



软件工程课程设计 考查报告

姓 名: 孙鹤轩
学 号: 20211000156
班 级: 191211
学 院: 计算机学院
题 目: 二次压降检测仪器鉴定系统数据库设计
指导 教师: 王勇

2023 年 12 月

目录

第 1 章系统背景意义.....	2
第 2 章系统需求分析概述.....	3
第 3 章系统数据模型描述.....	3
第 4 章系统数据库设计.....	4
4.1 表结构.....	4
4.2 SQL 语句设计.....	6
4.2.1 根据被检测仪器设备编号查询出二次压降检测仪器出厂信息.....	6
4.2.2 根据被检测仪器设备编号查询查询出该设备检测实验输入信息.....	6
4.2.3 根据被检测仪器设备编号查询出检测结果信息.....	7
4.2.4 根据被检测仪器设备编号查询出检测报告信息，检测报告包括：部分出厂信息、检测实验输入信息以及检测结果信息.....	7
第 5 章 springboot 开发路线.....	7
5.1 创建项目.....	7
5.2 数据库配置连接.....	9
5.3 Service 开发.....	9
5.4 Mapper 开发.....	9
5.5 Controller 开发.....	10
第 6 章总结.....	10

第 1 章系统背景意义

电力系统是现代社会的不可或缺的基础设施，为各行各业的正常运行提供稳定可靠的电力供应。在电力系统中，互感器是一种关键的电力设备，用于测量和传输电能。互感器的作用是将高电压（主电路）转换为低电压（次级电路），以便进行测量、保护和控制。它们广泛应用于变电站、输电线路、配电系统以及工业和商业用电中。

在互感器中，二次压降是一个重要的性能参数。二次压降是指互感器二次侧输出电压相对于输入电压的降低量。它直接影响到互感器的负载能力、传输效率和精确度。在正常运行情况下，互感器的二次压降应该在一定的范围内，以保证电能传输的稳定性和准确性。然而，由于互感器长期运行、负载变化、环境条件等因素的影响，互感器的二次压降可能会出现异常或超出规定范围。

互感器二次压降的异常或超标可能导致一系列问题。首先，如果二次压降过高，会降低互感器的负载能力，导致电能传输的效率下降。这可能导致供电不稳定、电压波动或过载等问题，影响电力系统的正常运行。其次，互感器二次压降异常可能导致电能计量的准确性下降，给能源计费带来困扰，并可能引发纠纷。此外，异常的二次压降还可能损害互感器本身，导致性能的进一步恶化甚至故障。因此，及时检测和检定互感器的二次压降是确保电力系统的稳定运行和供电质量的重要环节。

为了解决互感器二次压降的检测和评估问题，互感器二次压降检测仪检定系统应运而生。该系统由互感器二次压降检测仪和相关的校准设备组成。互感器二次压降检测仪是一种专门设计用于测量互感器二次压降的仪器，通过将其连接到互感器的二次侧，可以准确测量和记录互感器的二次压降数值。校准设备用于对互感器二次压降检测仪进行校准和验证，确保其测量结果的准确性和可靠性。

互感器二次压降检测仪检定系统的应用具有重要意义。首先，它可以帮助电力系统运维人员及时发现互感器二次压降异常或超标的情况。通过定期进行检定和校准，可以监测互感器的性能，并及时采取措施进行维修或更换，以保证电力系统的稳定运行。其次，该系统可以提高电能计量的准确性。通过确保互感器二次压降检测仪的准确性，可以提高电能计量的可靠性，避免计量误差带来的争议和纠纷。此外，互感器二次压降检测仪检定系统还可以提高电力系统的效率。准确测量互感器二次压降可以帮助优化电力系统的运行，减少能源损耗，提高能源利用效率，从而降低能源成本和环境影响。

互感器二次压降检测仪检定系统还对电力设备的安全运行起到关键作用。通过检定互感器二次压降检测仪，可以发现潜在的故障或安全隐患，及时采取措施进行修复或更换，保障电力设备的安全性。这有助于防止互感器故障引发的事故，减少停电和损坏设备的风险，保障电力系统和用户的安全。

总之，互感器二次压降检测仪检定系统在保证电力系统稳定性、提高电能计量准确性、提高电力系统效率以及保障电力设备安全性方面具有重要的意义和应用价值。它为电力系统运维人员提供了一种有效的工具，确保电力系统的安全运行和提高供电质量，并为相关研究和技术的发展提供了基础和指导。通过采用互感器二次压降检测仪检定系统，电力系统可以更好地满足日益增长的电力需求，为社会经济发展提供可靠的电力支持。

第 2 章系统需求分析概述

互感器二次压降检测仪器鉴定系统应主要完成鉴定实验与实验数据存储和统计查询，具体业务功能见下表：

功能名称	功能描述
接收输入数据	对输入的实验参数和设备信息进行封装
进行检测实验	根据输入信息开始实验，并封装实验结果
GUI 显示实验结果	以相对友好的界面显示实验结果
存储实验结果数据	保存实验结果到数据库中
存储被检测设备信息	保存被检测设备的相关信息
存储实验日志	保存实验日志（包括设备信息，实验结果，实验日期等），便于查询
查询实验日志	根据设备 id 查询实验日志，回显 GUI

表 2-1

综上所述，该系统应完成根据输入数据完成检测实验以及相关数据存储功能。

第 3 章系统数据模型描述

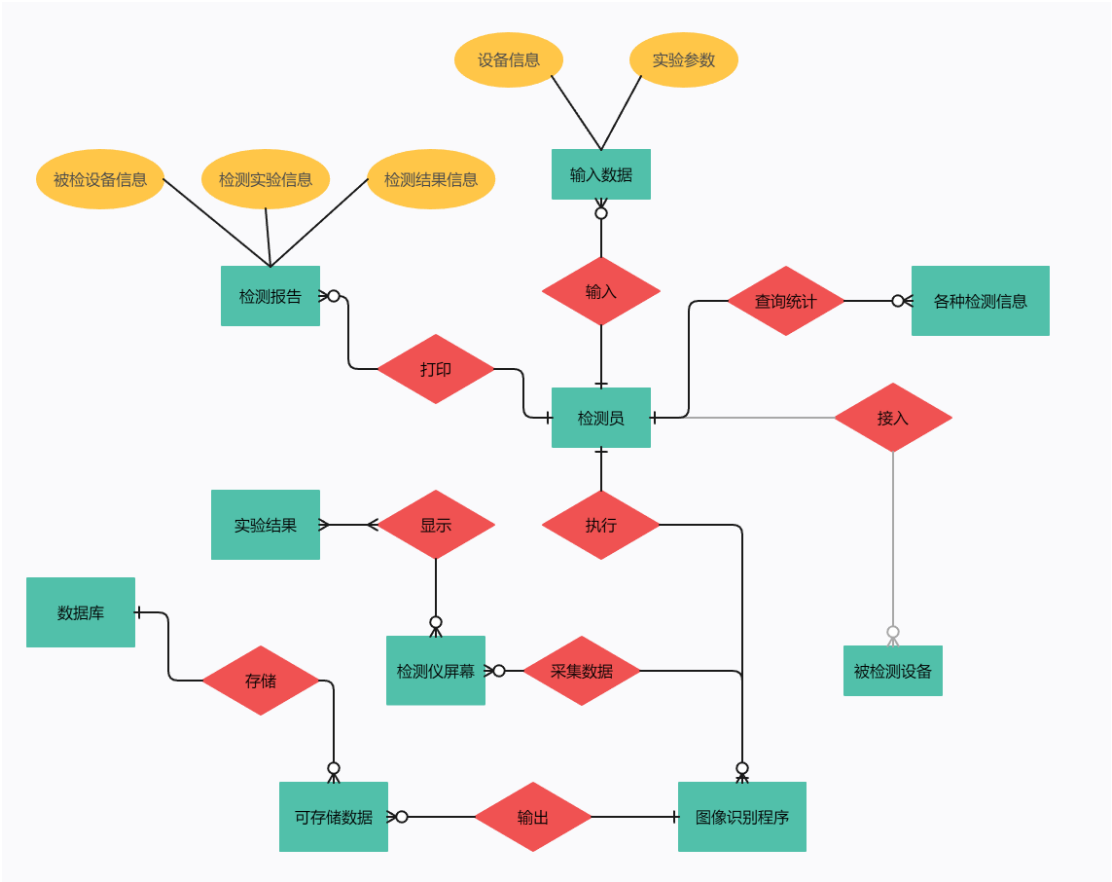


图 3-1 系统 ER 图

上图给出了互感器二次压降检测仪器鉴定系统的 ER 图，其描述如下：

1. 实体:

- 检测员: 表示操作系统的实体
- 被检测设备: 表示要鉴定的二次压降检测仪器实体
- 鉴定检测仪屏幕: 表示显示实验结果的屏幕实体
- 图像识别程序: 表示识别屏幕显示数据的程序实体
- 输入数据: 表示输入鉴定仪器的实验参数实体
- 实验结果: 表示鉴定仪器的鉴定结果实体
- 可存储数据: 表示经过图像识别程序整理后的数据实体
- 各种检测信息: 表示检测员根据实验结果统计的检测信息实体
- 检测报告: 表示经管理员打印的实验结果报告实体
- 数据库: 表示实验数据存储的存储实体

2. 属性:

- 输入数据的属性:
 - 设备信息: 表示被检测设备的设备信息 (厂家等)
 - 实验参数: 表示被检测设备的规格 (额定电压等)
- 检测报告的属性:
 - 被检设备信息: 表示被检测设备的设备信息 (厂家等)
 - 检测实验信息: 表示鉴定系统的实验信息 (输入, 日志等)
 - 检测结果信息: 表示鉴定系统的鉴定结果

3. 关系:

- 输入关系: 表示输入数据与检测员的关系, 由检测员输入数据到鉴定系统
- 打印关系: 表示检测报告与检测员的关系, 由检测员打印检测报告
- 查询统计关系: 表示检测信息与检测员的关系, 由检测员查询统计各种监测信息
- 执行关系: 表示图像处理程序与检测员的关系, 由检测员选择执行图像处理程序来提取实验结果数据
- 显示关系: 表示实验结果与检测屏幕的关系, 由检测屏幕显示实验结果
- 采集数据: 表示图像处理程序与检测屏幕的关系, 由图像处理程序采集检测屏幕显示的实验数据并整理成可存储数据
- 输出关系: 表示图像处理程序与可存储数据的关系, 由图像处理程序完成后输出可存储数据
- 存储关系: 表示可存储数据与数据库的关系, 由数据库存储可存储数据

第 4 章系统数据库设计

4.1 表结构

为满足题目描述要求以及前后端交互需求, 本次数据库设计共设计了四张表, 分别是 factory_info、trial_input、trial_result、trial_log, 其具体设计如下:

factory_info（厂家信息）表结构：

字段名	数据类型	是否允许 NULL 值	说明
id	bigint	否	设备编号、主键
name	varchar	否	设备名称
factory_name	varchar	否	厂家名称
origin	varchar	否	生产地
manufacture_time	varchar	否	出厂日期

trial_input（输入数据）表结构：

字段名	数据类型	是否允许 NULL 值	说明
current_level	varcahr	否	电流档位
voltage_level	varchar	否	电压档位
test_voltage	float	否	试验电压
protection_voltage	float	否	保护电压
protection_current	float	否	保护电流
trial_time	int	否	实验事件
regulator_speed	varchar	否	调压器速率
trial_log_time	datetime	否	实验记录时间

trial_result（实验结果）表结构：

字段名	数据类型	是否允许 NULL 值	说明
id	bigint	否	设备编号
Uab	float	否	电压 Uab
Ubc	float	否	电压 Ubc
Uca	float	否	电压 Uca
avgU	float	否	三相平均电压

Ia	float	否	电流 Ia
Ib	float	否	电流 Ib
Ic	float	否	电流 Ic
avgI	float	否	三相平均电流
trial_log_time	datetime	否	实验记录时间
name	varchar	否	设备名称

trial_log（实验日志）表结构：

字段名	数据类型	是否允许 NULL 值	说明
id	bigint	否	设备编号
name	varchar	否	设备名称
trial_time	datetime	否	日志记录时间

4.2 SQL 语句设计

4.2.1 根据被检测仪器设备编号查询出二次压降检测仪器出厂信息

```
select * from factory_info where id = #{id}
```

使用 select 查询语句查询,*表示查询所有字段,where id = #{id}为筛选条件,#{id}为 MyBatis 动态参数语法

4.2.2 根据被检测仪器设备编号查询出该设备检测实验输入信息

```
select * from trial_input where id = #{id}
```

同样使用 select 查询语句查询,含义与 4.2.1 一样

4.2.3 根据被检测仪器设备编号查询出检测结果信息

```
select * from trial_result where id = #{id}
```

同样使用 select 查询语句查询

4.2.4 根据被检测仪器设备编号查询出检测报告信息, 检测报告包括:

部分出厂信息、检测实验输入信息以及检测结果信息

```
select  
trial_result.id, trial_result.Uab, trial_result.Ubc, trial_result.Uca, trial_result.avgU, trial_result.Ia,  
trial_result.Ib, trial_result.Ic, trial_result.avgI, trial_result.trial_log_time, trial_result.name, trial_input  
.current_level, trial_input.voltage_level, trial_input.test_voltage, trial_input.protection_voltage,  
trial_input.protection_current, trial_input.trial_time, trial_input.regulator_speed, factory_info.factor  
y_name, origin, manufacture_time  
from trial_result
```

```
    join trial_input on trial_result.id = trial_input.id
```

```
    join factory_info on trial_result.id = factory_info.id
```

```
where trial_result.id = #{id}
```

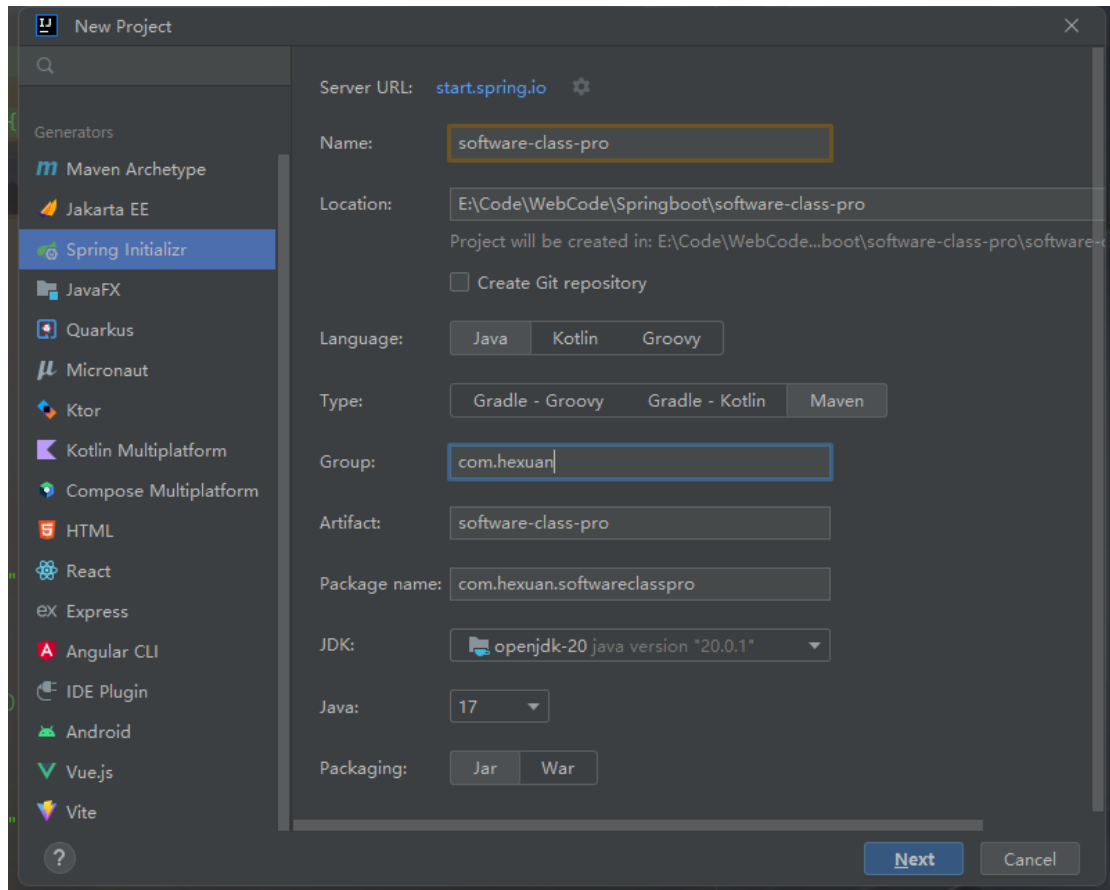
上述 sql 语句使用 select 对三张表进行联合查询通过 join 关键字指定通过设备编号 id 来查询

第 5 章 springboot 开发路线

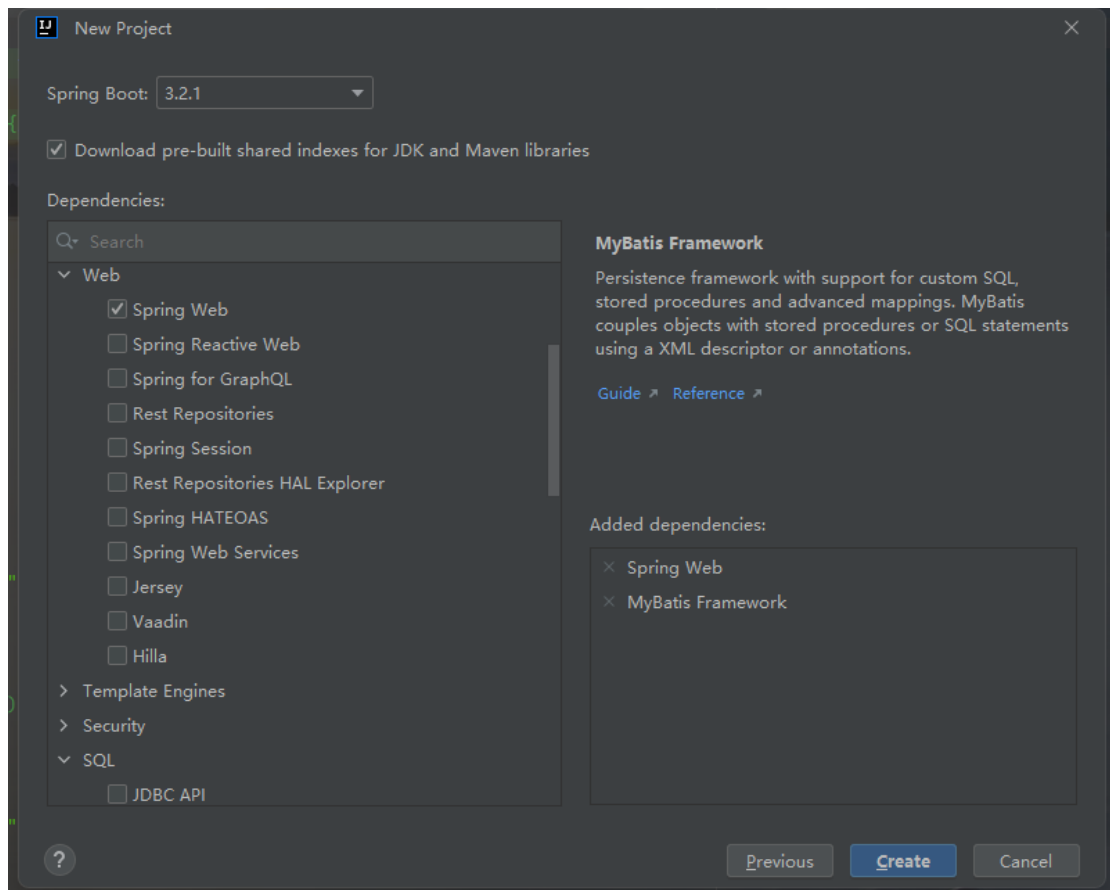
本次课程设计完成了前后端联调, 后端通过 springboot 框架开发

5.1 创建项目

通过 idea 的创建窗口创建 springboot 项目



选则 **springweb** 和 **MyBatis** 依赖模块



5.2 数据库配置连接

进入 `resource` 目录下的 `application.yml` 文件，添加以下配置
spring:

#数据库连接信息

datasource:

driver-class-name: com.mysql.cj.jdbc.Driver

url: jdbc:mysql://localhost:3306/your database name

username:your user name

password: your password

#Mybatis 配置

mybatis:

configuration:

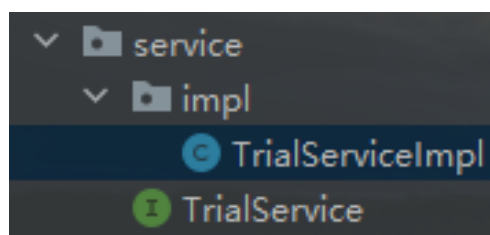
log-impl: org.apache.ibatis.logging.stdout.StdOutImpl

map-underscore-to-camel-case: true#驼峰命名

5.3 Service 开发

Service 的主要作用是调用持久层框架 MyBatis 进行数据库操作，负责将数据传输给 MyBatis 的 mapper 函数

- 编写 Service 接口：新建软件包 `service` 添加 `TrialService` 接口并添加注解 `@Service`
- 编写 Service 实现类，对 `TrialService` 中的方法进行实现



5.4 Mapper 开发

Mapper 的主要作用是对数据库进行操作，通过注解 `@Select`、`@Insert` 等注解在函数上来完成数据库操作，如下例：

```
@Select("select * from factory_info where id = #{id}")
```

```
FactoryInfo getFatoryInfo(Long id);
```

5.5 Controller 开发

Controller 的作用是处理前端发送过来的请求，格式如下：

```
@GetMapping("/getFactoryInfo/{id}")
public Result getFactoryInfo(@PathVariable Long id){
    log.info("获取出厂信息");
    FactoryInfo factoryInfo;
    factoryInfo = trialService.getFactoryInfo(id);
    return Result.success(factoryInfo);
}
```

通过注解 `@GetMapping` 注册控制器，`GetMapping` 在 `规定` 中处理查询请求，而上传请求常用 `@PostMapping` 注解，上述控制器请求路径采用的是路径参数。

第 6 章总结

本次课程设计主要进行了数据库设计和 SQL 语句设计，主要是进行查询操作，同时为了满足前后端开发的需求，我自行设计了许多 SQL 语句用于数据的插入等，使我对 mysql 数据库的操作以及数据库表结构设计思路更加熟练，同时对自学的 java 前后端开发知识有了验证，使我受益匪浅。