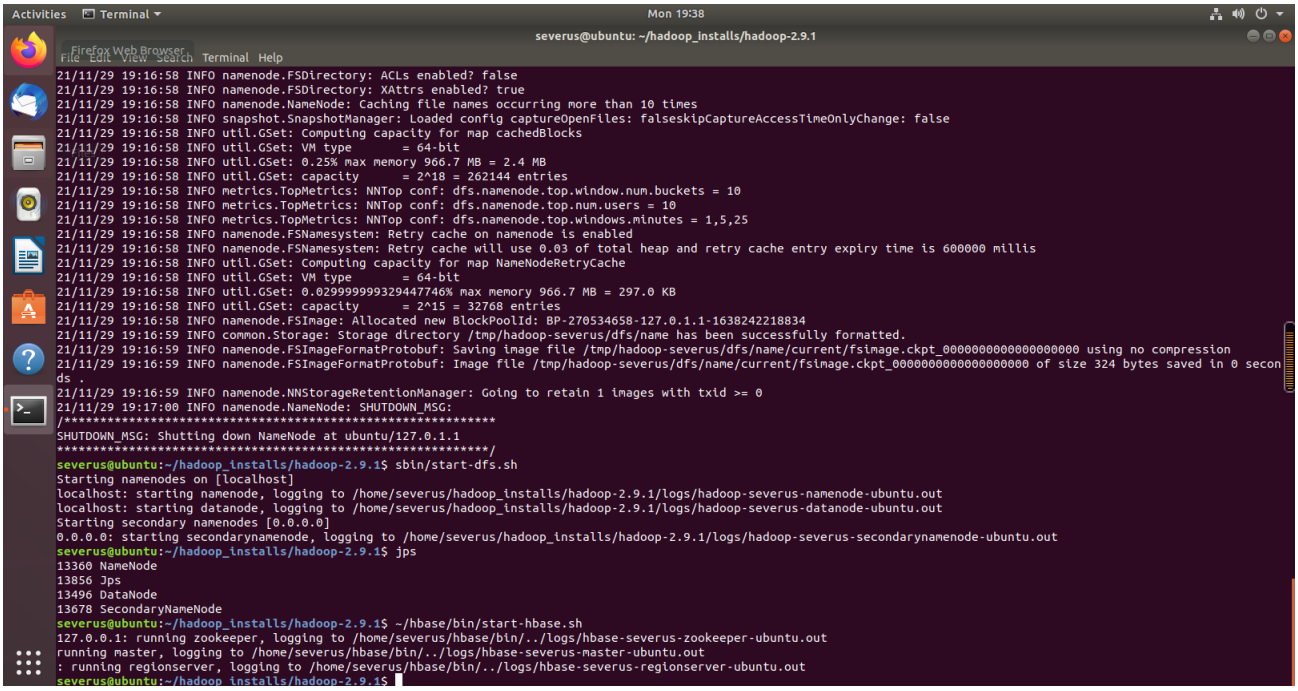
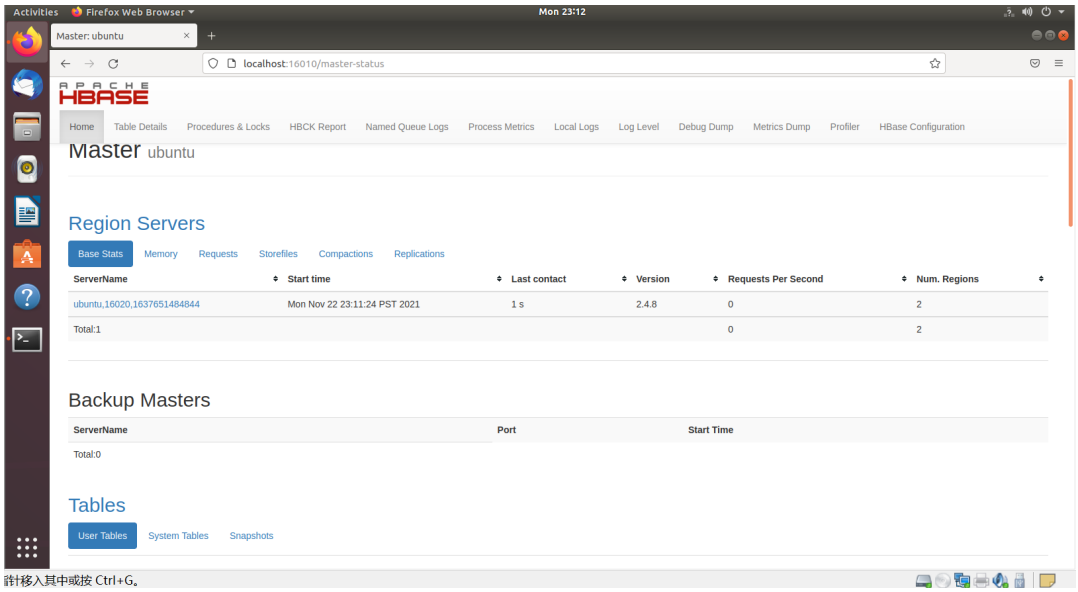
191098323 张峻珲 实验三

一. 环境配置

安装Hbase，成功运行伪分布模式





二. 设计Hbase表

以学生的学号为rowKey，每一行为一个学生的相关信息

初始表共4个列族，分别存储学生的个人信息和三门课程；Student列族共3列：S_Name，S_Sex，S_Age；每个课程列族各4列：C_No，C_Credit，C_Name，SC_Score

	Student			Math				Computer Science				English			
rowKey(S_No)	S_Name	S_Sex	S_Age	C_No	C_Credit	C_Name	SC_Score	C_No	C_Credit	C_Name	SC_Score	C_No	C_Credit	C_Name	SC_Score
2015001	Li Lei	male	23	123001	2	Math	86	123002	5	Computer Science		123003	3	English	69
2015002	Han Meimei	female	22	123001	2	Math		123002	5	Computer Science	77	123003	3	English	99
2015003	Zhang San	male	24	123001	2	Math	98	123002	5	Computer Science	95	123003	3	English	

三. 编写java程序

1. 创建表格

```
public static void createTable(String tableName, String[] familys) throws Exception{
    HBaseAdmin admin = newHBaseAdmin(conf);
    if (admin.tableExists(tableName)) {
        System.out.println("table"+tableName+"already exists!");
    }else {
        HTableDescriptor tableDesc =  new HTableDescriptor(tableName);
        for(int i=0;i < familys.length;i++)
        {
            tableDesc.addFamily(new HColumDescriptor(familys[i]));
        }
        admin.createTable(tableDesc);
        System.out.println("create table" + tableName + " success!");
    }
}
```

2. 添加数据

```
public static void addData (String tableName, String rowKey, String family, String qualifier, String value)
    throws Exception{
    try{
        HTable table = new HTable(conf, tableName);
        Put put  = new Put(Bytes.toBytes(rowkey));
        put.add(Bytes.toBytes(family),Bytes.toBytes(qualifier),Bytes.toBytes(value));
        table.put(put);
        System.out.println("insert record success!");
    } catch (IOException e){
        e.printStackTrace();
    }
}
```

3. 根据列族和列查询信息

```
public static void scanByFilter(String tableName, String family, String qualifer, String value)
    throws IOException{
    HTable table = new HTable(conf, tableName);
    Scan scan = new Scan();
    scan.addColumn(Bytes.toBytes(family), Byte.toBytes(qualifer));
    Filter filter = new SingleColumnValueFilter(Bytes.toBytes(family),Bytes.toBytes(qualifer),CompareOp.EQUAL, Bytes.toBytes(valu
    scan.setFilter(filter);
    ResultScanner result  = table.getScanner(scan);
    for(Result r: result) {
        for(KeyValue kv : r.raw()) {
            if (new String(kv.getQualifier()).equals(SC_Score)){
                System.out.println("value:" + new String(kv.getValue()));
            }
        }
    }
}
```

查询选修Computer Science的学生的成绩

```
HbaseController.scanByFilter("Student_info","Computer Science","C_Name","Computer Science");
```

4. 新增列族

```
public static void addColumn(String tablename, String Columnname){
    Admin adminManger = conn.getAdmin();
    adminManger.disableTable(tablename);
    HTableDescriptor htd = adminManger.getTableDescriptor(tablename);
    HColumnDescriptor f = new HColumnDescriptor(Columnname); //新增列族
    f.setTimeToLive(TIME_TO_LIVE);
    htd.addFamily(f);
    adminManger.addColumn(tablename, f);
    adminManger.enableTableAsync(tablename);
}
```

增加新的列族Contact和新列S_Email，并添加数据

```
HbaseController.addColumn("Student_info","Contact");
HbaseController.addData("Student_info","2015001","Contact","S_Email","lilie@qq.com");
HbaseController.addData("Student_info","2015002","Contact","S_Email","hmm@qq.com");
HbaseController.addData("Student_info","2015003","Contact","S_Email","zs@qq.com");
```

5. 删除信息

```
public static void deleteColumn(String tableName, String rowKey, String family){
    try{
        Table table = conn.getTable(TableName.valueOf(tableName));
        List<Delete> list = new ArrayList<Delete>();
        Delete del = new Delete(rowKey.getBytes());
        del.deleteFamily(Bytes.toBytes(family));
        list.add(del);
        table.delete(list);
        System.out.println("Del record:"+rowKey+"-"+family);
    }catch (IOException e){
        e.printStackTrace();
    }
}
```

删除学号为2015003的学生的选课记录

```
HbaseController.deleteColumn("Student_info","2015003","Math");
HbaseController.deleteColumn("Student_info","2015003","Computer Science");
HbaseController.deleteColumn("Student_info","2015003","English");
```

6. 删除表

```
public static void deleteTable(String tableName) throws IOException {
    init();
    TableName tn = TableName.valueOf(tableName);
    if (admin.tableExists(tn)) {
        admin.disableTable(tn);
        admin.deleteTable(tn);
    }
    close();
    System.out.println("删除了表："+tableName);
}
```

```
HbaseController.deleteTable("Student_info");
```

7. 完整代码

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.*;
import org.apache.hadoop.hbase.client.*;
import org.apache.hadoop.hbase.util.Bytes;
import org.apache.hadoop.conf.Configuration;
import java.io.IOException;
import java.util.Scanner;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class HbaseController {
    private static Configuration conf = null;
    private static Connection conn = null;
    static {
        conf = HbaseConfiguration.create();
        conf.set("hbase.zookeeper.quorum","localhost");
        conf.set("hbase.zookeeper.property.clientPort","2181");
    }

    public static void createTable(String tableName, String[] familys) throws Exception{
        HBaseAdmin admin = newHBaseAdmin(conf);
        if (admin.tableExists(tableName)) {
            System.out.println("table"+tableName+"already exists!");
        }else {
            HTableDescriptor tableDesc =  new HTableDescriptor(tableName);
            for(int i=0;i < familys.length;i++)
            {
```

```
                tableDesc.addFamily(new HColumDescriptor(familys[i]));
            }
            admin.createTable(tableDesc);
            System.out.println("create table" + tableName + " success!");
        }
    }

    public static void addData (String tableName, String rowKey, String family, String
qualifier, String value)
        throws Exception{
        try{
            HTable table = new HTable(conf, tableName);
            Put put  = new Put(Bytes.toBytes(rowkey));
            put.add(Bytes.toBytes(family),Bytes.toBytes(qualifier),Bytes.toBytes(value));
            table.put(put);
            System.out.println("insert record success!");
        } catch (IOException e){
            e.printStackTrace();
        }
    }

    public static void scanByFilter(String tableName, String family, String qualifer, String
value)
        throws IOException{
        HTable table = new HTable(conf, tableName);
        Scan scan = new Scan();
        scan.addColumn(Bytes.toBytes(family), Byte.toBytes(qualifer));
        Filter filter = new
SingleColumnValueFilter(Bytes.toBytes(family),Bytes.toBytes(qualifer),CompareOp.EQUAL,
Bytes.toBytes(value));
        scan.setFilter(filter);
        ResultScanner result  = table.getScanner(scan);
        for(Result r: result) {
            for(KeyValue kv : r.raw()) {
                if (new String(kv.getQualifier()).equals(SC_Score)){
                    System.out.println("value:" + new String(kv.getValue()));
                }
            }
        }
    }

    public static void addColumn(String tablename, String Columnname){
        Admin adminManger = conn.getAdmin();
        adminManger.disableTable(tablename);
        HTableDescriptor htd = adminManger.getTableDescriptor(tablename);
        HColumnDescriptor f = new HColumnDescriptor(Columnname); //新增列族
        f.setTimeToLive(TIME_TO_LIVE);
        htd.addFamily(f);
        adminManger.addColumn(tablename, f);
        adminManger.enableTableAsync(tablename);
    }

    public static void deleteColumn(String tableName, String rowKey, String family){
        try{
            Table table = conn.getTable(TableName.valueOf(tableName));
            List<Delete> list = new ArrayList<Delete>();
            Delete del = new Delete(rowKey.getBytes());
            del.deleteFamily(Bytes.toBytes(family));
            list.add(del);
            table.delete(list);
            System.out.println("Del record:"+rowKey+"-"+family);
```

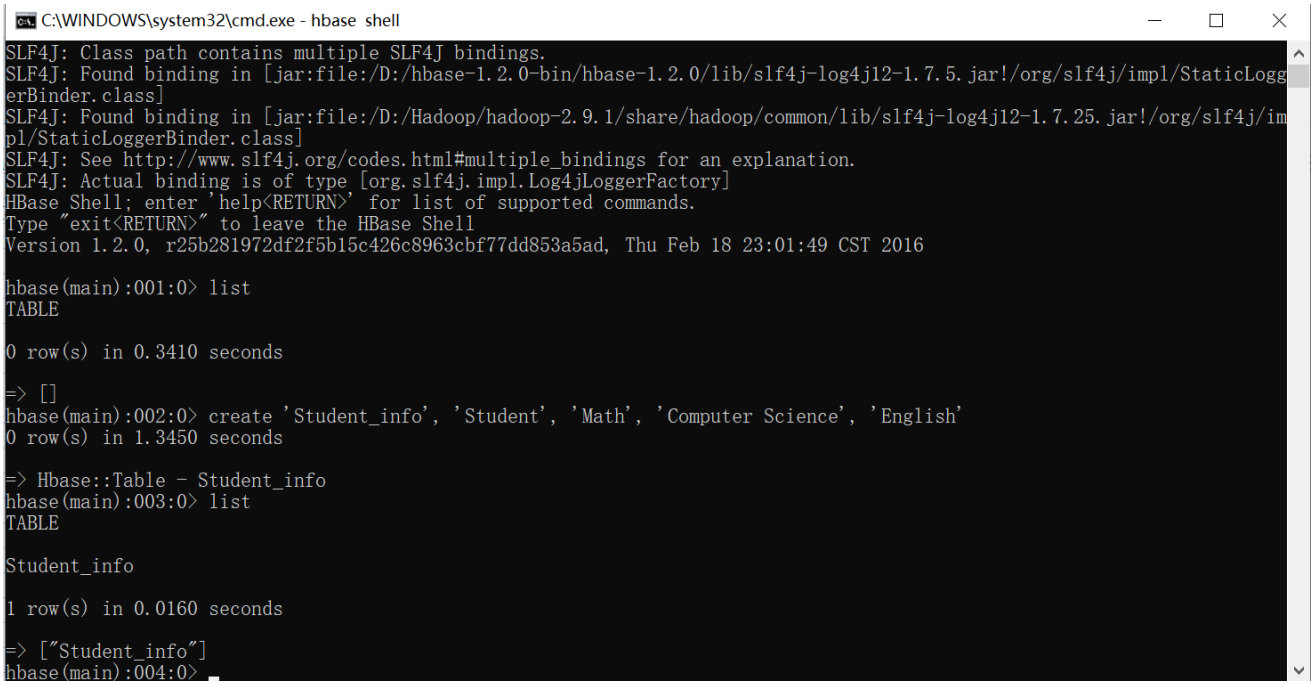
```
        }catch (IOException e){
            e.printStackTrace();
        }
    }

    public static void deleteTable(String tableName) throws IOException {
        init();
        TableName tn = TableName.valueOf(tableName);
        if (admin.tableExists(tn)) {
            admin.disableTable(tn);
            admin.deleteTable(tn);
        }
        close();
        System.out.println("删除了表："+tableName);
    }


    public static void main(String [] args)
    {
        try {
            String tablename = "Student_info";
            String[] familys = {"student","Math","Computer Science","English"};
            HbaseController.createTable(tablename,familys);
            // 添加学生信息
            HbaseController.addData("Student_info","2015001","student","S_No","2015001");
            HbaseController.addData("Student_info","2015001","student","S_Name","Li Lei");
            HbaseController.addData("Student_info","2015001","student","S_Sex","male");
            HbaseController.addData("Student_info","2015001","student","S_Age","23");

            HbaseController.addData("Student_info","2015002","student","S_No","2015002");
            HbaseController.addData("Student_info","2015002","student","S_Name","Han
Meimei");
            HbaseController.addData("Student_info","2015002","student","S_Sex","female");
            HbaseController.addData("Student_info","2015002","student","S_Age","22");

            HbaseController.addData("Student_info","2015003","student","S_No","2015003");
            HbaseController.addData("Student_info","2015003","student","S_Name","Zhang San");
            HbaseController.addData("Student_info","2015003","student","S_Sex","male");
            HbaseController.addData("Student_info","2015003","student","S_Age","24");

            HbaseController.addData("Student_info","2015001","Math","C_No","123001");
            HbaseController.addData("Student_info","2015001","Math","C_Credit","2");
            HbaseController.addData("Student_info","2015001","Math","SC_Score","86");
            HbaseController.addData("Student_info","2015001","Math","C_Name","Math");

            HbaseController.addData("Student_info","2015001","English","C_No","123003");
            HbaseController.addData("Student_info","2015001","English","C_Credit","3");
            HbaseController.addData("Student_info","2015001","English","SC_Score","69");
            HbaseController.addData("Student_info","2015001","English","C_Name","English");

            HbaseController.addData("Student_info","2015001","English","C_No","123003");
            HbaseController.addData("Student_info","2015001","English","C_Credit","3");
            HbaseController.addData("Student_info","2015001","English","SC_Score","69");
            HbaseController.addData("Student_info","2015001","English","C_Name","English");

            HbaseController.addData("Student_info","2015002","Computer
Science","C_No","123002");
            HbaseController.addData("Student_info","2015002","Computer
Science","C_Credit","5");
            HbaseController.addData("Student_info","2015002","Computer
Science","SC_Score","77");
```

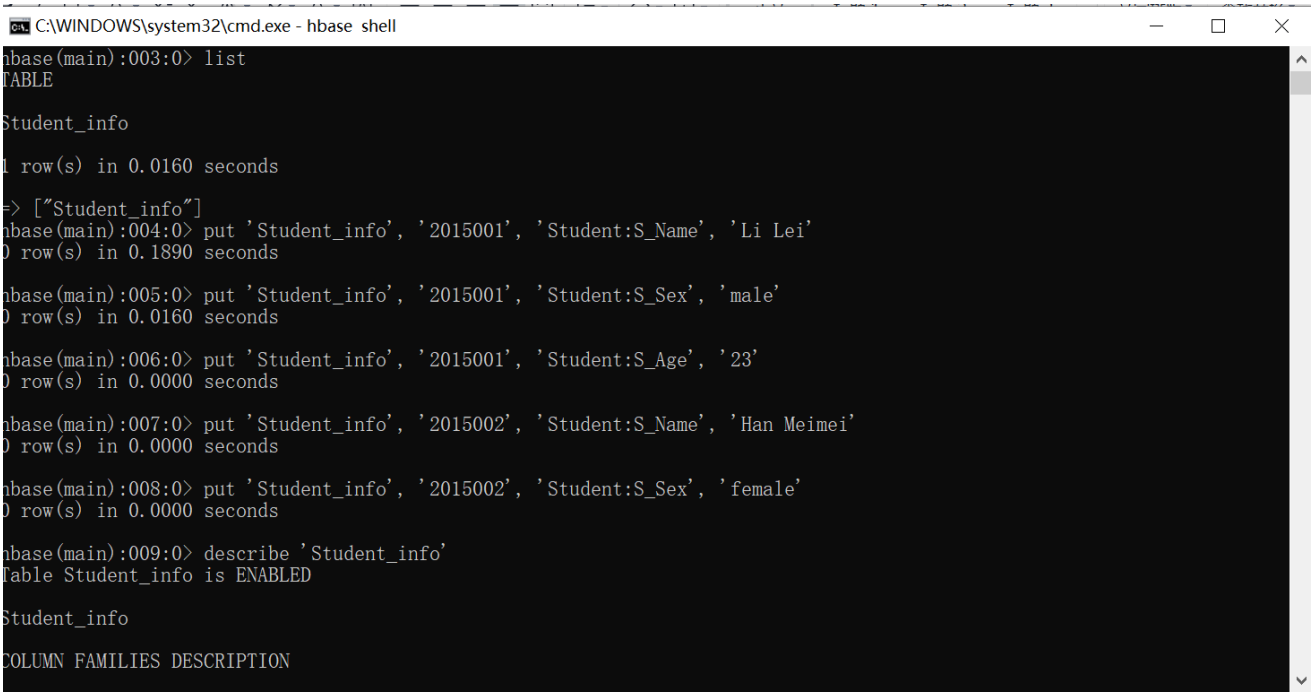
```
            HbaseController.addData("Student_info","2015002","Computer
Science","C_Name","Computer Science");

            HbaseController.addData("Student_info","2015002","English","C_No","123003");
            HbaseController.addData("Student_info","2015002","English","C_Credit","3");
            HbaseController.addData("Student_info","2015002","English","SC_Score","99");
            HbaseController.addData("Student_info","2015002","English","C_Name","English");

            HbaseController.addData("Student_info","2015003","Math","C_No","123001");
            HbaseController.addData("Student_info","2015003","Math","C_Credit","2");
            HbaseController.addData("Student_info","2015003","Math","SC_Score","98");
            HbaseController.addData("Student_info","2015003","Math","C_Name","Math");

            HbaseController.addData("Student_info","2015003","Computer
Science","C_No","123003");
            HbaseController.addData("Student_info","2015003","Computer
Science","C_Credit","5");
            HbaseController.addData("Student_info","2015003","Computer
Science","SC_Score","95");
            HbaseController.addData("Student_info","2015003","Computer
Science","C_Name","Computer Science");

            HbaseController.scanByFilter("Student_info","Computer Science","C_Name","Computer
Science");

            HbaseController.addColumn("Student_info","Contact");

 HbaseController.addData("Student_info","2015001","Contact","S_Email","lilie@qq.com");

 HbaseController.addData("Student_info","2015002","Contact","S_Email","hmm@qq.com");

 HbaseController.addData("Student_info","2015003","Contact","S_Email","zs@qq.com");

            HbaseController.deleteColumn("Student_info","2015003","Math");
            HbaseController.deleteColumn("Student_info","2015003","Computer Science");
            HbaseController.deleteColumn("Student_info","2015003","English");

            HbaseController.deleteTable("Student_info");
        }catch (Exception e){
            e.printStackTrace();
        }
    }
}
```

四. Shell完成操作

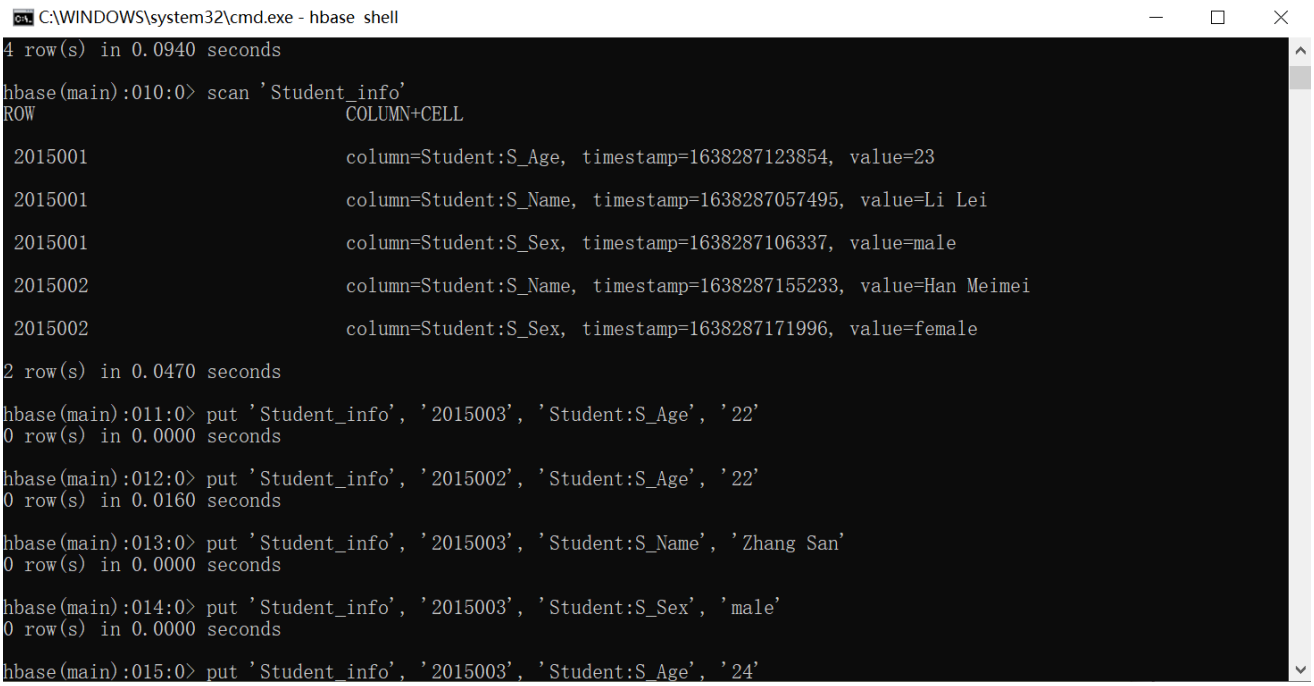
这一部分在windows下安装的Hbase下进行操作

1. 初始化

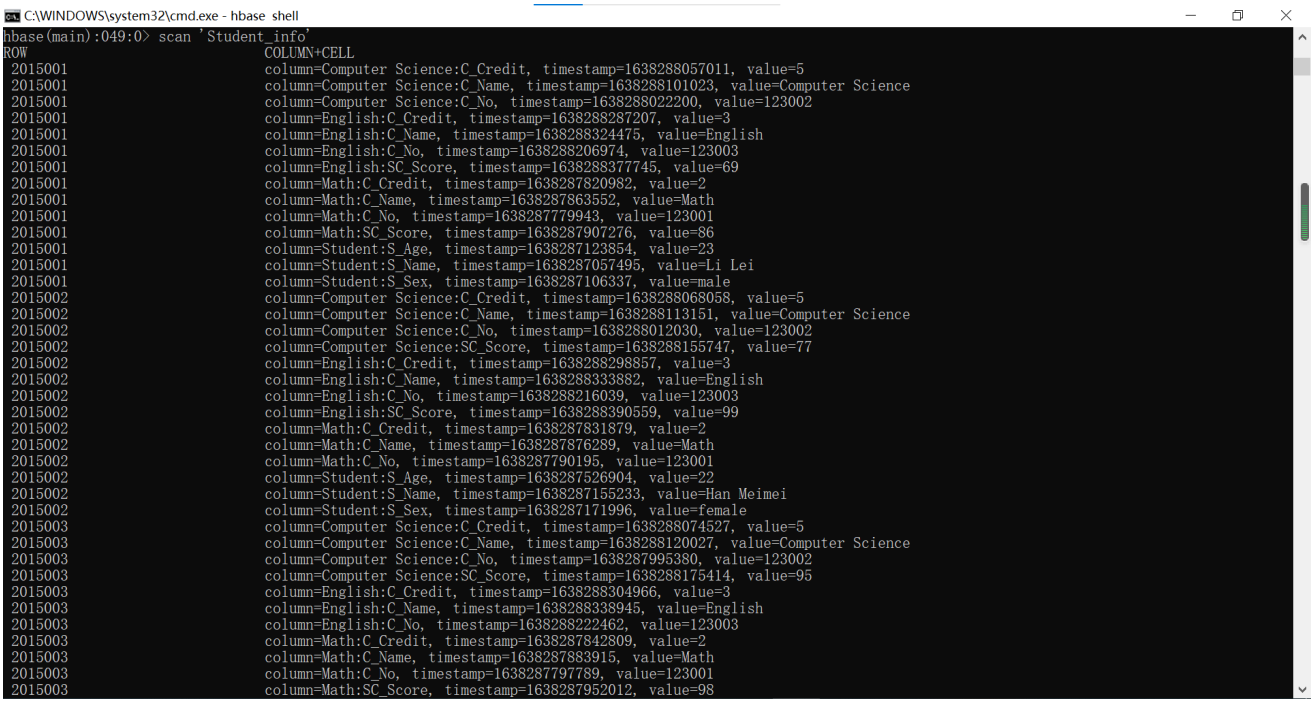




（不知为何这边用时变成了0，有点害怕，不过查看了一下没问题，于是继续）



完成



2. 查询Computer Science的学生的成绩

```
hbase(main):050:0> scan 'Student_info', {COLUMN=>'Computer Science:SC_Score'}
ROW                                   COLUMN+CELL

 2015002                              column=Computer Science:SC_Score, timestamp=1638288155747, value=77

 2015003                              column=Computer Science:SC_Score, timestamp=1638288175414, value=95

2 row(s) in 0.0160 seconds
```

3. 增加新的列族和新列Contact:Email，并添加数据

遇到报错，发现自己是输错了，卡在那里，通过>'指令退出重来

```
hbase(main):051:0> alter 'Student_info', Contact'
hbase(main):052:0'
hbase(main):053:0' >
hbase(main):054:0' >'
SyntaxError: (hbase):51: syntax error, unexpected tSTRING_BEG

alter 'Student_info', Contact'


hbase(main):055:0>
```

完成

```
hbase(main):056:0> alter 'Student_info', 'Contact'
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 1.9730 seconds

hbase(main):057:0> put 'Student_info', '2015001', 'Contact:S_Email', 'lilie@qq.com'
0 row(s) in 0.0000 seconds

hbase(main):058:0> put 'Student_info', '2015002', 'Contact:S_Email', 'hmm@qq.com'
0 row(s) in 0.0000 seconds

hbase(main):059:0> put 'Student_info', '2015003', 'Contact:S_Email', 'zs@qq.com'
0 row(s) in 0.0000 seconds
```

（遇到奇奇怪怪的问题，重新打开）

```
hbase(main):003:0> scan 'Student_info', {COLUMN=>'Contact:S_Email'}
ROW                                   COLUMN+CELL

 2015001                              column=Contact:S_Email, timestamp=1638289199213, value=lilie@qq.com

 2015002                              column=Contact:S_Email, timestamp=1638289225835, value=hmm@qq.com

 2015003                              column=Contact:S_Email, timestamp=1638289259321, value=zs@qq.com

3 row(s) in 0.0470 seconds
```

4. 删除学号为2015003的学生的选课记录

```
hbase(main):013:0> delete 'Student_info', '2015003', 'English:C_No'
0 row(s) in 0.0000 seconds

hbase(main):014:0> delete 'Student_info', '2015003', 'English:C_Name'
0 row(s) in 0.0000 seconds

hbase(main):015:0> delete 'Student_info', '2015003', 'English:C_Credit'
0 row(s) in 0.0150 seconds

hbase(main):016:0> get 'Student_info', '2015003'
COLUMN                                CELL

 Contact:S_Email                      timestamp=1638289259321, value=zs@qq.com

 Student:S_Age                        timestamp=1638287695841, value=24

 Student:S_Name                       timestamp=1638287557973, value=Zhang San

 Student:S_Sex                        timestamp=1638287593435, value=male

4 row(s) in 0.0160 seconds
```

5. 删除所创建的表

```
hbase(main):017:0> disable 'Student_info'
0 row(s) in 2.3300 seconds

hbase(main):018:0> drop 'Student_info'
0 row(s) in 1.2980 seconds

hbase(main):019:0> exists 'Student_info'
Table Student_info does not exist

0 row(s) in 0.0150 seconds
```

五. 遇到问题

```
hbase(main):060:0> scan 'Student_info', 'COLUMN=>'Contact:S_Email'
hbase(main):061:0' >'
SyntaxError: (hbase):60: syntax error, unexpected tCONSTANT

scan 'Student_info', 'COLUMN=>'Contact:S_Email'


hbase(main):062:0> scan 'Student_info', {'COLUMN=>'Contact:S_Email'}
hbase(main):063:1' >'
hbase(main):064:1> scan 'Student_info', {COLUMN=>'Contact:S_Email'}
hbase(main):065:1> >'
hbase(main):066:1' >'
hbase(main):067:1> scan 'Student_info', {COLUMN=>'Contact:S_Email'}
hbase(main):068:1> scan 'Student_info'
hbase(main):069:1> >+
```

这边输错了一次，然后就进入了1，之前做的都没有了，不知为何会这样，网上没有找到解决方案，只好重来；所幸重新进来之后，表格还在，于是继续操作

```
C:\WINDOWS\system32\cmd.exe - hbase shell
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0, r25b281972df2f5b15c426c8963cbf77dd853a5ad, Thu Feb 18 23:01:49 CST 2016

hbase(main):001:0> list
TABLE
Student_info
1 row(s) in 0.2500 seconds

=> ["Student_info"]
hbase(main):002:0> scan 'Student_info'
ROW                             COLUMN+CELL
 2015001                        column=Computer Science:C_Credit, timestamp=1638288057011, value=5
 2015001                        column=Computer Science:C_Name, timestamp=1638288101023, value=Computer Science
 2015001                        column=Computer Science:C_No, timestamp=1638288022200, value=123002
 2015001                        column=Contact:S_Email, timestamp=1638289199213, value=lilie@qq.com
 2015001                        column=English:C_Credit, timestamp=1638288287207, value=3
 2015001                        column=English:C_Name, timestamp=1638288324475, value=English
 2015001                        column=English:C_No, timestamp=1638288206974, value=123003
 2015001                        column=English:SC_Score, timestamp=1638288377745, value=69
 2015001                        column=Math:C_Credit, timestamp=1638287820982, value=2
 2015001                        column=Math:C_Name, timestamp=1638287863552, value=Math
 2015001                        column=Math:C_No, timestamp=1638287779943, value=123001
 2015001                        column=Math:SC_Score, timestamp=1638287907276, value=86
 2015001                        column=Student:S_Age, timestamp=1638287123854, value=23
 2015001                        column=Student:S_Name, timestamp=1638287057495, value=Li Lei
 2015001                        column=Student:S_Sex, timestamp=1638287106337, value=male
 2015002                        column=Computer Science:C_Credit, timestamp=1638288068058, value=5
 2015002                        column=Computer Science:C_Name, timestamp=1638288113151, value=Computer Science
 2015002                        column=Computer Science:C_No, timestamp=1638288012030, value=123002
 2015002                        column=Computer Science:SC_Score, timestamp=1638288155747, value=77
 2015002                        column=Contact:S_Email, timestamp=1638289225835, value=hmm@qq.com
 2015002                        column=English:C_Credit, timestamp=1638288298857, value=3
 2015002                        column=English:C_Name, timestamp=1638288333882, value=English
 2015002                        column=English:C_No, timestamp=1638288216039, value=123003
 2015002                        column=English:SC_Score, timestamp=1638288390559, value=99
 2015002                        column=Math:C_Credit, timestamp=1638287831879, value=2
 2015002                        column=Math:C_Name, timestamp=1638287876289, value=Math
 2015002                        column=Math:C_No, timestamp=1638287790195, value=123001
 2015002                        column=Student:S_Age, timestamp=1638287526904, value=22
 2015002                        column=Student:S_Name, timestamp=1638287155233, value=Han Meimei
```