



Become QA Auto

PROMETHEUS



Створення даних

Бутенко Сергій



План лекції



Формулювання задачі



Практика



Синтаксис INSERT



ЗАДАЧА



Задача

Зареєструвати новий товар в таблиці **products**
для його наступної реалізації

◎ INSERT. Загальна інформація.





INSERT



Products

id	name	description	quantity
1	солodka вода	з цукром	25
2	солodka вода	з цукрозамінником	10
3	молоко	натуральне незбиране	10



INSERT. Синтаксис



```
INSERT INTO products (id, name, description, quantity)  
VALUES (4, 'печиво', 'солодке', 30);
```



INSERT. Синтаксис



```
INSERT INTO products (id, name, description, quantity)  
VALUES (4, 'печиво', 'солодке', 30);
```



INSERT. Синтаксис



```
INSERT INTO products (id, name, description, quantity)  
VALUES (4, 'печиво', 'солодке', 30);
```




INSERT. Синтаксис



```
INSERT INTO products (id, name, description, quantity)  
VALUES (4, 'печиво', 'солодке', 30);
```



Задача

Створити тест, що перевіряє правильність даних після виконання запиту **INSERT** для таблиці **products**.
А саме створення нового продукту



Практика



EXPLORER

> OPEN EDITORS

✓ BECOME QA AUTO

- > __pycache__
- > .pytest_cache
- > config
- ▼ modules
 - > __pycache__
 - > api
- ▼ common
 - > __pycache__
 - + __init__.py
 - + database.py 2, U
- ▼ ui / page_objects
 - + __init__.py
 - + __init__.py U
- ▼ tests
 - > __pycache__
 - > api

```
database.py  test_database.py
modules > common > + database.py > ...
1  import sqlite3
2
3
4  class Database():
5
6      def __init__(self):
7          self.connection = sqlite3.connect(r'/home/sbutenko/rep
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 Versio
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
```



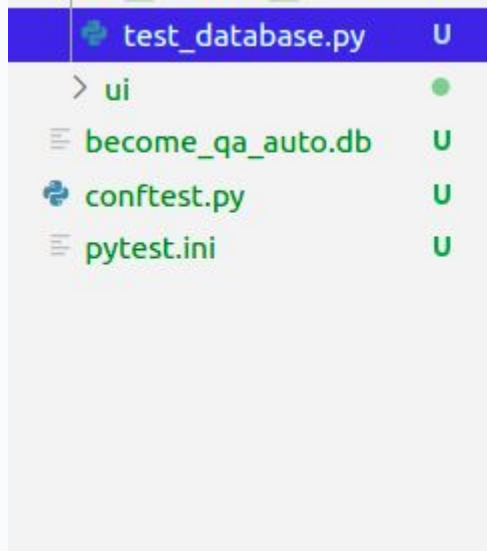
Практика



```
39     def insert_product(self, product_id, name, description, qnt):
40         query = f"INSERT INTO products (id, name, description, quantity) \
41             VALUES ({product_id}, '{name}', '{description}', {qnt})"
42         self.cursor.execute(query)
43         self.connection.commit()
```



Практика



```
36     assert water_qnt[0][0] == 25
37
38
39 @pytest.mark.database
40 def test_product_insert():
41     db = Database()
42     db.insert_product(4, 'печиво', 'солодке', 30)
43     water_qnt = db.select_product_qnt_by_id(4)
44
45     assert water_qnt[0][0] == 30
46
```



Практика



```
→ Become QA Auto git:(master) X 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 18 items / 13 deselected / 5 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')]
....
===== 5 passed, 13 deselected in 0.05s =====
```



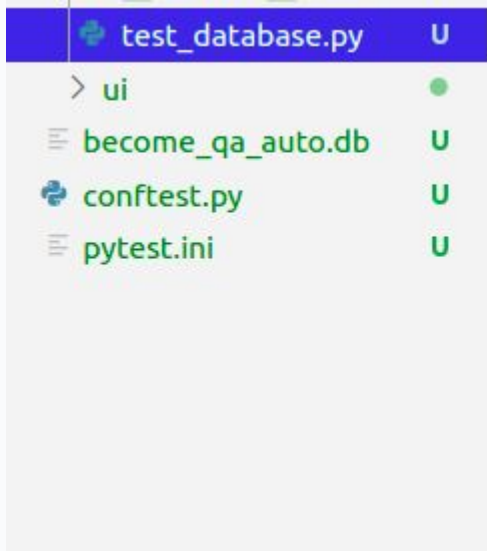
Практика



products	
id	int
name	text
description	text
quantity	int



Практика



```
36     assert water_qnt[0][0] == 25
37
38
39 @pytest.mark.database
40 def test_product_insert():
41     db = Database()
42     db.insert_product(4, 'печиво', 'солодке', 30)
43     water_qnt = db.select_product_qnt_by_id(4)
44
45     assert water_qnt[0][0] == 30
46
```




Практика



```
→ Become QA Auto git:(master) X 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 18 items / 13 deselected / 5 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')]
...F

===== FAILURES =====
_____ test_product_insert _____

@pytest.mark.database
def test_product_insert():
    db = Database()
    db.insert_product(4, 'печиво', 'солодке', 30)

tests/database/test_database.py:42:
```



Практика



```
test_product_insert

@pytest.mark.database
def test_product_insert():
    db = Database()
    > db.insert_product(4, 'печиво', 'солодке', 30)

tests/database/test_database.py:42:
-----
self = <modules.common.database.Database object at 0x7fe3db60ab20>
product_id = 4, name = 'печиво', description = 'солодке', qnt = 30

def insert_product(self, product_id, name, description, qnt):
    query = f"INSERT INTO products (id, name, description, quantity) \
        VALUES ({product_id}, '{name}', '{description}', {qnt})"
    > self.cursor.execute(query)
E     sqlite3.IntegrityError: UNIQUE constraint failed: products.id

modules/common/database.py:42: IntegrityError
===== short test summary info =====
FAILED tests/database/test_database.py::test_product_insert - sqlite3.Integri...
===== 1 failed, 4 passed, 13 deselected in 0.09s =====
```



INSERT. Синтаксис



```
INSERT OR REPLACE INTO products (id, name, description, quantity)  
VALUES (4, 'печиво', 'солодке', 30);
```



INSERT. Синтаксис



```
39     def insert_product(self, product_id, name, description, qnt):
40         query = f"INSERT OR REPLACE INTO products (id, name, description, quantity) \
41             VALUES ({product_id}, '{name}', '{description}', {qnt})"
42         self.cursor.execute(query)
43         self.connection.commit()
44     ..
```



INSERT. Синтаксис



```
→ Become QA Auto git:(master) X 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 18 items / 13 deselected / 5 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')]
....
===== 5 passed, 13 deselected in 0.05s =====
```



Підсумки



- ✓ Вивчили, що для створення даних використовується запит **INSERT INTO**
- ✓ Визначили, що ми не можемо вставляти одні і ті самі дані при наявності умови унікальності даних
- ✓ Ознайомилися з синтаксисом запиту **INSERT INTO**
- ✓ Написали ще один тест для нашої бази даних