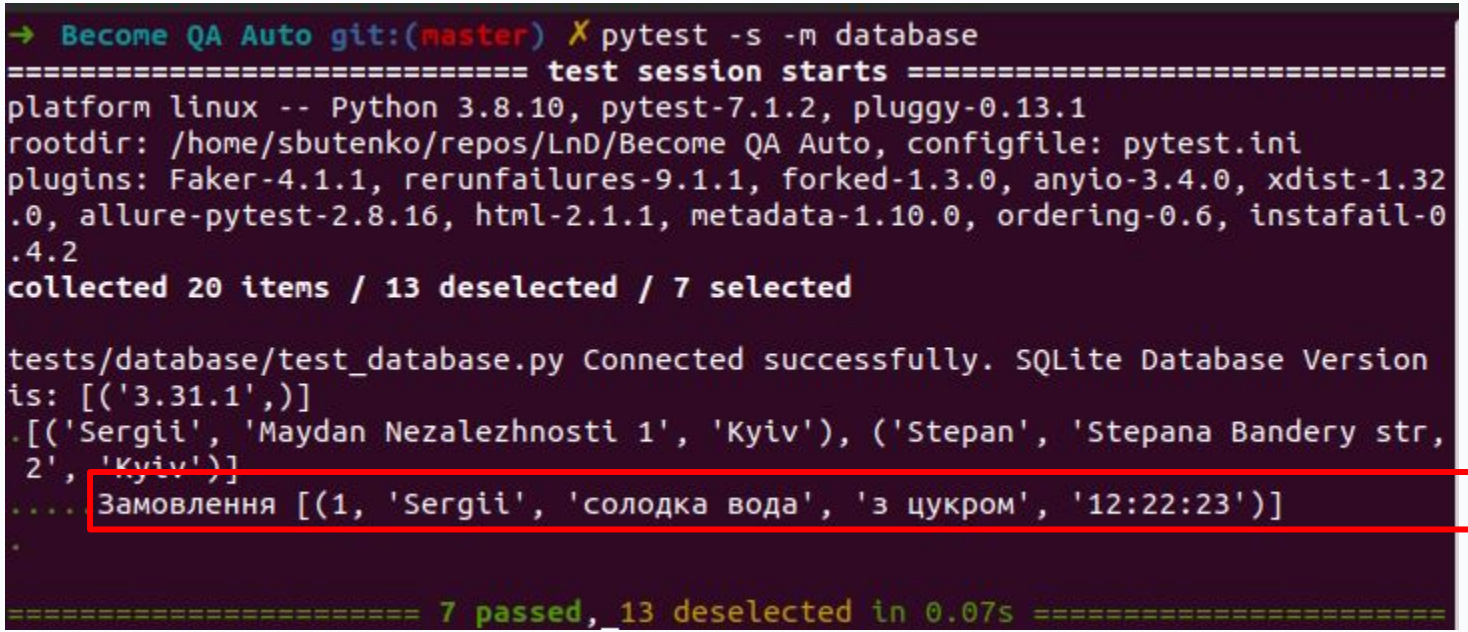
◎ Практика

```
50      def get_detailed_orders(self):
51          query = "SELECT orders.id, customers.name, products.name, \
52                   products.description, orders.order_date \
53                   FROM orders \
54                   JOIN customers ON orders.customer_id = customers.id \
55                   JOIN products ON orders.product_id = products.id"
56          self.cursor.execute(query)
57          record = self.cursor.fetchall()
58          return record
```

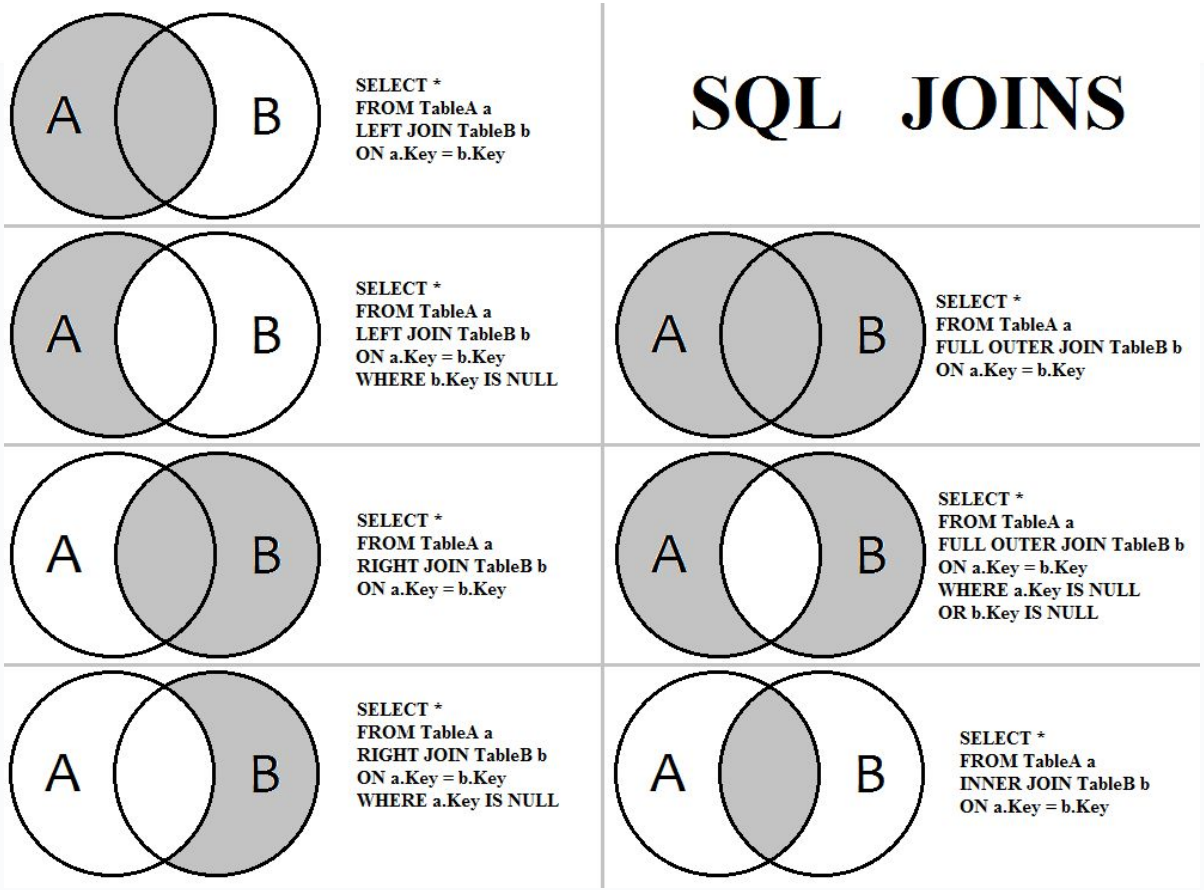
Практика

```
58  @pytest.mark.database
59  def test_detailed_orders():
60      db = Database()
61      orders = db.get_detailed_orders()
62      print("Замовлення", orders)
63      # Check quantity of orders equal to 1
64      assert len(orders) == 1
65
66      # Check struture of data
67      assert orders[0][0] == 1
68      assert orders[0][1] == 'Sergii'
69      assert orders[0][2] == 'солодка вода'
70      assert orders[0][3] == 'з цукром'
71
```

◎ Практика


→ Become QA Auto git:(master) ✗ pytest -s -m database
============================= test session starts ==============================
platform linux -- Python 3.8.10, pytest-7.1.2, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto, configfile: pytest.ini
plugins: Faker-4.1.1, rerunfailures-9.1.1, forked-1.3.0, anyio-3.4.0, xdist-1.32
.0, allure-pytest-2.8.16, html-2.1.1, metadata-1.10.0, ordering-0.6, instafail-0
.4.2
collected 20 items / 13 deselected / 7 selected
tests/database/test_database.py Connected successfully. SQLite Database Version
is: [('3.31.1',)]
.[('Sergii', 'Maydan Nezalezhnosti 1', 'Kyiv'), ('Stepan', 'Stepana Bandery str,
2', 'Kyiv')]
....Замовлення [(1, 'Sergii', 'солодка вода', 'з цукром', '12:22:23')]
.
======================= 7 passed, 13 deselected in 0.07s =======================

JOIN. Загальна інформація.



Підсумки

- Вивчили, що для поєднання даних з 2х і більше таблиць використовується команда **JOIN**
- Ознайомилися з синтаксисом команди **JOIN**
- Написали ще один тест для нашої бази даних