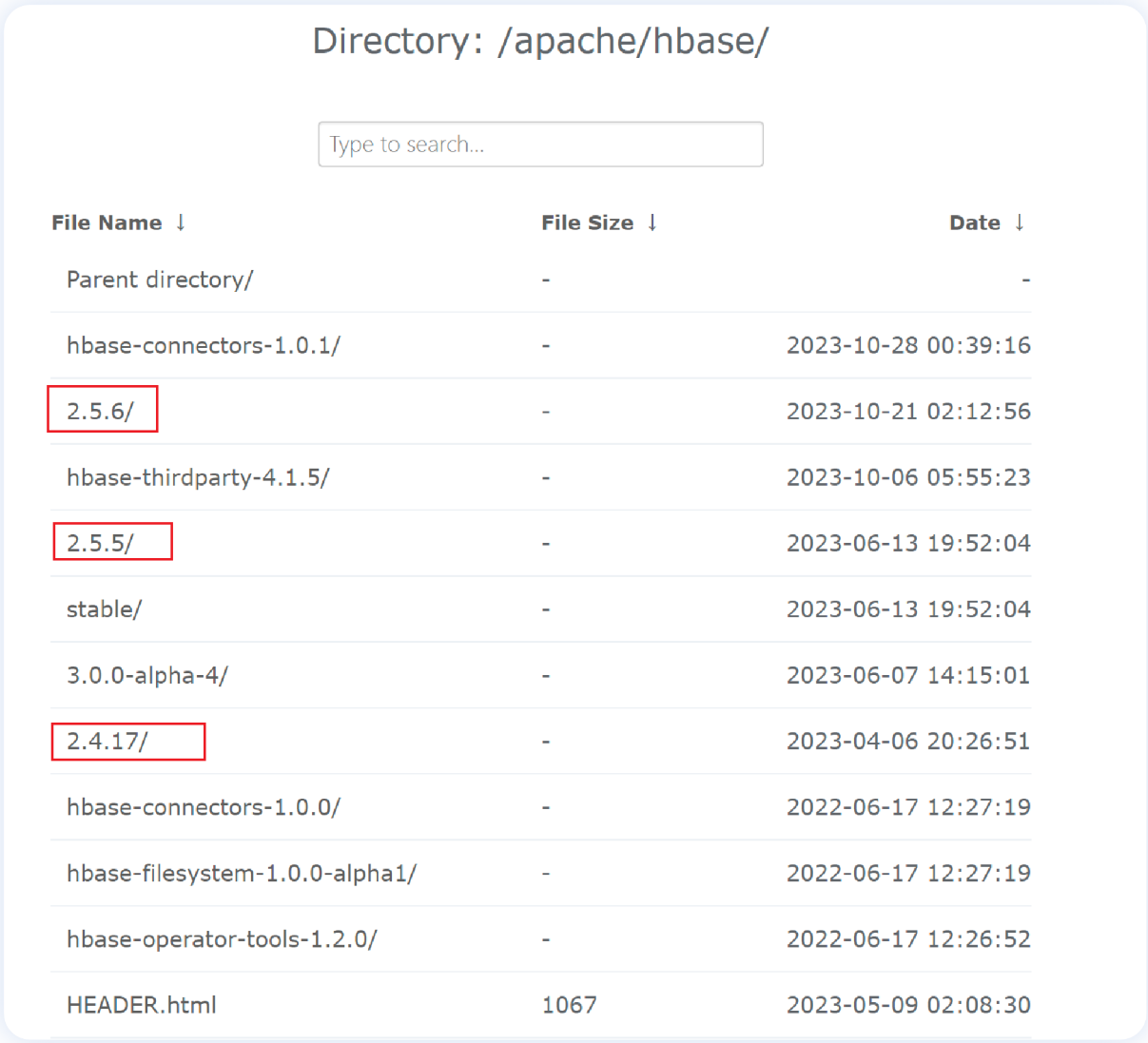
实验3

HBase安装

HBase下载

https://mirrors.nju.edu.cn/apache/hbase/2.4.8/hbase-2.4.8-bin.tar.gz有问题，查看https://mirrors.nju.edu.cn/apache/hbase/后发现现在只有如下版本，选择2.5.5版本

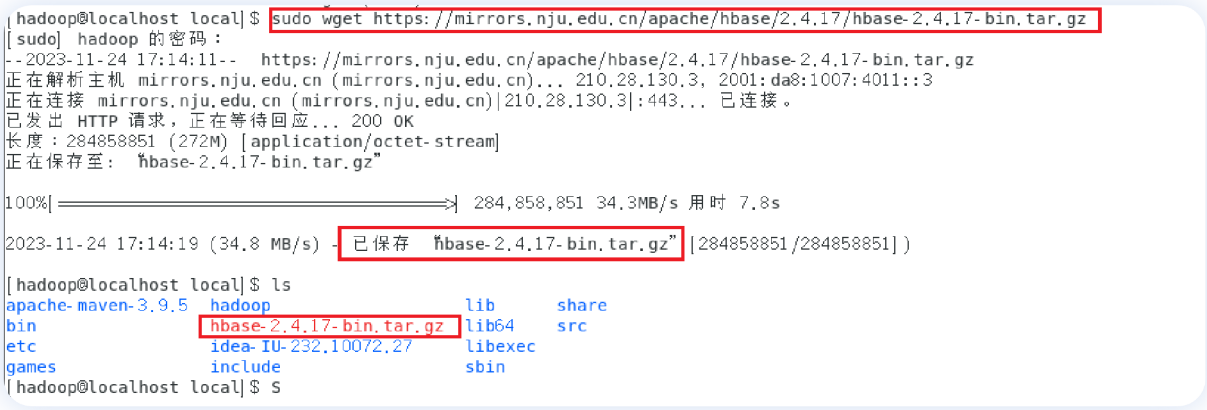


Directory: /apache/hbase/

Type to search...

File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
hbase-connectors-1.0.1/	-	2023-10-28 00:39:16
2.5.6/	-	2023-10-21 02:12:56
hbase-thirdparty-4.1.5/	-	2023-10-06 05:55:23
2.5.5/	-	2023-06-13 19:52:04
stable/	-	2023-06-13 19:52:04
3.0.0-alpha-4/	-	2023-06-07 14:15:01
2.4.17/	-	2023-04-06 20:26:51
hbase-connectors-1.0.0/	-	2022-06-17 12:27:19
hbase-filesystem-1.0.0-alpha1/	-	2022-06-17 12:27:19
hbase-operator-tools-1.2.0/	-	2022-06-17 12:26:52
HEADER.html	1067	2023-05-09 02:08:30

```
sudo wget https://mirrors.nju.edu.cn/apache/hbase/2.5.5/hbase-2.5.5-hadoop3-bin.tar.gz
```



解压：`sudo tar xzvf hbase-2.5.5-hadoop3-bin.tar.gz`

> `xzvf` 参数是解压tar.gz文件的参数

更改名称（hbase-2.5.5->hbase）

修改 `hbase-env.sh`

`vim hbase-env.sh`

添加如下内容：

```
#本机的java路径
export JAVA_HOME=/usr/lib/java-1.8.0/jdk1.8.0_381
#HBase 将管理自己的 ZooKeeper 实例
export HBASE_MANAGES_ZK=true
#为了避免 HBase 在启动时查找 Hadoop 的类路径，这一行很重要！！
export HBASE_DISABLE_HADOOP_CLASSPATH_LOOKUP="true"
```

修改 `hbase-site.xml`

`vim hbase-site.xml`

文件配置修改如下：

```
<!—修改成true 分布式—>
<property>
```

```
    <name>hbase.cluster.distributed</name>
    <value>true</value>
  </property>

  <property>
    <name>hbase.tmp.dir</name>
    <value>/usr/local/hbase/data/tmp</value>
  </property>

  <property>
    <name>hbase.unsafe.stream.capability.enforce</name>
    <value>false</value>
  </property>

  <!--HBase的数据保存在HDFS对应目录-->
  <property>
    <name>hbase.rootdir</name>
    <value>hdfs://localhost:9000/hbase</value>
  </property>

  <!--配置ZK的地址,有5个节点启用ZooKeeper-->
  <property>
     <name>hbase.zookeeper.quorum</name>
     <value>localhost:2181</value>
  </property>

<!--主节点和从节点允许的最大时间误差-->
<property>
  <name>hbase.master.maxclockskew</name>
  <value>180000</value>
</property>

<!--zookeeper数据目录-->
<property>
  <name>hbase.zookeeper.property.dataDir</name>
  <value>/usr/local/hbase/data/zookeeper</value>
</property>
```

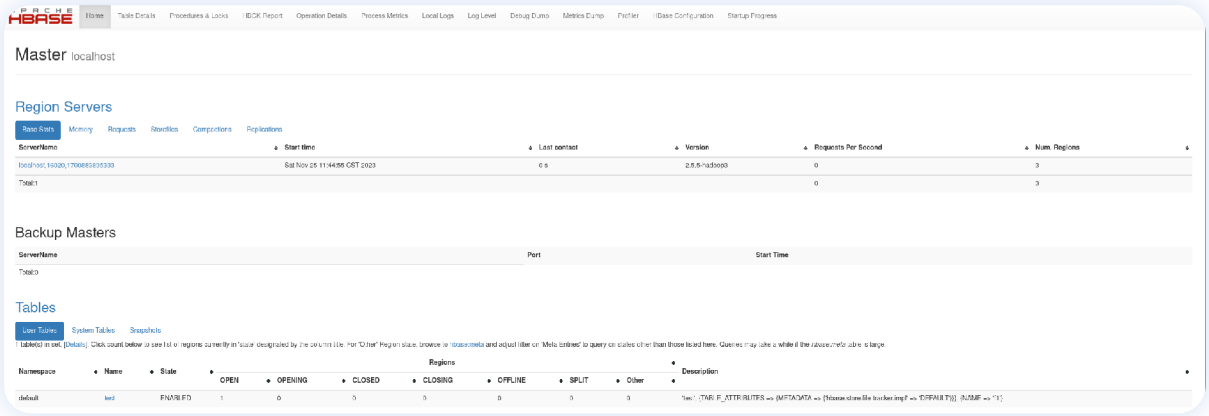
伪分布式运行

1. 先启动 `hadoop`
2. 启动 `Hbase`

   `./bin/start-hbase.sh`

```
hadoop@localhost ~]$ starthadoop
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [localhost.localdomain]
Starting resourcemanager
Starting nodemanagers
[hadoop@localhost ~]$ which hbase
/usr/local/hbase/bin/hbase
[hadoop@localhost ~]$ cd /usr/local/hbase/bin/
[hadoop@localhost bin]$ ./start-hbase.sh
localhost: running zookeeper, logging to /usr/local/hbase/bin/../logs/hbase-hadoop-zookeeper-localhost.localdomain.out
running master, logging to /usr/local/hbase/bin/../logs/hbase-hadoop-master-localhost.localdomain.out
: running regionserver, logging to /usr/local/hbase/bin/../logs/hbase-hadoop-regionserver-localhost.localdomain.out
[hadoop@localhost bin]$
```

```
[hadoop@localhost hbase]$ jps
9888 HMaster
9090 NodeManager
8451 DataNode
9763 HQuorumPeer
8309 NameNode
8709 SecondaryNameNode
10022 HRegionServer
8954 ResourceManager
11082 Jps
```



3. 关闭 `HBase`

   `./bin/stop-hbase.sh`

实验内容

Task1: 设计并创建合适的表

1. 进入shell

```
[hadoop@localhost hbase]$ hbase shell
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.5.5-hadoop3, r7ebd4381261fefd78fc2acf258a95184f4147cee, Thu Jun  1 17:59:44 PDT 2023
Took 0.0018 seconds
hbase:001:0>
```

2. 学生表的构建

```
hbase:006:0> create 'student', 'info'
Created table student
Took 2.2119 seconds
=> Hbase::Table - student
hbase:007:0> put 'student', '2120001', 'info:S_No', '2120001'
Took 0.2251 seconds
hbase:008:0> put 'student', '2120001', 'info:S_Name', 'Li Lei'
Took 0.0145 seconds
hbase:009:0> put 'student', '2120001', 'info:S_Sex', 'male'
Took 0.0139 seconds
hbase:010:0> put 'student', '2120001', 'info:S_Age', '20'
Took 0.0095 seconds
hbase:011:0> scan 'student'
ROW                                                   COLUMN+CELL
 2120001                                              column=info:S_Age, timestamp=2023-11-25T12:37:11.060, value=20
 2120001                                              column=info:S_Name, timestamp=2023-11-25T12:36:56.756, value=Li Lei
 2120001                                              column=info:S_No, timestamp=2023-11-25T12:36:40.139, value=2120001
 2120001                                              column=info:S_Sex, timestamp=2023-11-25T12:37:03.998, value=male
 row(s)
```

- 创建表 `create '<table_name>','<列族名>'`
- 添加一列 `put <table>,<rowkey>,<family:column>,<value>`
- 查看表： `scan '<table_name>'`

```
hbase:023:0> scan 'student'
ROW                                                   COLUMN+CELL
 2120001                                              column=info:S_Age, timestamp=2023-11-25T12:37:11.060, value=20
 2120001                                              column=info:S_Name, timestamp=2023-11-25T12:36:56.756, value=Li Lei
 2120001                                              column=info:S_No, timestamp=2023-11-25T12:36:40.139, value=2120001
 2120001                                              column=info:S_Sex, timestamp=2023-11-25T12:37:03.998, value=male
 2120002                                              column=info:S_Age, timestamp=2023-11-25T12:44:36.485, value=21
 2120002                                              column=info:S_Name, timestamp=2023-11-25T12:44:36.462, value=Han Meimei
 2120002                                              column=info:S_No, timestamp=2023-11-25T12:44:36.445, value=2120002
 2120002                                              column=info:S_Sex, timestamp=2023-11-25T12:44:36.476, value=female
 2120003                                              column=info:S_Age, timestamp=2023-11-25T12:45:00.630, value=20
 2120003                                              column=info:S_Name, timestamp=2023-11-25T12:45:00.595, value=Zhang Li
 2120003                                              column=info:S_No, timestamp=2023-11-25T12:45:00.572, value=2120003
 2120003                                              column=info:S_Sex, timestamp=2023-11-25T12:45:00.617, value=female
 2120004                                              column=info:S_Age, timestamp=2023-11-25T12:45:38.830, value=19
 2120004                                              column=info:S_Name, timestamp=2023-11-25T12:45:38.819, value=Li Ming
 2120004                                              column=info:S_No, timestamp=2023-11-25T12:45:38.796, value=2120004
 2120004                                              column=info:S_Sex, timestamp=2023-11-25T12:45:38.822, value=male
 row(s)
```

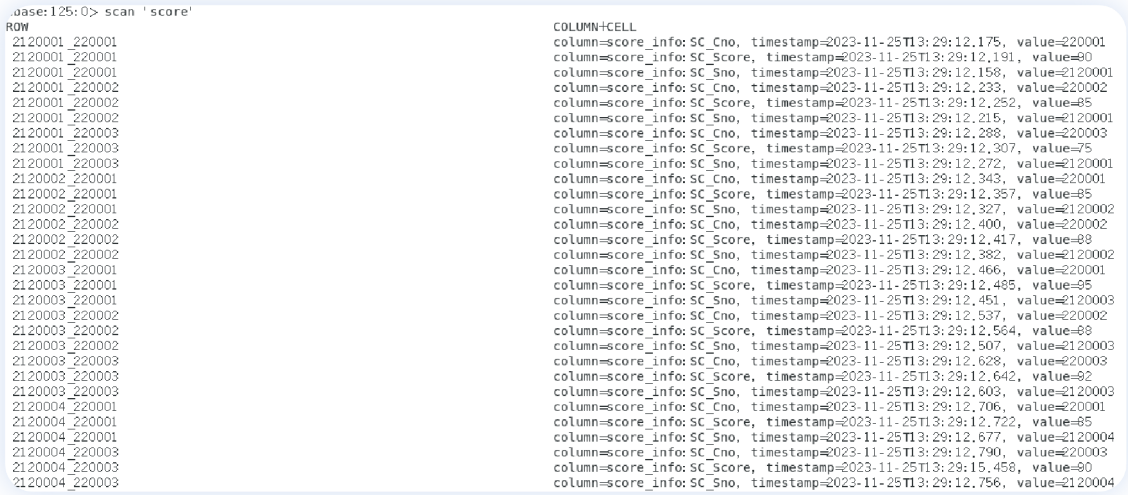
3. 创建课程表

与创建学生表类似

```
ROW                                                   COLUMN+CELL
 220001                                               column=course_info:C_Credit, timestamp=2023-11-25T13:32:19.382, value=4.0
 220001                                               column=course_info:C_Name, timestamp=2023-11-25T13:32:19.367, value=Math
 220001                                               column=course_info:C_No, timestamp=2023-11-25T13:32:19.330, value=220001
 220002                                               column=course_info:C_Credit, timestamp=2023-11-25T13:32:19.431, value=3.0
 220002                                               column=course_info:C_Name, timestamp=2023-11-25T13:32:19.417, value=English
 220002                                               column=course_info:C_No, timestamp=2023-11-25T13:32:19.400, value=220002
 220003                                               column=course_info:C_Credit, timestamp=2023-11-25T13:32:20.744, value=4.0
 220003                                               column=course_info:C_Name, timestamp=2023-11-25T13:32:19.473, value=BigData
 220003                                               column=course_info:C_No, timestamp=2023-11-25T13:32:19.459, value=220003
 row(s)
```

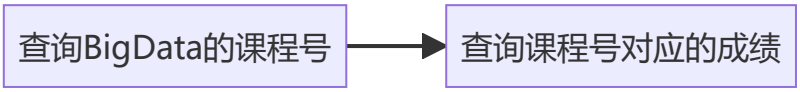
4. 创建选课表

> 注意，这里第一列不是唯一的，因此不能作为rowkey，这里我是用的是 `<SC_Sno>_<SC_Cno>` 保证rowkey的唯一性



Task2:查询选修BigData的学生的成绩

HBase是一个面向列的分布式数据库，因此它不支持内部关联查询。所以本任务使用多次使用过滤器查询数据

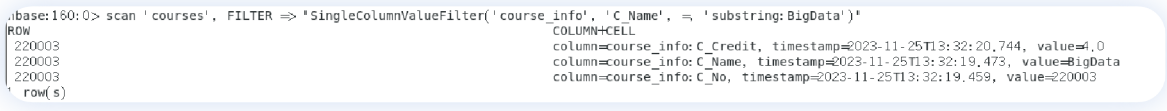


比较器	表达式缩写
BinaryComparator	binary:值
BinaryPrefixComparator	binaryprefix:值
BitComparator	bit:值
NullComparator	null
RegexStringComparator	regexstring:正则表达式
SubstringComparator	substring:值

1. 查询BigData的课程号

   这里使用值过滤器和子字符串比较器

```
scan 'courses', FILTER => "SingleColumnValueFilter('course_info',
'C_Name', =, 'substring:BigData')"
```



可以看到，BigData的课程号是220003

2. 查询220003的成绩

```
base:161:0> scan 'score', {FILTER => "SingleColumnValueFilter('score_info', 'SC_Cno', =, 'binary:220003')"}
ROW                                                    COLUMN+CELL
 2120001_220003                                        column=score_info:SC_Cno, timestamp=2023-11-25T13:29:12.288, value=220003
 2120001_220003                                        column=score_info:SC_Score, timestamp=2023-11-25T13:29:12.307, value=75
 2120001_220003                                        column=score_info:SC_Sno, timestamp=2023-11-25T13:29:12.272, value=2120001
 2120003_220003                                        column=score_info:SC_Cno, timestamp=2023-11-25T13:29:12.628, value=220003
 2120003_220003                                        column=score_info:SC_Score, timestamp=2023-11-25T13:29:12.642, value=92
 2120003_220003                                        column=score_info:SC_Sno, timestamp=2023-11-25T13:29:12.603, value=2120003
 2120004_220003                                        column=score_info:SC_Cno, timestamp=2023-11-25T13:29:12.790, value=220003
 2120004_220003                                        column=score_info:SC_Score, timestamp=2023-11-25T13:29:15.458, value=90
 2120004_220003                                        column=score_info:SC_Sno, timestamp=2023-11-25T13:29:12.756, value=2120004
 row(s)
```

可以看到

- 学号为2120001的同学成绩为75
- 学号为2120003的同学成绩为92
- 学号为2120004的同学成绩为94

3. 根据学号查询姓名

多次使用 `get 'student', '2120001'` 获取每位同学的姓名，最后可以看到

```
base:163:0> get 'student', '2120001'
COLUMN                                                 CELL
 info:S_Age                                            timestamp=2023-11-25T12:37:11.060, value=20
 info:S_Name                                           timestamp=2023-11-25T12:36:56.756, value=Li Lei
 info:S_No                                             timestamp=2023-11-25T12:36:40.139, value=2120001
 info:S_Sex                                            timestamp=2023-11-25T12:37:03.998, value=male
 row(s)
```

- Li Lei的同学成绩为75
- Zhang Li的同学成绩为92
- Li Ming的同学成绩为94

Task3:学生表增加电子邮件列

直接多次put即可。

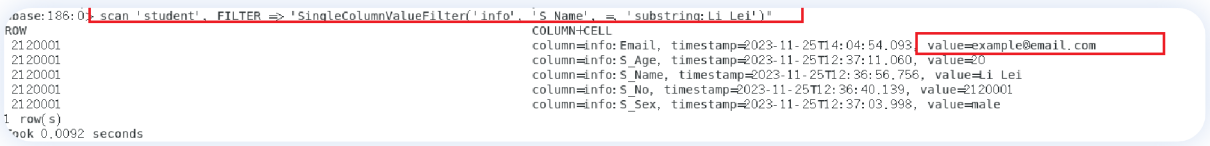
```
base:168:0> put 'student', '2120002', 'info:Email', 'example2@email.com'
Took 0.0040 seconds
hbase:169:0> put 'student', '2120003', 'info:Email', 'example2@email.com'
Took 0.0045 seconds
hbase:170:0> put 'student', '2120003', 'info:Email', 'example3@email.com'
Took 0.0049 seconds
hbase:171:0> put 'student', '2120004', 'info:Email', 'example4@email.com'
Took 0.0049 seconds
hbase:172:0> scan 'student'
ROW                                                    COLUMN+CELL
 2120001                                               column=info:Email, timestamp=2023-11-25T14:04:54.093, value=example@email.com
 2120001                                               column=info:S_Age, timestamp=2023-11-25T12:37:11.060, value=20
 2120001                                               column=info:S_Name, timestamp=2023-11-25T12:36:56.756, value=Li Lei
 2120001                                               column=info:S_No, timestamp=2023-11-25T12:36:40.139, value=2120001
 2120001                                               column=info:S_Sex, timestamp=2023-11-25T12:37:03.998, value=male
 2120002                                               column=info:Email, timestamp=2023-11-25T14:05:23.316, value=example2@email.com
 2120002                                               column=info:S_Age, timestamp=2023-11-25T12:44:36.485, value=21
 2120002                                               column=info:S_Name, timestamp=2023-11-25T12:44:36.462, value=Han Meimei
 2120002                                               column=info:S_No, timestamp=2023-11-25T12:44:36.445, value=2120002
 2120002                                               column=info:S_Sex, timestamp=2023-11-25T12:44:36.476, value=female
 2120003                                               column=info:Email, timestamp=2023-11-25T14:05:34.549, value=example3@email.com
 2120003                                               column=info:S_Age, timestamp=2023-11-25T12:45:00.630, value=20
 2120003                                               column=info:S_Name, timestamp=2023-11-25T12:45:00.595, value=Zhang Li
 2120003                                               column=info:S_No, timestamp=2023-11-25T12:45:00.572, value=2120003
 2120003                                               column=info:S_Sex, timestamp=2023-11-25T12:45:00.617, value=female
 2120004                                               column=info:Email, timestamp=2023-11-25T14:05:41.497, value=example4@email.com
 2120004                                               column=info:S_Age, timestamp=2023-11-25T12:45:38.830, value=19
 2120004                                               column=info:S_Name, timestamp=2023-11-25T12:45:38.815, value=Li Ming
 2120004                                               column=info:S_No, timestamp=2023-11-25T12:45:38.796, value=2120004
 2120004                                               column=info:S_Sex, timestamp=2023-11-25T12:45:38.822, value=male
4 row(s)
Took 0.0100 seconds
```

如果要新增一个列族，那么可以先用altert添加一个列族名，然后再put也可以

```
alter 'student', {NAME => 'Email'}
```

Task4:查询Li Lei的电子邮件；

该任务比较简单，就与Task2中的过滤相同使用即可，可以看到Li Lei 的电子邮箱是example@email.com



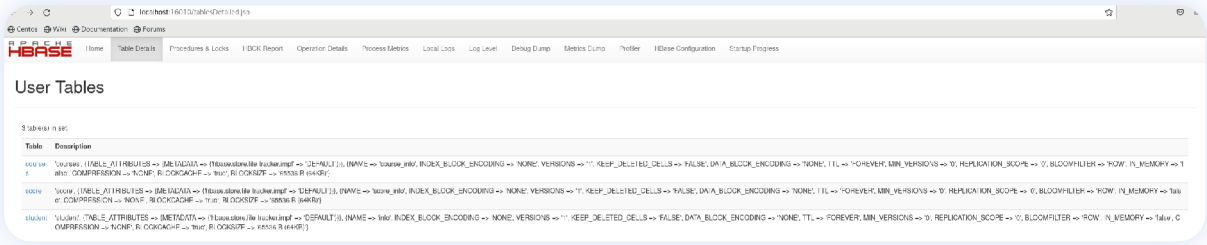
Task5: 删除所创建的表

这题就更简单了，正常使用语法即可

> 删除表操作：
>
> ```
> disable <table>
> drop <table>
> ```

```
hbase:187:0> list
TABLE
courses
score
student
test
4 row(s)
Took 0.0064 seconds
=> ["courses", "score", "student", "test"]
hbase:188:0> disable 'test'
Took 0.3293 seconds
hbase:189:0> drop 'test'
Took 0.1181 seconds
hbase:190:0> list
TABLE
courses
score
student
3 row(s)
Took 0.0053 seconds
=> ["courses", "score", "student"]
hbase:191:0>
```

Web UI截图



问题总结及解决方案

关于磁盘扩容问题

在之前我的/usr文件夹内内存不足，如果只用vmvare的内存分配也不能将内存正确扩充到我的工作文件夹下，因此参考了这篇文章，成功扩容。

Linux CentOS 7分配的磁盘空间不足，空间扩展方法，保姆级操作

Tips1:创建快捷启动方式：

修改环境变量：

```
sudo vim ~/.bashrc
```

添加如下别名：

```
alias starthadoop='/usr/local/hadoop/sbin/start-all.sh'
```

就可以快捷启动hadoop

Tips2:快速查找文件路径

可以用which快速查询路径

```
which hadoop
```

关于权限问题（很重要！）

之前配置的时候没有给hadoop用户设置管理员权限，在操作的时候都用sodo，这样在后续启动HBase的时候会出现一点问题。因此，**强烈建议**给用户设置hbase的管理员权限

```
sudo chown -R hadoop:hadoop hbase
```

- `chown` 是改变文件或目录所有者的命令。
- `-R` 选项表示递归地应用于目录及其所有子目录和文件。
- `hadoop:hadoop` 是目标所有者和所属组的标识。在这里，文件/目录的所有者将设置为 `hadoop` 用户，所属组也将设置为 `hadoop` 组。
- `hbase` 是要更改所有者的目标文件或目录的名称。

关于HBase和Hadoop冲突问题

之前运行的时候一直会出现SLF4J冲突什么的。后来解决方案是在env中添加了

```
export HBASE_DISABLE_HADOOP_CLASSPATH_LOOKUP="true"
```

好像就好使了

其他思考

HBase是一个基于Hadoop生态的分布式列型数据库,具有高可靠性、高并发读写和易扩展等优点,适合PB级大数据的存储。原因在于它基于HDFS文件系统实现了数据的高可靠存储,利用HBase的分布式架构可以对数据进行自动分区,实现对大规模数据的横向扩展。同时,它通过MemStore和SSTable的结构优化了对列式数据的存储 efficiency 和读取性能。这使得它可以提供每秒数十万次的读写吞吐。

但是,HBase作为一个非关系型NoSQL数据库,它也存在一定的缺陷。比如部署和使用复杂度较高,需要搭建整个Hadoop生态圈,不支持SQL,学习和使用成本较高。此外,HBase无法提供强一致性保证,在极端情况下可能会出现读取不一致数据的问题。而且HBase的读放大问题也会对某些读密集场景的性能产生一定影响。