

Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 2
з дисципліни «Бази даних. Частина 2»
«Практика використання сервера Redis »

Виконав
студент групи КП-82
Грицюк Олександр

Київ - 2021

1. Екранні форми

- режим звичайного користувача:

```

      ГОЛОВНЕ МЕНЮ

      1 - Зареєструватися
      2 - Увійти
      3 - Exit

```

```

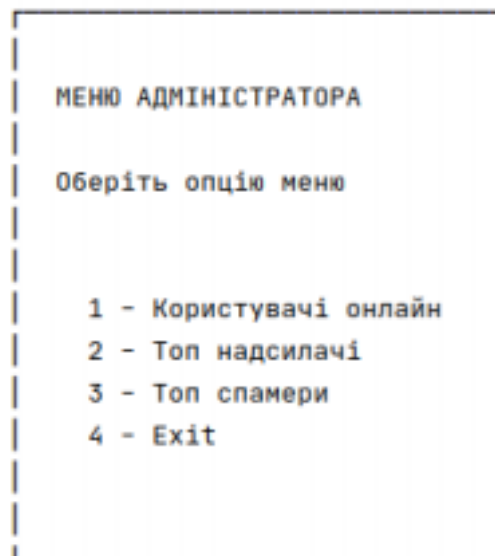
      КОРИСТВАЦЬКЕ МЕНЮ ДЛЯ <new_user>

      Оберіть опцію меню

      1 - Вийти
      2 - Надіслати повідомлення
      3 - Вхідні повідомлення
      4 - Статистика моїх повідомлень
      5 - Exit

```

- режим адміністратора



- режим емуляції

```
Введіть кількість користувачів для емуляції: 10
Користувач: wwilliamson
Користувач: mfisher
Користувач: batesdaniel
Користувач: norrisallison
Користувач: samuel76
Користувач: suttoncarrie
Користувач: elizabeth27
Користувач: jonathon08
Користувач: richardmurphy
Користувач: morristhomas
Натисніть ENTER для початку...█
```

```
користувач <richardmurphy`> надіслав повідомлення до <samuel76>
<Law fish above generation capital serve.>
```

```
користувач <morristhomas`> надіслав повідомлення до <elizabeth27>
<Each move east tax plant investment guy may ability.>
```

```
користувач <jonathon08`> надіслав повідомлення до <morristhomas>
<Poor attack arm financial meet ago add class whose.>
```

```
користувач <norrisallison`> надіслав повідомлення до <suttoncarrie>
<Action recently throw down food agency two they kitchen debate religious far.>
```

```
користувач <samuel76`> надіслав повідомлення до <morristhomas>
<Trouble car side pull note sure add hard young describe thus pick production.>
```

2. Обґрунтування вибору структур даних

Для представлення користувачів в системі було обрано структуру даних SET, адже вона дозволяє за час $O(1)$ здійснити пошук, чи є такий користувач, наприклад під час реєстрації чи аутентифікації. Так як маємо дві групи користувачів, які мають різні функціональні можливості, було вирішено створити два окремих SETа.

Так як адмін повинен мати можливість переглядати список наявних користувачів, а також їх порядок не важливий, то з цією метою також було обрано структуру даних SET.

Для того, щоби мати можливість спостерігати відсортований рейтинг користувачів відповідно до їх активності (надісланих повідомлень/спаму), було обрано структуру даних відсортований сет, тобто ZSET, адже саме він зберігає ключі у відсортованому вигляді за певним значенням.

Для зберігання даних про повідомлення, було вирішено для кожного екземпляра створювати HASH (бо він дозволяє запам'ятовувати набори ключ значення) і в ньому зберігати інформацію про юзернейми відправника та отримувача, а також саме повідомлення. Для того, щоб генерувати для кожної такої структури унікальний ключ, було використано звичайну строку, яку кожен раз при додаванні нового повідомлення збільшуємо на одиницю.

Для журналу повідомлень було використано Pub/Sub, адже завдяки цьому будь-хто може стати publisher-ом і відправляти повідомлення, які слухач відразу ж прочитає, адже він підписаний на відповідний канал. Це дає змогу створювати додатки, які в реалтаймі будуть оновлюватися, адже як тільки-но дані будуть опубліковані, клієнт зможе їх миттєво прочитати та відповідно опрацювати їх.

Для черги повідомлень було використано LIST, адже ця структура даних зберігає порядок, а операції вставляння в кінець та вилучення з початку займають $O(1)$ часу. Для відображення користувачу вхідних повідомлень (які надійшли саме йому) було обрано LIST, адже він дозволяє зберігати хронологічний порядок появи нових повідомлень. Для того, щоб користувач зміг

отримати статистику по статусам відправлених повідомлень, було відповідно створено SET для кожного статусу, в яких зберігаються усі повідомлення в системі відповідно до їх статусу. Також щоб зберігати відправлені користувачем повідомлення був також створений SET.

Щоби групувати, використовуються операції об'єднання множин статусів, які були обрані для того щоб їх переглянути, а потім робиться перетин утвореної множини з множиною відправлених повідомлень.

Це і дає змогу швидко порахувати відповідну кількість.