哈尔滨工业大学

**计算学部**

**2021年春季学期**

**《软件架构与中间件》课程**

**实验报告**

**Lab 3-4：数据层及表示层软件架构实验**

|  |  |  |
| --- | --- | --- |
| **姓名** | **学号** | **联系方式** |
| 许健 | 118371013 | 941197279@qq.com |
| 郭茁宁 | 1183710109 | v-zguo@microsoft.com |

目 录

[1 实验目的 1](#_Toc39004699)

[2 实验要求 1](#_Toc39004700)

[3 实验内容与过程 1](#_Toc39004701)

[3.1 Mycat数据库分库分表实验 1](#_Toc39004702)

[3.2 Sharding-JDBC数据库分库分表实验 8](#_Toc39004703)

[3.3 Redis数据缓存实验 13](#_Toc39004704)

[4 结对开发过程记录 18](#_Toc39004705)

[5 实验总结 19](#_Toc39004706)

# 实验目的

1）学习使用Mycat和Sharding-JDBC实现数据分库分表

2）学习使用Redis数据库实现数据缓存

3）能够灵活应用Mycat或Sharding-JDBC实现分库分表架构到实际系统

4）能够灵活应用Redis实现数据缓存架构到实际系统

5）学会数据表示层的设计及实现（Lab4）

# 实验要求

1）2人结对成组

2）实验3.1、3.2、3.3均为必做

3）结合《软件过程与工具》课程中进销存系统(或其他实际软件系统)进行数据层架构重构，实现根据业务垂直划分的数据库分库分表；面向海量数据带来的数据检索慢问题，实现数据库水平分片，达到数据检索的性能提升；利用缓存架构实现数据读取的性能提升。

4）应给出关键过程的细节

# 实验内容与过程

根据提供的实验指导书完成实验3.1-3.3，并回答相关问题。

## Mycat数据库分库分表实验

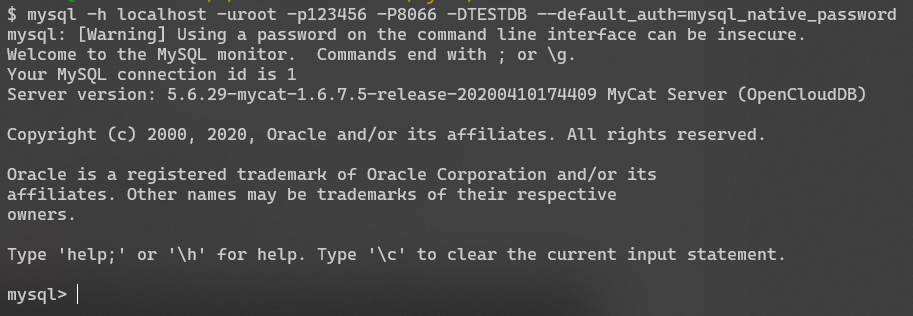
1）请给出Mycat配置安装过程中遇到的问题和解决方案。

问题：连接mycat时出现密码错误，如图。



原因：mysql8.0的默认密码验证不再是password，因此无法使用password来完成登陆验证。

解决方案：只需要在原命令后加入强制使用password验证的方法即可，添加参数 --default\_auth=mysql\_native\_password



2）请详析Mycat的分库分表原理和操作方法。

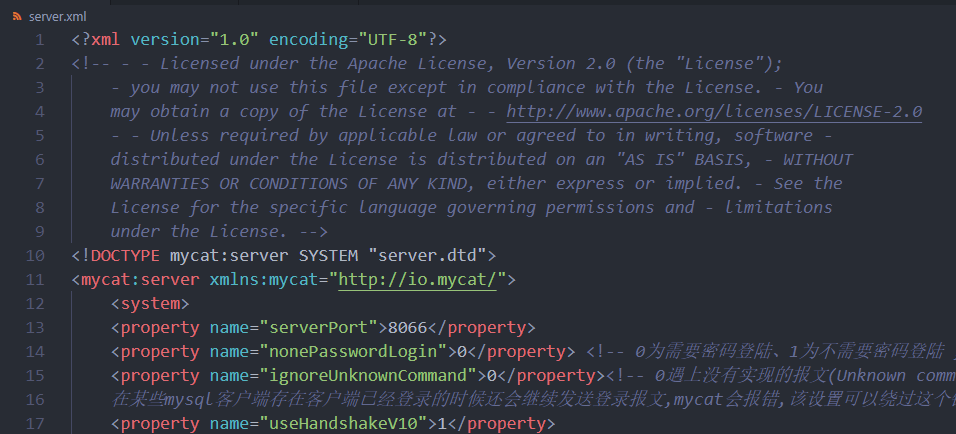
**分库分表原理**：

Mycat通过定义路由规则（其中定义分片字段和分片算法）来实现分库分表。Mycat拦截了用户发送过来的SQL语句，首先对SQL语句做一些特定的分析：如分片分析、路由分析、读写分离分析、缓存分析等，然后将此 SQL 发往后端的真实数据库，并将返回的结果做适当的处理，最终再返回给用户。当 Mycat收到一个SQL时，会先解析这个SQL，查找涉及到的表，然后看此表的定义，如果有分片规则， 则获取到SQL里分片字段的值，并匹配分片函数，得到该SQL对应的分片列表，然后将SQL发往这些分片去执行，最后收集和处理所有分片返回的结果数据，并输出到客户端。

**操作方法**：

首先下载安装jdk1.8，mysql8.0，mycat-1.6.7.5，下载mysql connect8.0的jar包，替换mycat/lib目录下原来的mysql connector

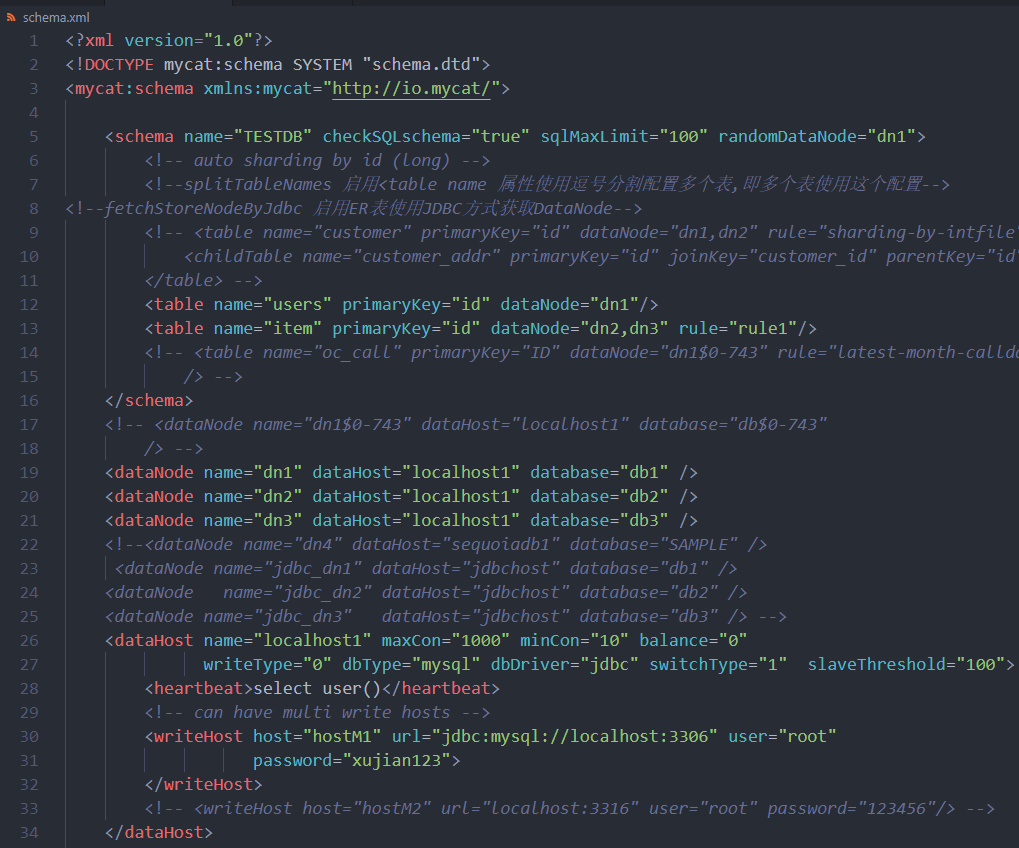
打开mycat/conf目录，编辑server.xml文件，配置服务端口为8066



配置Mycat的登录用户名和密码



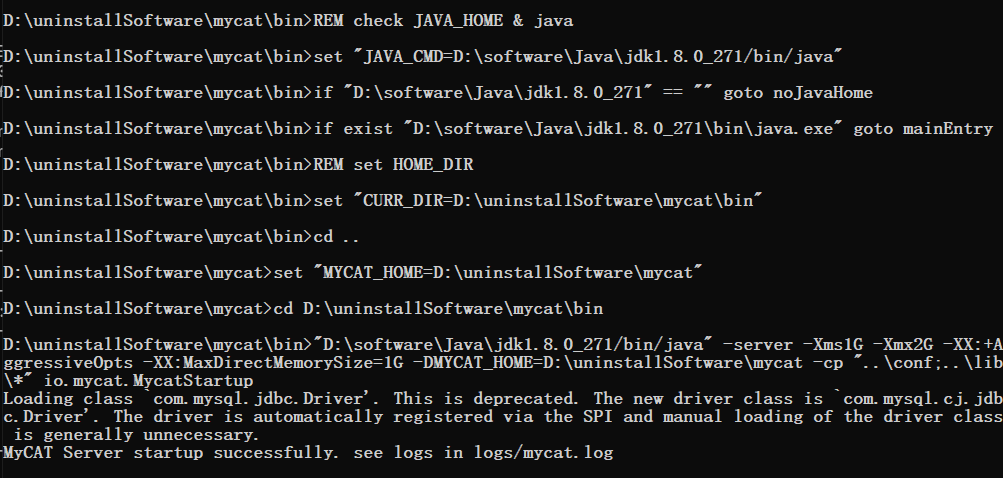
编辑schema.xml，配置使用的逻辑数据库的名字为TESTDB，配置数据库表为users和item两个表，配置它们的主键和物理数据库位置，以及切分表的规则，还要配置分片映射到物理数据库的地址。



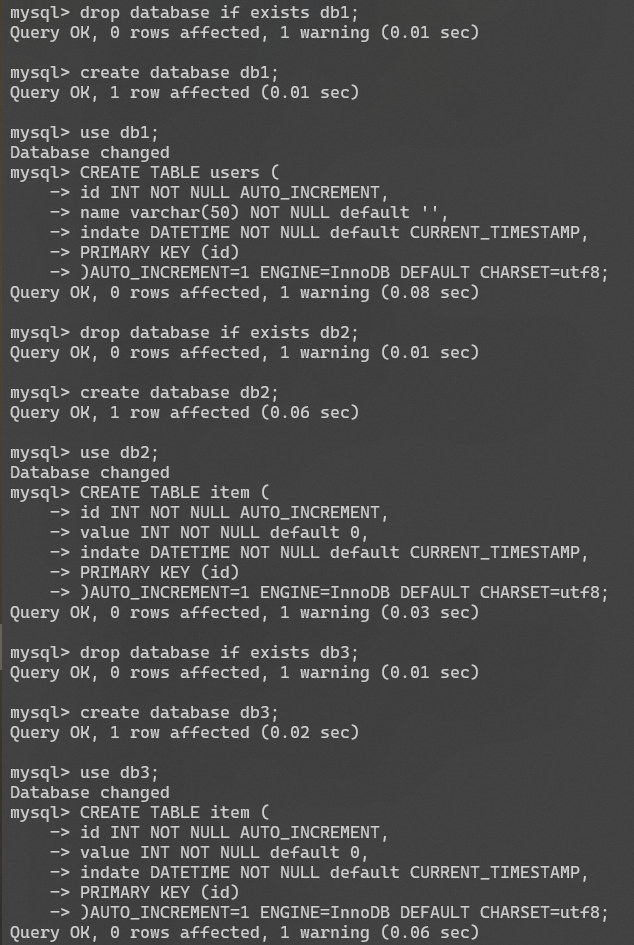
编辑rule.xml，配置切分数据库的规则

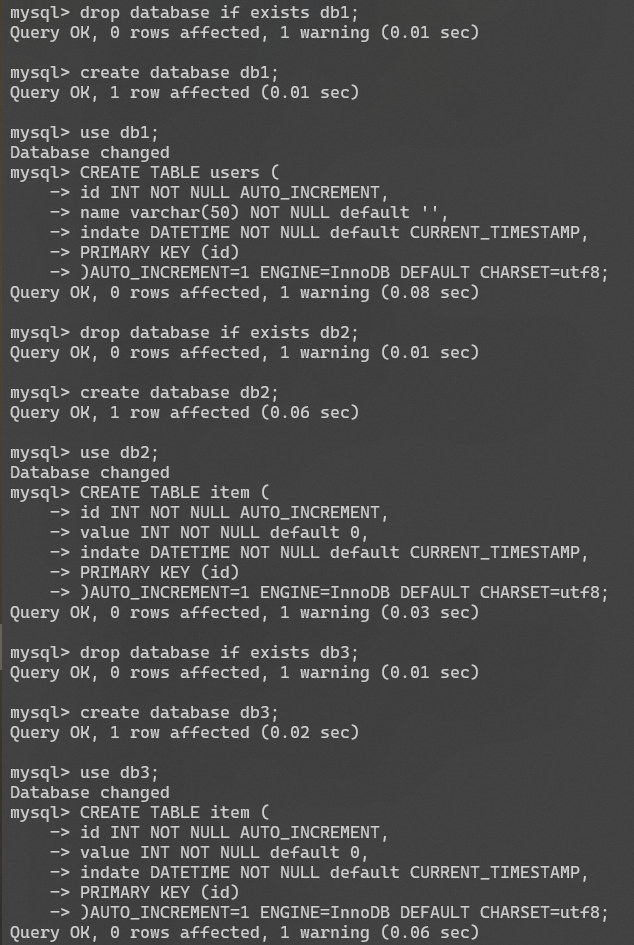


至此，mycat的配置完成，在windows下直接双击mycat/bin/startup\_nowrap.bat启动mycat服务，如下图即启动成功

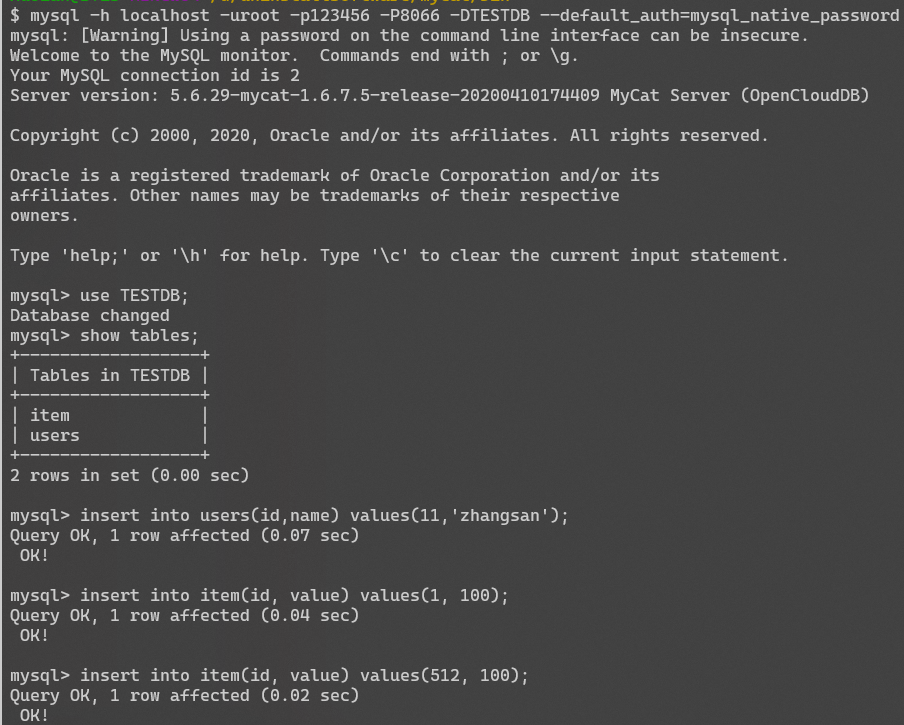


验证：在物理数据库上创建测试用表

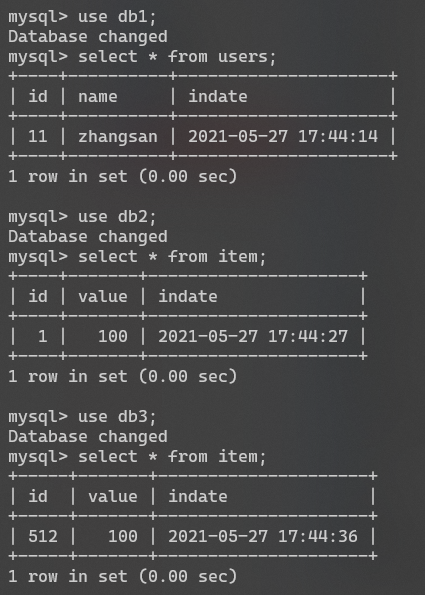




连接mycat，在数据库中插入三个元组



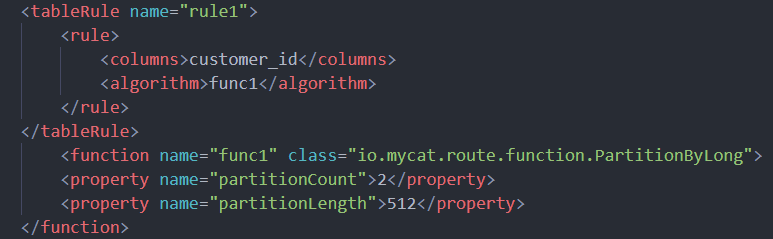
登录物理数据库查看结果

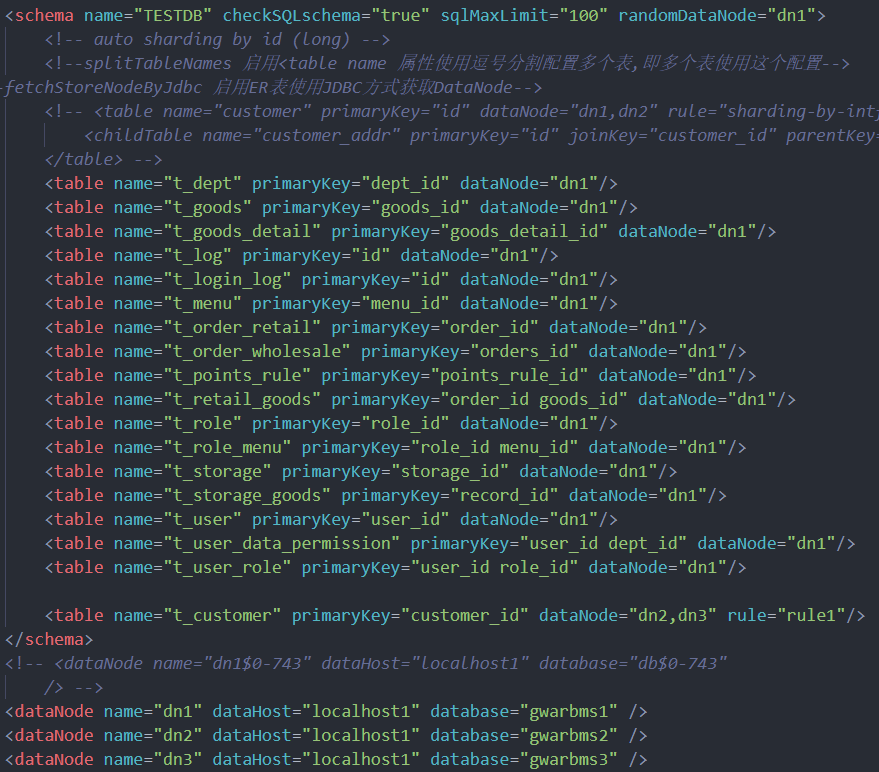


3）请在进销存系统(或其他实际软件系统)创建具有复杂表结构和含有较大数据量的数据库表， 并基于此库表描述分库分表的结果，且验证分库分表的效果。

对原系统的t\_customer表进行分库分表，其他表不变，配置如下

分片规则，单表最大长度为512



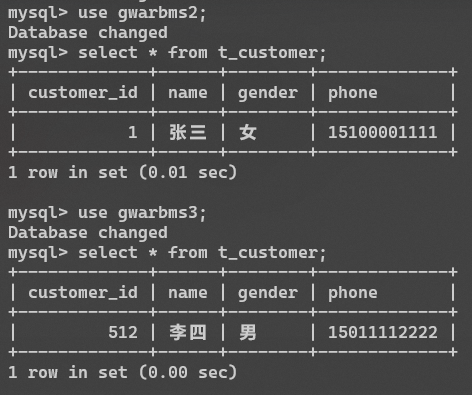


验证，登录mycat，插入两条数据

(1, 张三, 女, 15100001111) 和 (512, 李四, 男, 15011112222)



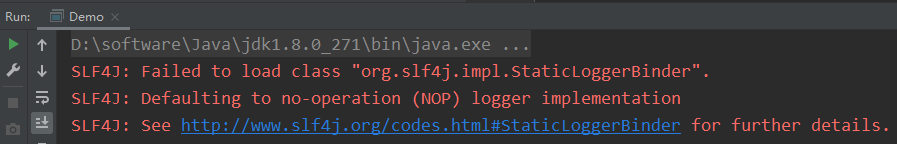
打开mysql查看插入结果，可以看到已经按照规则进行了分片



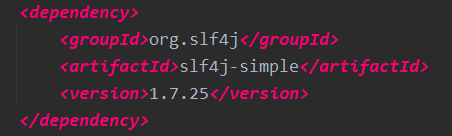
## Sharding-JDBC数据库分库分表实验

1）请给出Sharding-JDBC配置安装过程中遇到的问题和解决方案。

运行示例代码报错



经查资料，该报错是缺少相应的日志包导致的，并不会影响程序功能，可按照如下办法修改，在maven配置文件pom.xml中引入包



该问题解决

2）请详析Sharding-JDBC的分库分表原理和操作方法。

**分库分表原理**：当Sharding-JDBC接受到一条SQL语句时，会执行如下操作：

SQL解析--> SQL改写 --> SQL路由--> SQL执行 --> 结果归并，最终返回执行结果。

SQL解析过程分为词法解析和语法解析。词法解析器用于将SQL拆解为不可再分的原子符号。再使用语法解析器将SQL转换为抽象语法树。

SQL改写分为两部分，一部分是将分表的逻辑表名称替换为真实表名称。另一部

分是根据SQL解析结果替换一些在分片环境中不正确的功能。

SQL路由是根据分片规则配置，将SQL定位至真正的数据源。主要分为单表路

由、Binding表路由和笛卡尔积路由。

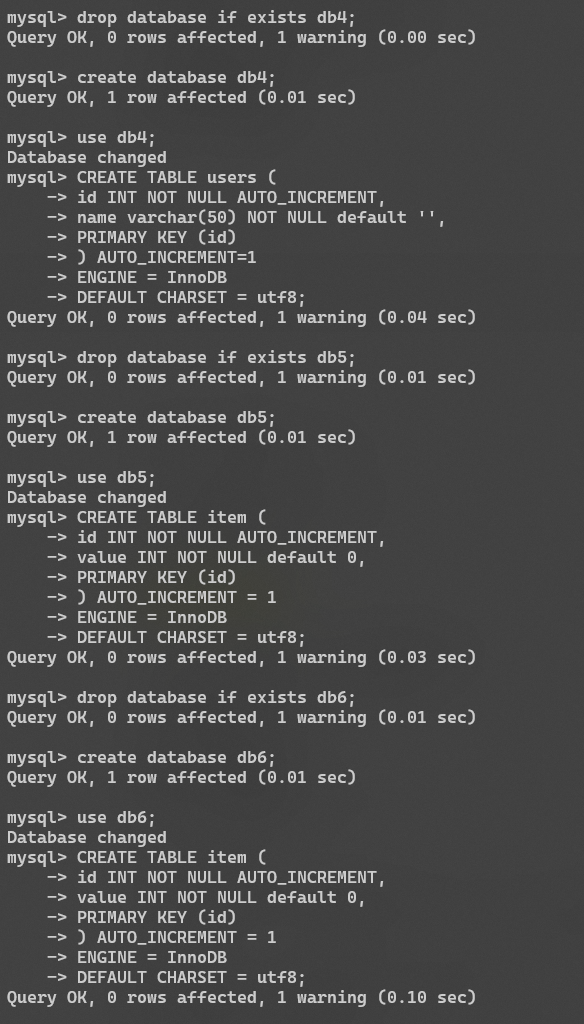
路由至真实数据源后，Sharding-JDBC将采用多线程并发执行SQL，并完成对

addBatch等批量方法的处理。

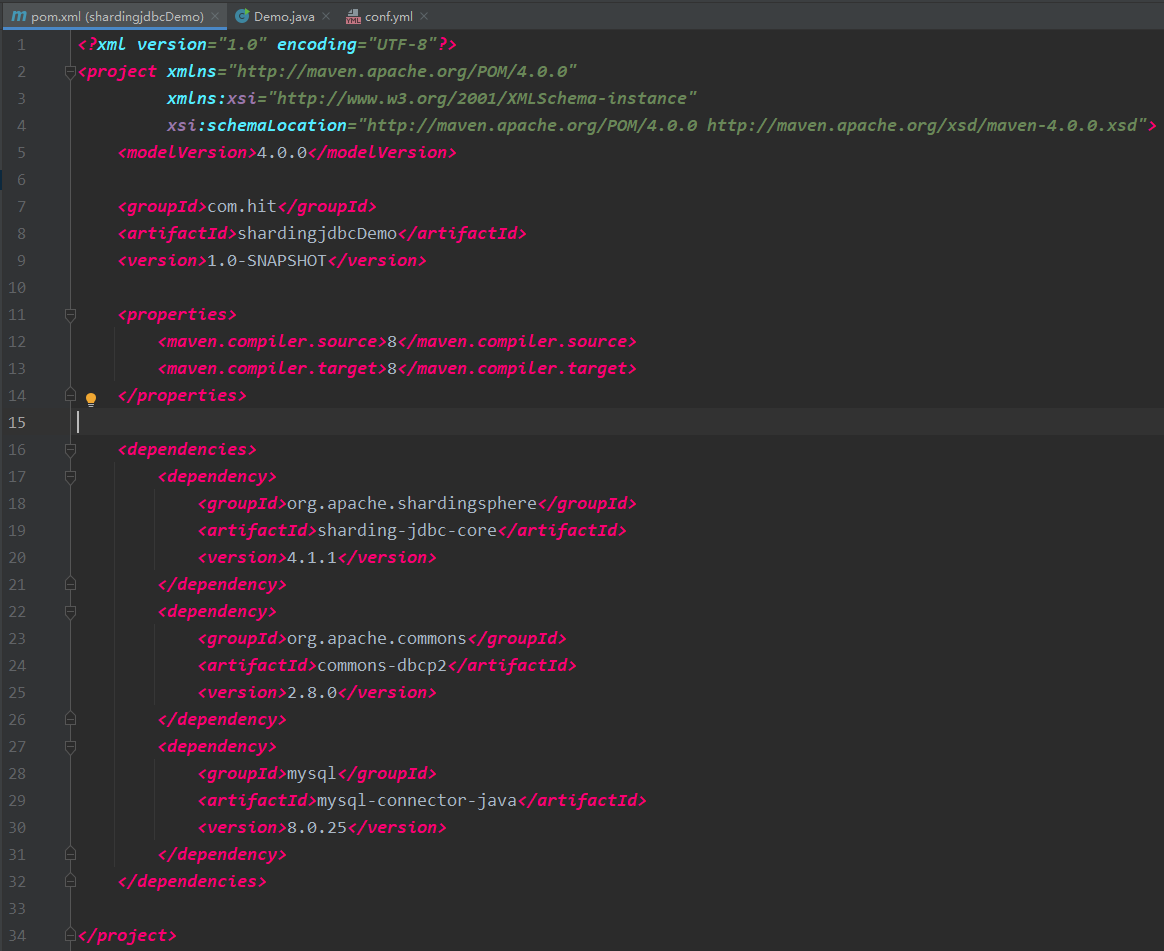
将从各个数据节点获取的多数据结果集，组合成为一个结果集并正确的返回至请求客户端。结果归并包括4类：普通遍历类、排序类、聚合类和分组类。每种类型都会先根据分页结果跳过不需要的数据。

**操作方法**：

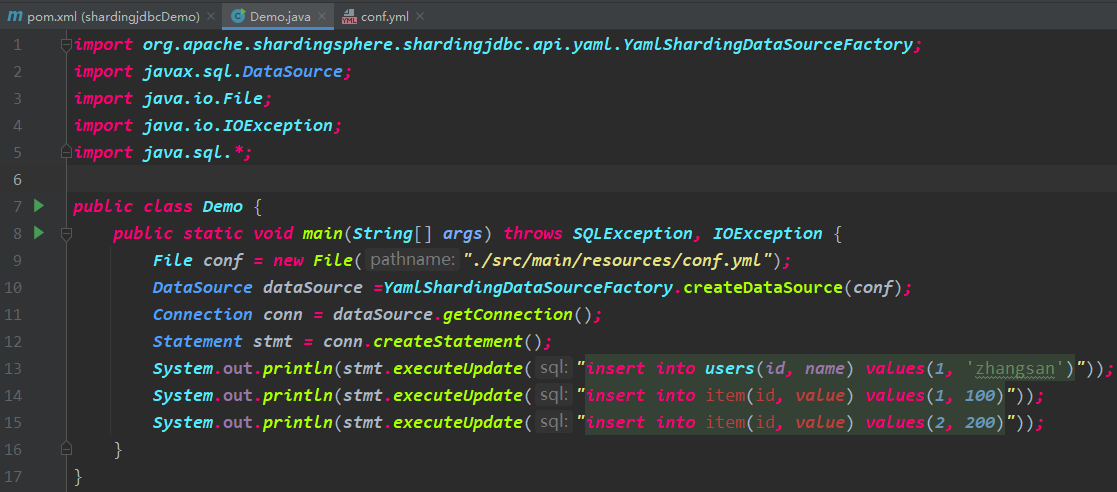
先登录物理数据库MySQL，创建数据库与表



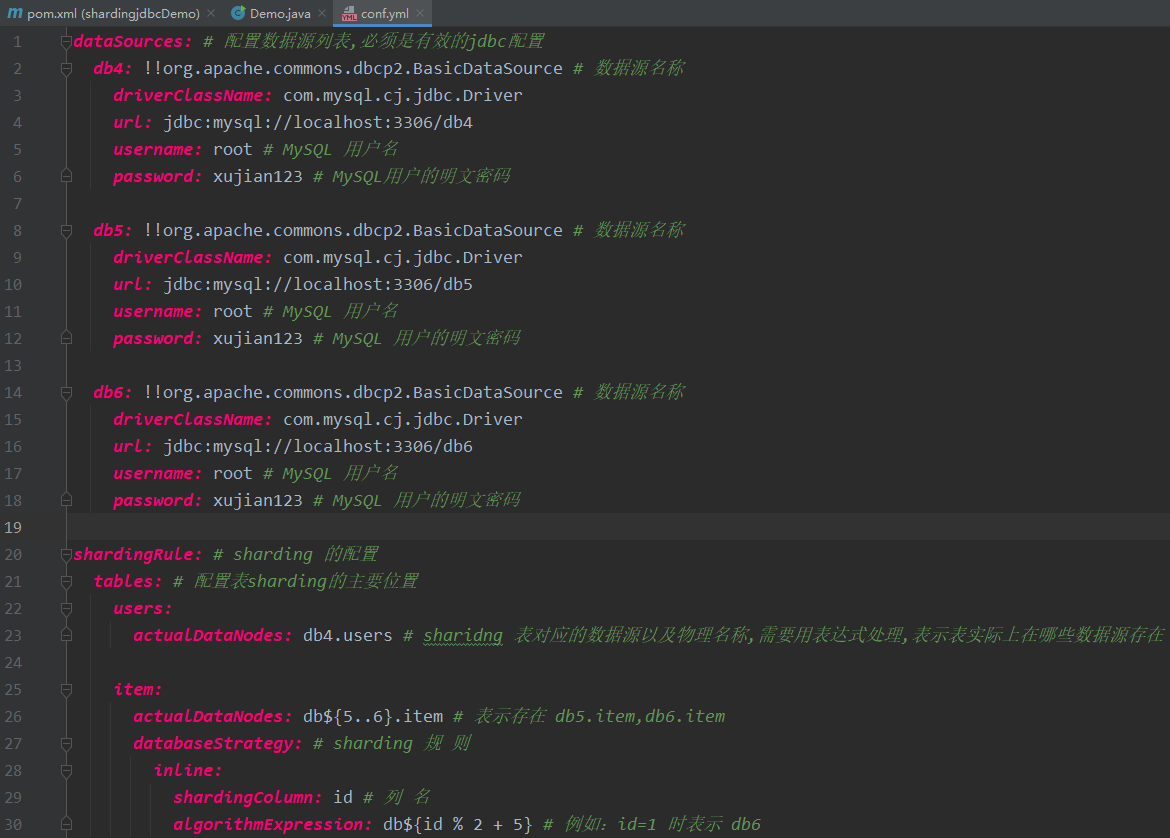
创建maven项目，修改配置文件



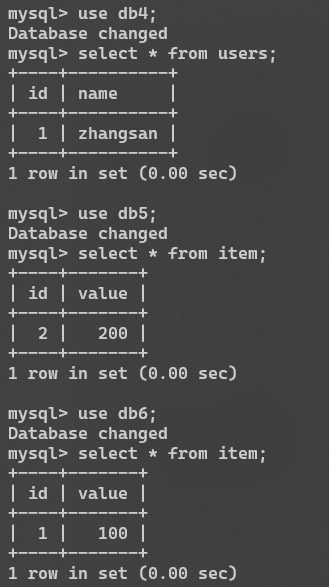
编写主程序



编写Sharding-JDBC的配置，包括分库分表规则等

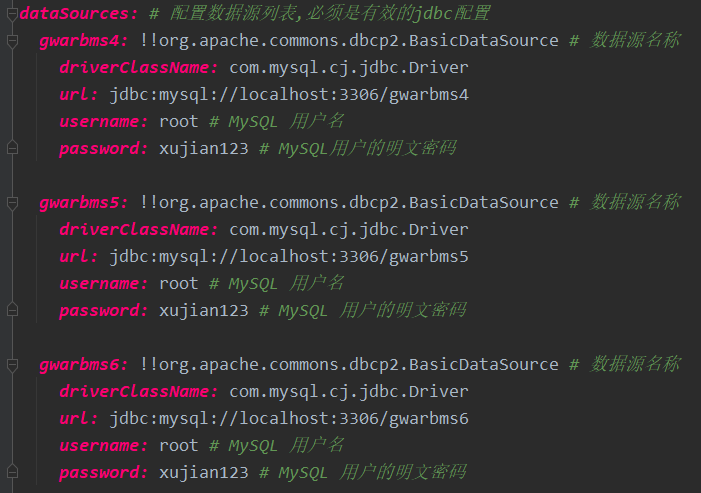


运行Demo.main() 后在mysql中查看结果



3）请在进销存系统(或其他实际软件系统)创建具有复杂表结构和含有较大数据量的数据库表， 并基于此库表描述分库分表的结果，且验证分库分表的效果。

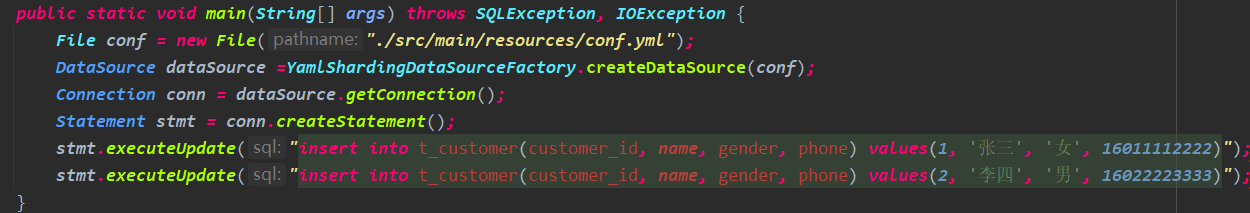
数据源配置



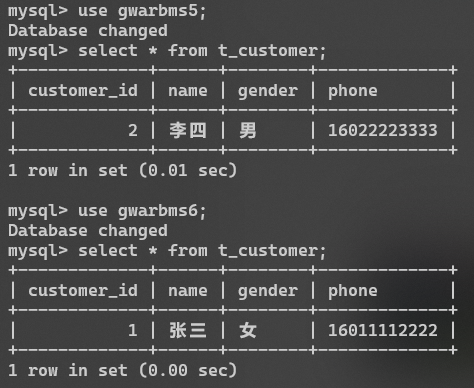
配置shrding规则，将customer表分库分表，按照id，偶数id的放入gwarbms5.t\_customer中，奇数id的放入gwarbms6.t\_customer中



编写测试主程序



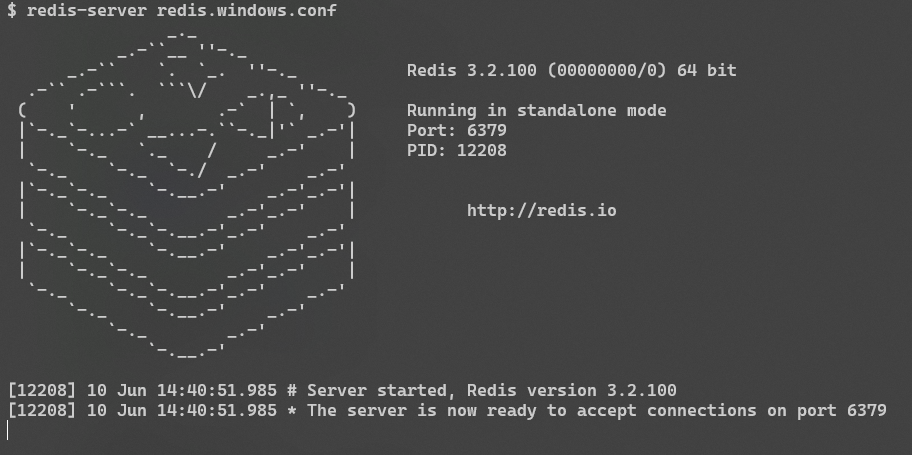
运行后在数据库里查看结果，与设计规则相符



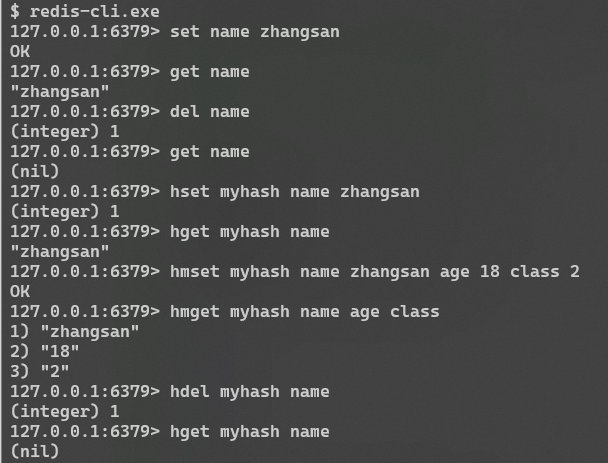
## Redis数据缓存实验

1）请给出Redis配置安装过程中遇到的问题和解决方案。

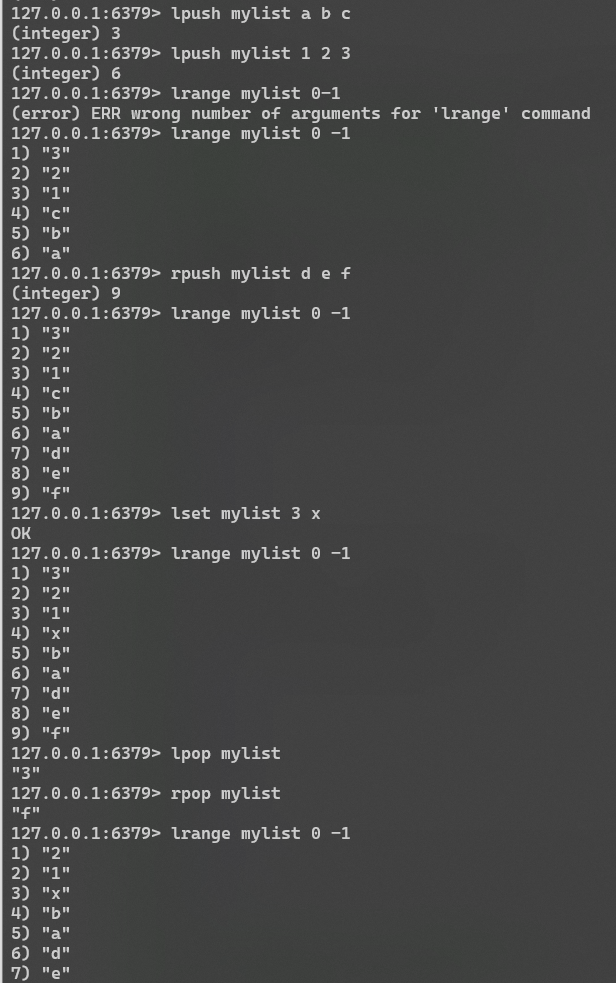
启动reids



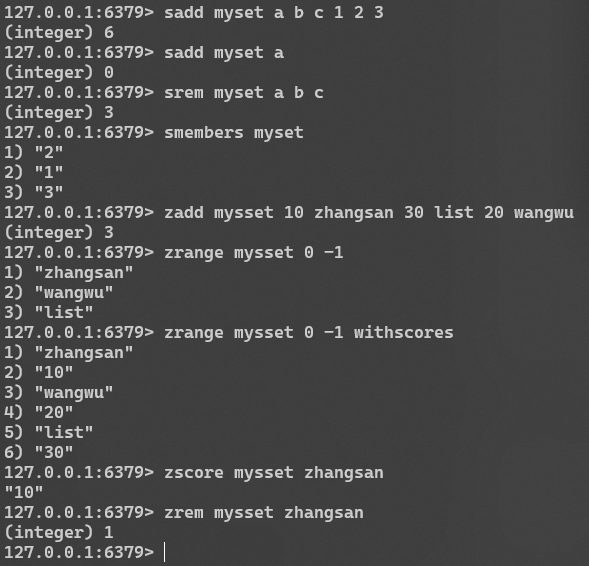
使用String和hash数据类型



使用List数据类型



使用Set和sorted set数据类型

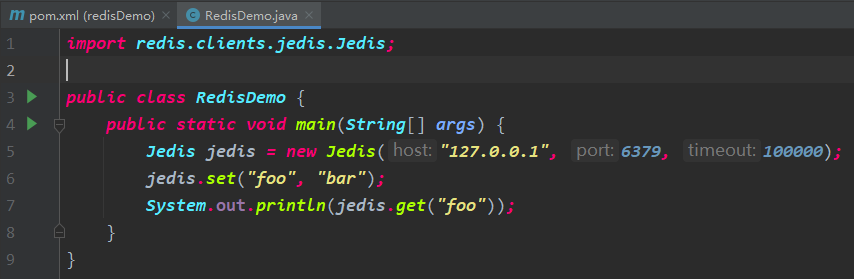


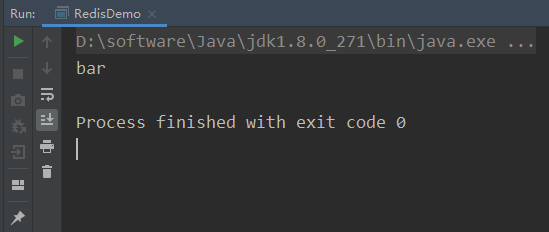
Jedis的使用练习

建立maven项目，修改配置文件pom.xml，添加jedis依赖如下



编写测试程序，并运行，结果如下





2）请详析Redis的缓存清洗策略，数据迁移及扩容策略，面向缓存雪崩、穿透等问题的策略。

**缓存清洗策略**：Redis默认情况下就是使用LRU策略的，当内存中写入的数据很满之后，就会使用LRU算法清理掉部分内存中的数据，将最近一段时间内，最少使用的一些数据给移除掉，然后让新的数据写入Redis缓存中。

**数据迁移及扩容策略**：有平滑迁移、一致性哈希、停机迁移等策略。

* 平滑迁移：在原集群和新集群上进行双写操作，从原集群迁移历史数据到新集群，然后将读操作从原集群切换到新集群上读，最后下线双写，完成迁移操作。
* 一致性哈希：一致性哈希将整个哈希值空间组织成一个虚拟的圆环（哈希环），整个空间按顺时针方向组织。将各个服务器使用Hash进行一个哈希，这样每台机器就能确定其在哈希环上的位置。将数据key使用相同的函数Hash计算出哈希值，并确定此数据在环上的位置，从此位置沿环顺时针“行走”，第一台遇到的服务器就是其应该定位到的服务器。
* 停机迁移：停机应用，先将应用停止服务；迁移历史数据，按照新的规则把历史数据迁移到新的缓存数据集群中；更改应用的数据源配置，指向新的缓存集群；重新启动应用。

**缓存雪崩策略**：

* 更新锁机制：对缓存更新操作进行加锁保护，保证只有一个线程能进行缓存更新。
* 失效时间分片机制：对不同的数据使用不同的失效时间，甚至对相同的数据、不同的请求使用不同的失效时间。
* 后台更新机制：由后台线程来更新缓存，并不是业务线程来更新缓存。

**缓存穿透策略**：通常将空值缓存起来，再次接收到同样的查询请求时，若命中缓存并值为空，就会直接返回，不会透传到数据库，避免缓存穿透。对恶意的查询攻击，可以对查询条件设置规则，不符合条件产生规则的直接拒绝

1. 请在进销存系统(或其他实际软件系统)设计一个简单场景，实现缓存读写操作，缓存更新操作，给出缓存的效果，分析2问题中相关策略的效果。

## 利用React进行表示层的设计及实现

1）利用React开发2-3个进销存系统(或其他实际软件系统)的交互界面。

# 结对开发过程记录

**（1）角色切换与任务分工**

表1-1结对开发角色与任务分工

| 日期 | 时间(HH:MM - HH:MM) | 驾驶员角色 | 领航员角色 | 本段时间的任务 |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**（2）工作日志**

由领航员负责记录，记录结对开发期间的遇到的问题、两人如何通过交流合作解决每个问题的。

表1-2 结对开发工作日志

| 日期/时间 | 问题描述 | 最终解决方法 | 交流过程 |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**（3）结对开发工作现场照片或线上讨论截图**

至少2张。

结对开发现场照片1 结对开发现场照片2

# 实验总结

【本实验的收获与不足，对数据层软件架构中各种问题的理解和认识】