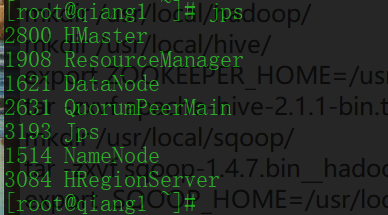
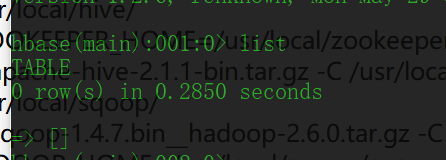
1、实验项目四 Hbase的模拟新浪微博功能

1. 实验内容与步骤



启动hadoop,zookeeper,hbase;

1. 创建hbase表，表名自行拟定，列族分别为attends(关注用户id),fans(粉丝用户id)



2、要求通过Hbase API完成hbase表的创建,且数据插入表全部使用hbase API完成

3、rowkey使用用户id



|  |
| --- |
| package com.hbase; 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;  */\*\*  \* Created by hadoopuser on 8/14/19.  \*/* public class hbase{    private static final String *TABLE\_NAME*="message";   public static final String *FAMILY\_NAME\_1* = "attends";  public static final String *FAMILY\_NAME\_2* = "fans";   //conf  private static Configuration getHBaseConfiguration() {  Configuration conf = HBaseConfiguration.*create*();  conf.set("hbase.zookeeper.quorum",  "qiang1:2181,qiang2:2181,qiang3:2181");  conf.set("zookeeper.znode.parent", "/hbase");  conf.set("hbase.rootdir", "hdfs://qiang1:8888/hbase");  return conf;  }   //createTable  private static void createTable(Configuration conf) throws IOException {  Connection connection = null;  Table table = null;  try {  connection = ConnectionFactory.*createConnection*(conf);  Admin admin = connection.getAdmin();   if (!admin.tableExists(TableName.*valueOf*(*TABLE\_NAME*))) {  //create table ,create family  HTableDescriptor tableDescriptor = new HTableDescriptor(TableName.*valueOf*(*TABLE\_NAME*));  HColumnDescriptor columnDescriptor\_1 = new HColumnDescriptor(Bytes.*toBytes*(*FAMILY\_NAME\_1*));  HColumnDescriptor columnDescriptor\_2 = new HColumnDescriptor(Bytes.*toBytes*(*FAMILY\_NAME\_2*));  tableDescriptor.addFamily(columnDescriptor\_1);  tableDescriptor.addFamily(columnDescriptor\_2);  admin.createTable(tableDescriptor);    } else {  System.*err*.println("table is exists!!!!!");  }  // rowkey attends fans // id name sex id name sex // 000001 000001 lihua female 000003 // 000001 000002 zhangsan male 000015 male // 000002 000004 male 000002 wanwu male // 000003 000006 female 000012 lisi male // 000004 000003 male 000023 liwu male  //put data  table = connection.getTable(TableName.*valueOf*(*TABLE\_NAME*));   Put put = new Put(Bytes.*toBytes*("000001")); //rowkey  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_1*), Bytes.*toBytes*("id"), Bytes.*toBytes*("000001"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_1*), Bytes.*toBytes*("name"), Bytes.*toBytes*("lihua"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_1*), Bytes.*toBytes*("sex"), Bytes.*toBytes*("female"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_2*), Bytes.*toBytes*("id"), Bytes.*toBytes*("000003"));    Put put1= new Put(Bytes.*toBytes*("000001")); //rowkey  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_1*), Bytes.*toBytes*("id"), Bytes.*toBytes*("000002"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_1*), Bytes.*toBytes*("name"), Bytes.*toBytes*("zhangsan"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_1*), Bytes.*toBytes*("sex"), Bytes.*toBytes*("male"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_2*), Bytes.*toBytes*("id"), Bytes.*toBytes*("000015"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_2*), Bytes.*toBytes*("sex"), Bytes.*toBytes*("male"));    Put put2 = new Put(Bytes.*toBytes*("000002"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_1*), Bytes.*toBytes*("id"), Bytes.*toBytes*("000004"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_1*), Bytes.*toBytes*("sex"), Bytes.*toBytes*("male"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_2*), Bytes.*toBytes*("id"), Bytes.*toBytes*("000002"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_2*), Bytes.*toBytes*("name"), Bytes.*toBytes*("wanwu"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_2*), Bytes.*toBytes*("sex"), Bytes.*toBytes*("male"));   Put put3 = new Put(Bytes.*toBytes*("000003"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_1*), Bytes.*toBytes*("id"), Bytes.*toBytes*("000006"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_1*), Bytes.*toBytes*("sex"), Bytes.*toBytes*("female"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_2*), Bytes.*toBytes*("id"), Bytes.*toBytes*("000012"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_2*), Bytes.*toBytes*("name"), Bytes.*toBytes*("lisi"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_2*), Bytes.*toBytes*("sex"), Bytes.*toBytes*("male"));   Put put4 = new Put(Bytes.*toBytes*("000004"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_1*), Bytes.*toBytes*("id"), Bytes.*toBytes*("000003"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_1*), Bytes.*toBytes*("sex"), Bytes.*toBytes*("male"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_2*), Bytes.*toBytes*("id"), Bytes.*toBytes*("000023"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_2*), Bytes.*toBytes*("name"), Bytes.*toBytes*("liwu"));  put.addColumn(Bytes.*toBytes*(*FAMILY\_NAME\_2*), Bytes.*toBytes*("sex"), Bytes.*toBytes*("male"));    table.put(put);  table.put(put1);  table.put(put2);  table.put(put3);  table.put(put4);    }catch (IOException e){  e.printStackTrace();  }finally {  //close  if (table != null) table.close();  if (connection != null) connection.close();  }  }    public static void main(String [] args){  Configuration conf=*getHBaseConfiguration*();  try {  *createTable*(conf);  } catch (IOException e) {  e.printStackTrace();  }finally {   }  }   } |

4、将数据插入上述表当中，数据内容如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| rowkey | attends | | | fans | | |
| id | name | sex | id | name | sex |
| 000001 | 000001 | lihua | female | 000003 |  |  |
| 000001 | 000002 | **zhangsan** | male | 000015 |  | male |
| 000002 | 000004 |  | male | 000002 | wanwu | male |
| 000003 | 000006 |  | female | 000012 | lisi | male |
| 000004 | 000003 |  | male | 000023 | liwu | male |