

项目实施规范报告

项目概述

今天我介绍一下这半个月来的工作，即一个入门项目，这个项目是在凯哥的指导下完成的。

本项目采用 Spring Boot 作为基础框架，结合 MyBatis-Plus 操作 Mysql 数据库，同时利用 Redis 进行缓存优化，旨在构建一个高效、稳定的学生成绩管理系统。

这个项目的名称是学生成绩管理系统，而且我这个项目从工程角度来讲是没有难度的，所以我从一些项目准则的角度来介绍这个项目。

这个项目的技术栈和公司类似，使用的是SpringBoot+Mybatis-Plus+Redis+Mysql。

下面我将从代码、数据库设计、缓存设计、测试和不足这几个方面进行介绍。

代码规范

首先是几个代码规范

1. 遵循 Java 编码规范，包括命名约定、注释规范等。

- 类名采用大驼峰命名法，方法名和变量名采用小驼峰命名法。
- 为关键代码提供详细的注释，提高代码的可读性。

第一是java编码规范主要包括命名和注释，类名采用大驼峰命名法，方法名和变量名采用小驼峰命名法。为关键代码提供详细的注释，提高代码的可读性。

2. Spring Boot 相关代码分层规范

- Controller 层负责接收请求和响应处理，参数校验和异常处理遵循统一规范。
- Service 层负责业务逻辑处理，保持事务的一致性和完整性。
- Mapper 层（MyBatis-Plus Mapper）专注于数据库操作，保证数据访问的准确性和高效性。

第二是，SpringBoot代码分层的规范，Controller 层负责接收请求和响应处理，Service 层负责业务逻辑处理，Mapper 层（MyBatis-Plus Mapper）专注于数据库操作。

3. MyBatis-Plus 代码的映射和SQL书写规范。

- Mapper接口使用与对应的数据库表相关且有意义的名称，例如 UserMapper 对应 user 表。（接口名与表名对应）
- Mapper 接口的方法命名清晰反映数据库操作的意图。（方法名与意图对应）
- XML文件的 namespace 要与对应的 Mapper 接口全路径一致。（XML的路径与Mapper对应）
- XML 映射文件中的 SQL 语句书写规范，避免复杂和难以维护的查询。

第三是，MyBatis-Plus 的映射和SQL书写规范。Mapper接口名与数据库表对应，Mapper接口的方法名与数据库操作意图对应，XML的命名空间与Mapper接口对应，XML中的SQL书写规范。

4. Redis 相关数据命名和客户端操作规范

- 键的命名遵循统一的命名空间和规则，便于管理和识别。
- 对 Redis 操作进行封装，提供统一的接口，避免直接在业务代码中操作 Redis 客户端。

第四是 Redis 相关数据的命名和客户端操作也要符合规范。

redis缓存中的键要符合命名空间或者方法名的规范，操作Redis的客户端需要进行封装。

数据库设计规范

接下来是数据库设计规范，这部分会和我的项目相关性高一点。

1. 数据类型选择

- 为每个列选择合适的数据类型，以节省存储空间并提高性能。

第一是数据类型选择：

为每个列选择合适的数据类型，以节省存储空间并提高性能。例如在我的项目中UserID使用int,Role使用的是enum。

2. 索引设计

- 在 Username 等经常用于查询的列上创建索引。
- 避免过度创建索引，以免影响插入和更新操作的性能。

第二是索引设计：

在经常用于查询的列上创建索引，以节省存储空间并提高性能。例如在我的项目中我为Username添加了索引，这样在登录的时候可以快速的查询，避免全表扫描。避免过度创建索引，以免影响插入和更新操作的性能。

3. 范式遵循

- 尽量遵循第三范式，减少数据冗余。
- 为了提高查询性能，在teacher表违反了第三范式。

第三是范式遵循：

尽量遵循第三范式，减少数据冗余。

但是必要的时候也可以不遵循，上次同事也介绍过，合适的冗余可以提高查询性能。例如在我的项目中teachername就违反了第三范式

4. 主键和外键

- 为每张表定义主键，确保数据的唯一性和完整性。
- 在存在关联关系的表之间建立外键，以维护数据的一致性。

第四是主键和外键

为每张表定义主键，确保数据的唯一性和完整性。

在存在关联关系的表之间建立外键。例如在我的项目中，teachers表和students表中的teacherID和studentID就是与Users表中的UserID关联的外键。

缓存设计规范

然后是缓存设计

1. 明确缓存的更新策略，如主动更新、定时更新或基于数据变更的被动更新。
 - 本项目设计为修改了表就直接删除缓存。

一个好的项目应明确缓存更新策略，比如主动更新、被动更新等等。我这个项目比较简单，缓存更新选择了直接删除

2. 设置合理的缓存过期时间，根据数据的热度和更新频率进行调整。

测试规范

接下来是测试

1. 编写单元测试、集成测试和接口测试，确保代码的质量和功能的正确性。
 - 对数据库操作和缓存操作进行充分的测试。

每个项目都要进行测试，也许完备测试，也许只测试某些模块。我对这个项目进行了单元测试和接口测试。

不足之处

最后是不足之处

1. 可扩展性
2. 并发能力
3. 功能简单

第一是可扩展性差

因为我没有企业开发的经验，只是在学校给老师做一些很杂的项目，这些项目都是为了交差完全没有考虑过可扩展性。

第二是并发能力，项目设计之初就没有考虑并发的問題，而且项目课程的数量也没有限制。第三是功能简单，功能简单从数据库和接口这两方面都可以体现出来 数据库简单：一共五张表：

用户表、老师表、学生表、成绩表和课程表 接口简单：全部都是用的mybatis-plus自带的CRUD函数，一条sql都没有编写，可想而知功能有多简单。