

数据库设计说明书

1 引言

1.1 编写目的

- 本数据库设计说明书是关于此烟草系统数据库设计，主要包括数据概念结构的设计、逻辑结构设计、物理结构设计以及数据字典、运行环境、安全设计等
- 读者范围：用户、系统设计人员、系统开发人员、系统测试人员、系统维护人员
- 本数据库设计说明书根据系统需求分析设计所编写
- 为团队开发提供参考基础

1.2 背景

传统的烟草生长过程需要耗费大量人力和时间来监测和调节影响烟草生长的因素，如温度、湿度、光照等。这种传统的人工管理方式存在效率低下、易受人为因素影响以及无法实时响应问题等诸多不足之处。为了解决这些问题并提高烟草的生长质量和产量，我们致力于开发一个智能烟草生长管理系统

1.3 定义

- MySQL：一个关系型数据库管理系统，在 WEB 应用方面，MySQL是最好的RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。
- Navicat：Navicat Premium 是一套可创建多个连接的数据库开发工具，让你从单一应用程序中同时连接 MySQL、Redis、MariaDB、MongoDB、SQL Server、Oracle、PostgreSQL 和 SQLite。
- 概念数据模型：一种面向用户、面向客观世界的模型，主要用来描述世界的概念化结构
- 逻辑数据模型：一种面向数据库系统的模型，是具体的DBMS所支持的数据模型
- 物理数据模型：一种面向计算机物理表示的模型，描述了数据在储存介质上的组织结构

1.4 参考资料

- [1] 柳伟卫.Spring Boot 企业级应用开发实战[J].北京大学出版社,2018年3月
- [2] 杨开振.深入浅出springBoot2.x.[J/OL].人民邮电出版社,2018年8月
- [3] 《数据库设计说明书》国标规范文本

- [4] 《数据库系统概论》王珊，萨师煊编著

2 外部设计

2.1 标识符和状态

表名	标识符或名称	描述信息	状态
alert_data	警报id	用于存放警报相关信息	使用
soil_data	传感器id	用于保存土壤数据	使用
automation_status_table	记录id	用于保存自动化操作记录	使用
device_state	记录id	用于保存设备状态记录	使用
light_data	传感器id	用于存储光照数据	使用
light_instructions	记录id	用来保存灯光操作记录	使用
opinions	意见id	用于保存用户反馈意见	使用
soil_data	传感器id	用于存储土壤数据	使用
temperature_data	传感器id	用于存储气温数据	使用
userinfo	用户id	用于存储用户信息数据	使用
users	用户id	用户存储用户数据	使用

2.2 开发环境

- 操作系统：windows 10
- 数据库版本：MySQL 8.0.24
- 开发集成环境：IntelliJ IDEA 2020.2.3 x64
- 开发语言：Java 1.8

2.3 开发规范

- 字符集采用utf8
- 排序规则用utf8_general_ci

- 按设计编辑字段名、类型、长度、Notnull、键等
- 注意数据库名和表名的大小写

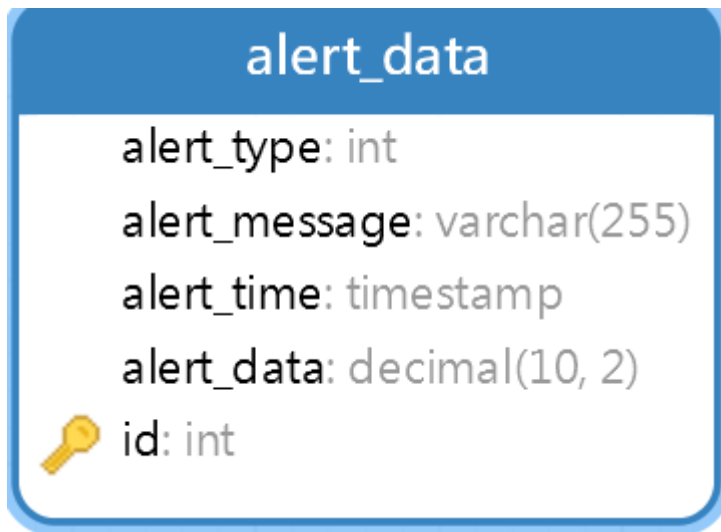
3 结构设计

3.1 概念结构设计

3.1.1 实体图

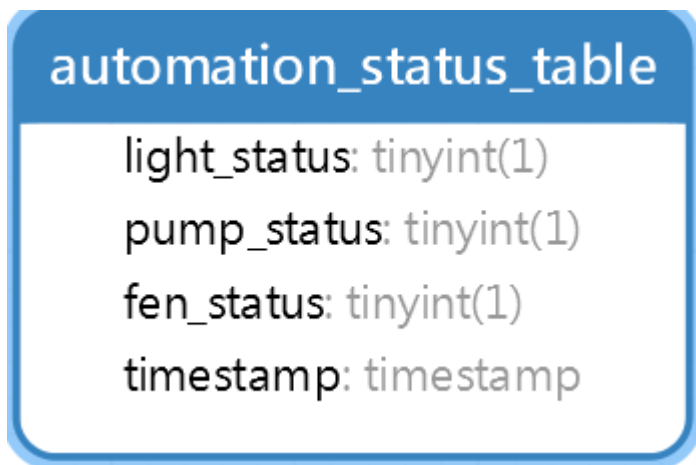
表名: `alert_data`

- 警报数据（警报类型，警报信息，警报时间，警报数据，ID）



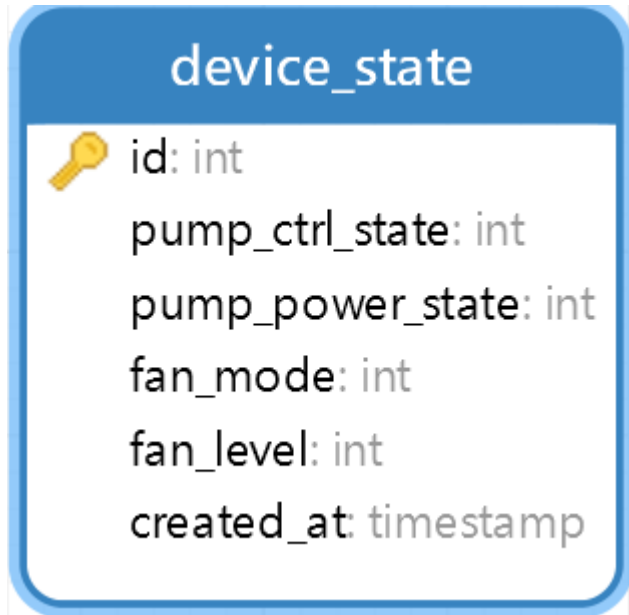
表名: `automation_status_table`

- 自动化状态数据（灯光状态，水泵状态，风机状态，时间戳）



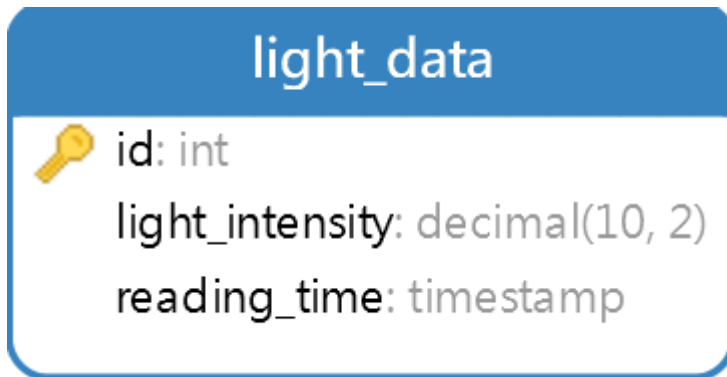
表名: `device_state`

- 设备状态数据 (ID, 水泵控制状态, 水泵电源状态, 风扇模式, 风扇级别, 创建时间)



表名: light_data

- 光照数据 (ID, 光照强度, 读取时间)



表名: light_instructions

- 灯光指令数据 (ID, 灯光模式, 灯光级别, 创建时间)

light_instructions



id: int

light_mode: int

light_level: int

created_at: timestamp

表名: opinions

- 用户意见反馈 (ID, 电话号码, 描述, 日期)

opinions



id: int

phone_number: varchar(255)

description: text

date: timestamp

表名: soil_data

- 土壤数据 (ID, 土壤湿度, 读取时间)

soil_data



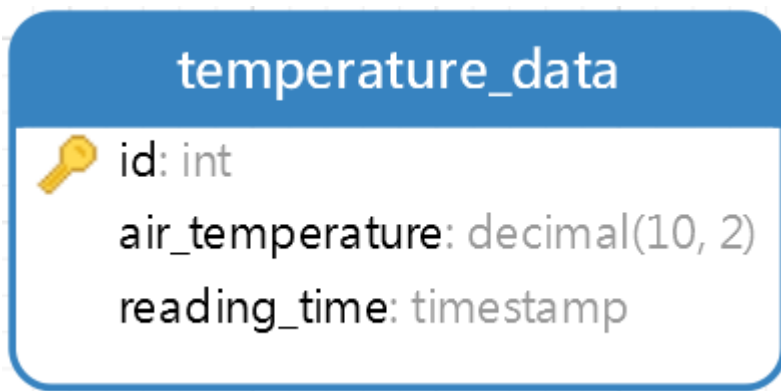
id: int

soil_humidity: decimal(10, 2)

reading_time: timestamp

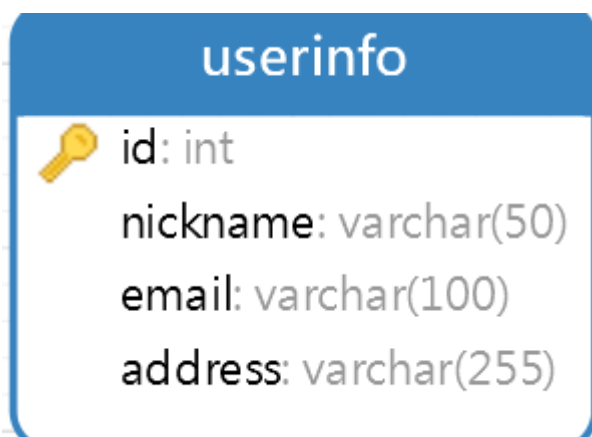
表名: temperature_data

- 温度数据 (ID, 空气温度, 读取时间)



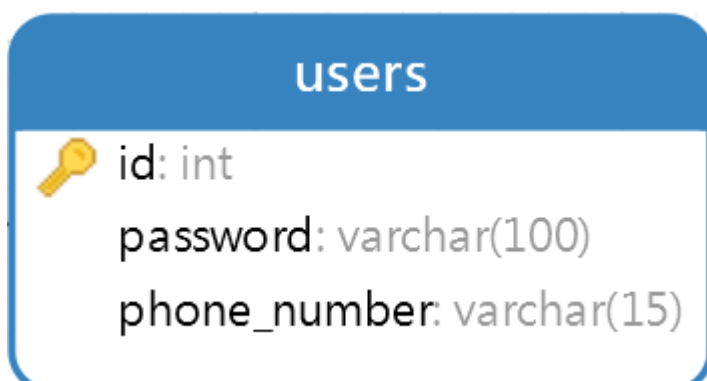
表名: userinfo

- 用户信息 (ID, 昵称, 电子邮件, 地址)

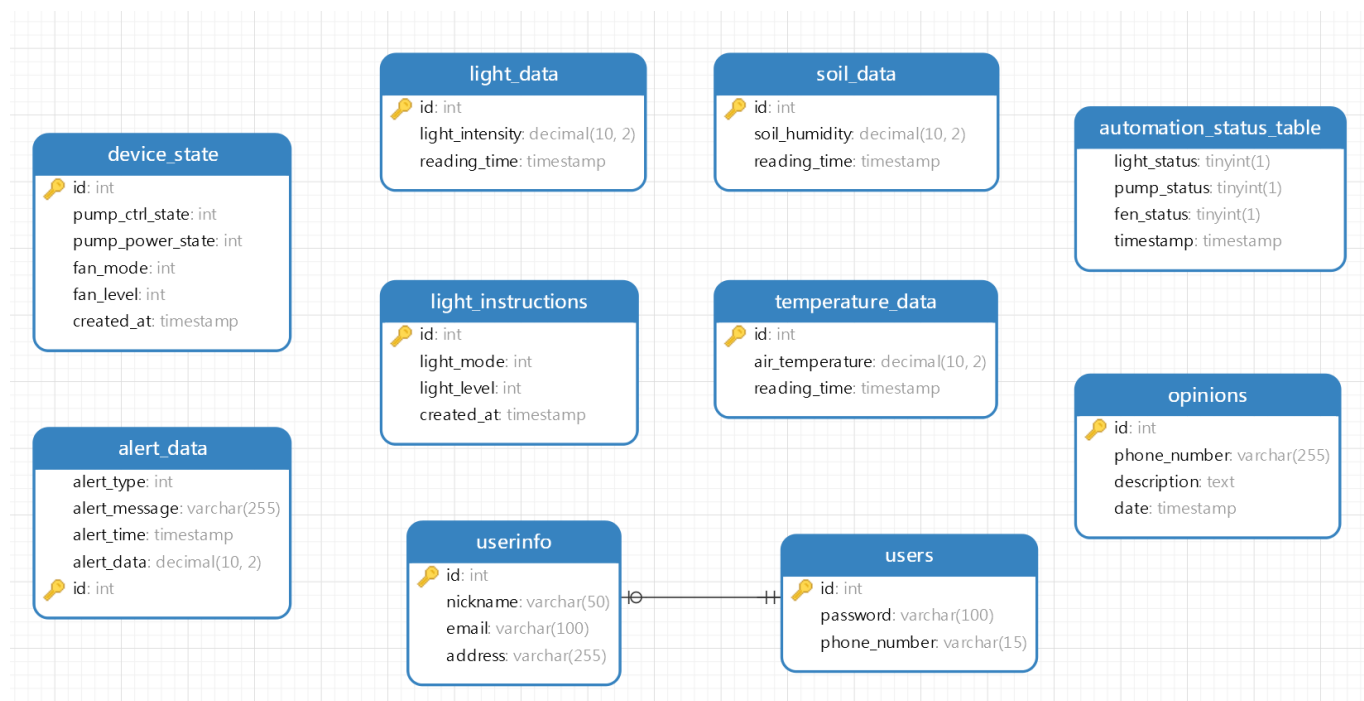


表名: users

- 用户数据 (ID, 密码, 电话号码)

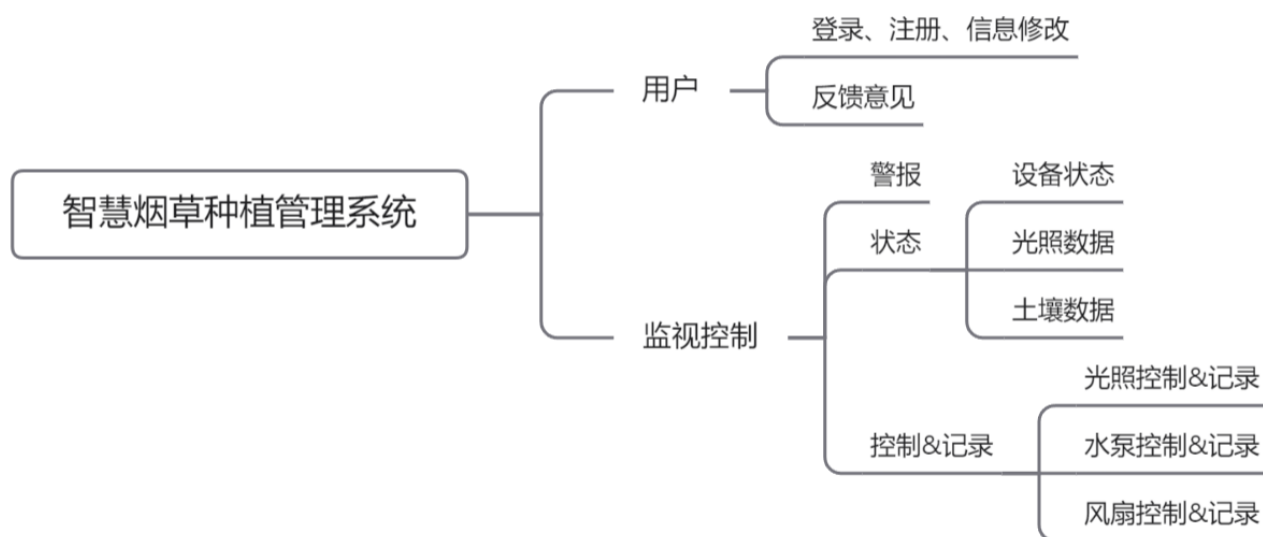


3.1.2 E-R图



3.2 逻辑结构设计

把上述原始数据进行分解、合并后重新组织起来的数据库全局逻辑结构，包括所确定的关键字和属性、重新确定的记录结构和文卷结构、所建立的各个文卷之间的相互关系，形成本数据库的数据库管理员视图。



4 运用设计

4.1 数据字典设计

对数据库设计中涉及到的各种项目，如数据项、记录、系、文卷、模式、子模式等一般要建立起数据字典，以说明它们的标识符，同义名及有关信息，在本节中要说明对此数据字典设计的基本考虑。 **表名:** alert_data

字段名	数据类型	长度	主键	非空	描述
alert_type	int			是	警报类型
alert_message	varchar(255)	255		是	警报信息
alert_time	timestamp				警报时间，默认为当前时间
alert_data	decimal(10,2)				警报数据
id	int		是	是	ID

表名: automation_status_table

字段名	数据类型	长度	主键	非空	描述
light_status	tinyint(1)	1			灯光状态
pump_status	tinyint(1)	1			水泵状态
fen_status	tinyint(1)	1			风机状态
timestamp	timestamp				时间戳，默认为当前时间

表名: light_data

字段名	数据类型	长度	主键	非空	描述
id	int		是	是	ID
light_intensity	decimal(10,2)				光照强度
reading_time	timestamp				读取时间，默认为当前时间

表名: light_instructions

字段名	数据类型	长度	主键	非空	描述
id	int		是	是	ID
light_mode	int				灯光模式
light_level	int				灯光级别
created_at	timestamp				创建时间，默认为当前时间

表名: opinions

字段名	数据类型	长度	主键	非空	描述
id	int		是	是	ID
phone_number	varchar(255)	255		是	电话号码
description	text			是	描述
date	timestamp				日期，默认为当前时间

表名: sensor_data

字段名	数据类型	长度	主键	非空	描述
id	int		是	是	ID
light_intensity	decimal(10,2)				光照强度
air_temperature	decimal(10,2)				空气温度
reading_time	timestamp				日期，默认为当前时间

表名: soil_data

字段名	数据类型	长度	主键	非空	描述
id	int		是	是	ID
soil_moisture	decimal(10,2)				土壤湿度
reading_time	timestamp				读取时间，默认为当前时间

表名: temperature_data

字段名	数据类型	长度	主键	非空	描述
id	int		是	是	ID
air_temperature	decimal(10,2)				空气温度
reading_time	timestamp				读取时间，默认为当前时间

表名: userinfo

字段名	数据类型	长度	主键	非空	描述
id	int		是	是	ID
nickname	varchar(255)	255			昵称
email	varchar(255)	255			电子邮件
address	varchar(255)	255			地址

表名: users

字段名	数据类型	长度	主键	非空	描述
id	int		是	是	ID
password	varchar(255)	255			密码，使用MD5加密存储
phone_number	varchar(255)	255			电话号码

4.2 安全保密设计

- 对数据库进行访问控制，仅授权用户能够访问相关表和数据
- 对敏感数据进行加密存储，如用户密码等
- 定期更新数据库软件和补丁，以保持系统安全性
- 设置防火墙和入侵检测系统，保护数据库服务器免受未经授权的访问和攻击