Київський національний університет ім. Тараса Шевченка

Кафедра мережевих та інтернет технологій

Лабораторна робота № 9

Дисципліна: бази даних та інформаційні системи

Тема: Розширені можливості Redis

Виконав: Студент групи МІТ-31

Пугач Назар

Мета: Закріпити знання про роботу з Redis та ознайомитися з розширеним функціоналом — транзакціями, скриптами на Lua, публікацією/підпискою (Pub/Sub) та потоками Redis Streams.

Хід роботи

1. Транзакції у Redis
    a. Ознайомтесь з командами MULTI, EXEC, DISCARD, WATCH.
    b. Створіть просту транзакцію, що додає кілька ключів.

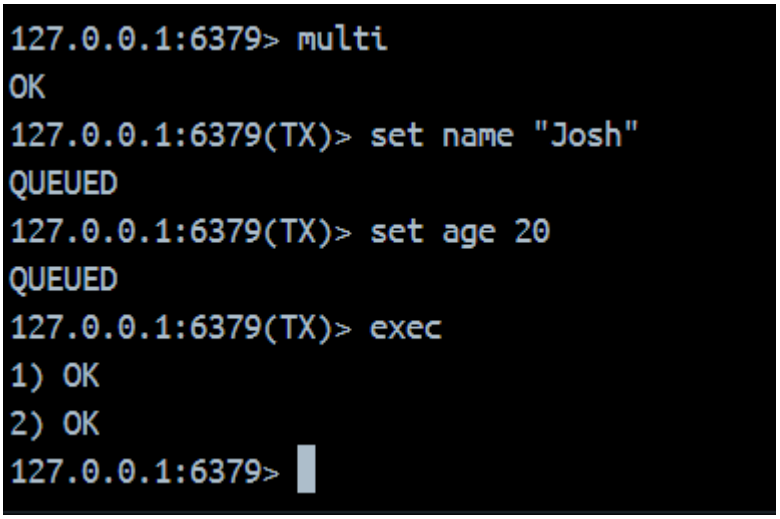

Рисунок 1 – проста транзакція.

    c. Спробуйте використати WATCH для імітації конкурентного доступу

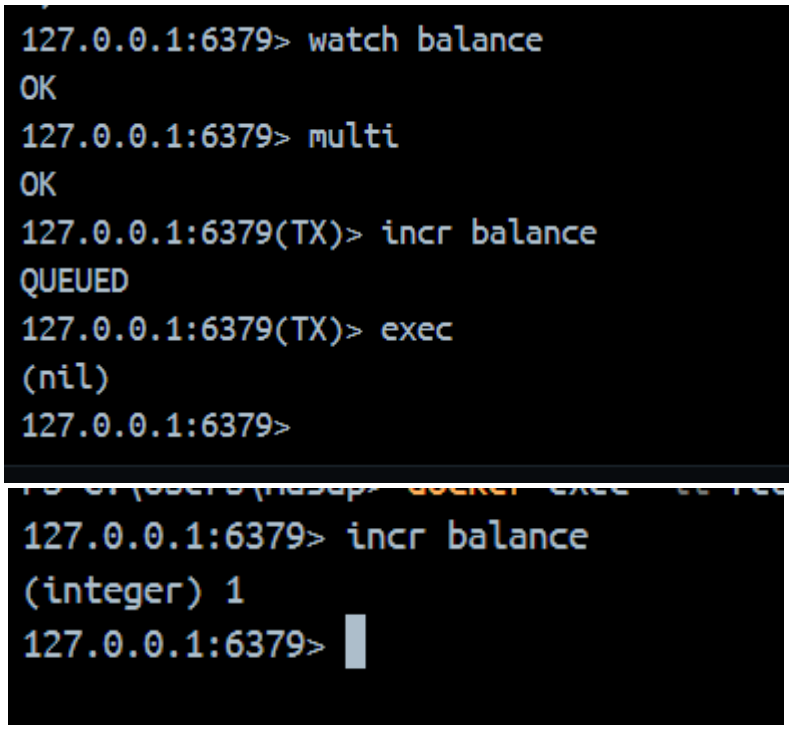

Рисунок 2 – імітація конкурентного доступу.

2. Lua-скрипти в Redis
    a. Ознайомтесь із командою EVAL.
    b. Напишіть простий Lua-скрипт, який перевіряє наявність ключа і створює його, якщо не існує.

```
EVAL "if redis.call('exists', KEYS[1]) == 0 then
  return redis.call('set', KEYS[1], ARGV[1])
else
  return 'exists'
```

end"
1 testkey "value"

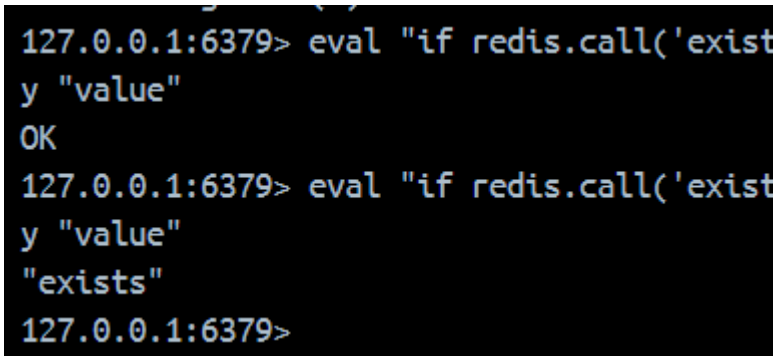

Рисунок 3 – результат роботи скрипта.

3. Механізм Pub/Sub
   a. В одному терміналі запустіть підписку на канал:

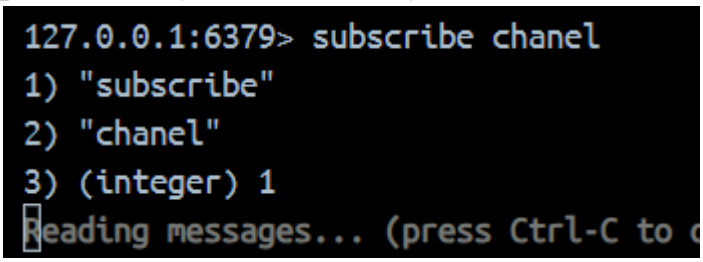

Рисунок 4 – підписка на канал.

   b. В іншому терміналі виконайте публікацію повідомлення:

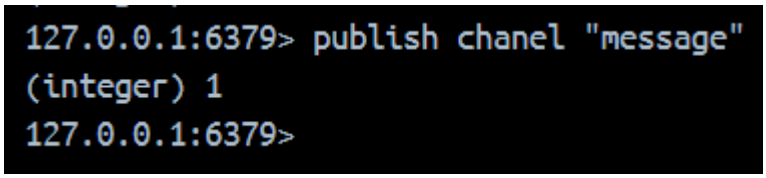

Рисунок 5 – публікація повідомлення.

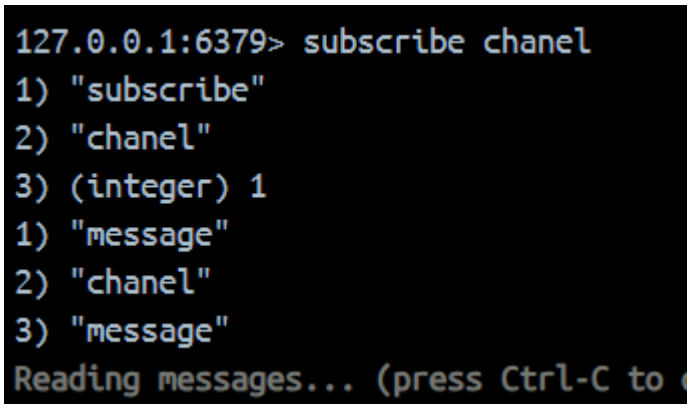

Рисунок 6 – отримане повідомлення в першому терміналі

4. Redis Streams.
   a. Додайте події до потоку
   b. Прочитайте останні події
   c. Створіть споживача потоку та зчитайте нові повідомлення

```
127.0.0.1:6379> XADD mystream * sensor-id 1234 temperature 19.8
"1745260502314-0"
127.0.0.1:6379> XRANGE mystream - +
1) 1) "1745260502314-0"
   2) 1) "sensor-id"
      2) "1234"
      3) "temperature"
      4) "19.8"
127.0.0.1:6379> XREAD COUNT 2 STREAMS mystream 0
1) 1) "mystream"
   2) 1) 1) "1745260502314-0"
         2) 1) "sensor-id"
            2) "1234"
            3) "temperature"
            4) "19.8"
127.0.0.1:6379>
```

Рисунок 7 – робота з потоками.

5. Напишіть скрипт (Python/JS), який реалізує логіку Pub/Sub або читає дані зі Stream.
    a. Python-код для роботи з Redis.

```python
import redis

r = redis.Redis(host='localhost', port=6379, db=0)


while True:
    msg = input("Enter message for 'chanel' (or 'exit'): ")
    if msg.lower() == 'exit':
        break
    r.publish('chanel', msg)
```

Рисунок 9 – Python код для роботи з Redis.

```
127.0.0.1:6379> subscribe chanel
1) "subscribe"
2) "chanel"
3) (integer) 1
1) "message"
2) "chanel"
3) "Hello from script!"
Reading messages... (press Ctrl-C to quit or any key to type command)
```

Рисунок 10 – Результат роботи коду.

Висновок: Під час виконання лабораторної роботи я ракріпив знання про роботу з Redis та ознайомився з розширеним функціоналом — транзакціями, скриптами на Lua, публікацією/підпискою (Pub/Sub) та потоками Redis Streams.