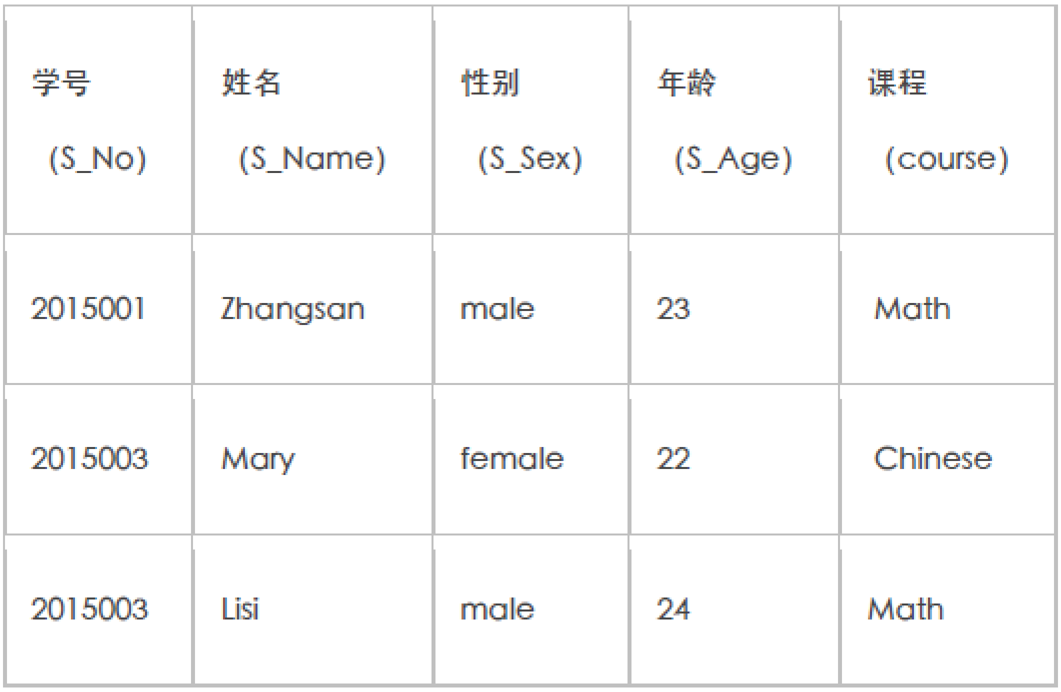
大规模分布式系统第五次作业——HBase的增删改查

16300200020 张言健

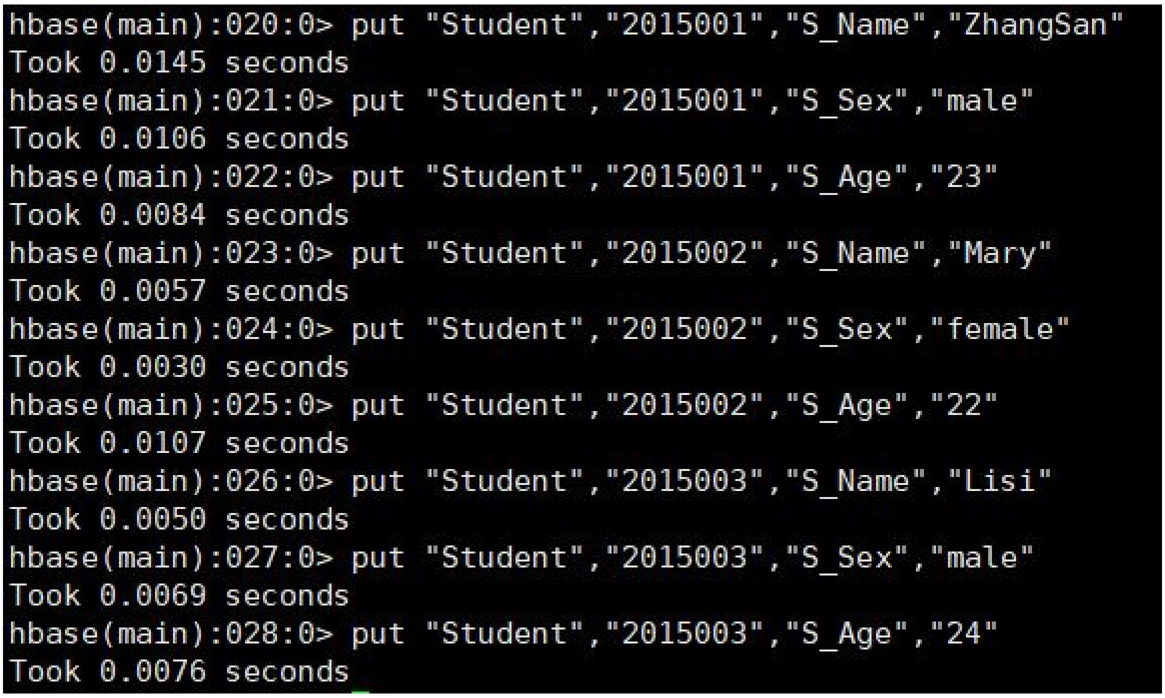
1. HBase的增删改

插入表中数据

```
create 'Student','S_Name','S_sex','S_Age'
```

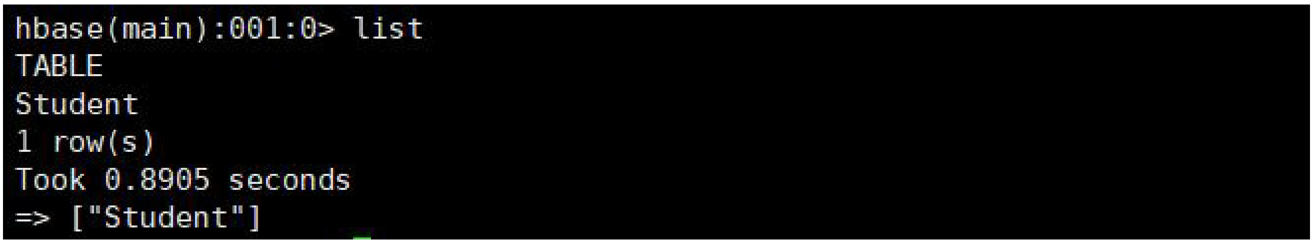


学号 （S_No）	姓名 （S_Name）	性别 （S_Sex）	年龄 （S_Age）	课程 （course）
2015001	Zhangsan	male	23	Math
2015003	Mary	female	22	Chinese
2015003	Lisi	male	24	Math



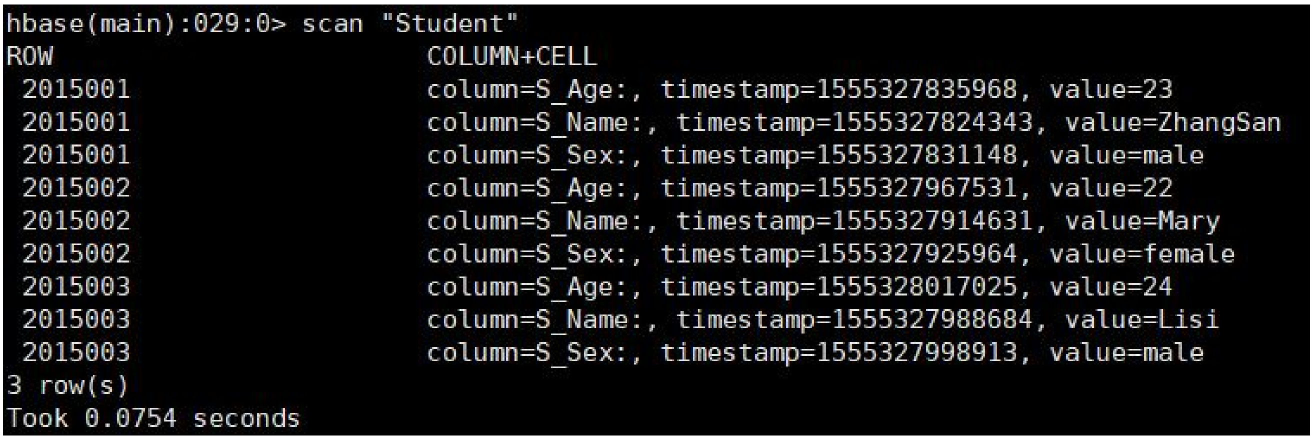
```
hbase(main):020:0> put "Student","2015001","S_Name","ZhangSan"
Took 0.0145 seconds
hbase(main):021:0> put "Student","2015001","S_Sex","male"
Took 0.0106 seconds
hbase(main):022:0> put "Student","2015001","S_Age","23"
Took 0.0084 seconds
hbase(main):023:0> put "Student","2015002","S_Name","Mary"
Took 0.0057 seconds
hbase(main):024:0> put "Student","2015002","S_Sex","female"
Took 0.0030 seconds
hbase(main):025:0> put "Student","2015002","S_Age","22"
Took 0.0107 seconds
hbase(main):026:0> put "Student","2015003","S_Name","Lisi"
Took 0.0050 seconds
hbase(main):027:0> put "Student","2015003","S_Sex","male"
Took 0.0069 seconds
hbase(main):028:0> put "Student","2015003","S_Age","24"
Took 0.0076 seconds
```

task1. 列出HBase 所有的表的相关信息



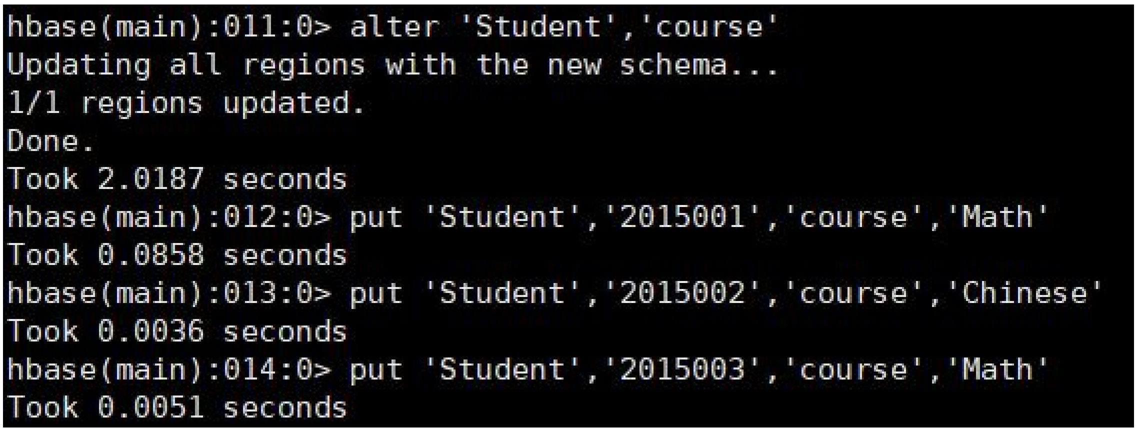
```
hbase(main):001:0> list
TABLE
Student
1 row(s)
Took 0.8905 seconds
=> ["Student"]
```

task2. 在终端打印出学生表的所有记录数据



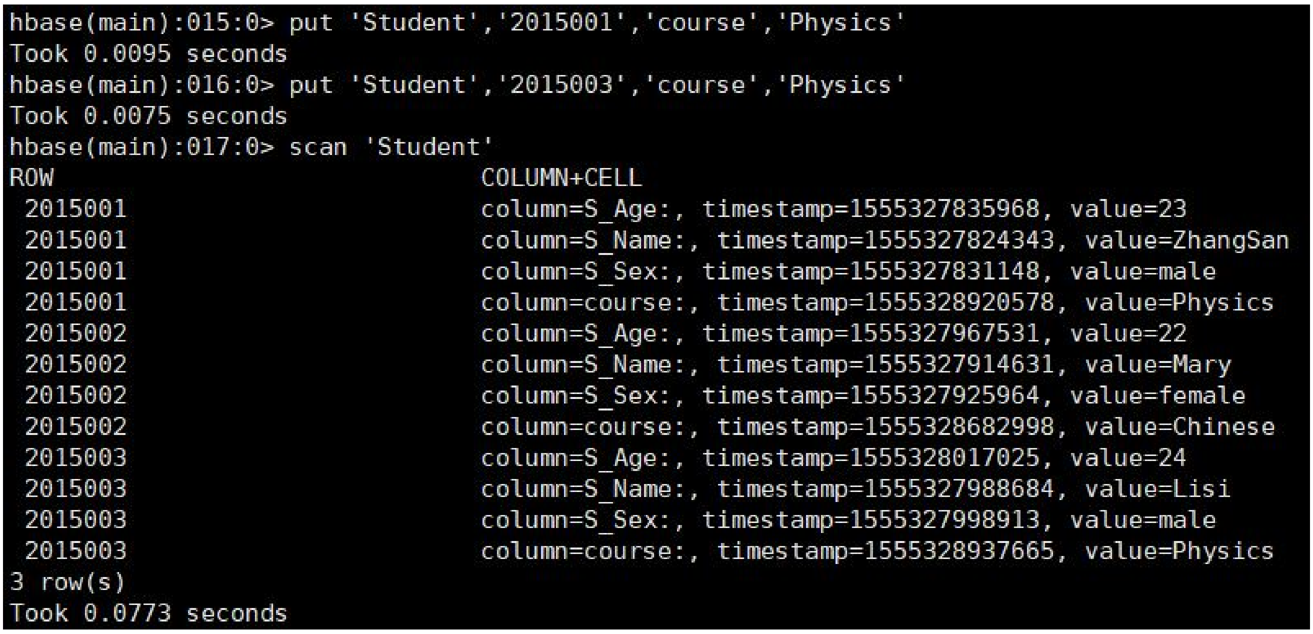
```
hbase(main):029:0> scan "Student"
ROW                        COLUMN+CELL
 2015001                   column=S_Age:, timestamp=1555327835968, value=23
 2015001                   column=S_Name:, timestamp=1555327824343, value=ZhangSan
 2015001                   column=S_Sex:, timestamp=1555327831148, value=male
 2015002                   column=S_Age:, timestamp=1555327967531, value=22
 2015002                   column=S_Name:, timestamp=1555327914631, value=Mary
 2015002                   column=S_Sex:, timestamp=1555327925964, value=female
 2015003                   column=S_Age:, timestamp=1555328017025, value=24
 2015003                   column=S_Name:, timestamp=1555327988684, value=Lisi
 2015003                   column=S_Sex:, timestamp=1555327998913, value=male
3 row(s)
Took 0.0754 seconds
```

task3. 向学生表添加课程列族



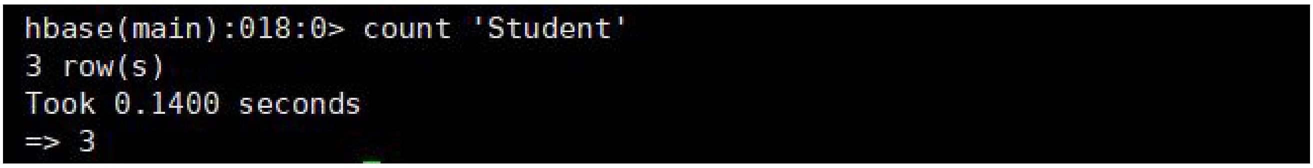
```
hbase(main):011:0> alter 'Student','course'
Updating all regions with the new schema...
1/1 regions updated.
Done.
Took 2.0187 seconds
hbase(main):012:0> put 'Student','2015001','course','Math'
Took 0.0858 seconds
hbase(main):013:0> put 'Student','2015002','course','Chinese'
Took 0.0036 seconds
hbase(main):014:0> put 'Student','2015003','course','Math'
Took 0.0051 seconds
```

task4. 将课程列族中的数学更换为物理



```
hbase(main):015:0> put 'Student','2015001','course','Physics'
Took 0.0095 seconds
hbase(main):016:0> put 'Student','2015003','course','Physics'
Took 0.0075 seconds
hbase(main):017:0> scan 'Student'
ROW                            COLUMN+CELL
 2015001                       column=S_Age:, timestamp=1555327835968, value=23
 2015001                       column=S_Name:, timestamp=1555327824343, value=ZhangSan
 2015001                       column=S_Sex:, timestamp=1555327831148, value=male
 2015001                       column=course:, timestamp=1555328920578, value=Physics
 2015002                       column=S_Age:, timestamp=1555327967531, value=22
 2015002                       column=S_Name:, timestamp=1555327914631, value=Mary
 2015002                       column=S_Sex:, timestamp=1555327925964, value=female
 2015002                       column=course:, timestamp=1555328682998, value=Chinese
 2015003                       column=S_Age:, timestamp=1555328017025, value=24
 2015003                       column=S_Name:, timestamp=1555327988684, value=Lisi
 2015003                       column=S_Sex:, timestamp=1555327998913, value=male
 2015003                       column=course:, timestamp=1555328937665, value=Physics
3 row(s)
Took 0.0773 seconds
```

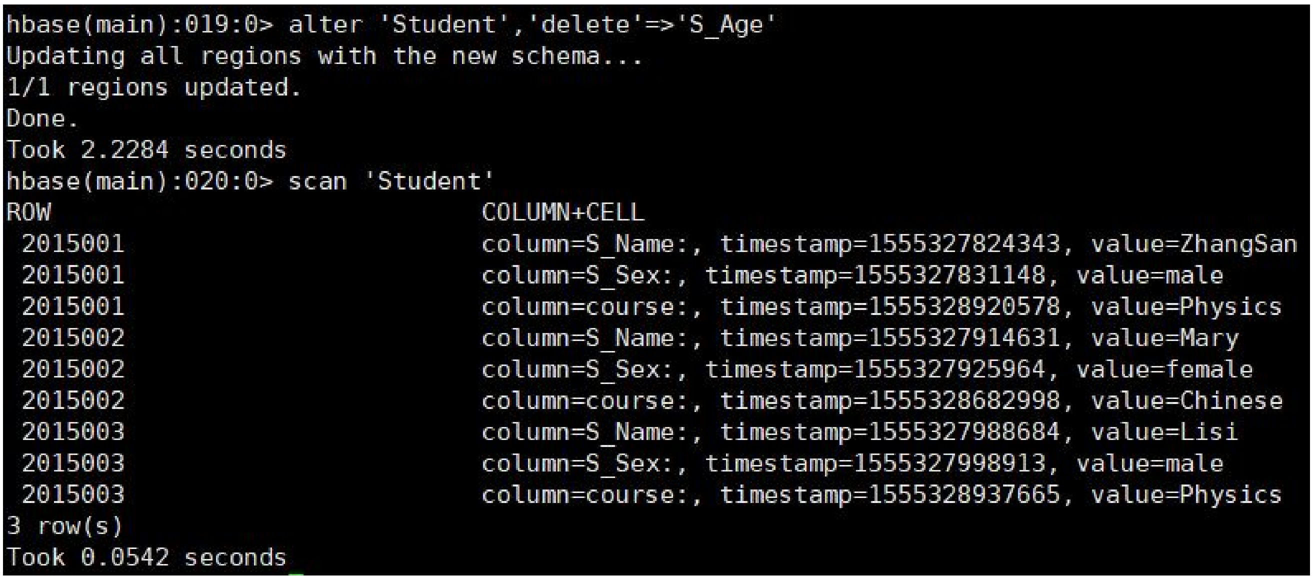
task5. 统计表的行数



```
hbase(main):018:0> count 'Student'
3 row(s)
Took 0.1400 seconds
=> 3
```

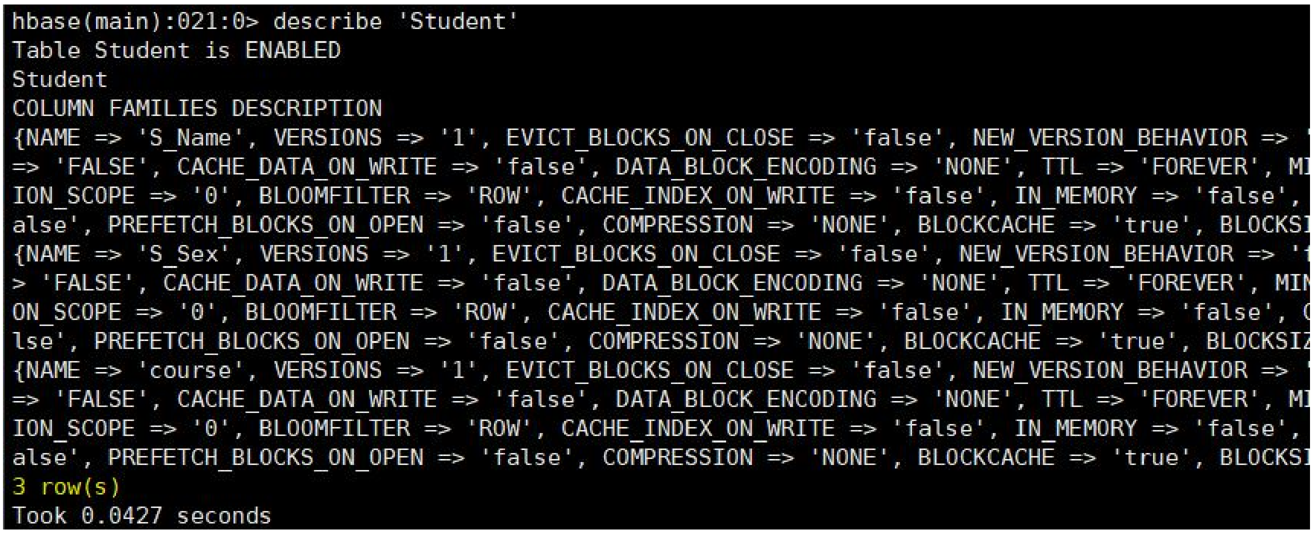
![count](D:\SP2(资料存储室)\复旦学习资料\大三下\分布式系统\第五次作业\count.JPG

task6. 删除年龄列



```
hbase(main):019:0> alter 'Student','delete'=>'S_Age'
Updating all regions with the new schema...
1/1 regions updated.
Done.
Took 2.2284 seconds
hbase(main):020:0> scan 'Student'
ROW                              COLUMN+CELL
 2015001                         column=S_Name:, timestamp=1555327824343, value=ZhangSan
 2015001                         column=S_Sex:, timestamp=1555327831148, value=male
 2015001                         column=course:, timestamp=1555328920578, value=Physics
 2015002                         column=S_Name:, timestamp=1555327914631, value=Mary
 2015002                         column=S_Sex:, timestamp=1555327925964, value=female
 2015002                         column=course:, timestamp=1555328682998, value=Chinese
 2015003                         column=S_Name:, timestamp=1555327988684, value=Lisi
 2015003                         column=S_Sex:, timestamp=1555327998913, value=male
 2015003                         column=course:, timestamp=1555328937665, value=Physics
3 row(s)
Took 0.0542 seconds
```

task7. 统计表的列数



```
hbase(main):021:0> describe 'Student'
Table Student is ENABLED
Student
COLUMN FAMILIES DESCRIPTION
{NAME => 'S_Name', VERSIONS => '1', EVICT_BLOCKS_ON_CLOSE => 'false', NEW_VERSION_BEHAVIOR => '
=> 'FALSE', CACHE_DATA_ON_WRITE => 'false', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MI
ION_SCOPE => '0', BLOOMFILTER => 'ROW', CACHE_INDEX_ON_WRITE => 'false', IN_MEMORY => 'false',
alse', PREFETCH_BLOCKS_ON_OPEN => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSI
{NAME => 'S_Sex', VERSIONS => '1', EVICT_BLOCKS_ON_CLOSE => 'false', NEW_VERSION_BEHAVIOR => 'f
> 'FALSE', CACHE_DATA_ON_WRITE => 'false', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN
ON_SCOPE => '0', BLOOMFILTER => 'ROW', CACHE_INDEX_ON_WRITE => 'false', IN_MEMORY => 'false', C
lse', PREFETCH_BLOCKS_ON_OPEN => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZ
{NAME => 'course', VERSIONS => '1', EVICT_BLOCKS_ON_CLOSE => 'false', NEW_VERSION_BEHAVIOR => '
=> 'FALSE', CACHE_DATA_ON_WRITE => 'false', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MI
ION_SCOPE => '0', BLOOMFILTER => 'ROW', CACHE_INDEX_ON_WRITE => 'false', IN_MEMORY => 'false',
alse', PREFETCH_BLOCKS_ON_OPEN => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSI
3 row(s)
Took 0.0427 seconds
```

2. 两表的自然拼接

给定两个表的信息

N1.txt

```
01 沐川文化艺术中心
02 上海浦东足球场
03 复合体育观演中心
```

D1.txt

```
2014/10/21  07:30          3,395,145 01-01 一层平面图.dwg
2014/10/21  07:29            924,099 01-02 二层平面图.dwg
2014/10/21  07:29            935,215 01-03 三-九层、十二-十五层平面图.dwg
```

由于在windows下载的两个文件并不是utf-8编码，因此需要先进行转换才能处理，否则会有如下报错

```
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xe3 in position 3: invalid
continuation byte
```

mapper.py

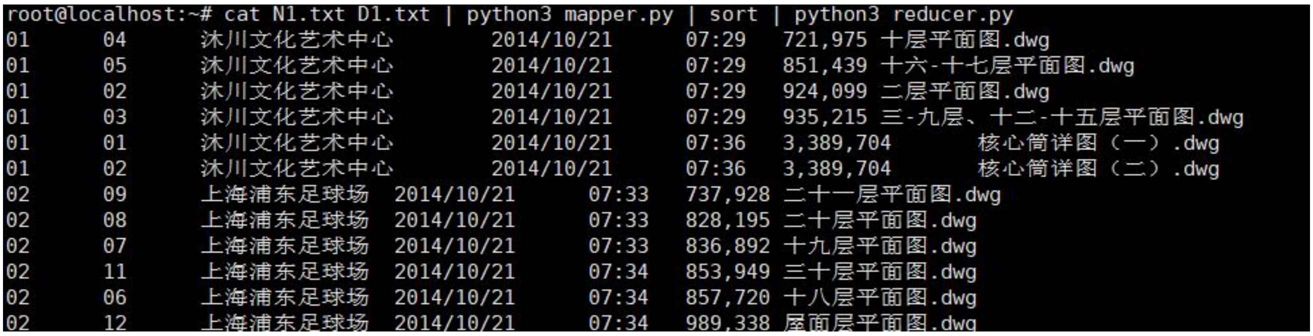
```
import sys
import io
import os
input_stream = sys.stdin
# the map_input_file will be set by hadoop
# filepath = os.environ["map_input_file"]  #也可使用map_input_file为“N1”或“D1”区别两个文件
# filename = os.path.split(filepath)[-1]
for line in input_stream:
    if line.strip()=="":
        continue
    # fields = line[:-1].split("\t")
    lineset = line[:-1].strip().split()
    # print(len(lineset))
    if len(lineset) == 2:  # 对于不规则的数据进行删除，此处使用词条个数来判断，便于本地测试
        location = lineset[1]
        idx = lineset[0]
        print(idx,"0",location)

    if len(lineset) > 4:
        date = lineset[0]
        time = lineset[1]
        filescale = lineset[2]
        subidx = lineset[3][3:5]
        idx = lineset[3][:2]
        project = lineset[4]
        print(idx,"1","\t".join((date,time,filescale,subidx, project)))
```

reducer.py

```
import sys
lastidx = ""
input_stream = sys.stdin
for line in input_stream:
    if line.strip() == "":
        continue
    lineset = line.strip().split()
    idx = lineset[0]
    if idx != lastidx:
        location = ""
        if lineset[1] == "0":  # 使用标记来源文件
            location = lineset[2]
    elif idx == lastidx:
        if lineset[1] == "1":
            date = lineset[2]
            time = lineset[3]
            filescale = lineset[4]
            subidx = lineset[5]
            file = lineset[6]
            if location:
                print("\t".join((lastidx, subidx, location, date, time, filescale,
file)))
    lastidx = idx
```

本地测试



```
root@localhost:~# cat N1.txt D1.txt | python3 mapper.py | sort | python3 reducer.py
01      04      沐川文化艺术中心        2014/10/21      07:29   721,975 十层平面图.dwg
01      05      沐川文化艺术中心        2014/10/21      07:29   851,439 十六-十七层平面图.dwg
01      02      沐川文化艺术中心        2014/10/21      07:29   924,099 二层平面图.dwg
01      03      沐川文化艺术中心        2014/10/21      07:29   935,215 三-九层、十二-十五层平面图.dwg
01      01      沐川文化艺术中心        2014/10/21      07:36   3,389,704       核心筒详图（一）.dwg
01      02      沐川文化艺术中心        2014/10/21      07:36   3,389,704       核心筒详图（二）.dwg
02      09      上海浦东足球场  2014/10/21      07:33   737,928 二十一层平面图.dwg
02      08      上海浦东足球场  2014/10/21      07:33   828,195 二十层平面图.dwg
02      07      上海浦东足球场  2014/10/21      07:33   836,892 十九层平面图.dwg
02      11      上海浦东足球场  2014/10/21      07:34   853,949 三十层平面图.dwg
02      06      上海浦东足球场  2014/10/21      07:34   857,720 十八层平面图.dwg
02      12      上海浦东足球场  2014/10/21      07:34   989,338 屋面层平面图.dwg
```

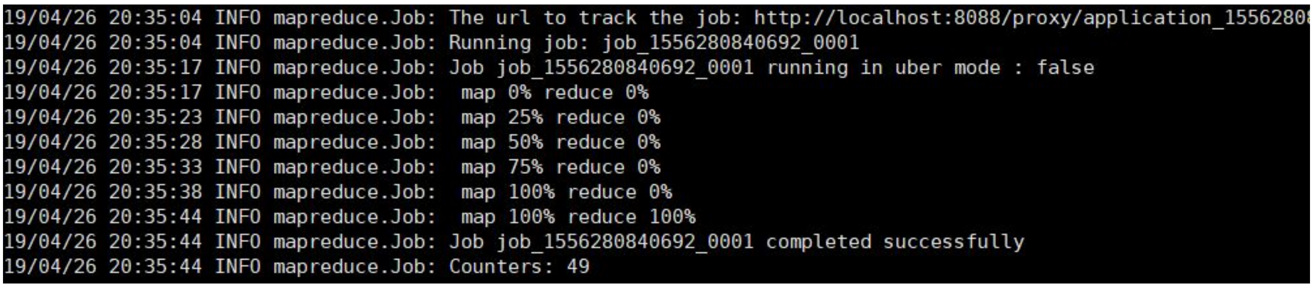
Hadoop测试

run.sh

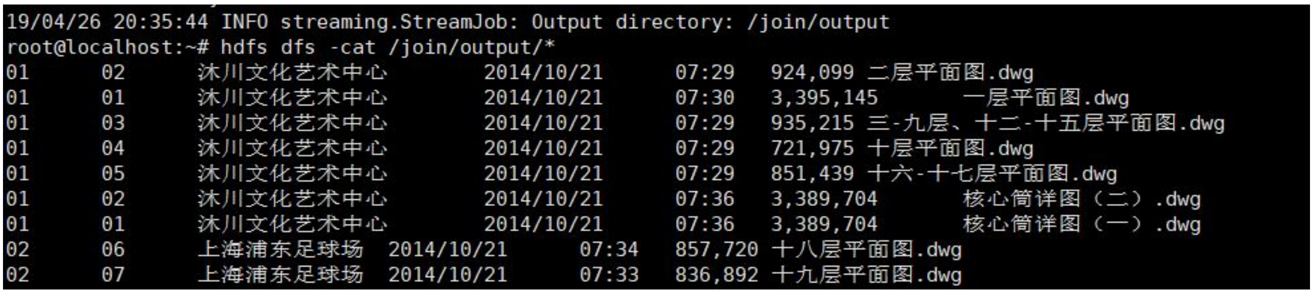
```
HADOOP_CMD="hadoop" #我的hadoop位置
STREAM_JAR_PATH="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.8.5.jar"
#streaming这个jar包的位置
INPUT_FILE_PATH_1="/join/input/*" #测试文件在hdfs中的位置。所以需要先将文件传入hdfs中
OUTPUT_PATH="/join/output" #文件输出目录（运行mr前一定不能存在，mr自己会创建）

$HADOOP_CMD fs -rmr -skipTrash $OUTPUT_PATH #删除原有的输出文件夹

#step 1.下面代码就是使用streaming框架的命令，具体参数就不解释了
$HADOOP_CMD jar $STREAM_JAR_PATH \
    -D mapred.map.tasks=3 \
    -D mapred.job.name="join_test" \
    -partitioner org.apache.hadoop.mapred.lib.KeyFieldBasedPartitioner \
        -input $INPUT_FILE_PATH_1 \
        -output $OUTPUT_PATH \
        -mapper "python3 mapper.py" \
        -reducer "python3 reducer.py" \
        -file ./mapper.py \
        -file ./reducer.py
```



```
19/04/26 20:35:04 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_155628084
19/04/26 20:35:04 INFO mapreduce.Job: Running job: job_1556280840692_0001
19/04/26 20:35:17 INFO mapreduce.Job: Job job_1556280840692_0001 running in uber mode : false
19/04/26 20:35:17 INFO mapreduce.Job:  map 0% reduce 0%
19/04/26 20:35:23 INFO mapreduce.Job:  map 25% reduce 0%
19/04/26 20:35:28 INFO mapreduce.Job:  map 50% reduce 0%
19/04/26 20:35:33 INFO mapreduce.Job:  map 75% reduce 0%
19/04/26 20:35:38 INFO mapreduce.Job:  map 100% reduce 0%
19/04/26 20:35:44 INFO mapreduce.Job:  map 100% reduce 100%
19/04/26 20:35:44 INFO mapreduce.Job: Job job_1556280840692_0001 completed successfully
19/04/26 20:35:44 INFO mapreduce.Job: Counters: 49
```



```
19/04/26 20:35:44 INFO streaming.StreamJob: Output directory: /join/output
root@localhost:~# hdfs dfs -cat /join/output/*
01      02      沐川文化艺术中心        2014/10/21      07:29   924,099 二层平面图.dwg
01      01      沐川文化艺术中心        2014/10/21      07:30   3,395,145       一层平面图.dwg
01      03      沐川文化艺术中心        2014/10/21      07:29   935,215 三-九层、十二-十五层平面图.dwg
01      04      沐川文化艺术中心        2014/10/21      07:29   721,975 十层平面图.dwg
01      05      沐川文化艺术中心        2014/10/21      07:29   851,439 十六-十七层平面图.dwg
01      02      沐川文化艺术中心        2014/10/21      07:36   3,389,704       核心筒详图（二）.dwg
01      01      沐川文化艺术中心        2014/10/21      07:36   3,389,704       核心筒详图（一）.dwg
02      06      上海浦东足球场  2014/10/21      07:34   857,720 十八层平面图.dwg
02      07      上海浦东足球场  2014/10/21      07:33   836,892 十九层平面图.dwg
```

测试成功

3. 设计数据库

连接后的数据表存入入Hbase数据库，请设计数据库rowkey和列

数据库的设计：

rowkeys	colume family	
	info	value
	info: idx info: sub_idx info: location info: date info: time info: file_scale info: file	idx sub_idx location date time file_scale file

每个数据单元的结果包含（idx，location, date, time, file_scale, sub_idx）

实现按文件名和项目编号可快速查询到文件信息

我们使用Thrift以及HappyBase来进行数据的处理，Apahce Thrift是FaceBook实现的一种高效的、支持多种语言的远程服务调用的框架。HappyBase是一个开发人员友好的Python库，可与HBase进行交互。

安装Thrift与HappyBase

```
wget https://www-us.apache.org/dist/thrift/0.12.0/thrift-0.12.0.tar.gz
tar -zxvf thrift-0.12.0.tar.gz
cd thrift-0.12.0/
./configure
make
make install
pip install thrift
pip install happybase
```

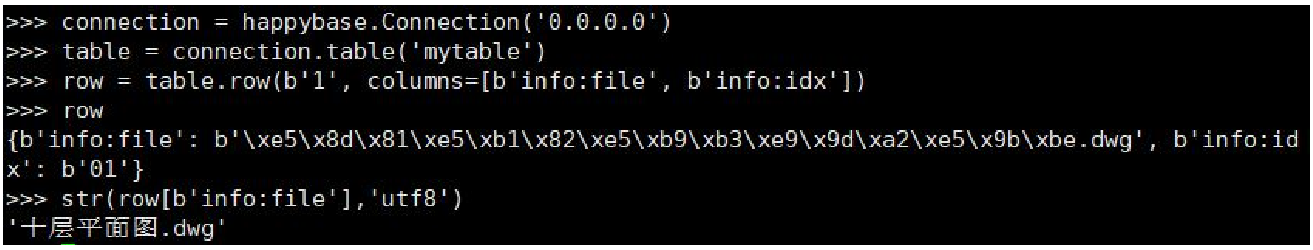
将已经处理的数据迁移到本地文件中

```
hdfs dfs -cat /join/output/* > output
```

hbase.py

```
import happybase
#连接数据库,ip地址取决于本地端口
connection = happybase.Connection('0.0.0.0')
connection.open()
connection.create_table(
    'mytable',
    {
        'info': dict()
    }
)
# rowkeys 采用递增顺序
rowkey = 1
# 生成 mytable 的连接
table = connection.table('mytable')
with open(r"output","r") as file:    # we can use scan 'myTable' to see the result
    line = file.readline().strip().split()
    while(line):
        [idx, subidx, name, date, time, filescale,  project] = line
        data = {
            'info:idx':idx,
            'info:sub_idx': subidx,
            'info:name': name,
            'info:date': date,
            'info:time': time,
            'info:file_scale': filescale,
            'info:project':  project
        }
        # print(data)
        table.put(str(rowkey),data)
        rowkey+=1
        line = file.readline().strip().split()
```

在交互式界面中输入如下字符即可得到查询结果：

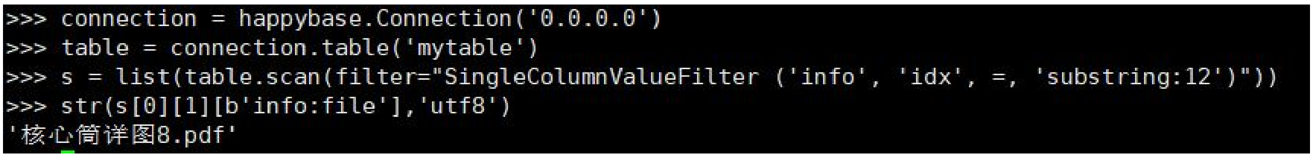


```
>>> connection = happybase.Connection('0.0.0.0')
>>> table = connection.table('mytable')
>>> row = table.row(b'1', columns=[b'info:file', b'info:idx'])
>>> row
{b'info:file': b'\xe5\x8d\x81\xe5\xb1\x82\xe5\xb9\xb3\xe9\x9d\xa2\xe5\x9b\xbe.dwg', b'info:id
x': b'01'}
>>> str(row[b'info:file'],'utf8')
'十层平面图.dwg'
```

此时数据库中保存的是十六进制的序列，我们可以用utf-8来解码

按项目编号分组输出

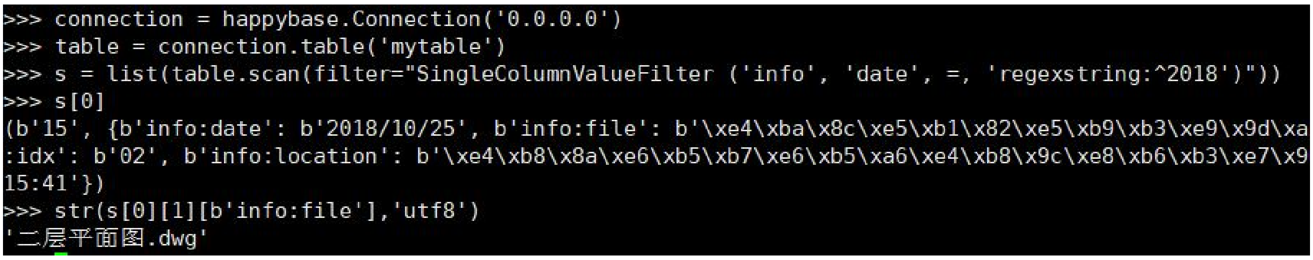
可以使用happybase的scan API来实现查询，输出结果如图，scan本身是返回一个generator对象，用list函数可以转换为python列表



```
>>> connection = happybase.Connection('0.0.0.0')
>>> table = connection.table('mytable')
>>> s = list(table.scan(filter="SingleColumnValueFilter ('info', 'idx', =, 'substring:12')"))
>>> str(s[0][1][b'info:file'],'utf8')
'核心筒详图8.pdf'
```

按图纸年份分组输出

由于此时记录的date的格式为'y/m/d'，因此使用年份作为开头的正则表达式可以提取出结果



```
>>> connection = happybase.Connection('0.0.0.0')
>>> table = connection.table('mytable')
>>> s = list(table.scan(filter="SingleColumnValueFilter ('info', 'date', =, 'regexstring:^2018')"))
>>> s[0]
(b'15', {b'info:date': b'2018/10/25', b'info:file': b'\xe4\xba\x8c\xe5\xb1\x82\xe5\xb9\xb3\xe9\x9d\xa
:idx': b'02', b'info:location': b'\xe4\xb8\x8a\xe6\xb5\xb7\xe6\xb5\xa6\xe4\xb8\x9c\xe8\xb6\xb3\xe7\x9
15:41'})
>>> str(s[0][1][b'info:file'],'utf8')
'二层平面图.dwg'
```