实验 3 熟悉常用的 HBase 操作

19084129-李奕澄

实验目的

1) 理解 HBase 在 Hadoop 体系结构种的角色
2) 熟练使用 HBase 操作常用的 Shell 命令
3) 熟悉 HBase 操作常用的 JavaAPI

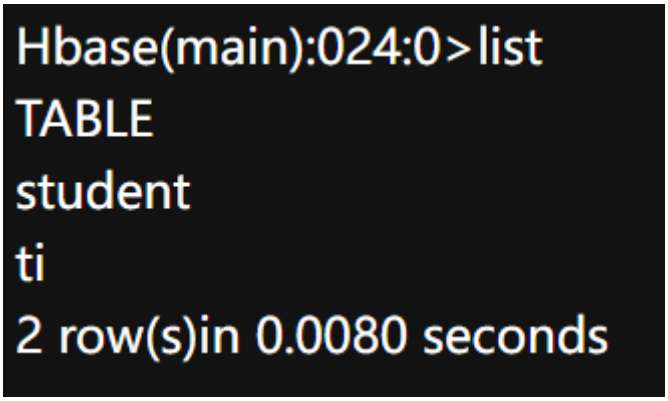
实验环境

Ubuntu 18.04
Hadoop 3.1.3

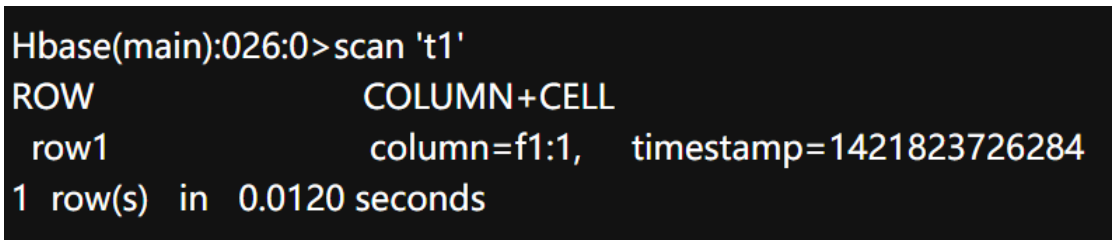
实验内容

1. 编程实现以下指定功能 H 并用 hadoop 提供的 HBase Shell 命令完成下列任务：

（1） 列出 HBase 所有的表的相关信息，例如表名；



（2） 在终端打印出指定的表的所有记录数据；



（3） 向已经创建好的表添加和删除指定的列族或列；

```
Hbase(main):027:0>alter  't1' , 'NAME'=>'f4'
Updating all regions with the new schema...
0/1 regions updated.
1/1 regions updated.
Done.
0  row(s) in  2.2170  seconds
```

```
Hbase(main):028:0>alter  't1' ,NAME=>'f4',METHOD=>'delete'
Updating all regions with the new schema...
0/1 regions updated.
1/1 regions updated.
Done.
0  row(s)  in  2.6540  seconds
```

（4） 清空指定的表的所有记录数据;

```
Hbase(main):029:0> truncate  't1'
Truncating  't1'  table  (it may take a while):
  -  Disabling table...
  -  Truncating table...
0  row(s)  in  1.7360  seconds
```

（5） 统计表的行数。

```
Hbase(main):032:0> count  't1'
0 row(s)in 0.0230 seconds


=>0
```

编程实现:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.HColumnDescriptor;
import org.apache.hadoop.hbase.HTableDescriptor;
```

```
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 java.io.IOException;

public class HBaseOperate {
      public static Configuration configuration;
      public static Connection connection;
      public static Admin admin;
      public static long ts;
public static void getData(String tableName)throws IOException{
         init();
         Table table connection.getTable(TableName.valueOf(tableName));
         Scanscan new Scan();
         ResultScanner scanner=table.getScanner(scan:/获取行的遍历器
         for(Result result:scanner)
            printRecoder(result);
      }
            close();
public static void printRecoder(Result result)throws IOException{
         for(Cell cell:result.rawCells()){
         System.out.print(行
   健"+newString(Bytes.toString(cell.getRowArray(),cell.getRowOffset(),cell.getRowLength())));
         System.out.print("列
         簇"+newString(
         Bytes.toString(cell.getFamilyArray(),cell.getFamilyOffset(),cell.getFamilyLength())));
         System.out.print("
         列"+new
         String(Bytes.toString(cell.getQualifierArray(),cell.getQualifierOffset(),cell.getQualifierLe
         ngth())));
         System.out.print("值"+new
         String(Bytes.toString(cell.getValueArray(),cell.getValueOffset(),cell.getValueLength())))
         System.out.printin("时间戳：  "+cell.getTimestamp(0:
public static void insterRow(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 deleRow(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 clearRows(String tableName)throws IOException{
      init();
      TableName tablename TableName.valueOf(tableName);
      admin.disableTable(tablename);
      admin.deleteTable(tablename);
      TableDescriptorBuilder
      tableDescriptor
      TableDescriptorBuilder.newBuilder(tablename);
      admin.createTable(tableDescriptor.build());
      close();
}
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();resultl=null;result=scanner.next()){
         num++;
         System.out.printin("行数："+num:
         scanner.close();
         close();
}
```

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（Student）表

S_No	S_Name	S_Sex	S_Age
2015001	Zhangsan	male	23
2015002	Mary	female	22
2015003	Lisi	male	24

```
Hbase(main):030:0>  create 'Student','S No','S Name','S Sex','S Age'
Hbase(main):031:0>  put 'Student','s001','S No','2015001'
Hbase(main):032:0>  put 'Student,'s001','S Name,'Zhangsan'
Hbase(main):033:0>  put 'Student','s001','S Sex','male
Hbase(main):034:0>  put 'Student,'s001,'S Age,23'
Hbase(main):035:0>  put 'Student','s002','S No',2015002
Hbase(main):036:0>  put 'Student','s002,'S Name,'Mary
Hbase(main):037:0>  put 'Student,'s002,'S Sex','female'
Hbase(main):038:0>  put 'Student','s002,'S Age,22'
Hbase(main):039:0>  put 'Student','s003','S No','2015003
Hbase(main):040:0>  put 'Student,'s003,'S Name',Lisi
Hbase(main):041:0>  put 'Student','s003,'S Sex','male
Hbase(main):042:0>  put 'Student,'s003','S Age,24'
```

课程（Course）表

C_No	C_Name	C_Credit
123001	Math	2.0
123002	Computer Science	5.0
123003	English	3.0

```
Hbase(main):056:0>  create 'Course','C No,'C Name','C Credit'
Hbase(main):057:0>  put 'Course','c001,'C No,'123001'
Hbase(main):058:0>  put 'Course','c001','C Name,'Math'
Hbase(main):059:0>  put 'Course,'c001'C Credit','2.0'
Hbase(main):060:0>  put 'Course','c002,'C No,'123002'
Hbase(main):061:0>  put 'Course,'c002,'C_Name,'Computer'
Hbase(main):062:0>  put 'Course','c002,'C Credit,'5.0'
Hbase(main):063:0>  put 'Course','c003','C No,'123003'
Hbase(main):064:0>  put 'Course,'c003','C_Name,'English
Hbase(main):065:0>  put 'Course,'c003','C Credit.  '3.0'
```

选课（SC）

SC_Sno	SC_Cno	SC_Score
2015001	123001	86
2015001	123003	69
2015002	123002	77
2015002	123003	99
2015003	123001	89
2015003	123002	95

```
Hbase(main):066:0>  create 'SC,'SC Sno','SC Cno','SC Score'
Hbase(main):067:0>  put'SC,'sc001,'SC Sno',2015001'
Hbase(main):068:0>  put 'SC,'sc001,'SC Cno,123001'
Hbase(main):069:0>  put 'SC,'sc001','SC Score',86'
Hbase(main):070:0>  put 'SC,'sc002,'SC Sno','2015001
Hbase(main):071:0>  put 'SC,'sc002,'SC Cno','123003'
Hbase(main):072:0>  put 'SC,'sc002,'SC Score','69'
Hbase(main):073:0>  put 'SC,'sc003','SC Sno',2015002
Hbase(main):074:0>  put 'SC,'sc003,'SC Cno,123002
Hbase(main):075:0>  put 'SC,'sc003,'SC Score',77
Hbase(main):076:0>  put 'SC,'sc004,'SC Sno',2015002
Hbase(main):077:0>  put 'SC,'sc004,'SC Cno,'123003'
Hbase(main):078:0>  put 'SC,'sc004,'SC Score','99'
```

同时，完成以下指定功能

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

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 files 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

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

4、modifyData(String tableName, String row, String column, String val)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。

5、deleteRow(String tableName, String row)删除表 tableName 中 row 指定的行的记录。

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.HColumnDescriptor;
import org.apache.hadoop.hbase.HTableDescriptor;
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 java.io.IOException;

public class HBaseOperate {
    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;
    public static long ts;

public static void createTable(String tableName,String[]fields)throws IOException
    init();
    TableName tablename TableName.valueOf(tableName);

    if(admin.tableExists(tablename)){
    System.out.printin("table is exists!");
    admin.disableTable(tablename);
    admin.deleteTable(tablename);
  }
    TableDescriptorBuilder
    tableDescriptor
    TableDescriptorBuilder.newBuilder(tablename);
    for(String str:fields)

tableDescriptor.setColumnFamily(ColumnFamilyDescriptorBuilder.newBuilder(Bytes.toBytes(s
tr))
    .build());
    admin.createTable(tableDescriptor.build())
    }
    close();
}
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 I=fields.length;i++){
        Put put new Put(row.getBytes());
        Stringn]cols fields[i].split(":");
        put.addColumn(cols[o].getBytes(),cols[1].getBytes(),values[i].getBytes());
        table.put(put);
    }
        table.close();
        close();
}
public static void scanColumn(String tableName,String column)throws IOException
        admin new HBaseAdmin(conf);
        HTable table new HTable(conf,Bytes.toBytes(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();
}
public static void modifyData(String tableName,String row,String column,String val)throws
IOException{
        admin new HBaseAdmin(conf);
        HTable table new HTable(conf,Bytes.toBytes(tableName));
        Put put new Put(row.getBytes());
        Scanscan new Scan();
        ResultScanner resultScanner table.getscanner(scan);
        for (Result r:resultScanner)
          {
              for (Cell cell r.getColumnCells(row.getBytes(),column.getBytes()))
            {
                        ts=cell.getTimestamp();
              }
          }
         put.add(row.getBytes(),column.getBytes(),ts,val.getBytes());
         table.put(put);
         table.close();
         close();
}
public static void deleteRow(String tableName,String row)throws IOException{
        admin new HBaseAdmin(conf);
        HTable table new HTable(conf,Bytes.toBytes(tableName));
        Deletedelete new Delete(row.getBytes());
        table.delete(delete);
```

```
        table.close();
        close();
    }
        public static void main(String[] args) {
            String[] fields = {"Score"};
            try {
                createTable("person", fields);
            } catch (IOException e) {
                e.printStackTrace();
            }

            String[] fields = {"Score:Math", "Score:Computer Science", "Score:English"};
            String[] values = {"99", "80", "100"};
            try {
                addRecord("person", "Score", fields, values);
            } catch (IOException e) {
                e.printStackTrace();
            }

            try {
                scanColumn("Student", "S_No");
            } catch (IOException e) {
                e.printStackTrace();
            }

            try {
                modifyData("person", "Score", "Math", "100");
            } catch (IOException e) {
                e.printStackTrace();
            }

            try {
                deleteRow("person", "Score");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
```