FBDP-实验三

邵一淼 191098180

需求分析

要求包含以下四张表的数据：

学生表(student)

学号S_No	姓名S_Name	性别S_Sex	年龄S_Age
2015001	Li Lei	male	23
2015002	Han Meimei	female	22
2015003	Zhang San	male	24

课程(course)

课程号C_No	课程名C_Name	学分C_Credit
123001	Math	2
123002	Computer Science	5
123003	English	3

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

学生表(student)，增加联系方式

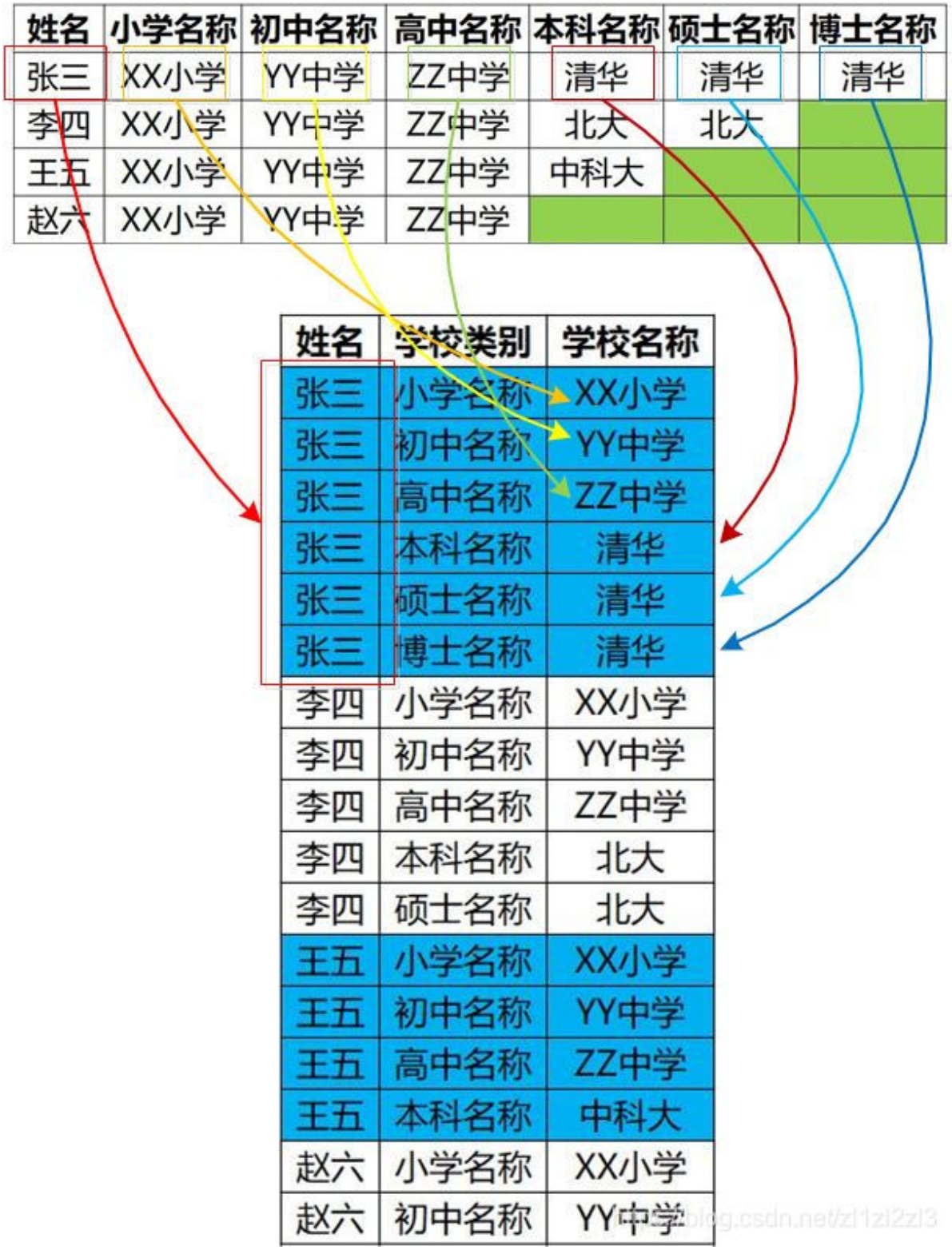
学号S_No	姓名S_Name	性别S_Sex	年龄S_Age	联系方式（S_Email)
2015001	Li Lei	male	23	lilie@qq.com
2015002	Han Meimei	female	22	hmm@qq.com
2015003	Zhang San	male	24	zs@qq.com

最开始的思路如下：

学号	姓名	选课	成绩	...
2015001	Li Lei	Math	86	
2015001	Li Lei	English	69	
2015002	Han Meimei	Computer Sciece	77	
...				
2015003	Zhang San	Computer Science	95	

这虽然是很一般的正常思路，但是**不适合列式存储**，反而比较适合行式存储

在阅读了许多列式存储的例子之后，尤其是下图



姓名	小学名称	初中名称	高中名称	本科名称	硕士名称	博士名称
张三	XX小学	YY中学	ZZ中学	清华	清华	清华
李四	XX小学	YY中学	ZZ中学	北大	北大	
王五	XX小学	YY中学	ZZ中学	中科大		
赵六	XX小学	YY中学	ZZ中学			

姓名	学校类别	学校名称
张三	小学名称	XX小学
张三	初中名称	YY中学
张三	高中名称	ZZ中学
张三	本科名称	清华
张三	硕士名称	清华
张三	博士名称	清华
李四	小学名称	XX小学
李四	初中名称	YY中学
李四	高中名称	ZZ中学
李四	本科名称	北大
李四	硕士名称	北大
王五	小学名称	XX小学
王五	初中名称	YY中学
王五	高中名称	ZZ中学
王五	本科名称	中科大
赵六	小学名称	XX小学
赵六	初中名称	YY中学

逐渐找到了感觉

将每一行的rowKey定义为学号，每个学生仅占用一行，学生个人信息作为一个列族，三门学科分别作为一个列族，最后表格如下所示：

student				Math				Computer				English			
学号	姓名	性别	年龄	课程号	课程名	学分	成绩	课程号	课程名	学分	成绩	课程号	课程名	学分	成绩
2015001	Li Lei	male	23	123001	Math	2	86					123003	English	3	69
2015002	Han Meim	female	22					123002	Computer	5	77	123003	English	3	99
2015003	Zhang Sa	male	24	123001	Math	2	98	123002	Computer	5	95				

代码

代码文件已上传到[Fairy-Miaomiao/HBase (github.com)]

Java程序

Java程序代码见JavaCode文件夹

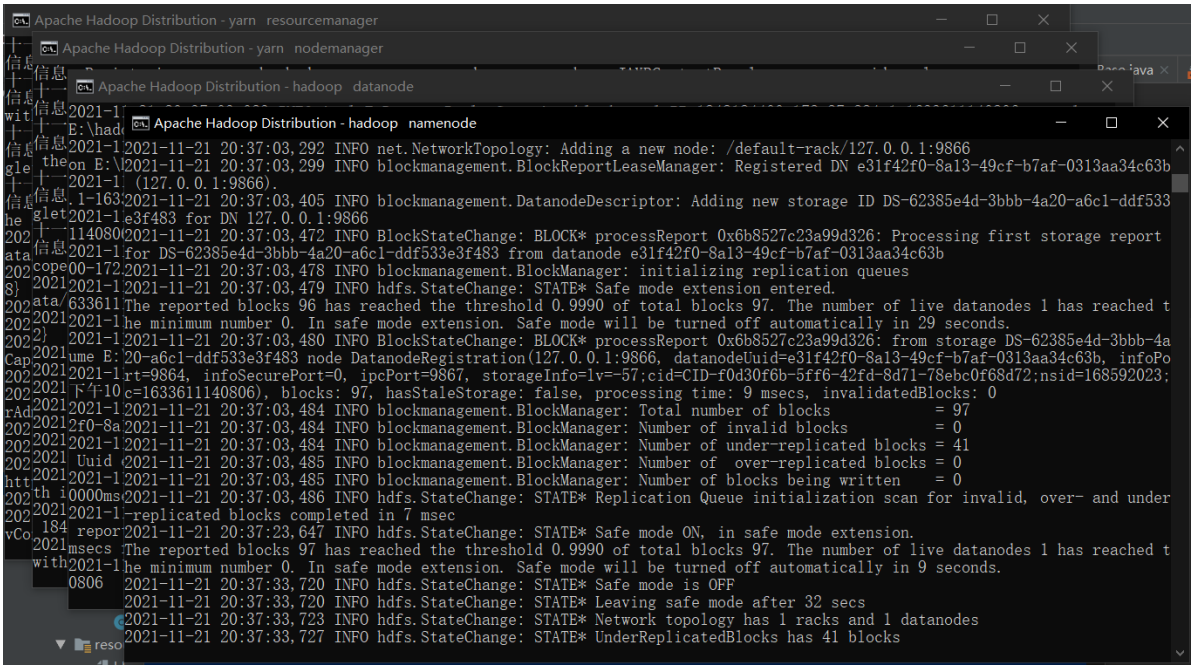
类名	功能
HBaseMain	入口类，通过调用HBaseFunction中的方法来完成一些表操作
HBaseFunction	实现了一些表操作的方法，如创建表、添加数据、查询数据、新增列、删除记录等

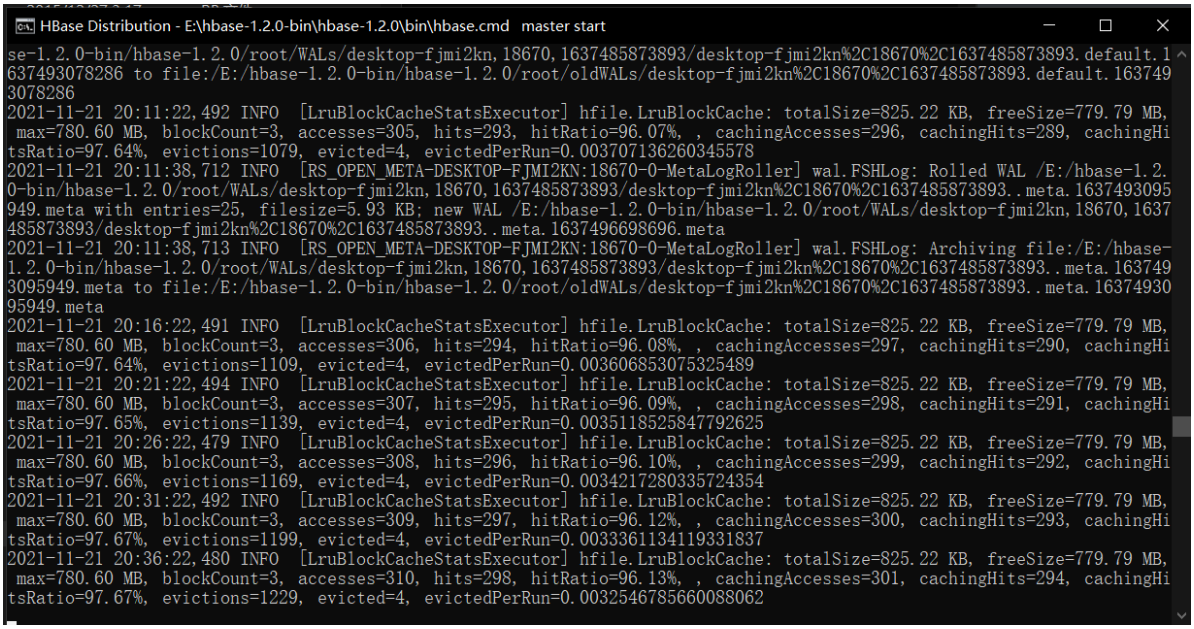
shell命令

shell代码见shell.txt

运行截图

启动hadoop和HBase





Java程序运行

```
Finish create table: StuClss
add 2015001:Student-S_No:2015001 in table:StuClss
add 2015001:Student-S_Name:Li Lei in table:StuClss
add 2015001:Student-S_Sex:male in table:StuClss
add 2015001:Student-S_Age:23 in table:StuClss
add 2015002:Student-S_No:2015002 in table:StuClss
add 2015002:Student-S_Name:Han Meimei in table:StuClss
add 2015002:Student-S_Sex:female in table:StuClss
add 2015002:Student-S_Age:22 in table:StuClss
add 2015003:Student-S_No:2015003 in table:StuClss
add 2015003:Student-S_Name:Zhang San in table:StuClss
add 2015003:Student-S_Sex:male in table:StuClss
add 2015003:Student-S_Age:24 in table:StuClss
add 2015001:Math-C_No:123001 in table:StuClss
add 2015001:Math-C_Name:Math in table:StuClss
add 2015001:Math-C_Credit:2 in table:StuClss
add 2015001:English-C_No:123003 in table:StuClss
add 2015001:English-C_Name:English in table:StuClss
add 2015001:English-C_Credit:3 in table:StuClss
add 2015002:Computer-C_No:123002 in table:StuClss
add 2015002:Computer-C_Name:Computer Science in table:StuClss
add 2015002:Computer-C_Credit:5 in table:StuClss
add 2015002:English-C_No:123003 in table:StuClss
add 2015002:English-C_Name:English in table:StuClss
add 2015002:English-C_Credit:3 in table:StuClss
add 2015003:Math-C_No:123001 in table:StuClss
add 2015003:Math-C_Name:Math in table:StuClss
add 2015003:Math-C_Credit:2 in table:StuClss
add 2015003:Computer-C_No:123002 in table:StuClss
add 2015003:Computer-C_Name:Computer Science in table:StuClss
add 2015003:Computer-C_Credit:5 in table:StuClss
add 2015001:Math-SC_Score:86 in table:StuClss
add 2015001:English-SC_Score:69 in table:StuClss
```

```
add 2015001:English-SC_Score:69 in table:StuClss
add 2015002:Computer-SC_Score:77 in table:StuClss
add 2015002:English-SC_Score:99 in table:StuClss
add 2015003:Math-SC_Score:98 in table:StuClss
add 2015003:Computer-SC_Score:95 in table:StuClss
以下是选修Computer Science的学生和他们的成绩:
学号为 2015002 的学生的Computer Science成绩为 77
学号为 2015003 的学生的Computer Science成绩为 95
add 2015001:Contact-Email:lilie@qq.com in table:StuClss
add 2015002:Contact-Email:hmm@qq.com in table:StuClss
add 2015003:Contact-Email:zs@qq.com in table:StuClss
Del record:2015003-Math ... Done.
Del record:2015003-Computer ... Done.
Del record:2015003-English ... Done.
Delete 2015003 information
Finish delete table:StuClss
```

shell运行

创建表

```
hbase(main):001:0> create 'StuClass','Student','Math','Computer','English'
0 row(s) in 1.5120 seconds

=> Hbase::Table - StuClass
hbase(main):002:0> list
TABLE

StuClass

1 row(s) in 0.0210 seconds

=> ["StuClass"]
hbase(main):003:0> put 'StuClass','2015001','Student:S_No','2015001'
0 row(s) in 0.1370 seconds

hbase(main):004:0> put 'StuClass','2015001','Student:S_Name','Li Lei'
0 row(s) in 0.0080 seconds

hbase(main):005:0> put 'StuClass','2015001','Student:S_Sex','male'
0 row(s) in 0.0040 seconds

hbase(main):006:0> put 'StuClass','2015001','Student:S_Age','23'
0 row(s) in 0.0030 seconds

hbase(main):007:0> scan 'StuClass',{COLUMN=>'Student'}
ROW                           COLUMN+CELL

 2015001                      column=Student:S_Age, timestamp=1637462890457, value=23

 2015001                      column=Student:S_Name, timestamp=1637462866552, value=Li Lei
```

```
hbase(main):018:0> put 'StuClass','2015002','Student:S_Sex','female'
0 row(s) in 0.0040 seconds

hbase(main):019:0> put 'StuClass','2015002','Student:S_Age','22'
0 row(s) in 0.0040 seconds

hbase(main):020:0> scan 'StuClass',{COLUMN=>'Student'}
ROW                           COLUMN+CELL

 2015001                      column=Student:S_Age, timestamp=1637462890457, value=23

 2015001                      column=Student:S_Name, timestamp=1637462866552, value=Li Lei

 2015001                      column=Student:S_No, timestamp=1637462642183, value=2015001

 2015001                      column=Student:S_Sex, timestamp=1637462880747, value=male

 2015002                      column=Student:S_Age, timestamp=1637463296862, value=22

 2015002                      column=Student:S_Name, timestamp=1637463015248, value=Han Meimei

 2015002                      column=Student:S_No, timestamp=1637462987218, value=2015002

 2015002                      column=Student:S_Sex, timestamp=1637463288923, value=female

2 row(s) in 0.0230 seconds

hbase(main):021:0> put 'StuClass','2015003','Student:S_No','2015003'
0 row(s) in 0.0050 seconds
```

```
hbase(main):021:0> put 'StuClass','2015003','Student:S_No','2015003'
0 row(s) in 0.0050 seconds

hbase(main):022:0> put 'StuClass','2015003','Student:S_Name','Zhang San'
0 row(s) in 0.0040 seconds

hbase(main):023:0> put 'StuClass','2015003','Student:S_Sex','male'
0 row(s) in 0.0030 seconds

hbase(main):024:0> put 'StuClass','2015003','Student:S_Age','24'
0 row(s) in 0.0050 seconds
```

```
hbase(main):025:0> scan 'StuClass',{COLUMN=>'Student'}
ROW                           COLUMN+CELL

 2015001                      column=Student:S_Age, timestamp=1637462890457, value=23

 2015001                      column=Student:S_Name, timestamp=1637462866552, value=Li Lei

 2015001                      column=Student:S_No, timestamp=1637462642183, value=2015001

 2015001                      column=Student:S_Sex, timestamp=1637462880747, value=male

 2015002                      column=Student:S_Age, timestamp=1637463296862, value=22

 2015002                      column=Student:S_Name, timestamp=1637463015248, value=Han Meimei

 2015002                      column=Student:S_No, timestamp=1637462987218, value=2015002

 2015002                      column=Student:S_Sex, timestamp=1637463288923, value=female

 2015003                      column=Student:S_Age, timestamp=1637463591349, value=24

 2015003                      column=Student:S_Name, timestamp=1637463566876, value=Zhang San

 2015003                      column=Student:S_No, timestamp=1637463550560, value=2015003

 2015003                      column=Student:S_Sex, timestamp=1637463579489, value=male

3 row(s) in 0.0520 seconds

hbase(main):026:0>
```

```
hbase(main):026:0> put 'StuClass','2015001','Math:C_No','123001'
0 row(s) in 0.0090 seconds

hbase(main):027:0> put 'StuClass','2015001','Math:C_Name','Math'
0 row(s) in 0.0060 seconds

hbase(main):028:0> put 'StuClass','2015001','Math:C_Credit','2'
0 row(s) in 0.0030 seconds

hbase(main):029:0> put 'StuClass','2015001','English:C_Credit','3'
0 row(s) in 0.0030 seconds

hbase(main):030:0> put 'StuClass','2015001','English:C_Name','English'
0 row(s) in 0.0050 seconds

hbase(main):031:0> put 'StuClass','2015001','English:C_No','123003'
0 row(s) in 0.0030 seconds

hbase(main):032:0> put 'StuClass','2015002','English:C_No','123003'
0 row(s) in 0.0030 seconds

hbase(main):033:0> put 'StuClass','2015002','English:C_Name','English'
0 row(s) in 0.0020 seconds

hbase(main):034:0> put 'StuClass','2015002','English:C_Credit','3'
0 row(s) in 0.0030 seconds

hbase(main):035:0> put 'StuClass','2015002','Computer:C_Credit','5'
0 row(s) in 0.0040 seconds

hbase(main):036:0> put 'StuClass','2015002','Computer:C_Name','Computer Science'
0 row(s) in 0.0040 seconds

hbase(main):037:0> put 'StuClass','2015002','Computer:C_No','123002'
0 row(s) in 0.0040 seconds

hbase(main):038:0> put 'StuClass','2015003','Computer:C_No','123002'
0 row(s) in 0.0040 seconds
```

```
hbase(main):040:0> put 'StuClass','2015003','Computer:C_Name','Computer Science'
0 row(s) in 0.0030 seconds

hbase(main):041:0> put 'StuClass','2015003','Computer:C_Credit','5'
0 row(s) in 0.0050 seconds

hbase(main):042:0> put 'StuClass','2015003','Math:C_Credit','2'
0 row(s) in 0.0030 seconds

hbase(main):043:0> put 'StuClass','2015003','Math:C_Name','Math'
0 row(s) in 0.0020 seconds

hbase(main):044:0> put 'StuClass','2015003','Math:C_No','123001'
0 row(s) in 0.0040 seconds
```

```
hbase(main):045:0> scan 'StuClass',{COLUMN=>'Math','Computer','English'}
SyntaxError: (hbase):45: syntax error, unexpected ','

scan 'StuClass',{COLUMN=>'Math','Computer','English'}


hbase(main):046:0> scan 'StuClass',{COLUMN=>['Math','Computer','English']}
ROW                           COLUMN+CELL
 2015001                      column=English:C_Credit, timestamp=1637463789372, value=3
 2015001                      column=English:C_Name, timestamp=1637463800747, value=English
 2015001                      column=English:C_No, timestamp=1637463813766, value=123003
 2015001                      column=Math:C_Credit, timestamp=1637463774529, value=2
 2015001                      column=Math:C_Name, timestamp=1637463761389, value=Math
 2015001                      column=Math:C_No, timestamp=1637463734918, value=123001
 2015002                      column=Computer:C_Credit, timestamp=1637463872564, value=5
 2015002                      column=Computer:C_Name, timestamp=1637463901412, value=Computer Science
 2015002                      column=Computer:C_No, timestamp=1637463917539, value=123002
 2015002                      column=English:C_Credit, timestamp=1637463853868, value=3
 2015002                      column=English:C_Name, timestamp=1637463837990, value=English
 2015002                      column=English:C_No, timestamp=1637463831744, value=123003
 2015003                      column=Computer:C_Credit, timestamp=1637463969270, value=5
 2015003                      column=Computer:C_Name, timestamp=1637463960875, value=Computer Science
 2015003                      column=Computer:C_No, timestamp=1637463946529, value=123002
```

```
hbase(main):047:0> put 'StuClass','2015001','Math:SC_Score','86'
0 row(s) in 0.0060 seconds

hbase(main):048:0> put 'StuClass','2015001','English:SC_Score','69'
0 row(s) in 0.0060 seconds

hbase(main):049:0> put 'StuClass','2015002','English:SC_Score','99'
0 row(s) in 0.0050 seconds

hbase(main):050:0> put 'StuClass','2015002','Computer:SC_Score','77'
0 row(s) in 0.0120 seconds

hbase(main):051:0> put 'StuClass','2015003','Computer:SC_Score','95'
0 row(s) in 0.0050 seconds

hbase(main):052:0> put 'StuClass','2015003','Math:SC_Score','98'
0 row(s) in 0.0050 seconds
```

```
hbase(main):053:0> scan 'StuClass',{COLUMN=>['Math:SC_Score','Computer:SC_Score','English:SC_Score']}
ROW                           COLUMN+CELL
 2015001                      column=English:SC_Score, timestamp=1637464346473, value=69
 2015001                      column=Math:SC_Score, timestamp=1637464314760, value=86
 2015002                      column=Computer:SC_Score, timestamp=1637464374292, value=77
 2015002                      column=English:SC_Score, timestamp=1637464361988, value=99
 2015003                      column=Computer:SC_Score, timestamp=1637464394051, value=95
 2015003                      column=Math:SC_Score, timestamp=1637464404400, value=98
3 row(s) in 0.0430 seconds
```

查询选修Computer Science的学生的成绩

```
hbase(main):054:0> scan 'StuClass',{COLUMN=>'Computer:SC_Score'}
ROW                           COLUMN+CELL
 2015002                      column=Computer:SC_Score, timestamp=1637464374292, value=77
 2015003                      column=Computer:SC_Score, timestamp=1637464394051, value=95
2 row(s) in 0.0140 seconds
```

增加新的列族和新列Contact:Email,并添加数据

（发现少截图了一行代码，最终代码以shell.txt为准）

```
hbase(main):056:0> put 'StuClass','2015001','Contact:Email','lilie@qq.com'
0 row(s) in 0.0050 seconds

hbase(main):057:0> put 'StuClass','2015002','Contact:Email','hmm@qq.com'
0 row(s) in 0.0030 seconds

hbase(main):058:0> put 'StuClass','2015003','Contact:Email','zs@qq.com'
0 row(s) in 0.0060 seconds

hbase(main):059:0> scan 'StuClass',{COLUMN=>'Contact:Email'}
ROW                           COLUMN+CELL

 2015001                      column=Contact:Email, timestamp=1637464636075, value=lilie@qq.com

 2015002                      column=Contact:Email, timestamp=1637464647539, value=hmm@qq.com

 2015003                      column=Contact:Email, timestamp=1637464659956, value=zs@qq.com

3 row(s) in 0.0170 seconds

hbase(main):060:0>
```

删除学号为2015003的学生的选课记录

```
hbase(main):060:0> delete 'StuClass','2015003','Math:C_No'
0 row(s) in 0.0310 seconds

hbase(main):061:0> delete 'StuClass','2015003','Math:C_Name'
0 row(s) in 0.0020 seconds

hbase(main):062:0> delete 'StuClass','2015003','Math:C_Credit'
0 row(s) in 0.0040 seconds

hbase(main):063:0> delete 'StuClass','2015003','Math:SC_Score'
0 row(s) in 0.0030 seconds

hbase(main):064:0> delete 'StuClass','2015003','Computer:SC_Score'
0 row(s) in 0.0180 seconds

hbase(main):065:0> delete 'StuClass','2015003','Computer:C_No'
0 row(s) in 0.0100 seconds

hbase(main):066:0> delete 'StuClass','2015003','Computer:C_Name'
0 row(s) in 0.0030 seconds

hbase(main):067:0> delete 'StuClass','2015003','English:C_Name'
0 row(s) in 0.0020 seconds

hbase(main):068:0> delete 'StuClass','2015003','English:C_No'
0 row(s) in 0.0030 seconds

hbase(main):069:0> delete 'StuClass','2015003','English:C_Credit'
0 row(s) in 0.0020 seconds

hbase(main):070:0> delete 'StuClass','2015003','English:SC_Score'
0 row(s) in 0.0020 seconds

hbase(main):071:0> delete 'StuClass','2015003','Computer:C_Credit'
0 row(s) in 0.0040 seconds

hbase(main):072:0> get 'StuClass','20115003','Math','Computer','English'
COLUMN                        CELL

0 row(s) in 0.0290 seconds
```

删除所创建的表

```
hbase(main):073:0> disable 'StuClass'
0 row(s) in 2.3390 seconds

hbase(main):074:0> drop 'StuClass'
0 row(s) in 1.2910 seconds

hbase(main):075:0> list
TABLE

0 row(s) in 0.0040 seconds

=> []
hbase(main):076:0>
```

问题总结及解决方案

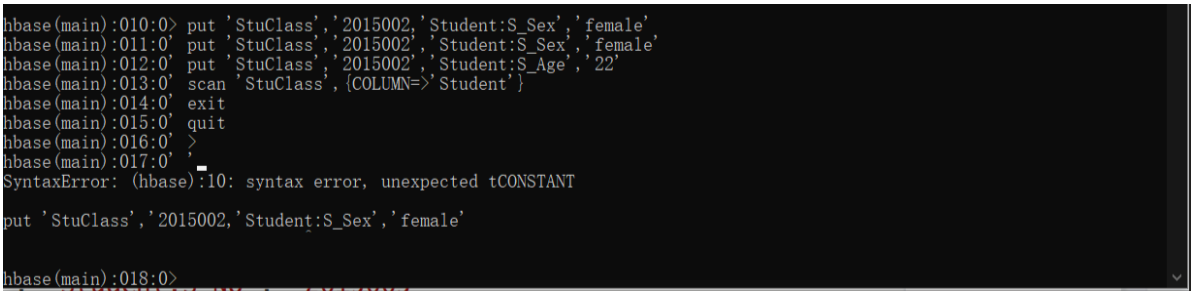
安装过程出人意料地顺利，没有遇到版本不兼容问题。

1、hbase shell中的>变成 '

问题描述：在添加数据时，输入了以下命令之后

```
put 'StuClass','2015002,'Student:S_Sex','female'
```

hbase shell中的>就变成了 '，如下图所示，输入其他命令也没有反应



解决方案：经查询资料得知hbase shell中不同符号的含义

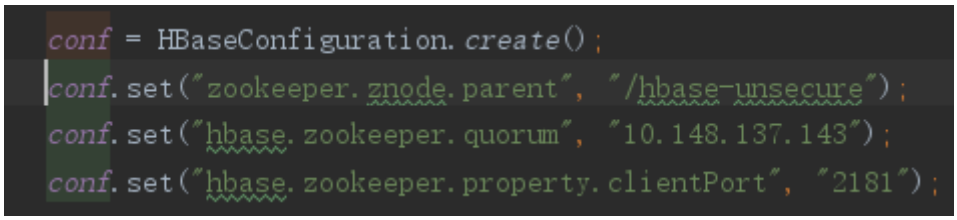
	含义
hbase(main):021:0*	表示还没输入完整的操作命令
hbase(main):021:0' hbase(main):021:0''	表示操作命令中的单引号或者双引号没有承兑
hbase(main)021:0>	表示刚刚执行完命令，还没有输入下一句操作命令

原先输入的命令因为失误少了一个单引号，所以只要再单独输入一个单引号就能恢复正常

2、org.apache.hadoop.hbase.client.RetriesExhaustedException: Can't get the locations

问题描述：运行Java程序时报错，

解决方案：在conf中多增加一行zookepper.znode.parent就能解决



其他思考

本实验的其他构建表格方式

本实验中，我创建的表格如下

student				Math				Computer				English			
学号	姓名	性别	年龄	课程号	课程名	学分	成绩	课程号	课程名	学分	成绩	课程号	课程名	学分	成绩
2015001	Li Lei	male	23	123001	Math	2	86					123003	English	3	69
2015002	Han Meim	female	22					123002	Computer	5	77	123003	English	3	99
2015003	Zhang Sa	male	24	123001	Math	2	98	123002	Computer	5	95				

做完实验回顾一番，不免开始思考，有没有其他创建方式呢?

显然是有的，如下图，rowKey为课程号，课程作为一个列族，每个学生又分别作为一个列族

Course			Li Lei					Han Meimei					Zhang San				
课程号	课程名	学分	学号	姓名	性别	年龄	成绩	学号	姓名	性别	年龄	成绩	学号	姓名	性别	年龄	成绩
123001	Math	2	2015001	Li Lei	male	23	86						2015003	Zhang San	male	24	98
123002	Computer	5						2015002	Han Meime	female	22	77	2015003	Zhang San	male	24	98
123003	English	3	2015001	Li Lei	male	23	69	2015002	Han Meime	female	22	99					

列式存储的优点在哪?

如前文需求分析所提及的那样，似乎在excel等软件十分普及的现在，大家的第一反应都觉得像excel那样的行式存储更易于理解和使用，而列式存储不仅不易理解，在存储数据时也很难想到（对我来说）。

阅读了许多材料，以下仅总结一些我能读懂或者说能产生共鸣的，至于一些看上去很高端但是我可能还没有涉及到的暂不说明

列式存储的优点：

- 同一列存放在一起，数据类型相同，则更好的进行压缩
- 同一列存放在一起，则排序更加方便，基于排序方便，where某一列会更加快

- 列式存储适合**"针对列"的查询**：
  比如select rowid from table_name，因为只会读取图中的第1个绿色部分的数据（查询时只有涉及到的字段会被读取），而select * from table_name limit 1则需要读取column-based stores**所有**绿色部分的数据（虽然目的就是要查询第1行的数据）；但是不适用于insert/update操作比较多的场景，比如当插入1个row时，由于列式存储导致同一个row的数据被分散在多个数据块中，因此需要去遍历所有数据块的数据。此外由于同一个字段连续存储（同一列的内容有很多值是重复的，可以压缩），因此更加便于编码压缩。

  综合来看，列式存储比较适合大数据量（压缩比高）、分析型操作（针对少数几列）；不适合频率较高的删除（全列检索）、更新（重新压缩）操作。

为什么需要HBase?

问题：关系数据库已经流行很多年，并且Hadoop已经有了HDFS和MapReduce,为什么需要HBase?

1、首先是HBase和HDFS的关系：

HDFS是Hadoop分布式文件系统。
HBase的数据通常存储在HDFS上。HDFS为HBase提供了高可靠性的底层存储支持。
Hbase是Hadoop database即Hadoop数据库。它是一个适合于非结构化数据存储的数据库，HBase基于列的而不是基于行的模式。
HBase是Google Bigtable的开源实现，类似Google Bigtable利用GFS作为其文件存储系统，HBase利用Hadoop HDFS作为其文件存储系统；Google运行MapReduce来处理Bigtable中的海量数据，HBase同样利用Hadoop MapReduce来处理HBase中的海量数据。
HDFS为HBase提供了高可靠性的底层存储支持，Hadoop MapReduce为HBase提供了高性能的计算能力，Zookeeper为HBase提供了稳定服务和failover机制（通俗地说，即当A无法为客户服务时，系统能够自动地切换，使B能够及时地顶上继续为客户提供服务，且客户感觉不到这个为他提供服务的对象已经更换）。Pig和Hive还为HBase提供了高层语言支持，使得在HBase上进行数据统计处理变的非常简单。 Sqoop则为HBase提供了方便的RDBMS（关系型数据库）数据导入功能，使得传统数据库数据向HBase中迁移变的非常方便。

2、HBASE本身作为一个分布式数据库
HBase 本身其实可以完全不要考虑 HDFS 的，可以只把 HBase 当作是一个分布式高并发 k-v 存储系统，只不过它底层的文件系统是通过 HDFS 来支持的罢了。换做其他的分布式文件系统也是一样的，不影响 HBase 的本质。甚至如果不考虑文件系统的分布式或稳定性等特性的话，完全可以用简单的本地文件系统，甚至内存文件系统来代替。（确实，单机版的HBase也可以正常使用）

HBase 在 HDFS 之上提供了：

①、高并发实时随机写，通过 LSM（内存+顺序写磁盘）的方式提供了 HDFS 所不拥有的实时随机写及修改功能

②、高并发实时点读及扫描了解一下 LSM 算法，在文件系统之上有数据库，在业务层面，HBase 完全可以独立于 HDFS 来理解

3、HBASE可以满足大规模数据的实时处理需求
HDFS面向批量访问模式，不是随机访问模式Hadoop可以很好地解决大规模数据的离线批量处理问题，但是，受限于Hadoop MapReduce编程框架的高延迟数据处理机制，使得Hadoop无法满足大规模数据实时处理应用的需求

●传统的通用关系型数据库无法应对在数据规模剧增时导致的系统扩展性和性能问题(分库分表也不能很好解决)

●传统关系数据库在数据结构变化时一般需要停机维护;空列浪费存储空间

因此，业界出现了一类面向半结构化数据存储和处理的高可扩展、低写入/查询延迟的系统，例如键值数据库、文档数据库和列族数据库(如BigTable和HBase等)

HBase已经成功应用于互联网服务领域和传统行业的众多在线式数据分析处理系统中

参考资料

hbase通过idea操作api_会飞的鱼-CSDN博客

HBase的配置 - 简书 (jianshu.com)

Java在HBase数据库创建表*chszs*的专栏-*CSDN*博客hbase java建表