暨南大学本科实验报告专用纸

课程名称 云计算实验 成绩评定

实验项目名称 hbase 数据库 java API 编程 指导教师 魏林锋

实验项目编号 实验项目类型 设计 实验地点 N116

学生姓名 陈宇 学号 2020101642

学院 信息科学技术 系 计算机 专业 软件工程

实验时间 2022 年 11 月 16 日 上 午～ 11 月 16 日 上 午

8.1 实验目的

1）理解分布式列族数据库 HBase 工作原理。

2）通过实验掌握分布式列族数据库 HBase 的 JAVA API 编程。

8.2 实验内容

完成分布式列族数据库 HBase 的 JAVA API 编程。

8.3 实验环境

已经配置完成的 Hadoop 完全分布式环境。已经配置完成的 Zookeeper 集群模式环境。已经配置完成的 HBase 集群模式环境。环境配置如下：

Hadoop01：192.168.8.91 Hadoop02：192.168.8.92 Hadoop03：192.168.8.93 管理员用户：root / admin@1 Hadoop 用户：hadoop / hadoop

8.4 实验步骤

1、启动 Hadoop 集群。命令如下：

```
[root@master conf]# cd
```

[root@master ~]# su hadoop

[hadoop@master ~]$ start-all.sh

master 节点：

```
[hadoop@master ~]$ jps
4306 ResourceManager
3971 SecondaryNameNode
4435 NodeManager
4996 QuorumPeerMain
5444 HRegionServer
3782 DataNode
3639 NameNode
5496 Jps
5261 HMaster
```

slave1 节点：

```
[hadoop@slave1 ~]$ jps
3489 DataNode
4117 QuorumPeerMain
4246 HRegionServer
4343 Jps
3642 NodeManager
```

slave2 节点：

```
[hadoop@slave2 ~]$ jps
3649 NodeManager
4324 Jps
4073 QuorumPeerMain
3499 DataNode
4205 HRegionServer
```

2、启动 Zookeeper 集群。命令如下：

[hadoop@master ~]$ zkServer.sh start

[hadoop@master ~]$ jps

```
[hadoop@master ~]$ jps
4306 ResourceManager
3971 SecondaryNameNode
4435 NodeManager
4996 QuorumPeerMain
5444 HRegionServer
3782 DataNode
3639 NameNode
5496 Jps
5261 HMaster
```

[hadoop@slave1 ~]$ zkServer.sh start

[hadoop@slave1 ~]$jps

```
[hadoop@slave1 ~]$ jps
3489 DataNode
4117 QuorumPeerMain
4246 HRegionServer
4343 Jps
3642 NodeManager
```

[hadoop@slave2 ~]$ zkServer.sh start

[hadoop@slave2 ~]$jps

```
[hadoop@slave2 ~]$ jps
3649 NodeManager
4324 Jps
4073 QuorumPeerMain
3499 DataNode
4205 HRegionServer
```

[hadoop@master ~]$ zkServer.sh status

```
[hadoop@master ~]$ zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /usr/zookeeper/bin/../conf/zoo.cfg
Mode: follower
```

[hadoop@ slave1 ~]$ zkServer.sh status

```
[hadoop@slave1 ~]$  zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /usr/zookeeper/bin/../conf/zoo.cfg
Mode: follower
```

[hadoop@ slave2 ~]$ zkServer.sh status

```
[hadoop@slave2 ~]$ zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /usr/zookeeper/bin/../conf/zoo.cfg
Mode: leader
```

3、启动 HBase 集群。命令如下：

[hadoop@master ~]$ start-hbase.sh

Master 节点：

```
[hadoop@master ~]$ jps
4306 ResourceManager
3971 SecondaryNameNode
4435 NodeManager
4996 QuorumPeerMain
5444 HRegionServer
3782 DataNode
3639 NameNode
5496 Jps
5261 HMaster
```

Slave1 节点：

```
[hadoop@slave1 ~]$ jps
3489 DataNode
4117 QuorumPeerMain
4246 HRegionServer
4343 Jps
3642 NodeManager
```

Slave2 节点：

```
[hadoop@slave2 ~]$ jps
3649 NodeManager
4324 Jps
4073 QuorumPeerMain
3499 DataNode
4205 HRegionServer
```

4、在 Eclipse 中创建 MapReduce 项目，项目名为 TestDemo。在 Hbase Java API 运行时，需要导入 Hbase 包，导入步骤如下：

1）右击 Project，右键选择 Build Path->Configure Build Path

2）选择 Add Library。

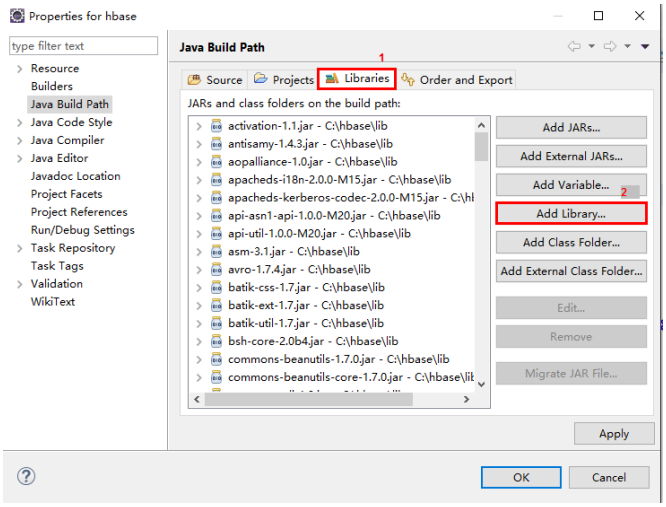

图 11-1

3）选择 User Library，点击 Next（如图 11-2 所示）.

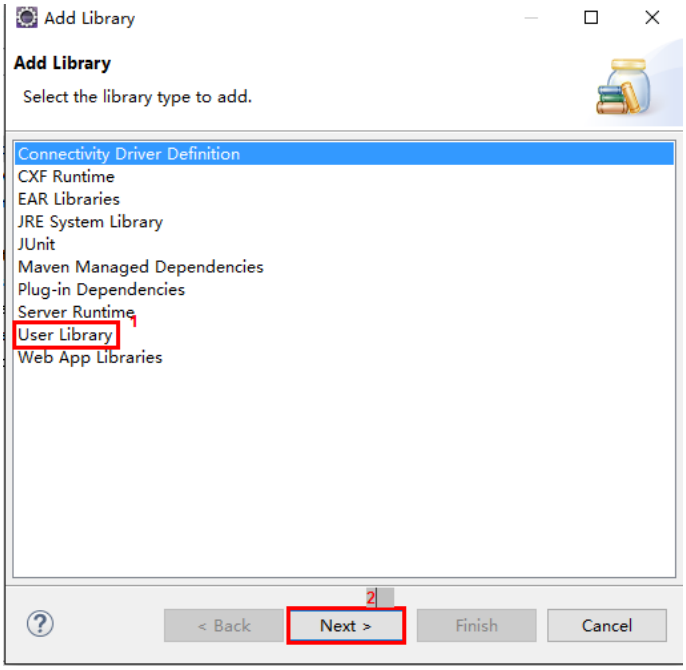

图 11-2

4）点击 User Libraries（如图 11-3 所示）.

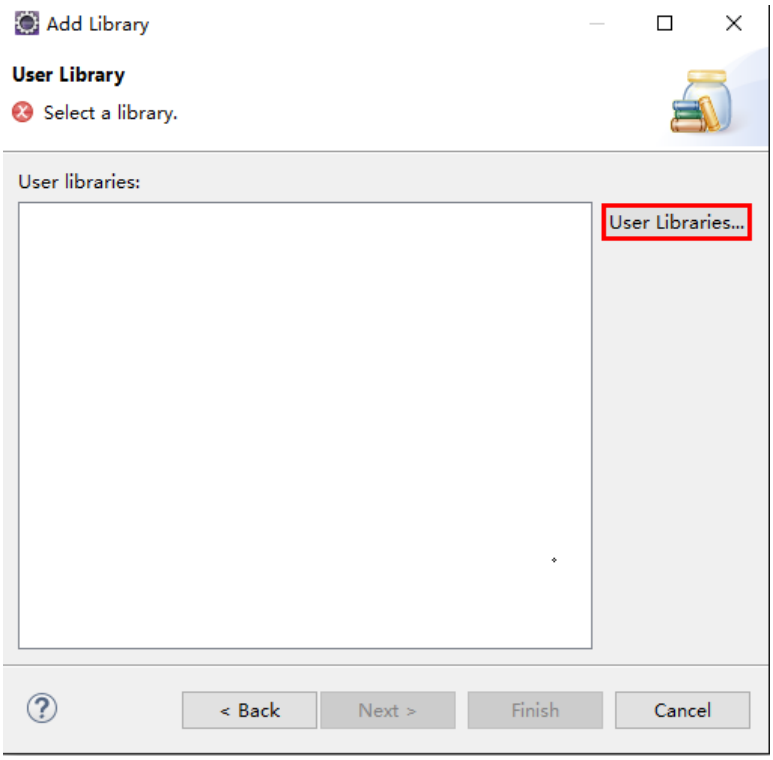

图 11-3

5）点击 New（如图 11-4 所示）

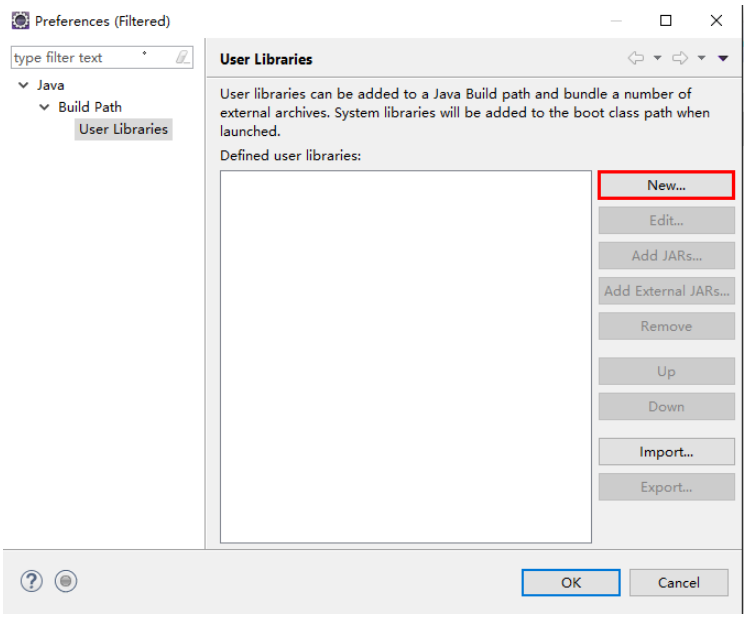

图 11-4

6）输入 library name 为 Hbase（name 为任意），点击 OK（如图 11-6 所示）.

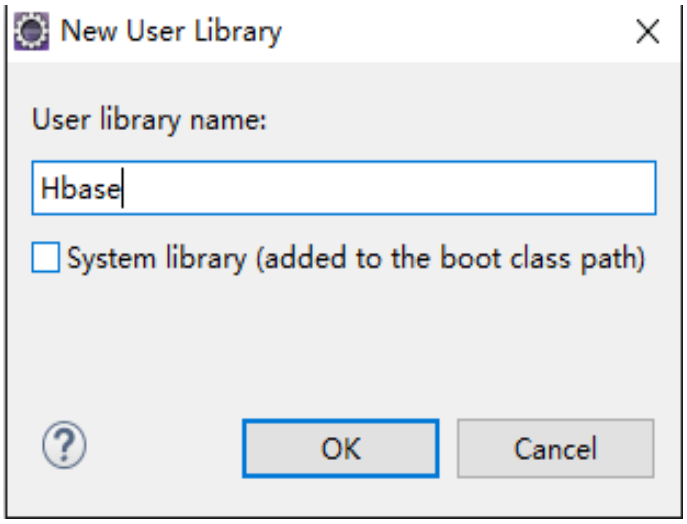

图 11-6

7）点击 Add External JARS，把 hbase 的 lib 文件夹的 jar 文件导入到 Hbase 这个库中（如图 11-7 所示）.

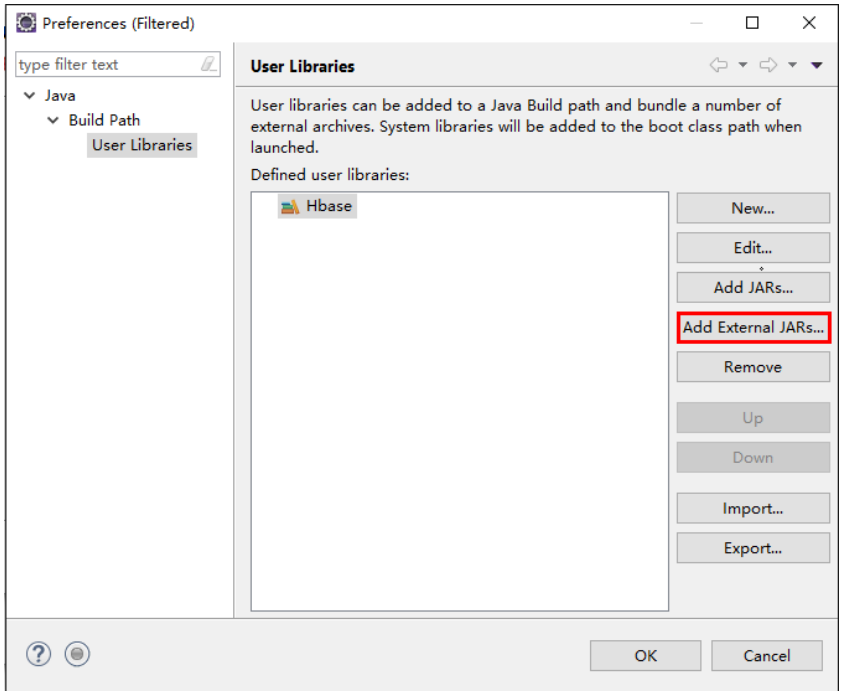

图 11-7

8）然后点击 OK（如图 11-8 所示）

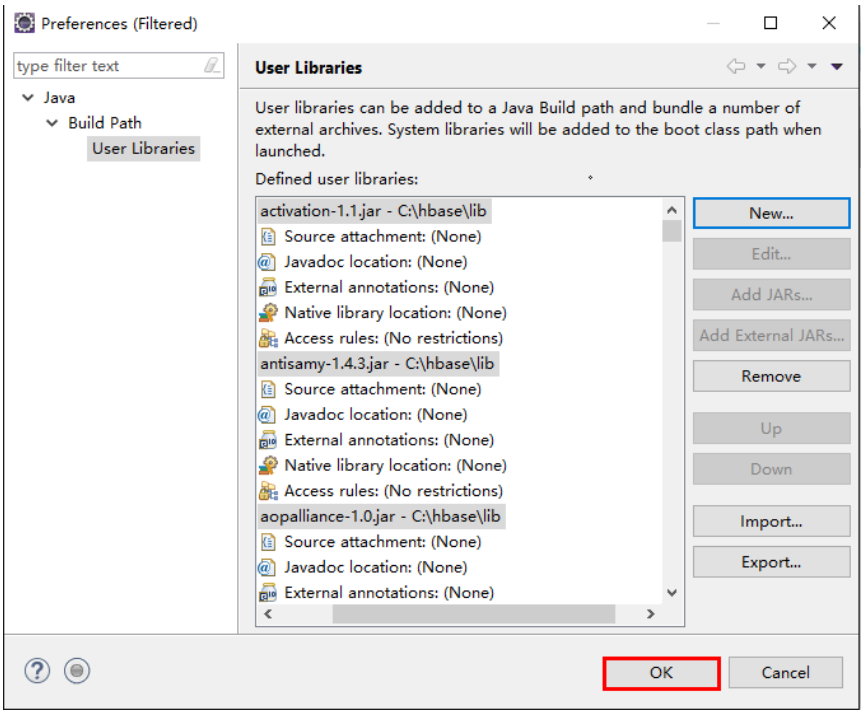

图 11-8

9）点击 hbase->Finish（如图 11-9 所示）.

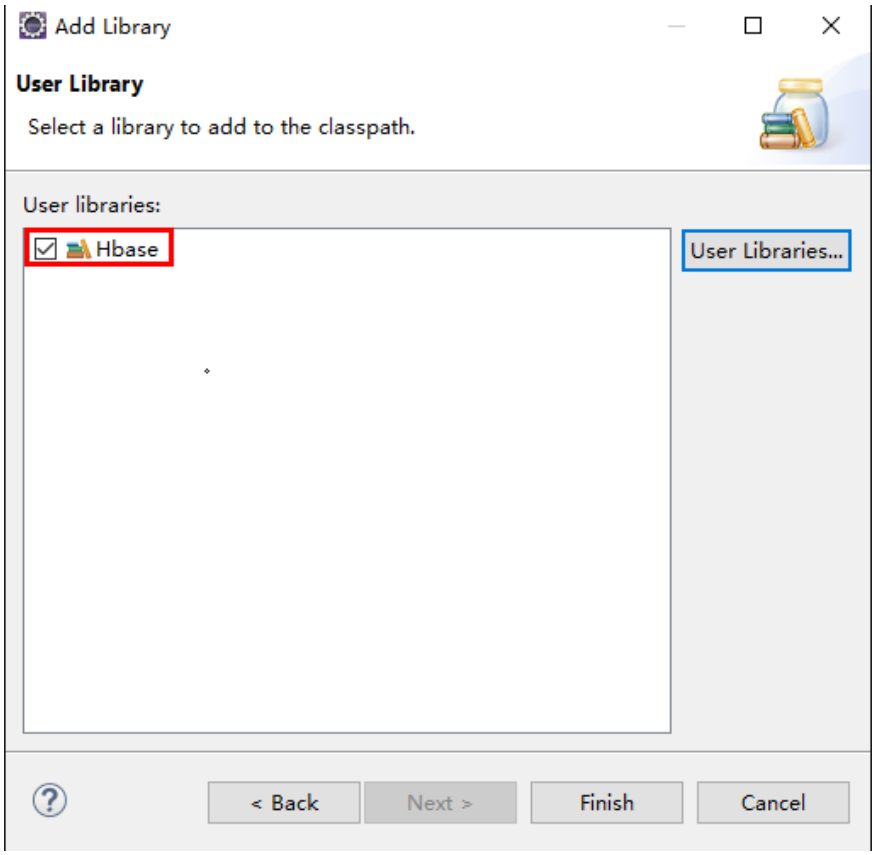

图 11-9

10）在项目 HBase 下增加一个文件夹 conf，将 Hbase 集群的 conf 目录下的配置文件 hbase-site.xml(从 Linux 上/usr/hbase/conf/habase-site.xml 下载)复制到该目录下，然后选择项目属性在Libraries->Add Class Folder,将刚刚增加的conf目录选上(如图11-10、11-11 所示）.

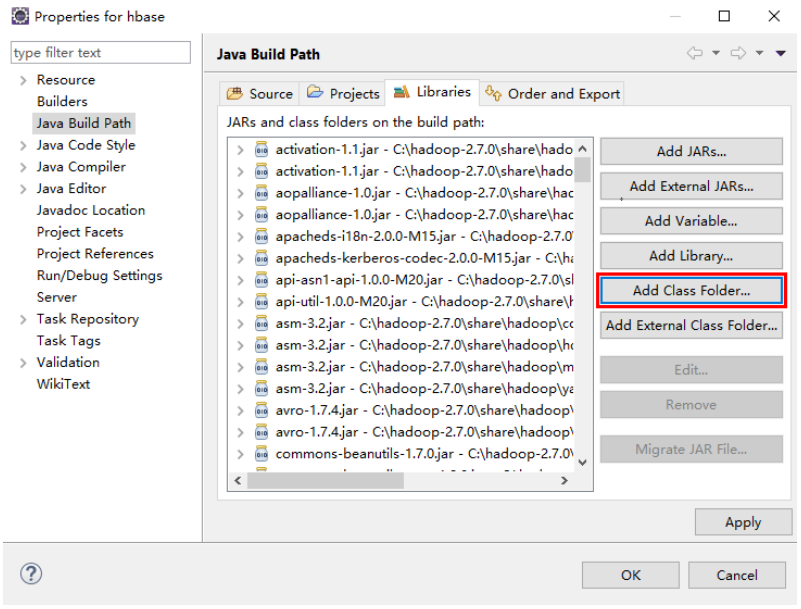

图 11-10

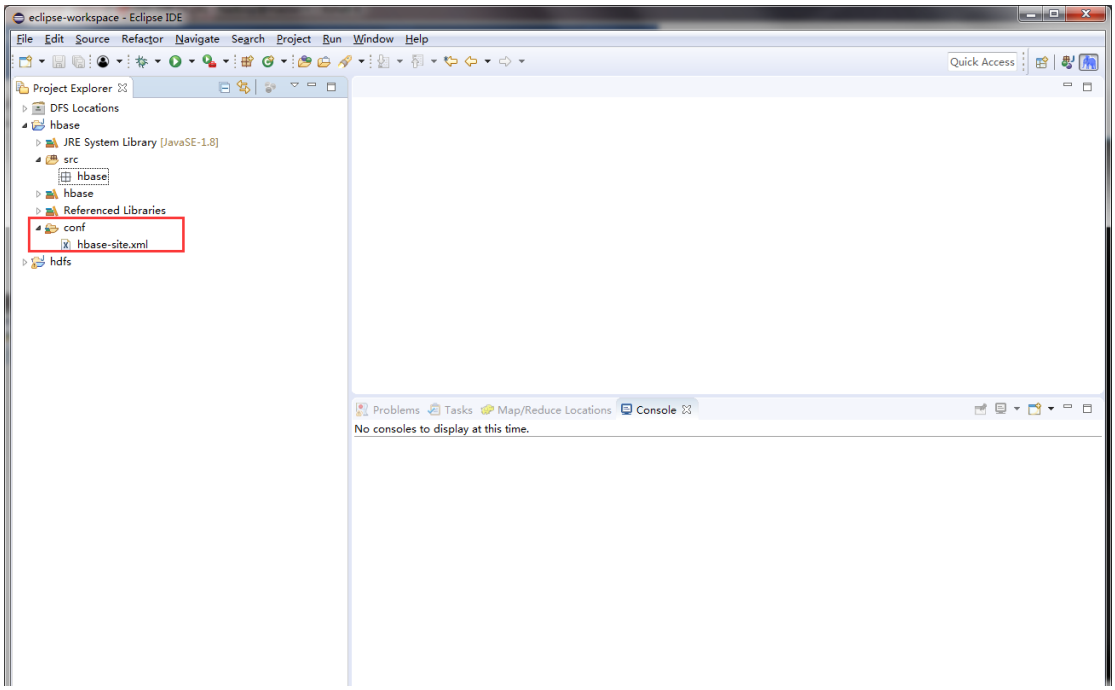

图 11-11

11）在 C:\Windows\System32\drivers\etc\hosts 文件中添加三项，如因权限原因不能修改 hosts 文件，可以把它剪切到系统桌面上进行修改后，再贴回到

C:\Windows\System32\drivers\etc\目录中。命令如下：

```
192.168.8.91    master
192.168.8.92    slave1
192.168.8.93    slave2
```

5、导入完 Hbase 包后，通过 Hbase Java API 编程实现 HBase 的各种操作。

1）创建 Student 表。代码如下：

```
package hbase;
import java.io.IOException;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.HColumnDescriptor;
import org.apache.hadoop.hbase.HTableDescriptor;
import org.apache.hadoop.hbase.client.HBaseAdmin;
import org.apache.hadoop.hbase.TableName;

import org.apache.hadoop.conf.Configuration;

public class create_student_table {
    public static void main(String[] args) throws IOException {
        // 初始化配置文件
        Configuration con = new Configuration();
        con.set("hbase.zookeeper.quorum", "master,slave1,slave2");

        // 实例化 HBaseAdmin
        HBaseAdmin admin = new HBaseAdmin(con);

        // 创建表
        HTableDescriptor tableDescriptor = new
HTableDescriptor(TableName.valueOf("student"));

        // 创建列族
        tableDescriptor.addFamily(new HColumnDescriptor("XH"));
        tableDescriptor.addFamily(new HColumnDescriptor("NA"));

        // 创建表
        admin.createTable(tableDescriptor);
        System.out.println("StudentTable Created");
    }
}
```

2）查看 HBase 中所有的表。代码如下：

```
package hbase;
```

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HTableDescriptor;
import org.apache.hadoop.hbase.MasterNotRunningException;
import org.apache.hadoop.hbase.client.HBaseAdmin;

public class List_student_table {
    public static void main(String args[])throws MasterNotRunningException, IOException{
        // 初始化配置文件
        Configuration con = new Configuration();
        con.set("hbase.zookeeper.quorum", "master,slave1,slave2");

        // 实例化 HBaseAdmin
        HBaseAdmin admin = new HBaseAdmin(con);

        // 使用 HBaseAdmin 对象获取所有表的列表
        HTableDescriptor[] tableDescriptor =admin.listTables();
        for (int i=0; i<tableDescriptor.length;i++ ){
            System.out.println(tableDescriptor[i].getNameAsString());   //打印所有的表名，
想当与做了 list 操作
        }
    }
}
```

3）禁用 Student 表。代码如下：

```
package hbase;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.MasterNotRunningException;
import org.apache.hadoop.hbase.client.HBaseAdmin;

public class disable_student_table {
    public static void main(String args[]) throws MasterNotRunningException, IOException{
        // 初始化配置文件
        Configuration con = new Configuration();
        con.set("hbase.zookeeper.quorum", "master,slave1,slave2");

        // 实例化 HBaseAdmin
        HBaseAdmin admin = new HBaseAdmin(con);

       // 判断 student 是否可用
```

```
        Boolean bool = admin.isTableDisabled("student");
        System.out.println(bool);

        // 将 student 表设置为不可用
        if(!bool){
            admin.disableTable("student");
            System.out.println("studentTable disabled");
        }
    }
}       // 使用 HBaseAdmin 对象获取所有表的列表
        HTableDescriptor[] tableDescriptor =admin.listTables();
        for (int i=0; i<tableDescriptor.length;i++ ){
            System.out.println(tableDescriptor[i].getNameAsString());  //打印所有的表名，
想当与做了 list 操作
        }
    }
}
```

4）启用 Student 表。代码如下：

```
package hbase;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.MasterNotRunningException;
import org.apache.hadoop.hbase.client.HBaseAdmin;

public class enable_student_table {
    public static void main(String args[]) throws MasterNotRunningException, IOException{
        // 初始化配置文件
        Configuration con = new Configuration();
        con.set("hbase.zookeeper.quorum", "master,slave1,slave2");

        // 实例化 HBaseAdmin
        HBaseAdmin admin = new HBaseAdmin(con);

        // 判断 student 是否可用
        Boolean bool = admin.isTableEnabled("student");
        System.out.println(bool);

        // 启用 student 表
        if(!bool){
            admin.enableTable("student");
```

```
                System.out.println("studentTable Enabled");
            }
        }
}
```

5）添加 Student 表的列族。代码如下：

```
package hbase;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.HColumnDescriptor;
import org.apache.hadoop.hbase.MasterNotRunningException;
import org.apache.hadoop.hbase.client.HBaseAdmin;

public class addColomn_student_table {
        public static void main(String args[]) throws MasterNotRunningException, IOException{
                // 初始化配置文件
                Configuration con = new Configuration();
                con.set("hbase.zookeeper.quorum", "master,slave1,slave2");

                // 实例化 HBaseAdmin
                HBaseAdmin admin = new HBaseAdmin(con);

                // 实例化描述符类
                HColumnDescriptor columnDescriptor1 = new HColumnDescriptor("sex");
                HColumnDescriptor columnDescriptor2 = new HColumnDescriptor("age");

                // 添加列族
                admin.addColumn("student", columnDescriptor1);
                admin.addColumn("student", columnDescriptor2);
                System.out.println("studentcoloumn added");
            }
}
```

6）删除 Student 表的列族。代码如下：

```
package hbase;

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

```
import org.apache.hadoop.hbase.client.HBaseAdmin;

public class deleteColoumn_student_table {
    public static void main(String args[]) throws MasterNotRunningException, IOException{
        // 初始化配置文件
        Configuration con = new Configuration();
        con.set("hbase.zookeeper.quorum", "master,slave1,slave2");

        // 实例化 HBaseAdmin
        HBaseAdmin admin = new HBaseAdmin(con);

        // 删除列族
        admin.deleteColumn("student","sex");
        System.out.println("studentcoloumn deleted");
    }
}
```

7）验证 Student 表是否存在。代码如下：

```
package hbase;

import java.io.IOException;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.client.HBaseAdmin;

public class exists_student_table {
    public static void main(String args[])throws IOException{

        // 初始化配置文件
        Configuration con = new Configuration();
        con.set("hbase.zookeeper.quorum", "master,slave1,slave2");

        // 实例化 HBaseAdmin
        HBaseAdmin admin = new HBaseAdmin(con);

        // 判断表是否存在
        boolean bool = admin.tableExists("student");
        System.out.println(bool);
    }
}
```

8）删除 Student 表。代码如下：

```
package hbase;

import java.io.IOException;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.client.HBaseAdmin;

public class delete_student_table {
	public static void main(String[] args) throws IOException {
		// 初始化配置文件
		Configuration con = new Configuration();
		con.set("hbase.zookeeper.quorum", "master,slave1,slave2");

		// 实例化 HBaseAdmin
		HBaseAdmin admin = new HBaseAdmin(con);

		// 禁用 student 表
		admin.disableTable("student");

		// 删除 student 表
		admin.deleteTable("student");
		System.out.println("studentTable deleted");
	}
}
```

9）创建表和查询表信息综合实例。代码如下：

```
package hbase;

import com.google.common.base.Preconditions;
import com.google.common.base.Strings;
import com.google.common.collect.Lists;
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.HBaseAdmin;
import org.junit.Test;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.io.IOException;
import java.util.Iterator;
```

```
import java.util.List;

public class CreateHBaseTable {

    /*创建表单*/
    public void createTable(HBaseAdmin admin, String tableName, List<String> columnNames) throws IOException {
        Preconditions.checkArgument(!Strings.isNullOrEmpty(tableName), "table name is not allowed null or empty !");
        Preconditions.checkArgument((null != columnNames && columnNames.size() > 0), "colume is not allowed empty !");
        if (null == admin) {
            throw new IllegalStateException("admin is empty !");
        }

        HTableDescriptor hTableDescriptor = new HTableDescriptor(TableName.valueOf(tableName));
        for (String colName : columnNames) {
            hTableDescriptor.addFamily(new HColumnDescriptor(colName));
        }
        admin.createTable(hTableDescriptor);
    }

    /*查询所有表单*/
    public List<String> scanTables(HBaseAdmin admin) {
        HTableDescriptor[] hTableDescriptors = new HTableDescriptor[0];
        if (null == admin) {
            throw new IllegalStateException("admin is empty !");
        }
        try {
            hTableDescriptors = admin.listTables();
        } catch (IOException e) {
            e.printStackTrace();
        }
        List<String> tmpList = Lists.newArrayList();
        for (HTableDescriptor hTableDescriptor : hTableDescriptors) {
            tmpList.add(String.valueOf(hTableDescriptor.getNameAsString()));
        }
        return tmpList;
    }

    public static void main(String[] args) throws IOException {
        Configuration conf = new Configuration();
        conf.set("hbase.zookeeper.quorum", "master,slave1,slave2");
```

```
            HBaseAdmin admin = new HBaseAdmin(conf);
            List list = Lists.newArrayList();
            list.add("Student_col");
            list.add("Student_col1");
            list.add("Student_col2");
            list.add("Student_col3");
            CreateHBaseTable createHbaseTable = new CreateHBaseTable();
            createHbaseTable.createTable(admin, "Student", list);

            List<String> tableNames = createHbaseTable.scanTables(admin);
            Iterator iterator = tableNames.iterator();
            while (iterator.hasNext()) {
                System.out.println(String.valueOf(iterator.next()));
            }
        }
}
```

10）HBase 的 JDBC 操作。代码如下：

```
package hbase;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Map.Entry;
import java.util.NavigableMap;
import java.util.Set;

import org.apache.commons.io.IOUtils;
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 org.apache.hadoop.hbase.client.Get;
import org.apache.hadoop.hbase.client.Put;
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;
```

```
/**
 * JDBC 连接 Hbase 测试
 *
 * @author lucky
 *
 */
public class HBaseJavaAPI {

	/**
	 * 列族
	 */
	private static final String FAMILY_NAMES = "a,b,c,d,e,f";
	/**
	 * 表名
	 */
	private static final String TABLE_NAME = "bigTab";
	/**
	 * 连接对象
	 */
	private static Connection conn;
	/**
	 * Hbase 集群地址，多个地址用“,”分开
	 */
	private static final String HOSTS = "master,slave1,slave2";

	static {
		Configuration conf = HBaseConfiguration.create();
		conf.set("hbase.zookeeper.quorum", "master,slave1,slave2");
		try {
			conn = ConnectionFactory.createConnection(conf);
			createTable(TABLE_NAME, FAMILY_NAMES);
		} catch (IOException e) {
			e.printStackTrace();
		}
	}

	/**
	 * 创建表
	 *
	 * @param tableName
	 *                表名
	 * @param familyNames
	 *                列族，多个列族用“,”分开
	 * @throws IOException
```

```
     */
    private static void createTable(String tableName, String familyNames) throws IOException {
        Admin admin = null;
        TableName tn = TableName.valueOf(tableName);

        admin = conn.getAdmin();
        if (!admin.tableExists(tn)) {
            HTableDescriptor descriptor = new HTableDescriptor(tn);
            for (String familyName : familyNames.split(",")) {

                descriptor.addFamily(new HColumnDescriptor(familyName));
            }
            admin.createTable(descriptor);
            System.out.println("创建表：" + tableName);
        }

    }

    /**
     * 删除表
     *
     * @param tableName
     *                  表名
     * @throws IOException
     */
    private static void dropTable(String tableName) throws IOException {
        Admin admin = null;
        try {
            TableName tn = TableName.valueOf(tableName);
            admin = conn.getAdmin();
            if (admin.tableExists(tn)) {
                admin.disableTable(tn);
                admin.deleteTable(tn);
                System.out.println("删除表：" + tableName);
            }
        } finally {
            IOUtils.closeQuietly(admin);
        }

    }

    /**
     * 添加数据
     *
```

```
     * @param tableName
     *                  表名
     * @param rowKey
     *                  行号
     * @param datas
     *                  数据
     * @throws IOException
     */
    private static void add(String tableName, String rowKey, List<String[]> datas) throws IOException {
        Table table = conn.getTable(TableName.valueOf(tableName));
        try {
            Put list = new Put(rowKey.getBytes());
            for (String[] data : datas) {
                list.addColumn(data[0].getBytes(), data[1].getBytes(), data[2].getBytes());
            }
            table.put(list);
        }
        catch(IOException ex) {
            ex.printStackTrace();
        }
        finally {
            if (table != null) {
                IOUtils.closeQuietly(table);
            }
        }
    }

    /**
     * 查看所有数据
     *
     * @param tableName
     * @throws IOException
     */
    private static void scan(String tableName) throws IOException {
        Table table = conn.getTable(TableName.valueOf(tableName));
        try {
            ResultScanner rs = table.getScanner(new Scan());
            for (Result r : rs) {
                NavigableMap<byte[], NavigableMap<byte[], NavigableMap<Long, byte[]>>> map = r.getMap();
                Set<Entry<byte[], NavigableMap<byte[], NavigableMap<Long, byte[]>>>> set = map.entrySet();
                for (Entry<byte[], NavigableMap<byte[], NavigableMap<Long, byte[]>>>
```

```
entry : set) {

                        Set<Entry<byte[], NavigableMap<Long, byte[]>>> entrySet =
entry.getValue().entrySet();
                        for (Entry<byte[], NavigableMap<Long, byte[]>> entry2 : entrySet) {
                            System.out.print(new String(r.getRow()));
                            System.out.print("\t");
                            System.out.print(new String(entry.getKey()));
                            System.out.print(":");
                            System.out.print(new String(entry2.getKey()));
                            System.out.print(" => ");
                            System.out.println(new
String(entry2.getValue().firstEntry().getValue()));
                        }
                    }
                }
            }
            catch(IOException ex) {
                ex.printStackTrace();
            }
            finally {
                if (table != null) {
                    IOUtils.closeQuietly(table);
                }
            }
        }

        /**
         * 根据 rowKey 查询
         *
         * @param tableName
         * @param rowKey
         * @throws IOException
         */
        private static void scanByrowKey(String tableName, String rowKey) throws IOException {

            Table table = conn.getTable(TableName.valueOf(tableName));
            try {
                Result r = table.get(new Get(rowKey.getBytes()));

                NavigableMap<byte[], NavigableMap<byte[], NavigableMap<Long, byte[]>>>
map = r.getMap();
                Set<Entry<byte[], NavigableMap<byte[], NavigableMap<Long, byte[]>>>> set =
map.entrySet();
```

```
                for (Entry<byte[], NavigableMap<byte[], NavigableMap<Long, byte[]>>> entry :
set) {

                    Set<Entry<byte[], NavigableMap<Long, byte[]>>> entrySet =
entry.getValue().entrySet();
                    for (Entry<byte[], NavigableMap<Long, byte[]>> entry2 : entrySet) {
                        System.out.print(new String(r.getRow()));
                        System.out.print("\t");
                        System.out.print(new String(entry.getKey()));
                        System.out.print(":");
                        System.out.print(new String(entry2.getKey()));
                        System.out.print(" => ");
                        System.out.println(new String(entry2.getValue().firstEntry().getValue()));
                    }
                }
            }
            catch(IOException ex) {
                ex.printStackTrace();
            }
            finally {
                if (table != null) {
                    IOUtils.closeQuietly(table);
                }
            }
        }

        public static void main(String[] args) throws IOException {
            List<String[]> list = new ArrayList<>();
            list.add("a, a1, www1".split(","));
            list.add("a, a2, www2".split(","));
            list.add("a, a3, www3".split(","));
            list.add("a, a4, www4".split(","));
            list.add("b, a1, www5".split(","));
            list.add("b, a2, www6".split(","));
            list.add("b, a3, www7".split(","));
            list.add("b, a4, www8".split(","));
            list.add("c, a5, www9".split(","));
            list.add("c, a6, www10".split(","));
            list.add("c, a1, www11".split(","));
            list.add("c, a2, www12".split(","));
            list.add("d, a3, www13".split(","));
            list.add("e, a4, www14".split(","));
            list.add("f, a5, www15".split(","));
            add(TABLE_NAME, "100001", list);
```

```
            list = new ArrayList<>();
            list.add("a, a1, vvv1".split(","));
            list.add("a, a2, vvv2".split(","));
            list.add("a, a3, vvv3".split(","));
            list.add("a, a4, vvv4".split(","));
            list.add("b, a1, vvv5".split(","));
            list.add("b, a2, vvv6".split(","));
            list.add("b, a3, vvv7".split(","));
            list.add("b, a4, vvv8".split(","));
            list.add("c, a5, vvv9".split(","));
            list.add("c, a6, vvv10".split(","));
            list.add("c, a1, vvv11".split(","));
            list.add("c, a2, vvv12".split(","));
            list.add("d, a3, vvv13".split(","));
            list.add("e, a4, vvv14".split(","));
            list.add("f, a5, vvv15".split(","));
            add(TABLE_NAME, "100002", list);

            scan(TABLE_NAME);
            System.out.println();
            scanByrowKey(TABLE_NAME, "100002");
            dropTable(TABLE_NAME);
        }
}
```