**广 州 商 学 院**

**实验报告（第 4 次）**

实验名称 基于mysql数据库分表实现

实验时间 2025.04.08

同组同学 唐璇 小组分工 全职

**一、实验目的**

1. 优化数据库性能：通过分表技术，将数据分散到不同的表中，减少单表数据量，从而提高查询性能和处理速度。

2. 提高系统扩展性：分表能够更好地支持数据量增长，避免单一数据库表带来的性能瓶颈，增强系统的扩展性。

3. 分散数据存储负载：通过分表将数据分散到多个物理表或数据库，减少单一表的IO压力，提高系统的整体吞吐量。

4. 降低数据库操作复杂度：通过合理的分表设计，可以降低对大数据表的操作复杂度，提升数据处理效率，避免全表扫描带来的性能问题。

5. 提高高并发处理能力：分表能够使多个数据库实例或表同时处理不同的数据请求，减少单表锁竞争，提高系统的并发处理能力。

**二、实验仪器设备或材料**

1. 计算机：运行Windows、Linux或macOS操作系统的计算机。

2. Go语言环境：已安装并配置好Go语言环境，包括GOPATH、GOROOT等环境变量的设置。

3. 文本编辑器或IDE：如VS Code、GoLand等，用于编写Go代码。

4. 网络连接：用于下载相关库和依赖。

5. Mysql数据库：用于创建数据表、实验所需要的资源数据。

**三、实验原理**

1. 水平分表原理：将大表按某一字段（如ID、时间等）进行拆分，数据被分布到多个物理表中，每个表存储不同范围的数据，从而降低单个表的数据量，提高查询效率。

2. 垂直分表原理：将一个表中的不同列拆分到不同的表中，通常按照数据的访问频率或功能模块进行分割，优化表的读写性能。

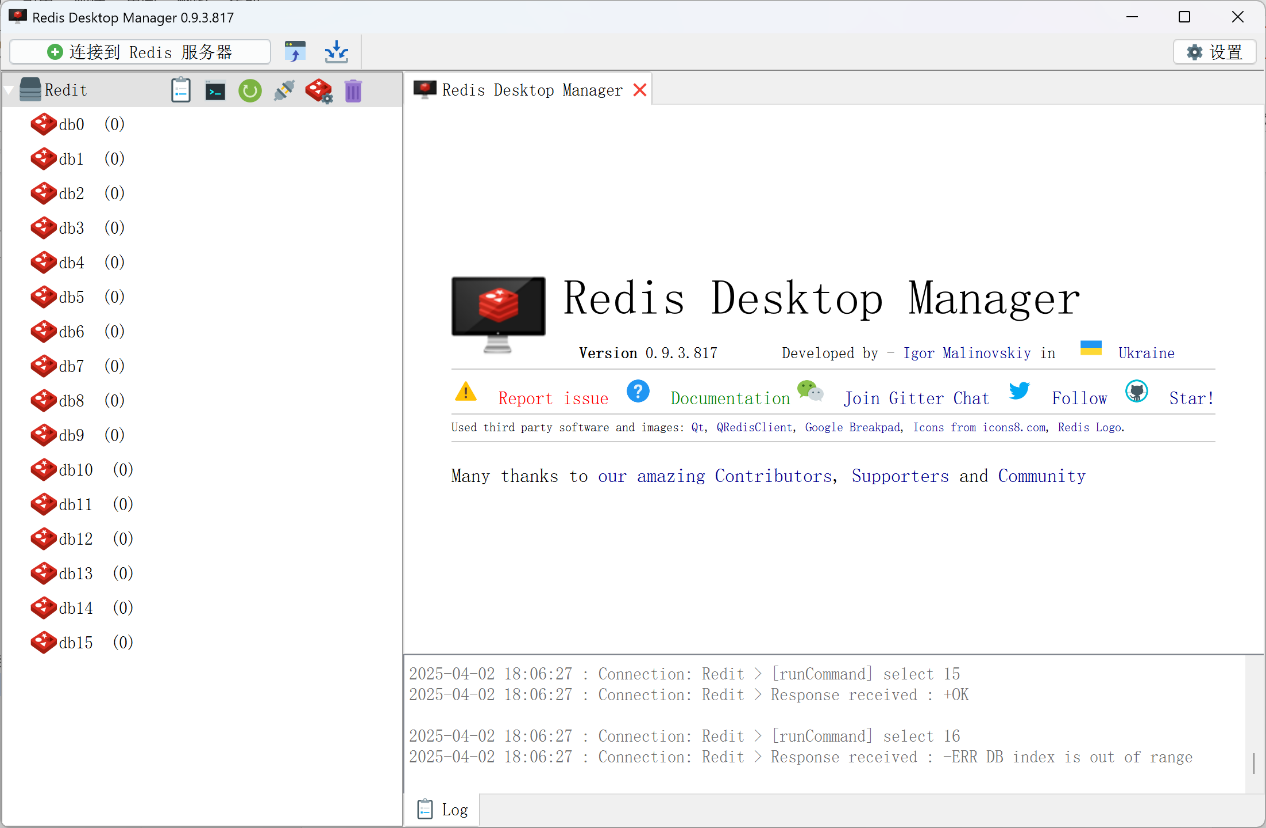
3.分表策略：可以根据不同的业务需求选择合适的分表策略，如范围分表、哈希分表、时间分表等。每种策略适用于不同的场景和数据特点。

4.分布式数据库架构：分表可以在不同的物理数据库实例上实现，通过数据分片将数据分布到多个节点，提高数据访问速度和可扩展性。

5.查询路由与合并：分表后，在进行查询时需要通过某些算法（如哈希算法、范围算法等）来确定数据存储在哪个表中，查询时需要对多个分表的结果进行合并处理。

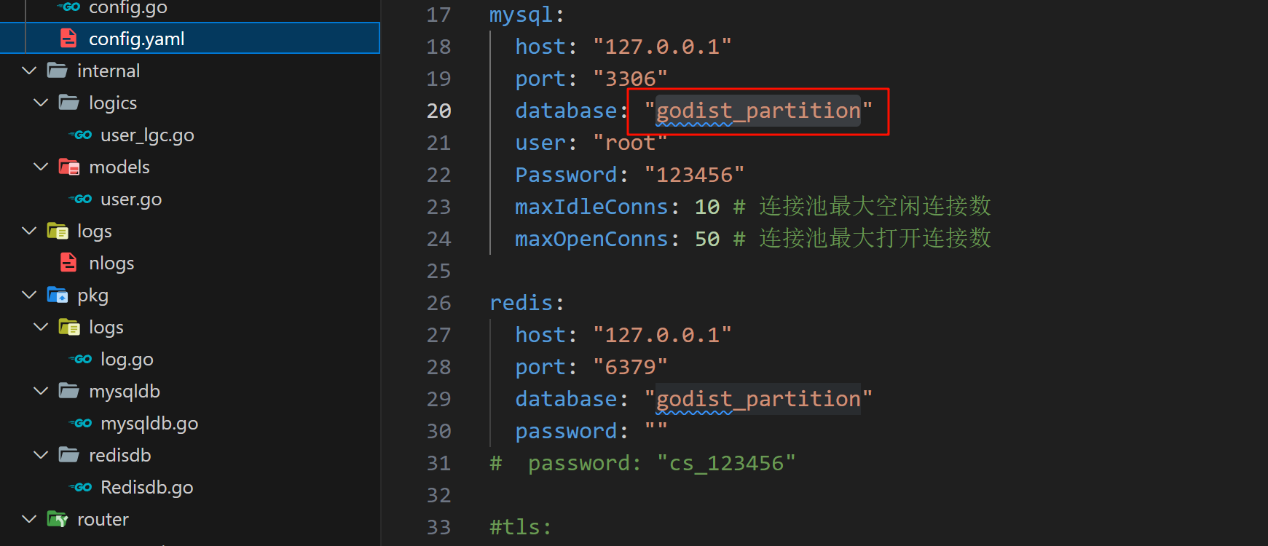
**四、实验内容与步骤**

1.确定Redis、MySQL等服务的正常启动

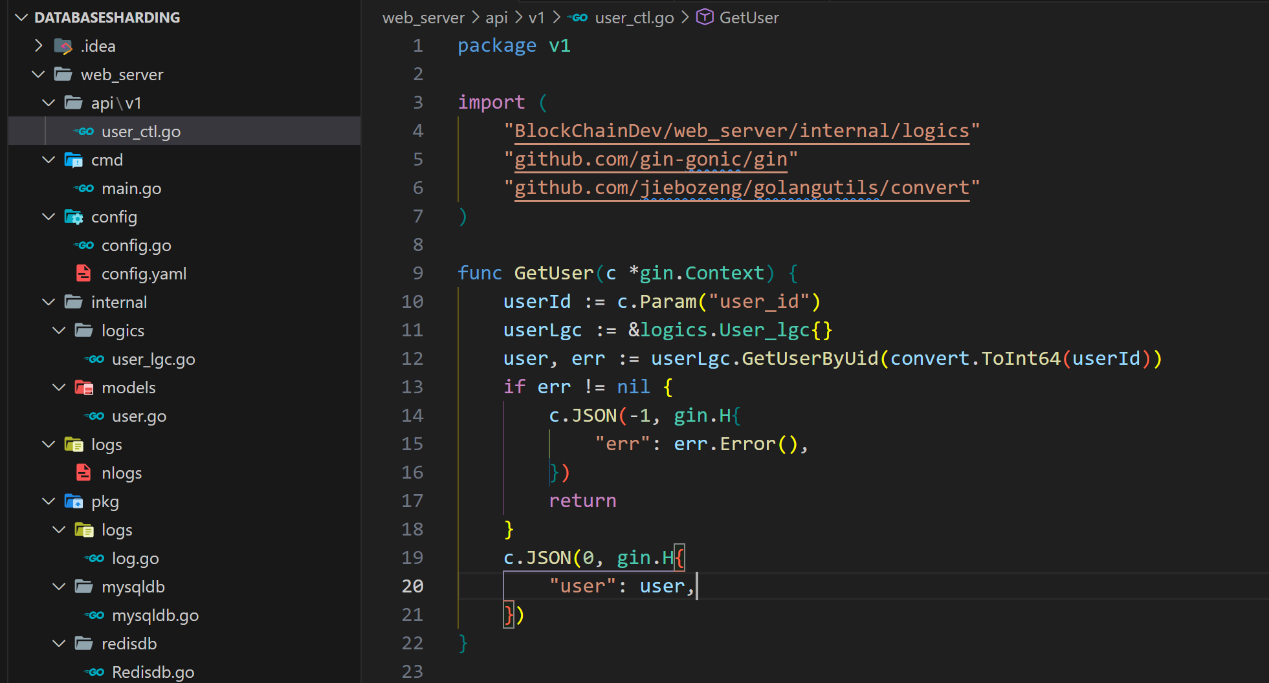


2. 移植之前实验的建立相关包，并创建相关文件

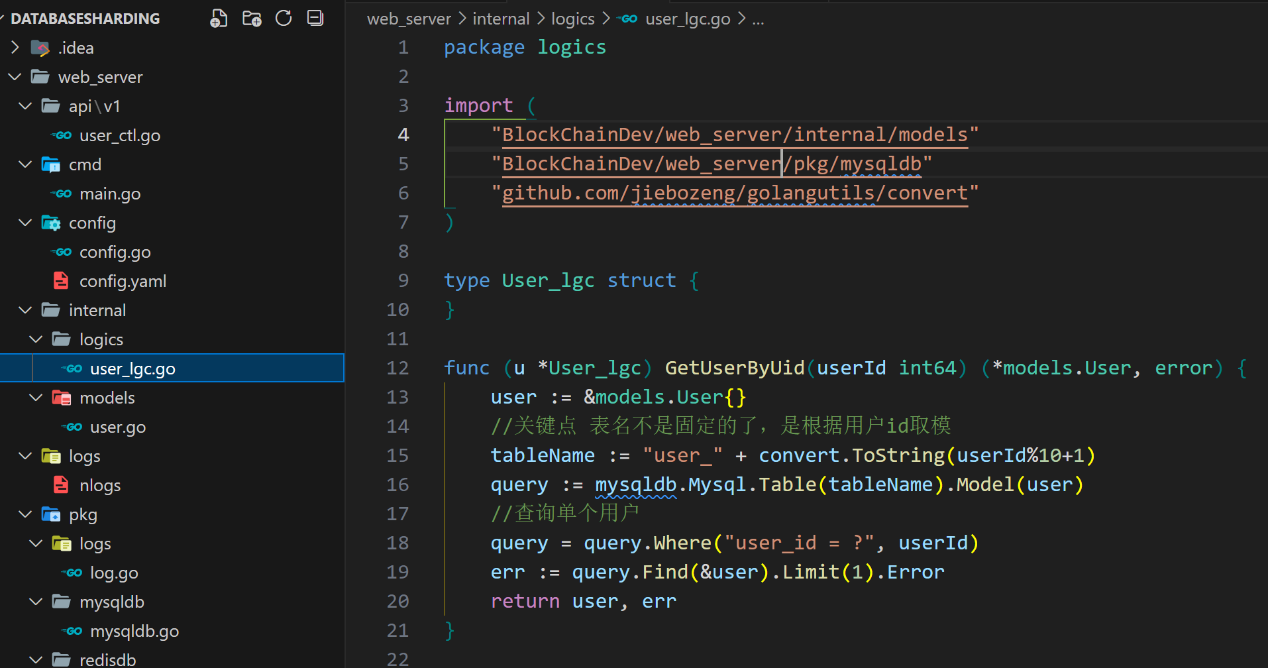
Config.yaml



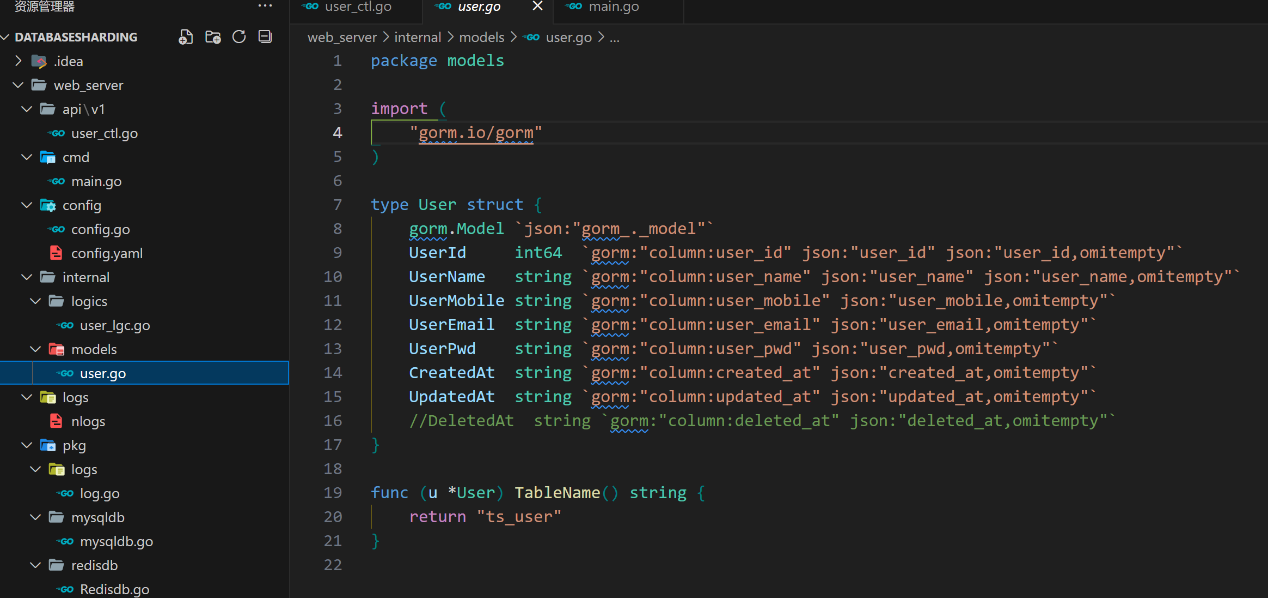
web\_server\api\v1\user\_ctl.go



web\_server\internal\logics\user\_lgc.go

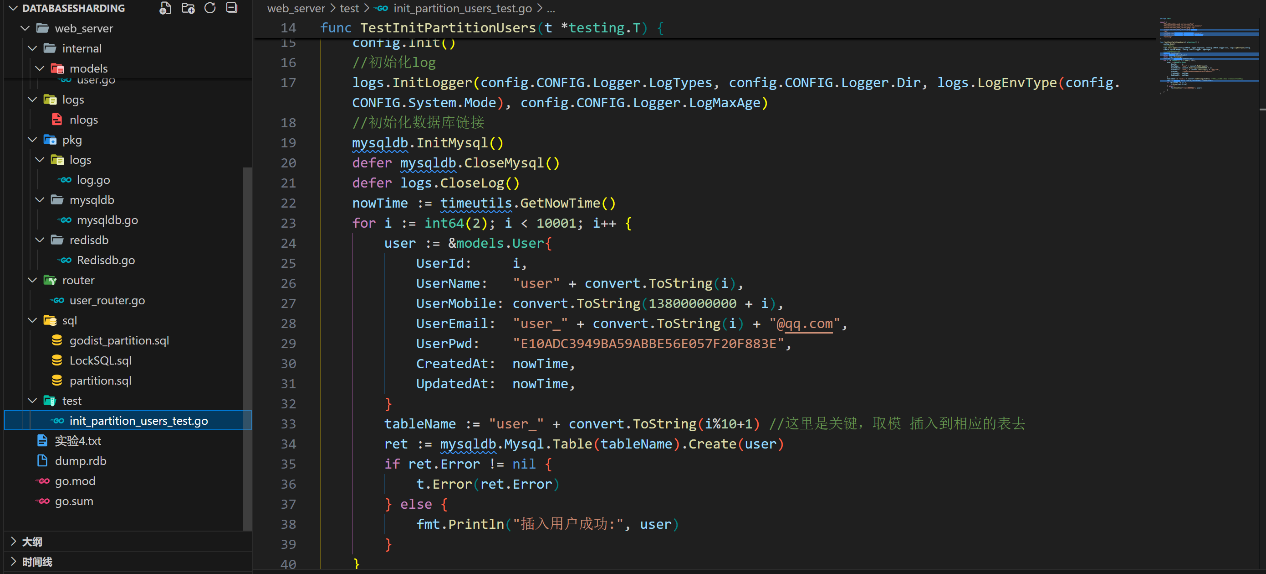


web\_server\internal\models\user.go

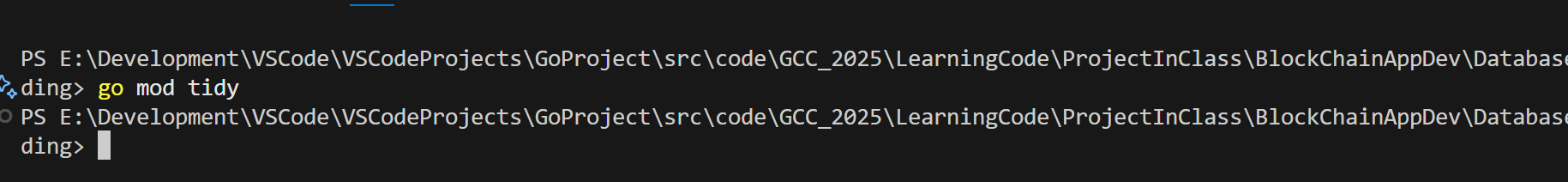


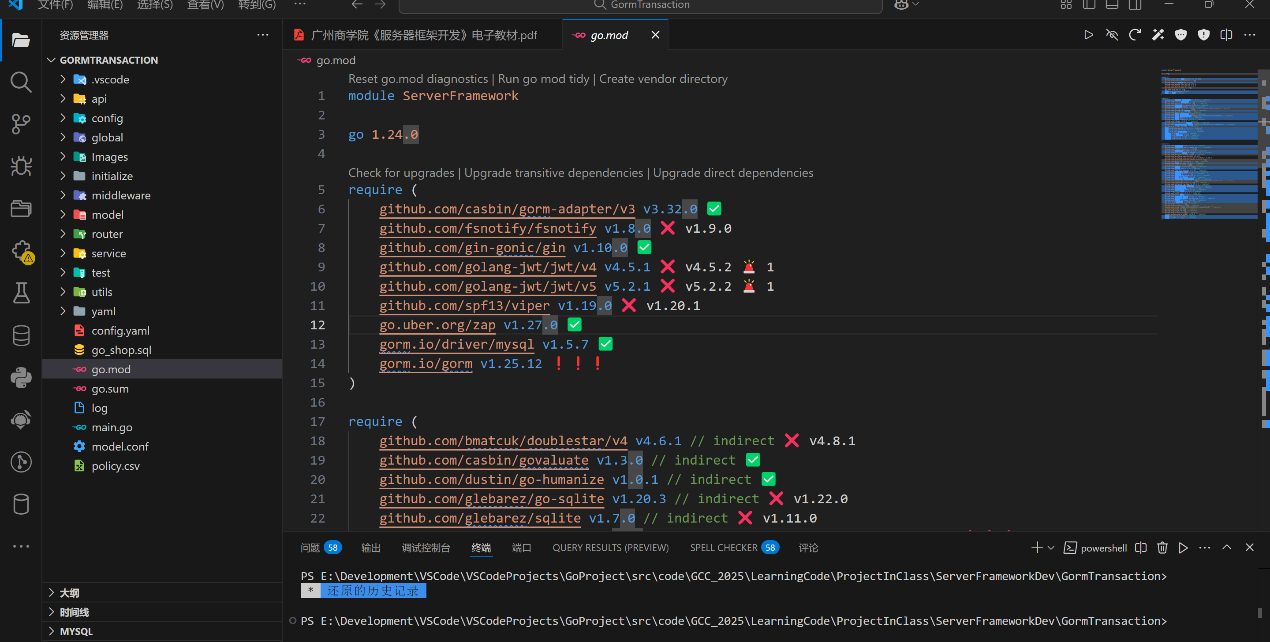
3.添加测试包test，用于将用户数据插入到数据库中

web\_server\test\init\_partition\_users\_test.go(这里最具有挑战性)



4.go mod tidy一下来获取所需的第三方包

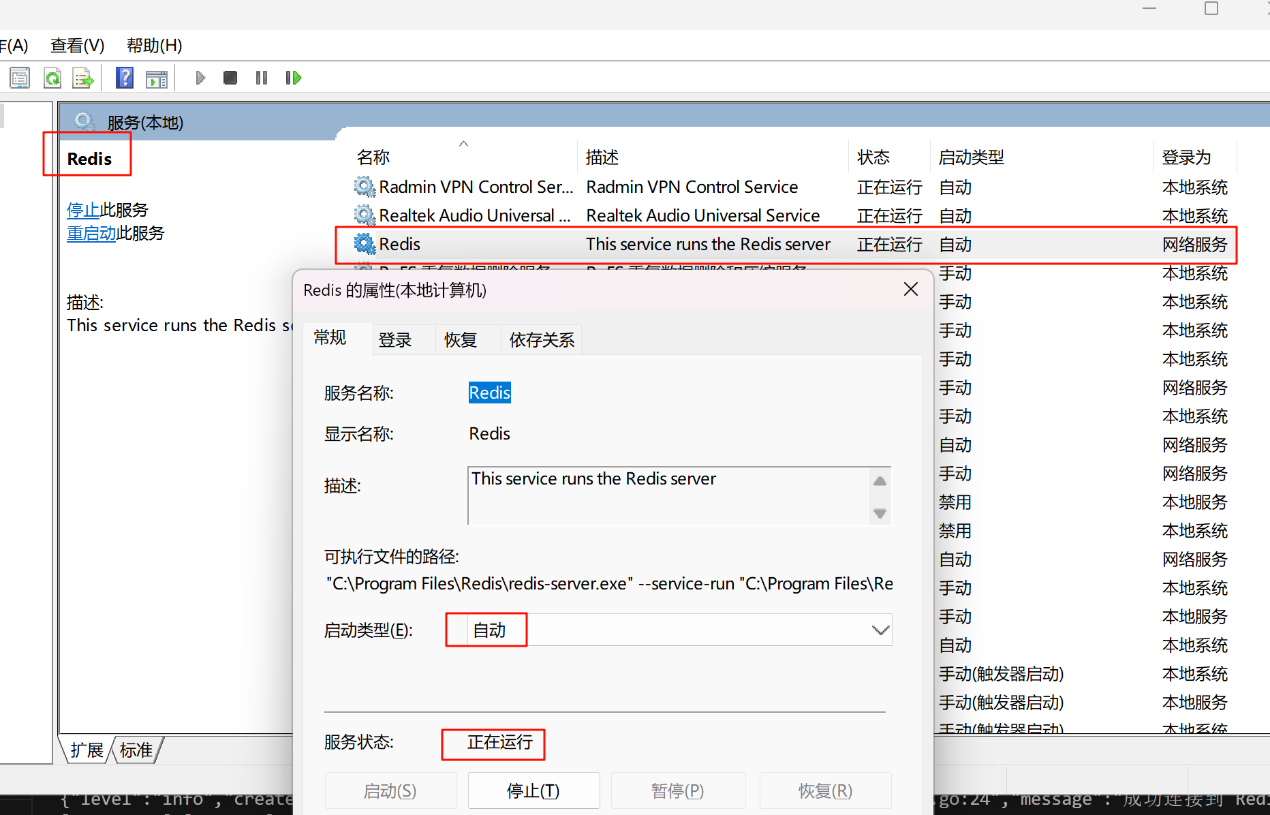




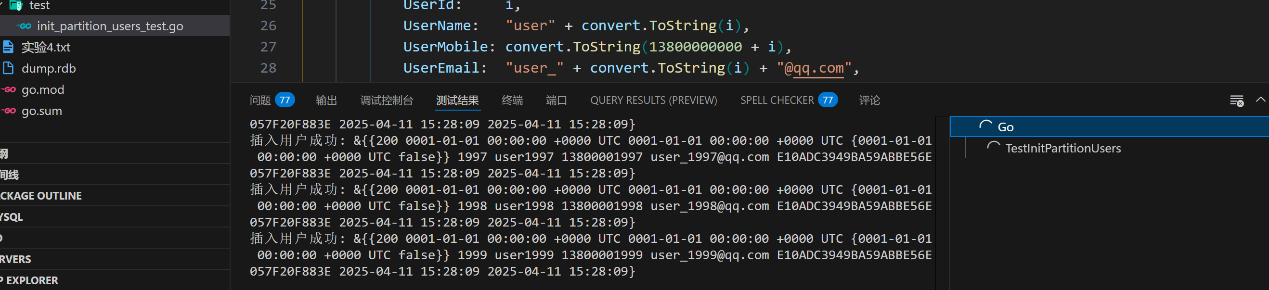
基本准备完毕，开始实验测试

**五、实验结果与分析**

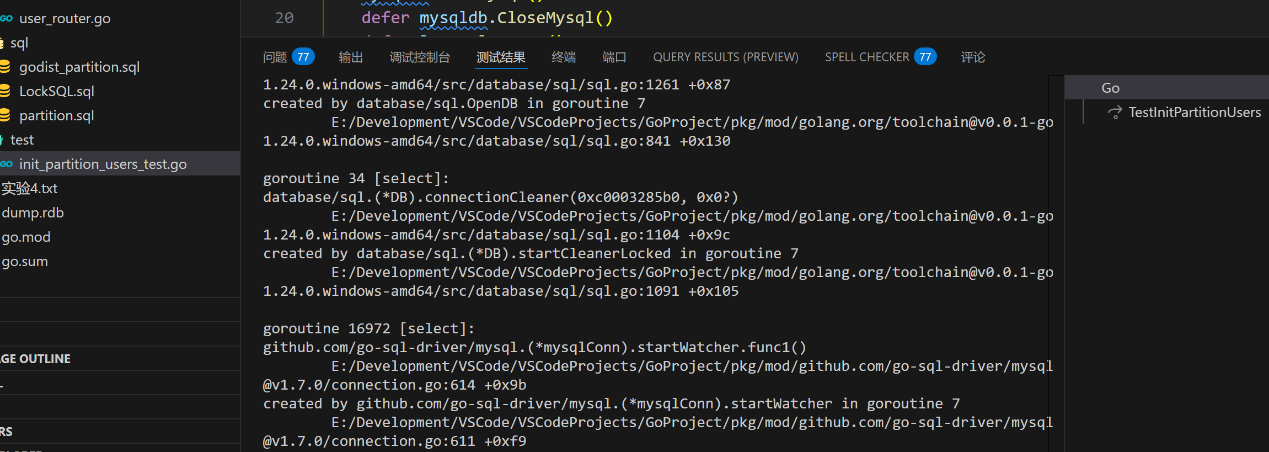
1.启动服务（Redis已设置为开机自启动，故无需redis-server）

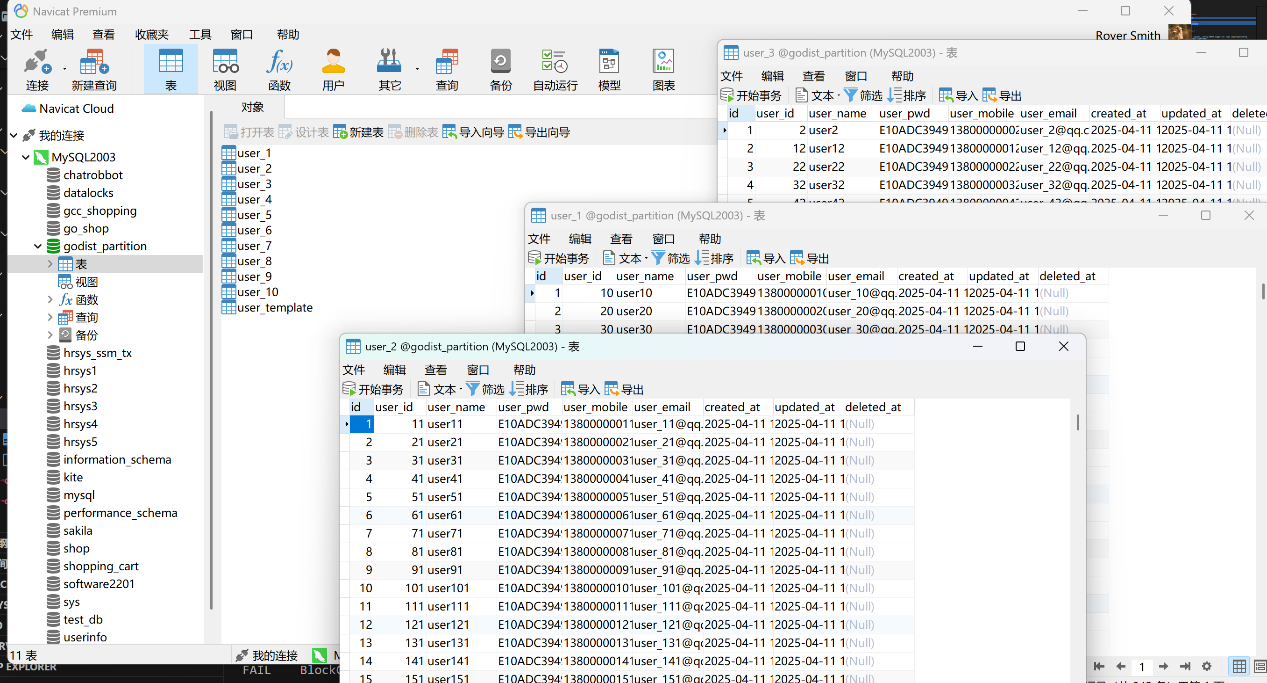


2.切换到init\_partition\_users\_test.go，点击run test调用

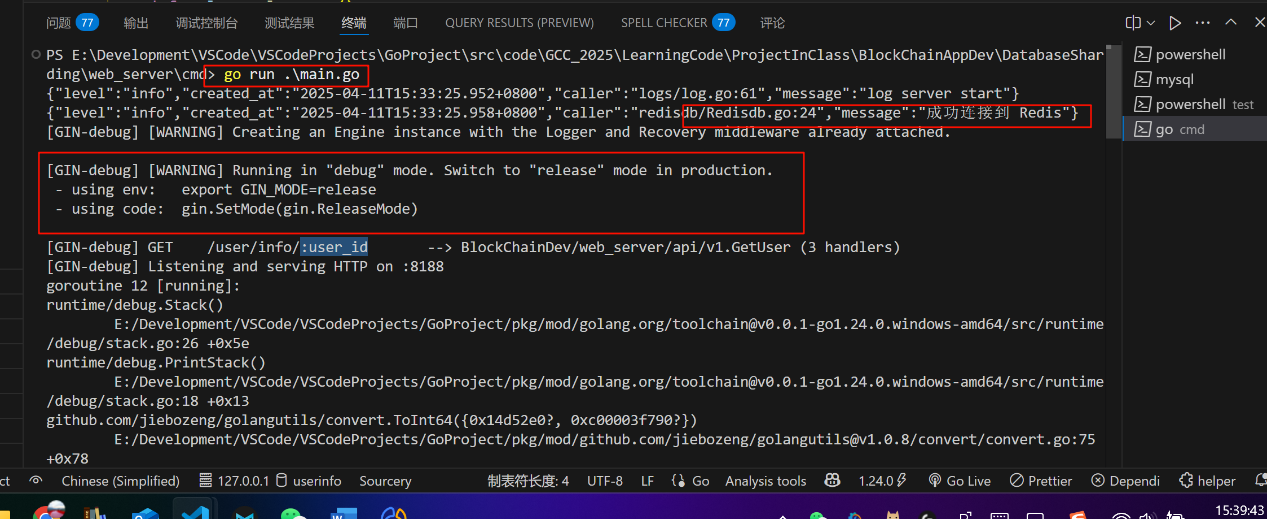
**添加数据中**

**添加数据完成**

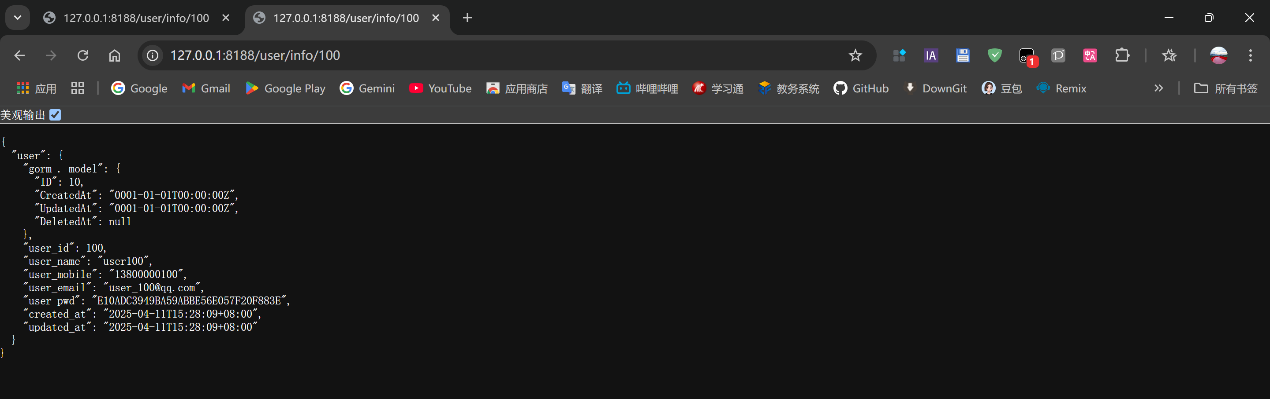




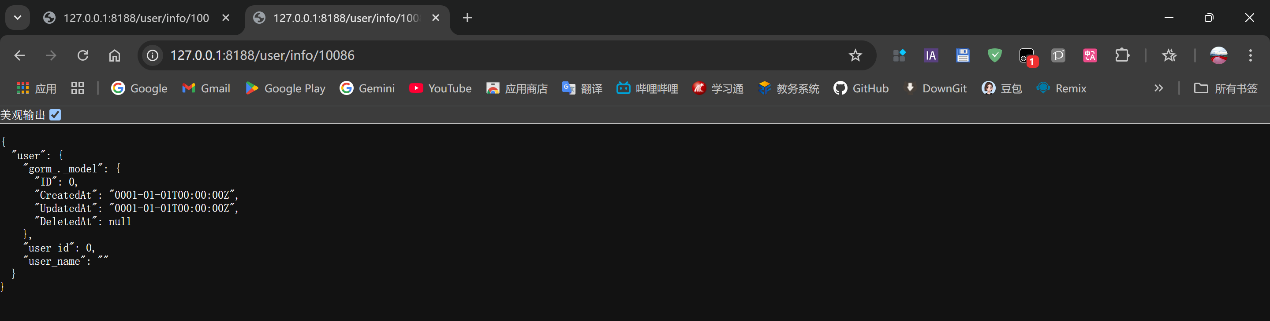
3.启动客户端（go run .\main.go）



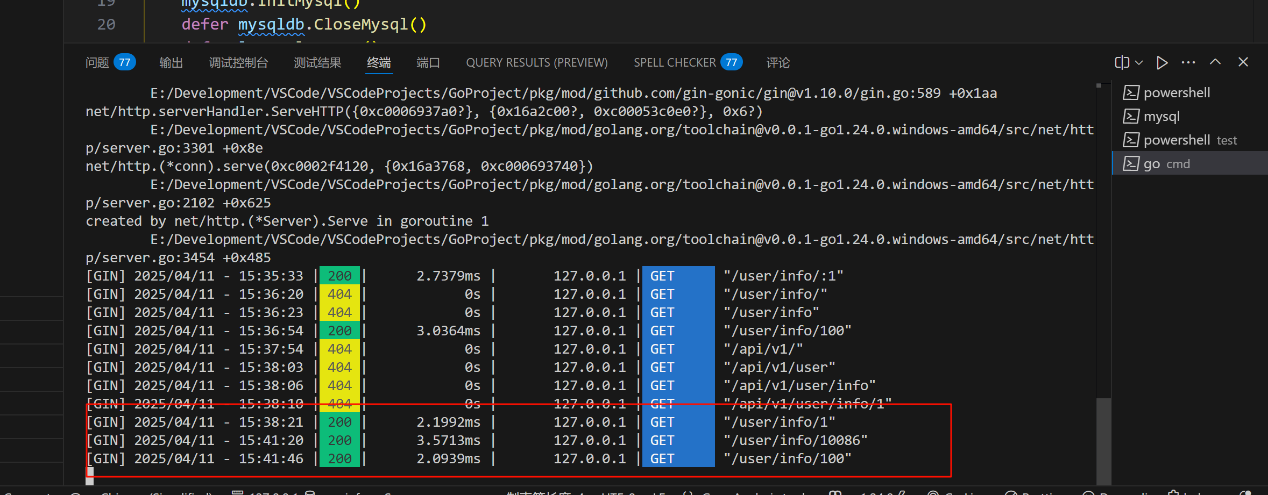
4. 返回浏览器查看GET内容（http://127.0.0.1:8188/user/info/100）



多个展示（id：10086）



5.返回终端查看Gin调用情况



**六、结论与体会**

通过本次实验，我切实掌握了MySQL数据库分表技术的核心要点与实践方法。实验中，依据用户ID取模进行分表操作，成功将大量用户数据分散存储，有效提升了数据处理效率。这不仅加深了我对分表原理和策略的理解，也让我认识到分表在优化数据库性能、提高系统扩展性方面的显著作用。

同时，我也遇到了一些问题，如查询路由和数据插入逻辑的处理等，通过解决这些问题，我的实践能力得到了锻炼，为后续开发大规模数据库应用积累了宝贵经验。

**七、教师评语**