

**数学与信息学院学生实验报告**

**实验课程名称：** 大数据技术概论 **教师：**   **\_\_**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验项目名称** | NoSQL和关系数据库的操作比较 | | | **实验成绩** |  |
| **学生姓名** |  | **学 号** |  | **年级专业班级** |  |
| **小组成员** |  | | | **实验日期** |  |

## 实验四：NoSQL和关系数据库的操作比较

### 一、实验目的

* 理解四种数据库(MySQL、HBase、Redis和MongoDB)的概念以及不同点；
* 熟练使用四种数据库操作常用的Shell命令；
* 熟悉四种数据库操作常用的Java API。

### 二、实验平台

* 操作系统：Linux（建议Ubuntu18.04）；
* Hadoop版本：3.2.0；
* MySQL版本：自定义；
* HBase版本：2.2.1；
* Redis版本：自定义；
* MongoDB版本：自定义；
* JDK版本：openjdk-1.8；
* Java IDE：Eclipse；

### 三、实验步骤

**1.数据库简介及其安装教程**

|  |  |  |
| --- | --- | --- |
| 数据库名称 | 简介 | 安装教程 |
|  | MySQL是一个[关系型数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511)，由瑞典MySQL AB 公司开发，目前属于 [Oracle](https://baike.baidu.com/item/Oracle) 旗下产品。MySQL 是最流行的[关系型数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511)之一，在 WEB 应用方面，MySQL [RDBMS](https://baike.baidu.com/item/RDBMS/1048260) (Relational Database Management System，关系数据库管理系统) 应用软件之一。 | <https://www.runoob.com/?s=MYSQL%E5%AE%89%E8%A3%85> |
|  | Redis（全称：Remote Dictionary Server 远程字典服务）是一个开源的使用ANSI [C语言](https://baike.baidu.com/item/C%E8%AF%AD%E8%A8%80)编写、支持网络、可基于内存亦可持久化的日志型、Key-Value [非关系型数据库](https://baike.baidu.com/item/%E9%9D%9E%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93)，并提供多种语言的API。从2010年3月15日起，Redis的开发工作由VMware主持。从2013年5月开始，Redis的开发由Pivotal赞助。 | <https://www.runoob.com/redis/redis-install.html> |
|  | MongoDB是一个基于分布式文件存储 [1]  的数据库。由[C++](https://baike.baidu.com/item/C%2B%2B" \t "_blank)语言编写。旨在为WEB应用提供可扩展的高性能数据存储解决方案。  MongoDB是一个介于[关系数据库](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E6%95%B0%E6%8D%AE%E5%BA%93)和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。它支持的数据结构非常松散，是类似[json](https://baike.baidu.com/item/json)的[bson](https://baike.baidu.com/item/bson" \t "_blank)格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是它支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立[索引](https://baike.baidu.com/item/%E7%B4%A2%E5%BC%95" \t "_blank)。 | <https://www.runoob.com/mongodb/mongodb-window-install.html> |

**2.实验内容**

**（一） MySQL数据库操作**

学生表Student

|  |  |  |  |
| --- | --- | --- | --- |
| Name | English | Math | Computer |
| zhangsan | 69 | 86 | 77 |
| lisi | 55 | 100 | 88 |

1. 根据上面给出的Student表，在MySQL数据库中完成如下操作：

**准备工作：**

（1）用管理员身份cmd进入MySQL的/bin目录，输入命令开启MySQL服务：





（2）**登陆MySQL**

mysql>语法如下：*mysql -u用户名 -p用户密码*

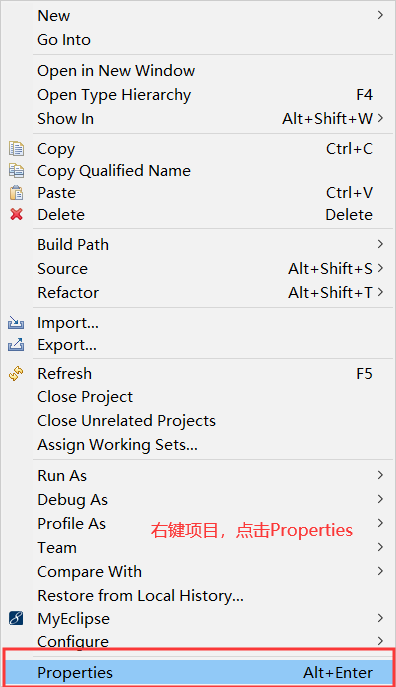
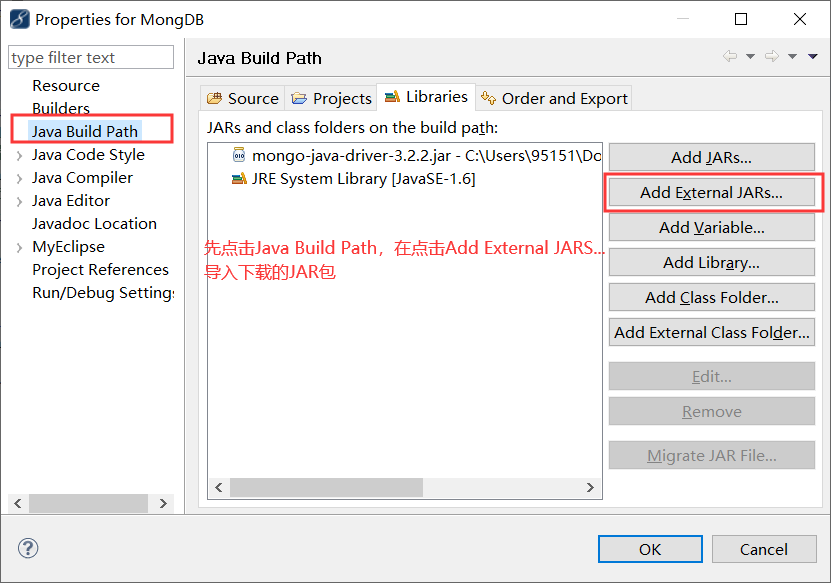


（3）**导入MySQL的java包进行编程：**

进入菜鸟教程下载需要的jar包：

<https://www.runoob.com/java/java-mysql-connect.html>

在MyEclipse导入jar包：

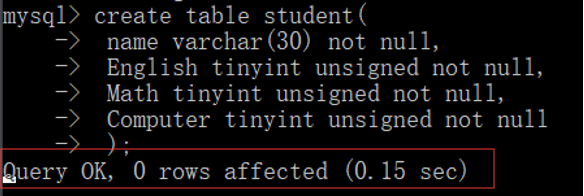
**开始MySQL实验：**

1. 在MySQL中创建Student表，并录入数据；

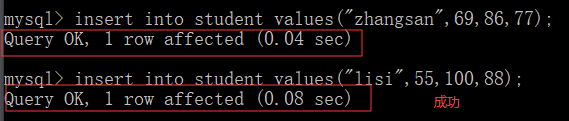
·要先创建数据库，然后使用你创建的数据库创建表：



·创建学生表：



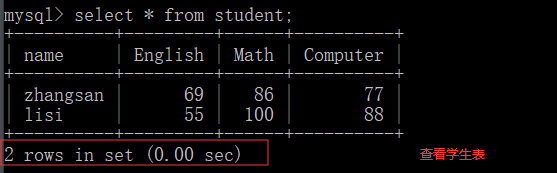
·向 Student 表中插入两条记录的 SQL 语句如下：



1. 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;

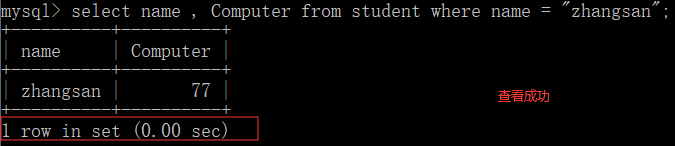
·上述 SQL 语句执行后的结果截图如下：



1. 查询zhangsan的Computer成绩；

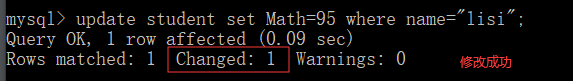
·查询 zhangsan 的 Computer 成绩的 SQL 语句如下：select name , Computer from student where name = "zhangsan";

·上述语句执行后的结果截图如下：



（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"; 上述 SQL 语句执行结果截图如下：



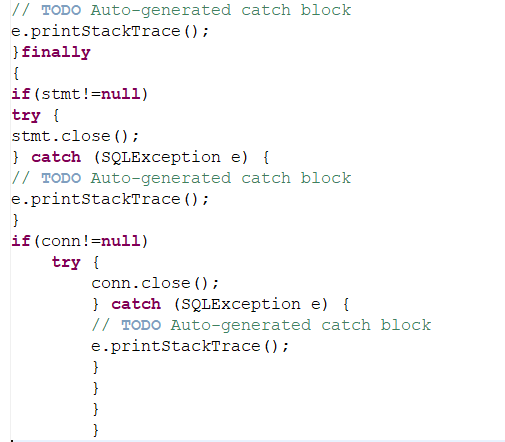
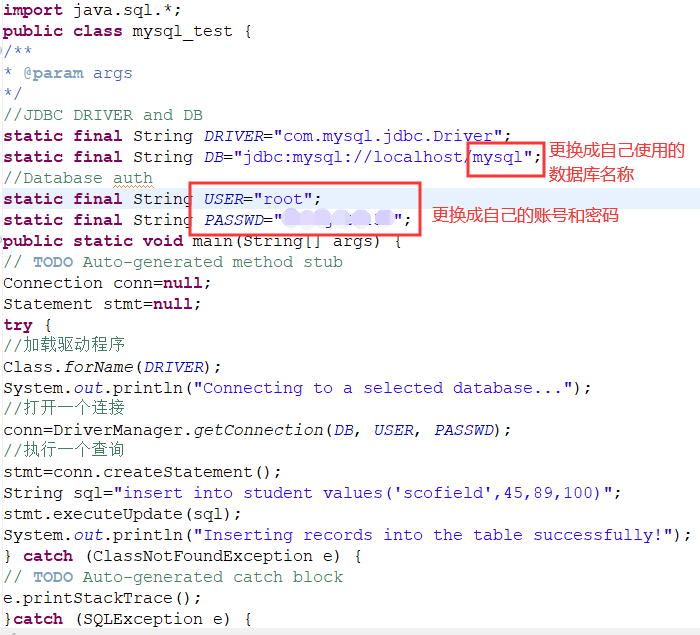
1. 根据上面已经设计出的Student表，使用MySQL的JAVA客户端编程实现以下操作：

（1）向Student表中添加如下所示的一条记录：

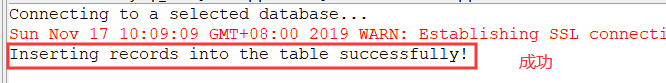
|  |  |  |  |
| --- | --- | --- | --- |
| scofield | 45 | 89 | 100 |

**·代码：**

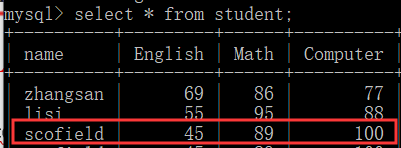
**要注意把代码中的账户，密码和数据库换成自己所使用的，执行成功后进入MySQL服务器查询是否成功。**



**·在MyEclipse上的执行结果：**



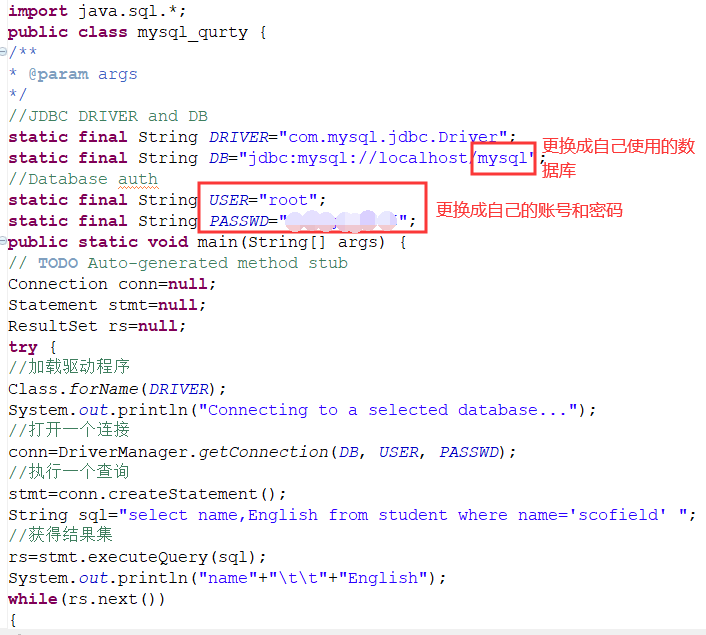
**·在MySQL客户端查询是否添加成功：**

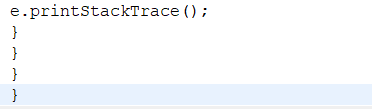
**（成功）**

（2）获取scofield的English成绩信息

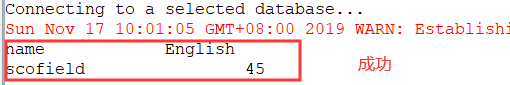
**·代码：**

**要注意把代码中的账户，密码和数据库换成自己所使用的。**





**·在MyEclipse上的执行结果：**



**（二）HBase数据库操作**

学生表Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| name | score | | | |
| English | Math | Computer | |
| zhangsan | 69 | 86 | | 77 |
| lisi | 55 | 100 | | 88 |

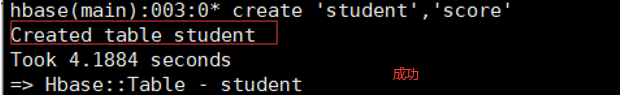
**准备工作：**

1.开启HBase 2.进入HBase的Shell模式

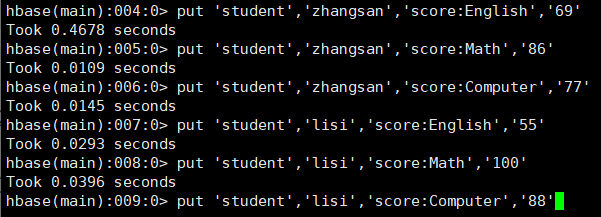
1.根据上面给出的学生表Student的信息，执行如下操作：

1. 用Hbase Shell命令创建学生表Student；

·创建 Student 表的命令如下：

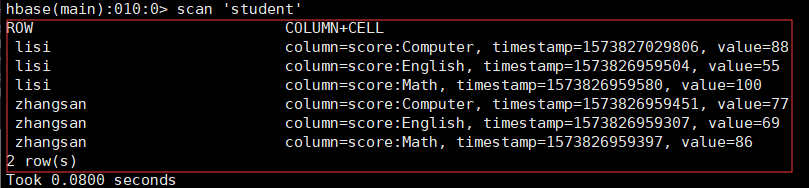


·向 Student 表中插入上面表格数据的命令如下：



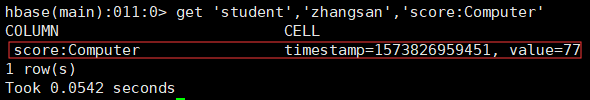
1. 用scan命令浏览Student表的相关信息；

·用 scan 指令浏览 Student 表相关信息的命令如下：



1. 查询zhangsan的Computer成绩；

·查询 zhangsan 的 Computer 成绩的命令如下：

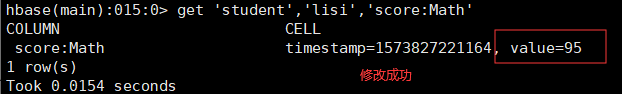


（4）修改lisi的Math成绩，改为95。

·修改 lisi 的 Math 成绩的命令如下：



·查看一下是否修改成功：



2.根据上面已经设计出的Student表，用HBase API编程实现以下操作：

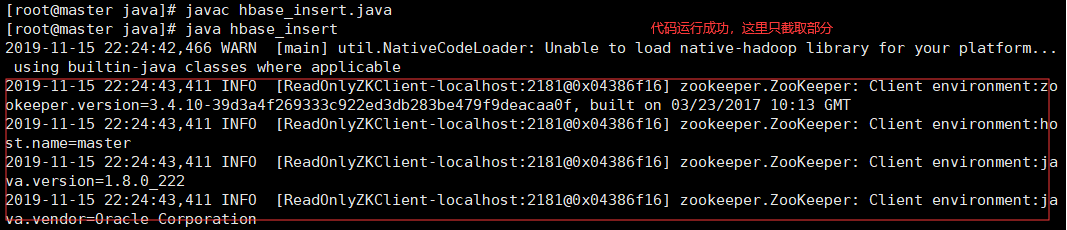
（1）添加数据：English:45 Math:89 Computer:100

|  |  |  |  |
| --- | --- | --- | --- |
| scofield | 45 | 89 | 100 |

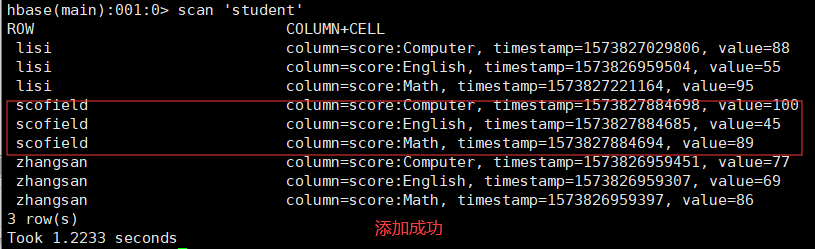
·实现添加数据的 Java 代码如下：

|  |  |
| --- | --- |
|  |  |

·编译代码：



·执行完上述代码以后，可以用 scan 命令输出数据库数据，以检验是否插入成功，执行结果截图如下：



1. 获取scofield的English成绩信息。

**·代码：**

|  |  |
| --- | --- |
|  |  |

**·执行结果：**



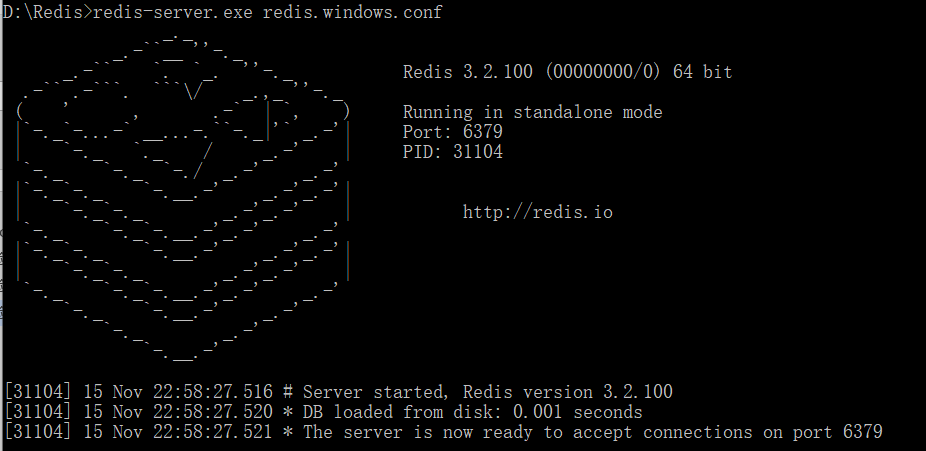
**（三）Redis数据库操作**

Student键值对如下：

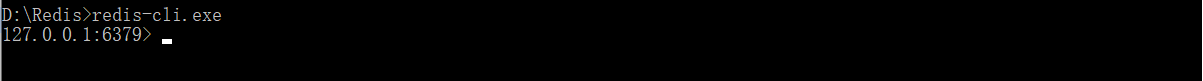
|  |
| --- |
| zhangsan:｛  English: 69  Math: 86  Computer: 77  ｝  lisi:｛  English: 55  Math: 100  Computer: 88  ｝ |

**准备工作：**

**（1）输入redis-server.exe redis.windows.conf启动redis以及配置文件**



**（2）再打开一个cmd窗口作为客户端，输入命令redis-cli.exe运行redis-cli程序**



**（3）导入Redis的jar包：**

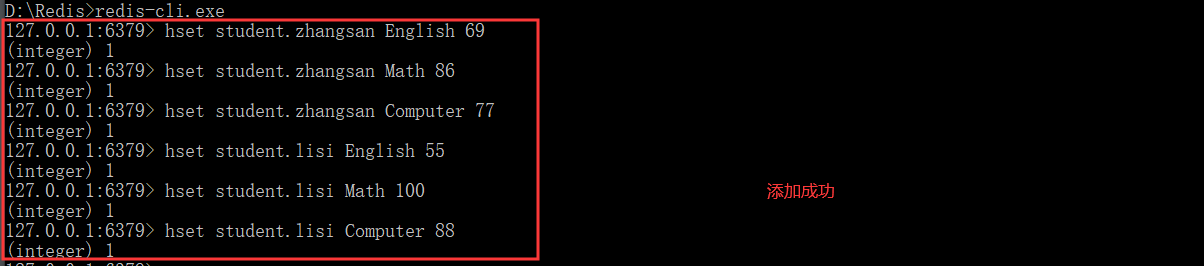
进入菜鸟教程下载需要的jar包，导入JAR包和MySQL一样。

**开始实验！！！**

1. 根据上面给出的键值对，完成如下操作：

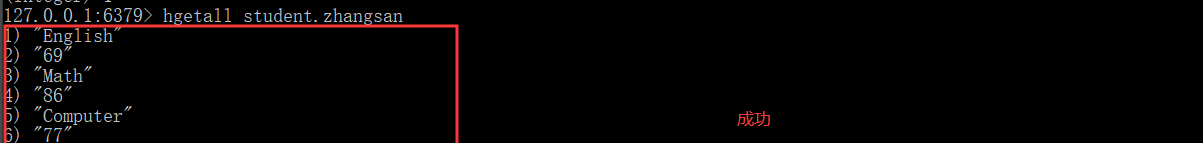
（1）用Redis的哈希结构设计出学生表Student（键值可以用student.zhangsan和student.lisi来表示两个键值属于同一个表）；

·插入上述键值对的命令如下：

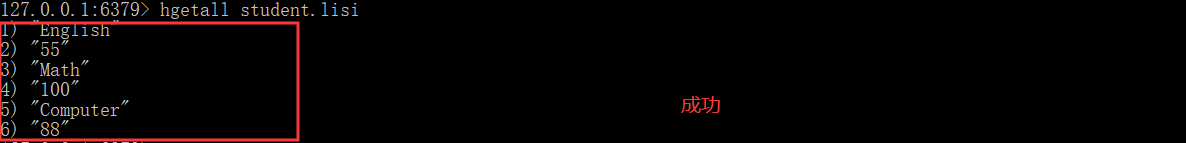


（2）用hgetall命令分别输出zhangsan和lisi的成绩信息；

·查询 zhangsan 成绩信息的命令如下：



·查询 lisi 成绩信息的命令如下：



（3）用hget命令查询zhangsan的Computer成绩；

·查询 zhangsan 的 Computer 成绩的命令如下：



（4）修改lisi的Math成绩，改为95。

·修改 lisi 的 Math 成绩的命令如下：



· 查看是否修改成功：

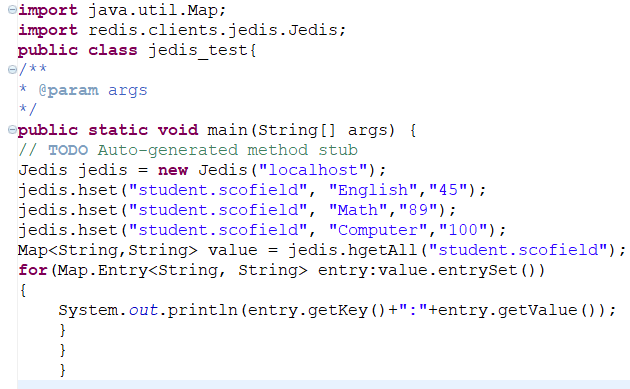


2.根据上面已经设计出的学生表Student，用Redis的JAVA客户端编程(jedis)，实现如下操作：

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

该数据对应的键值对形式如下：

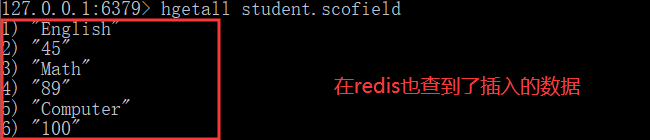
|  |
| --- |
| scofield:｛  English: 45  Math: 89  Computer: 100  ｝ |

·完成添加数据操作的 Java 代码如下：

·在 MyEclipse 中执行上述代码后，在 MyEclipse 控制台输出的信息截图如下：

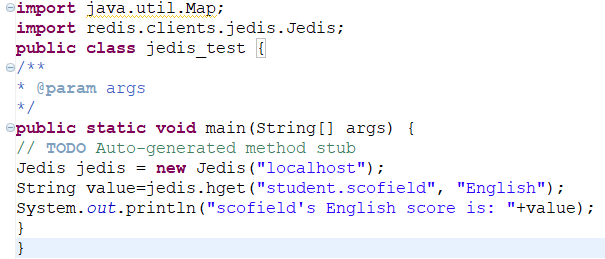


·在redis查询是否插入成功：



（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：



·在 Eclipse 中执行上述代码后，在 Eclipse 控制台输出的信息截图如下：



**（四）MongoDB数据库操作**

Student文档如下:

|  |
| --- |
| {  “name”: “zhangsan”,  “score”: {  “English”: 69,  “Math”: 86,  “Computer”: 77  }  }  {  “name”: “lisi”,  “score”: {  “English”: 55,  “Math”: 100,  “Computer”: 88  }  } |

**准备工作**

**1.** **启动 MongoDB 服务**

进入MongoDB的安装目录，输入mongodb://localhost

**2.进入MongoDB**

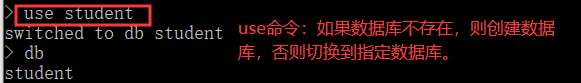
**输入mongo**

**开始实验！！！**

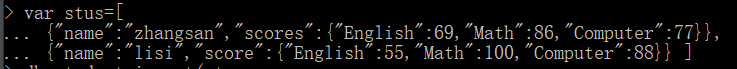
1.根据上面给出的文档，完成如下操作：

（1）用MongoDB Shell设计出student集合；

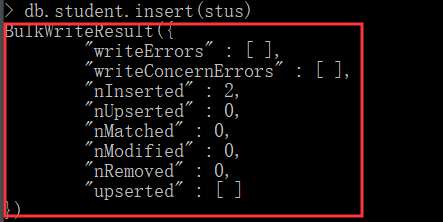
·首先，切换到 student 集合，命令如下：



·其次，定义包含上述两个文档的数组，命令如下：

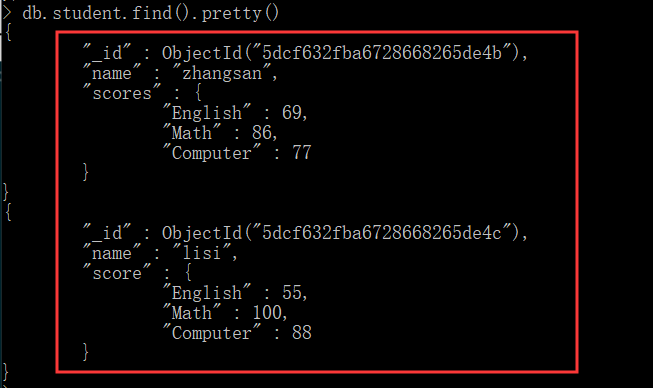


·最后，调用如下命令插入数据库：



（2）用find()方法输出两个学生的信息

·用 find()方法输出两个学生信息的命令如下：

；

（3）用find()方法查询zhangsan的所有成绩(只显示score列)；

·用 find 函数查询 zhangsan 的所有成绩的命令如下：



（4）修改lisi的Math成绩，改为95。

·修改 lisi 的 Math 成绩的命令如下：



· 查询是否修改成功：

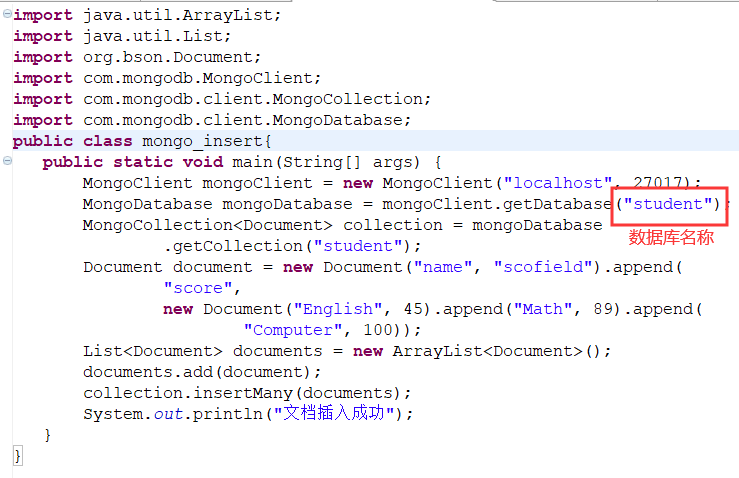


2.根据上面已经设计出的Student集合，用MongoDB的Java客户端编程，实现如下操作：

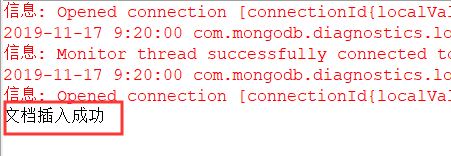
（1）添加数据：English:45 Math:89 Computer:100

与上述数据对应的文档形式如下：

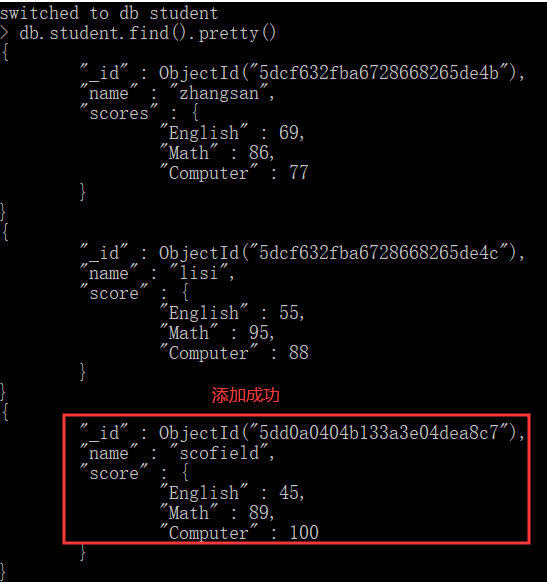
|  |
| --- |
| {  “name”: “scofield”,  “score”: {  “English”: 45,  “Math”: 89,  “Computer”: 100  }  } |

·代码： 

·执行结果：



·可以使用 find()方法验证数据是否已经成功插入到 MongoDB 数据库中，具体命令及其执行结果截图如下：

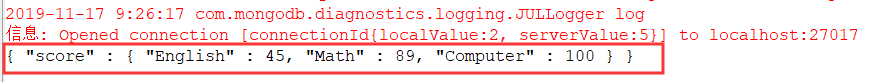


（2）获取scofield的所有成绩成绩信息(只显示score列)

·代码：



·执行结果：



**总结**

一.本次实验的个人心得体会：

这次实验相对于前几次实验较为简单，只需要去菜鸟教程下载几个数据库进行简单的查找，添加，删除操作即可。而编程部分只需要在MyEclipse导入连接所需要的JAR包即可，代码参考林子雨老师的进行细微修改即可。做完这次实验后，确实对关系型数据库和非关系型数据库的区别有进一步的理解，非关系型数据库严格上不是一种数据库，应该是一种数据结构化存储方法的集合，可以是文档或者键值对等。

这次实验还是比较愉快的！！！

二.四种关系型数据库和非关系型数据库的个人总结

1.MySQL：

·MySQL各语句的执行顺序

from... where..  select ... .group by... having.... order by... limit

·on和where的选择条件的区别：  
on后面的是连接条件，代表两个表建立关系所遵循的规则

where后面的可以看作是筛选条件，是对最终结果集进行过滤所遵循的规则

·SQL语句分类：DDL**语句，DML（Data Manipulation Language）语句：数据操纵语句，DCL（Data Control Language）语句：数据控制语句。**

**2.HBase：**

·HBase个人理解  
a.HBASE是一个分布式-数据库，可以提供数据的实时随机读写。  
b.HBASE是一个no-sql的数据库，需要用java来控制

**3.Redis：**

1.以 值-键，形式存储数据，旨在减轻Mysql 等真实业务系统的数据库的访问压力；

2.redis存储的数据，存储在内存中，需注意内存分配，默认的内存不够，会报错

3.redis可以配置集群，可以配置主从，可以配置占用内存大小

**4.MongDB：**

**·优势：更高的写负载，处理很大的规模的单表，不可靠环境保证高可用性，使用基于位置的数据查询，查的更快。**

**·劣势：mongodb不支持事务操作，mongodb占用空间过大。**

三.[关系型数据库和非关系型数据库 对比](https://www.cnblogs.com/nucdy/p/7872012.html)

1.关系型数据库和非关系型数据库在使用场景上差别比较大，所以并不存在孰强孰弱，只有结合自身的业务特点才能发挥出这两类数据库的优势，下面说说这两类数据库的一些特点：  
首先一般非关系型数据库是基于CAP模型，而传统的关系型数据库是基于ACID模型的：

1. 数据存储结构：

首先关系型数据库一般都有固定的表结构，并且需要通过DDL语句来修改表结构，不是很容易进行扩展，而非关系型数据库的存储机制就有很多了，比如基于文档的，K-V键值对的，还有基于图的等，对于数据的格式十分灵活没有固定的表结构，方便扩展，因此如果业务的数据结构并不是固定的或者经常变动比较大的，那么非关系型数据库是个好的选择

2. 可扩展性

传统的关系型数据库给人一种横向扩展难，不好对数据进行分片等，而一些非关系型数据库则原生就支持数据的水平扩展(比如mongodb的sharding机制)，并且这可能也是很多NoSQL的一大卖点，其实象Mysql这种关系型数据库的水平扩展也并不是难，即使NoSQL水平扩展容易但对于向跨分片进行joins这种场景都没有什么太好的解决办法，不管是关系型还是非关系型数据库，解决水平扩展或者跨分片Joins这种场景，在应用层和数据库层中间加一层中间件来做数据处理也许是个好的办法

3. 数据一致性

非关系型数据库一般强调的是数据最终一致性，而不没有像ACID一样强调数据的强一致性，从非关系型数据库中读到的有可能还是处于一个中间态的数据，因此如果你的业务对于数据的一致性要求很高，那么非关系型数据库并不一个很好的选择，非关系型数据库可能更多的偏向于OLAP场景，而关系型数据库更多偏向于OLTP场景

-------------------------------------------------------------------------------

本文参考自CSDN博主，原文链接: <https://blog.csdn.net/qq_33472765/article/details/81515251>

二.