

# Лабораторная работа №8: Знакомство с NoSQL БД (на примере Redis)

---

1. Цель работы Освоить основы работы с NoSQL базой данных Redis. Получить практические навыки выполнения основных операций CRUD, работы с различными типами данных и сравнения производительности с реляционной СУБД.
2. Используемый стек технологий

База данных:\*\* Redis

Язык программирования: Python 3

Библиотека: redis-py

Реляционная СУБД: SQLite (для сравнения)

ОС: Linux (Ubuntu)

Инструменты: Redis CLI, Python IDE

3. Теоретические сведения

**Redis** — высокопроизводительная NoSQL база данных типа «ключ-значение» с возможностью хранения различных структур данных.

## Ключевые особенности Redis:

- **In-memory data storage:** Данные хранятся в оперативной памяти
- **Поддержка структур данных:** строки, списки, множества, хеши, сортированные множества
- **Персистентность:** Возможность сохранения данных на диск
- **Репликация:** Поддержка мастер-слейв репликации
- **Высокая производительность:** До сотен тысяч операций в секунду

## Сравнение с реляционными БД:

- **Гибкость схемы данных:** Отсутствие жесткой схемы
- **Производительность:** Значительное преимущество для определенных сценариев
- **Масштабируемость:** Простое горизонтальное масштабирование

4. Ход выполнения работы

## Часть 1: Установка и настройка Redis

---

## 1.1 Установка Redis на Ubuntu/Debian

```
echo " НАЧАЛО УСТАНОВКИ REDIS" echo ""*50

echo " Обновление списка пакетов..." sudo apt update

echo " Установка Redis..." sudo apt install -y redis-server

echo " Redis установлен!"
```

## 1.2 Настройка Redis

```
echo "\n Проверка конфигурационного файла Redis..." sudo ls -la /etc/redis/ echo ""

echo " Содержимое конфигурационного файла redis.conf (первые 30 строк):" sudo head -30
/etc/redis/redis.conf

echo "\n Настройка безопасности Redis..." echo "Рекомендуемые изменения в /etc/redis/redis.conf:"
echo "1. bind 127.0.0.1 ::1 (разрешить только локальные подключения)" echo "2. requirepass
ваш_пароль (установить пароль, опционально для тестов)" echo "3. protected-mode yes (включить
защищенный режим)" echo ""

echo " Проверка текущих настроек безопасности:" sudo grep -E "^(bind|requirepass|protected-
mode)" /etc/redis/redis.conf
```

## 1.3 Запуск и проверка работы Redis

```
echo "\n Запуск службы Redis..." sudo systemctl start redis-server

echo " Включение автозагрузки Redis..." sudo systemctl enable redis-server

echo " Проверка статуса службы Redis..." sudo systemctl status redis-server --no-pager

echo "\n Проверка работы Redis через redis-cli..." redis-cli ping
```

## 1.4 Дополнительная проверка работоспособности

```
echo "\n Информация о сервере Redis:" redis-cli INFO SERVER | grep -E "
(redis_version|process_id|tcp_port|uptime)"

echo "\n Базовая статистика Redis:" redis-cli INFO STATS | grep -E "
(total_connections_received|total_commands_processed|instantaneous_ops_per_sec)"

echo "\n Использование памяти:" redis-cli INFO MEMORY | grep -E "
(used_memory_human|used_memory_peak_human|mem_fragmentation_ratio)"

echo "\n Быстрый тест производительности (PING 100 раз):" time redis-cli -r 100 PING > /dev/null
echo "Тест завершен!"
```

Часть 2: Базовые операции CRUD с помощью redis-cli 2.1 Вход в интерактивный режим redis-cli

```
echo "\n ЗАПУСК ИНТЕРАКТИВНОГО РЕЖИМА REDIS-CLI" echo "="*50 echo "Введите следующие команды в redis-cli:" echo ""
```

## 2.2 Работа со строками (Strings)

```
cat > redis_strings_demo.txt << 'EOF' echo "\n РАБОТА СО СТРОКАМИ (STRINGS)" echo "="*40
```

```
SET user:1000 "John Doe" SET user:1001 "Jane Smith" SET user:1002 "Bob Johnson" SET user:1003 "Alice Williams"
```

```
GET user:1000 GET user:1001
```

```
EXISTS user:1000 EXISTS user:9999
```

```
SET session:abc123 "user_data_here" EX 60 TTL session:abc123
```

```
SET counter:visits 0 INCR counter:visits INCR counter:visits INCRBY counter:visits 5 GET counter:visits  
DECR counter:visits
```

```
APPEND user:1000 " - Manager" GET user:1000
```

```
STRLEN user:1000
```

```
GETRANGE user:1000 0 3
```

```
SETNX user:1004 "New User" SETNX user:1000 "Should Not Change" GET user:1004 GET user:1000
```

```
MGET user:1000 user:1001 user:1002 user:1003
```

```
DEL user:1003 EXISTS user:1003
```

```
echo "\n Список всех ключей с префиксом user:" KEYS user:* EOF
```

```
echo "Выполнение операций со строками..." redis-cli < redis_strings_demo.txt
```

## 2.3 Работа с хешами (Hashes)

```
cat > redis_hashes_demo.txt << 'EOF' echo "\n РАБОТА С ХЕШАМИ (HASHES)" echo "="*40
```

```
HSET user:profile:1000 name "John" age 30 email "john@example.com" city "New York" job "Developer"
```

```
HGETALL user:profile:1000
```

```
HGET user:profile:1000 name HGET user:profile:1000 age HMGET user:profile:1000 name email job
```

```
HEXISTS user:profile:1000 name HEXISTS user:profile:1000 phone
```

```
HKEYS user:profile:1000 HVALS user:profile:1000
```

```
HLEN user:profile:1000
```

HSET user:profile:1000 age 31 HGET user:profile:1000 age

HINCRBY user:profile:1000 age 1 HGET user:profile:1000 age

HSETNX user:profile:1000 phone "123-456-7890" HSETNX user:profile:1000 name "Should Not Change" HGETALL user:profile:1000

HSET user:profile:1001 name "Jane" age 25 email "jane@example.com" city "London" job "Designer"

HSET user:profile:1002 name "Bob" age 35 email "bob@example.com" city "Tokyo" job "Manager"

HDEL user:profile:1000 phone HGETALL user:profile:1000

echo "\n Список всех хешей пользователей:" KEYS user:profile:\*

EOF

echo "\nВыполнение операций с хешами..." redis-cli < redis\_hashes\_demo.txt

## 2.4 Работа со списками (Lists)

cat > redis\_lists\_demo.txt << 'EOF' echo "\n РАБОТА СО СПИСКАМИ (LISTS)" echo "=="\*40"

LPUSH tasks:user:1000 "Write report" LPUSH tasks:user:1000 "Check email" LPUSH tasks:user:1000 "Attend meeting"

LRANGE tasks:user:1000 0 -1

RPUSH tasks:user:1000 "Send invoices" LRANGE tasks:user:1000 0 -1

LLEN tasks:user:1000

LINDEX tasks:user:1000 0 LINDEX tasks:user:1000 -1

LSET tasks:user:1000 1 "Check urgent email" LRANGE tasks:user:1000 0 -1

LPOP tasks:user:1000 LRANGE tasks:user:1000 0 -1

RPOP tasks:user:1000 LRANGE tasks:user:1000 0 -1

LPUSH tasks:user:1000 "Task A" LPUSH tasks:user:1000 "Task B" LPUSH tasks:user:1000 "Task C"

LRANGE tasks:user:1000 0 -1 LTRIM tasks:user:1000 0 2 LRANGE tasks:user:1000 0 -1

LPUSH messages:queue "Message 1" LPUSH messages:queue "Message 2" LPUSH messages:queue "Message 3" BRPOP messages:queue 5

EOF

echo "\nВыполнение операций со списками..." redis-cli < redis\_lists\_demo.txt

## 2.5 Работа с множествами (Sets)

cat > redis\_sets\_demo.txt << 'EOF' echo "\n РАБОТА С МНОЖЕСТВАМИ (SETS)" echo "=="\*40"

SADD tags:article:1000 "python" "programming" "tutorial" "redis" SADD tags:article:1001 "redis" "database" "nosql" "tutorial" SADD tags:article:1002 "python" "machine-learning" "ai"

SMEMBERS tags:article:1000

SISMEMBER tags:article:1000 "python" SISMEMBER tags:article:1000 "java"

SCARD tags:article:1000

SREM tags:article:1000 "tutorial" SMEMBERS tags:article:1000

SMOVE tags:article:1001 tags:article:1002 "database" SMEMBERS tags:article:1001 SMEMBERS tags:article:1002

echo "\nОбъединение (UNION):" SUNION tags:article:1000 tags:article:1001 tags:article:1002

echo "\nПересечение (INTERSECTION):" SINTER tags:article:1000 tags:article:1001

echo "\nРазность (DIFFERENCE):" SDIFF tags:article:1000 tags:article:1001

SRANDMEMBER tags:article:1000 SRANDMEMBER tags:article:1000 2 # Два случайных элемента  
EOF

echo "\nВыполнение операций с множествами..." redis-cli < redis\_sets\_demo.txt

## 2.6 Работа с упорядоченными множествами (Sorted Sets)

cat > redis\_sorted\_sets\_demo.txt << 'EOF' echo "\n РАБОТА С УПОРЯДОЧЕННЫМИ  
МНОЖЕСТВАМИ (SORTED SETS)" echo "=="\*40"

ZADD leaderboard 1500 "player:1000" ZADD leaderboard 1800 "player:1001" ZADD leaderboard 1200 "player:1002" ZADD leaderboard 2100 "player:1003" ZADD leaderboard 1700 "player:1004"

ZRANGE leaderboard 0 -1 WITHSCORES

ZREVRANGE leaderboard 0 -1 WITHSCORES

ZRANK leaderboard "player:1001" # Ранг по возрастанию ZREVRANK leaderboard "player:1001" #  
Ранг по убыванию

ZSCORE leaderboard "player:1001"

ZINCRBY leaderboard 100 "player:1000" ZSCORE leaderboard "player:1000"

ZRANGEBYSCORE leaderboard 1500 2000 WITHSCORES

ZCOUNT leaderboard 1500 2000

ZREVRANGE leaderboard 0 2 WITHSCORES

ZREM leaderboard "player:1002" ZRANGE leaderboard 0 -1 WITHSCORES

```
ZADD article:views 150 "article:1000" ZADD article:views 89 "article:1001" ZADD article:views 324  
"article:1002" ZADD article:views 45 "article:1003"
```

```
ZINCRBY article:views 1 "article:1000" # Увеличение просмотров ZREVRANGE article:views 0 -1  
WITHSCORES # Самые популярные статьи
```

```
EOF
```

```
echo "\nВыполнение операций с упорядоченными множествами..." redis-cli <  
redis_sorted_sets_demo.txt
```