

Колоночные хранилища

Объектные базы данных. Лекция 4

Колоночные хранилища

- Колоночные хранилища схожи с традиционными реляционными СУБД. Модель данных хранилищ данного типа подразумевает хранение значений как неинтерпретируемых байтовых массивов, адресуемых кортежами <ключ строки, ключ столбца, метка времени>.
- Типичным применением этого вида СУБД является веб-индексирование, а также задачи, связанные с большими данными, с пониженными требованиями к согласованности данных.
- Примеры СУБД: HBase, Cassandra, Hypertable, SimpleDB

HBase



- HBase — нереляционная распределенная база данных с открытым исходным кодом.
- Разработчик: Apache Software Foundation (в рамках проекта Hadoop)
- HBase написана на языке Java.
- Первый выпуск: 2006 г.
- Последняя версия: 1.1.0.1 (21 мая 2015 года)
- Лицензия: Apache License 2.0
- HBase используется в Facebook (платформа обмена сообщениями), StumbleUpon (хранилище данных, работающее в режиме реального времени, и обработка аналитических данных), Twitter и др.

facebook



HBase



- Хорошо подходит для систем оперативной аналитической обработки.
- Версионирование данных, заложенное в структуре хранения.
- В HBase изначально заложена отказоустойчивость. Использование упреждающей записи в журнал и распределенной конфигурации позволяет HBase быстро восстанавливаться после отказов отдельных серверов.

Представление данных в HBase

- **Словарь (*map*)** – набор пар ключ-значение.
- Пример:

red: “#F00”

blue: “#00F”

yellow: “#FF0”

square: “4”

triangle: “3”

Представление данных в HBase

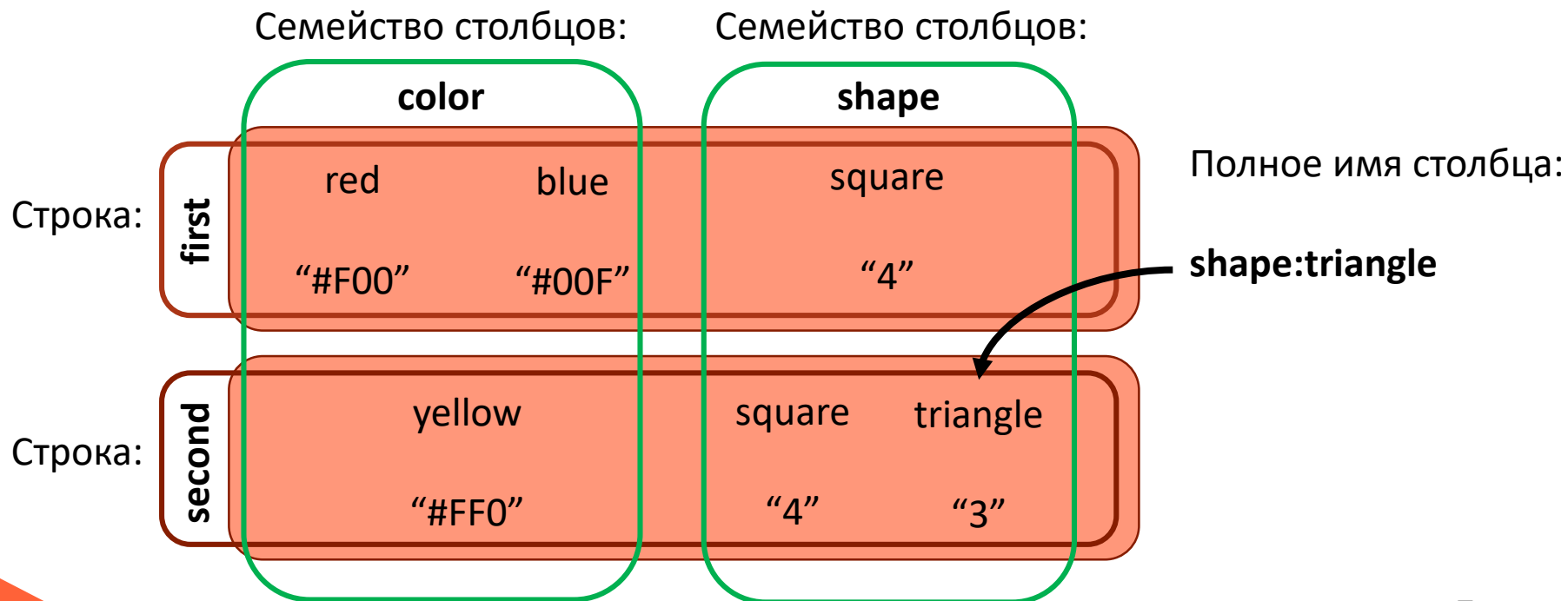
- **Словарь (map)** – набор пар ключ-значение.
- **Строка данных** – словарь, в котором ключи называются **столбцами**, а значения интерпретируются как массивы байтов.
- Пример:

Столбцы данных для строки first:
red, blue, square

Строка:	first	red	blue	square
		"#F00"	"#00F"	"4"
Строка:	second	yellow	square	triangle
		"#FF0"	"4"	"3"

Представление данных в HBase

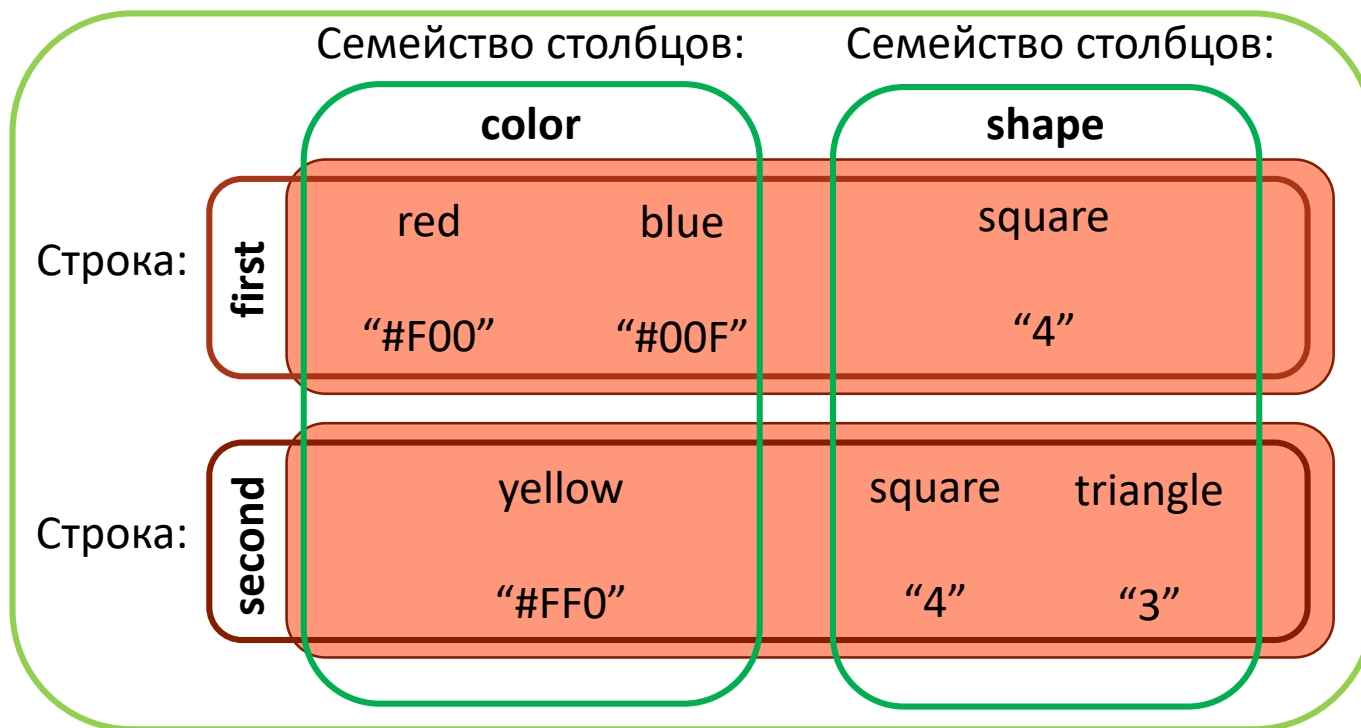
- **Словарь (map)** – набор пар ключ-значение.
- **Строка данных** – словарь, в котором ключи называются **столбцами**, а значения интерпретируются как массивы байтов.
- Столбцы группируются в **семейства столбцов**, полное имя столбца состоит из двух частей: имя семейства и **квалификатор столбца** ('family:qualifier').
- Пример:



Представление данных в HBase

- **Словарь (map)** – набор пар ключ-значение.
- **Строка данных** – словарь, в котором ключи называются **столбцами**, а значения интерпретируются как массивы байтов.
- Столбцы группируются в **семейства столбцов**, полное имя столбца состоит из двух частей: имя семейства и *квалификатор столбца* ('family:qualifier').
- **Таблица** – набор строк данных.

Таблица
Pictures:

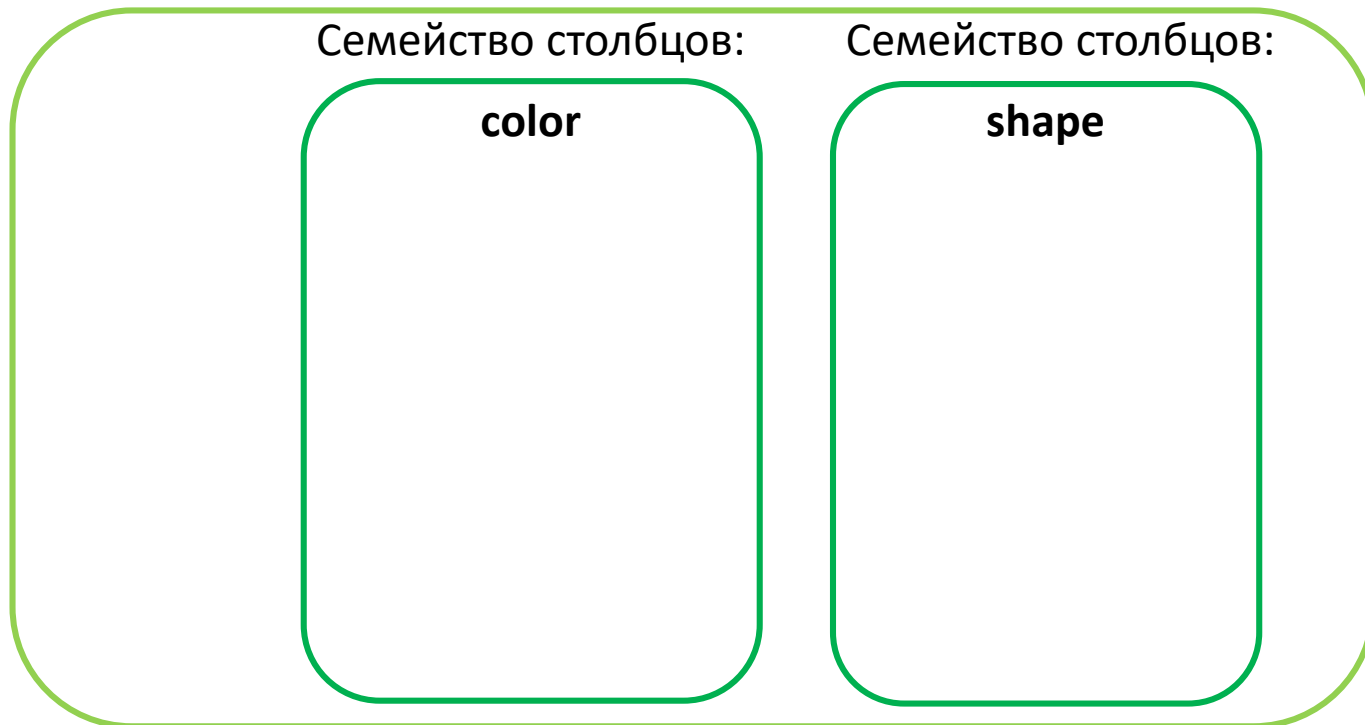


Создание таблицы Pictures

Создаем таблицу `Pictures` с семействами столбцов `color` и `shape` (таблица создается пустой без строк и столбцов):

```
hbase> create 'Pictures', 'color', 'shape'
```

Таблица
Pictures:



Вставка, обновление и выборка данных

Вставка данных в таблицу:

```
put 'table name', 'rowid', 'colfamily:colname', 'value'
```

где:

- `table name` - название таблицы;
- `rowid` - ключ строки данных в таблице;
- `colfamily:colname` - полное имя столбца, состоящее из названия семейства столбцов `colfamily` и квалификатора столбца `colname`;
- `value` - данные.

Вставка данных в таблицу Pictures

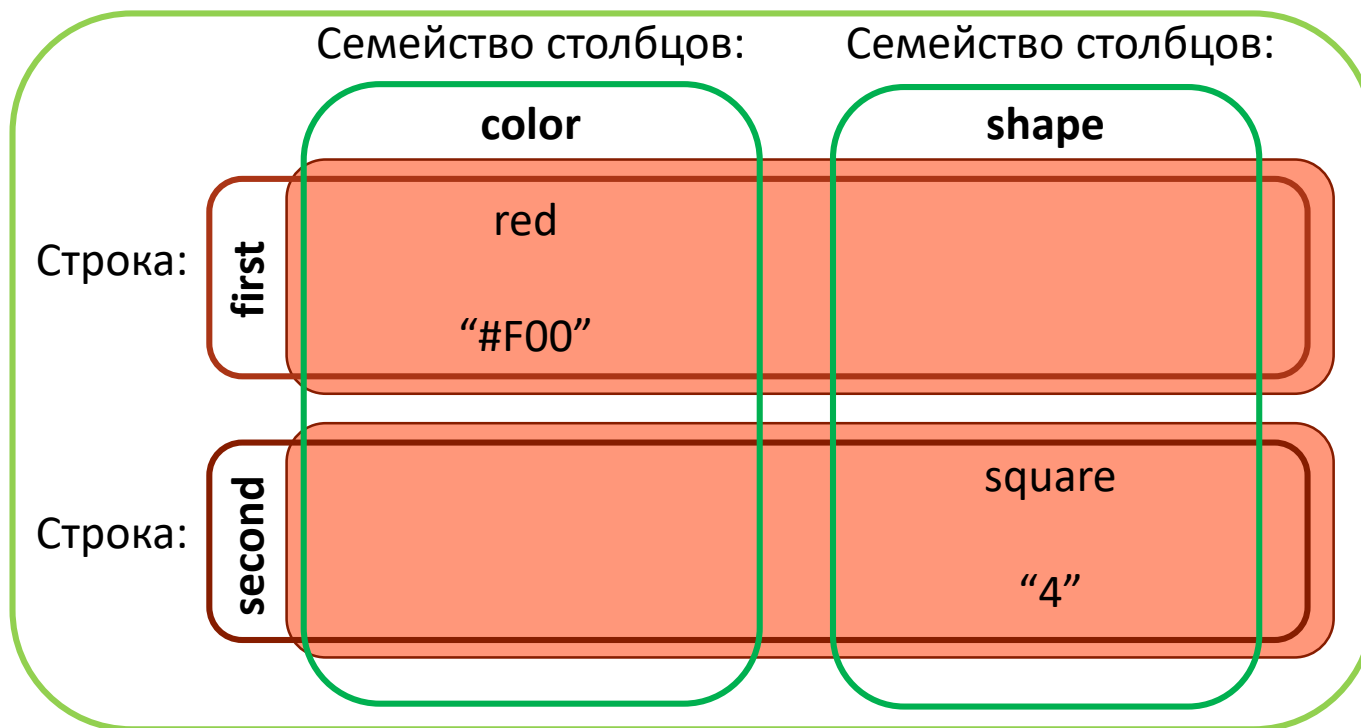
Вставка в таблицу Pictures новой строки с ключом first, содержащей столбец color:red со значением "#F00":

```
hbase> put 'Pictures', 'first', 'color:red', "#F00"
```

Вставка в таблицу Pictures новой строки с ключом second, содержащей столбец shape:square со значением "4":

```
hbase> put 'Pictures', 'second', 'shape:square', "4"
```

Таблица
Pictures:



Вставка, обновление и выборка данных

Обновление данных производится также командой **put**:

```
put 'table name', 'rowid', 'colfamily:colname', 'value'
```

Для обновления данных достаточно в поле `value` указать новое значение для существующего столбца таблицы.

Версионирование данных:

HBase хранит вместе с каждым значением данных **целочисленную временную метку** – количество миллисекунд, прошедших с момента «эпохи» (00:00:00 UTC 1 января 1970). Когда в ячейку записывается новое значение, старое не стирается, а остается вместе со своей временной меткой.

Обновление данных в таблице Pictures

Обновление в таблице Pictures столбца color:red строки с ключом first:

```
hbase> put 'Pictures', 'first', 'color:red', "#FF0000"
```

Таблица
Pictures:

Семейство столбцов:		Семейство столбцов:	
color		shape	
Строка:	first	red "#F00" 1417521885277 red "#FF0000" 1417521885289	
Строка:	second		square "4"

Вставка, обновление и выборка данных

Выборка данных из таблицы:

```
get 'table name', 'rowid' [, {COLUMN => 'colfamily:colname'}]
```

где:

- `table name` - название таблицы;
- `rowid` - ключ строки данных в таблице;
- `colfamily:colname` - полное имя столбца, состоящее из названия семейства столбцов `colfamily` и квалификатора столбца `colname`.

Выборка данных из таблицы Pictures

```
hbase> get 'Pictures', 'first'
```

```
COLUMN      CELL
```

```
color: red timestamp = 1417521885277, value = #F00
```

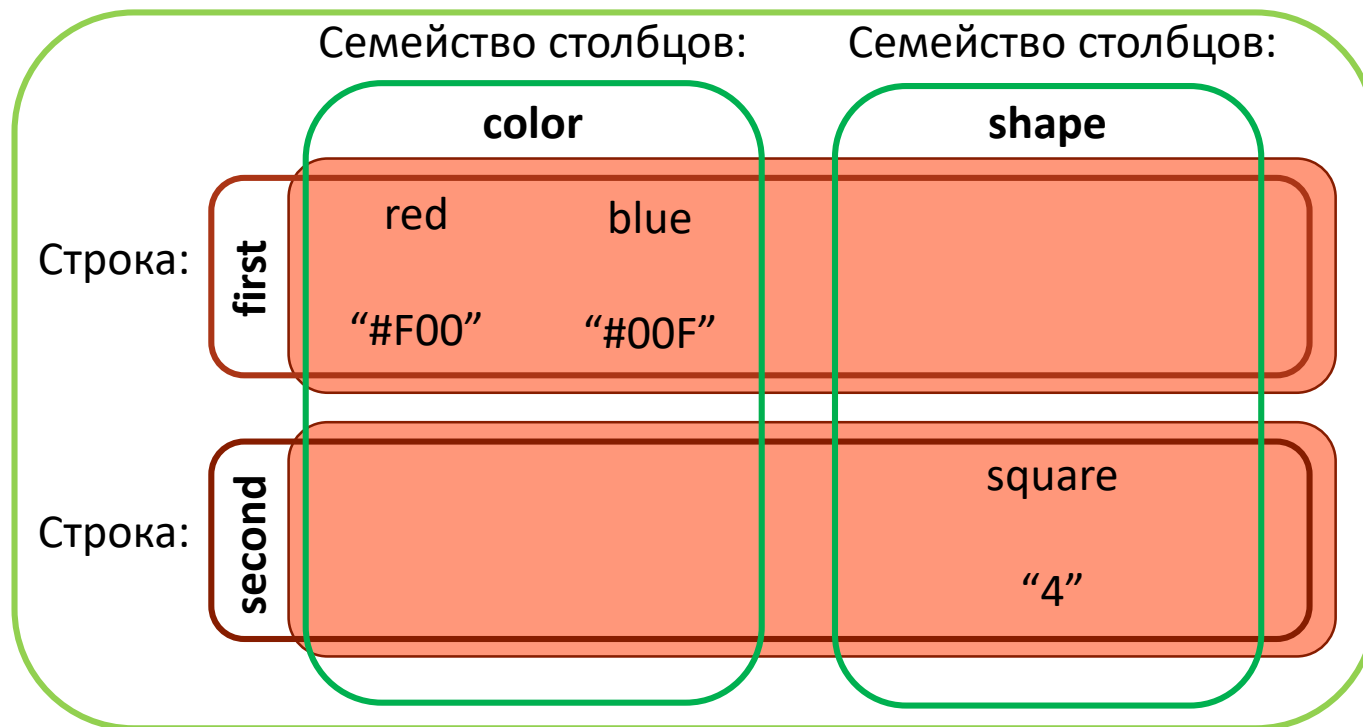
```
color: blue timestamp = 1417521885275, value = #00F
```

```
hbase> get 'Pictures', 'first', {COLUMN => 'color:blue'}
```

```
COLUMN      CELL
```

```
color: blue timestamp = 1417521885275, value = #00F
```

Таблица
Pictures:



Изменение таблиц

```
hbase> alter 'Pictures', { NAME => 'size', VERSIONS =>
hbase* org.apache.hadoop.hbase.HConstants::ALL_VERSIONS }
```

Таблица
Pictures:

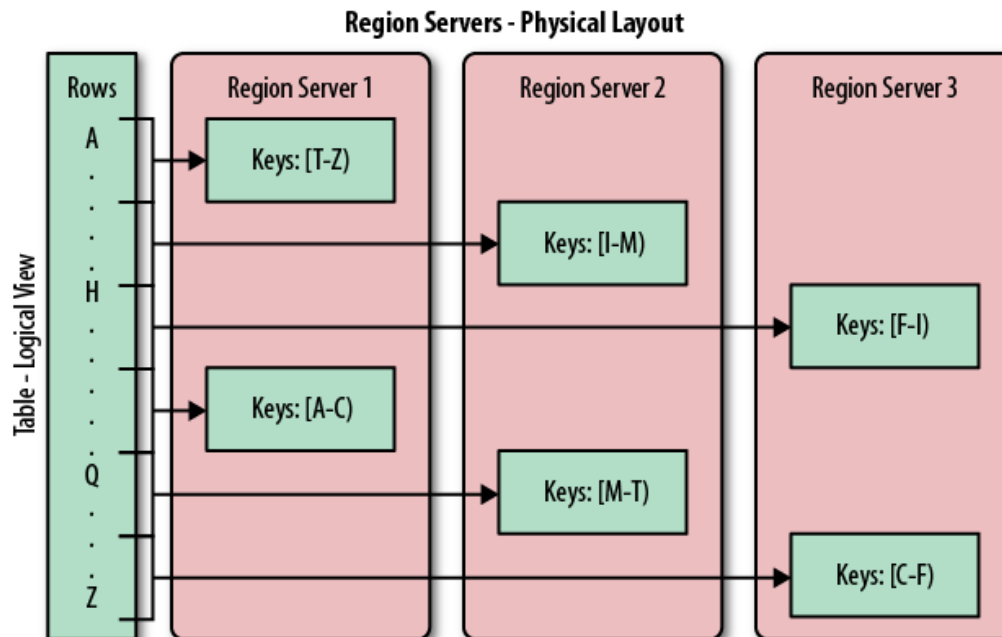
Семейства столбцов:			
color		shape	size
Строка:	first	red "#F00"	blue "#00F"
Строка:	second	square "4"	

Отказоустойчивость

- **Журнал упреждающей записи (WAL)** - это стандартный метод обеспечения целостности данных. Основная идея WAL состоит в том, что изменения в файлах с данными (где находятся таблицы и индексы) должны записываться только после того, как эти изменения были занесены в журнал, т. е. после того как записи журнала, описывающие данные изменения, будут сохранены на постоянное устройство хранения.
- Упреждающая запись в журнал называется **журналированием**.
- В HBase упреждающая запись применяется для защиты от отказов узлов. Информация в WAL заносится до фиксации результатов операций редактирования (put и increment) на диске.

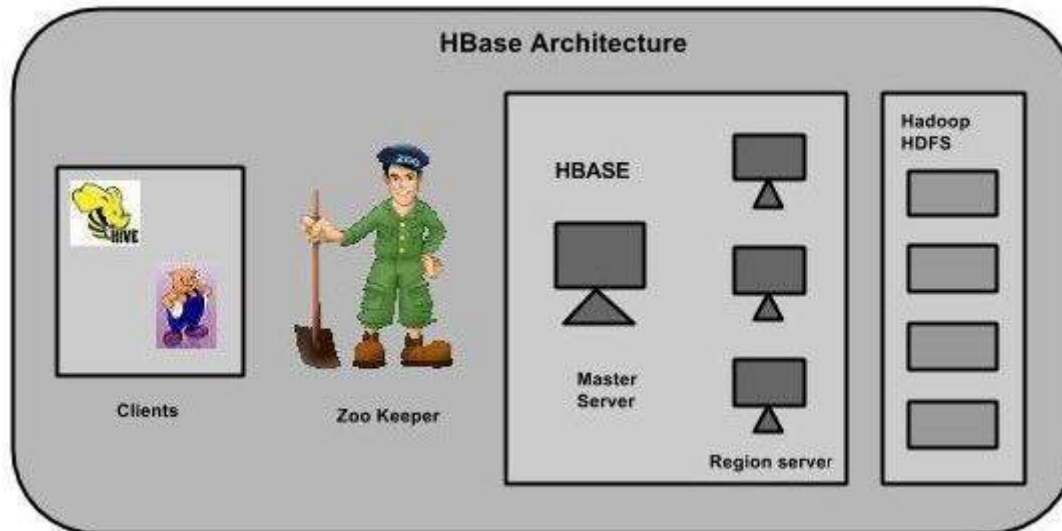
Регионы в HBase

- В HBase строки хранятся отсортированными по ключу.
- **Регионом** называется множество строк, определяемое начальным ключом (включительно) и конечным (исключительно). Регионы не перекрываются, и каждому из них назначается региональный сервер в кластере.



Распределение регионов по серверам

- Распределением регионов по региональным серверам отвечает *главный* узел, который часто называют **HMaster**. Главный сервер может одновременно выполнять функции регионального сервера.
- Если региональный сервер выходит из строя, то главный сервер вмешивается и передает кому-то ответственность за обслуживание регионов, назначенных отказавшему узлу. Новый смотритель заглянет в WAL и посмотрит, нужны ли какие-нибудь восстановительные действия. Если же выходит из строя главный сервер, то его роль принимает на себя один из региональных серверов.



Режимы работы HBase

- 1) автономный режим с одной-единственной машиной;
- 2) псевдораспределенный режим с одним узлом, «притворяющимся» кластером;
- 3) полностью распределенный режим с кластером совместно работающих узлов.