



数据库及软件工程课程设计报告

**课程名称：数据库及软件工程课程设计**

**姓名学号：陈威臻 1806030134、武文浩 1806030127** **郭晓彤 1803010606、 陈 辉 1806010213**

**专业班级：18级计算机专业2，3班**

**2021/7/6**

# 前言

由于现在的科学技术水平在一直进步，计算机科学的发展也变得越来越成熟，其强大的功能已为人们深刻认识，它在人类社会的各个领城有着不同凡响的作用并越发的对人类的生活提供便利。

作为计算机的一部分，数据库是一个专门用于对现实世界的各种数据进行存储的地方，而数据库管理系统风提供了强大的功能方便人们对数据的管理和存储，同时实现了数据的备份。特别是在关系数据库理论建立后数据库极大地方便了人们对数据的管理和操作。同时数据库已经成为一个国家信息技术发展水平的衡量。同时它具有着手工管理所无法比拟的优点.例如:检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成本低等。迅速的建立客户信息，避免在以往的人工填写客户信息造成的各种误差，使得客户信息能更方便的管理，节约人力资源，减少公司开销。这些优点能够极大地提高客户关系管理的效率，也是企业的科学化、正规化管理，与世界接轨的重要条件。

而软件工程与数据库课程设计的目的和意义就是为了使我们熟练地掌握软件工程中数据库设计的方法，并用它来解决实际生活中的问题，进一步理解和掌握课堂.上所学的各种数据库设计所使用的概念模型和数据库设计的流程及通用方法，掌握软件设计的基本内容和设计方法，并培养进行规范化软件设计的能力，掌握使用各种计算机资料和有关参考资料，提高进行程序设计的基本能力。

**目录**

[前言 2](#_Toc76499683)

[目录 3](#_Toc76499684)

[一、概述： 4](#_Toc76499685)

[1.项目背景： 4](#_Toc76499686)

[2.编写目的 4](#_Toc76499687)

[3.软件定义 4](#_Toc76499688)

[4.开发环境 5](#_Toc76499689)

[二、需求分析： 5](#_Toc76499690)

[1.数据流图 5](#_Toc76499691)

[（1）DFD顶层 5](#_Toc76499692)

[（2）1层 6](#_Toc76499693)

[（3）2层 6](#_Toc76499694)

[2.数据字典 7](#_Toc76499695)

[三、概念结构设计： 9](#_Toc76499696)

[四、逻辑结构设计 10](#_Toc76499697)

[1.关系表： 10](#_Toc76499698)

[2.数据表： 10](#_Toc76499699)

[五、软件功能设计 12](#_Toc76499700)

[1.系统功能结构图及其所完成的任务： 12](#_Toc76499701)

[2.项目使用方法： 12](#_Toc76499702)

[六、代码设计和界面设计 13](#_Toc76499703)

[1.代码设计： 13](#_Toc76499704)

[2.界面设计: 14](#_Toc76499705)

[七、小结 17](#_Toc76499706)

[八、相关资料及参考文献： 18](#_Toc76499707)

[九、附录：课程设计程序核心代码 19](#_Toc76499708)

[1.User.sql 19](#_Toc76499709)

[2.Project.sql 20](#_Toc76499710)

[3.Location.sql 21](#_Toc76499711)

# 一、概述：

## 1.项目背景：

我国目前的能源供应形式严峻，状况为煤炭比重过大，环境压力沉重;人均能耗远低于世界平均水平，系统效率低，原料等浪费大，能源技术也落后，产品能耗高。所以我国的能源供应面临严峻挑战:

一、化石能源可持续供应能力遭遇严重挑战；

二、能源决策国际环境复杂化。

对国外石油资源依赖度快速增大，长远来看，能源资源及其供应能力将对我国能源系统的。可持续性构成严重威胁。从能源资源、环境保护的角度，如此高的能源需求量，如果继续维持目前的能源构架是绝对不可行的。因此在大力提高高效的同时，积极开发和利用可再生能源，特别是资源量最大、分布最普遍的太阳能将是我国的必由之路。

现在光伏发电已经从解决边远地区的用电和特殊用电转向并网发电和建筑结合供电的方向发展，逐步发挥替代能源的作用，并且发展十分迅速。在近年的各种可再生能源中，并网光伏的增长速度最快，年平均增长率达到了百分之八十多。现在的科学技术水平在一直进步，计算机科学的发展也变得越来越成熟，所以有必要学习并开发改进光伏发电项目管理系统。

## 2.编写目的

通过DBMS的各种子系统来实现用户登录、退出，判断用户权限是否足够，查询、增加、更改项目的有关信息以及查看用户列表等操作，提高光伏发电项目管理员工作信息报送及反馈的工作效率，减轻管理员的劳动负担；提高信息的及时性和汇总统计信息的准确性。

## 3.软件定义

它是一种操纵和管理数据库的大型软件，是用于建立、使用和维护数据库，简称DBMS。它对数据库进行统一的管理和控制，以保证数据库的安全性和完整性。用户通过DBMS访问数据库中的数据，数据库管理员也通过DBMS进行数据库的维护工作。成分是数据库的“维护和更新”。有了这个成分，可以随时删除过时的数据元素,增加新的数据元素，因而新增加的数据可以成为数据库中可以操作的一部分。另外，这个成分可以在一定数据元素中用作修改数据区。对于每个查询者来说，可以通过自动化的方式定期地获得执行维持和更新的结果。

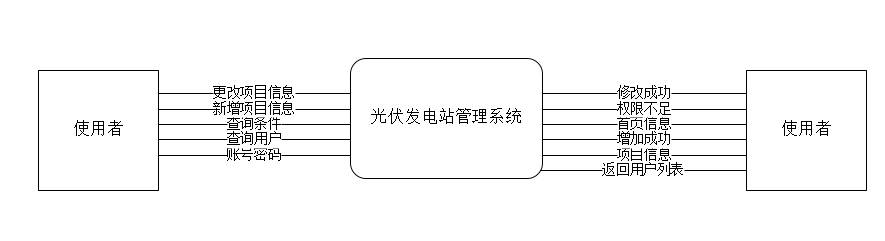
## 4.开发环境

Mysql8.0  
  
Jdk1.8  
  
Springboot

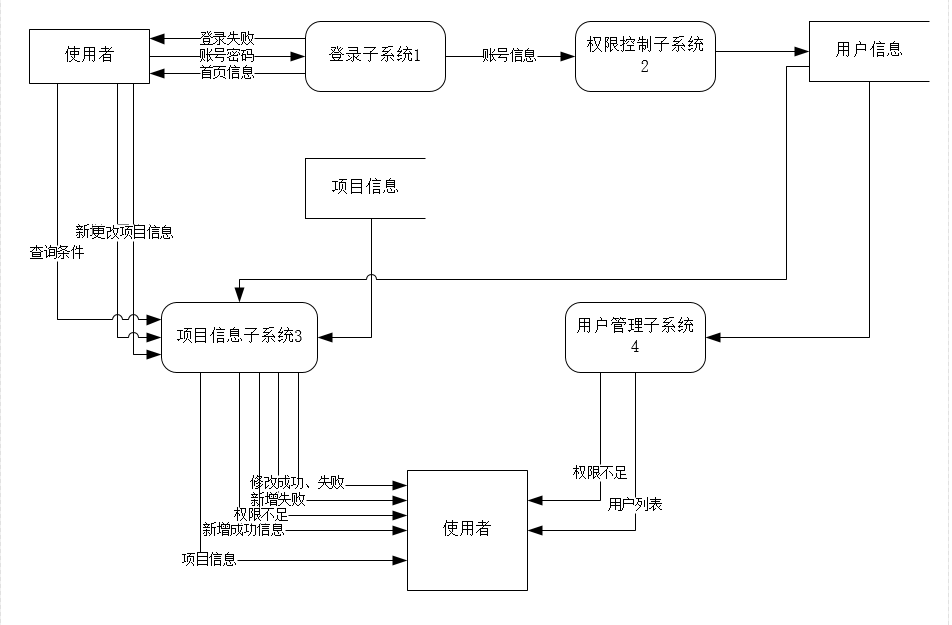
# 二、需求分析：

## 1.数据流图

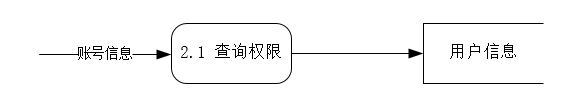
### （1）DFD顶层

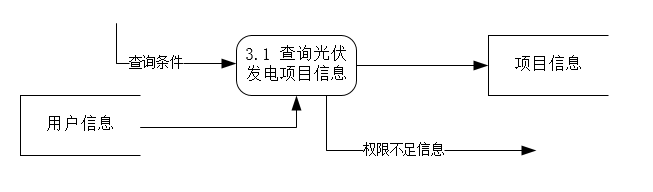


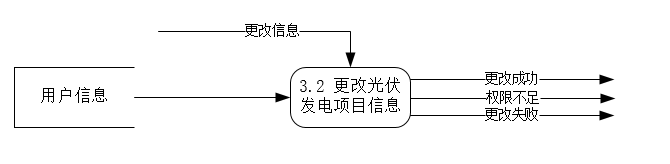
### （2）1层

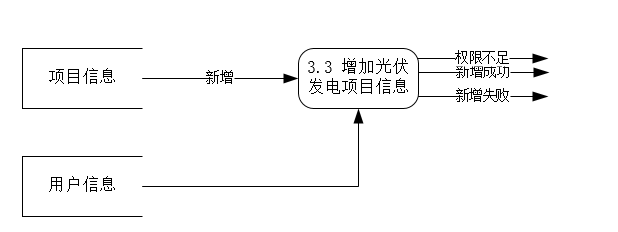


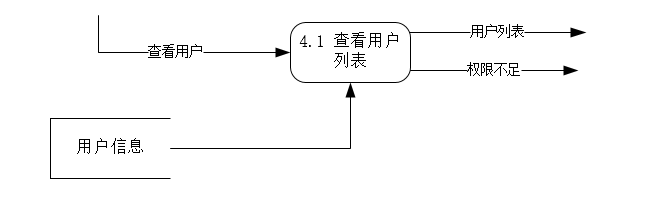
### （3）2层











## 2.数据字典

数据流：账号密码 项目信息 用户列表

存储字典：用户信息 项目信息

数据加工：查询光伏发电项目信息 更改光伏发电项目信息 增加光伏发电项目信息 查看用户列表 查询权限

数据流字典

①数据流名称：账号密码

数据结构：

01 账号密码

02 password varchar

02 account varchar

用途：使用者通过账号密码登录

来源：使用者输入

去向：登录子系统

出现时间：使用者登录时

②数据流名称：项目信息

数据结构：

01 项目信息

02 id int

02 name varchar

02 type varchar

02 stage varchar

02 organization\_light varchar

02 organization\_pressure varchar

02 date date

用途：描述项目的具体信息

来源：查询光伏发电项目信息

去向：使用者

出现时间：查询结果返回给使用者时

排列方式：按项目id升序排列

③数据流名称：用户列表

数据结构：

01 用户列表

02 id int

02 account varchar

02 type int

用途：描述用户的具体信息

来源：查看用户列表 查询权限

去向：使用者

出现时间：查询结果返回给使用者时

排列方式：按用户id升序排列

数据存储字典

①存储文件名：用户信息

数据结构：

01 用户信息

02 id int

02 account varchar

02 password varchar

02 type int

流入的数据流：

流出的数据流：用户列表

涉及的处理名：查询光伏发电项目信息 更改光伏发电项目信息 增加光伏发电项目信息 查看用户列表 查询权限

排列方式：按用户id升序排列

②存储文件名：项目信息

数据结构：

01 项目信息

02 id int

02 name varchar

02 type varchar

02 stage varchar

02 organization\_light varchar

02 organization\_pressure varchar

02 date date流入的数据流：

流入的数据流：更改的项目信息

流出的数据流：项目信息

涉及的处理名：查询光伏发电项目信息 更改光伏发电项目信息 增加光伏发电项目信息

排列方式：按项目id升序排列

数据加工描述

①加工名：查询光伏发电项目信息

输入：用户列表 查询条件

输出：项目信息

加工定义：在用户有权限时从项目信息库中查询符合条件的光伏发电项目，无权限则返回权限不足信息

激发条件：有查询命令时

②加工名：更改光伏发电项目信息

输入：用户列表 更改信息

输出：失败信息或无输出

加工定义：在用户有权限时更改项目信息库相关的光伏发电项目，无权限则返回权限不足信息，更改失败

激发条件：有更改命令时

③加工名：增加光伏发电项目信息

输入：用户列表 新增项目信息

输出：失败信息或无输出

加工定义：在用户有权限新增光伏发电项目并传入项目信息库，无权限则返回权限不足信息，新增失败

激发条件：有新增命令时

④加工名：查看用户列表

输入：查询用户列表命令 用户列表

输出：符合查询条件的用户信息或失败信息

加工定义：当用户权限符合时将用户信息库中符合查询条件的的用户列表输出给使用者，否则返回权限不足

激发条件：有查看用户列表操作时

⑤加工名：登录子系统

输入：账号密码 用户列表

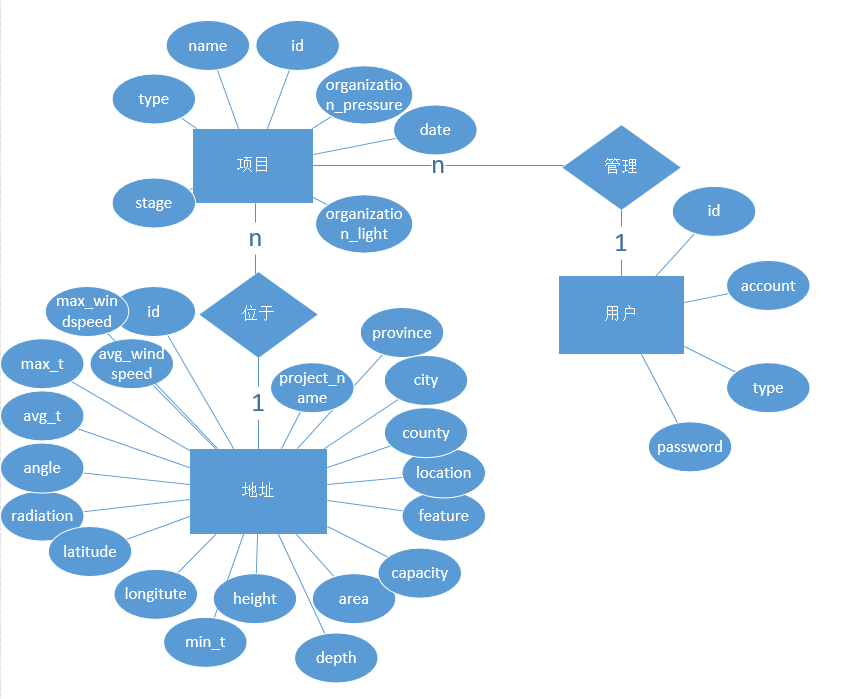
输出：首页信息或无输出

加工定义：当账号密码符合用户信息库中的某一用户的账号密码时，用户成功登录，若不符合则返回首页信息

激发条件：使用者提交账号密码后

# 三、概念结构设计：

ER模型图：



# 四、逻辑结构设计

## 1.关系表：

**Project**（id,name,type,stage.organization\_light,organization\_pressure,date，locate，manage）

**Location**(id,project\_name,province,city,couty,location,feature,capacity,area,height,longitude,latitude,radiation,angle,avg\_t,max\_t,min\_t,depth,avg\_windspeed,max\_windspeed)

**User**(id,account,password,type)

## 2.数据表：

User表

字段定义

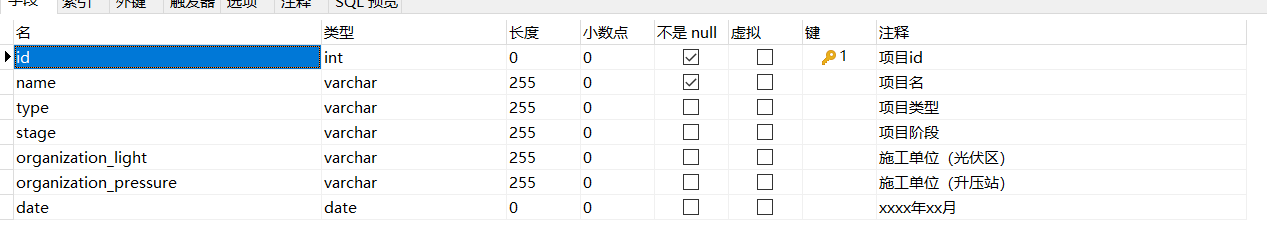


索引

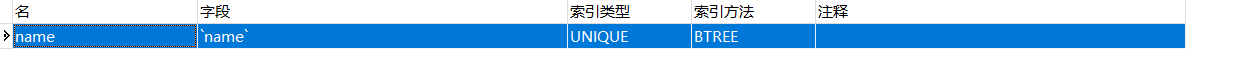


Project表

字段定义

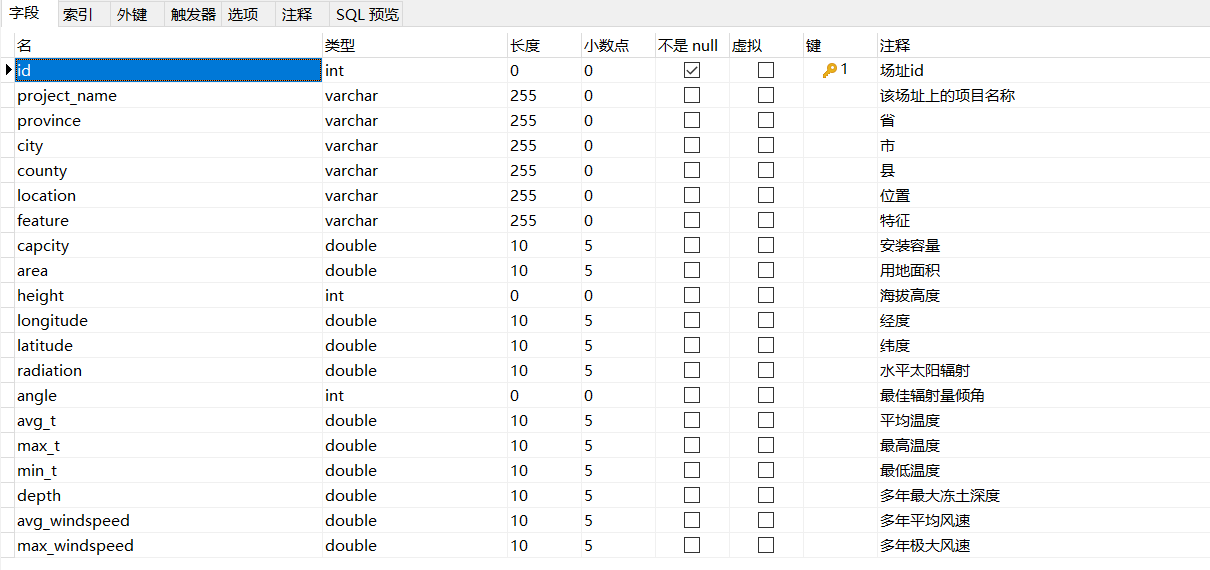


索引

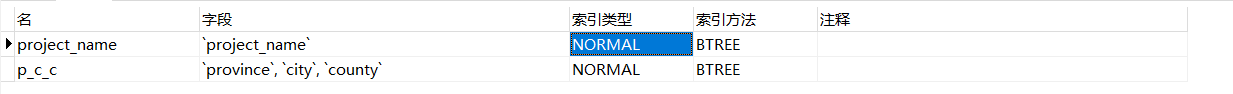


Location表

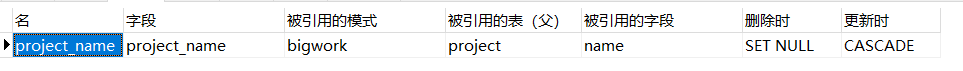
字段定义



索引

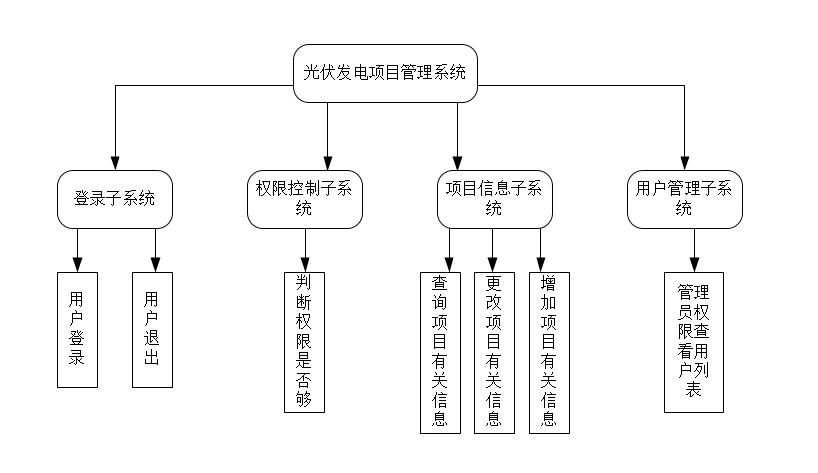


外键



# 五、软件功能设计

## 1.系统功能结构图及其所完成的任务：



## 2.项目使用方法：

（1）运行user.sql,project.sql,location.sql,在数据库中创建这三张表

（2） 修改src/main/resources/application.properties文件

**spring.datasource.url**=**jdbc:mysql://localhost:3306/bigwork?characterEncoding=utf-8&useSSL=false&serverTimezone=Hongkong  
spring.datasource.username**=**root  
spring.datasource.password**=**123456**

bigwork修改为你所创建的数据库的名称

root修改为你的数据库用户名

123456修改为你的数据库密码

# 六、代码设计和界面设计

## 1.代码设计：

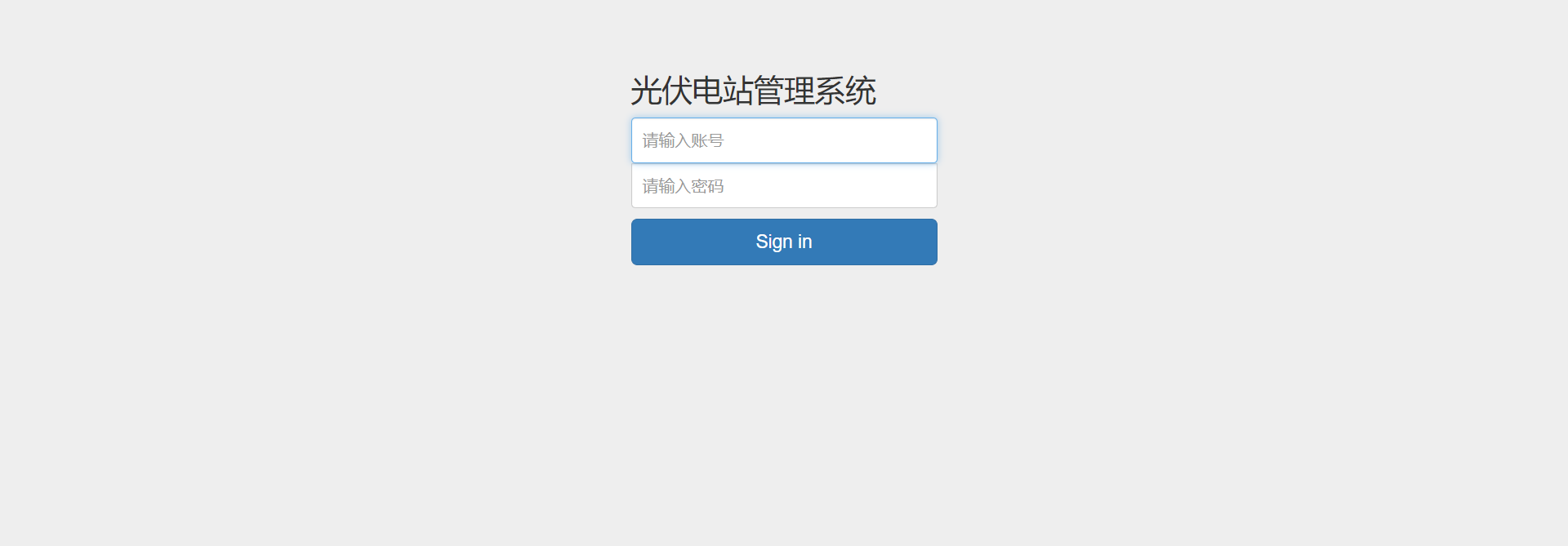
@Override  
**public boolean** preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) **throws** Exception {  
 *//每次请求发送时，从cookie中取得账号名，并查数据库返回user对象，放入hostHolder中* String account = CookieUtil.*getValue*(request, **"ticket"**);  
 **if** (account!=**null**){  
 User user = **userService**.findUserByAccount(account);  
 **hostHolder**.setUsers(user);  
 }  
 **return true**;  
}  
  
@Override  
**public void** postHandle(HttpServletRequest request, HttpServletResponse response, Object handler, ModelAndView modelAndView) **throws** Exception {  
 *//从hostHolder中取到保存的user对象，并添加到modelandview对象中传给前端* User user = **hostHolder**.getUser();  
 **if** (user!=**null**&&modelAndView!=**null**){  
 modelAndView.addObject(**"loginUser"**, user);  
 }  
}  
  
@Override  
**public void** afterCompletion(HttpServletRequest request, HttpServletResponse response, Object handler, Exception ex) **throws** Exception {  
 *//模板渲染完之后清空hostHolder* **hostHolder**.clear();  
}

新增场址

@PostMapping(value = **"/addLocation"**)  
**public** String addLocation(Model model,Location location){  
 *//判断外键projectName属性的引用完整性* String projectName = location.getProjectName();  
 **if** (projectName!=**null**&&!StringUtils.*isBlank*(projectName)){  
 Project project = **projectService**.findProjectByName(projectName);  
 **if**(project==**null**){  
 model.addAttribute(**"msg"**,**"该项目不存在"**);  
 **return "/addLocation"**;  
 }  
 **if**(**locationService**.selectLocationByProjectName(projectName)!=**null**){  
 model.addAttribute(**"msg"**,**"该项目已在其他场址"**);  
 **return "/addLocation"**;  
 }  
 }  
  
//判断输入值是否合法  
 Double longitude = location.getLongitude();  
 **if** (longitude!=**null**){  
 **if**(longitude <0|| longitude >180){  
 model.addAttribute(**"msg"**,**"经度不符合标准"**);  
 **return "/addLocation"**;  
 }  
 }  
  
 Double latitude = location.getLatitude();  
 **if** (latitude!=**null**){  
 **if** (latitude <0|| latitude >90){  
 model.addAttribute(**"msg"**,**"纬度不符合标准"**);  
 **return "/addLocation"**;  
 }  
 }  
  
 **int** row = **locationService**.insertLocation(location);  
 model.addAttribute(**"msg"**,**"插入数据成功"**);  
 **return "/addLocation"**;  
}

## 2.界面设计:

登陆界面

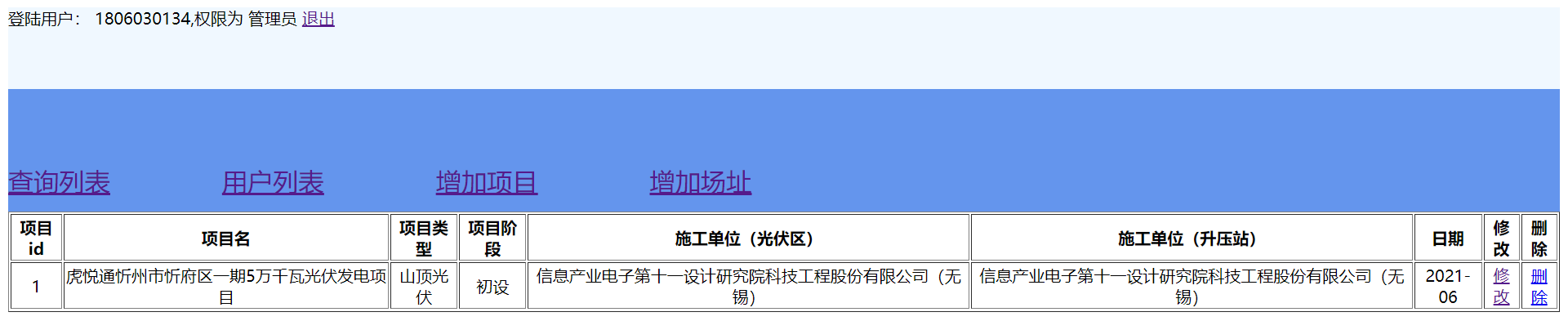


管理员：

查询场址



查询项目



新增项目



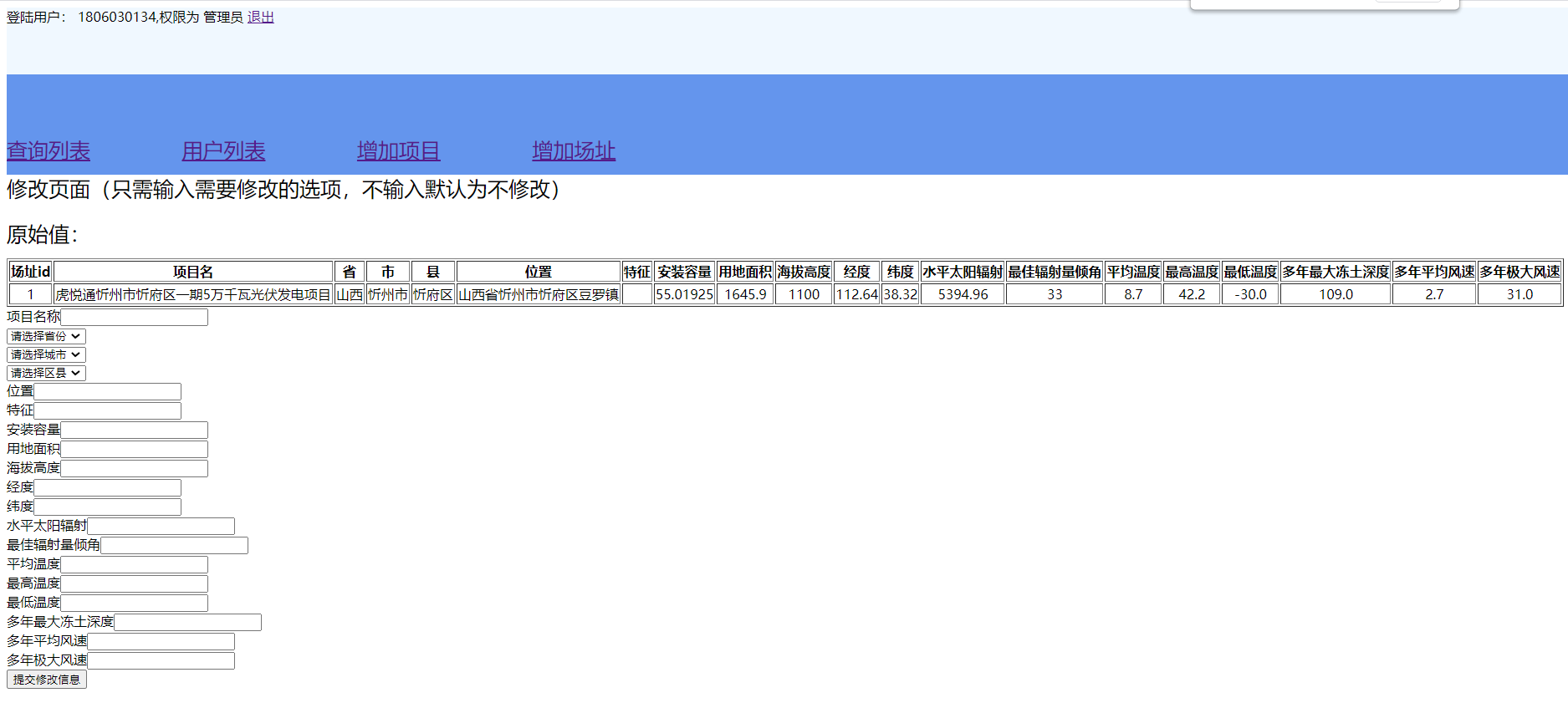
新增场址



修改项目信息

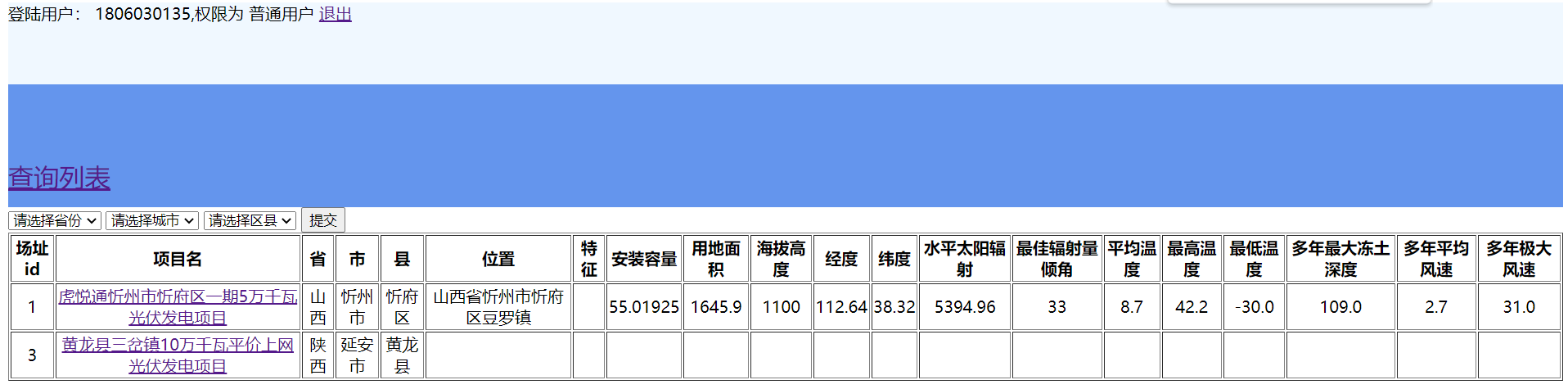


修改场址信息

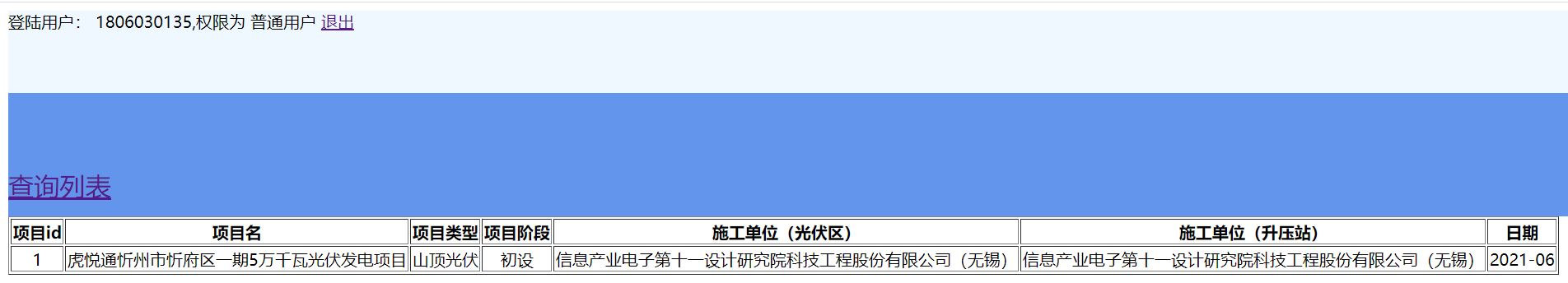


普通用户：

查询场址信息



查看项目信息：



# 七、小结

1.陈威臻：

通过做这次的光伏电站管理系统，我对于Springboot，Mybatis，Thymeleaf的操作更加熟练了也对java中的threadlocal对象有了一定的了解.唯一的遗憾就在于因为时间比较紧张以及还没有熟练掌握Springsecurity的原因，没有用上Springsecurity框架进行权限管理，而仅仅是根据权限将增删改的按钮隐藏，带来一定的风险。

2.陈辉：

我的任务主要就是写文档，了解项目需求分析及项目背景等等，同时及时发现小组实践之中的问题并指出等等，通过这次课程设计发现这其中需要的很多知识我们没有接触过，去网上查资料的时候发现有很多已经学过的东西我们没有理解到位，感觉一阵头大，不能很好的用来解决问题，这就需要我们不断的大量的实践，通过不断的在网上自学，不断地发现问题，思考问题，进而解决问题。同时也发现我前边所学到的仅仅是皮毛，还有很多需要我掌握的东西我根本不知道。在实践中我将理论课所学到的知识融会贯通，同时也学到不少很实用的东西。

实践给了两周时间，两个星期的时间非常快就过去了，这两个星期不敢说我自己获得了多少知识，但起码是了解了项目开发的部分过程。虽说上过数据库上过管理信息系统等相关的课程，但是没有亲身经历过相关的设计工作细节。这次实践确实给我们提供了一个很好的机会。

我们通过在网上学习而且在实践中应用sql语言，对数据库的创建、修改、删除方法有了一定的了解，通过导入表和删除表、更改表学会了对于表的一些操作，为了建立一个关系数据库信息管理系统，必须得经过逻辑设计、物理设计、需求分析、概念设计、系统调试、维护以及系统评价的一般过程，为毕业设计打下基础。但很多事情不是想象中的那么简单的，它涉及到的各种实体、属性、数据流程、数据处理等等。我相信自己之后再遇到相关内容的时候能更加充实自己的知识。

3. 郭晓彤：

通过本次课程设计，我们将数据库的构建与软件工程所学的内容相结合，完成了一个比较大型的系统。在本次实践中，我们将书本上的东西带入了实践，认识到一个数据库的建立过程，以及在构建时并不像教材中的那么简单，经过了许许多多的报错与配置，但也收获了配置相关程序的经验，以后可以方便进行数据库相关的工作。实践中我们也遇到了许多其他问题，包括一些功能如何实现、如何优化代码效率等等，团队是大型开发项目中必不可少的部分，本次实践也让我们进一步的学会了如何分工协作完成任务。此外，我也对这两门专业课有了更深的理解，在SQL语句嵌入、使用DDL语句定义数据等等方面运用的更加熟练，相信未来也会继续使用这些技术来创造更大的项目。

4. 武文浩：

在数据库与软件工程课程设计中，我实践了两门专业课的融合，也认识到在大项目中是不同知识的融汇，仅仅掌握一门技术是不可能实现一个模型、一个系统的。唯有把许许多多专业课学懂学透才能互相融合，让这些技术共同实现预期目标。我们在项目中运用软件工程的知识对需求进行分析以及模型设计，并对大家进行合理的分工，在测试中也用到了一些基本的边界值之类的测试方法，而数据库则偏向于构建的部分，利用JAVA结合SQL语言实现对数据库的增删查改，满足使用者需要。本次课设就像实际生活中的一些程序的缩影，只有熟练应用各种专业技能才能完善程序，满足需求，这次课设我不仅学到了专业技能，更多的则是开发素质。

# 八、相关资料及参考文献：

【1】Roger S.Pressman, Bruce R.Maxim著，郑人杰，马素霞 等译。《软件工程》（原书第八版）。北京：机械工业出版社，2016.12（2017.7重印）。

【2】《数据库系统教程》上册/王能斌。--2版。北京：电子工业出版社，2008.5

【3】唐学忠，王文。《Visual Basic程序设计教程》北京:中国电力出版社，2002。

【4】《新能源光伏发电项目施工管理研究》，徐运何，江苏林洋电力服务有限公司南京分公司，大众标准化. 2021,(05)

# 九、附录：课程设计程序核心代码

## 1.User.sql

/\*

Navicat Premium Data Transfer

Source Server : cccwz

Source Server Type : MySQL

Source Server Version : 80021

Source Host : localhost:3306

Source Schema : bigwork

Target Server Type : MySQL

Target Server Version : 80021

File Encoding : 65001

Date: 05/07/2021 20:04:20

\*/

SET NAMES utf8mb4;

SET FOREIGN\_KEY\_CHECKS = 0;

-- ----------------------------

-- Table structure for user

-- ----------------------------

DROP TABLE IF EXISTS `user`;

CREATE TABLE `user` (

`id` int(0) UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '主键，自增id',

`account` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL COMMENT '账号',

`password` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL COMMENT '密码',

`type` int(0) NOT NULL DEFAULT 1 COMMENT '权限1-用户 0-管理员',

PRIMARY KEY (`id`) USING BTREE,

UNIQUE INDEX `account`(`account`) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 5 CHARACTER SET = utf8 COLLATE = utf8\_general\_ci ROW\_FORMAT = Dynamic;

SET FOREIGN\_KEY\_CHECKS = 1;

## 2.Project.sql

/\*

Navicat Premium Data Transfer

Source Server : cccwz

Source Server Type : MySQL

Source Server Version : 80021

Source Host : localhost:3306

Source Schema : bigwork

Target Server Type : MySQL

Target Server Version : 80021

File Encoding : 65001

Date: 05/07/2021 20:04:14

\*/

SET NAMES utf8mb4;

SET FOREIGN\_KEY\_CHECKS = 0;

-- ----------------------------

-- Table structure for project

-- ----------------------------

DROP TABLE IF EXISTS `project`;

CREATE TABLE `project` (

`id` int(0) UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '项目id',

`name` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL COMMENT '项目名',

`type` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '项目类型',

`stage` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '项目阶段',

`organization\_light` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '施工单位（光伏区）',

`organization\_pressure` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '施工单位（升压站）',

`date` date NULL DEFAULT NULL COMMENT 'xxxx年xx月',

PRIMARY KEY (`id`) USING BTREE,

UNIQUE INDEX `name`(`name`) USING BTREE,

INDEX `id`(`id`, `name`) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 5 CHARACTER SET = utf8 COLLATE = utf8\_general\_ci ROW\_FORMAT = Dynamic;

SET FOREIGN\_KEY\_CHECKS = 1;

## 3.Location.sql

/\*

Navicat Premium Data Transfer

Source Server : cccwz

Source Server Type : MySQL

Source Server Version : 80021

Source Host : localhost:3306

Source Schema : bigwork

Target Server Type : MySQL

Target Server Version : 80021

File Encoding : 65001

Date: 05/07/2021 20:04:07

\*/

SET NAMES utf8mb4;

SET FOREIGN\_KEY\_CHECKS = 0;

-- ----------------------------

-- Table structure for location

-- ----------------------------

DROP TABLE IF EXISTS `location`;

CREATE TABLE `location` (

`id` int(0) NOT NULL AUTO\_INCREMENT COMMENT '场址id',

`project\_name` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '该场址上的项目名称',

`province` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '省',

`city` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '市',

`county` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '县',

`location` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '位置',

`feature` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '特征',

`capcity` double(10, 5) NULL DEFAULT NULL COMMENT '安装容量',

`area` double(10, 5) NULL DEFAULT NULL COMMENT '用地面积',

`height` int(0) NULL DEFAULT NULL COMMENT '海拔高度',

`longitude` double(10, 5) NULL DEFAULT NULL COMMENT '经度',

`latitude` double(10, 5) NULL DEFAULT NULL COMMENT '纬度',

`radiation` double(10, 5) NULL DEFAULT NULL COMMENT '水平太阳辐射',

`angle` int(0) NULL DEFAULT NULL COMMENT '最佳辐射量倾角',

`avg\_t` double(10, 5) NULL DEFAULT NULL COMMENT '平均温度',

`max\_t` double(10, 5) NULL DEFAULT NULL COMMENT '最高温度',

`min\_t` double(10, 5) NULL DEFAULT NULL COMMENT '最低温度',

`depth` double(10, 5) NULL DEFAULT NULL COMMENT '多年最大冻土深度',

`avg\_windspeed` double(10, 5) NULL DEFAULT NULL COMMENT '多年平均风速',

`max\_windspeed` double(10, 5) NULL DEFAULT NULL COMMENT '多年极大风速',

PRIMARY KEY (`id`) USING BTREE,

INDEX `project\_name`(`project\_name`) USING BTREE,

INDEX `p\_c\_c`(`province`, `city`, `county`) USING BTREE,

CONSTRAINT `project\_name` FOREIGN KEY (`project\_name`) REFERENCES `project` (`name`) ON DELETE SET NULL ON UPDATE CASCADE

) ENGINE = InnoDB AUTO\_INCREMENT = 9 CHARACTER SET = utf8 COLLATE = utf8\_general\_ci ROW\_FORMAT = Dynamic;

SET FOREIGN\_KEY\_CHECKS = 1;