版本问题

# 查看

在HBase中 一个row对应的相同的列只会有一行。使用scan 或get 得到都是最新的数据，如果我们对这某一row所对应的列进行了更改操作后，并不会多生成一条数据，

不会像数据库一样，插入时多生成一条记录，在HBase中对同一条数据的修改或插入都只是put操作，最终看到的都是最新的数据，其它的数据在不同的version中保存，

默认是隐藏的，通过时间戳区分，Hbase默认保存最近的三个版本，如何才能看到这些旧版本的数据了？

## 插入测试数据：

[sql] view plaincopy

hbase(main):026:0> put 'testtable1','row-1','colfam1:qual1','你好，中国'

0 row(s) in 0.0200 seconds

hbase(main):027:0> put 'testtable1','row-1','colfam1:qual1','你好，广州'

0 row(s) in 0.0130 seconds

hbase(main):027:0> put 'testtable1','row-1','colfam1:qual1','welcome，hbase'

0 row(s) in 0.0130 seconds

## 下面3个命令，只显示最近的三个版本

[sql] view plaincopy

hbase(main):015:0> get 'testtable1','row-1',{COLUMN=>'colfam1:qual1',VERSIONS=>10}

COLUMN CELL

colfam1:qual1 timestamp=1410943676361, value=welcome\xEF\xBC\x8Chbase

colfam1:qual1 timestamp=1410942935244, value=\xE4\xBD\xA0\xE5\xA5\xBD\xEF\xBC\x8C\xE5\xB9\xBF\xE5\xB7\x9E

colfam1:qual1 timestamp=1410942917285, value=\xE4\xBD\xA0\xE5\xA5\xBD\xEF\xBC\x8C\xE4\xB8\xAD\xE5\x9B\xBD

3 row(s) in 0.0270 seconds

hbase(main):016:0> scan 'testtable1', {COLUMN=>'colfam1:qual1',VERSIONS=>10}

ROW COLUMN+CELL

row-1 column=colfam1:qual1, timestamp=1410943676361, value=welcome\xEF\xBC\x8Chbase

row-1 column=colfam1:qual1, timestamp=1410942935244, value=\xE4\xBD\xA0\xE5\xA5\xBD\xEF\xBC\x8C\xE5\xB9

\xBF\xE5\xB7\x9E

row-1 column=colfam1:qual1, timestamp=1410942917285, value=\xE4\xBD\xA0\xE5\xA5\xBD\xEF\xBC\x8C\xE4\xB8

\xAD\xE5\x9B\xBD

1 row(s) in 0.0300 seconds

hbase(main):017:0> scan 'testtable1', {FILTER => "PrefixFilter ('row-1')",COLUMN=>'colfam1:qual1',VERSIONS=>10}

ROW COLUMN+CELL

row-1 column=colfam1:qual1, timestamp=1410943676361, value=welcome\xEF\xBC\x8Chbase

row-1 column=colfam1:qual1, timestamp=1410942935244, value=\xE4\xBD\xA0\xE5\xA5\xBD\xEF\xBC\x8C\xE5\xB9

\xBF\xE5\xB7\x9E

row-1 column=colfam1:qual1, timestamp=1410942917285, value=\xE4\xBD\xA0\xE5\xA5\xBD\xEF\xBC\x8C\xE4\xB8

\xAD\xE5\x9B\xBD

1 row(s) in 0.0220 seconds

## 下面2个命令，可以显示所有的版本

[sql] view plaincopy

hbase(main):018:0> scan 'testtable1',{FILTER => "(QualifierFilter (>=, 'binary:qual1')))",RAW => true, VERSIONS => 10}

ROW COLUMN+CELL

row-1 column=colfam1:qual1, timestamp=1410943676361, value=welcome\xEF\xBC\x8Chbase

row-1 column=colfam1:qual1, timestamp=1410942935244, value=\xE4\xBD\xA0\xE5\xA5\xBD\xEF\xBC\x8C\xE5\xB9

\xBF\xE5\xB7\x9E

row-1 column=colfam1:qual1, timestamp=1410942917285, value=\xE4\xBD\xA0\xE5\xA5\xBD\xEF\xBC\x8C\xE4\xB8

\xAD\xE5\x9B\xBD

row-1 column=colfam1:qual1, timestamp=1410936055137, value=\xE4\xB8\xAD\xE5\x9B\xBD\xE7\xAC\xAC\xE4\xB8

\x89\xE6\x96\xB9\xE7\x9A\x84\xE4\xBA\x8C\xE6\x96\xB9\xE7\x9A\x84

row-1 column=colfam1:qual1, timestamp=1410936031157, value=\xE4\xB8\xAD\xE5\x9B\xBD\xE7\xAC\xAC\xE4\xB8

\x89\xE6\x96\xB9\xE7\x9A\x84

1 row(s) in 0.0290 seconds

hbase(main):019:0> scan 'testtable1',{FILTER => "PrefixFilter ('row-1')",RAW => true, VERSIONS => 10}

ROW COLUMN+CELL

row-1 column=colfam1:qual1, timestamp=1410943676361, value=welcome\xEF\xBC\x8Chbase

row-1 column=colfam1:qual1, timestamp=1410942935244, value=\xE4\xBD\xA0\xE5\xA5\xBD\xEF\xBC\x8C\xE5\xB9

\xBF\xE5\xB7\x9E

row-1 column=colfam1:qual1, timestamp=1410942917285, value=\xE4\xBD\xA0\xE5\xA5\xBD\xEF\xBC\x8C\xE4\xB8

\xAD\xE5\x9B\xBD

row-1 column=colfam1:qual1, timestamp=1410936055137, value=\xE4\xB8\xAD\xE5\x9B\xBD\xE7\xAC\xAC\xE4\xB8

\x89\xE6\x96\xB9\xE7\x9A\x84\xE4\xBA\x8C\xE6\x96\xB9\xE7\x9A\x84

row-1 column=colfam1:qual1, timestamp=1410936031157, value=\xE4\xB8\xAD\xE5\x9B\xBD\xE7\xAC\xAC\xE4\xB8

\x89\xE6\x96\xB9\xE7\x9A\x84

row-1 column=colfam2:col-0, timestamp=1410935938913, value=val-1.0

row-1 column=colfam2:col-1, timestamp=1410935938921, value=val-1.1

row-1 column=colfam2:col-2, timestamp=1410935938927, value=val-1.2

row-1 column=colfam2:col-3, timestamp=1410935938929, value=val-1.3

row-1 column=colfam2:col-4, timestamp=1410935938932, value=val-1.4

row-1 column=colfam2:col-5, timestamp=1410935938935, value=val-1.5

row-1 column=colfam2:col-6, timestamp=1410935938937, value=val-1.6

row-1 column=colfam2:col-7, timestamp=1410935938939, value=val-1.7

row-1 column=colfam2:col-8, timestamp=1410935938941, value=val-1.8

row-1 column=colfam2:col-9, timestamp=1410935938944, value=val-1.9

1 row(s) in 0.0690 seconds

## 用java代码测试：

[java] view plaincopy

package client;

// cc GetExample Example application retrieving data from HBase

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.hbase.HBaseConfiguration;

import org.apache.hadoop.hbase.client.Get;

import org.apache.hadoop.hbase.client.HTable;

import org.apache.hadoop.hbase.client.Result;

import org.apache.hadoop.hbase.util.Bytes;

import util.HBaseHelper;

import java.io.IOException;

import org.apache.hadoop.hbase.KeyValue;

import java.util.List;

public class GetExample {

public static void main(String[] args) throws IOException {

// vv GetExample

Configuration conf = HBaseConfiguration.create(); // co GetExample-1-CreateConf Create the configuration.

conf.set("hbase.zookeeper.property.clientPort", "2181");

conf.set("hbase.zookeeper.quorum", "jifeng01");

conf.set("zookeeper.znode.parent", "/hbase");

/\*/ ^^ GetExample

HBaseHelper helper = HBaseHelper.getHelper(conf);

if (!helper.existsTable("testtable1")) {

helper.createTable("testtable1", "colfam1");

}

\*/

//vv GetExample

HTable table = new HTable(conf, "testtable1"); // co GetExample-2-NewTable Instantiate a new table reference.

Get get = new Get(Bytes.toBytes("row-1")); // co GetExample-3-NewGet Create get with specific row.

get.setMaxVersions();

get.addColumn(Bytes.toBytes("colfam1"), Bytes.toBytes("qual1"));

Result result = table.get(get);

List<KeyValue> list = result.list();

for(final KeyValue kv:list){

// System.out.println("value: "+ kv+ " str: "+Bytes.toString(kv.getValue()));

System.out.println(String.format("row:%s, family:%s, qualifier:%s, qualifiervalue:%s, timestamp:%s.",

Bytes.toString(kv.getRow()),

Bytes.toString(kv.getFamily()),

Bytes.toString(kv.getQualifier()),

Bytes.toString(kv.getValue()),

kv.getTimestamp()));

}

/\*

get.addColumn(Bytes.toBytes("colfam1"), Bytes.toBytes("qual1")); // co GetExample-4-AddCol Add a column to the get.

Result result = table.get(get); // co GetExample-5-DoGet Retrieve row with selected columns from HBase.

byte[] val = result.getValue(Bytes.toBytes("colfam1"),

Bytes.toBytes("qual1")); // co GetExample-6-GetValue Get a specific value for the given column.

System.out.println("Value: " + Bytes.toString(val)); // co GetExample-7-Print Print out the value while converting it back.

\*/

// ^^ GetExample

}

}

输出结果：

[sql] view plaincopy

row:row-1, family:colfam1, qualifier:qual1, qualifiervalue:welcome，hbase, timestamp:1410943676361.

row:row-1, family:colfam1, qualifier:qual1, qualifiervalue:你好，广州, timestamp:1410942935244.

row:row-1, family:colfam1, qualifier:qual1, qualifiervalue:你好，中国, timestamp:1410942917285.

# 删除

删除指定版本的数据：

delete 'testtable1','row-1','colfam1:qual1',1433337394363

注意：如果是删除最新的版本，那么将查不出数据了。

# 列族操作

## 增加

1、表置为不可用：disable 'scores'

2、增加列族：alter 'scores',NAME=>'info'

3、表可用：enable 'scores'

## 删除

alter 't1′， NAME => 'f1′， METHOD => 'delete'

alter 't1′， 'delete' => 'f1′