**Методы анализа данных**

**Практическая работа №6**

**Работа с Apache HBase**

**Доступ к данным с помощью HBase**

В этой лабораторной работе выполняются базовые операции с созданием, удалением и изменением таблицы в оболочке HBase. Вы будете использовать оболочку для размещения и получения данных в HBase.

## Начнем с оболочки HBase

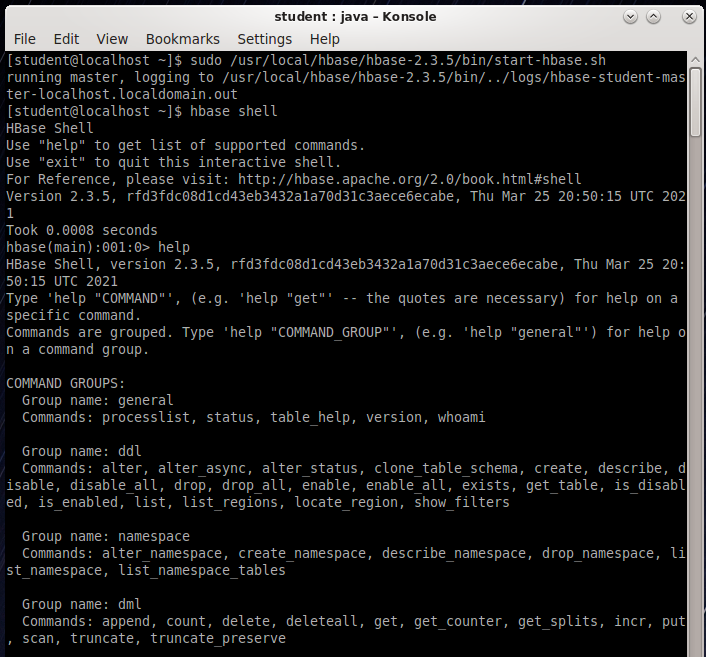
### Запустите оболочку HBase. Выполните команду справки и просмотрите основную информацию об использовании оболочки HBase.

sudo /usr/local/hbase/hbase-2.3.5/bin/start-hbase.sh

hbase shell

﻿hbase(main):001:0> help

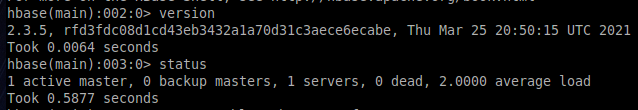
Примечание: Команды HBase пишутся после символа-приглашения ">".



### Отображение версии и состояния для базового использования

﻿hbase(main):001:0> version

hbase(main):001:0> status



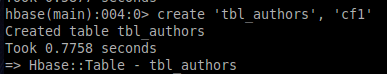
Вы можете увидеть версию Hbase (2.3.5) и автономное выполнение (1 активный мастер, 1 сервер).

### Используйте команду create для создания новой таблицы. Необходимо указать имя таблицы и имя семейства столбцов.

#### Имя таблицы: tbl\_authors, Семейство столбцов (Column Family): cf1

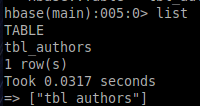
hbase(main):001:0> create ‘tbl\_authors’, ‘cf1’

Примечание: Имена таблиц, строки, столбцы должны быть заключены в кавычки.



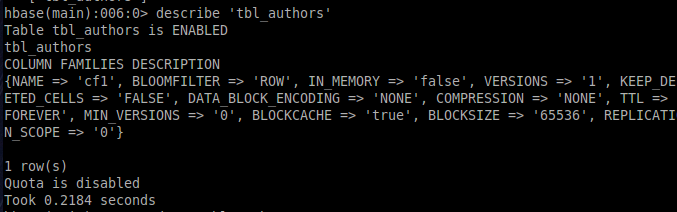
### Посмотреть список созданных таблиц можно с помощью команды list:

hbase(main):002:0> list



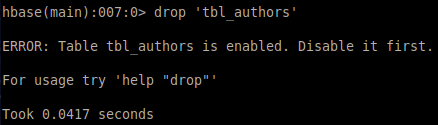
### Используйте команду describe для просмотра подробных сведений, включая настройки по умолчанию.

hbase(main):001:0> describe ‘tbl\_authors’



### Удалите только что созданную таблицу. Выведите все таблицы, чтобы убедиться, что таблица была успешно удалена

hbase(main):001:0> drop ‘tbl\_authors’



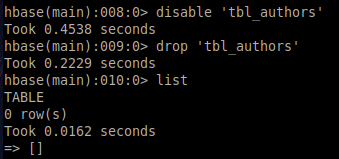
Примечание: Получислоь? Дело в том, что таблица tbl\_authors включена. Сначала отключите её, чтобы затем удалить её.

### Отключение таблицы tbl\_authors для её удаления.

hbase(main):003:0> disable ‘tbl\_authors’

hbase(main):004:0> drop ‘tbl\_authors’

hbase(main):005:0> list



### Создайте тестовую таблицу и используйте команду put. Здесь мы вставляем три значения, по одному за раз.

#### Входные данные представлены в следующем формате.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Column family | | |
| Rowkey | cf1:a | cf1:b | cf1:c |
| rk1 | A |  |  |
| rk2 |  | B |  |
| rk3 |  |  | C |

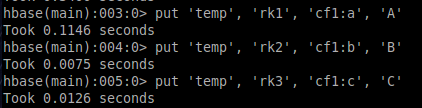
#### Имя таблицы: temp, Семейство столбцов (Column Family): cf1

hbase(main):003:0> ﻿create 'temp', 'cf1'

hbase(main):004:0> ﻿put 'temp', 'rk1', 'cf1:a', 'A'

hbase(main):005:0> ﻿put 'temp', 'rk2', 'cf1:b', 'B'

hbase(main):006:0> ﻿put 'temp', 'rk3', 'cf1:c', 'C'



### Подсчет количества строк во временной таблице HBase, созданной в предыдущем шаге

hbase(main):003:0> ﻿count ‘temp’

### 

### Добавим еще значений. Каким будет общее количество строк в таблице temp после выполнения следующих команд?

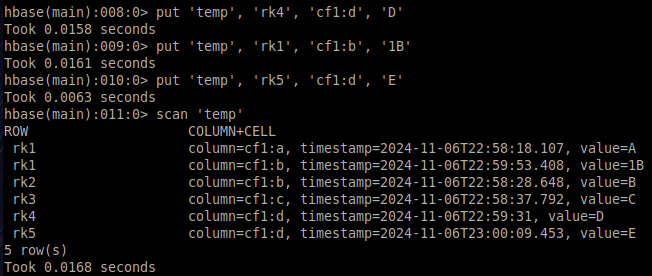
hbase(main):003:0> ﻿ ﻿put 'temp', 'rk4', 'cf1:d', 'D'

hbase(main):003:0> ﻿ ﻿put 'temp', 'rk1', 'cf1:b', '1B'

hbase(main):003:0> ﻿ ﻿put 'temp', 'rk5', 'cf1:d', 'E'

hbase(main):003:0> ﻿scan ‘temp’

Ниже приведен результат сканирования данных в таблице TEMP. Общий список — 6, но фактическое количество строк — 5.

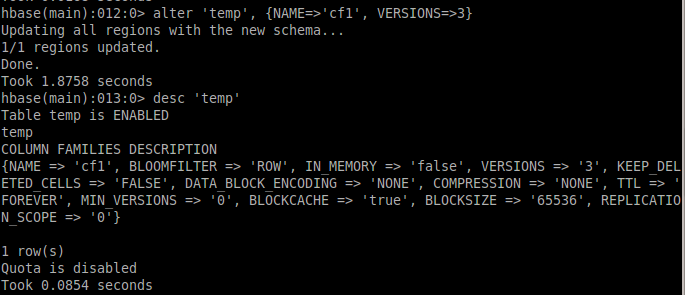


Примечание: В cf1 есть две строчки с ROWKEY rk=1.

### Измените атрибут version cf1 в таблице temp на 3 и проверьте, правильно ли отражено изменение.

hbase(main):003:0> ﻿ ﻿alter ‘temp’, {NAME=>'cf1', VERSIONS=>3}

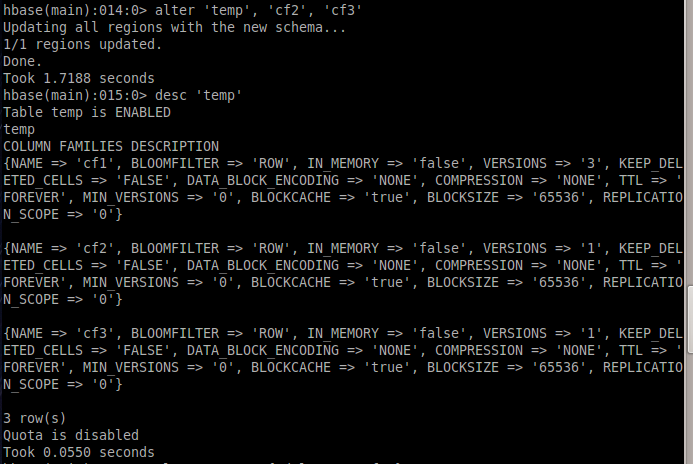
hbase(main):004:0> desc ‘temp’



### Измените временную таблицу, добавив семейство столбцов cf2 и cf3 с помощью команды ALTER.

hbase(main):003:0> ﻿ ﻿﻿alter 'temp', 'cf2', 'cf3'

### Выполните команду desc, чтобы проверить, добавлены ли cf2 и cf3.



### Удалите только что добавленный cf3 и проверьте результат.

hbase(main):003:0>﻿ alter 'temp', {'delete' => 'cf3'}

hbase(main):004:0> desc ‘temp’

### Удалите временную таблицу. Выведите все таблицы, чтобы убедиться, что таблица была успешно удалена.

hbase(main):005:0> disable ‘temp’

hbase(main):006:0> drop ‘temp’

hbase(main):007:0> list

### Выйдите из оболочки HBase с помощью команды quit

hbase(main):004:0> quit

### Создайте файл с командами HBase (при копировании могут возникнуть проблемы с кавычками).

$vi rubyscript.rb

﻿create 'temp', 'cf1'

put 'temp', 'rk1', 'cf1:a', 'A'

put 'temp', 'rk2', 'cf1:b', 'B'

put 'temp', 'rk3', 'cf1:c', 'C'

put 'temp', 'rk4', 'cf1:d', 'D'

alter 'temp', {NAME=>'cf1', VERSIONS=>3}

alter 'temp', 'cf2', 'cf3'

put 'temp', 'rk1', 'cf1:b', '1B'

put 'temp', 'rk5', 'cf1:d', 'E'

scan 'temp'

get 'temp', 'rk1'

count 'temp'

exit

### Теперь этот файл можно выполнить из оболочки HBase.

hbase shell

hbase(main):001:0> require ‘./rubyscript.rb’

텍스트이(가) 표시된 사진

자동 생성된 설명

Примечание: выше приведен экран результатов выполнения rubyscript.rb. Если в скрипте нет оператора exit, HBase остается в состоянии оболочки.

### Можно также передавать команды в оболочку HBase в неинтерактивном режиме с помощью команды echo и оператора | (pipe).

echo "describe 'temp'" | hbase shell -n

**Доступ к данным с помощью команд DML**

В этой лабораторной работе вы будете использовать команды для вставки, извлечения, сканирования и удаления строк.

## Операции CRUD (Вставка, выбор, обновление, удаление).

### Запустите оболочку HBase:

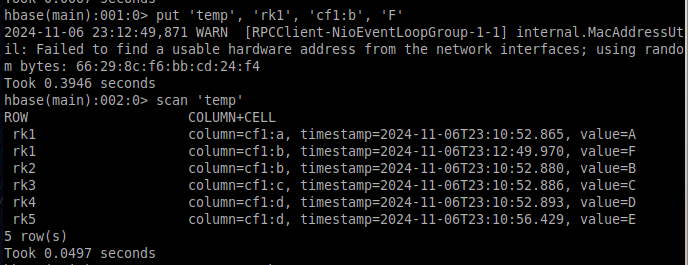
﻿hbase shell

### Введите данные со следующими условиями и отобразите результат.

RowKey:rk1, column descriptor : b и значение: F

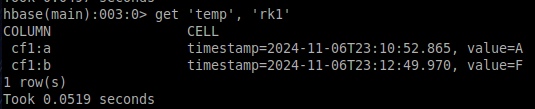
hbase(main):003:0> ﻿ ﻿put ‘temp’, ‘rk1’, ‘cf1:b’, ‘F’

hbase(main):004:0> scan ‘temp’



### Получение строки с ключом rk1 из временной таблицы. Get ‘’

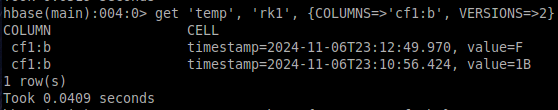
Hbase(main):003:0> ﻿ ﻿get ‘temp’, ‘rk1’



### Получение двух значений предыдущих версий столбца 'b' в строке rk1.

Hbase(main):003:0> ﻿ ﻿get ‘temp’, ‘rk1’, ﻿{COLUMNS => ‘cf1:b’, VERSIONS=>2}

hbase(main):004:0> get ‘temp’, ‘rk1’, {COLUMNS => ‘cf1:b’}



Примечание: Отображается первое входное значение 1B и последнее входное значение F. Если версия не указана, отображается только окончательное значение.

### Просмотр всей таблицы, но отображение только столбцов 'a', 'b', 'c'.

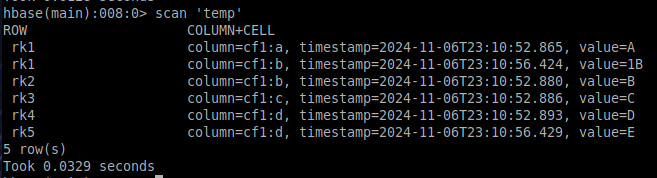
Hbase(main):003:0> ﻿ ﻿ ﻿scan ‘temp’, {COLUMNS => [‘cf1:b’, ‘cf1:a’, ‘cf1:c’]}

### Удаление ячейки из столбца 'b' из таблицы temp с ключом строки = rk1.

Hbase(main):003:0> ﻿ ﻿ ﻿ ﻿delete ‘temp’, ‘rk1’, ‘cf1:b’

### Убедитесь, что указанное значение в столбце 'b' было удалено. Если это не так, повторите команду выше.

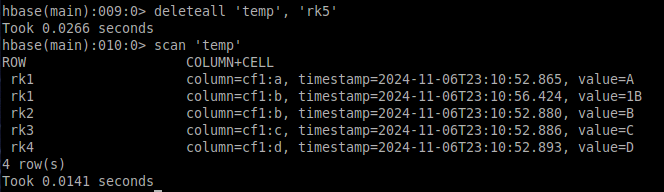
﻿Hbase(main):004:0> scan ‘temp’



### Удаление всей строки из временной таблицы в ключе строки rk1.

﻿Hbase(main):012:0> deleteall ‘temp’, ‘rk5’

### Проверьте, что строка с rk1 была удалена из временной таблицы.



## Использование MIN\_VERSIONS и TTL

### Выведите описание таблицы temp и проверьте значение TTL столбцов из cf2. Значение TTL (Time-To-Live) по умолчанию — FOREVER, что означает, что срок действия версий ячейки никогда не истекает. 텍스트, 모니터, 화면, 실내이(가) 표시된 사진 자동 생성된 설명

### Измените значение с FOREVER на 30 секунд. Это означает, что версия будет удалена через 30 секунд после вставки данных в таблицу..

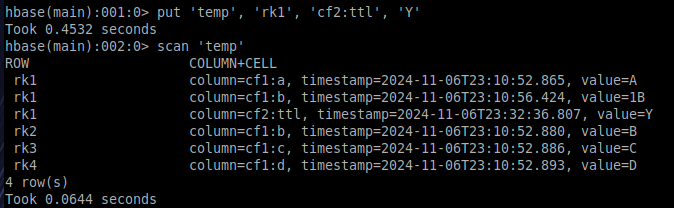
﻿Base(main):005:0> alter ‘temp’, NAME => ‘cf2’, TTL=> 30

### Проверьте, что TTL изменено с FOREVER на 30 секунд. 텍스트, 모니터, 실내, 화면이(가) 표시된 사진 자동 생성된 설명

### Вставим данные со следующими условиями и отобразим результаты.

RowKey:rk1, Column Family: cf2, column descriptor : ttl и значение Y

﻿hbase(main):007:0> put ‘temp’, ‘rk1’, ‘cf2:ttl’, ‘Y’



### Подождав не менее 30 секунд, выполните команду сканирования еще раз, чтобы узнать, истек ли срок действия вставленной строки и была ли она удалена. 텍스트, 실내이(가) 표시된 사진 자동 생성된 설명

### Установите TTL на 10 секунд и установите MIN\_VERSIONS на 1 дляcf2.

Hbase(main):014:0> alter ‘temp’, NAME => ‘cf2’, TTL=> 10, MIN\_VERSIONS => 1

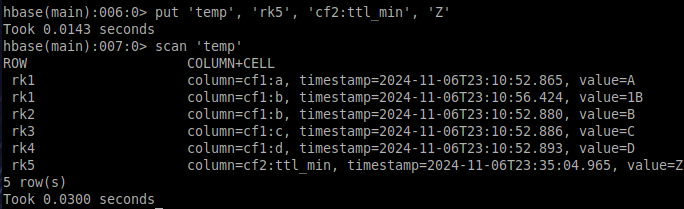
Примечание: Когда версия истекает с опцией MIN\_VERSION, она не будет удалена, если это единственная оставшаяся версия ячейки.

### Проверим значения TTL и MIN\_VERSIONS в cf2. 텍스트, 모니터, 실내, 화면이(가) 표시된 사진 자동 생성된 설명

### Вставим данные со следующими условиями и отобразим результаты.

RowKey:rk5, Column Family: cf2, column descriptor : ttl\_min and value: Z

﻿ ﻿hbase(main):016:0> put ‘temp’, ‘rk5’, ‘cf2:ttl\_min’, ‘Z’



### Подождав не менее 10 секунд, выполните команду сканирования еще раз, чтобы узнать, истек ли срок действия вставленной строки и осталась ли она. 텍스트, 실내, 전자기기, 키보드이(가) 표시된 사진 자동 생성된 설명

**Самостоятельная работа с HBase**

## Напишите команду для создания таблицы, отвечающей необходимым требованиям

#### Имя таблицы: **movie**

#### Содержит описание трех фильмов, каждому фильму соответствует столбец.

#### Все столбцы находятся в семействе title.

#### 

#### Название столбцов соответствуют названиям фильмов (выберите по желанию).

#### В ячейках таблицы описание.

#### 

#### Имя таблицы: **ranking**

#### Содержит информацию об оценках пользователей трех фильмов (опять же, фильмы - столбцы)

#### В ячейках таблицы оценки.

#### 

### Используйте команду alter, чтобы добавить семейство столбцов (column family) fl\_title для названий фильмов на другом языке.

### 

### Убедитесь, что семейство столбцов fl\_title добавлено.

### 

## Напишите команды, отвечающие следующим условиям

### Создайте таблицу user по следующим условиям:

#### Имя таблицы: **user**

#### Семейство столбцов: info

#### Создайте строку с Row key: rk100

#### Столбец age содержит значение 20

#### Столбец gender содержит значение F

#### Столбец zip содержит значение 18730

#### 

### Отобразите строку с ключом строки rk100 из пользовательской таблицы.

### 

### Кроме того, вставьте строку c rk100 в таблицу user со следующими атрибутами:

#### Row key: rk100

#### Семейство столбцов: info

#### Столбец age - значение 30

#### Row key: rk100

#### Семейство столбцов: info

#### Столбец age - значение 40

#### 

### Отобразите строки с ключом строки rk100 из пользовательской таблицы.

### В пользовательской таблице получите все старые версии столбца age в строке с ключом rk100.

### Отобразите всю таблицу с помощью команды сканирования.

### В таблице пользователей удалите столбец info:age строки с ключом строки rk100.

### Убедитесь, что столбец age удален.

### Удалите все строки с ключом строки rk100 из таблицы user.

### Убедитесь, что строка с ключом rk100 удалена из таблицы user.