

**数学与信息学院学生实验报告**

**实验课程名称：** 大数据技术概论 **教师： \_\_**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验项目名称** | **熟悉常用的HBase操作** | | | **实验成绩** |  |
| **学生姓名** |  | **学 号** | **29** | **年级专业班级** |  |
| **小组成员** | **无** | | | **实验日期** |  |

# 1. 实验目的和要求

## 1.1 实验目的

* 理解HBase在Hadoop体系结构中的角色；
* 熟练使用HBase操作常用的Shell命令；
* 熟悉HBase操作常用的Java API。

## 1.2 实验软硬件环境

* 操作系统： Ubuntu19.04；
* Hadoop版本：3.2.0；
* HBase版本：2.2.1；
* JDK版本：jdk-13；
* Java IDE：Eclipse 4.3。

## 1.3 实验要求

① 在Linux系统上安装Hbase，记录软件安装、测试的过程及结果。

②用Hadoop提供的HBase Shell命令完成指定任务。

③ 编程实现相应的功能

# 2. 实验记录

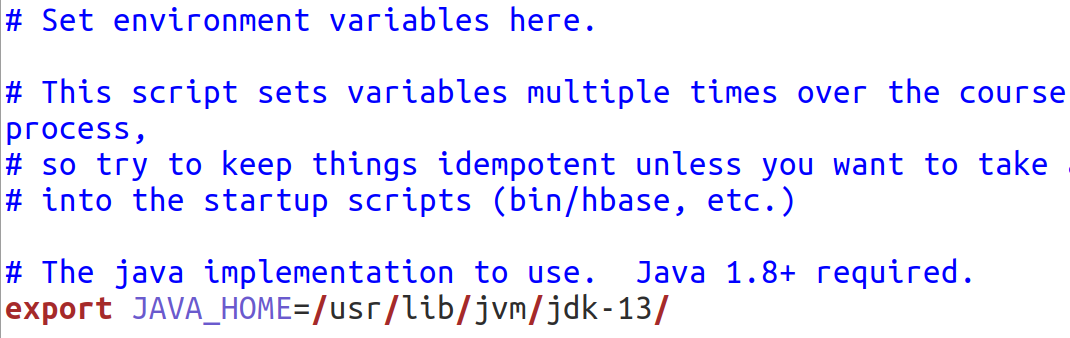
## 2.1 安装Hbase

**1.**在以下链接下载Hbase(很快的)

**http://mirrors.tuna.tsinghua.edu.cn/apache/hbase/2.2.1/hbase-2.2.1-bin.tar.gz**

下载之后解压到用户主目录下

**2.**在启动HBase之前设置环境变量，在**conf / hbase-env.sh**文件中进行设置，把**JAVA\_HOME**的路径改成自己java的具体路径，找到它的一种方法是使用在终端输入**whereis java**



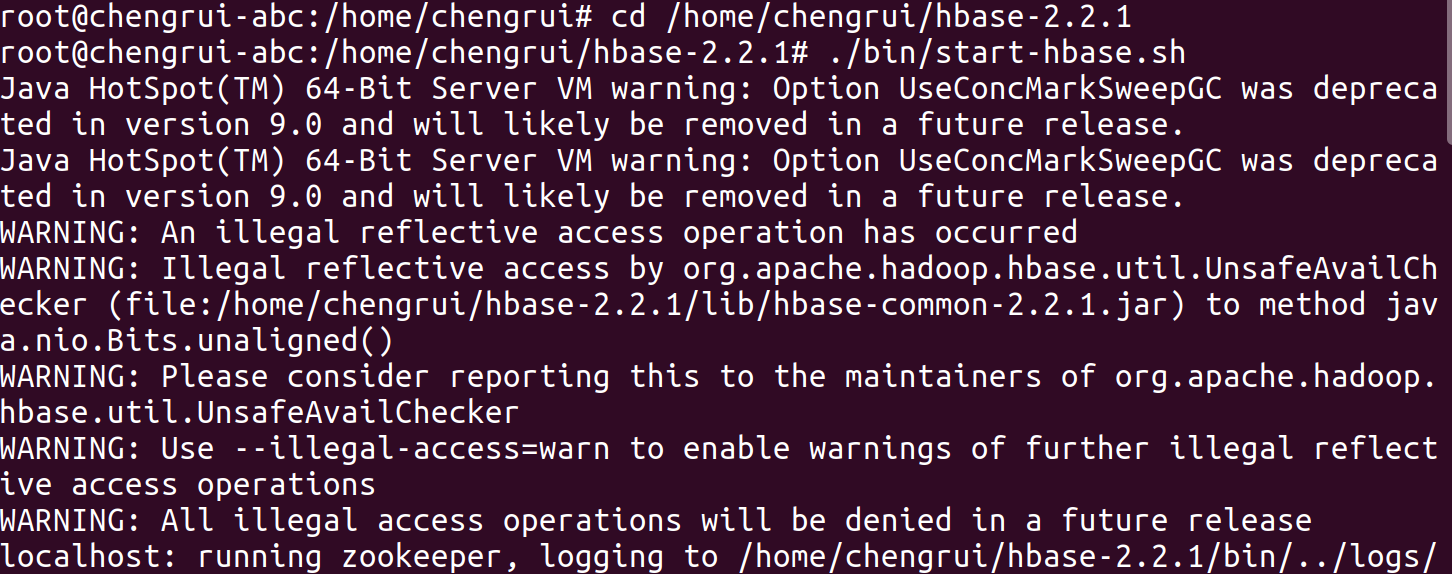
注意这里的export前的**#**一定要删除，否则后面启动Hbase时会报以下这个错：



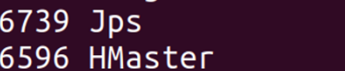
**3.**编辑**conf / hbase-site.xml**（这是Hbase的主要配置文件），configuration中加入下面这一段。这么做的目的是在本地文件系统上指定HBase和ZooKeeper写入数据的目录。



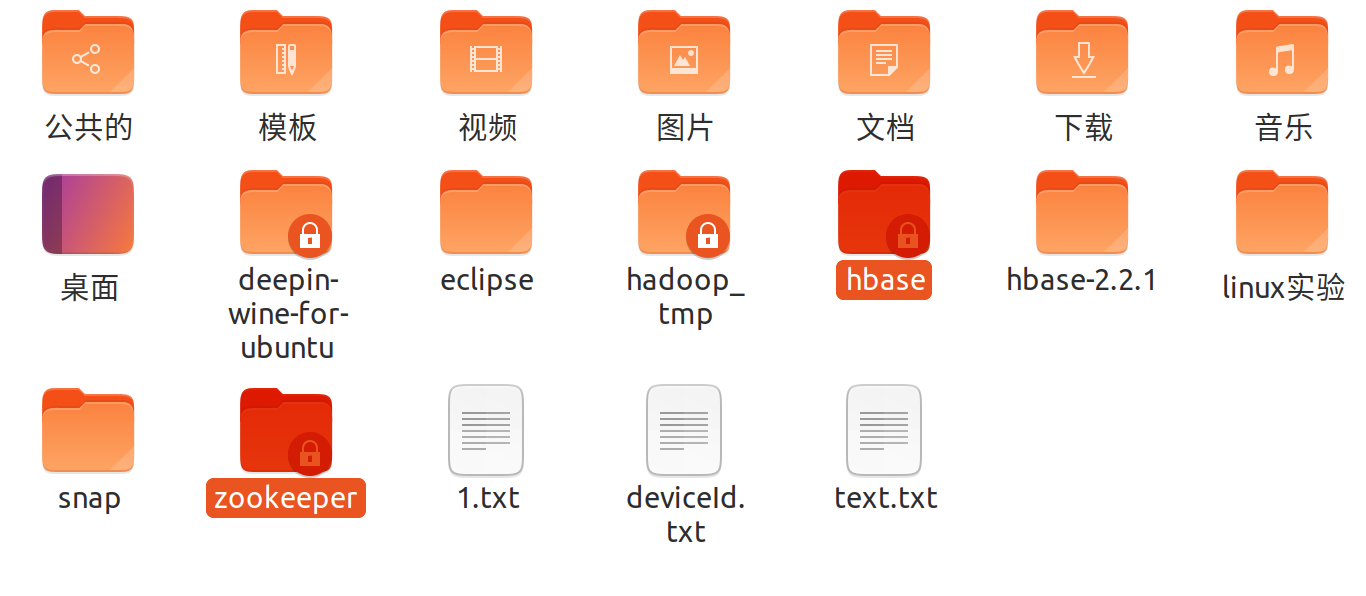
**4.**使用终端在hbase-2.2.1文件夹下，输入**./bin/start-hbase.sh**，开启hbase。



输入**Jps**，看看有没有出现hmaster



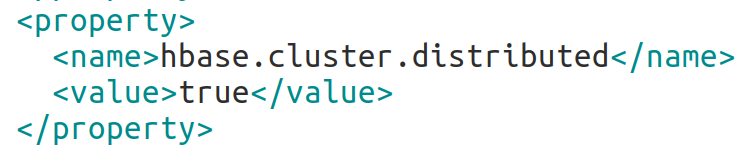
**5.**主目录下自动生成了这两个目录，他们是指定HBase和ZooKeeper写入数据的目录



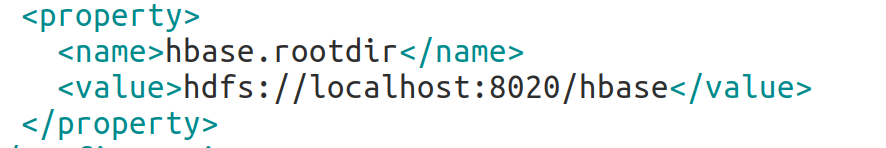
**接下来配置伪分布式：**

**1.**输入**./bin/stop-hbase.sh**停止hbase

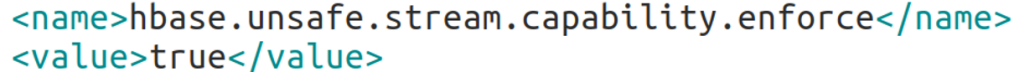
**2.**编辑**hbase-site.xml**配置。首先添加以下属性，该属性指示HBase以分布式模式运行



接下来，**hbase.rootdir**从本地文件系统更改为HDFS实例的地址。

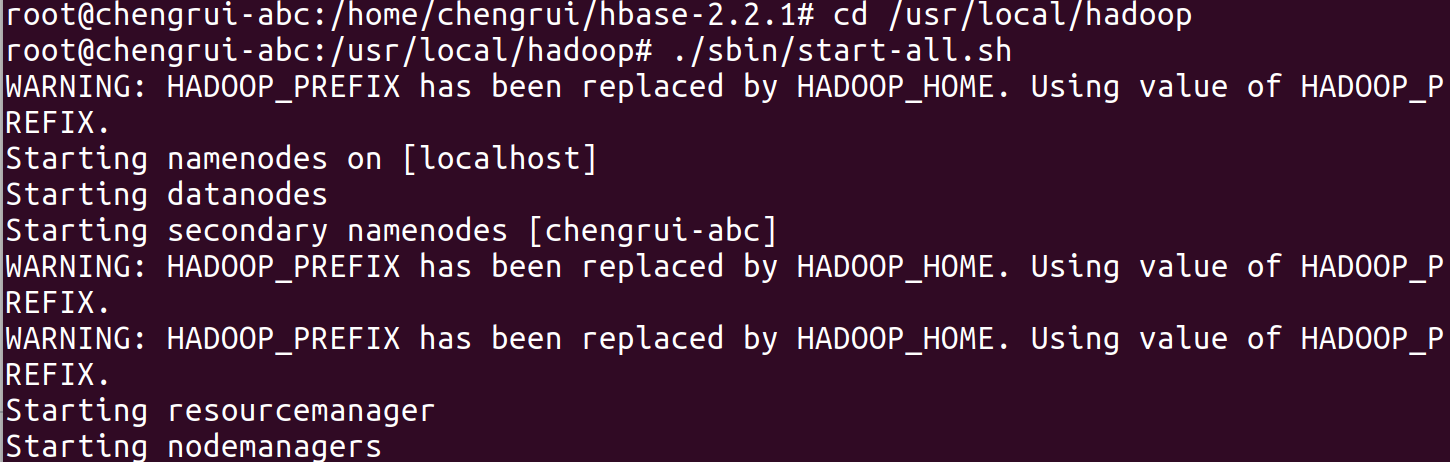


将**hbase.unsafe.stream.capability.enforce**设置为**true**。



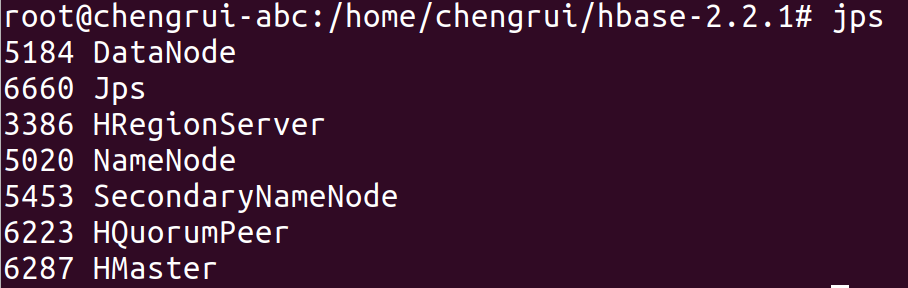
后面如果有遇到hmaster出现几秒后消失的问题，则把它改回false，反正我没遇到。

**3.**打开hadoop

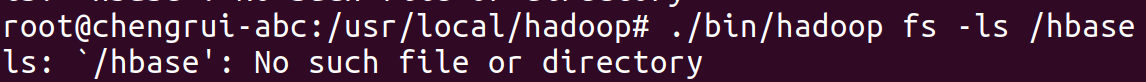


**4.**启动hbase，输入**jps**发现Hmaster没有出现。

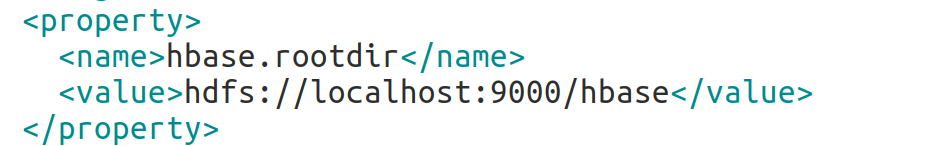
原因是没有删除原先的**rootdir**，导致rootdir重复，确保rootdir唯一后出现了：



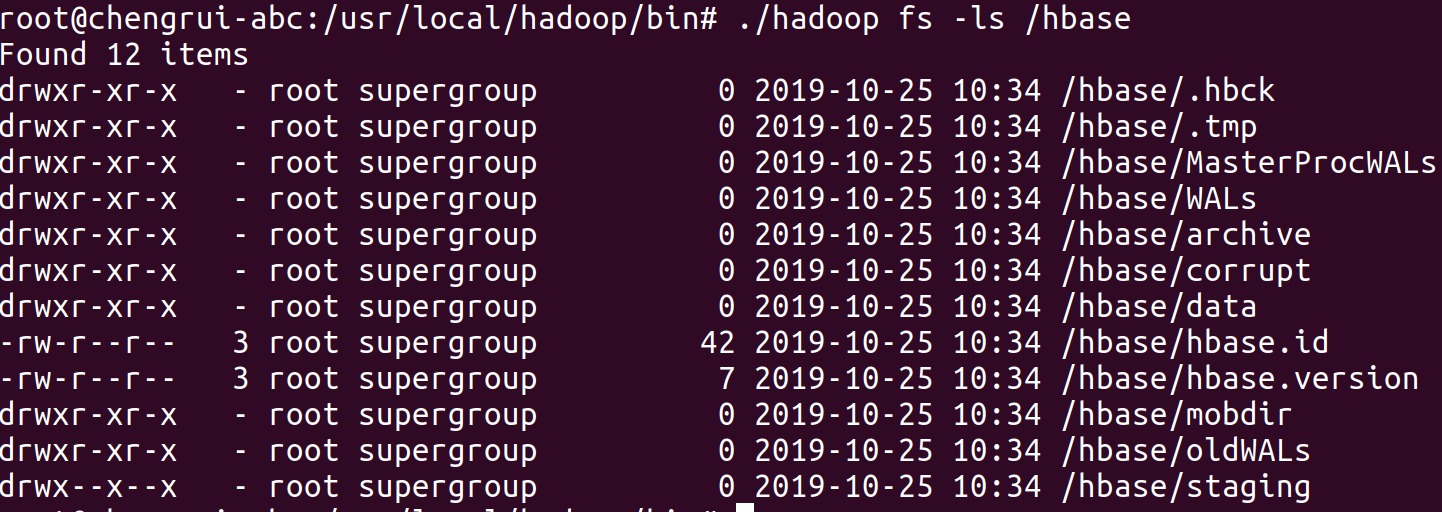
**5.**检查HDFS中的HBase目录。发现报错：



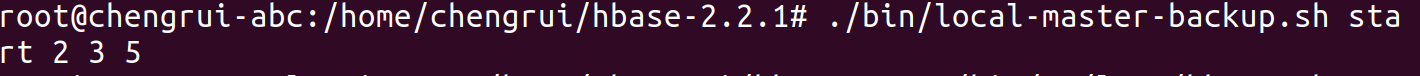
原因是8020没有改成HDFS在本地主机上的端口，我的是9000



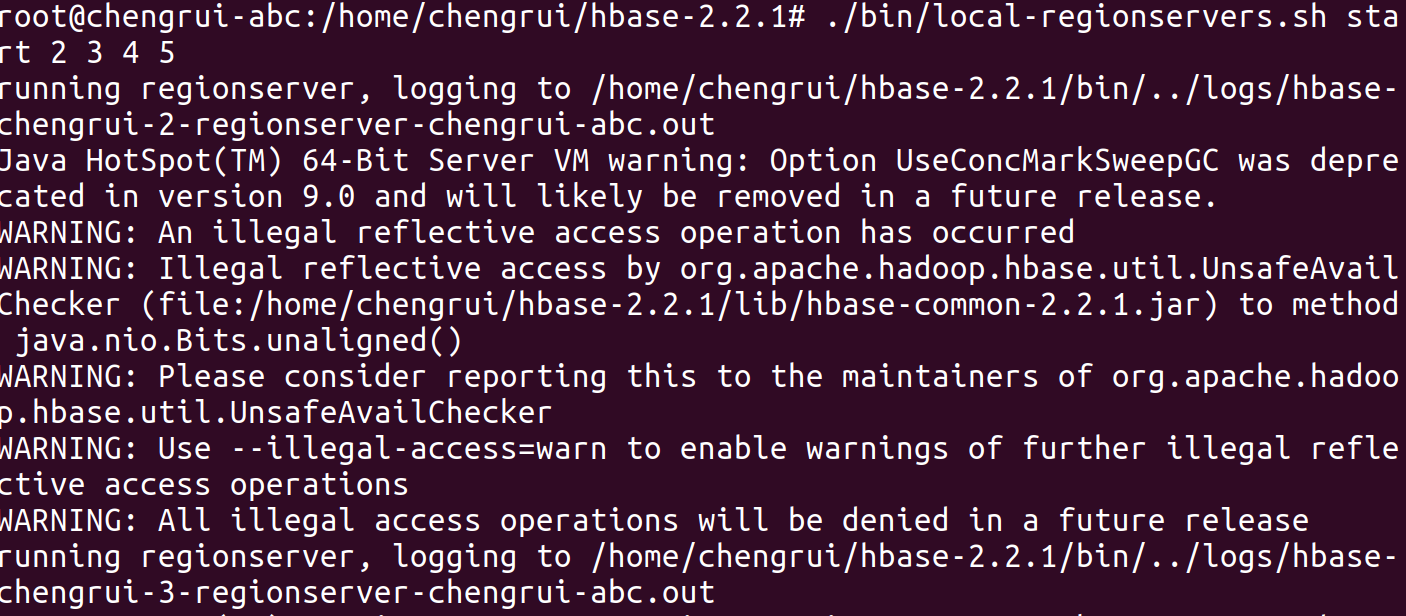
修改后再次查看：



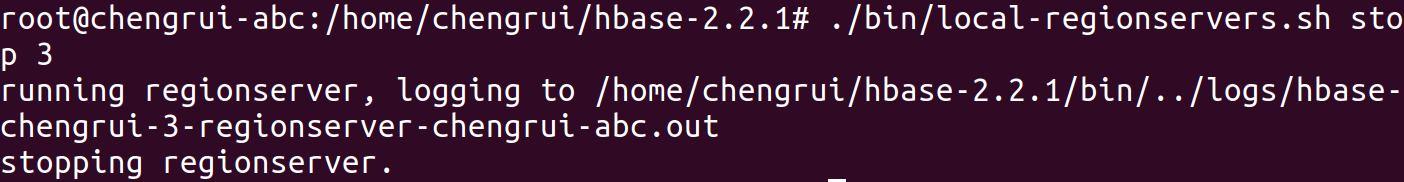
**6.**输入**./bin/local-master-backup.sh start 2 3 5，**启动3个备份服务器。



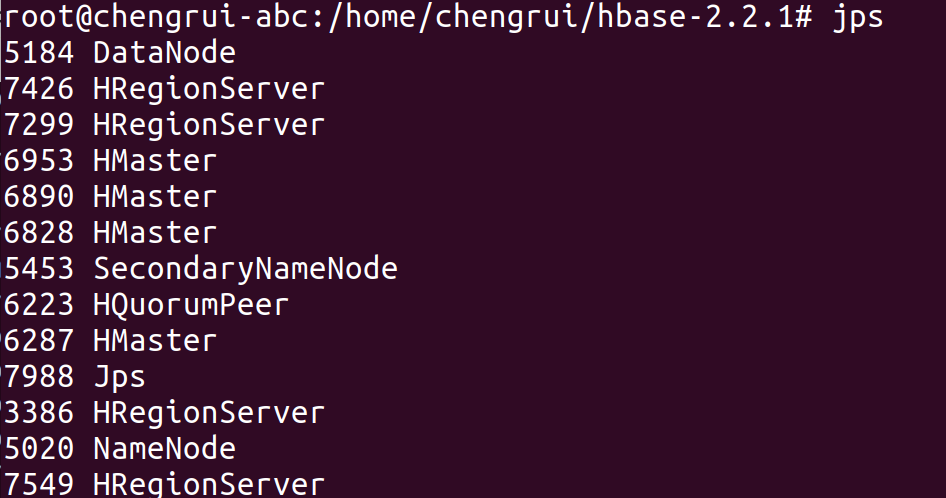
**7.**输入**./bin/local-regionservers.sh start 2 3 4 5**，启动四个附加的RegionServer。



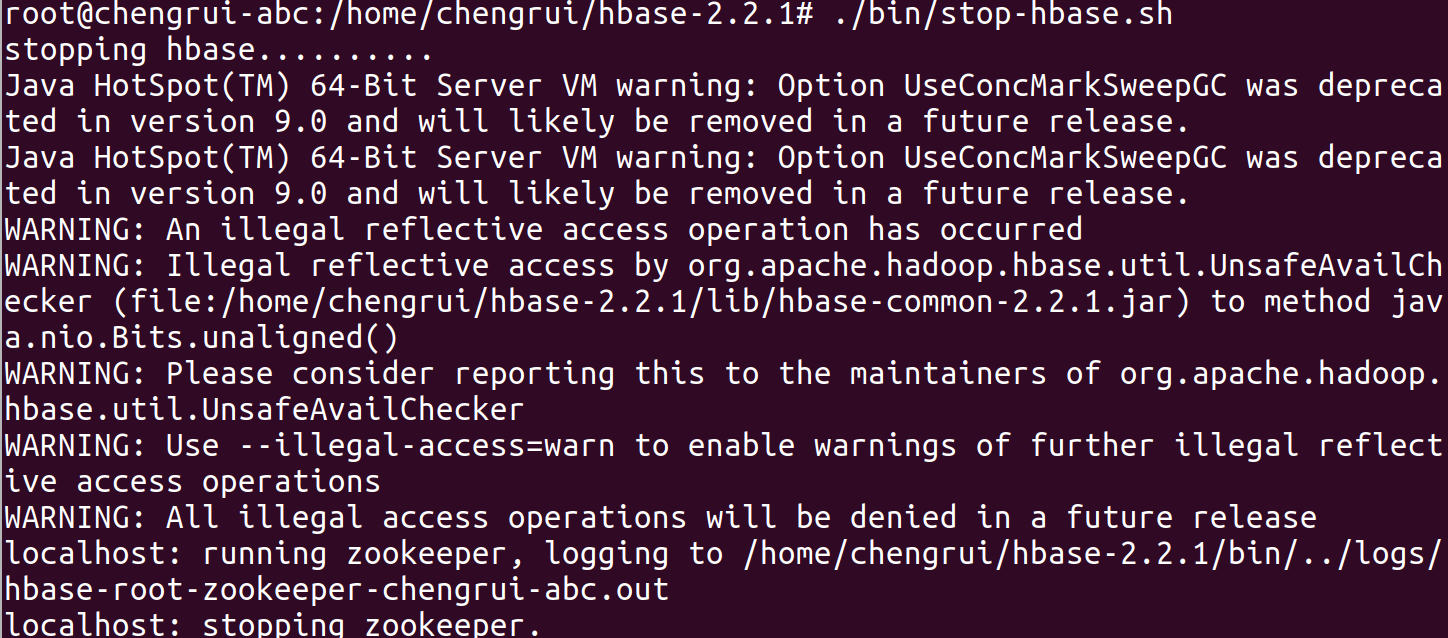
**8.**停止RegionServer，输入**./bin/local-regionservers.sh stop 3**命令停止3：



**9.**输入**jps，**查看是否开启成功



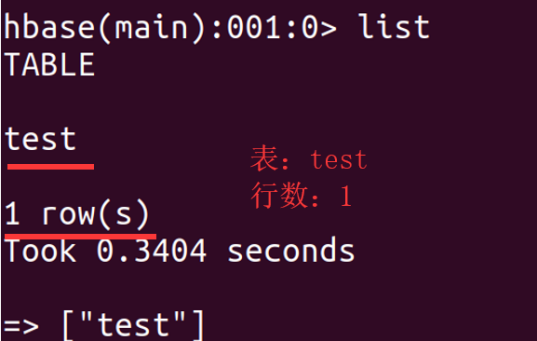
**10.**输入**./bin/start-hbase.sh，**关闭hbase



## 2.2 HBase Shell命令操作

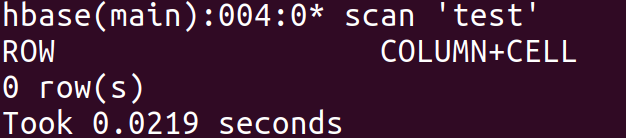
1. 编程实现以下指定功能，并用Hadoop提供的HBase Shell命令完成相同任务：
2. 列出HBase所有的表的相关信息，例如表名；

**list**



1. 在终端打印出指定的表的所有记录数据；

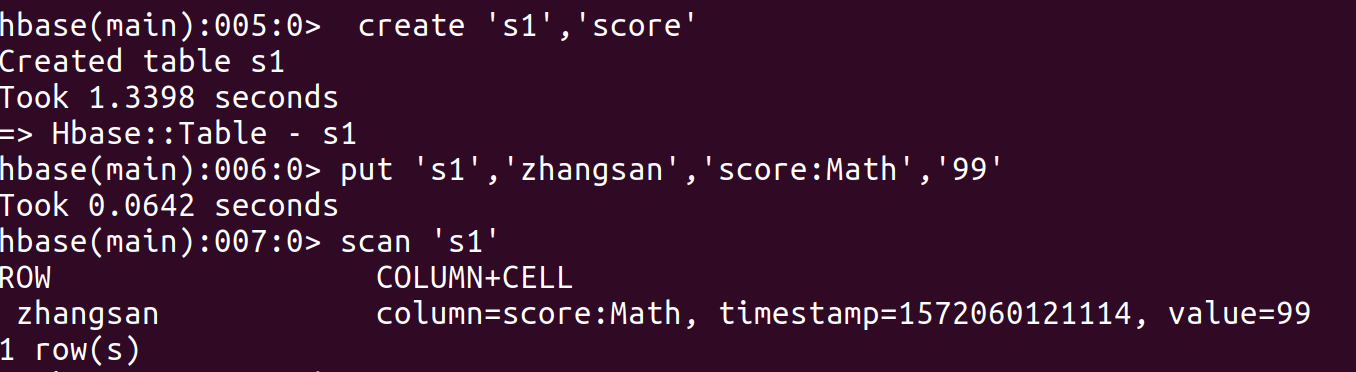
**scan ’表名’**



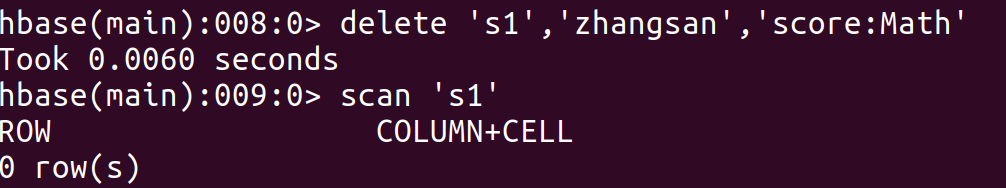
1. 向已经创建好的表添加和删除指定的列族或列；

**创建表：create ‘表名’ ‘列族名’**

**添加列：put ‘表名’, ‘行键’, ‘列名’, ‘值’**



**删除列：delete ‘表名’, ‘行键’, ‘列名’**

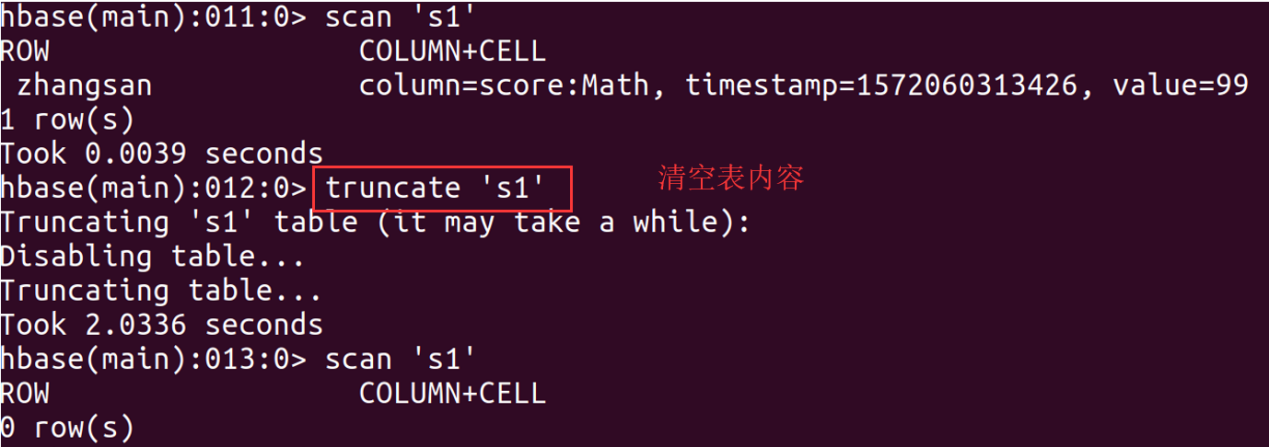


1. 清空指定的表的所有记录数据；

**scan**查看表，发现是有内容的

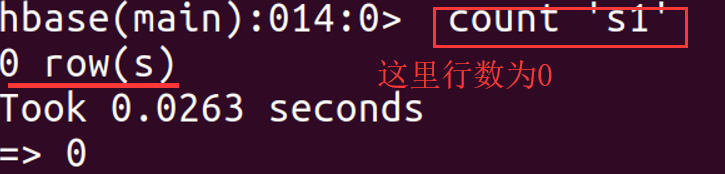
然后清空表的所有记录数据：**truncate ‘表名’**

**scan**再次查看，发现已经没有了数据



1. 统计表的行数。

**count ‘表名’**



（二）HBase数据库操作

1. 现有以下关系型数据库中的表和数据，要求将其转换为适合于HBase存储的表并插入数据：

学生表（Student）

|  |  |  |  |
| --- | --- | --- | --- |
| 学号（S\_No） | 姓名（S\_Name） | 性别（S\_Sex） | 年龄（S\_Age） |
| 2015001 | Zhangsan | male | 23 |
| 2015003 | Mary | female | 22 |
| 2015003 | Lisi | male | 24 |

课程表（Course）

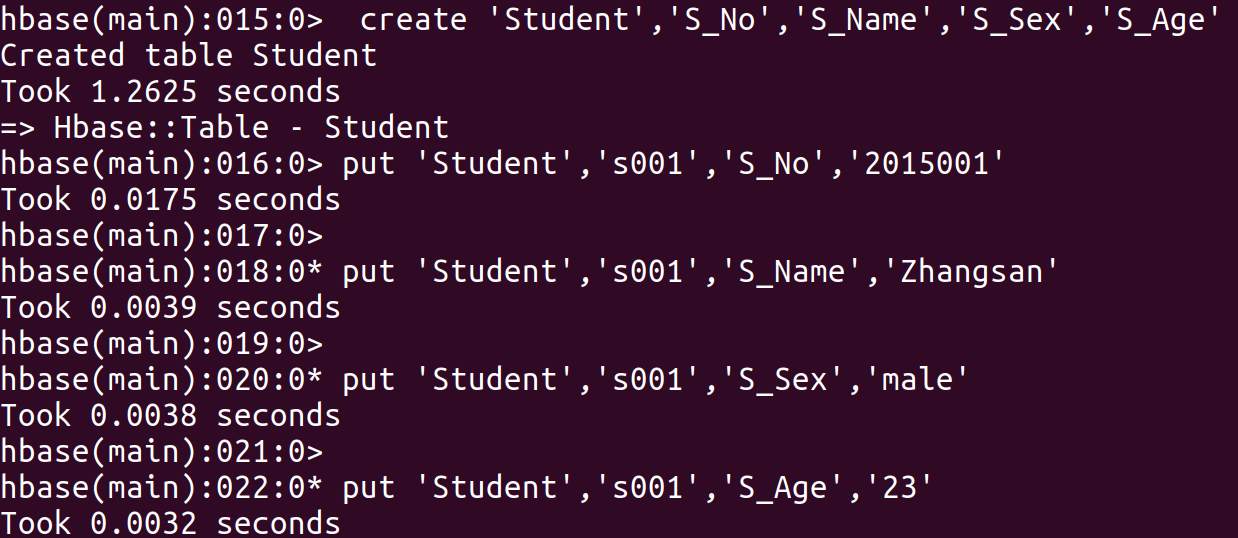
|  |  |  |
| --- | --- | --- |
| 课程号（C\_No） | 课程名（C\_Name） | 学分（C\_Credit） |
| 123001 | Math | 2.0 |
| 123002 | Computer Science | 5.0 |
| 123003 | English | 3.0 |

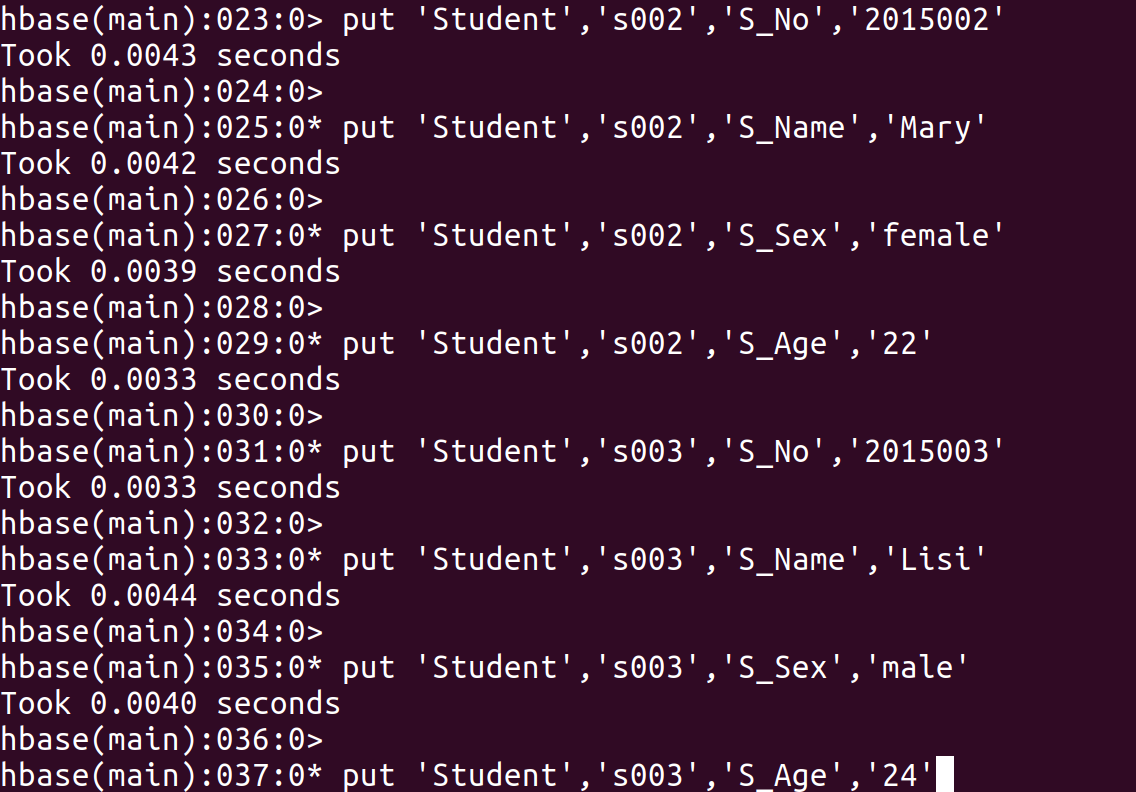
选课表（SC）

|  |  |  |
| --- | --- | --- |
| 学号（SC\_Sno） | 课程号（SC\_Cno） | 成绩（SC\_Score） |
| 2015001 | 123001 | 86 |
| 2015001 | 123003 | 69 |
| 2015002 | 123002 | 77 |
| 2015002 | 123003 | 99 |
| 2015003 | 123001 | 98 |
| 2015003 | 123002 | 95 |

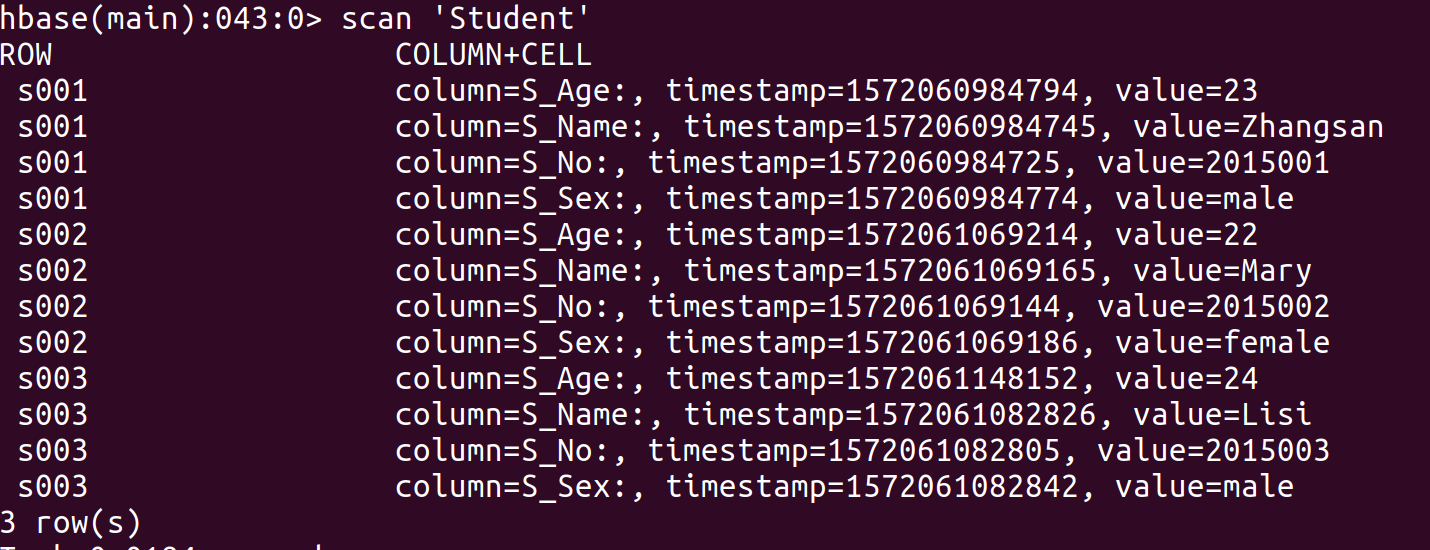
先用**create**创建，再用**put**插入数据，具体代码如下：

**Student表：**

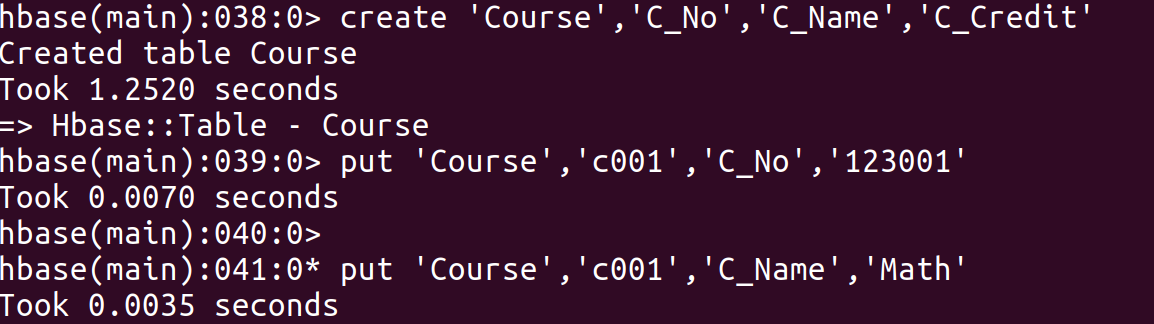


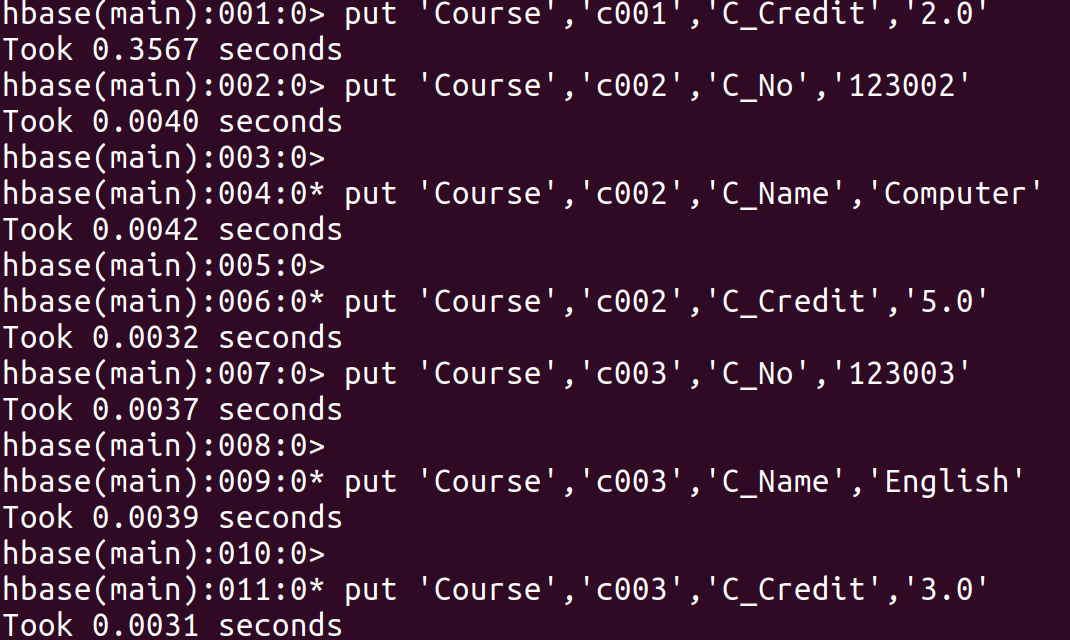


查看结果：**scan ‘Student’**

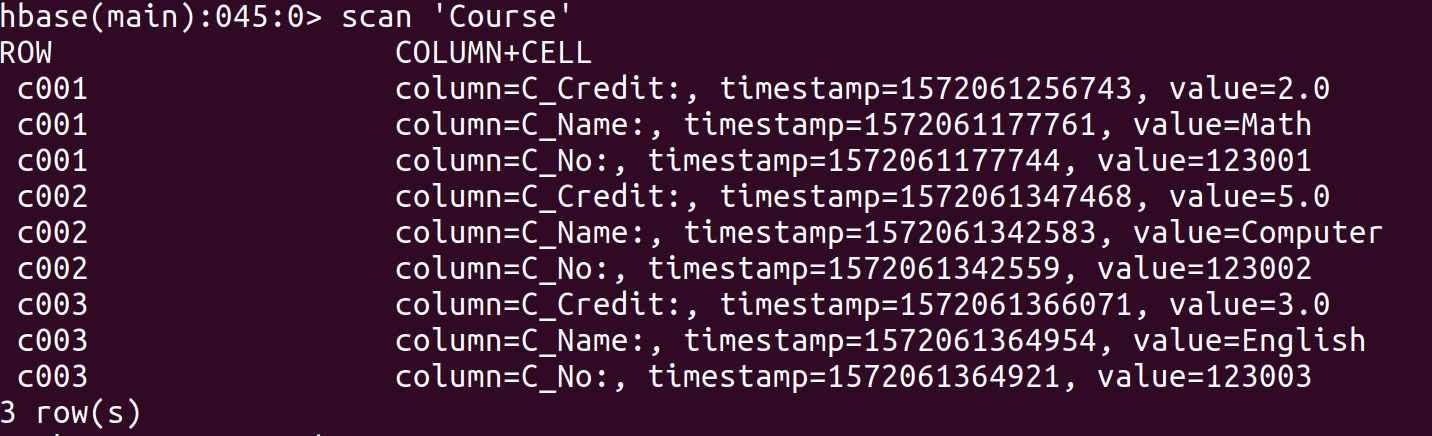


**Course表**

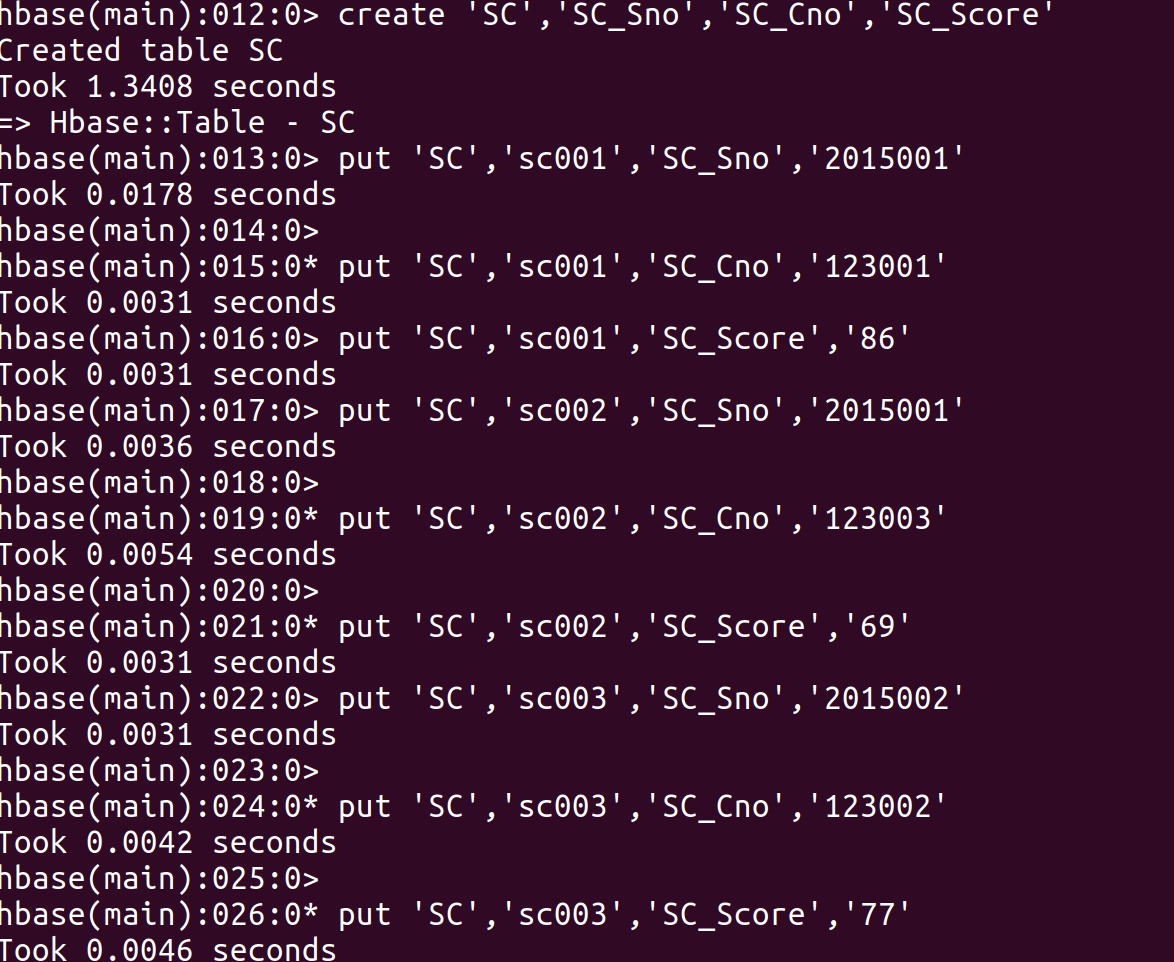


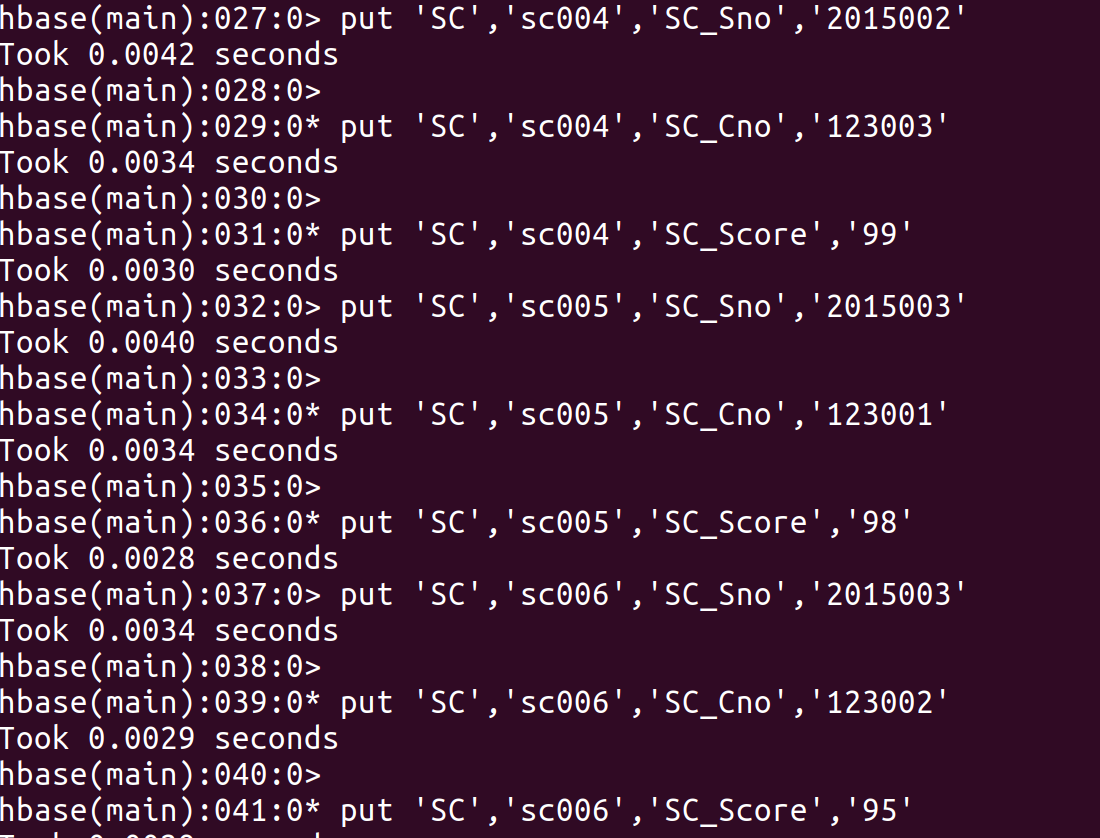


查看结果：**scan ‘Course’**

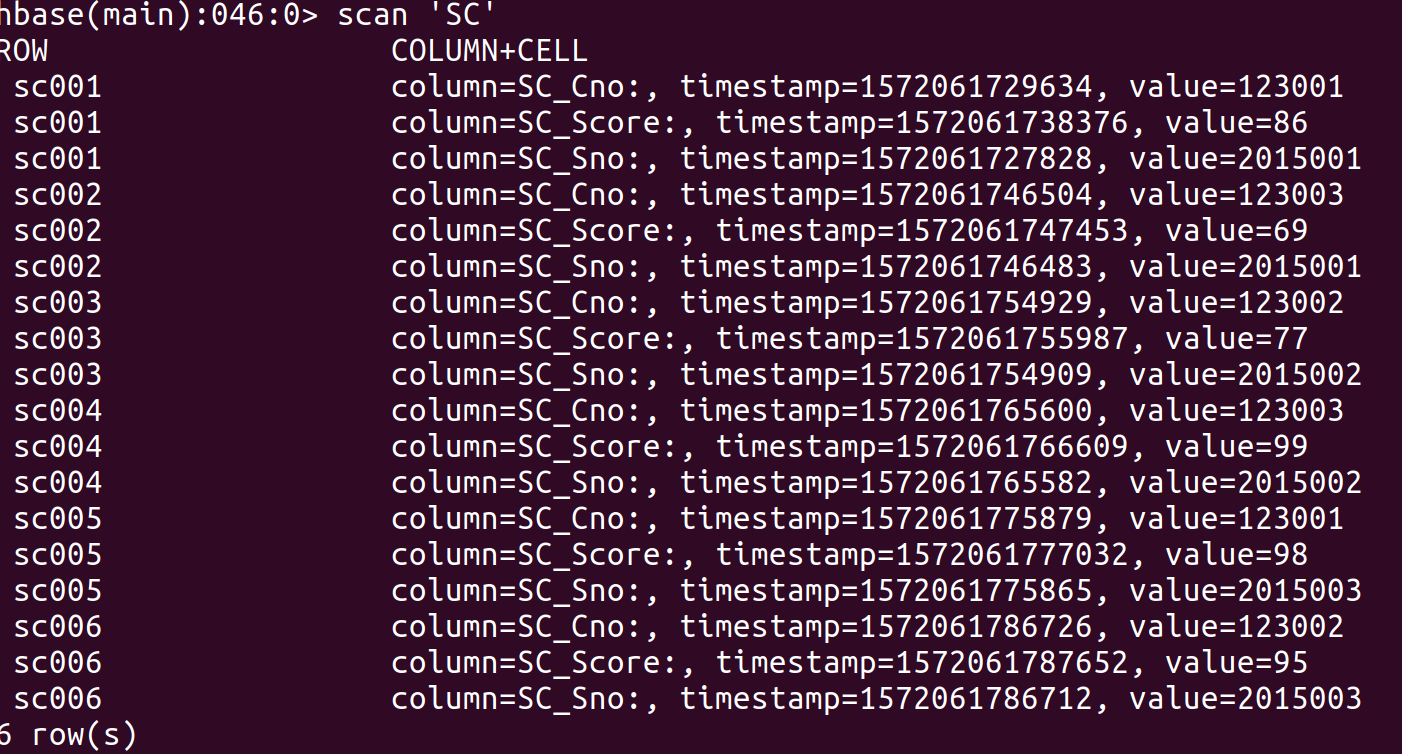


**SC表：**





查看结果：**scan ‘SC’**



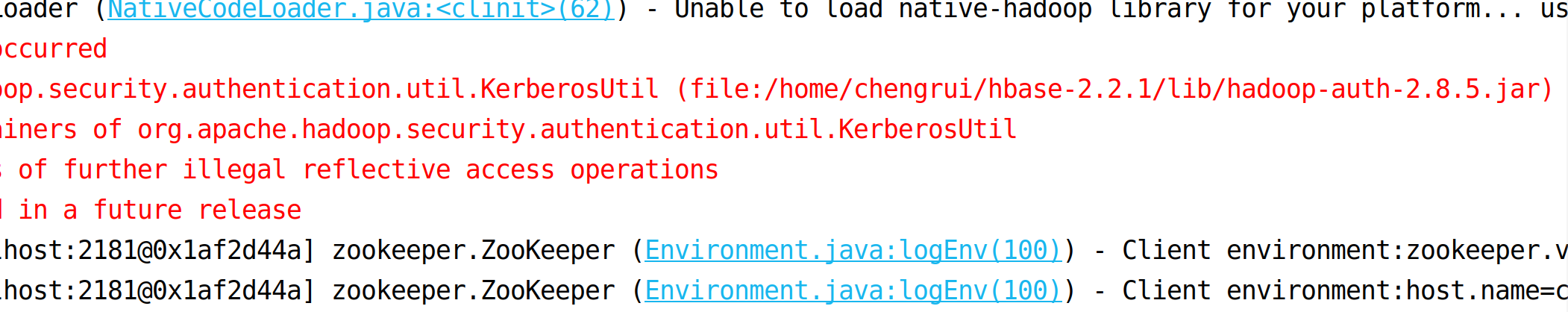
## 2.3 编程实现相应的功能

（1）createTable(String tableName, String[] fields)

创建表，参数tableName为表的名称，字符串数组fields为存储记录各个字段名称的数组。要求当HBase已经存在名为tableName的表的时候，先删除原有的表，然后再创建新的表。

为了在eclipse下运行，先导入hbase-2.2.1目录下的lib下和lib子目录下所有包。

报错



我怀疑是Hadoop包的版本问题，于是我删除名字中带有hadoop的包，然后导入自己版本hadoop的包，就可以运行了。（林子雨老师的视频说要导入配置文件，但我没导入发现也可以运行，可能是我导入的包比较全的原因吧，这里不做深究。）

**代码：**

package hbase;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.hbase.\*;

import org.apache.hadoop.hbase.client.\*;

public class Test1 {

public static Configuration configuration;

public static Connection connection;

public static Admin admin;

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

String fields[]= {"ziduan1","ziduan2","ziduan3"};

createTable("test1",fields);

}

public static void init(){

configuration = HBaseConfiguration.create();

configuration.set("hbase.rootdir","hdfs://localhost:9000/hbase");

try{

connection = ConnectionFactory.createConnection(configuration);

admin = connection.getAdmin();

}catch (IOException e){

e.printStackTrace();

}

}

//关闭连接

public static void close(){

try{

if(admin != null){

admin.close();

}

if(null != connection){

connection.close();

}

}catch (IOException e){

e.printStackTrace();

}

}

public static void createTable(String tableName,String[] fields) throws IOException {

init();

TableName tablename = TableName.valueOf(tableName);

if(admin.tableExists(tablename)){

System.out.println("表已存在，正在删除");

admin.disableTable(tablename);

admin.deleteTable(tablename);//删除原来的表

System.out.println("删除成功");

}

HTableDescriptor hTableDescriptor= new HTableDescriptor(tablename);

for(String str:fields){

HColumnDescriptor hColumnDescriptor = new HColumnDescriptor(str);

hTableDescriptor.addFamily(hColumnDescriptor);

}

admin.createTable(hTableDescriptor);

System.out.println("表创建成功");

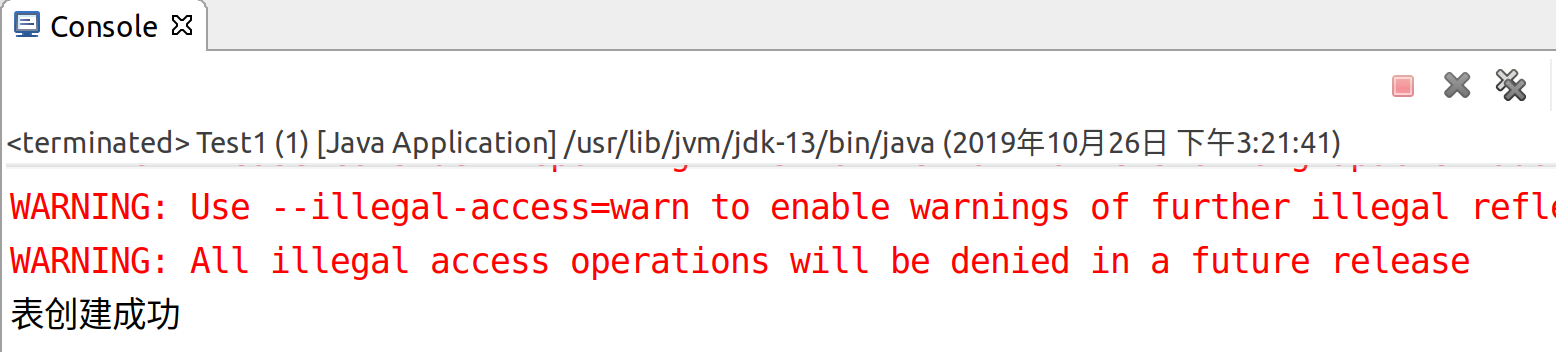
close();

}

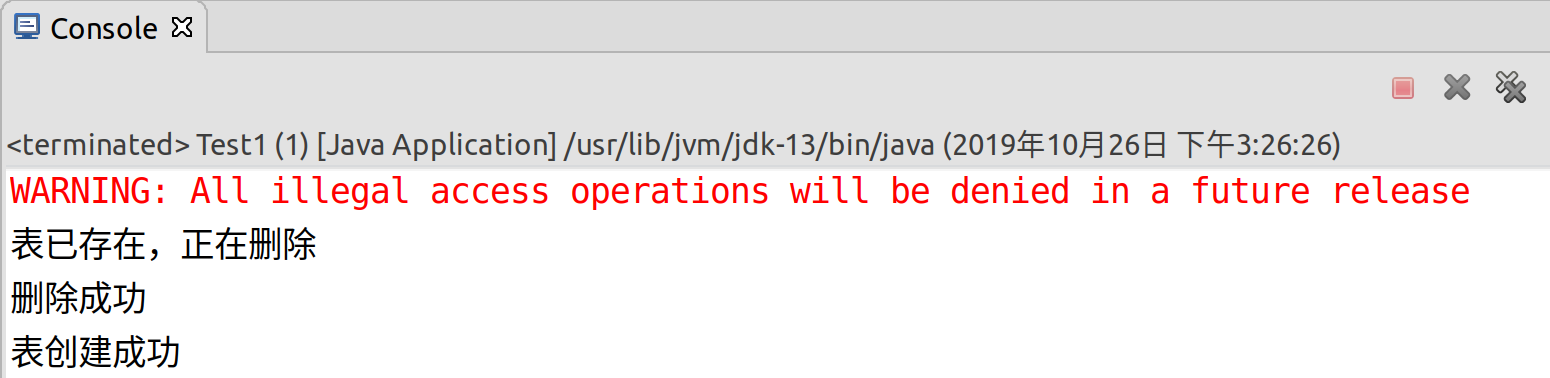
}

**运行结果**

表不存在时：



表已存在时:

（2）addRecord(String tableName, String row, String[] fields, String[] values)

向表tableName、行row（用S\_Name表示）和字符串数组fields指定的单元格中添加对应的数据values。其中，fields中每个元素如果对应的列族下还有相应的列限定符的话，用“columnFamily:column”表示。例如，同时向“Math”、“Computer Science”、“English”三列添加成绩时，字符串数组fields为{“Score:Math”, ”Score:Computer Science”, ”Score:English”}，数组values存储这三门课的成绩。

**代码：**

package hbase;

import java.io.IOException;

import java.util.Scanner;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.hbase.\*;

import org.apache.hadoop.hbase.client.\*;

public class Test2 {

public static Configuration configuration;

public static Connection connection;

public static Admin admin;

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

String fields[]= {"Score:Math", "Score:Computer Science", "Score:English"};

System.out.println("输入姓名，数学成绩，电脑成绩，英语成绩");

Scanner sc=new Scanner(System.in);

String name = sc.next();

String values[]= {sc.next(), sc.next(), sc.next()};

addRecord("test2",name,fields,values);

}

public static void init(){

configuration = HBaseConfiguration.create();

configuration.set("hbase.rootdir","hdfs://localhost:9000/hbase");

try{

connection = ConnectionFactory.createConnection(configuration);

admin = connection.getAdmin();

}catch (IOException e){

e.printStackTrace();

}

}

//关闭连接

public static void close(){

try{

if(admin != null){

admin.close();

}

if(null != connection){

connection.close();

}

}catch (IOException e){

e.printStackTrace();

}

}

public static void addRecord(String tableName,String row,String[] fields,String[] values) throws

IOException {

init();

Table table = connection.getTable(TableName.valueOf(tableName));

for(int i = 0;i < fields.length;i++){

Put put = new Put(row.getBytes());

String[] cols = fields[i].split(":");//列族，列名

put.addColumn(cols[0].getBytes(), cols[1].getBytes(), values[i].getBytes());

table.put(put);

System.out.println("添加成功");

}

table.close();

close();

}

}

**运行结果：**



（3）scanColumn(String tableName, String column)

浏览表tableName某一列的数据，如果某一行记录中该列数据不存在，则返回null。要求当参数column为某一列族名称时，如果底下有若干个列限定符，则要列出每个列限定符代表的列的数据；当参数column为某一列具体名称（例如“Score:Math”）时，只需要列出该列的数据。

**代码：**

package hbase;

import java.io.IOException;

import java.util.Scanner;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.hbase.\*;

import org.apache.hadoop.hbase.client.\*;

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

public class Test3 {

public static Configuration configuration;

public static Connection connection;

public static Admin admin;

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

Scanner sc=new Scanner(System.in);

System.out.println("请输入表名和列名：");

String tablename =sc.next();

String columname =sc.next();

scanColumn(tablename,columname);

}

public static void init(){

configuration = HBaseConfiguration.create();

configuration.set("hbase.rootdir","hdfs://localhost:9000/hbase");

try{

connection = ConnectionFactory.createConnection(configuration);

admin = connection.getAdmin();

}catch (IOException e){

e.printStackTrace();

}

}

//关闭连接

public static void close(){

try{

if(admin != null){

admin.close();

}

if(connection != null){

connection.close();

}

}catch (IOException e){

e.printStackTrace();

}

}

public static void scanColumn(String tableName,String column)throws IOException{

init();

Table table = connection.getTable(TableName.valueOf(tableName));//获取相关表信息

try {

Scan scan = new Scan();

if (column.lastIndexOf(":") == -1) {//列族

scan.addFamily(Bytes.toBytes(column));//指定需要的family

} else {

String[] cols = column.split(":");//列族，列名

scan.addColumn(cols[0].getBytes(), cols[1].getBytes());

}

ResultScanner scanner = table.getScanner(scan);

for (Result result = scanner.next(); result != null; result = scanner.next()){

showCell(result);

}

}catch (Exception e) {

System.out.println("null");

}

table.close();

close();

}

//输出

public static void showCell(Result result){

Cell[] cells = result.rawCells();

for(Cell cell:cells){

System.out.println("RowName:"+new String(CellUtil.cloneRow(cell))+" ");

System.out.println("Timetamp:"+cell.getTimestamp()+" ");

System.out.println("column Family:"+new String(CellUtil.cloneFamily(cell))+" ");

System.out.println("row Name:"+new String(CellUtil.cloneQualifier(cell))+" ");

System.out.println("value:"+new String(CellUtil.cloneValue(cell))+" ");

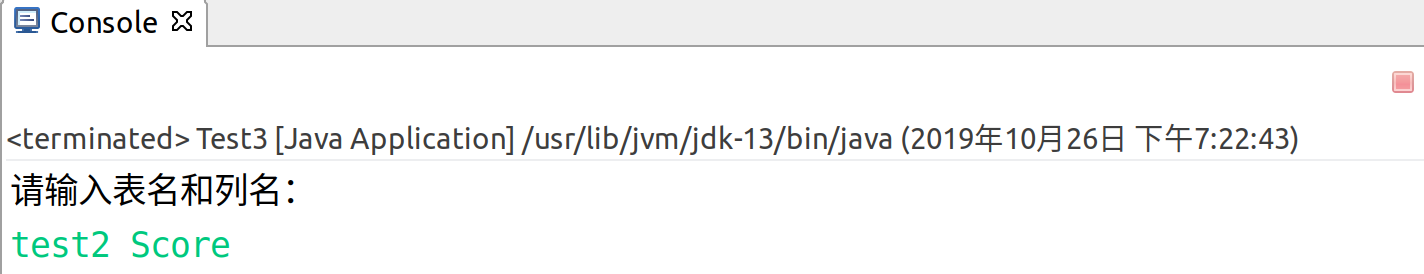
}

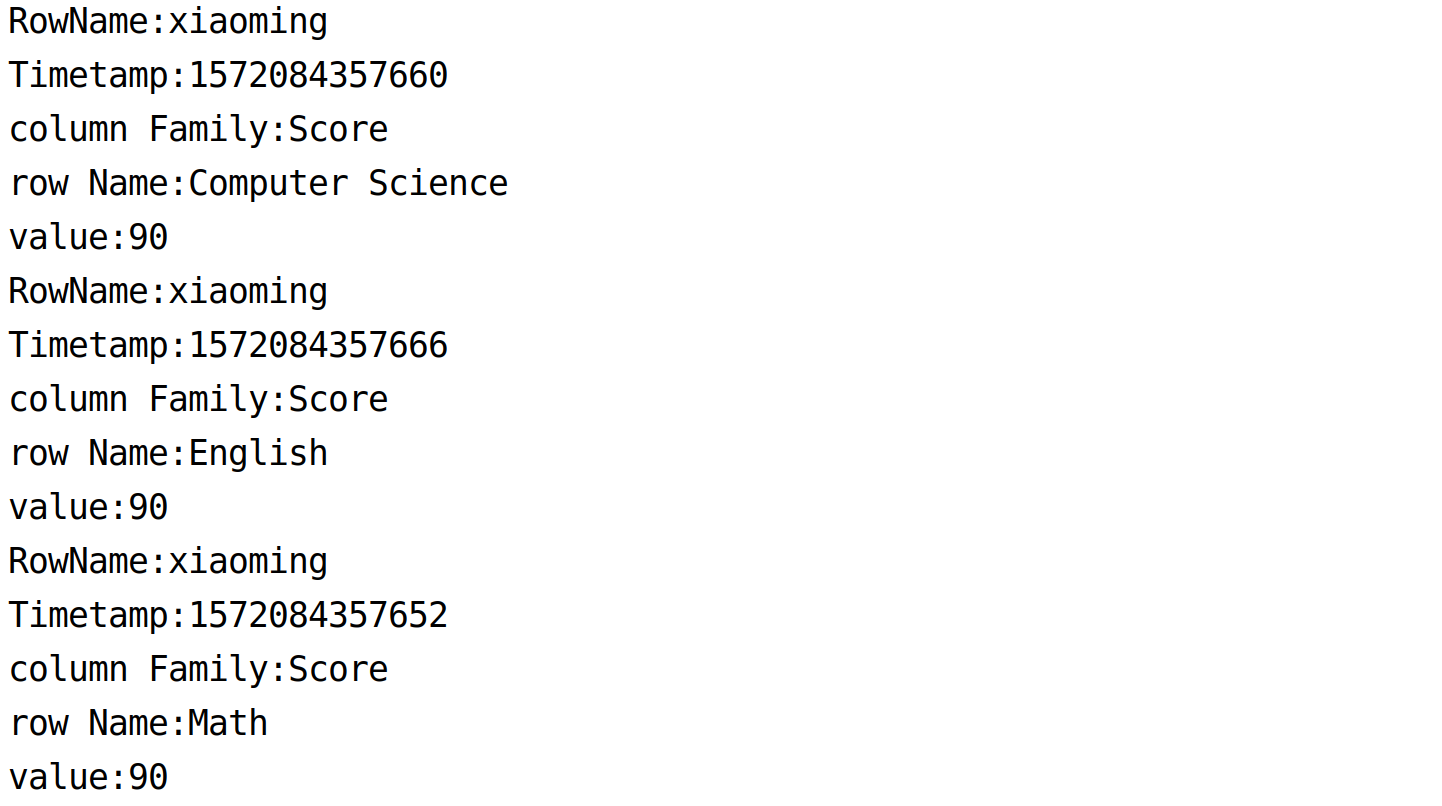
}

}

**运行结果：**

**指定列族：**





**指定列：**



（4）modifyData(String tableName, String row, String column)

修改表tableName，行row（可以用学生姓名S\_Name表示），列column指定的单元格的数据。

**代码：**

package hbase;

import java.io.IOException;

import java.util.Scanner;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.hbase.\*;

import org.apache.hadoop.hbase.client.\*;

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

public class Test4 {

public static Configuration configuration;

public static Connection connection;

public static Admin admin;

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

Scanner sc=new Scanner(System.in);

System.out.println("请输入表名,行键，列名，新值：");

String tableName =sc.next();

String rowname =sc.next();

String columname =sc.next();

String val =sc.next();

modifyData(tableName,rowname,columname,val);

}

public static void init(){

configuration = HBaseConfiguration.create();

configuration.set("hbase.rootdir","hdfs://localhost:9000/hbase");

try{

connection = ConnectionFactory.createConnection(configuration);

admin = connection.getAdmin();

}catch (IOException e){

e.printStackTrace();

}

}

public static void close(){

try{

if(admin != null){

admin.close();

}

if(connection != null){

connection.close();

}

}catch (IOException e){

e.printStackTrace();

}

}

public static void modifyData(String tableName,String row,String column,String val)throws

IOException{

init();

Table table = connection.getTable(TableName.valueOf(tableName));

Put put = new Put(row.getBytes());

if (column.lastIndexOf(":") != -1) {//列族

String[] cols = column.split(":");//列族，列名

put.addColumn(cols[0].getBytes(),cols[1].getBytes(),val.getBytes());

}

else put.addColumn(column.getBytes(),null,val.getBytes());

System.out.println("新值已修改为："+val);

table.put(put);

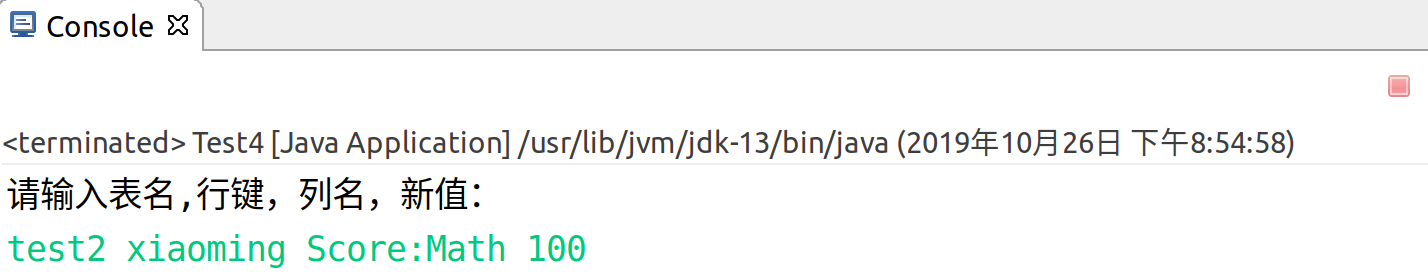
table.close();

close();

}

}

**运行结果：**



（5）deleteRow(String tableName, String row)

删除表tableName中row指定的行的记录。

**代码：**

package hbase;

import java.io.IOException;

import java.util.Scanner;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.hbase.\*;

import org.apache.hadoop.hbase.client.\*;

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

public class Test5 {

public static Configuration configuration;

public static Connection connection;

public static Admin admin;

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

Scanner sc=new Scanner(System.in);

System.out.println("请输入表名,行键：");

String tableName =sc.next();

String rowname =sc.next();

deleteRow(tableName,rowname);

}

public static void init(){

configuration = HBaseConfiguration.create();

configuration.set("hbase.rootdir","hdfs://localhost:9000/hbase");

try{

connection = ConnectionFactory.createConnection(configuration);

admin = connection.getAdmin();

}catch (IOException e){

e.printStackTrace();

}

}

public static void close(){

try{

if(admin != null){

admin.close();

}

if(connection != null){

connection.close();

}

}catch (IOException e){

e.printStackTrace();

}

}

public static void deleteRow(String tableName,String row)throws IOException{

init();

Table table = connection.getTable(TableName.valueOf(tableName));

Delete delete = new Delete(row.getBytes());

table.delete(delete);

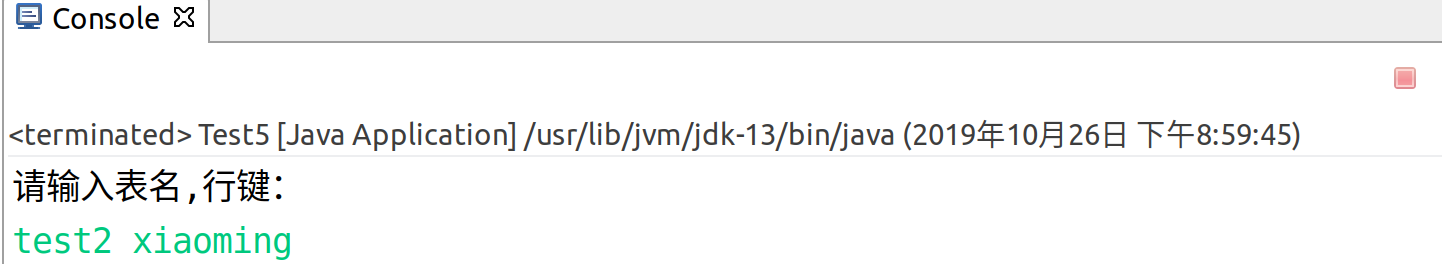
table.close();

close();

}

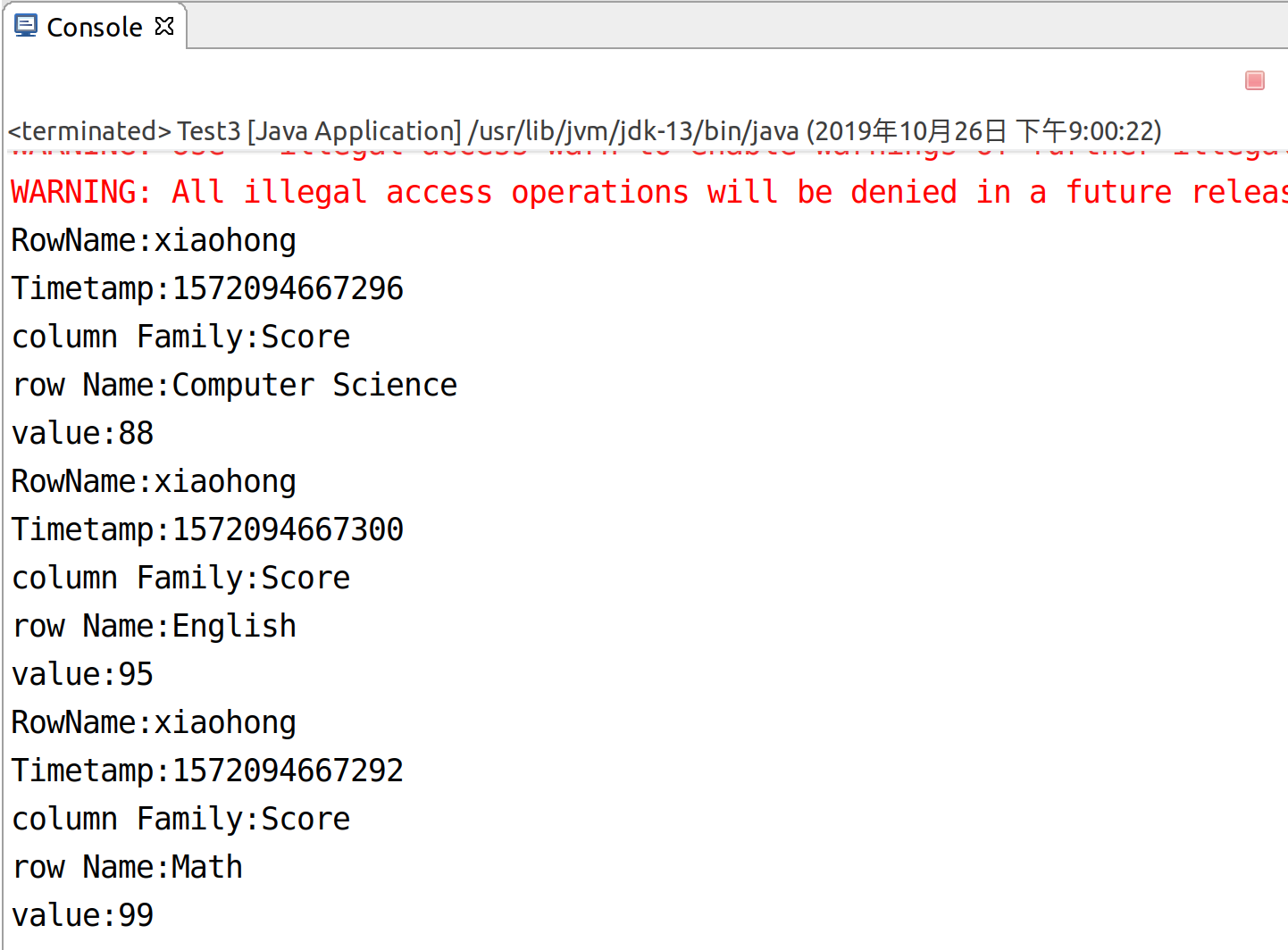
}

**运行结果：**



使用第三题的代码查看效果

发现只有小红的信息了，小明已被删除



# 3. 实验总结

## 3.1 安装部分

我的安装过程主要是按照老师给的官方教程，但因为一些原因比如：粗心和一时疏忽，遇到了一些问题：

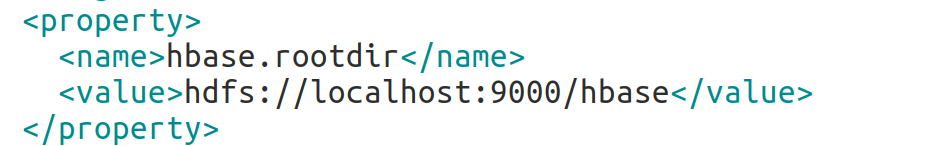
1．在编辑conf / hbase-env.sh中的JAVA\_HOME时没把export前的#去掉，导致后面在开启hbase时报错：



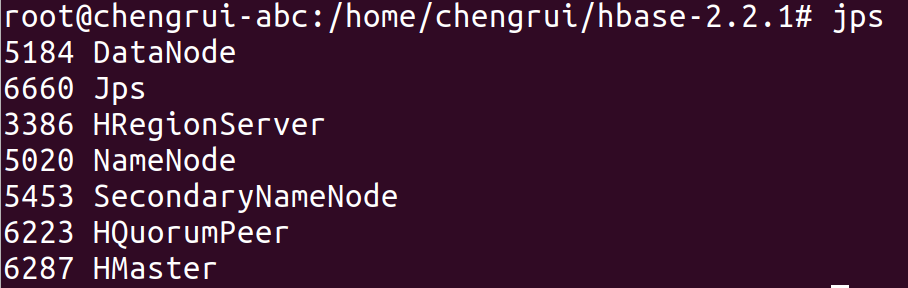
**解决方法：**去掉#就好了。

2. 配置伪分布式时，启动hbase，输入jps发现Hmaster没有出现。

原因是：配置**conf / hbase-site.xml时**没有删除原先的**rootdir**，导致rootdir重复

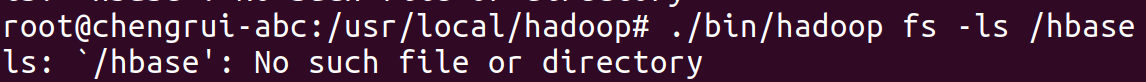
**解决方法**：确保rootdir唯一后出现了，就是删除当初配的本地路径，只留下

修改完再重启，就出现了，有时候修改完配置文件后重启：



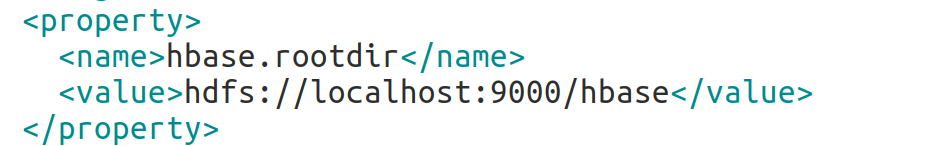
3. 有时候修改完配置文件后重启hbase没生效，只要重启ubuntu就行。

4. 检查HDFS中的HBase目录。发现报错：

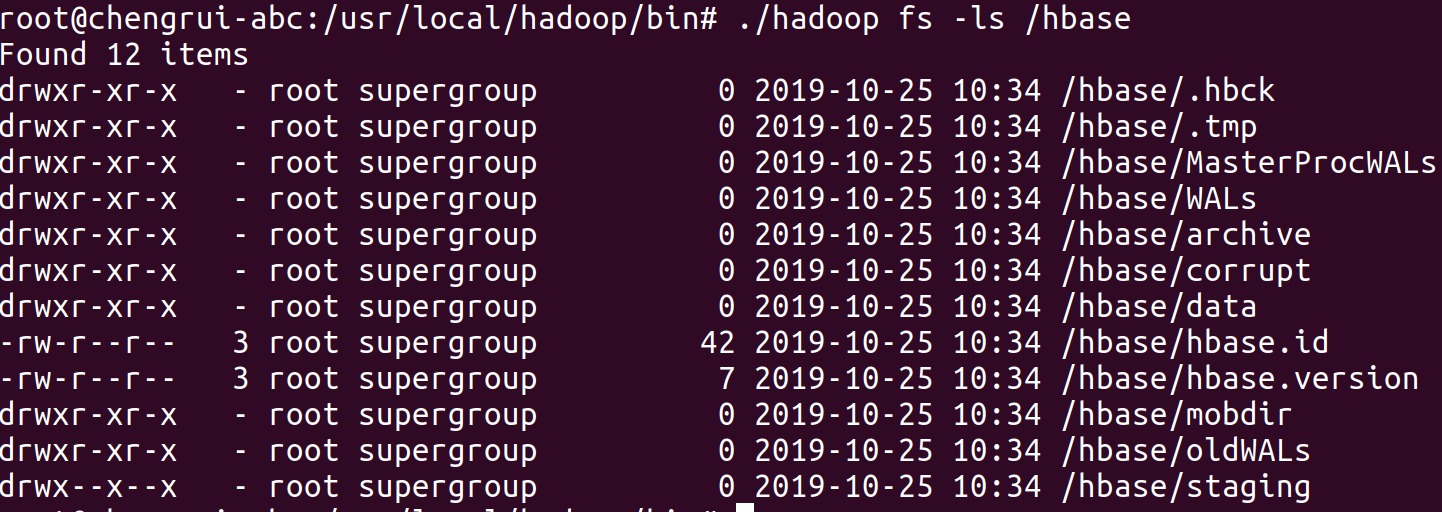


原因是：配置**conf / hbase-site.xml的rootdir**时，8020没有改成HDFS在本地主机上的端口，我的是9000。

**解决方式：**改成HDFS在本地主机上的端口



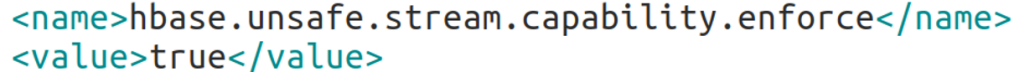
修改后再次查看：



5. 我的安装过程相对比较顺利，当时在帮助同学的时候也遇到了一些不同的问题。

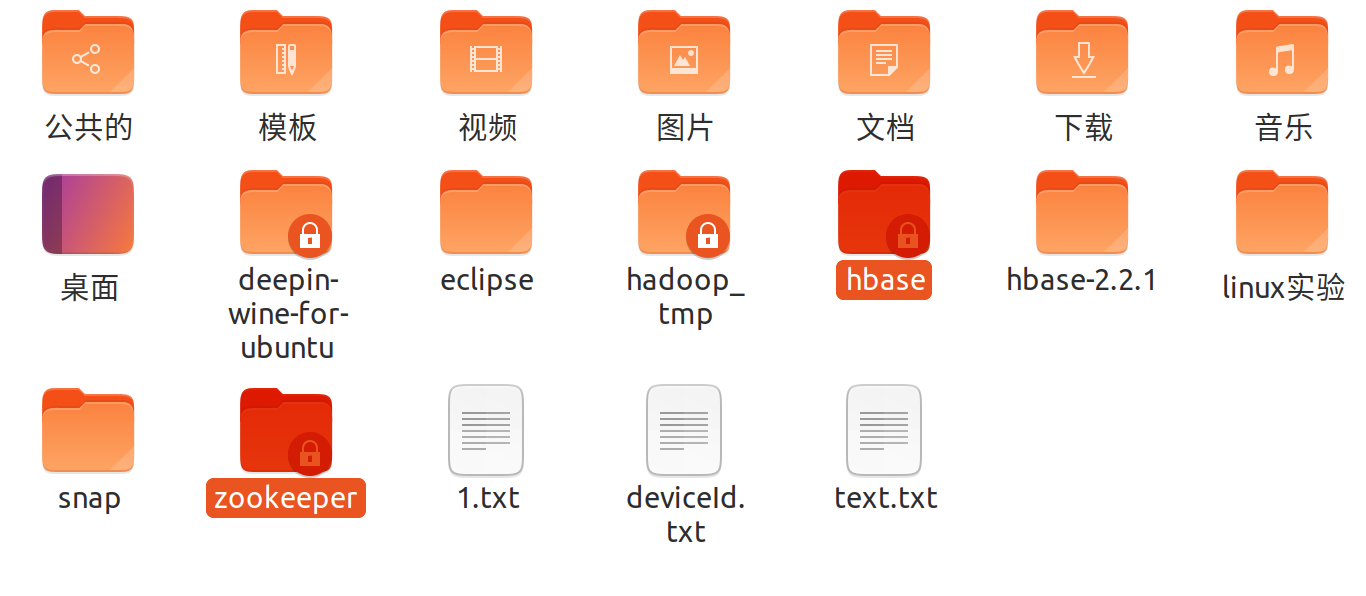
5.1 如果有遇到hmaster出现几秒后消失的问题

**解决方式：**



把值改回false。

5.2. 有的同学配置完本机的hbase后，第一次运行时，没有出现下面的hbase目录



原因是：把hbase-2.2.1改名成hbase了，导致生成保存数据的hbase目录时冲突了，所以建议解压后不要把hbase-2.2.1改名成hbase。

5.3 在执行hbase的shell命令时，如果什么命令都报hmaster的错时，先确定hmaster是否存在，把前面提到的true改为false后如果还没出现hmaster，可以试着先启动hadoop，再启动hbase，我帮同学亲测有效。

5.4 如果其他命令比如list，version可以执行，而就是建不了表，网上有说是节点时间问题，重新设置时间就行，但是我试过不行。我感觉是因为重装hbase的时候，没卸载干净的原因，于是把本地的zookeeper目录删除，重启生成一个新的zookeeper后，发现就成功了。

## 3.2 代码部分

1. 在eclipse运行hbase的Java程序前要导入hbase的包，我根据网上说的导入lib下的所有包，报错。怀疑是hadoop包的版本和本地不同的问题，于是就又导入了hadoop的包，就成功了！后来才知道原来林子雨老师导完包后还导入了配置文件，于是我怀疑有两种方法可以解决问题：一种就是导入hbase的包后导入配置文件，第二种就是跟我一样导入hbase中除了Hadoop开头的包后，再导入本地Hadoop的包。

2. 编程题的第三题有一个要求：当参数column为某一列族名称时，如果底下有若干个列限定符，则要列出每个列限定符代表的列的数据；当参数column为某一列具体名称（例如“Score:Math”）时，只需要列出该列的数据。

直接使用scan.addFamily(Bytes.toBytes(column));虽然可以执行，但是会报错。于是我改成

if (column.lastIndexOf(":") == -1) {//列族

scan.addFamily(Bytes.toBytes(column));//指定需要的family

} else {

String[] cols = column.split(":");//列族，列名

scan.addColumn(cols[0].getBytes(), cols[1].getBytes());

}

判断输入的是列族还是列，列就使用addColumn方法，列族就使用addFamily。

后面两题类似，列和列族分开讨论。