

# Проектирование NoSQL - основы Redis

## Операции

Redis, как система управления базами данных, построенная на структуре ключ-значение, оперирует данными, используя простые, но мощные команды для манипуляции с парами ключ-значение. Основное взаимодействие с базой данных включает в себя две базовые операции: установку значения по ключу (SET) и получение значения по ключу (GET).

1. Установка значения (SET): Команда `SET` используется для сохранения значения по ключу. Если ключ уже существует, его значение будет перезаписано. Простейший пример команды SET:

```
SET mykey "Hello"
```

Эта команда устанавливает значение "Hello" для ключа "mykey". Если операция прошла успешно, Redis вернет ответ "OK".

2. Получение значения (GET): Для получения значения, которое было сохранено по ключу, используется команда `GET` :

```
GET mykey
```

Эта команда вернет значение, ассоциированное с "mykey". Если ключ не существует, Redis вернет специальное значение `nil` .

Кроме этих базовых команд, Redis предоставляет широкий спектр дополнительных операций для работы с ключами и значениями:

- EXPIRE/PERSIST: Для установки времени жизни ключа (после истечения которого ключ будет автоматически удален) используется команда `EXPIRE` . Команда `PERSIST` удаляет время жизни, делая ключ постоянным.
- DEL: Удаляет один или несколько ключей и их значения.
- MSET/MGET: Позволяет установить или получить значения для нескольких ключей за одну операцию.
- INCR/DECR: Увеличивает или уменьшает числовое значение строкового ключа на единицу. Если ключ не содержит числовое значение, Redis вернет ошибку.

Redis не ограничивается хранением только строковых значений. Как обсуждалось ранее, он может управлять разнообразными типами данных, и для каждого из этих типов существуют специфические команды, позволяющие эффективно их использовать. Все эти операции выполняются очень быстро, так как данные хранятся в оперативной памяти. Это делает Redis идеальным решением для задач, где требуется высокая скорость доступа к данным, таких как кэширование, сессии, очереди сообщений и многое другое.

## Практика

Онлайн сервис <https://try.redis.io/> может заменить нам локальный Redis. Зайдите туда и начните практику.

Создание 5 ключей, назначение значений и TTL

1. Создадим 5 ключей и назначим им значения в Redis. Давайте предположим, что наши ключи и значения будут следующими:

- Ключ `user:1` с значением `John`
- Ключ `user:2` с значением `Alice`
- Ключ `product:101` с значением `Laptop`
- Ключ `product:102` с значением `Smartphone`
- Ключ `order:123` с значением `Order123`

Для этого выполним по очереди следующие команды Redis:

```
SET user:1 John
SET user:2 Alice
SET product:101 Laptop
SET product:102 Smartphone
SET order:123 Order123
```

2. Теперь мы назначим время жизни (TTL) для каждого из ключей. Предположим, что мы хотим установить TTL 1 час (3600 секунд) для каждого ключа, по очереди выполнив команды:

```
EXPIRE user:1 3600
```

```
EXPIRE user:2 3600
```

```
EXPIRE product:101 3600
```

```
EXPIRE product:102 3600
```

```
EXPIRE order:123 3600
```

Просмотр списка всех ключей. Для просмотра списка всех ключей в Redis, выполним следующую команду:

```
KEYS *
```

Эта команда вернет список всех ключей, которые у нас есть в Redis.

Просмотр списка всех ключей, значений и TTL. Чтобы просмотреть список всех ключей, их значений и TTL, мы можем использовать следующую команду:

```
SCAN 0 COUNT 100
```

Эта команда вернет нам список ключей и их значений, и мы сможем увидеть, сколько времени осталось до истечения TTL для каждого ключа.