Mysql

1. 根据题目给出的Student表，在MySQL数据库中完成如下操作：

（1）在MySQL中创建Student表，并录入数据；

创建Student表的SQL语句如下：

```
CREATE TABLE student(
    NAME VARCHAR(30) NOT NULL,
    English TINYINT UNSIGNED NOT NULL,
    Math TINYINT UNSIGNED NOT NULL,
    Computer TINYINT UNSIGNED NOT NULL
);
```

向Student表中插入两条记录的SQL语句如下：

```
insert into student values("zhangsan",69,86,77);
insert into student values("lisi",55,100,88);
```

（2）用SQL语句输出Student表中的所有记录；

输出Student表中的所有记录的SQL语句如下：

```
select * from student;
```

上述SQL语句执行后的结果截图如图所示。

```
mysql> use student;
Database changed
mysql> select * from student;
+----------+---------+------+----------+
| name     | English | Math | Computer |
+----------+---------+------+----------+
| zhangsan |      69 |   86 |       77 |
| lisi     |      55 |  100 |       88 |
+----------+---------+------+----------+
```

（3）查询zhangsan的Computer成绩；

查询zhangsan的Computer成绩的SQL语句如下：

```
select name , Computer from student where name = "zhangsan";
```

```
mysql> select name , Computer from student where name = "zhangsan";
+----------+----------+
| name     | Computer |
+----------+----------+
| zhangsan |       77 |
+----------+----------+
1 row in set (0.00 sec)
```

（4）修改lisi的Math成绩，改为95。

修改lisi的Math成绩的SQL语句如下：

```
update student set Math=95 where name="lisi";
```

```
mysql> update student set Math=95 where name="lisi";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select name,Math from student where name = "lisi";
+------+------+
| name | Math |
+------+------+
| lisi |   95 |
+------+------+
```

2. 使用MySQL的JAVA客户端编程实现以下操作:

向Student表添加上述记录的Java代码如下：

```
package com.xusheng.nosql.mysql;
import java.sql.*;
public class mysql_test {

    /**
     * @param xusheng
     */
    //JDBC DRIVER and DB
    static final String  DRIVER="com.mysql.jdbc.Driver";
    static final String DB="jdbc:mysql://localhost/student?useSSL=false";
    //Database auth
    static final String USER="root";
    static final String PASSWD="root";

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Connection conn=null;
        Statement stmt=null;
        try {
            //加载驱动程序
            Class.forName(DRIVER);
```

```
            System.out.println("Connecting to a selected database...");
            //打开一个连接
            conn=DriverManager.getConnection(DB, USER, PASSWD);
            //执行一个查询
            stmt=conn.createStatement();
            String sql="insert into student values('scofield',45,89,100)";
            stmt.executeUpdate(sql);
            System.out.println("Inserting records into the table successfully!");
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }finally
        {
            if(stmt!=null)
                try {
                    stmt.close();
                } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            if(conn!=null)
                try {
                    conn.close();
                } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
        }
    }
}
```

（2）获取scofield的English成绩信息

获取scofield的English成绩信息的Java代码如下：

```
package com.xusheng.nosql.mysql;
import java.sql.*;
public class mysql_qurty {

    /**
     * @param args
     */
    //JDBC DRIVER and DB
    static final String  DRIVER="com.mysql.jdbc.Driver";
    static final String DB="jdbc:mysql://localhost/student?useSSL=false";
    //Database auth
    static final String USER="root";
```

```
    static final String PASSWD="root";

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Connection conn=null;
        Statement stmt=null;
        ResultSet rs=null;
        try {
            //加载驱动程序
            Class.forName(DRIVER);
            System.out.println("Connecting to a selected database...");
            //打开一个连接
            conn=DriverManager.getConnection(DB, USER, PASSWD);
            //执行一个查询
            stmt=conn.createStatement();
            String sql="select name,English from student where name='scofield' ";
            //获得结果集
            rs=stmt.executeQuery(sql);
            System.out.println("name"+"\t\t"+"English");
            while(rs.next())
            {
                System.out.print(rs.getString(1)+"\t\t");
                System.out.println(rs.getInt(2));
            }
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }finally
        {
            if(rs!=null)
                try {
                    rs.close();
                } catch (SQLException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                }
            if(stmt!=null)
                try {
                    stmt.close();
                } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            if(conn!=null)
                try {
                    conn.close();
```

```
                } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            }
        }
    }
```

HBase

（1）用Hbase Shell命令创建学生表Student；

创建Student表的命令如下：

```
create 'student','score'
```

向Student表中插入上面表格数据的命令如下：

```
put 'student','zhangsan','score:English','69'
put 'student','zhangsan','score:Math','86'
put 'student','zhangsan','score:Computer','77'
put 'student','lisi','score:English','55'
put 'student','lisi','score:Math','100'
put 'student','lisi','score:Computer','88'
```

```
=> ["Course", "SC", "Student", "s1", "stu"]
hbase(main):006:0> create 'student','score'
Created table student
Took 1.3216 seconds
=> Hbase::Table - student
hbase(main):007:0> put 'student','zhangsan','score:English','69'
Took 0.2069 seconds
hbase(main):008:0> put 'student','zhangsan','score:Math','86'
Took 0.0081 seconds
hbase(main):009:0> put 'student','zhangsan','score:Computer','77'
Took 0.0063 seconds
hbase(main):010:0> put 'student','lisi','score:English','55'
Took 0.0109 seconds
hbase(main):011:0> put 'student','lisi','score:Math','100'
Took 0.0080 seconds
hbase(main):012:0> put 'student','lisi','score:Computer','88'
```

（2）用scan命令浏览Student表的相关信息；

用scan指令浏览Student表相关信息的命令如下：

```
scan 'student'
```

```
hbase(main):013:0> scan 'student'
ROW                     COLUMN+CELL
 lisi                   column=score:Computer, timestamp=1652956844379, value=88
 lisi                   column=score:English, timestamp=1652956811984, value=55
 lisi                   column=score:Math, timestamp=1652956820435, value=100
 zhangsan               column=score:Computer, timestamp=1652956805094, value=77
 zhangsan               column=score:English, timestamp=1652956789644, value=69
 zhangsan               column=score:Math, timestamp=1652956797579, value=86
2 row(s)
```

（3）查询zhangsan的Computer成绩；

查询zhangsan的Computer成绩的命令如下：

```
get 'student','zhangsan','score:Computer'
```

```
hbase(main):014:0> get 'student','zhangsan','score:Computer'
COLUMN                  CELL
 score:Computer         timestamp=1652956805094, value=77
1 row(s)
```

（4）修改lisi的Math成绩，改为95。

修改lisi的Math成绩的命令如下：

```
put 'student','lisi','score:Math','95'
```

```
hbase(main):015:0> put 'student','lisi','score:Math','95'
Took 0.0072 seconds
hbase(main):016:0> get 'student','lisi','score:Math'
COLUMN                  CELL
 score:Math             timestamp=1652957016474, value=95
1 row(s)
Took 0.0077 seconds
```

```
package com.xusheng.nosql.hbase;

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 hbase_insert {

    /**
     * @param xusheng
     */
```

```
    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        configuration  = HBaseConfiguration.create();
        //configuration.set("hbase.rootdir","hdfs://localhost:9000/hbase");
        //configuration.set("hbase.rootdir","hdfs://hadoop102:8020/HBase");
        configuration.set("hbase.zookeeper.quorum","hadoop102,hadoop103,hadoop104");

        try{
            connection = ConnectionFactory.createConnection(configuration);
            admin = connection.getAdmin();
        }catch (IOException e){
            e.printStackTrace();
        }
        try {
            insertRow("student","scofield","score","English","45");
            insertRow("student","scofield","score","Math","89");
            insertRow("student","scofield","score","Computer","100");
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        close();
    }
    public static void insertRow(String tableName,String rowKey,String colFamily,
                                 String col,String val) throws IOException {
        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();
    }
    public static void close(){
        try{
            if(admin != null){
                admin.close();
            }
            if(null != connection){
                connection.close();
            }
        }catch (IOException e){
            e.printStackTrace();
        }
    }
}
```

```
hbase(main):019:0> scan 'student'
ROW                     COLUMN+CELL
 lisi                   column=score:Computer, timestamp=1652956844379, value=88
 lisi                   column=score:English, timestamp=1652956811984, value=55
 lisi                   column=score:Math, timestamp=1652957016474, value=95
 scofield               column=score:Computer, timestamp=1652957280050, value=100
 scofield               column=score:English, timestamp=1652957279677, value=45
 scofield               column=score:Math, timestamp=1652957279686, value=89
 zhangsan               column=score:Computer, timestamp=1652956805094, value=77
 zhangsan               column=score:English, timestamp=1652956789644, value=69
 zhangsan               column=score:Math, timestamp=1652956797579, value=86
3 row(s)
```

```java
package com.xusheng.nosql.hbase;

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.Put;
import org.apache.hadoop.hbase.client.Result;
import org.apache.hadoop.hbase.client.Table;

public class hbase_query {

    /**
     * @param args
     */
    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        configuration  = HBaseConfiguration.create();
        //configuration.set("hbase.rootdir","hdfs://localhost:9000/hbase");
        //configuration.set("hbase.rootdir","hdfs://hadoop102:8020/HBase");
        configuration.set("hbase.zookeeper.quorum","hadoop102,hadoop103,hadoop104");

        try{
            connection = ConnectionFactory.createConnection(configuration);
            admin = connection.getAdmin();
        }catch (IOException e){
            e.printStackTrace();
        }
        try {
```

```
            getData("student","scofield","score","English");
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        close();
    }
    public static void getData(String tableName,String rowKey,String colFamily,
                               String col)throws  IOException{
        Table table = connection.getTable(TableName.valueOf(tableName));
        Get get = new Get(rowKey.getBytes());
        get.addColumn(colFamily.getBytes(),col.getBytes());
        Result result = table.get(get);
        showCell(result);
        table.close();
    }
    public static void showCell(Result result){
        Cell[] cells = result.rawCells();
        for(Cell cell:cells){
            System.out.println("RowName:"+new String(CellUtil.cloneRow(cell))+" ");
            System.out.println("Timetamp:"+cell.getTimestamp()+" ");
            System.out.println("column Family:"+new
String(CellUtil.cloneFamily(cell))+" ");
            System.out.println("row Name:"+new
String(CellUtil.cloneQualifier(cell))+" ");
            System.out.println("value:"+new String(CellUtil.cloneValue(cell))+" ");
        }
    }
    public static void close(){
        try{
            if(admin != null){
                admin.close();
            }
            if(null != connection){
                connection.close();
            }
        }catch (IOException e){
            e.printStackTrace();
        }
    }
}
```

```
RowName:scofield
Timetamp:1652957279677
column Family:score
row Name:English
value:45
```

Redis

Student键值对如下：

```
zhangsan: {
      English: 69
      Math: 86
      Computer: 77
}
lisi: {
      English: 55
      Math: 100
      Computer: 88
}
```

1. 根据上面给出的键值对，完成如下操作：

（1）用Redis的哈希结构设计出学生表Student（键值可以用student.zhangsan和student.lisi来表示两个键值属于同一个表）；

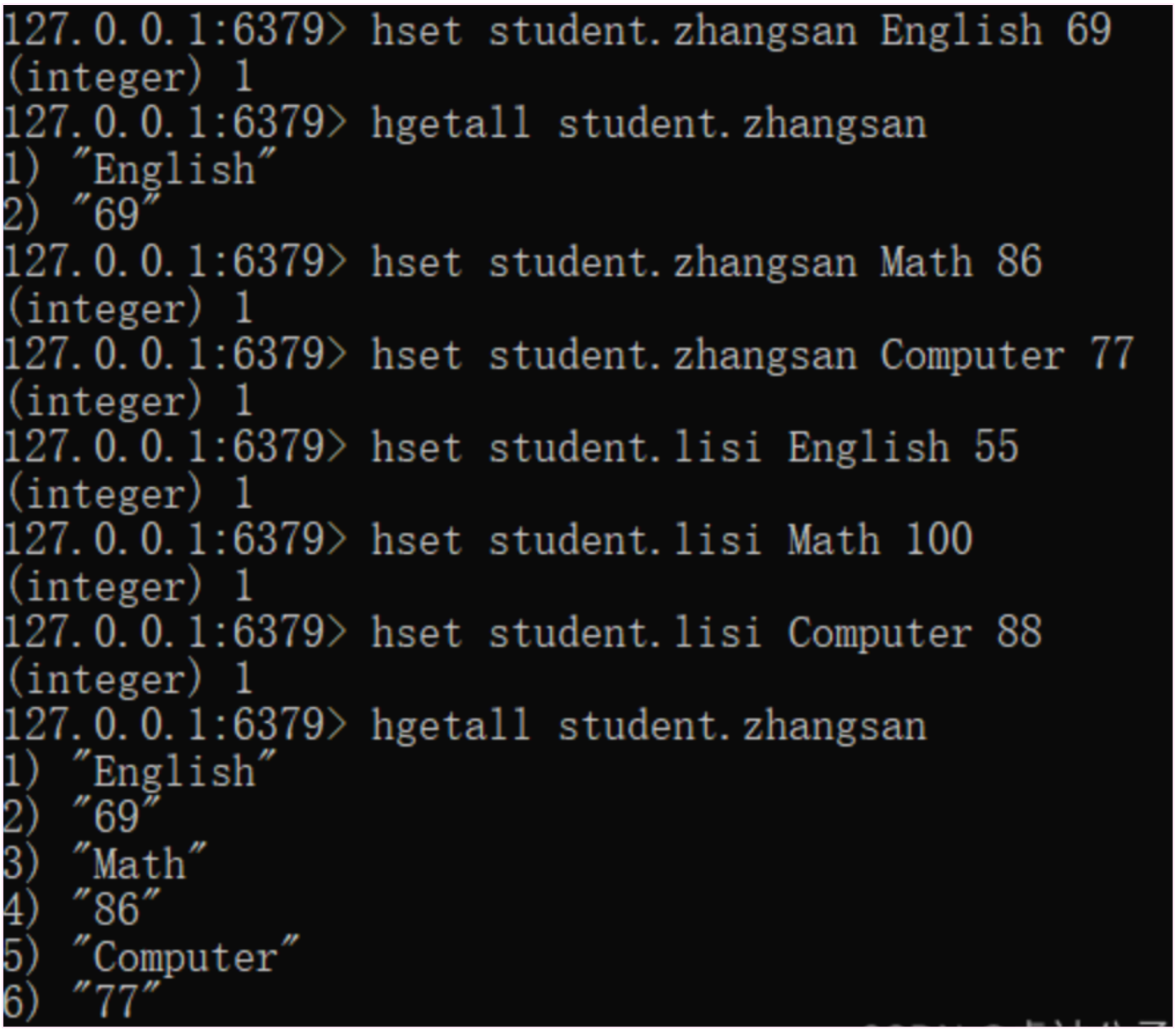
插入上述键值对的命令如下：

```
hset student.zhangsan English 69
hset student.zhangsan Math 86
hset student.zhangsan Computer 77
hset student.lisi English 55
hset student.lisi Math 100
hset student.lisi Computer 88
```

（2）用hgetall命令分别输出zhangsan和lisi的成绩信息；

查询zhangsan成绩信息的命令如下：

```
hgetall student.zhangsan
```



查询lisi成绩信息的命令如下：

```
hgetall student.lisi
```

（3）用hget命令查询zhangsan的Computer成绩；

查询zhangsan的Computer成绩的命令如下：

```
hget student.zhangsan Computer
```

4）修改lisi的Math成绩，改为95。

修改lisi的Math成绩的命令如下：

```
hset student.lisi Math 95
```

2. 用Redis的JAVA客户端编程(jedis)，实现如下操作：

（1）添加数据：English:45 Math:89 Computer:100

该数据对应的键值对形式如下：

```
scofield: {
       English: 45
       Math: 89
       Computer: 100
}
```

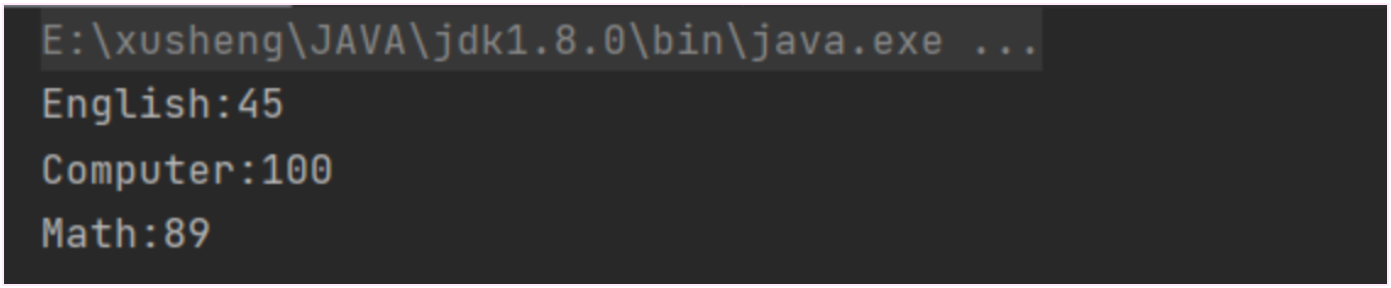
完成添加数据操作的Java代码如下：

```
package com.xusheng.nosql.redis;
import java.util.Map;
import redis.clients.jedis.Jedis;

public class jedis_test {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Jedis jedis = new Jedis("localhost");
        jedis.hset("student.scofield", "English","45");
        jedis.hset("student.scofield", "Math","89");
        jedis.hset("student.scofield", "Computer","100");
        Map<String,String>  value = jedis.hgetAll("student.scofield");
        for(Map.Entry<String, String> entry:value.entrySet())
        {
            System.out.println(entry.getKey()+":"+entry.getValue());
        }
    }
}
```

在idea中执行程序时，在idea控制台输出的信息截图如图所示:



（2）获取scofield的English成绩信息

获取scofield的English成绩信息的Java代码如下：

```
package com.xusheng.nosql.redis;
```

```java
import java.util.Map;
import redis.clients.jedis.Jedis;

public class jedis_query {

    /**
     * @param xusheng
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Jedis jedis = new Jedis("localhost");
        String value=jedis.hget("student.scofield", "English");
        System.out.println("scofield's English score is:    "+value);
    }
}
```