

## Лабораторна робота № 8 Бази даних та інформаційні системи

**Тема:** Робота з Redis

**Мета:** Закріпити розуміння роботи Redis та навчитися використовувати його основні можливості.

**Виконала:** студентка групи MIT-31, Панченко Владислава

Встановлення Redis на Windows за допомогою Docker і тестування підключення командою PING:

```
C:\Users\Lenovo>docker run --name redis-container -d -p 6379:6379 redis
Unable to find image 'redis:latest' locally
latest: Pulling from library/redis
254e724d7786: Pull complete
a131b527c2eb: Pull complete
e14bd717af36: Pull complete
1d380dfb74a4: Pull complete
c9e771e39c60: Pull complete
3781e1e8cbc4: Pull complete
4f4fb700ef54: Pull complete
f5340b1d1605: Pull complete
Digest: sha256:7df1eeff67eb0ba84f6b9d2940765a6bb1158081426745c185a03b1507de6a09
Status: Downloaded newer image for redis:latest
12e430d00fe4f2efafdb1c40e2c1d9587583158cc99a3fe7222177e530aab21e

C:\Users\Lenovo>docker exec -it redis-container redis-cli
127.0.0.1:6379> PING
PONG
127.0.0.1:6379>
```

Операції з рядками. Створюємо ключ зі своїм іменем командою SET, отримуємо його значення командою GET. Виконуємо команду INCR mycounter двічі та перевіряємо значення mycounter командою GET.

```
127.0.0.1:6379> SET student "Vladyslava Panchenko"
OK
127.0.0.1:6379> GET student
"Vladyslava Panchenko"
127.0.0.1:6379> INCR mycounter
(integer) 1
127.0.0.1:6379> INCR mycounter
(integer) 2
127.0.0.1:6379> GET mycounter
"2"
```

### Структури даних. Список List.

```
127.0.0.1:6379> LPUSH tasks "Task1"
(integer) 1
127.0.0.1:6379> LPUSH tasks "Task2"
(integer) 2
127.0.0.1:6379> LRANGE tasks 0 -1
1) "Task2"
2) "Task1"
127.0.0.1:6379> LPOP tasks
"Task2"
```

### Множина Set.

```
127.0.0.1:6379> SADD tech:set "Redis"
(integer) 1
127.0.0.1:6379> SADD tech:set "PostgreSQL"
(integer) 1
127.0.0.1:6379> SADD tech:set "MongoDB"
(integer) 1
127.0.0.1:6379> SMEMBERS tech:set
1) "Redis"
2) "PostgreSQL"
3) "MongoDB"
127.0.0.1:6379> SISMEMBER tech:set "Redis"
(integer) 1
```

### Хеш Hash.

```
127.0.0.1:6379> HSET profile name "Vladyslava Panchenko"
(integer) 1
127.0.0.1:6379> HSET profile city "Poltava"
(integer) 1
127.0.0.1:6379> HGET profile name
"Vladyslava Panchenko"
127.0.0.1:6379> HGETALL profile
1) "name"
2) "Vladyslava Panchenko"
3) "city"
4) "Poltava"
```

### Sorted Set.

```
127.0.0.1:6379> ZADD scores 85 "Student1"
(integer) 1
127.0.0.1:6379> ZADD scores 92 "Student2"
(integer) 1
127.0.0.1:6379> ZADD scores 74 "Student3"
(integer) 1
127.0.0.1:6379> ZREVRANGE scores 0 -1 WITHSCORES
1) "Student2"
2) "92"
3) "Student1"
4) "85"
5) "Student3"
6) "74"
127.0.0.1:6379> ZRANK scores "Student1"
(integer) 1
```

Працюємо з TTL. Після 10 секунд ключ зникає.

```
127.0.0.1:6379> SET temp:data "Hello" EX 10
OK
127.0.0.1:6379> GET temp:data
"Hello"
127.0.0.1:6379> TTL temp:data
(integer) 1
127.0.0.1:6379> GET temp:data
(nil)
```

Міні програма на Python. Програма повинна збільшувати лічильник, зчитувати список задач та публікувати повідомлення.

```
import redis

r = redis.Redis(host='localhost', port=6379, decode_responses=True)

#Лічильник
r.incr('mycounter')
print('Лічильник:', r.get('mycounter'))

#Список задач
r.lpush('tasks', 'Task1')
r.lpush('tasks', 'Task2')
print('Задачі:', r.lrange('tasks', 0, -1))

#Публікація повідомлення
r.publish('channel', 'Hello from Python!')
```

Результат запуску програми:

```
Лічильник: 3
Задачі: ['Task2', 'Task1', 'Task1']
```

```
127.0.0.1:6379> SUBSCRIBE channel
1) "subscribe"
2) "channel"
3) (integer) 1
1) "message"
2) "channel"
3) "Hello from Python!"
```

Де можна використати Redis у реальних проєктах:

- Кешування (щоб пришвидшити доступ до часто використовуваних даних)
- Сесії користувачів (у веб застосунках)
- Черги завдань (наприклад, Celery з Redis)
- Pub/Sub системи (для сповіщень, чату тощо)
- Leaderboard-и в іграх (Sorted Set)
- Мікросервіси (як посередник між ними)

### Запитання для самоперевірки

#### 1. Як встановити Redis у Windows/Linux?

У Linux командами:

`sudo apt update` — оновлення списку пакетів  
`sudo apt install redis-server -y` — встановлення Redis  
`sudo systemctl status redis` — перевірка статусу служби  
`redis-cli` — запуск Redis

У Windows за допомогою WSL:

`wsl --install` — встановлення WSL, якщо немає

Далі ті ж команди, що й для Linux.

У Windows за допомогою Docker:

`docker run --name redis-container -d -p 6379:6379 redis`  
`docker exec -it redis-container redis-cli`

#### 2. Які основні команди для роботи з рядками в Redis?

`SET key value` — встановити значення  
`GET key` — отримати значення  
`INCR key` — збільшити значення (лічильник)  
`EXISTS key` — перевірити існування  
`DEL key` — видалити ключ

#### 3. Як працювати з списками, множинами, хешами та Sorted Set?

Список (List):

`LPUSH list value` — додати на початок  
`LRANGE list 0 -1` — вивести всі елементи  
`LPOP list` — видалити перший елемент

Множина (Set):

`SADD set value` — додати значення  
`SMEMBERS set` — отримати всі елементи

SISMEMBER set value — перевірити наявність

Хеш (Hash):

HSET hash field value — встановити поле

HGET hash field — отримати значення поля

HGETALL hash — отримати всі пари поле–значення

Відсортована множина (Sorted Set):

ZADD zset score member — додати елемент з рейтингом

ZREVRANGE zset 0 -1 WITHSCORES — вивести у зворотному порядку

ZRANK zset member — отримати позицію елемента

#### 4. Як встановити час життя ключа в Redis?

SET key value EX seconds — створити ключ із TTL

TTL key — дізнатися, скільки секунд залишилося до видалення

#### 5. Які можливі сценарії використання Redis у реальних застосунках?

- Кешування (щоб пришвидшити доступ до часто використовуваних даних)
- Сесії користувачів (у веб застосунках)
- Черги завдань (наприклад, Celery з Redis)
- Pub/Sub системи (для сповіщень, чату тощо)
- Leaderboard-и в іграх (Sorted Set)
- Мікросервіси (як посередник між ними)

**Висновок:** ознайомилися з основними можливостями Redis, навчилися працювати з основними типами даних Redis, було протестовано механізм публікації/підписки (Pub/Sub) та створено просту програму для взаємодії з Redis у Python.

**Додаток:** файл lab8.py до завдання з міні-програмою в Python.