PROMETHEUS

Фільтрація даних

Бутенко Сергій

План лекції

- Загальна інформація
- Практика
- Синтаксис WHERE

Фільтрування даних

Customers

id	name	address	city	postalCode	country
1	Sergii	Maydan Nezalezhnosti 1	Kyiv	3127	Ukraine
2	Stepan	Stepana Bandery str, 2	Kyiv	2055	Ukraine

Products

id	name	description	quantity
1	солодка вода	з цукром	10
2	солодка вода	з цукрозамінником	10
3	молоко	натуральне незбиране	10

Фільтрування даних. Синтаксис

```
SELECT column1, column2, columnN FROM table_name
WHERE column1 = ‘SEARCH_CRITERIA’
```

Фільтрування даних. Синтаксис

```
SELECT column1, column2, columnN FROM table_name
WHERE column1 = ‘SEARCH_CRITERIA’
```

Фільтрування даних. Синтаксис

```
SELECT column1, column2, columnN FROM table_name
WHERE column1 = ‘SEARCH_CRITERIA’
```

Задача

Задача

Визначити адресу для доставки товару покупцю з іменем Сергій

WHERE. запит

customers	
id	**int**
name	text
address	text
city	text
postalCode	text
country	text

WHERE. запит


```
SELECT address, city, postalCode, country
FROM customers
```

WHERE. запит


```
SELECT address, city, postalCode, country
FROM customers
WHERE name = ‘Sergii’
```

WHERE. запит

address	city	postalCode	country
Maydan Nezalezhnosti 1	Kyiv	3127	Ukraine

WHERE. Умови

Оператор	Опис
=	Дорівнює значенню
>	Більше ніж
<	Менше ніж
>=	Більше дорівнює ніж
<=	Менше дорівнює ніж
<>	Не дорівнює

```
SELECT address, city, postalCode, country FROM customers
WHERE name = 'Sergii'
```

◎ Практика

EXPLORER

OPEN EDITORS

BECOME QA AUTO

- __pycache__
- .pytest_cache
- config
- modules
  - __pycache__
  - api
  - common
    - __pycache__
    - __init__.py
    - database.py 1, U
  - ui
- __init__.py U
- tests
  - __pycache__
  - api
  - database
    - __pycache__

database.py

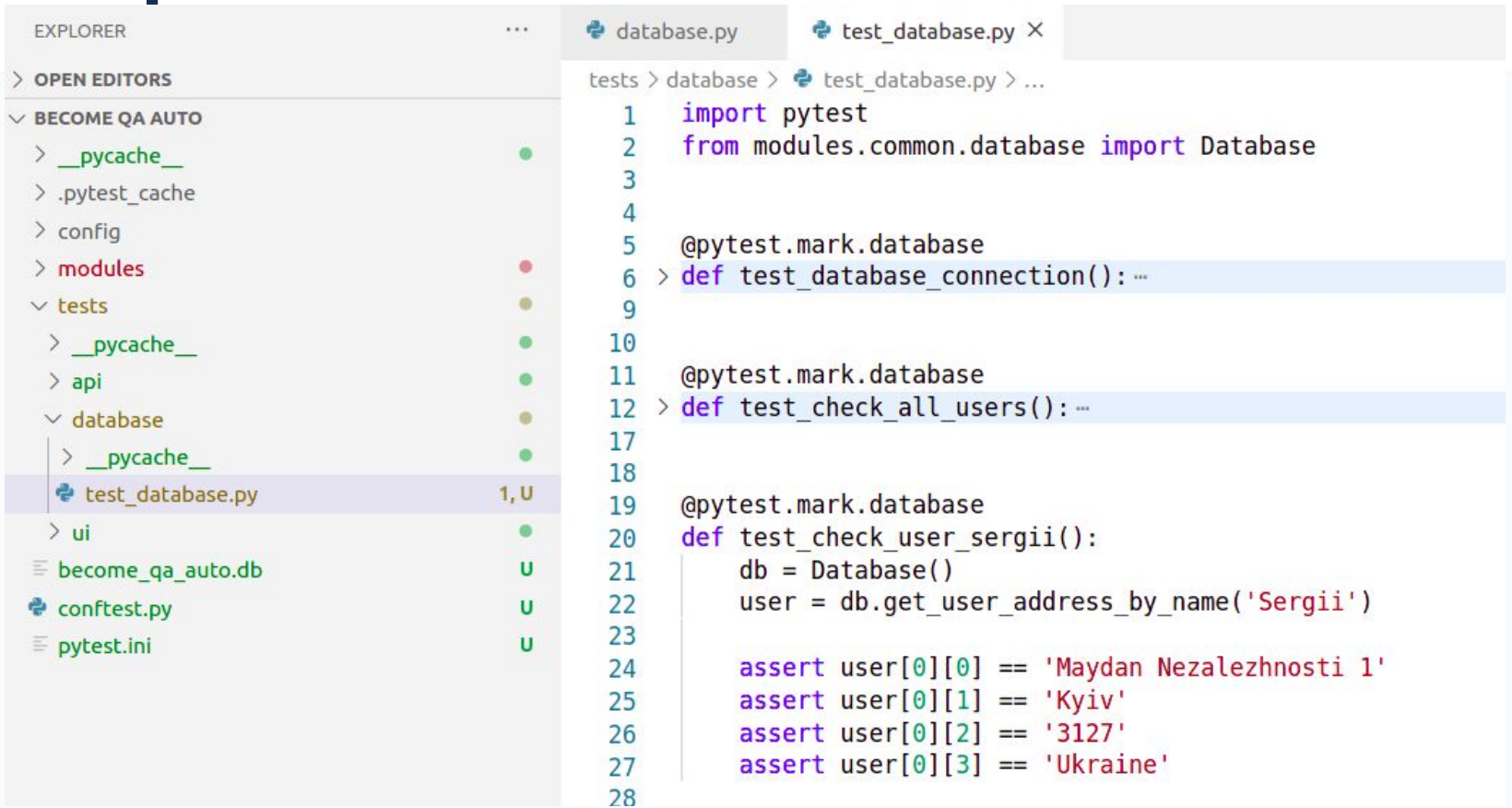
modules > common > database.py > ...

```python
import sqlite3


class Database():

    def __init__(self):
        self.connection = sqlite3.connect(r'/home/sbutenko/repos/LnD/Become QA Auto'
        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 Version is: {record}")

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

```

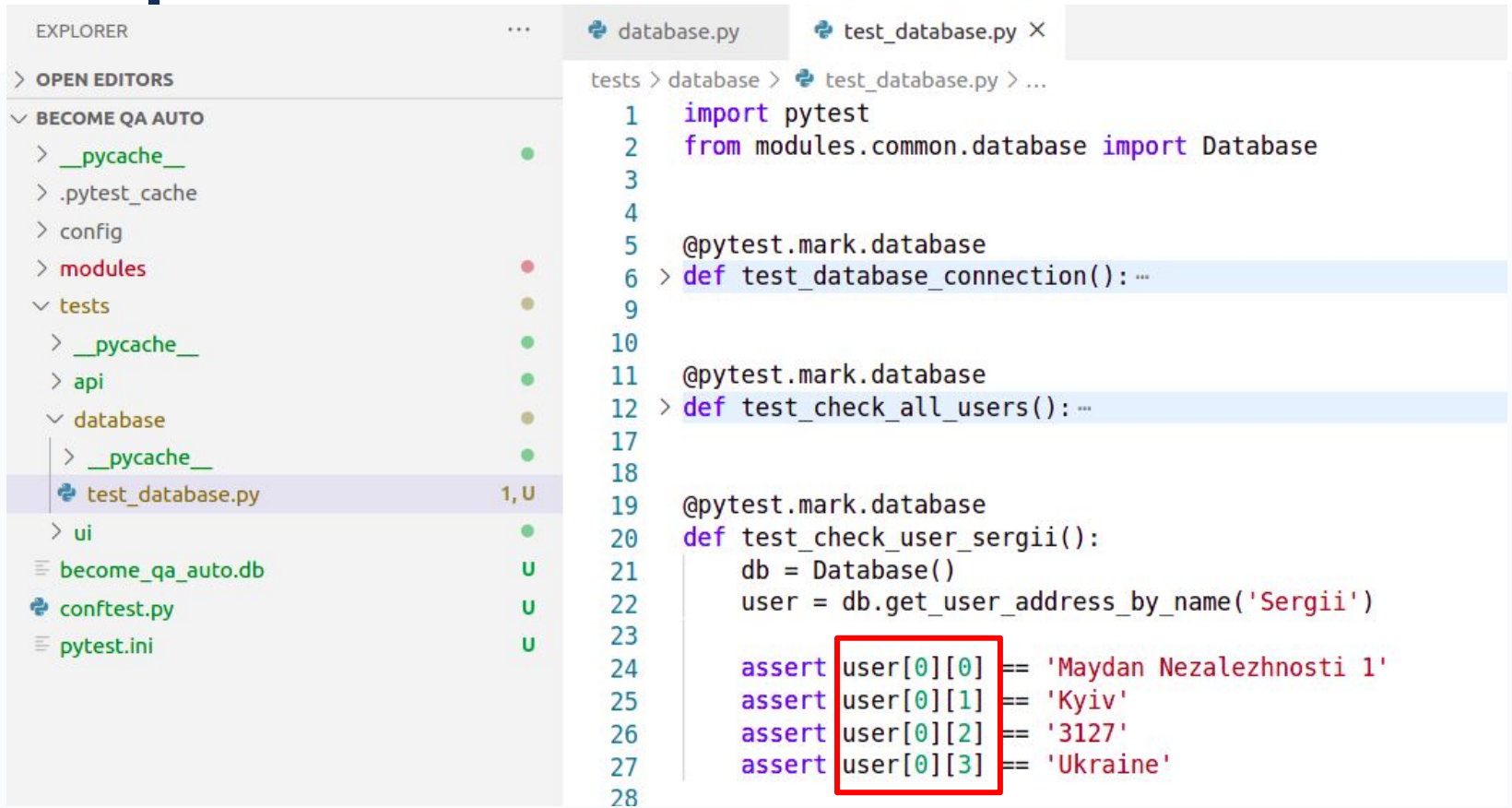
Практика

```
7         self.connection = sqlite3.connect(r'/home/sbutenko/repos/LnD/Become QA Auto' + r'/become_qa
8         self.cursor = self.connection.cursor()
9
10    def test_connection(self):
11        sqlite_select_Query = "SELECT sqlite_version();"
12        self.cursor.execute(sqlite_select_Query)
13        record = self.cursor.fetchall()
14        print(f"Connected successfully. SQLite Database Version is: {record}")
15
16    def get_all_users(self):
17        query = "SELECT name, address, city FROM customers"
18        self.cursor.execute(query)
19        record = self.cursor.fetchall()
20        return record
21
22    def get_user_address_by_name(self, name):
23        query = f"SELECT address, city, postalCode, country FROM customers WHERE name = '{name}'"
24        self.cursor.execute(query)
25        record = self.cursor.fetchall()
26        return record
27
```

◎ Практика


EXPLORER
OPEN EDITORS
BECOME QA AUTO
__pycache__
.pytest_cache
config
modules
tests
__pycache__
api
database
__pycache__
test_database.py
1, U
ui
become_qa_auto.db
U
conftest.py
U
pytest.ini
U
database.py
test_database.py
tests > database > test_database.py > ...
1 import pytest
2 from modules.common.database import Database
3
4
5 @pytest.mark.database
6 def test_database_connection(): ...
9
10
11 @pytest.mark.database
12 def test_check_all_users(): ...
17
18
19 @pytest.mark.database
20 def test_check_user_sergii():
21 db = Database()
22 user = db.get_user_address_by_name('Sergii')
23
24 assert user[0][0] == 'Maydan Nezalezhnosti 1'
25 assert user[0][1] == 'Kyiv'
26 assert user[0][2] == '3127'
27 assert user[0][3] == 'Ukraine'
28

◎ Практика


EXPLORER
OPEN EDITORS
BECOME QA AUTO
__pycache__
.pytest_cache
config
modules
tests
__pycache__
api
database
__pycache__
test_database.py 1, U
ui
become_qa_auto.db U
conftest.py U
pytest.ini U
database.py
test_database.py
tests > database > test_database.py > ...
1 import pytest
2 from modules.common.database import Database
3
4
5 @pytest.mark.database
6 def test_database_connection():
9
10
11 @pytest.mark.database
12 def test_check_all_users():
17
18
19 @pytest.mark.database
20 def test_check_user_sergii():
21 db = Database()
22 user = db.get_user_address_by_name('Sergii')
23
24 assert user[0][0] == 'Maydan Nezalezhnosti 1'
25 assert user[0][1] == 'Kyiv'
26 assert user[0][2] == '3127'
27 assert user[0][3] == 'Ukraine'
28

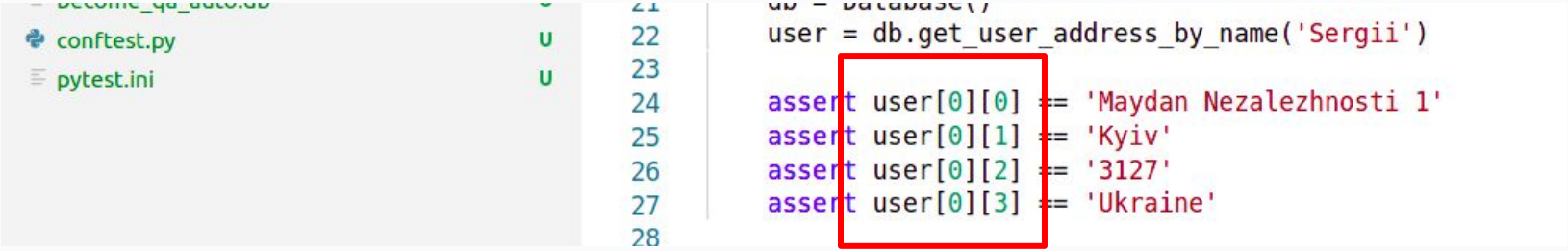
◎ Практика

```
21
22      def get_user_address_by_name(self, name):
23          query = f"SELECT address, city, postalCode, country FROM customers WHERE name = '{name}'"
24          self.cursor.execute(query)
25          record = self.cursor.fetchall()
26          return record
27
```

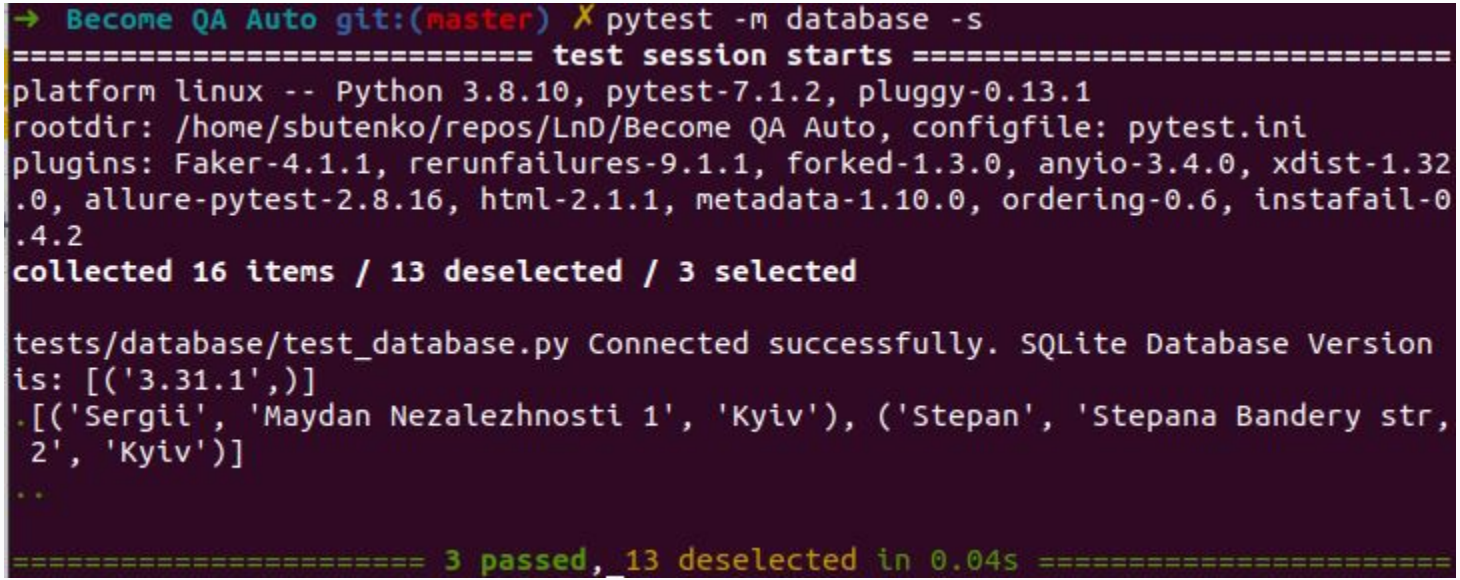

```
21      db = database()
22      user = db.get_user_address_by_name('Sergii')
23
24      assert user[0][0] == 'Maydan Nezalezhnosti 1'
25      assert user[0][1] == 'Kyiv'
26      assert user[0][2] == '3127'
27      assert user[0][3] == 'Ukraine'
28
```

◎ Практика

```
21
22      def get_user_address_by_name(self, name):
23          query = f"SELECT address, city, postalCode, country FROM customers WHERE name = '{name}'"
24          self.cursor.execute(query)
25          record = self.cursor.fetchall()
26          return record
27
```



Практика


→ Become QA Auto git:(master) ✗ pytest -m database -s
============================= 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 16 items / 13 deselected / 3 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')]
..
====================== 3 passed, 13 deselected in 0.04s ======================

Підсумки

- Вивчили, що дані з таблиці під час операцій отримання, зміни чи видалення можна фільтрувати за допомогою команди **WHERE**
- Ознайомилися з синтаксисом команди **WHERE**
- Написали ще один тест для нашої бази даних