**Set**

Команда используется для установки ключа и его значения, с дополнительными необязательными параметрами для указания срока действия записи значения ключа. Давайте установим ключ foo со значением hello world:

SET foo "hello world"

**Get**

Команда используется для получения значения, связанного с ключом. Если запись значения ключа превысила срок действия, будет возвращено nil:

GET foo

"hello world"

Если истечет время жизни записи:

GET foo

(nil)

**EXISTS**

Эта команда проверяет, существует ли что то с данным ключом. Она возвращает 1 если объект существует или 0 если нет:

127.0.0.1:6379> EXISTS foo

(integer) 1

**FLUSHALL**

Эта команда полностью удаляет все данные в текущем сеансе:

127.0.0.1:6379> FLUSHALL

**GETSET**

Команда возвращает текущее значение и устанавливает новое. Используется для атомарного управления данными:

127.0.0.1:6379> SET foo "hello"

OK

127.0.0.1:6379> GETSET foo "world"

"hello"

127.0.0.1:6379> GET foo

"world"

#### DEL

Команда удаляет ключ и соответствующее значение:

127.0.0.1:6379> DEL foo

(integer) 1

#### APPEND

Команда добавляем в соотвествующий ключ дополнительное значение. Возвращает количество символов итогового значения:

127.0.0.1:6379> SET foo "hello"

OK

127.0.0.1:6379> APPEND foo " world"

(integer) 11

127.0.0.1:6379> GET foo

"hello world"

#### KEYS

Возвращает все ключи из базы по указанному шаблону:

127.0.0.1:6379> KEYS '\*'

1) "boo"

2) "foo"

Может случиться так, что **KEYS** в ответ показывает:

(empty list or set)

Если абсолютно уверены, что сервер Redis который используете, тот на котором у есть данные, возможно подключаетесь к неправильному экземпляру базы данных. Как упоминается в документации Redis, новые подключения по умолчанию подключаются к **db 0**, а ключ находится в **db 1**. Чтобы выбрать нужную базу данных, используйте **SELECT**:

127.0.0.1:6379> SELECT 1

127.0.0.1:6379> KEYS '\*'

#### INCR / DECR

Инкремент / декримент. Значение ключа **integer**, хотя в базе храниться все равно строка, можно увеличить или уменьшить на **1**. Если использовать команду **INCR** с несуществующем значением то создаться новый ключ со значением 1:

127.0.0.1:6379> SET foo 1

OK

127.0.0.1:6379> INCR foo

(integer) 2

127.0.0.1:6379> INCR foo

(integer) 3

127.0.0.1:6379> DECR foo

(integer) 2

#### TTL

Когда ключ установлен с истечением срока действия, например **SET foo EX 10**, эту команду можно использовать для просмотра оставшегося времени:

127.0.0.1:6379> SET foo 10 EX 10

OK

127.0.0.1:6379> TTL foo

(integer) 6

127.0.0.1:6379> TTL foo

(integer) 5

127.0.0.1:6379> TTL foo

(integer) 2

127.0.0.1:6379> TTL foo

(integer) -2

#### PERSIST

Если мы передумаем об истечении срока действия ключа, мы можем использовать эту команду, чтобы удалить период истечения срока действия:

127.0.0.1:6379> PERSIST foo

(integer) 1

127.0.0.1:6379> TTL foo

(integer) -1

127.0.0.1:6379> GET foo

"bar"

#### RENAME

Эта команда используется для переименования ключей на нашем сервере Redis:

127.0.0.1:6379> RENAME foo foo2

OK

127.0.0.1:6379> GET foo

(nil)

127.0.0.1:6379> GET foo2

"bar"