

1. HBase命令行演示

1.1. HBase命令使用初准备

1.1.1. 先进入hbase shell命令行

1.1.2. help

1.1.3. 语法规则

2. HBase命令使用实战

1. HBase命令行演示

概述：大数据生态里的各种软件，基本都会给出shell命令行操作。所以在拿到新上手的软件的时候先找到怎么进入命令行，然后相应要想到help命令，查看命令帮助。别着急一股脑儿扎进去敲各种命令，这就是思路，思路很重要。

1.1. HBase命令使用初准备

下面按照我的思路来进行练习：

1.1.1. 先进入hbase shell命令行

在你安装的随意台服务器节点上，执行命令：**hbase shell**，会进入到你的hbase shell客户端

```
[bigdata@bigdata02 ~]# hbase shell
```

1.1.2. help

进入之后先别着急，先看一下提示。其实是不是有一句很重要的话：

```
HBase Shell; enter 'help<RETURN>' for list of supported commands.  
Type "exit<RETURN>" to leave the HBase Shell
```

意在告诉怎么获得帮助，怎么退出客户端

```
help  获取所有命令提示  
help "dm1"  获取一组命令的提示  
help "put"  获取一个单独命令的提示帮助
```

exit 退出 hbase shell 客户端

1.1.3. 语法规则

名称	命令表达式
创建表	create '表名称', '列名称1', '列名称2', '列名称N'
添加记录	put '表名称', '行名称', '列名称:', '值'
查看记录	get '表名称', '行名称'
查看表中的记录总数	count '表名称'
删除记录	delete '表名', '行名称', '列名称'
删除一张表	先要屏蔽该表, 才能对该表进行删除, 第一步 disable '表名称' 第二步 drop '表名称'
查看所有记录	scan "表名称"
查看某个表某个列中所有数据	scan "表名称", ['列名称:']
更新记录	就是重写一遍进行覆盖

2. HBase命令使用实战

1、显示hbase中的表列表：list

```
list
```

2、创建表

创建一张hbase表, 表名叫做user, 该表有info和data两个列簇, 注意, 创建表的时候不用指定列的信息, 插入数据的时候才需要指定key-value的信息, 这个key就是列

```
create 'user', 'info', 'data'
```

也可以这样写:

```
create 'user',{NAME=>'info'},{NAME=>'data'}
```

创建一张表叫做user_info, 包含两个列簇base_info和extra_info, 并且分别指定这两个列簇的数据版本号3和1

```
create 'user_info',{NAME=>'base_info',VERSIONS=>3 },
```

```
{NAME=>'extra_info',VERSIONS=>1}
```

```
create 'user',{NAME=>'info',VERSIONS=>3 },{NAME=>'data',VERSIONS=>2}
```

3、查看表的详细信息：desc或者describe

```
desc "user_info"
```

或者:

```
describe "user_info"
```

4、往表中插入数据：put

向user表中插入信息, row key为rk0001, 列簇info中添加name列标示符, 值为zhangsan
put 'user', 'rk0001', 'info:name', 'zhangsan'

向user表中插入信息, row key为rk0001, 列簇info中添加gender列标示符, 值为female
put 'user', 'rk0001', 'info:gender', 'female'

向user表中插入信息, row key为rk0001, 列簇info中添加age列标示符, 值为20
put 'user', 'rk0001', 'info:age', 20

向user表中插入信息, row key为rk0001, 列簇data中添加pic列标示符, 值为picture
put 'user', 'rk0001', 'data:pic', 'picture'

再插入几条其他数据:

```
put 'user', 'rk0002', 'info:name', 'fanbingbing'
put 'user', 'rk0002', 'info:gender', 'female'
put 'user', 'rk0002', 'info:nationality', '中国'
```

插入一堆示例数据, 以便后面测试:

```
put 'user_info', 'user0000', 'base_info:name', 'luoyufeng'
put 'user_info', 'user0000', 'base_info:age', '18'
put 'user_info', 'user0000', 'base_info:gender', 'female'
put 'user_info', 'user0000', 'extra_info:size', '34'
put 'user_info', 'user0001', 'base_info:name', 'zhangsan'
put 'user_info', 'user0001', 'base_info:name', 'luoyufeng'
put 'user_info', 'user0001', 'base_info:name', 'zhangsan'
```

```
put 'user_info', 'zhangsan_20150701_0001', 'base_info:name', 'zhangsan1'
put 'user_info', 'zhangsan_20150701_0002', 'base_info:name', 'zhangsan2'
put 'user_info', 'zhangsan_20150701_0003', 'base_info:name', 'zhangsan3'
put 'user_info', 'zhangsan_20150701_0004', 'base_info:name', 'zhangsan4'
put 'user_info', 'zhangsan_20150701_0005', 'base_info:name', 'zhangsan5'
put 'user_info', 'zhangsan_20150701_0006', 'base_info:name', 'zhangsan6'
put 'user_info', 'zhangsan_20150701_0007', 'base_info:name', 'zhangsan7'
put 'user_info', 'zhangsan_20150701_0008', 'base_info:name', 'zhangsan8'
```

```
put 'user_info', 'zhangsan_20150701_0001', 'base_info:age', '21'
put 'user_info', 'zhangsan_20150701_0002', 'base_info:age', '22'
put 'user_info', 'zhangsan_20150701_0003', 'base_info:age', '23'
put 'user_info', 'zhangsan_20150701_0004', 'base_info:age', '24'
put 'user_info', 'zhangsan_20150701_0005', 'base_info:age', '25'
put 'user_info', 'zhangsan_20150701_0006', 'base_info:age', '26'
put 'user_info', 'zhangsan_20150701_0007', 'base_info:age', '27'
put 'user_info', 'zhangsan_20150701_0008', 'base_info:age', '28'
```

```
put 'user_info', 'zhangsan_20150701_0001', 'extra_info:Hobbies', 'music'
put 'user_info', 'zhangsan_20150701_0002', 'extra_info:Hobbies', 'sport'
put 'user_info', 'zhangsan_20150701_0003', 'extra_info:Hobbies', 'music'
put 'user_info', 'zhangsan_20150701_0004', 'extra_info:Hobbies', 'sport'
put 'user_info', 'zhangsan_20150701_0005', 'extra_info:Hobbies', 'music'
put 'user_info', 'zhangsan_20150701_0006', 'extra_info:Hobbies', 'sport'
put 'user_info', 'zhangsan_20150701_0007', 'extra_info:Hobbies', 'music'
```

```
put 'user_info', 'baiyc_20150716_0001', 'base_info:name', 'baiyc1'
put 'user_info', 'baiyc_20150716_0002', 'base_info:name', 'baiyc2'
put 'user_info', 'baiyc_20150716_0003', 'base_info:name', 'baiyc3'
put 'user_info', 'baiyc_20150716_0004', 'base_info:name', 'baiyc4'
put 'user_info', 'baiyc_20150716_0005', 'base_info:name', 'baiyc5'
put 'user_info', 'baiyc_20150716_0006', 'base_info:name', 'baiyc6'
```

```

put 'user_info', 'baiyc_20150716_0007', 'base_info:name', 'baiyc7'
put 'user_info', 'baiyc_20150716_0008', 'base_info:name', 'baiyc8'

put 'user_info', 'baiyc_20150716_0001', 'base_info:age', '21'
put 'user_info', 'baiyc_20150716_0002', 'base_info:age', '22'
put 'user_info', 'baiyc_20150716_0003', 'base_info:age', '23'
put 'user_info', 'baiyc_20150716_0004', 'base_info:age', '24'
put 'user_info', 'baiyc_20150716_0005', 'base_info:age', '25'
put 'user_info', 'baiyc_20150716_0006', 'base_info:age', '26'
put 'user_info', 'baiyc_20150716_0007', 'base_info:age', '27'
put 'user_info', 'baiyc_20150716_0008', 'base_info:age', '28'

put 'user_info', 'baiyc_20150716_0001', 'extra_info:Hobbies', 'music'
put 'user_info', 'baiyc_20150716_0002', 'extra_info:Hobbies', 'sport'
put 'user_info', 'baiyc_20150716_0003', 'extra_info:Hobbies', 'music'
put 'user_info', 'baiyc_20150716_0004', 'extra_info:Hobbies', 'sport'
put 'user_info', 'baiyc_20150716_0005', 'extra_info:Hobbies', 'music'
put 'user_info', 'baiyc_20150716_0006', 'extra_info:Hobbies', 'sport'
put 'user_info', 'baiyc_20150716_0007', 'extra_info:Hobbies', 'music'
put 'user_info', 'baiyc_20150716_0008', 'extra_info:Hobbies', 'sport'

```

5、查询数据: get

获取user表中row key为rk0001的所有信息

```
get 'user', 'rk0001'
```

获取user表中row key为rk0001, info列簇的所有信息

```
get 'user', 'rk0001', 'info'
```

获取user表中row key为rk0001, info列簇的name、age列标示符的信息

```
get 'user', 'rk0001', 'info:name', 'info:age'
```

获取user表中row key为rk0001, info和data列簇的信息

```
get 'user', 'rk0001', 'info', 'data'
```

```
get 'user', 'rk0001', {COLUMN => ['info', 'data']}
```

```
get 'user', 'rk0001', {COLUMN => ['info:name', 'data:pic']}
```

获取user表中row key为rk0001, 列簇为info, 版本号最新5个的信息

```
get 'user', 'rk0001', {COLUMN => 'info', VERSIONS => 2}
```

```
get 'user', 'rk0001', {COLUMN => 'info:name', VERSIONS => 5}
```

```
get 'user', 'rk0001', {COLUMN => 'info:name', VERSIONS => 5, TIMERANGE =>
[1392368783980, 1392380169184]}
```

```
get 'user_info', 'user0001', {COLUMN => 'base_info:name', VERSIONS => 3}
```

获取user表中row key为rk0001, cell的值为zhangsan的信息

```
get 'user', 'rk0001', {FILTER => "ValueFilter(=, 'binary:zhangsan')"}'
```

获取user表中row key为rk0001, 列标示符中含有a的信息

```
get 'user', 'rk0001', {FILTER => "(QualifierFilter(=, 'substring:a'))"}'
```

6、查询数据: scan

查询user_info表中的所有信息

```
scan 'user_info'
```

查询user_info表中的指定列簇的所有信息

```
scan 'user_info', {COLUMNS => 'base_info'}
scan 'user_info', {COLUMNS => 'base_info:name'}
scan 'user_info', {COLUMNS => ['base_info', 'extra_info']}
scan 'user_info', {COLUMNS => ['base_info:name', 'extra_info:Hobbies']}
```

查询user_info表中的指定列簇为base_info的所有版本信息

```
scan 'user_info', {COLUMNS => 'base_info'}          ##只查询最新值
scan 'user_info', {COLUMNS => 'base_info', VERSIONS => 5}
```

查询user表中列簇为info和data的信息

```
scan 'user', {COLUMNS => ['info', 'data']}
scan 'user', {COLUMNS => 'info:name'}
scan 'user', {COLUMNS => ['info:name', 'data:pic']}
```

查询user_info表中列簇为base_info、列标示符为name的信息，并且版本最新的5个

```
scan 'user_info', {COLUMNS => 'base_info:name', VERSIONS => 5}
```

查询user表中列簇为info和data且列标示符中含有a字符的信息

```
scan 'user', {COLUMNS => ['info', 'data']}
scan 'user', {COLUMNS => ['info', 'data'], FILTER =>
(QualifierFilter(=, 'substring:a'))"}
```

查询表名为user_info表中列簇为base_info，rowkey的起始偏移范围是[baiyc_20150716_0003, baiyc_20150716_0006)的数据

```
scan 'user_info', {COLUMNS => 'base_info', STARTROW => 'baiyc_20150716_0003',
ENDROW => 'baiyc_20150716_0006'}
```

查询user表中rowkey以rk字符开头的

```
scan 'user', {FILTER=>"PrefixFilter('rk')"} }
```

查询user_info表中指定时间戳范围的数据

```
scan 'user_info', {TIMERANGE => [1540882871681,1540882888540]}
```

7、删除数据: delete

删除记录

```
delete 'user', 'rk0001'    ##不能一口气删除一个rowkey所对应的所有key-value
```

删除字段

```
delete 'user', 'rk0001', 'info:name'
```

删除user_info表rowkey为用户0000，列标示符为info:name的数据

```
get 'user_info', 'user0000', 'base_info:name'
delete 'user_info', 'user0000', 'base_info:name'    ##实现删除，其他两句为辅助测试语句
put 'user_info', 'user0000', 'base_info:name', 'luoyufeng'
```

删除user表rowkey为rk0001，列标示符为info:name，timestamp为1392383705316的数据

```
delete 'user', 'rk0001', 'info:name', 1392383705316
```

8、修改表结构：alter

添加两个列簇f2和f3

```
alter 'user_info', NAME => 'f2'
```

```
alter 'user_info', NAME => 'f3'
```

删除一个列簇f2:

```
alter 'user_info', NAME => 'f2', METHOD => 'delete'
```

或

```
alter 'user_info', 'delete' => 'f2'
```

添加列簇f1同时删除列簇f3

```
alter 'user_info', {NAME => 'f1'}, {NAME => 'f3', METHOD => 'delete'}
```

将user_info表的base_info列簇版本号改为5

```
alter 'user_info', NAME => 'base_info', VERSIONS => 5
```

9、修改数据

严格来说，HBase没有修改数据的显示操作，重复插入就相当于修改操作

10、清空表：truncate

清空user表中的数据

```
truncate 'user'
```

11、停用表/启用表：disable和enable

首先停用user表

```
disable 'user'
```

启用表

```
enable 'user'
```

12、删除表：drop

删除表一定要停用，然后删除

```
disable 'user'
```

```
drop 'user'
```

13、过滤器操作

```
get 'user', 'rk0001', {FILTER => "ValueFilter(=, 'binary:中国')"}
get 'user', 'rk0001', {FILTER => "(QualifierFilter(=,'substring:a'))"}

scan 'user', {COLUMNS => 'info:name'}
scan 'user', {COLUMNS => ['info', 'data'], FILTER =>
(QualifierFilter(=,'substring:a'))"}
scan 'user', {COLUMNS => 'info', STARTROW => 'rk0001', ENDROW => 'rk0003'}
scan 'user', {COLUMNS => 'info', STARTROW => '20140201', ENDROW => '20140301'}
scan 'user', {COLUMNS => 'info:name', TIMERANGE => [1395978233636,
1395987769587]}
```