福建師範大學
FUJIAN NORMAL UNIVERSITY

计算机与网络空间安全学院学生实验报告

实验课程名称：　　大数据导论　　教师：　　林鑫泓

实验名称	熟悉常用的 HBase 操作			实验成绩	
学生姓名	叶建行	学　号	116052020005	年级专业班级	2020 级软件工程一班
小组成员	无			实验日期	2022 年 10 月 21

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：Linux（建议 Ubuntu18.04），
- Hadoop 版本：建议 3.2.1，
- HBase 版本：建议 2.2.6，
- JDK 版本：openjdk-8。

1.3 实验要求

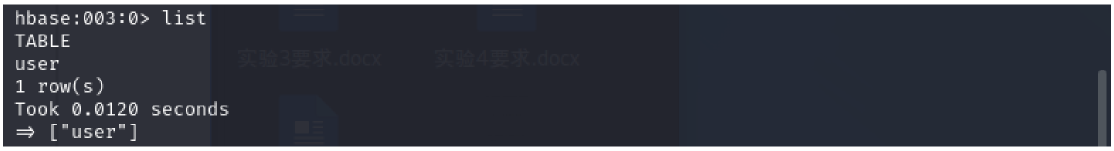
- 理解 HBase 在 Hadoop 体系结构中的角色，
- 熟练使用 HBase 操作常用的 Shell 命令，
- 熟悉 HBase 操作常用的 Java API。

2 实验记录

2.1 Linux 操作

（一）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：

（1） 列出 HBase 所有的表的相关信息，例如表名；



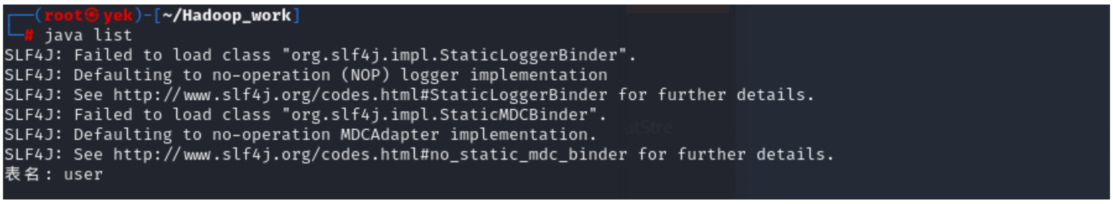
```
hbase:003:0> list
TABLE
user
1 row(s)
Took 0.0120 seconds
=> ["user"]
```

实验 java 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 java.io.IOException;
import java.util.List;

public class list {
    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;
    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(null != connection){
                connection.close();
```

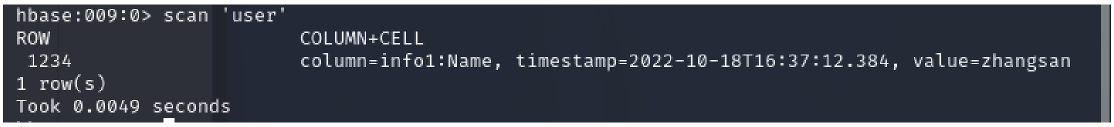
```
              }
          }catch (IOException e){
              e.printStackTrace();
          }
      }

      public static void listTables() throws IOException {
          init();
          List<TableDescriptor> tableDescriptors =
admin.listTableDescriptors();
          for(TableDescriptor tableDescriptor : tableDescriptors){
              TableName tableName = tableDescriptor.getTableName();
              System.out.println("表名: " + tableName);
          }
          close();
      }
      public static void main(String[] args) {
          try {
              listTables();
          } catch (IOException e) {
              throw new RuntimeException(e);
          }
      }
  }
```

实验结果：



```
┌──(root㉿yek)-[~/Hadoop_work]
└─# java list
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
表名: user
```

（2）　在终端打印出指定的表的所有记录数据；

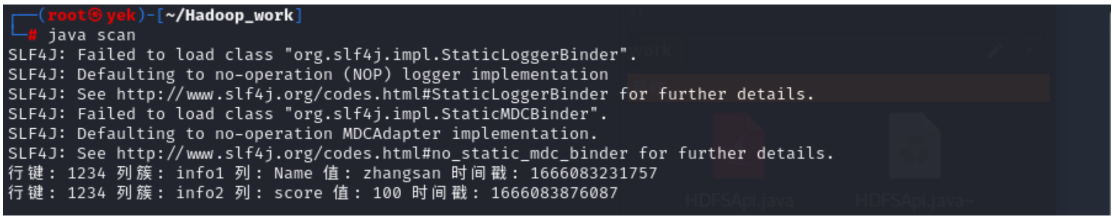


```
hbase:009:0> scan 'user'
ROW                    COLUMN+CELL
 1234                  column=info1:Name, timestamp=2022-10-18T16:37:12.384, value=zhangsan
1 row(s)
Took 0.0049 seconds
```

实验 Java 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 java.io.IOException;
  import java.util.List;

  public class scan {
      public static Configuration configuration;
      public static Connection connection;
      public static Admin admin;

      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 (null != connection) {
                  connection.close();
              }
          } catch (IOException e) {
              e.printStackTrace();
          }
      }

      public static void getData(String tableName) throws IOException {
          init();
          Table table =
connection.getTable(TableName.valueOf(tableName));
          Scan scan = new Scan();
          ResultScanner scanner = table.getScanner(scan);
          for (Result result : scanner) {
              printRecoder(result);
          }
          close();
      }

   public  static void printRecoder(Result result)throws IOException{
       for(Cell cell:result.rawCells()){
              System.out.print(行键: "+new
String(Bytes.toString(cell.getRowArray(),cell.getRowOffset(),
cell.getRowLength())));
              System.out.print(" 列簇: "+new
String(Bytes.toString(cell.getFamilyArray(),cell.getFamilyOffset(),
cell.getFamilyLength()) ));
              System.out.print(" 列: "+new
String(Bytes.toString(cell.getQualifierArray(),cell.getQualifierOffset(
), cell.getQualifierLength())));
              System.out.print(" 值: "+new
String(Bytes.toString(cell.getValueArray(),cell.getValueOffset(),
cell.getValueLength())));
              System.out.println(" 时间戳:
"+cell.getTimestamp());
          }
      }

      public static void main(String[] args) {
          try {
              getData("user");
          } catch (IOException e) {
              throw new RuntimeException(e);
          }
      }
  }
```

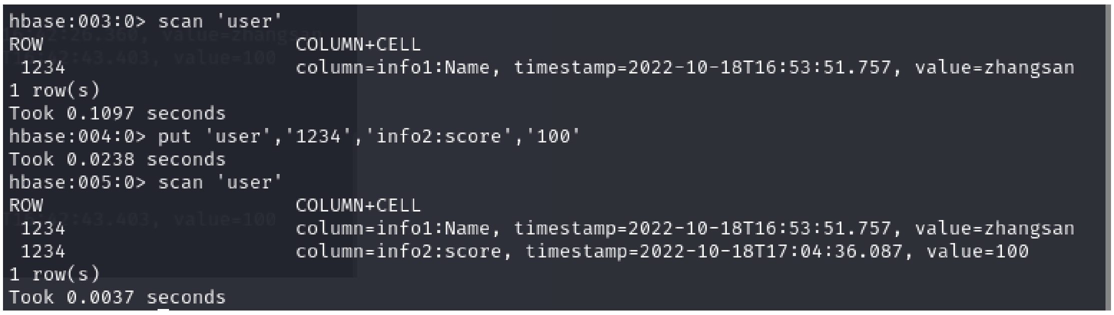
实验结果：



```
┌──(root㉿yek)-[~/Hadoop_work]
└─# java scan
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
行键: 1234 列簇: info1 列: Name 值: zhangsan 时间戳: 1666083231757
行键: 1234 列簇: info2 列: score 值: 100 时间戳: 1666083876087
```

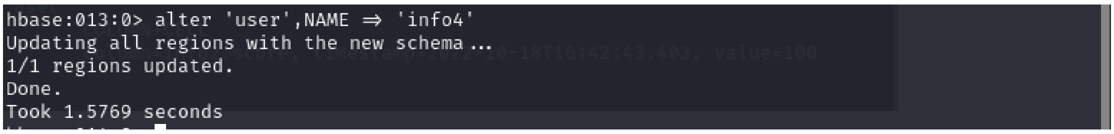
（3） 向已经创建好的表添加和删除指定的列族或列；

添加指定列：



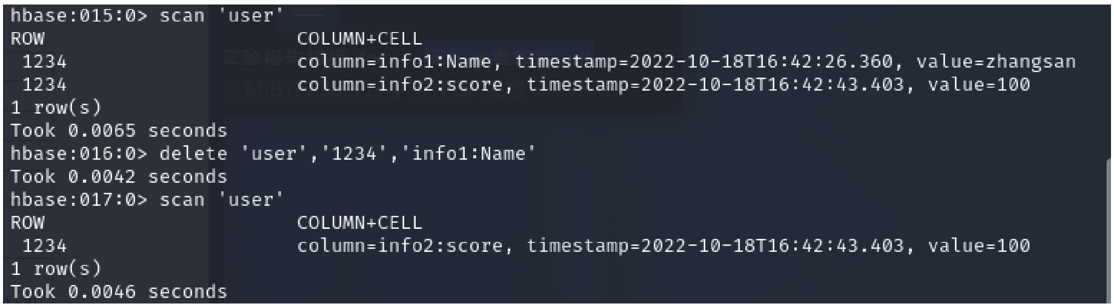
```
hbase:003:0> scan 'user'
ROW                         COLUMN+CELL
 1234                       column=info1:Name, timestamp=2022-10-18T16:53:51.757, value=zhangsan
1 row(s)
Took 0.1097 seconds
hbase:004:0> put 'user','1234','info2:score','100'
Took 0.0238 seconds
hbase:005:0> scan 'user'
ROW                         COLUMN+CELL
 1234                       column=info1:Name, timestamp=2022-10-18T16:53:51.757, value=zhangsan
 1234                       column=info2:score, timestamp=2022-10-18T17:04:36.087, value=100
1 row(s)
Took 0.0037 seconds
```

添加指定列族：



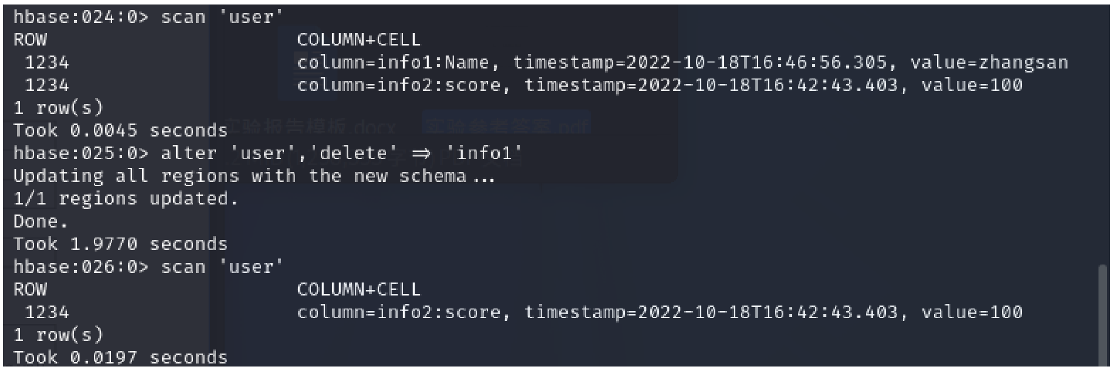
```
hbase:013:0> alter 'user',NAME => 'info4'
Updating all regions with the new schema...
1/1 regions updated.
Done.
Took 1.5769 seconds
```

删除指定列：



```
hbase:015:0> scan 'user'
ROW                      COLUMN+CELL
 1234                    column=info1:Name, timestamp=2022-10-18T16:42:26.360, value=zhangsan
 1234                    column=info2:score, timestamp=2022-10-18T16:42:43.403, value=100
1 row(s)
Took 0.0065 seconds
hbase:016:0> delete 'user','1234','info1:Name'
Took 0.0042 seconds
hbase:017:0> scan 'user'
ROW                      COLUMN+CELL
 1234                    column=info2:score, timestamp=2022-10-18T16:42:43.403, value=100
1 row(s)
Took 0.0046 seconds
```

删除指定列族：



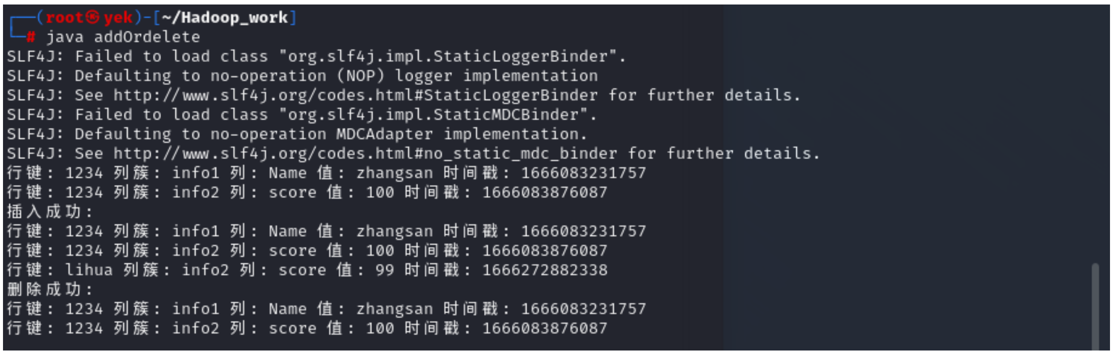
```
hbase:024:0> scan 'user'
ROW                      COLUMN+CELL
 1234                    column=info1:Name, timestamp=2022-10-18T16:46:56.305, value=zhangsan
 1234                    column=info2:score, timestamp=2022-10-18T16:42:43.403, value=100
1 row(s)
Took 0.0045 seconds
hbase:025:0> alter 'user','delete' => 'info1'
Updating all regions with the new schema...
1/1 regions updated.
Done.
Took 1.9770 seconds
hbase:026:0> scan 'user'
ROW                      COLUMN+CELL
 1234                    column=info2:score, timestamp=2022-10-18T16:42:43.403, value=100
1 row(s)
Took 0.0197 seconds
```

实验 Java 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 java.io.IOException;
import java.util.List;

public class addOrdelete {
    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;
    public static void init(){
        configuration  = HBaseConfiguration.create();
        configuration.set("hbase.rootdir","hdfs://localhost:9000/hbase
");
```

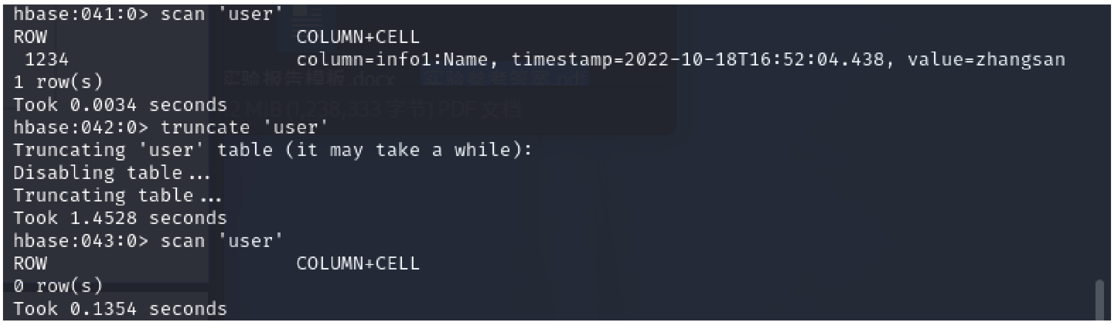
```
          try{
              connection =
ConnectionFactory.createConnection(configuration);
              admin = connection.getAdmin();
          }catch (IOException e){
              e.printStackTrace();
          }
      }
      public static void close(){
          try{
              if(admin != null){
                  admin.close();
              }
              if(null != connection){
                  connection.close();
              }
          }catch (IOException e){
              e.printStackTrace();
          }
      }
       public static void insertRow(String tableName,String
rowKey,String colFamily,String col,String val) throws IOException {
          init();
          Table table =
connection.getTable(TableName.valueOf(tableName));
          Put put = new Put(rowKey.getBytes());
          put.addColumn(colFamily.getBytes(), col.getBytes(),
val.getBytes());
          table.put(put);
          table.close();
          close();
      }
      public static void deleteRow(String tableName,String rowKey,String
colFamily,String col) throws IOException {
          init();
          Table table =
connection.getTable(TableName.valueOf(tableName));
          Delete delete = new Delete(rowKey.getBytes());
          delete.addFamily(Bytes.toBytes(colFamily));
          delete.addColumn(Bytes.toBytes(colFamily),Bytes.toBytes(col));
          table.delete(delete);
          table.close();
          close();
      }
      public static void getData(String tableName)throws  IOException{
          init();
          Table table =
connection.getTable(TableName.valueOf(tableName));
          Scan scan = new Scan();
          ResultScanner scanner = table.getScanner(scan);
          for (Result result:scanner){
              printRecoder(result);
          }
          close();
      }
      public  static void printRecoder(Result result)throws IOException{
          for(Cell cell:result.rawCells()){
              System.out.print("行键: "+new
String(Bytes.toString(cell.getRowArray(),cell.getRowOffset(),
cell.getRowLength())));
              System.out.print(" 列簇: "+new
String( Bytes.toString(cell.getFamilyArray(),cell.getFamilyOffset(),
cell.getFamilyLength()) ));
              System.out.print(" 列: "+new
String(Bytes.toString(cell.getQualifierArray(),cell.getQualifierOffset(
), cell.getQualifierLength())));
              System.out.print(" 值: "+new
String(Bytes.toString(cell.getValueArray(),cell.getValueOffset(),
cell.getValueLength())));
              System.out.println(" 时间戳:
"+cell.getTimestamp());
          }
      }
      public static void main(String[] args) {
          try {
              getData("user");
              insertRow("user","lihua","info2","score","99");
              System.out.println("插入成功: ");
              getData("user");
              deleteRow("user","lihua","info2","score");
              System.out.println("删除成功: ");
              getData("user");
          } catch (IOException e) {
              throw new RuntimeException(e);
          }
      }
  }
```

实验结果：



```
┌──(root㉿yek)-[~/Hadoop_work]
└─# java addOrdelete
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
行键: 1234 列簇: info1 列: Name 值: zhangsan 时间戳: 1666083231757
行键: 1234 列簇: info2 列: score 值: 100 时间戳: 1666083876087
插入成功:
行键: 1234 列簇: info1 列: Name 值: zhangsan 时间戳: 1666083231757
行键: 1234 列簇: info2 列: score 值: 100 时间戳: 1666083876087
行键: lihua 列簇: info2 列: score 值: 99 时间戳: 1666272882338
删除成功:
行键: 1234 列簇: info1 列: Name 值: zhangsan 时间戳: 1666083231757
行键: 1234 列簇: info2 列: score 值: 100 时间戳: 1666083876087
```

（4）　清空指定的表的所有记录数据；

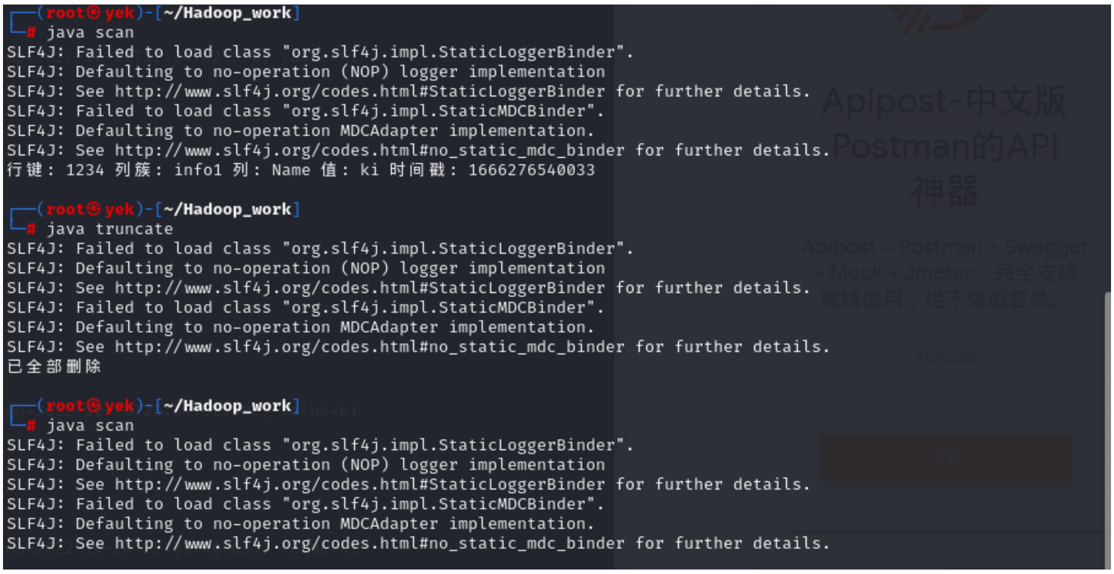


```
hbase:041:0> scan 'user'
ROW                        COLUMN+CELL
 1234                      column=info1:Name, timestamp=2022-10-18T16:52:04.438, value=zhangsan
1 row(s)
Took 0.0034 seconds
hbase:042:0> truncate 'user'
Truncating 'user' table (it may take a while):
Disabling table...
Truncating table...
Took 1.4528 seconds
hbase:043:0> scan 'user'
ROW                        COLUMN+CELL
0 row(s)
Took 0.1354 seconds
```

实验 Java 代码：

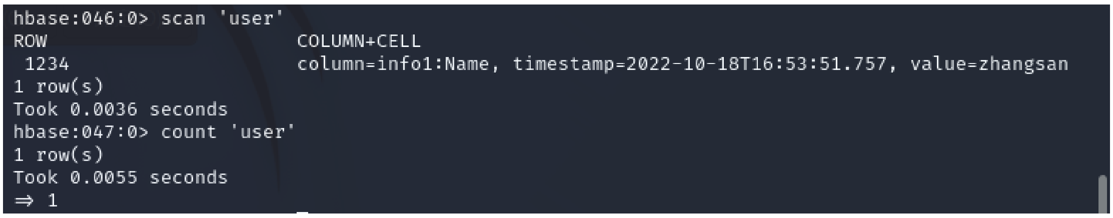
```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.*;
import org.apache.hadoop.hbase.client.*;
import org.apache.hadoop.hbase.util.Bytes;
import java.io.IOException;
import java.util.List;
import java.util.ArrayList;

public class truncate {
    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;
    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(null != connection){
                  connection.close();
              }
          }catch (IOException e){
              e.printStackTrace();
          }
      }
        public static void clearRows(String tableName)throws
IOException{
          init();
          TableName tablename = TableName.valueOf(tableName);
          Table table = connection.getTable(tablename);

          HTableDescriptor tableDescriptor1 = table.getTableDescriptor();
          HColumnDescriptor[] columnFamilies =
tableDescriptor1.getColumnFamilies();

            // 创建集合用于存放ColumnFamilyDescriptor 对象
          List<ColumnFamilyDescriptor> families = new ArrayList<>();
          // 将每个familyName 对应的ColumnFamilyDescriptor 对象添加到
families 集合中保存
          for (HColumnDescriptor columnFamily : columnFamilies){
                byte[] name = columnFamily.getName();
                String value = Bytes.toString(name);
                families.add(ColumnFamilyDescriptorBuilder.newBuilder(va
lue.getBytes()).build());
          }
          admin.disableTable(tablename);
          admin.deleteTable(tablename);
          TableDescriptorBuilder tableDescriptor =
TableDescriptorBuilder.newBuilder(tablename);
          admin.createTable(tableDescriptor.setColumnFamilies(families).
build());
          close();
      }
      public static void main(String[] args) {
          try {
              clearRows("user");
              System.out.println("已全部删除");
          } catch (IOException e) {
              throw new RuntimeException(e);
          }
      }
  }
```

实验结果：



```
┌──(root㉿yek)-[~/Hadoop_work]
└─# java scan
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
行键: 1234 列簇: info1 列: Name 值: ki 时间戳: 1666276540033

┌──(root㉿yek)-[~/Hadoop_work]
└─# java truncate
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
已全部删除

┌──(root㉿yek)-[~/Hadoop_work]
└─# java scan
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
```

（5） 统计表的行数。



```
hbase:046:0> scan 'user'
ROW                      COLUMN+CELL
 1234                    column=info1:Name, timestamp=2022-10-18T16:53:51.757, value=zhangsan
1 row(s)
Took 0.0036 seconds
hbase:047:0> count 'user'
1 row(s)
Took 0.0055 seconds
⇒ 1
```

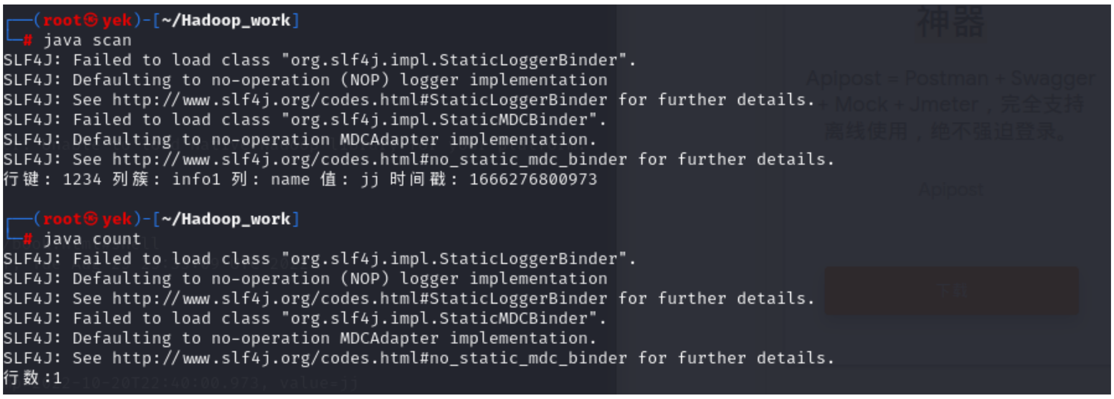
实验 Java 代码：

```
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 java.io.IOException;
import java.util.List;
import java.util.ArrayList;

public class count {
    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;
```

```
      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(null != connection){
                  connection.close();
              }
          }catch (IOException e){
              e.printStackTrace();
          }
      }
         public static void countRows(String tableName)throws
IOException{
          init();
          Table table =
connection.getTable(TableName.valueOf(tableName));
          Scan scan = new Scan();
          ResultScanner scanner = table.getScanner(scan);
          int num = 0;
          for (Result result =
scanner.next();result!=null;result=scanner.next()){
              num++;
          }
          System.out.println("行数:"+ num);
          scanner.close();
          close();
      }
      public static void main(String[] args) {
          try {
              countRows("user");
```

```
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}
```

实验结果：



```
┌──(root㉿yek)-[~/Hadoop_work]
└─# java scan
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
行键: 1234 列簇: info1 列: name 值: jj 时间戳: 1666276800973

┌──(root㉿yek)-[~/Hadoop_work]
└─# java count
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
行数:1
```

（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

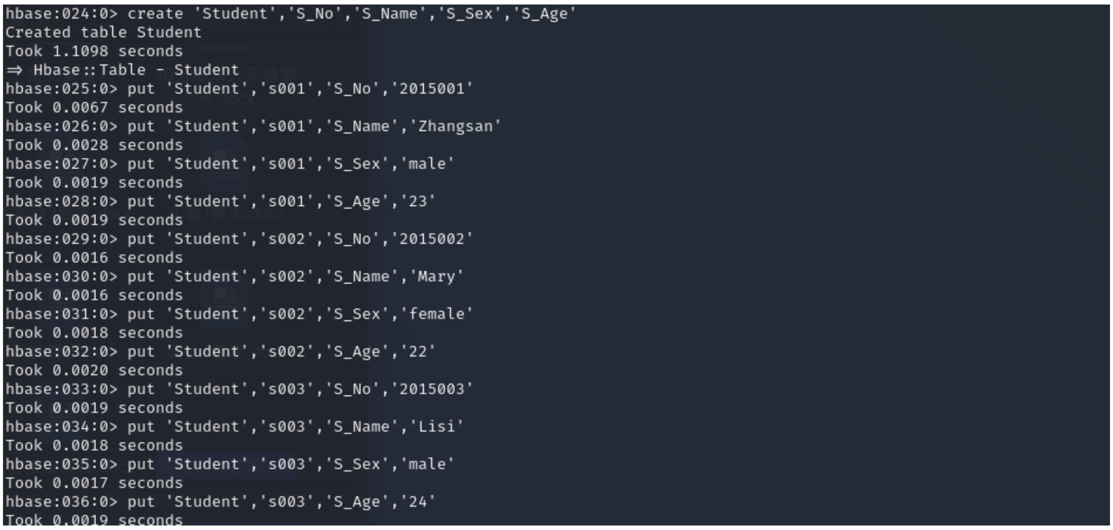
选课表（SC）

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

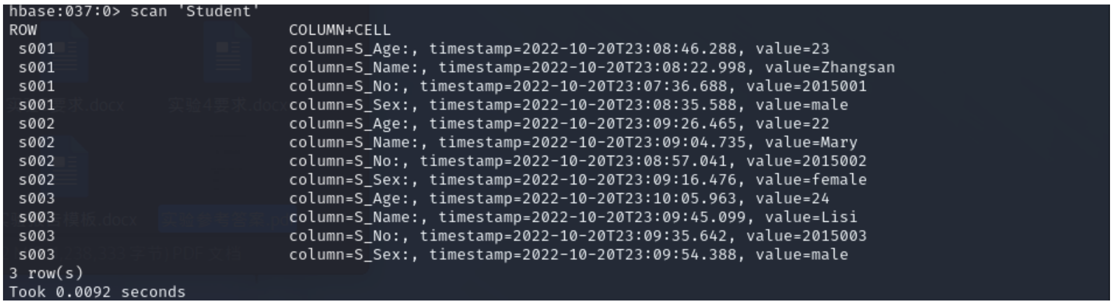
2015003	123001	98
2015003	123002	95

实验代码及结果：

学生表：

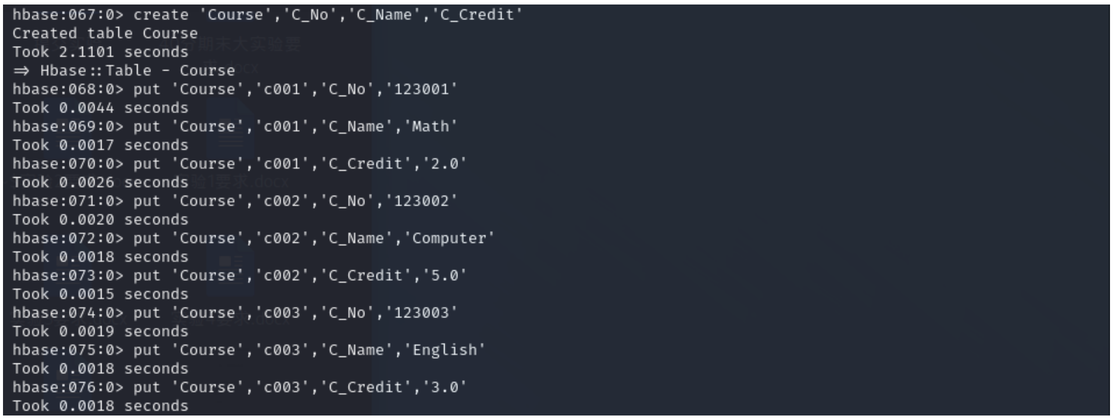


```
hbase:024:0> create 'Student','S_No','S_Name','S_Sex','S_Age'
Created table Student
Took 1.1098 seconds
=> Hbase::Table - Student
hbase:025:0> put 'Student','s001','S_No','2015001'
Took 0.0067 seconds
hbase:026:0> put 'Student','s001','S_Name','Zhangsan'
Took 0.0028 seconds
hbase:027:0> put 'Student','s001','S_Sex','male'
Took 0.0019 seconds
hbase:028:0> put 'Student','s001','S_Age','23'
Took 0.0019 seconds
hbase:029:0> put 'Student','s002','S_No','2015002'
Took 0.0016 seconds
hbase:030:0> put 'Student','s002','S_Name','Mary'
Took 0.0016 seconds
hbase:031:0> put 'Student','s002','S_Sex','female'
Took 0.0018 seconds
hbase:032:0> put 'Student','s002','S_Age','22'
Took 0.0020 seconds
hbase:033:0> put 'Student','s003','S_No','2015003'
Took 0.0019 seconds
hbase:034:0> put 'Student','s003','S_Name','Lisi'
Took 0.0018 seconds
hbase:035:0> put 'Student','s003','S_Sex','male'
Took 0.0017 seconds
hbase:036:0> put 'Student','s003','S_Age','24'
Took 0.0019 seconds
```

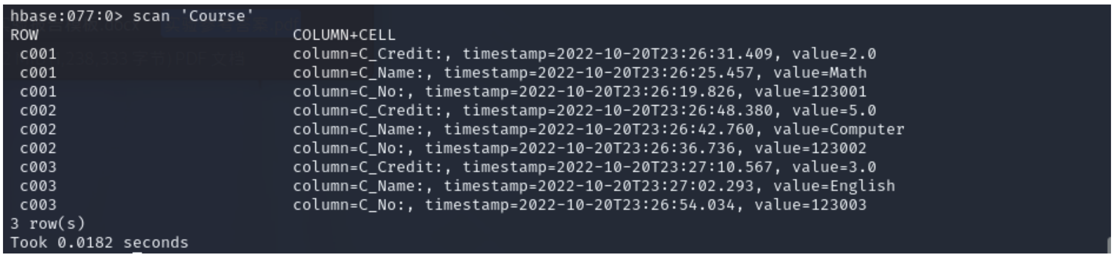


```
hbase:037:0> scan 'Student'
ROW                           COLUMN+CELL
 s001                         column=S_Age:, timestamp=2022-10-20T23:08:46.288, value=23
 s001                         column=S_Name:, timestamp=2022-10-20T23:08:22.998, value=Zhangsan
 s001                         column=S_No:, timestamp=2022-10-20T23:07:36.688, value=2015001
 s001                         column=S_Sex:, timestamp=2022-10-20T23:08:35.588, value=male
 s002                         column=S_Age:, timestamp=2022-10-20T23:09:26.465, value=22
 s002                         column=S_Name:, timestamp=2022-10-20T23:09:04.735, value=Mary
 s002                         column=S_No:, timestamp=2022-10-20T23:08:57.041, value=2015002
 s002                         column=S_Sex:, timestamp=2022-10-20T23:09:16.476, value=female
 s003                         column=S_Age:, timestamp=2022-10-20T23:10:05.963, value=24
 s003                         column=S_Name:, timestamp=2022-10-20T23:09:45.099, value=Lisi
 s003                         column=S_No:, timestamp=2022-10-20T23:09:35.642, value=2015003
 s003                         column=S_Sex:, timestamp=2022-10-20T23:09:54.388, value=male
3 row(s)
Took 0.0092 seconds
```

课程表

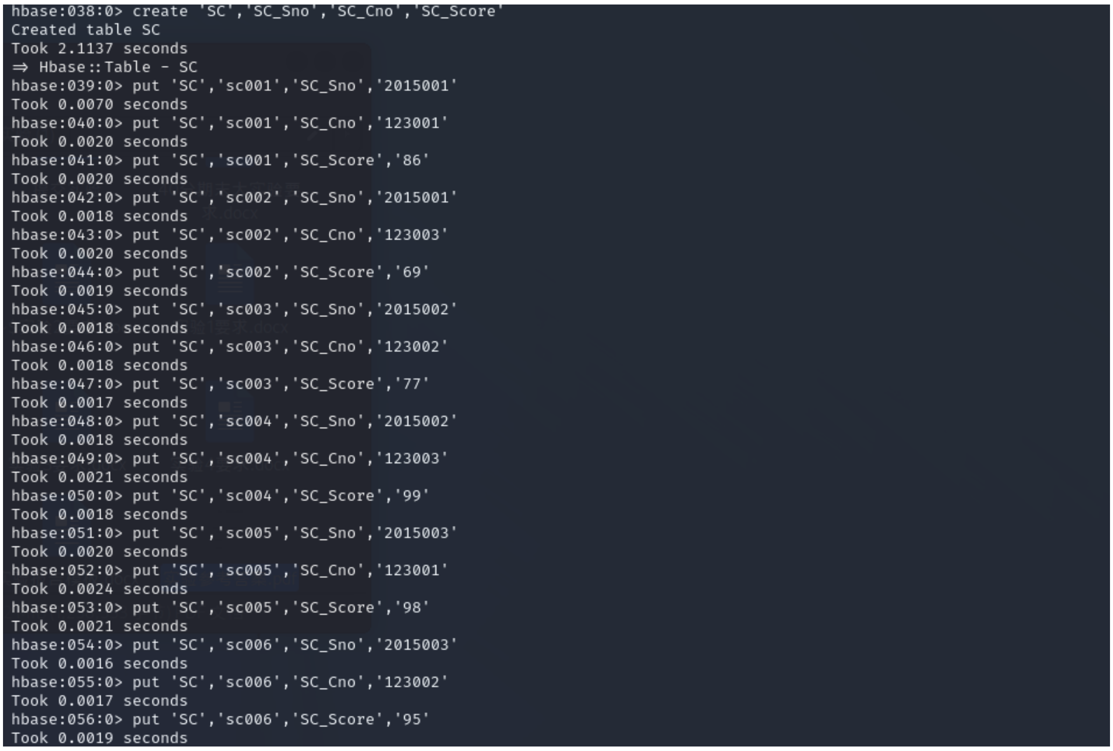


```
hbase:067:0> create 'Course','C_No','C_Name','C_Credit'
Created table Course
Took 2.1101 seconds
=> Hbase::Table - Course
hbase:068:0> put 'Course','c001','C_No','123001'
Took 0.0044 seconds
hbase:069:0> put 'Course','c001','C_Name','Math'
Took 0.0017 seconds
hbase:070:0> put 'Course','c001','C_Credit','2.0'
Took 0.0026 seconds
hbase:071:0> put 'Course','c002','C_No','123002'
Took 0.0020 seconds
hbase:072:0> put 'Course','c002','C_Name','Computer'
Took 0.0018 seconds
hbase:073:0> put 'Course','c002','C_Credit','5.0'
Took 0.0015 seconds
hbase:074:0> put 'Course','c003','C_No','123003'
Took 0.0019 seconds
hbase:075:0> put 'Course','c003','C_Name','English'
Took 0.0018 seconds
hbase:076:0> put 'Course','c003','C_Credit','3.0'
Took 0.0018 seconds
```

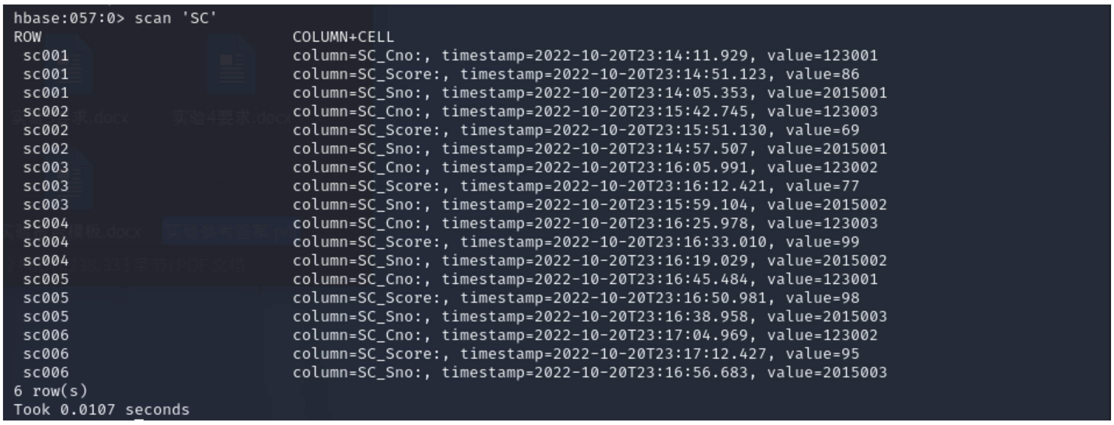


```
hbase:077:0> scan 'Course'
ROW                             COLUMN+CELL
 c001                           column=C_Credit:, timestamp=2022-10-20T23:26:31.409, value=2.0
 c001                           column=C_Name:, timestamp=2022-10-20T23:26:25.457, value=Math
 c001                           column=C_No:, timestamp=2022-10-20T23:26:19.826, value=123001
 c002                           column=C_Credit:, timestamp=2022-10-20T23:26:48.380, value=5.0
 c002                           column=C_Name:, timestamp=2022-10-20T23:26:42.760, value=Computer
 c002                           column=C_No:, timestamp=2022-10-20T23:26:36.736, value=123002
 c003                           column=C_Credit:, timestamp=2022-10-20T23:27:10.567, value=3.0
 c003                           column=C_Name:, timestamp=2022-10-20T23:27:02.293, value=English
 c003                           column=C_No:, timestamp=2022-10-20T23:26:54.034, value=123003
3 row(s)
Took 0.0182 seconds
```

选课表：



```
hbase:038:0> create 'SC','SC_Sno','SC_Cno','SC_Score'
Created table SC
Took 2.1137 seconds
=> Hbase::Table - SC
hbase:039:0> put 'SC','sc001','SC_Sno','2015001'
Took 0.0070 seconds
hbase:040:0> put 'SC','sc001','SC_Cno','123001'
Took 0.0020 seconds
hbase:041:0> put 'SC','sc001','SC_Score','86'
Took 0.0020 seconds
hbase:042:0> put 'SC','sc002','SC_Sno','2015001'
Took 0.0018 seconds
hbase:043:0> put 'SC','sc002','SC_Cno','123003'
Took 0.0020 seconds
hbase:044:0> put 'SC','sc002','SC_Score','69'
Took 0.0019 seconds
hbase:045:0> put 'SC','sc003','SC_Sno','2015002'
Took 0.0018 seconds
hbase:046:0> put 'SC','sc003','SC_Cno','123002'
Took 0.0018 seconds
hbase:047:0> put 'SC','sc003','SC_Score','77'
Took 0.0017 seconds
hbase:048:0> put 'SC','sc004','SC_Sno','2015002'
Took 0.0018 seconds
hbase:049:0> put 'SC','sc004','SC_Cno','123003'
Took 0.0021 seconds
hbase:050:0> put 'SC','sc004','SC_Score','99'
Took 0.0018 seconds
hbase:051:0> put 'SC','sc005','SC_Sno','2015003'
Took 0.0020 seconds
hbase:052:0> put 'SC','sc005','SC_Cno','123001'
Took 0.0024 seconds
hbase:053:0> put 'SC','sc005','SC_Score','98'
Took 0.0021 seconds
hbase:054:0> put 'SC','sc006','SC_Sno','2015003'
Took 0.0016 seconds
hbase:055:0> put 'SC','sc006','SC_Cno','123002'
Took 0.0017 seconds
hbase:056:0> put 'SC','sc006','SC_Score','95'
Took 0.0019 seconds
```



```
hbase:057:0> scan 'SC'
ROW                             COLUMN+CELL
 sc001                          column=SC_Cno:, timestamp=2022-10-20T23:14:11.929, value=123001
 sc001                          column=SC_Score:, timestamp=2022-10-20T23:14:51.123, value=86
 sc001                          column=SC_Sno:, timestamp=2022-10-20T23:14:05.353, value=2015001
 sc002                          column=SC_Cno:, timestamp=2022-10-20T23:15:42.745, value=123003
 sc002                          column=SC_Score:, timestamp=2022-10-20T23:15:51.130, value=69
 sc002                          column=SC_Sno:, timestamp=2022-10-20T23:14:57.507, value=2015001
 sc003                          column=SC_Cno:, timestamp=2022-10-20T23:16:05.991, value=123002
 sc003                          column=SC_Score:, timestamp=2022-10-20T23:16:12.421, value=77
 sc003                          column=SC_Sno:, timestamp=2022-10-20T23:15:59.104, value=2015002
 sc004                          column=SC_Cno:, timestamp=2022-10-20T23:16:25.978, value=123003
 sc004                          column=SC_Score:, timestamp=2022-10-20T23:16:33.010, value=99
 sc004                          column=SC_Sno:, timestamp=2022-10-20T23:16:19.029, value=2015002
 sc005                          column=SC_Cno:, timestamp=2022-10-20T23:16:45.484, value=123001
 sc005                          column=SC_Score:, timestamp=2022-10-20T23:16:50.981, value=98
 sc005                          column=SC_Sno:, timestamp=2022-10-20T23:16:38.958, value=2015003
 sc006                          column=SC_Cno:, timestamp=2022-10-20T23:17:04.969, value=123002
 sc006                          column=SC_Score:, timestamp=2022-10-20T23:17:12.427, value=95
 sc006                          column=SC_Sno:, timestamp=2022-10-20T23:16:56.683, value=2015003
6 row(s)
Took 0.0107 seconds
```

2. 请编程实现以下功能：

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

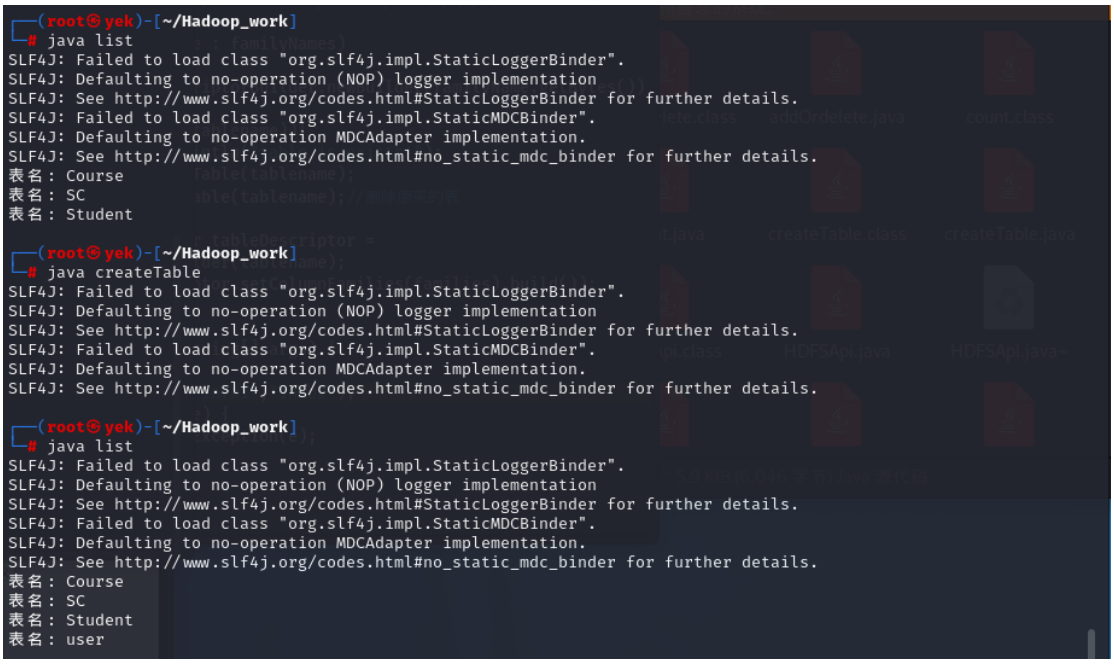
创建表，参数 tableName 为表的名称，字符串数组 fields 为存储记录各个字段名称的数组。要求当 HBase 已经存在名为 tableName 的表的时候，先删除原有的表，然后再创建新的表。

实验 Java 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 java.io.IOException;
  import java.util.List;
  import java.util.ArrayList;

  public class createTable {
      public static Configuration configuration;
      public static Connection connection;
      public static Admin admin;
      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(null != connection){
                  connection.close();
              }
          }catch (IOException e){
              e.printStackTrace();
          }
      }
        public static void clearRows(String tableName,String...
familyNames)throws IOException{
          init();
          TableName tablename = TableName.valueOf(tableName);
            // 创建集合用于存放ColumnFamilyDescriptor 对象
          List<ColumnFamilyDescriptor> families = new ArrayList<>();
          // 将每个familyName 对应的ColumnFamilyDescriptor 对象添加到
families 集合中保存
          for (String familyName : familyNames)
              families.add(ColumnFamilyDescriptorBuilder.newBuilder(fami
lyName.getBytes()).build());
          if(admin.tableExists(tablename)){
          System.out.println("table is exists!");
          admin.disableTable(tablename);
          admin.deleteTable(tablename);//删除原来的表
      }
          TableDescriptorBuilder tableDescriptor =
TableDescriptorBuilder.newBuilder(tablename);
admin.createTable(tableDescriptor.setColumnFamilies(families).build());

          close();
      }
      public static void main(String[] args) {
          try {
              clearRows("user","info1","info2");
          } catch (IOException e) {
              throw new RuntimeException(e);
          }
      }
  }
```

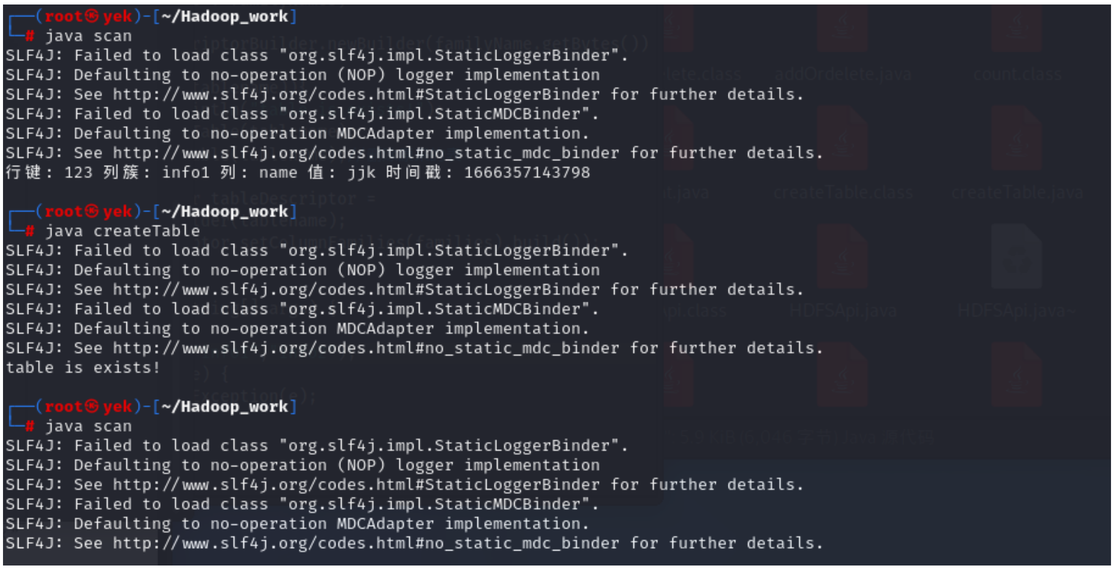
实验结果：

1. 表不存在时创建表



```
┌──(root㉿yek)-[~/Hadoop_work]
└─# java list
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
表名： Course
表名： SC
表名： Student

┌──(root㉿yek)-[~/Hadoop_work]
└─# java createTable
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.

┌──(root㉿yek)-[~/Hadoop_work]
└─# java list
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
表名： Course
表名： SC
表名： Student
表名： user
```

2. 表存在时删除原表，创建新表



```
┌──(root㉿yek)-[~/Hadoop_work]
└─# java scan
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
行键：123 列簇：info1 列：name 值：jjk 时间戳：1666357143798

┌──(root㉿yek)-[~/Hadoop_work]
└─# java createTable
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
table is exists!

┌──(root㉿yek)-[~/Hadoop_work]
└─# java scan
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
```

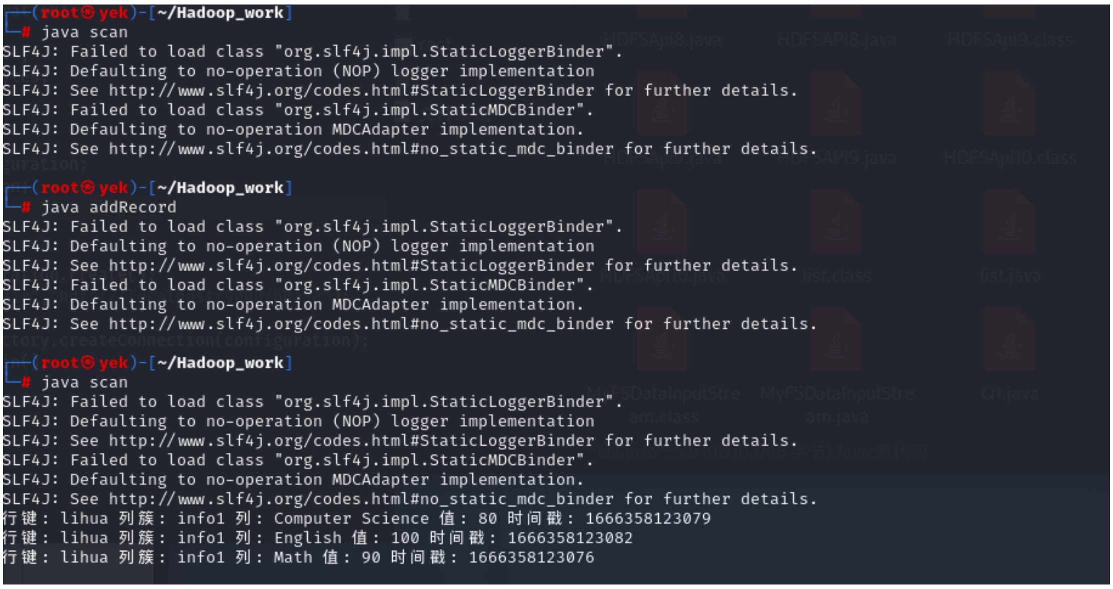
（2）addRecord(String tableName, String row, String[] fields, String[] values)

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。

实验 Java 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 java.io.IOException;

public class addRecord {
    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;
    //建立连接
    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(":");
              put.addColumn(cols[0].getBytes(), cols[1].getBytes(),
values[i].getBytes());
              table.put(put);
          }
          table.close();
          close();
      }
      public static void main(String[] args) {
          try {
              String[] columns = {"info1:Math", "info1:Computer Science",
"info1:English"};
              String[] values = {"90","80","100"};
              addRecord("user","lihua",columns,values);
          } catch (IOException e) {
              throw new RuntimeException(e);
          }
      }
  }
```

实验结果：



```
┌──(root㉿yek)-[~/Hadoop_work]
└─# java scan
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.

┌──(root㉿yek)-[~/Hadoop_work]
└─# java addRecord
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.

┌──(root㉿yek)-[~/Hadoop_work]
└─# java scan
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
行键: lihua 列簇: info1 列: Computer Science 值: 80 时间戳: 1666358123079
行键: lihua 列簇: info1 列: English 值: 100 时间戳: 1666358123082
行键: lihua 列簇: info1 列: Math 值: 90 时间戳: 1666358123076
```

（3）scanColumn(String tableName, String column)

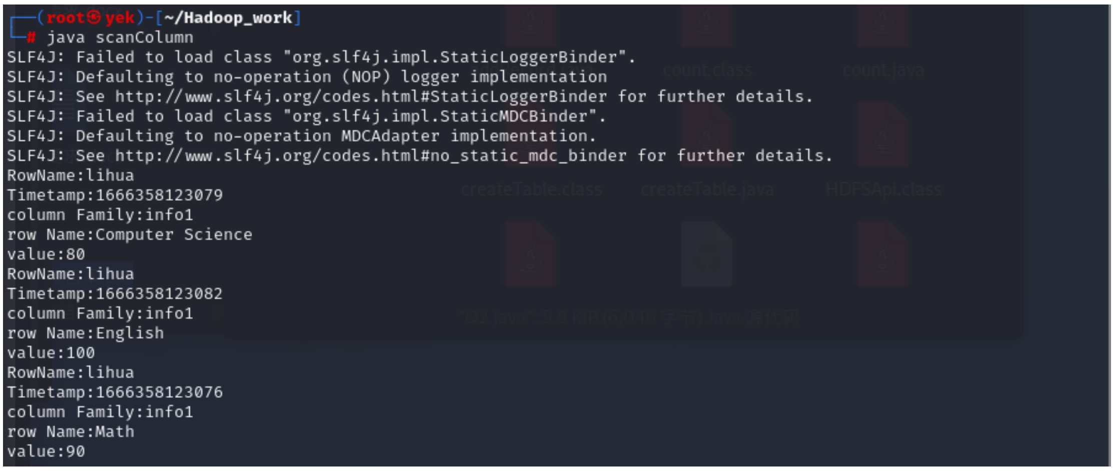
浏览表 tableName 某一列的数据，如果某一行记录中该列数据不存在，则返回 null。要求当参数 column 为某一列族名称时，如果底下有若干个列限定符，则要列出每个列限定符代表的列的数据；当参数 column 为某一列具体名称（例如“Score:Math”）时，只需要列出该列的数据。

实验 Java 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 java.io.IOException;

public class scanColumn {
    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;
    //建立连接
    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 scanColumn(String tableName,String
column)throws  IOException{
          init();
          Table table =
connection.getTable(TableName.valueOf(tableName));
          Scan scan = new Scan();
          scan.addFamily(Bytes.toBytes(column));
          ResultScanner scanner = table.getScanner(scan);
          for (Result result = scanner.next(); result != null; result =
scanner.next()){
              showCell(result);
          }
          table.close();
          close();
      }
      //格式化输出
      public static void showCell(Result result){
          Cell[] cells = result.rawCells();
          for(Cell cell:cells){
              System.out.println("RowName:"+new
String(Bytes.toString(cell.getRowArray(),cell.getRowOffset(),
cell.getRowLength()))+" ");
```

```
              System.out.println("Timetamp:"+cell.getTimestamp()+" ");
              System.out.println("column Family:"+new
String(Bytes.toString(cell.getFamilyArray(),cell.getFamilyOffset(),
cell.getFamilyLength()))+" ");
              System.out.println("row Name:"+new
String(Bytes.toString(cell.getQualifierArray(),cell.getQualifierOffset(
), cell.getQualifierLength()))+" ");
              System.out.println("value:"+new
String(Bytes.toString(cell.getValueArray(),cell.getValueOffset(),
cell.getValueLength()))+" ");
          }
      }
      public static void main(String[] args) {
          try {
            scanColumn("user","info1");
          } catch (IOException e) {
              throw new RuntimeException(e);
          }
      }
  }
```

实验结果：



```
┌──(root㉿yek)-[~/Hadoop_work]
└─# java scanColumn
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
RowName:lihua
Timetamp:1666358123079
column Family:info1
row Name:Computer Science
value:80
RowName:lihua
Timetamp:1666358123082
column Family:info1
row Name:English
value:100
RowName:lihua
Timetamp:1666358123076
column Family:info1
row Name:Math
value:90
```

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

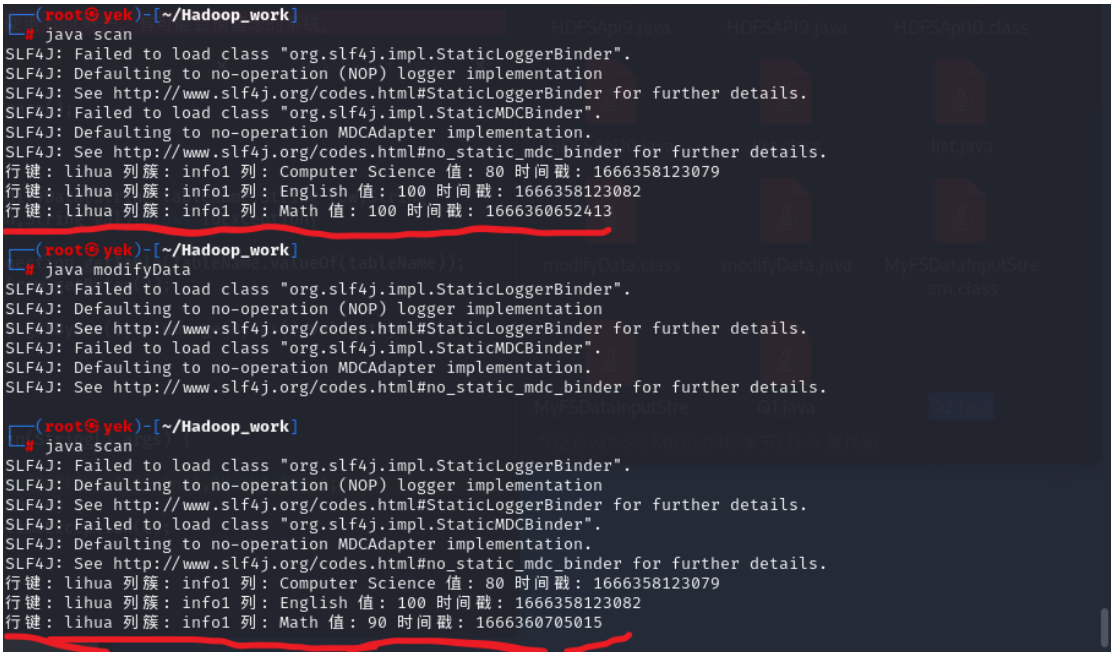
修改表 tableName，行 row（可以用学生姓名 S_Name 表示），列 column 指定的单元格的数据。

实验 Java 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 java.io.IOException;

  public class modifyData {
      public static Configuration configuration;
      public static Connection connection;
      public static Admin admin;
      //建立连接
      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 modifyData(String tableName,String row,String
columnfamily,String column,String val)throws IOException{
          init();
          Table table =
connection.getTable(TableName.valueOf(tableName));
          Put put = new Put(row.getBytes());
```

```
            put.addColumn(columnfamily.getBytes(),column.getBytes(),val.ge
tBytes());
            table.put(put);
            table.close();
            close();
        }
        public static void main(String[] args) {
            try {
                modifyData("user","lihua","info1","Math","90");
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
        }
    }
```

实验结果：



```
┌──(root㉿yek)-[~/Hadoop_work]
└─# java scan
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
行键: lihua 列簇: info1 列: Computer Science 值: 80 时间戳: 1666358123079
行键: lihua 列簇: info1 列: English 值: 100 时间戳: 1666358123082
行键: lihua 列簇: info1 列: Math 值: 100 时间戳: 1666360652413

┌──(root㉿yek)-[~/Hadoop_work]
└─# java modifyData
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.

┌──(root㉿yek)-[~/Hadoop_work]
└─# java scan
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
行键: lihua 列簇: info1 列: Computer Science 值: 80 时间戳: 1666358123079
行键: lihua 列簇: info1 列: English 值: 100 时间戳: 1666358123082
行键: lihua 列簇: info1 列: Math 值: 90 时间戳: 1666360705015
```

（5）deleteRow(String tableName, String row)

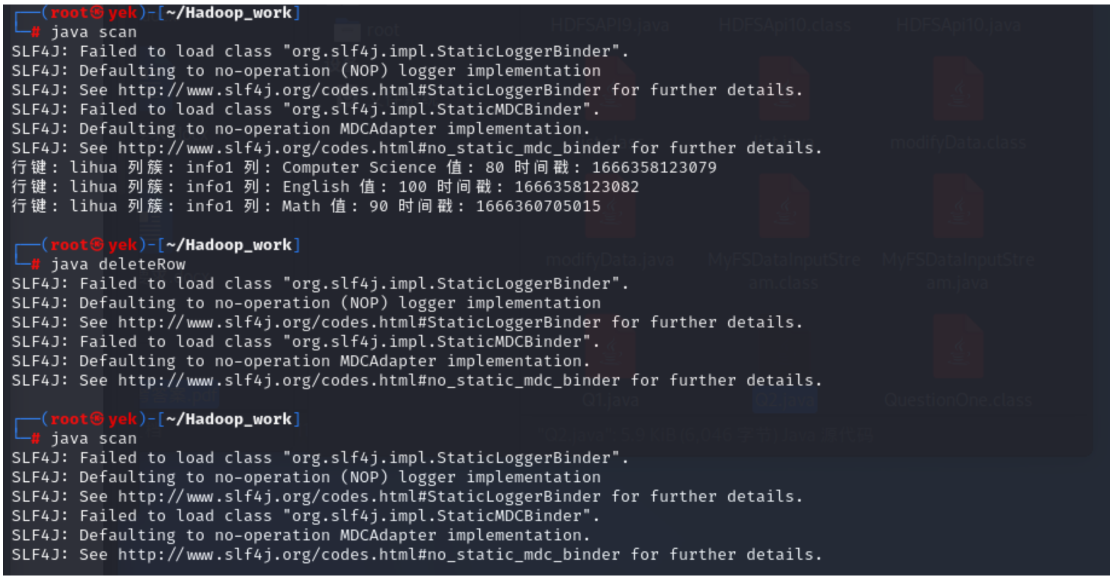
删除表 tableName 中 row 指定的行的记录。

实验 Java 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 java.io.IOException;

public class deleteRow {
    public static Configuration configuration;
```

```java
    public static Connection connection;
    public static Admin admin;
    //建立连接
    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 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();
    }
    public static void main(String[] args) {
        try {
            deleteRow("user","lihua");
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
        }
}
```

实验结果：



```
┌──(root㉿yek)-[~/Hadoop_work]
└─# java scan
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
行键: lihua 列簇: info1 列: Computer Science 值: 80 时间戳: 1666358123079
行键: lihua 列簇: info1 列: English 值: 100 时间戳: 1666358123082
行键: lihua 列簇: info1 列: Math 值: 90 时间戳: 1666360705015

┌──(root㉿yek)-[~/Hadoop_work]
└─# java deleteRow
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.

┌──(root㉿yek)-[~/Hadoop_work]
└─# java scan
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
SLF4J: Defaulting to no-operation MDCAdapter implementation.
SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
```

3 实验总结

练习使用利用 hbase shell 脚本编写 hbase 的常用命令，加深了我对 hbase 的理解，以及对 hbase 操作的熟练度，能更快的使用这些命令。通过 hbase Api 进行编程操作，加深 了我对 hbase 的理解