

(1) 列出 HBase 所有的表的相关信息，例如表名

```
hbase(main):001:0> list
```

(2) 在终端打印出指定的表的所有记录数据

查看记录数据:

```
hbase(main):001:0> scan '表名'
```

查看表的信息:

```
hbase(main):001:0> describe '表名'
```

(3) 向已经创建好的表添加和删除指定的列族或列

添加列族或列:

```
hbase(main):001:0> alter '表名','NAME'=>'列名'
```

删除列族或列:

```
hbase(main):001:0> alter '表名','NAME'=>'列名',METHOD=>'delete'
```

(4) 清空指定的表的所有记录数据

```
hbase(main):001:0> drop '表名'
```

(5) 统计表的行数

```
hbase(main):001:0> count '表名'
```

2. 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:

学号	姓名	性别	年龄
2015001	Zhangsan	male	23
2015002	Mary	female	22
2015003	Lisi	male	24

课程表:

课程号	课程名	学分
123001	Math	2.0
123002	Computer Science	5.0
123003	English	3.0

选课表：

学号	课程号	成绩
2015001	123001	86
2015001	123003	69
2015002	123002	77
2015002	123003	99
2015003	123001	98
2015003	123002	95

创建三个表格

```
hbase(main):001:0> create 'Student','S_No','S_Name','S_Sex','S_Age'
Created table Student
Took 3.5968 seconds
=> Hbase::Table - Student
hbase(main):002:0> create 'Course','C_No','C_Name','C_Credit'
Created table Course
Took 2.3274 seconds
=> Hbase::Table - Course
hbase(main):003:0> create 'SC','SC_Sno','SC_Cno','SC_Score'
Created table SC
Took 2.3036 seconds
=> Hbase::Table - SC
hbase(main):004:0> list
TABLE
Course
SC
Student
3 row(s)
Took 0.1286 seconds
=> ["Course", "SC", "Student"]
```

‘Student’ 表中添加数据

```
hbase(main):005:0> put 'Student','1','S_No','2015001'
Took 0.4452 seconds
hbase(main):006:0> put 'Student','1','S_Name','Zhangsan'
Took 0.0174 seconds
hbase(main):007:0> put 'Student','1','S_Sex','male'
Took 0.0218 seconds
hbase(main):008:0> put 'Student','1','S_Age','23'
Took 0.0097 seconds
hbase(main):009:0> put 'Student','2','S_No','2015002'
Took 0.0254 seconds
hbase(main):010:0> put 'Student','1','S_Name','Mary'
Took 0.0138 seconds
hbase(main):011:0> put 'Student','1','S_Name','Zhangsan'
Took 0.0086 seconds
hbase(main):012:0> put 'Student','2','S_Name','Mary'
Took 0.0113 seconds
hbase(main):013:0> put 'Student','2','S_Sex','female'
Took 0.0069 seconds
hbase(main):014:0> put 'Student','2','S_Age','22'
Took 0.0325 seconds
hbase(main):015:0> put 'Student','3','S_No','2015003'
Took 0.0081 seconds
hbase(main):016:0> put 'Student','3','S_Name','Lisi'
Took 0.0226 seconds
hbase(main):017:0> put 'Student','3','S_Sex','male'
Took 0.0176 seconds
hbase(main):018:0> put 'Student','3','S_Age','24'
Took 0.0116 seconds
hbase(main):019:0> scan 'Student'
ROW COLUMN+CELL
1 column=S_Age:, timestamp=1589036922246, value=23
1 column=S_Name:, timestamp=1589037013215, value=Zhangsan
1 column=S_No:, timestamp=1589036873321, value=2015001
1 column=S_Sex:, timestamp=1589036911682, value=male
2 column=S_Age:, timestamp=1589037052378, value=22
2 column=S_Name:, timestamp=1589037023148, value=Mary
2 column=S_No:, timestamp=1589036971568, value=2015002
2 column=S_Sex:, timestamp=1589037038278, value=female
3 column=S_Age:, timestamp=1589037096256, value=24
3 column=S_Name:, timestamp=1589037078823, value=Lisi
3 column=S_No:, timestamp=1589037067953, value=2015003
3 column=S_Sex:, timestamp=1589037087126, value=male
3 row(s)
```

‘Course’ 表中添加数据

```
hbase(main):020:0> put 'Course','1','C_No','123001'
Took 0.0450 seconds
hbase(main):021:0> put 'Course','1','C_Name','Math'
Took 0.0182 seconds
hbase(main):022:0> put 'Course','1','C_Credit','2.0'
Took 0.0115 seconds
hbase(main):023:0> put 'Course','2','C_No','123002'
Took 0.0093 seconds
hbase(main):024:0> put 'Course','2','C_Name','Computer Science'
Took 0.0093 seconds
hbase(main):025:0> put 'Course','2','C_Credit','5.0'
Took 0.0226 seconds
hbase(main):026:0> put 'Course','3','C_No','123003'
Took 0.0124 seconds
hbase(main):027:0> put 'Course','3','C_Name','English'
Took 0.0110 seconds
hbase(main):028:0> put 'Course','3','C_Credit','3.0'
Took 0.0080 seconds
hbase(main):029:0> scan 'Course'
ROW                                COLUMN+CELL
1                                  column=C_Credit:, timestamp=1589038396372, value=2.0
1                                  column=C_Name:, timestamp=1589038376174, value=Math
1                                  column=C_No:, timestamp=1589038356825, value=123001
2                                  column=C_Credit:, timestamp=1589038445580, value=5.0
2                                  column=C_Name:, timestamp=1589038432094, value=Computer Science
2                                  column=C_No:, timestamp=1589038408390, value=123002
3                                  column=C_Credit:, timestamp=1589038498579, value=3.0
3                                  column=C_Name:, timestamp=1589038476012, value=English
3                                  column=C_No:, timestamp=1589038456482, value=123003
3 row(s)
```

‘SC’ 表中添加数据

```
hbase(main):034:0> put 'SC','2','SC_Cno','123003'
Took 0.0079 seconds
hbase(main):035:0> put 'SC','2','SC_Score','69'
Took 0.0069 seconds
hbase(main):036:0> put 'SC','3','SC_Sno','2015002'
Took 0.0082 seconds
hbase(main):037:0> put 'SC','3','SC_Cno','123002'
Took 0.0098 seconds
hbase(main):038:0> put 'SC','3','SC_Score','77'
Took 0.0084 seconds
hbase(main):039:0> put 'SC','4','SC_Sno','2015002'
Took 0.0114 seconds
hbase(main):040:0> put 'SC','4','SC_Cno','123003'
Took 0.0078 seconds
hbase(main):041:0> put 'SC','4','SC_Score','99'
Took 0.0165 seconds
hbase(main):042:0> put 'SC','5','SC_Sno','2015003'
Took 0.0090 seconds
hbase(main):043:0> put 'SC','5','SC_Cno','123001'
Took 0.0082 seconds
hbase(main):044:0> put 'SC','5','SC_Score','98'
Took 0.0060 seconds
hbase(main):045:0> put 'SC','6','SC_Sno','2015003'
Took 0.0078 seconds
hbase(main):046:0> put 'SC','6','SC_Cno','123002'
Took 0.0070 seconds
hbase(main):047:0> put 'SC','6','SC_Score','95'
Took 0.0093 seconds
hbase(main):048:0> scan 'SC'
ROW                                COLUMN+CELL
1                                  column=SC_Cno:, timestamp=1589038718730, value=123001
1                                  column=SC_Score:, timestamp=1589038744924, value=86
1                                  column=SC_Sno:, timestamp=1589038696553, value=2015001
2                                  column=SC_Cno:, timestamp=1589038775504, value=123003
2                                  column=SC_Score:, timestamp=1589038788841, value=69
2                                  column=SC_Sno:, timestamp=1589038764997, value=2015001
3                                  column=SC_Cno:, timestamp=1589038840982, value=123002
3                                  column=SC_Score:, timestamp=1589038851598, value=77
3                                  column=SC_Sno:, timestamp=1589038808686, value=2015002
4                                  column=SC_Cno:, timestamp=1589038879786, value=123003
4                                  column=SC_Score:, timestamp=1589038889079, value=99
4                                  column=SC_Sno:, timestamp=1589038863007, value=2015002
5                                  column=SC_Cno:, timestamp=1589038914911, value=123001
5                                  column=SC_Score:, timestamp=1589038924007, value=98
5                                  column=SC_Sno:, timestamp=1589038903552, value=2015003
6                                  column=SC_Cno:, timestamp=1589038945785, value=123002
6                                  column=SC_Score:, timestamp=1589038954728, value=95
6                                  column=SC_Sno:, timestamp=1589038935585, value=2015003
6 row(s)
```

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

创建表，参数 tableName 为表的名称，字符串数组 fields 为存储记录各个域名称的数组。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。

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.TableName;
import org.apache.hadoop.hbase.client.ColumnFamilyDescriptor;
import org.apache.hadoop.hbase.client.ColumnFamilyDescriptorBuilder;
import org.apache.hadoop.hbase.client.Connection;
import org.apache.hadoop.hbase.client.Admin;
import org.apache.hadoop.hbase.client.ConnectionFactory;
import org.apache.hadoop.hbase.client.TableDescriptorBuilder;
```

```

import org.apache.hadoop.hbase.util.Bytes;
import java.io.IOException;

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

    public static void init(){//建立连接
        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(){//关闭连接
        try{
            if(admin != null){
                admin.close();
            }
            if(connection != null){
                connection.close();
            }
        }catch(IOException e){
            e.printStackTrace();
        }
    }
    public static void createTable(String tableName,String[] fields) throws
IOException{
        init();
        TableName tablename = TableName.valueOf(tableName);//定义表名
        if(admin.tableExists(tablename)){
            System.out.println("table is exists!");
            admin.disableTable(tablename);
            admin.deleteTable(tablename);
        }
        TableDescriptorBuilder tableDescriptor =
TableDescriptorBuilder.newBuilder(tablename);
        for(int i=0;i<fields.length;i++){
            ColumnFamilyDescriptor family =
ColumnFamilyDescriptorBuilder.newBuilder(Bytes.toBytes(fields[i])).build();
            tableDescriptor.setColumnFamily(family);
        }
        admin.createTable(tableDescriptor.build());
        close();
    }
}

```

```

public static void main(String[] args){
    String[] fields = {"id","score"};
    try{
        createTable("test",fields);
    }catch(IOException e){
        e.printStackTrace();
    }
}
}

```

(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。

当向表添加数据时，我们需要一个Put对象，在Put对象之前我们需要获取Table对象，这样才能对指定的表进行操作

```

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.TableName;
import org.apache.hadoop.hbase.client.Admin;
import org.apache.hadoop.hbase.client.Connection;
import org.apache.hadoop.hbase.client.ConnectionFactory;
import org.apache.hadoop.hbase.client.Put;
import org.apache.hadoop.hbase.client.Table;

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

    public static void init(){//建立连接
        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(){//关闭连接
    try{
        if(admin != null){
            admin.close();
        }
        if(connection != null){
            connection.close();
        }
    }catch(IOException e){
        e.printStackTrace();
    }
}

public static void addRecord(String tableName,String row,String[] fields,String[]
values) throws IOException{
    init();//连接Hbase
    Table table = connection.getTable(TableName.valueOf(tableName));//表连接
    Put put = new Put(row.getBytes());//创建put对象
    for(int i=0;i<fields.length;i++){
        String[] cols = fields[i].split(":");
        if(cols.length == 1){
            put.addColumn(fields[i].getBytes(),"".getBytes(),values[i].getBytes());
        }
        else{
            put.addColumn(cols[0].getBytes(),cols[1].getBytes(),values[i].getBytes());
        }
        table.put(put);//向表中添加数据
    }
    close();//关闭连接
}

public static void main(String[] args){
    String[] fields = {"Score:Math","Score:Computer Science","Score:English"};
    String[] values = {"90","90","90"};
    try{
        addRecord("grade","S_Name",fields,values);
    }catch(IOException e){
        e.printStackTrace();
    }
}
}

```

(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。

```

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;

```



```

import org.apache.hadoop.hbase.Cell;
import org.apache.hadoop.hbase.CellUtil;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.TableName;
import org.apache.hadoop.hbase.client.Admin;
import org.apache.hadoop.hbase.client.Connection;
import org.apache.hadoop.hbase.client.ConnectionFactory;
import org.apache.hadoop.hbase.client.Get;
import org.apache.hadoop.hbase.client.Result;
import org.apache.hadoop.hbase.client.ResultScanner;
import org.apache.hadoop.hbase.client.Scan;
import org.apache.hadoop.hbase.client.Table;
import org.apache.hadoop.hbase.util.Bytes;

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

    public static void init(){//建立连接
        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(){//关闭连接
        try{
            if(admin != null){
                admin.close();
            }
            if(connection != null){
                connection.close();
            }
        }catch(IOException e){
            e.printStackTrace();
        }
    }

    public static void showResult(Result result){
        Cell[] cells = result.rawCells();
        for(int i=0;i<cells.length;i++){
            System.out.println("RowName:"+new String(CellUtil.cloneRow(cells[i])));//打印行
            键
            System.out.println("ColumnName:"+new
String(CellUtil.cloneQualifier(cells[i])));//打印列名

```

```

        System.out.println("Value:"+new String(CellUtil.cloneValue(cells[i]))); //打印值
        System.out.println("Column Family:"+new
String(CellUtil.cloneFamily(cells[i]))); //打印列簇
        System.out.println();
    }
}
public static void scanColumn(String tableName,String column){
    init();
    try {
        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()){
            showResult(result);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    finally{
        close();
    }
}
public static void main(String[] args){
    scanColumn("Student","S_Age");
}
}

```

(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。

```

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.TableName;
import org.apache.hadoop.hbase.client.Admin;
import org.apache.hadoop.hbase.client.Connection;
import org.apache.hadoop.hbase.client.ConnectionFactory;
import org.apache.hadoop.hbase.client.Put;
import org.apache.hadoop.hbase.client.Table;

public class modifyData {
    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;
    public static void init(){ //建立连接
        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(){//关闭连接
try{
    if(admin != null){
        admin.close();
    }
    if(connection != null){
        connection.close();
    }
}catch(IOException e){
    e.printStackTrace();
}
}
public static void modifyData(String tableName,String row,String column,String
value) throws IOException{
    init();
    Table table = connection.getTable(TableName.valueOf(tableName));
    Put put = new Put(row.getBytes());
    String[] cols = column.split(":");
    if(cols.length == 1){
        put.addColumn(column.getBytes(),"".getBytes(), value.getBytes());
    }
    else{
        put.addColumn(cols[0].getBytes(), cols[1].getBytes(), value.getBytes());
    }
    table.put(put);
    close();
}
public static void main(String[] args){
    try{
        modifyData("Student","1","S_Name","Tom");
    }
    catch(Exception e){
        e.printStackTrace();
    }
}
}

```

(5) deleteRow(String tableName, String row)

删除表 tableName 中 row 指定的行的记录。

```

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.TableName;
import org.apache.hadoop.hbase.client.Admin;
import org.apache.hadoop.hbase.client.Connection;
import org.apache.hadoop.hbase.client.ConnectionFactory;
import org.apache.hadoop.hbase.client.Delete;
import org.apache.hadoop.hbase.client.Table;

public class deleteRow {
    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;
    public static void init(){//建立连接
        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(){//关闭连接
        try{
            if(admin != null){
                admin.close();
            }
            if(connection != null){
                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);
        close();
    }
    public static void main(String[] args){
        try{
            deleteRow("Student","3");
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}

```

}

}

}