数据库及安全课程设计报告

企业工程管理系统

姓 名：郑召作

学 号：20161001451

指导教师：章丽平

时 间：2019/07/10

目录

[第一章 需求分析 5](#_Toc13697994)

[§1.1信息要求 5](#_Toc13697995)

[§1.2处理要求 5](#_Toc13697996)

[§1.3安全性与完整性要求 7](#_Toc13697997)

[1.3.1 数据库的安全性 7](#_Toc13697998)

[1.3.2 数据库完整性 7](#_Toc13697999)

[第二章 概念结构设计 8](#_Toc13698000)

[§2.1 分E-R图 8](#_Toc13698001)

[§2.2 总E-R图 9](#_Toc13698002)

[第三章 逻辑结构设计 10](#_Toc13698003)

[§3.1 关系模式 10](#_Toc13698004)

[3.1.1具有简单属性的强实体集的表示： 10](#_Toc13698005)

[3.1.2弱实体集的表示 10](#_Toc13698006)

[3.1.3 联系集的表示 11](#_Toc13698007)

[§3.2 确定函数依赖： 11](#_Toc13698008)

[§3.3 数据库范式 11](#_Toc13698009)

[§3.4 定义不同视图 12](#_Toc13698010)

[§3.5 关系模式 12](#_Toc13698011)

[第四章 数据库的实施和维护 14](#_Toc13698012)

[§4.1 应用程序总框架 14](#_Toc13698013)

[§4.2 招标工程模块 15](#_Toc13698014)

[§4.2 项目零件需求添加模块 17](#_Toc13698015)

[§4.3 项目零件购买模块 18](#_Toc13698016)

[§4.4 供应商供应情况模块 22](#_Toc13698017)

[§4.5 项目阶段查看与修改模块 25](#_Toc13698018)

[第五章 结束语 27](#_Toc13698019)

[§5.1实习过程中遇到的问题以及解决方法。 27](#_Toc13698020)

[§5.2实习的体会 28](#_Toc13698021)

# 第一章 需求分析

本企业项目管理系统的目的是为了帮助公司、企业以及企业中的每个个体创造更大的价值，提高管理效益，实现经济效益最大化，从而为公司企业节约更多的人力、物力、财力。我们也希望在这个系统的开发过程中提高自已对数据管理的能力。

## §1.1信息要求

指用户需要从数据库中获得信息的内容与性质。数据库中需要存储哪些数据。

1、企业需要从外部实体获取招标项目，并将招标项目的名字以及详细描述记录在数据库中，方便做出招标决策。

2、企业需要记录当前企业所承包的项目，并实时跟踪项目进展，项目信息包括项目的名称，项目的规模，项目所需资金，项目负责人，项目所需本企业中员工数目，项目所在地点。

3、企业需要记录企业已经完成的项目，当一个项目的状态为完成时，需要把此项目加入到历史项目中。

4、系统需要从外部信息实体获取供应商信息，比如供应商名称、供应商联系方式、供应商地点等。

5、系统需要获取一个零件表单，记录所有的零件信息，表中只记录零件类型以及零件的唯一ID号。

6、系统需要记录零件的供应情况，此表连接供应商信息表以及零件信息表，除了记录这两个表的属性以外，还应包括供应此零件的颜色，可供应此零件的数目，以及开出的零件零售价。

7、系统还需要记录项目所需的零件情况，包括属性项目id，零件id，零件颜色，所需零件数目以及供应商id和供应商已提供的零件数目。

8、另外，系统需要记录企业的员工信息，包括员工的姓名、登陆该系统的密码以及在该企业中的职位。

9、还需记录每个员工参与了哪些项目，作为员工实体和项目实体之间的联系。

## §1.2处理要求

用户需要完成什么处理功能，对处理的响应时间有什么要求。（给出功能模块图）

1. 决策部门处理功能：

* 查看招标工程：从外部实体获取并记录到招标工程中，并可实现企业员工查看当前已有的招标工程。
* 项目招标处理：企业决策部门负责人可进行投标来获取工程，可简化为设置一个决策部门经理该经理拥有获取一个工程的权利。
* 删除招标工程：企业决策部门负责人在招标一个工程之后，需要将该工程从招标工程中删去，防止出现二次招标的情况。
* 添加并修改企业项目：经理拥有获取一个工程的权利后，需要将该工程的详细内容，例如：工程的名称、工程的规模、工程所需资金等，存储到相应的数据库中。

1. 项目部门：

* 查看企业工程：企业项目部门可查看已经添加到企业的工程信息；
* 制定零件清单：制定初始计划，并在技术中为某一工程添加其所需要零件的种类，以及该类零件的颜色和数量；
* 增加零件种类和数量：为某一工程动态地增加其所需零件种类和数量，并将该类信息写入零件清单中；
* 减少零件种类和数量：为某一工程动态地减少所需零件种类和数量，并将该类信息写入零件清单中；

(3) 采购部门：

* 查看零件清单：采购部门可以通过输入工程的信息，来查看相应工程中项目部门开出的零件清单；
* 查看供应商信息：采购部门可以查看以及录入系统的供应商信息，包括供应商的联系方式以及地点；
* 可查看供应商供应零件信息：供应商会供应一定数量、颜色的零件，还能看到供应商供应该零件的价格
* 采购零件：通过工程开出的零件清单，采购部门查找供应商信息决定从哪位供应商那购买零件。若符合要求的供应商拥有的零件库存可一满足要求则：若只有一位供应商供应该种零件则向该供应商购买信息入库，若有多家供应商，则需要选择价格最低的那个供应商。
* 供应组合查询：该部门的经理可以对供应商的信息进行组合查询，例如找出在北京的可供应红色F型零件的供应商。

（4）项目负责人功能：

* 查看零件清单：项目负责人可以通过输入工程的信息，以及自己的用户名以及密码，验证成功来查看相应工程中项目部门开出的零件清单；
* 查看项目信息：项目负责人有权利看到项目所有信息，并能检索所有信息以及信息的组合，以及工程花在购买零件上的总费用。
* 更改工程状况：项目负责人可以对工程当前状况进行修改
* 添加历史项目：当接收到项目部门发来的项目完成信息后由决策部门经理验收后需要保存到历史库中。

## §1.3安全性与完整性要求

### 1.3.1 数据库的安全性

数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或破坏。由于在数据库系统中大量数据集中存放，而且为许多最终用户直接共享，从而使安全性问题更为突出，本作品将实现以下数据库安全要求：

1. **数据加密：**对存储和传输的数据进行加密处理，从而使得不知道解密算法的人无法获知数据的内容。

**实现方案：**主要是通过散列函数实现用户口令的存储。哈希算法又称摘要算法、散列算法，它通过一个函数，把任意长度的数据转换成一个固定的长度的数据串，它是一种单向密码体制，无法从逆向还原出原始明文。防止黑客可以通过窃取到的用户名和口令，登录用户账号，侵害用户权益。

1. **用户标识和鉴别**：该方法由系统提供一定的方式让员工标识自己身份。每次员工要求进入系统时，由系统进行核对，通过鉴定后才提供系统的使用权。

**实现方案：**员工通过输入自静态口令以及自己的员工号，验证成功后，可获得相应访问权限，静态口令的实际就是一个口令字，口令字是一个受保护的字符串，通常用于个人身份的认证。

1. **审计：**建立审计日志，把员工对数据库的所有操作自动记录下来放人审计日志中，能够帮助企业管理层事后生成合规报告、事故追根溯源，同时加强内外部数据库网络行为记录，提高数据资产安全。

**实现方案：**MySQL可以利用审计跟踪的信息，重现导致数据库现有状况的一系列事件，找出非法存取数据的人、时间和内容等。

### 1.3.2 数据库完整性

数据库完整性（Database Integrity）是指数据库中数据在逻辑上的一致性、正确性、有效性和相容性。完整性的用于保证数据库中数据的正确性。系统在进行更新、插入或删除等操作时都要检查数据的完整性，核实其约束条件，即关系模型的完整性规则，本作品将会从接下四类完整性约束来进行阐述：

1. **实体完整性**：实体完整性指表中行的完整性。主要用于保证操作的数据（记录）非空、唯一且不重复。

**实现方案**：对于每个数据库中的每个关系，我选择能够为每个元组进行唯一标识的一个或多个属性组合，有且仅有一个主键，每一个主键值必须唯一，而且不允许为“空”（NULL）或重复。

1. **域完整性**：是指数据库表中的列必须满足某种特定的数据类型或约束。其中约束又包括取值范围、精度等规定。

**实现方案**：通过对企业中员工的每个输入进行规范约束，假如员工输入不满足输入要求，则拒绝更新数据库。

1. **参照完整性**：对于永久关系的相关表，在更新、插入或删除记录时，如果只改其一，就会影响数据的完整性，对于更新、插入或删除表间数据的完整性，统称为参照完整性。

**实现方案**：通过使用单个字段或多个字段作为外键，而外键约束就是引用字段必须在被引用字段中存在，除非引用字段部分为NULL或全部为NULL，比如在供应商供应表中，我使用供应商ID参考供应商、零件ID参考零件表。

1. **用户定义完整性**：是对数据表中字段属性的约束，包括字段的值域、字段的类型和字段的有效规则（如小数位数）等约束，是由确定关系结构时所定义的字段的属性决定的。

**实现方案**：通过使用CHECK 约束，来将约束表中某一个或者某些列中可接受的数据值或者数据格式，比如该企业在一个工程完成前可以接收第二个工程但要考虑到该企业是否有足够的人手可同时进行若干工程的实施。

# 第二章 概念结构设计

## §2.1 分E-R图

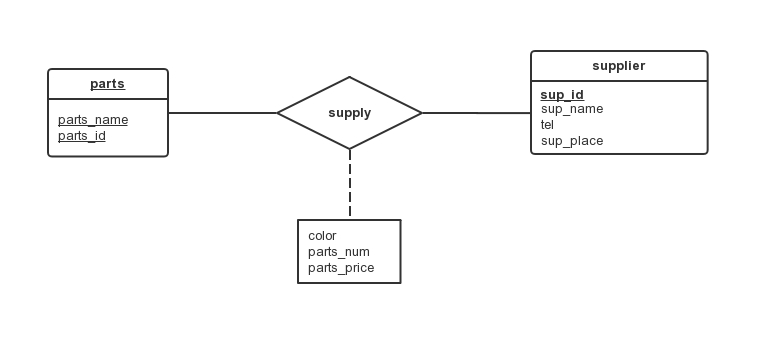


图1-1 零件供应ER分图

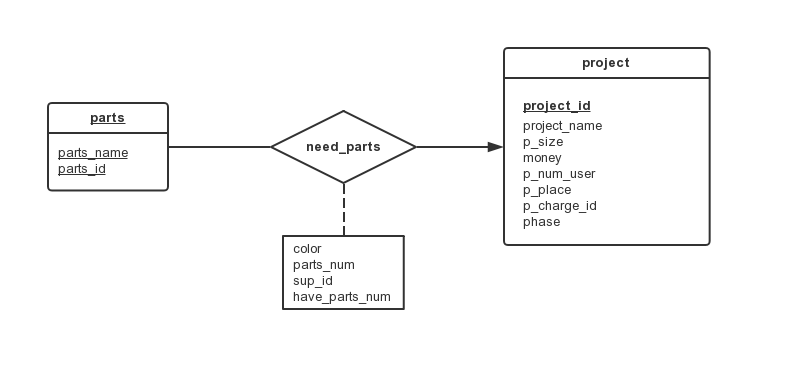


图1-2 项目零件供需ER分图

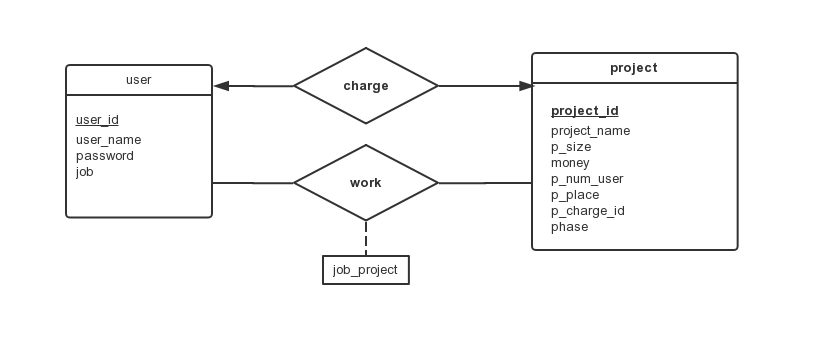


图1-3 用户工程ER分图

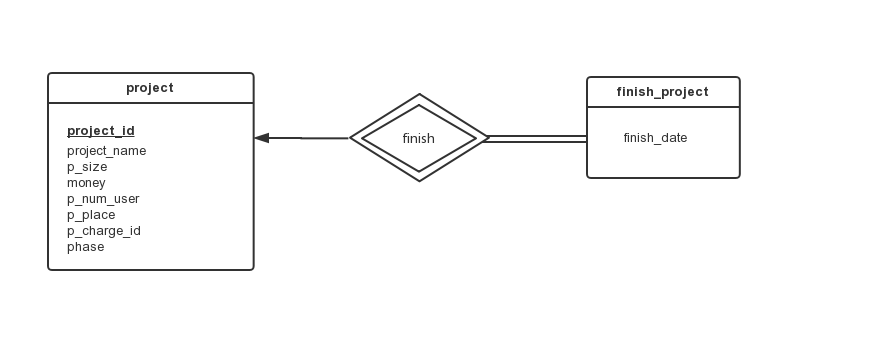


图1-4 包含历史工程弱实体集的工程ER分图

## §2.2 总E-R图

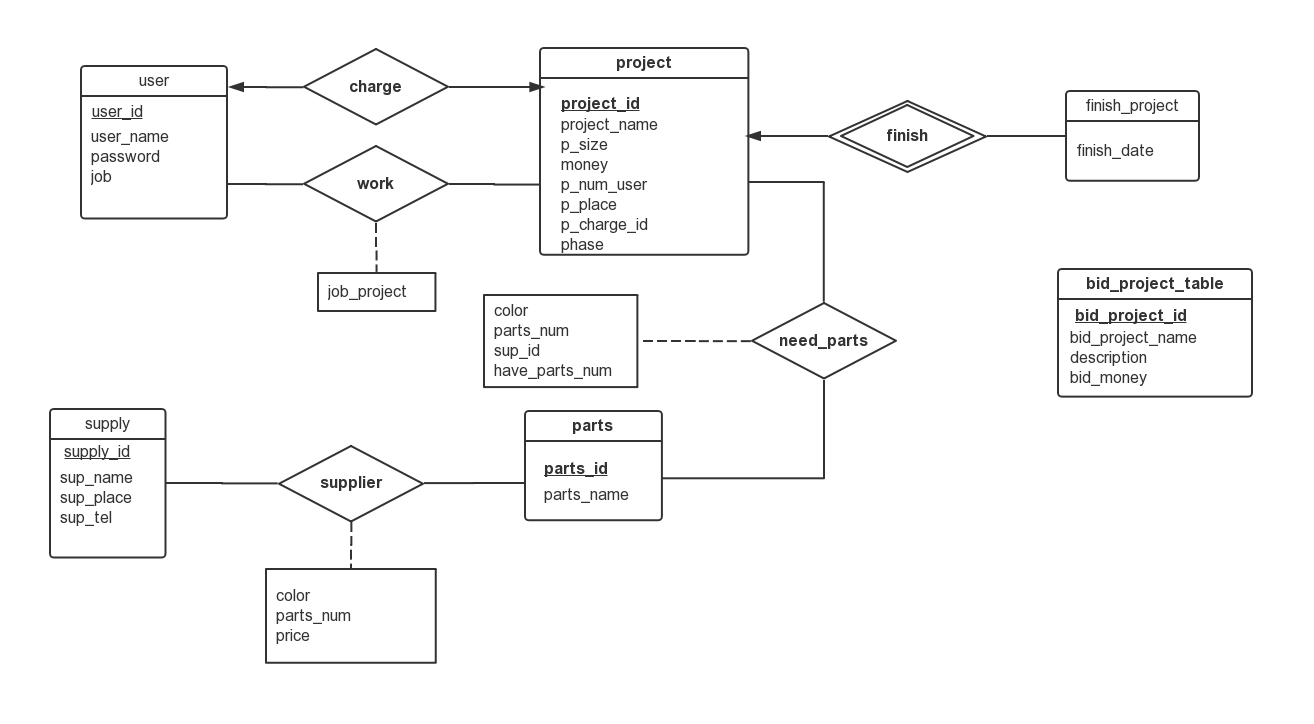


图1-5 总体E-R图

# 第三章 逻辑结构设计

## §3.1 关系模式

我将上文得到的符合ER数据库模式的数据库表示为一些关系模式的集合。在接下来的设计过程中，我将每个实体集及以及对于每个联系集，都设计了唯一的关系模式，关系模式名即为对应的实体集或联系集的名称。

### 3.1.1具有简单属性的强实体集的表示：

对于从强实体集转化而来的模式，强实体集的主码就是生成的模式的主码。这是因为每个元组都对应于实体集中的一个特定实体。

例如，考虑图1-5中的ER图的实体集supplier。该实体集有四个属性：sup\_id、sup\_name、sup\_place和sup\_place。我使用名为supplier的模式来表示这个实体集，它有三个属性：

supplier(sup\_id, sup\_name,sup\_place,sup\_place)

由于供应商ID是实体集的主码，因此它也是关系模式的主码，对于图1-5，从这些强实体集转换而来的模式为：

parts(parts\_id, parts\_name)

user( user\_id，user\_name，password，job)

bid\_project\_table(bid\_project\_id, bid\_name, bid\_moey, description)

project(project\_id, project \_name,p\_size, p\_money, p\_num\_user,p\_place, p\_charge\_id, phase)

### 3.1.2弱实体集的表示

对于从弱实体集转换而来的模式，该模式的主码由其所以来的强实体集的主码与弱实体集的分辨码组成。除了创建主码之外，还要在关系上建立外码约束，该约束指明强实体集关系的主码。外码约束保证表示弱实体的每个元组都有个表示相应强实体的元组与之对应。

以图1-5所示ER图中的弱实体集history\_project为例说明。该实体集有属性finish\_date，实体集所依赖的实体集project 的主码是project\_id。因此，用来表示history\_project的模式具有下面的属性:

history\_project( project\_id , finish\_date)

该主码由实体集project的主码和history\_project的分辨符组成。

### 3.1.3 联系集的表示

联系集主码的构成依赖于同联系集相关联的属性集合，对于每个联系集，从所有相关实体集中取所有主码属性能够用来标识一个指定元组。其他属性来自与联系集的属性。

图1-5所示ER图中所有联系集的关系模式如下：

supply(parts\_id, sup\_id, color, parts\_num, parts\_price)

work(user\_id, project\_id, job)

needs\_parts(project\_id, parts\_id, color, parts\_num, sup\_id, have\_parts\_num)

## §3.2 确定函数依赖：

函数依赖是某个属性集决定另一个属性集时，称另一属性集依赖于该属性集。它表征一个属性或属性集合的值对另一个属性或属性集合的值的依赖性。

1. supplier关系模式的函数依赖：

sup\_id →  sup\_name , sup\_place , sup\_tel

1. parts关系模式的函数依赖：

parts\_id → parts\_name

1. user关系模式的函数依赖：

user\_id → user\_name，password，job

1. bid\_project\_table关系模式的函数依赖：

bid\_project\_id → bid\_name, bid\_moey, description

1. history\_project关系模式的函数依赖：

project\_id → finish\_date

1. supply关系模式的函数依赖：

parts\_id, sup\_id, color → parts\_num, parts\_price

1. work关系模式的函数依赖：

user\_id, project\_id →job\_project

1. needs\_parts关系模式的函数依赖：

project\_id, parts\_id, color → parts\_num, sup\_id, have\_parts\_num

1. project关系模式的函数依赖：

project\_id → project \_name,p\_size, p\_money, p\_num\_user,p\_place, p\_charge\_id, phase

## §3.3 数据库范式

数据库能达到较满意的范式之一是BCNF，它消除所有基于函数依赖能发现的冗余，值得注意的是，一个数据库设计属于BCNF的条件是，构成该设计的关系模式集中的每个模式都属于BCNF。

通过对模式计算，由于我设计的所有关系模式都只含主码作为被依赖对象的函数依赖。所以每个模式都是属于BCNF的，则我设计的数据库是满足BCNF的。

## §3.4 定义不同视图

通过将概念模型转化为全局逻辑模型后，还应根据局部应用需求的特点，设计不同的视图，对不同级别的用户定义不同的视图，以保证系统安全性。

比如，在本数据库中，我们对project和need\_parts上定义了两个不同的视图，第一个是为采购部门所实现的视图，通过该视图，采购部门经理仅能访问到该工程所需的所有零件清单信息；第一个是为项目负责人设计的视图，通过该视图可以访问到两个关系进行自然连接后的所有属性，包括工程的进展、人员分配、工程规模等等，这些信息是通过视图一访问不到的。

## §3.5 关系模式

各表关系模式如下：

招标项目（bid\_project\_table）：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段描述 | 备注 |
| **bid\_project\_id** | int(6) | 招标项目ID | primary key |
| **bid\_name** | char(10) | 招标项目名字 |  |
| **bid\_moey** | int(11) | 招标项目开价 |  |
| **description** | char(20) | 招标项目详细描述 |  |

企业项目表（project）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段描述 | 备注 |
| **project\_id** | int(6) | 项目ID | primary key |
| **project \_name** | char(20) | 项目名字 |  |
| **p\_size** | char(5) | 项目规模 |  |
| **p\_money** | int(10) | 项目金额 |  |
| **p\_num\_user** | int(10) | 项目所需人数 |  |
| **p\_place** | char(20) | 项目地点 |  |
| **p\_charge\_id** | int(10) | 项目负责人ID | 员工表表外键 |
| **phase** | char(20) | 项目当前阶段 |  |

企业历史项目表（history\_project）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段描述 | 备注 |
| **project\_id** | int(6) | 历史项目ID | primary key |
| **finish\_date** | date | 历史项目完成时间 |  |

零件信息表（parts）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段描述 | 备注 |
| **parts\_id** | char(6) | 零件ID | primary key |
| **parts\_name** | char(20) | 零件名称 |  |

供应商信息表（supplier）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段描述 | 备注 |
| **sup\_id** | int(6) | 供应商ID | primary key |
| **sup\_name** | char(20) | 供应商名字 |  |
| **sup\_place** | char(20) | 供应商地点 |  |
| **sup\_tel** | char(20) | 供应商联系电话 |  |

供应信息表（supply）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段描述 | 备注 |
| **parts\_id** | char(6) | 零件ID | primary key 零件信息表外键 |
| **sup\_id** | int(6) | 供应商ID | primary key供应商信息表外键 |
| **Color** | char(10) | 零件颜色 | primary key |
| **parts\_num** | int(6) | 零件供应数量 |  |
| **parts\_price** | int(10) | 零件零售价 | Not null |

员工表（user）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段描述 | 备注 |
| **user\_id** | int(6) | 员工号 | primary key |
| **user\_name** | char(20) | 登陆用户名 |  |
| **password** | varchar(20) | 登陆密码 |  |
| **job** | varchar(20) | 职位 |  |

工作表（work）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段描述 | 备注 |
| **user\_id** | int(6) | 员工号 | primary key员工表外键 |
| **project\_id** | int(6) | 历史项目ID | primary key项目信息表外键 |
| **job** | varchar(20) | 职位 |  |

项目需要零件表（needs\_parts）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段描述 | 备注 |
| **project\_id** | int(6) | 项目ID | primary key 项目信息表外键 |
| **parts\_id** | char(10) | 零件ID | primary key零件信息表外键 |
| **color** | char(10) | 零件颜色 | primary key |
| **parts\_num** | int(6) | 需要零件数量 |  |
| **sup\_id** | int(6) | 供应商ID | primary key 供应商信息表外键 |
| **have\_parts\_num** | int(11) | 已收到零件 |  |

# 第四章 数据库的实施和维护

## §4.1 应用程序总框架

本数据库应用是发生在应用程序内，用户与数据库之间的交互都是通过该应用程序间接发生，由于Web已成为最广泛使用的数据库用户界面，因此本应用程序是基于Web实现的企业工程管理系统，详细的开发环境与语言如下：

* 开发语言：HTML、PHP、JavaScript
* 渲染语言：CSS
* Linux面板：宝塔
* 开发环境：CentOS操作系统、Nginx反向代理
* 数据库：MySQL关系型数据库

接下来我将按招标工程、项目零件需求添加、项目零件购买、供应商供应情况、项目阶段查看与修改五大模块分析所编写的应用程序“企业工程管理系统”，系统结构图如下：

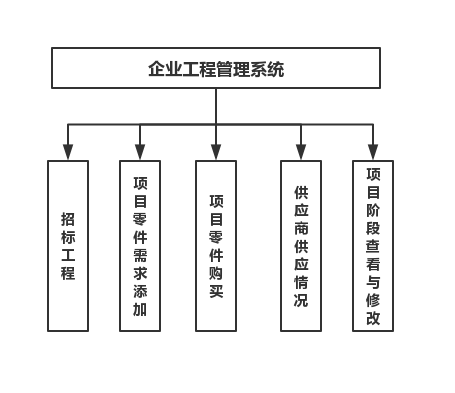


图4-1 系统模块图

## §4.2 招标工程模块

功能描述：该模块进行投标获取工程。可简化为设置一个决策部门经理该经理拥有获取一个工程的权利并需要将该工程的详细内容存储到相应的数据库中，并给工程部门发送信息要求工程部门开始实施工程。

详细思路：待招标的工程应当通过其他方式最终存储在数据库中，因此无需考虑这部分数据的录入，只需实现决策部门能够查看待招标工程信息，需要考虑的是决策部门经理需要将完成招标的工程从招标工程bid\_project\_table表中删去，通过部门经理进行信息添加，最终将结果录入到企业工程project表中，添加的信息包括项目规模、分配给该项目的人数、负责人ID、项目地点等等。招标完成后，显示工程信息到屏幕上。

实现过程中，我需要从前端获取到决策部门经理的输入，即需要招标工程的ID以及编辑工程信息，接受到决策部门经理的输入后，JavaScript将其提交给后端PHP，后端通过传入的参数编写数据库语言，然后在后端执行数据库语言来实现bid\_project-\_table表和project表的更新与检索。

文件介绍：

1、decision.php：企业投标工程查看、企业工程投标决策

2、bid\_succ.php：工程投标后信息修改

3、bid\_finish.php：企业所有工程信息查看

SQL语句：

1、查看所有招标工程：

select \* from bid \_project\_table

2、通过工程ID查看招标工程：

select \* from bid\_project\_table where bid\_project\_id = $bid\_project\_id

3、删除招标工程：

delete from bid\_project\_table where bid\_project\_id = $bid\_project\_id

4、添加企业工程：

insert into project values ( $p\_id , $p\_name , $p\_size , $p\_money , $p\_num\_user , $p\_place , $p\_charge\_id )

5、查看所有企业工程：

select \* from project

实验截图：



图4-2 招标信息查看



图4-3 工程招标

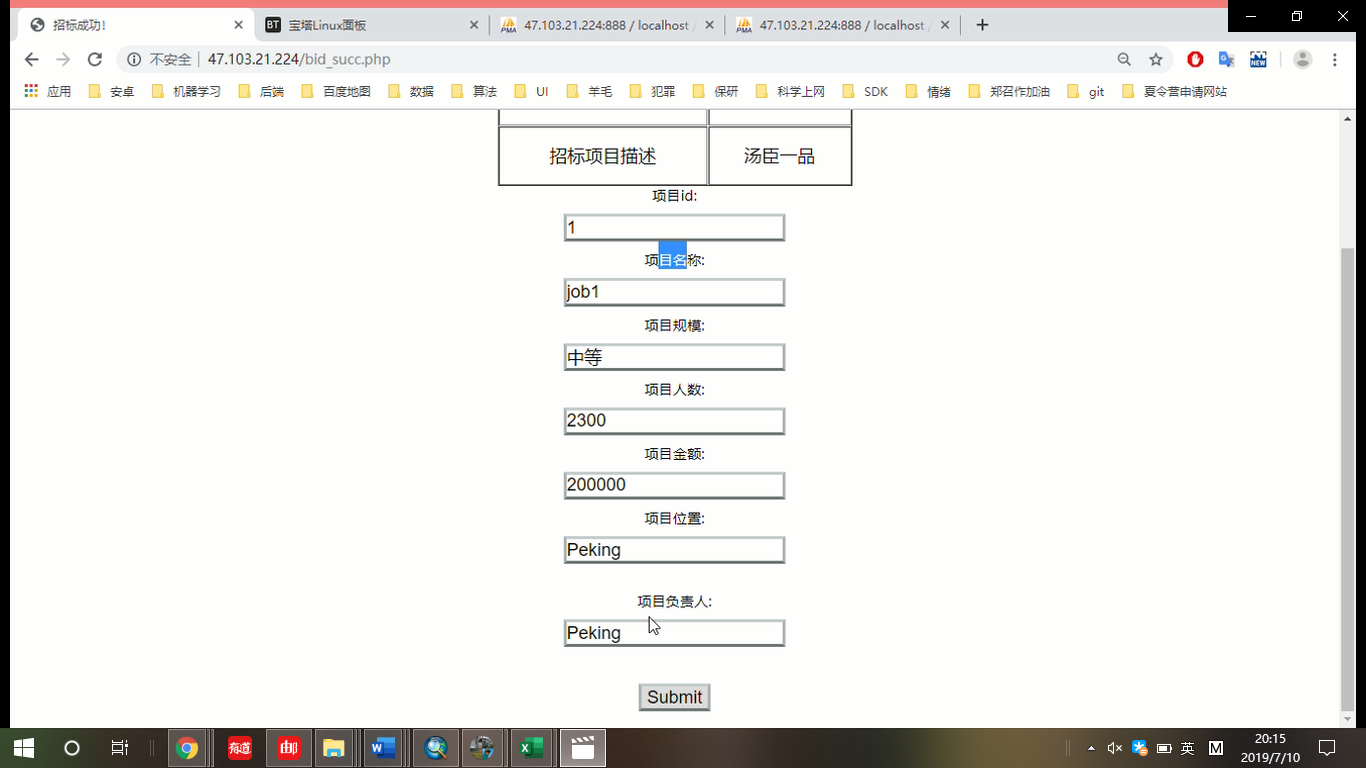


图4-4 项目信息提交

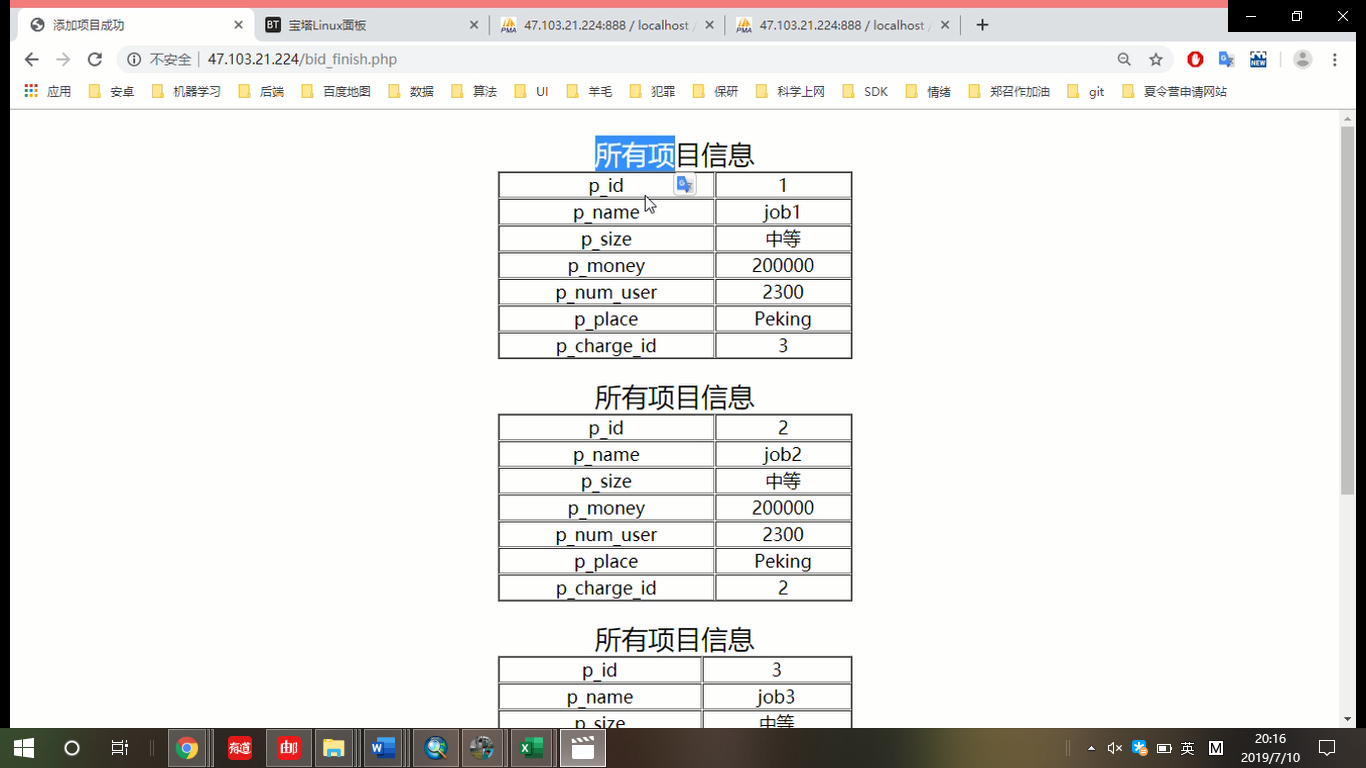


图4-5 项目信息查看

## §4.2 项目零件需求添加模块

功能描述：当项目部门收到决策部门的项目清单时候，开始项目的准备工作，项目部门经理将为第一阶段工程开出使用的零件清单，并将该清单发送给采购部门。采购清单中需要录入工程号、零件型号、零件颜色、需要零件数目等信息。在工程实施过程中项目经理有权利增加和减少零件的种类和数目。

详细思路：我需要从前端获取到项目部门经理的输入，包括需要添加零件的工程ID、零件ID、零件数量、零件颜色，然后通过POST发送Http请求，请求发送给后端PHP服务器，通过POST传入的参数编写零件需求添加操纵语言，然后在后端执行数据库语言来实现need\_parts表的更新与检索。

文件介绍：

1、scedule.html：零件信息添加提交

2、add\_parts.php：零件信息添加完成

SQL语句：

1、添加零件需求：

insert into need\_parts values( $\_POST['p\_id'] , $\_POST['parts\_id'] , $\_POST['color'] , $\_POST['parts\_nums'] , null , 0);

实验截图：

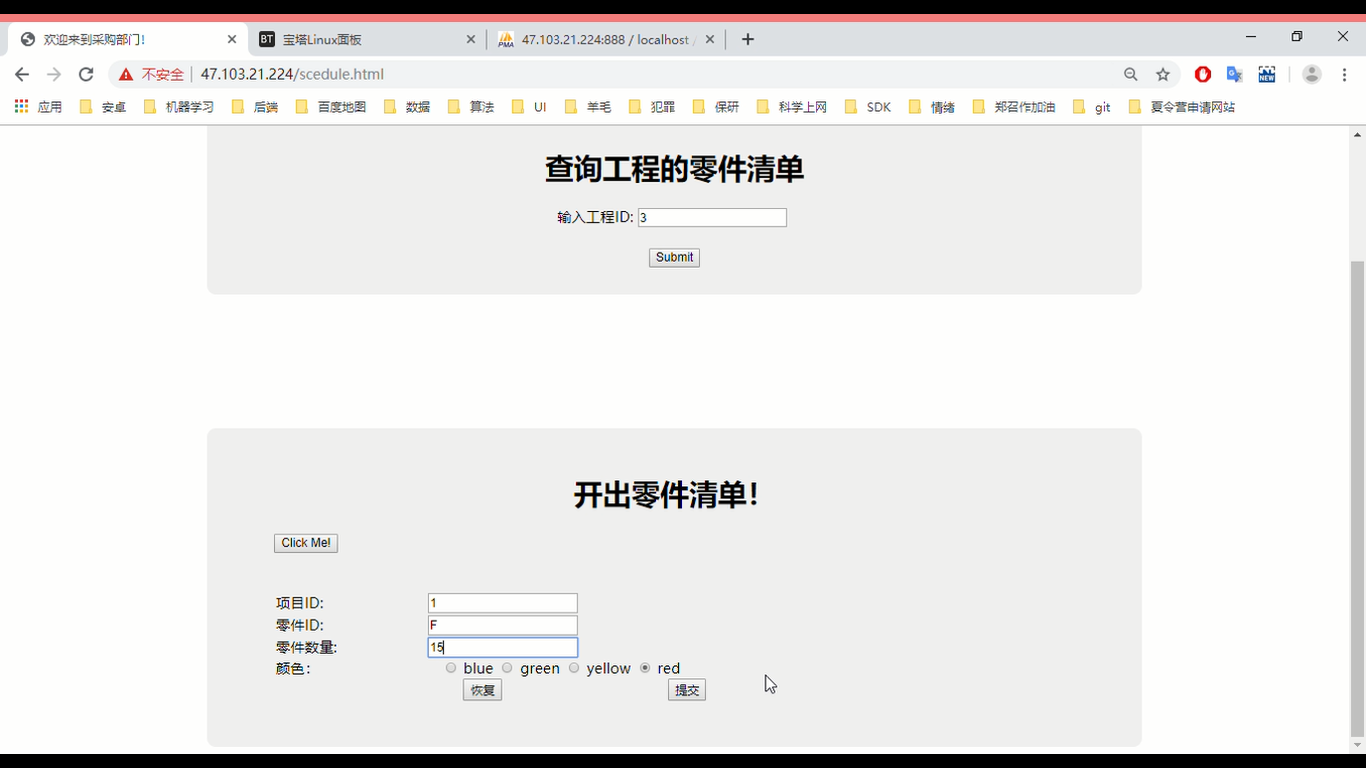


图4-6 零件需求添加



图4-7 零件需求添加成功

## §4.3 项目零件购买模块

功能描述：主要负责零件的采购。收到项目管理部门的采购信息后开始零件的采购查找供应商信息决定从哪位供应商那购买零件。若符合要求的供应商拥有的零件库存可一满足要求则：若只有一位供应商供应该种零件则向该供应商购买信息入库，若有多家供应商，则需要进行选择，选择价格最低的供应商提供的零件。

详细思路：首先采购部门需要输入企业工程ID，查看当前工程所需且还未进行采购的零件，对于每个零件，会显示出它所需的颜色、所需零件数量，并在每个零件下方都会有一个输入框来对采购部门开放，部门经理可以按照所需信息进行输入，也可以手动输入，增加或减少零件数量，最后进行购买，服务器会自动为零件选择最合适的供应商并自动完成购买。

实现过程中，我需要从前端获取到采购部门经理的输入，即需要进行采购的企业工程ID，接着通过POST发送Http请求，并返回零件信息结果到屏幕上，然后部门经理修改供需信息或直接按照零件清单发起零件购买请求，购买成功后，刷新页面，可以发现，已经购买了由供应商开出的价格最低的零件，在此过程中将其提交给后端，后端通过传入的参数编写数据库语言，然后在后端执行数据库语言来实现supply表、parts、supplier和need\_parts表的更新与检索。

文件介绍：

1、scedule.html：工程零件清单查询

2、need\_parts.php：修改供需信息或直接按照零件清单发起零件购买请求

3、buy\_parts.php：采购部门按要求进行零件采购

SQL语句：

1、查看企业工程所需零件：

select \* from need\_parts where project\_id = ".$p\_id

2、找出可供应某零件并满足要求的供应商信息：

select \*

from supplier natural join supply

where parts\_id = $parts\_id and color = $color

and parts\_num>=$parts\_num

3、找出可供应某零件并满足要求、且开出价格最低的供应商信息：

select \*

from supplier natural join supply

where parts\_id = $parts\_id and color = $color and parts\_num>= $parts\_num and parts\_price <= all

( select parts\_price

from supply

where parts\_id = $parts\_id and color = $color and parts\_num>=$parts\_num)

实验截图：

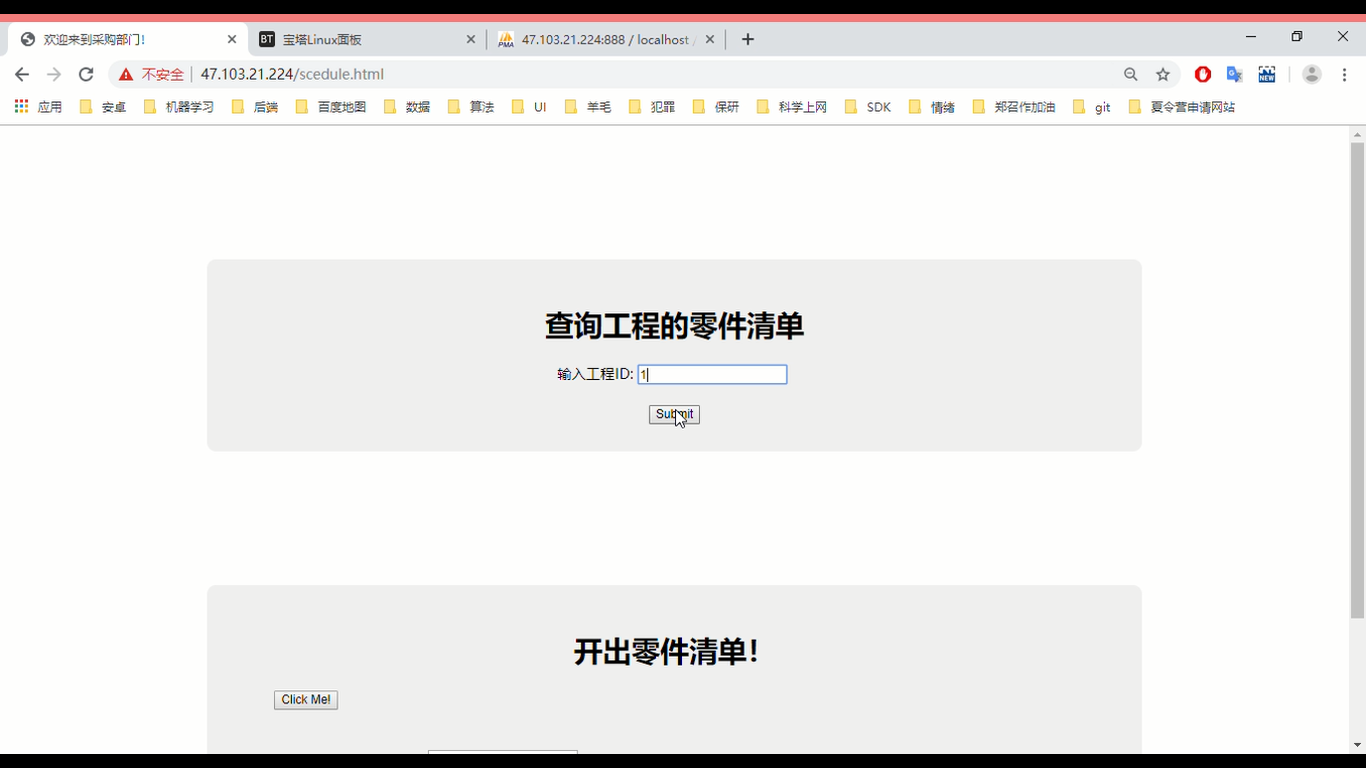


图4-8 输入工程ID界面

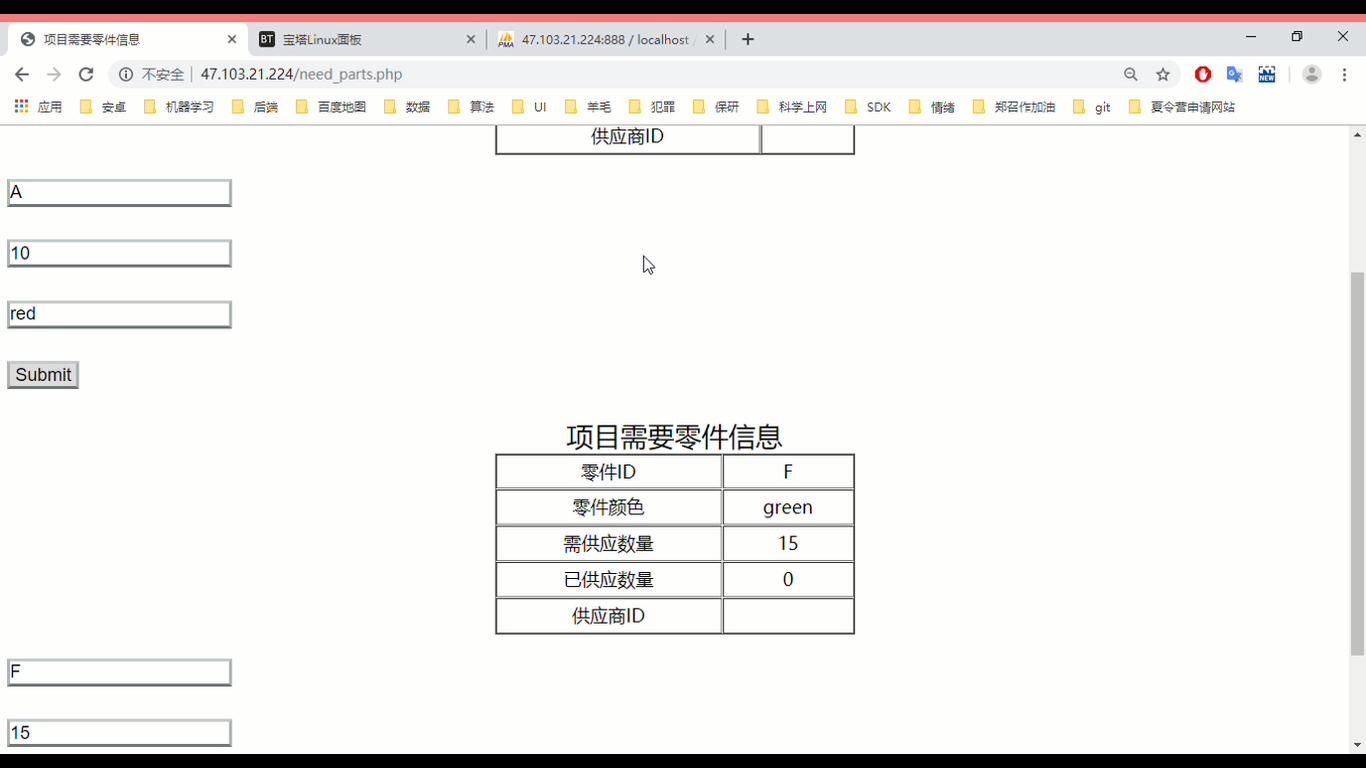


图4-9 项目零件清单界面

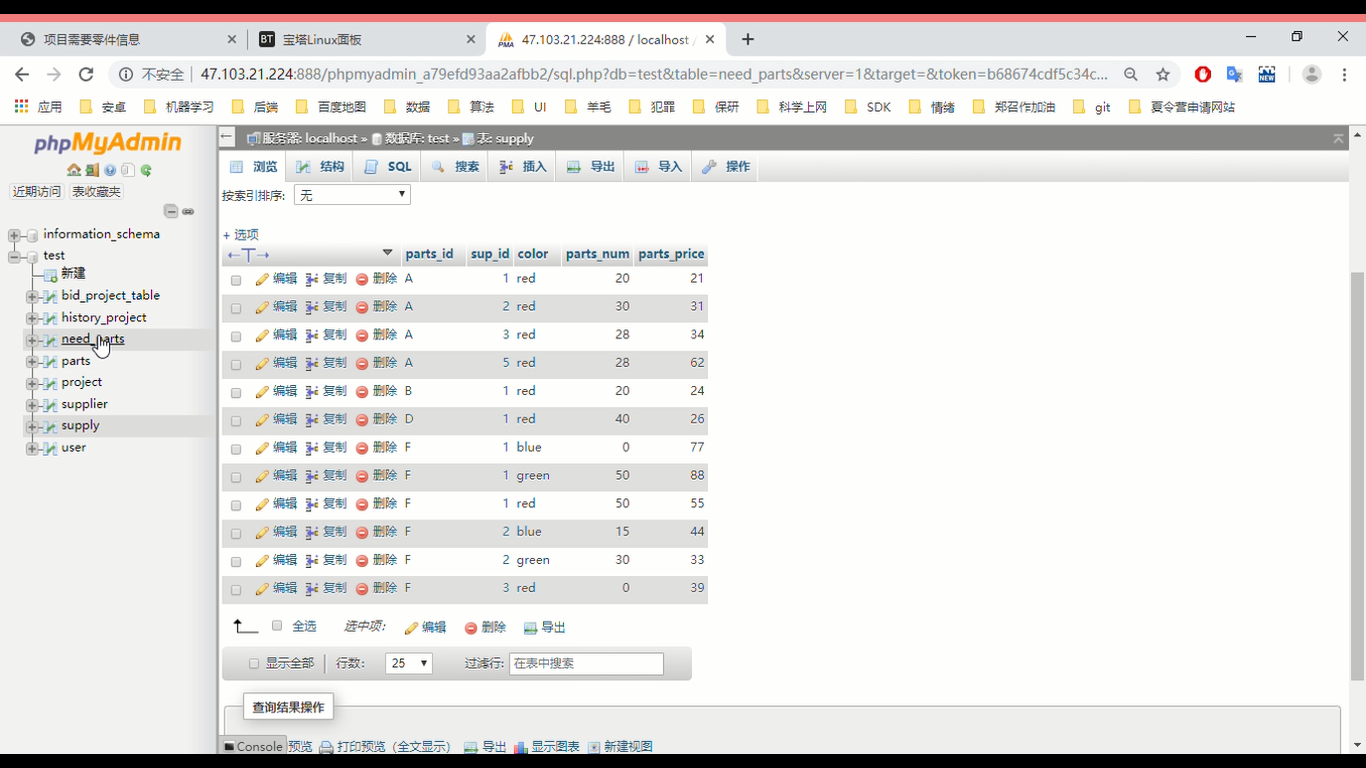


图4-10 供应商零件供应情况数据



图4-11 零件采购界面1

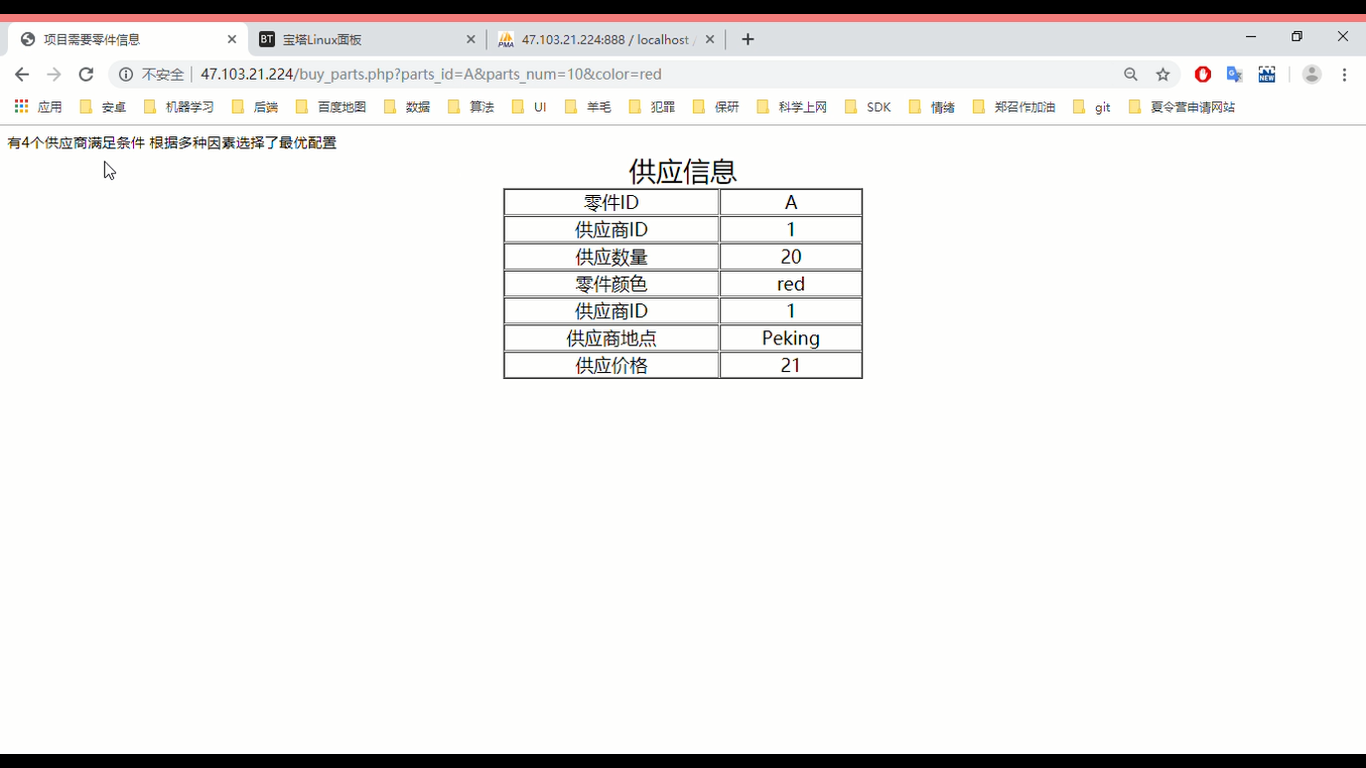


图4-12 零件供应采购成功1

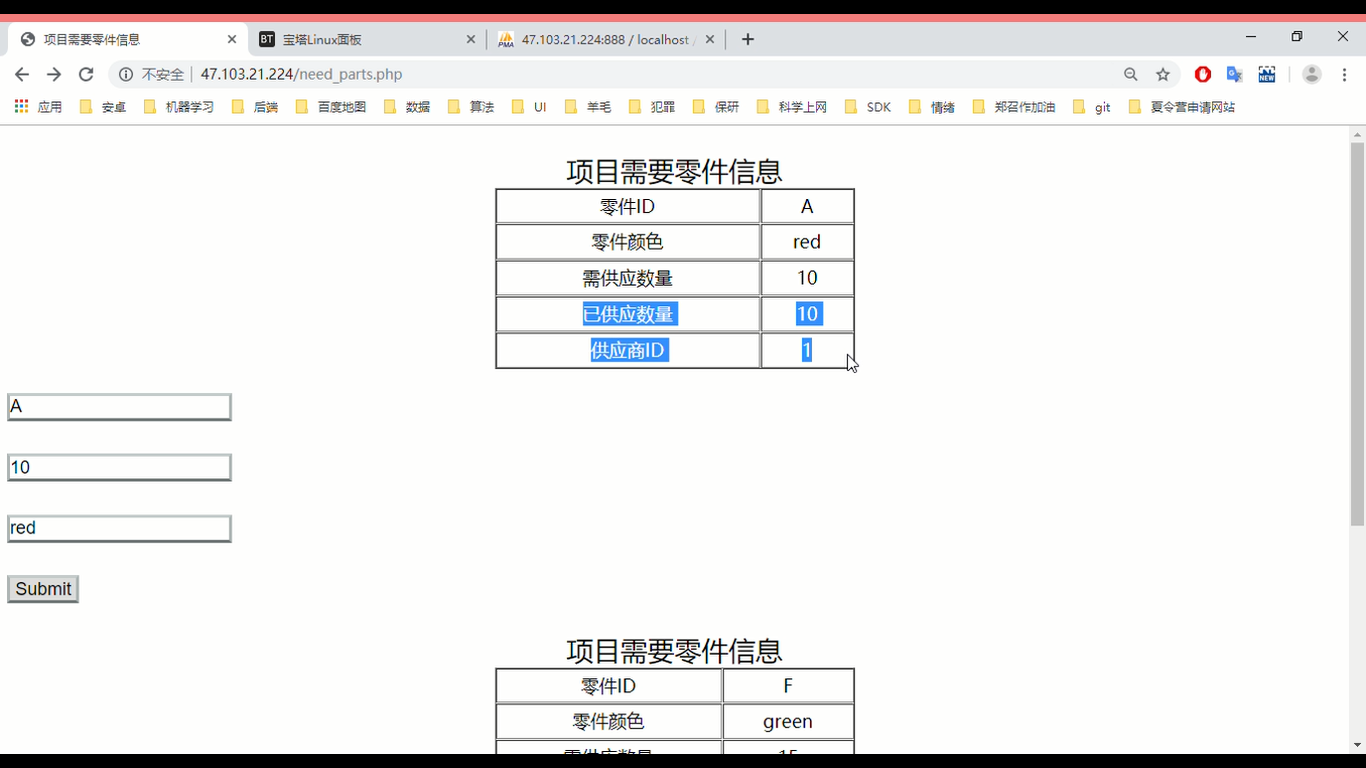


图4-13 返回零件采购界面1

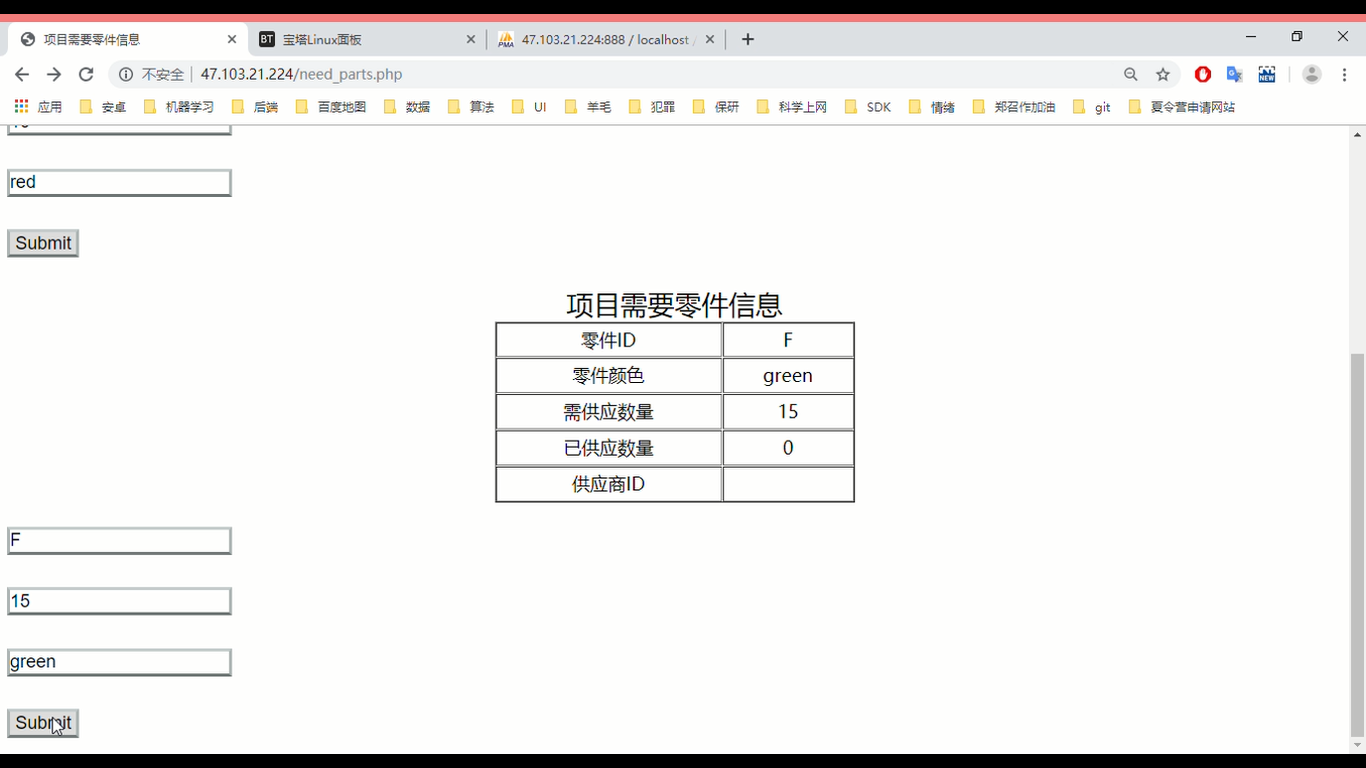


图4-14 零件采购界面2

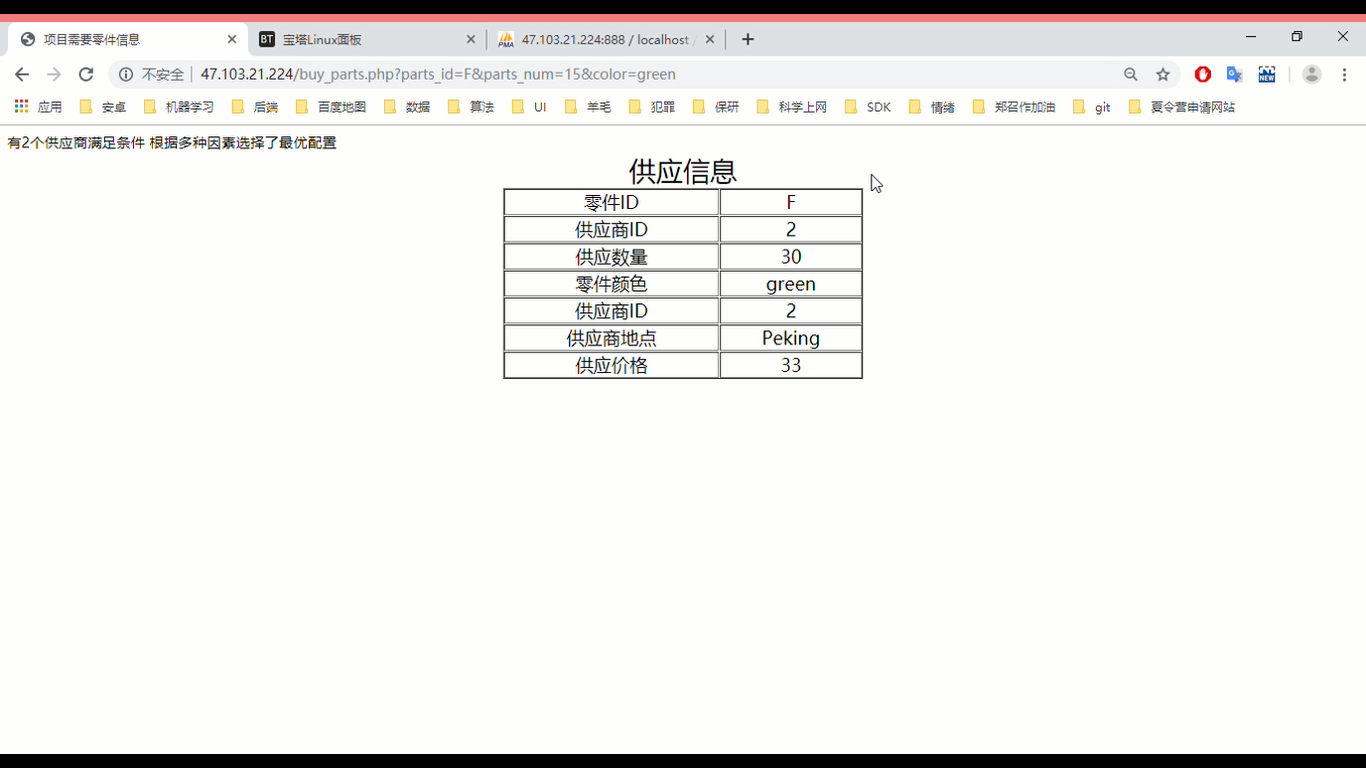


图4-15 零件供应采购成功2

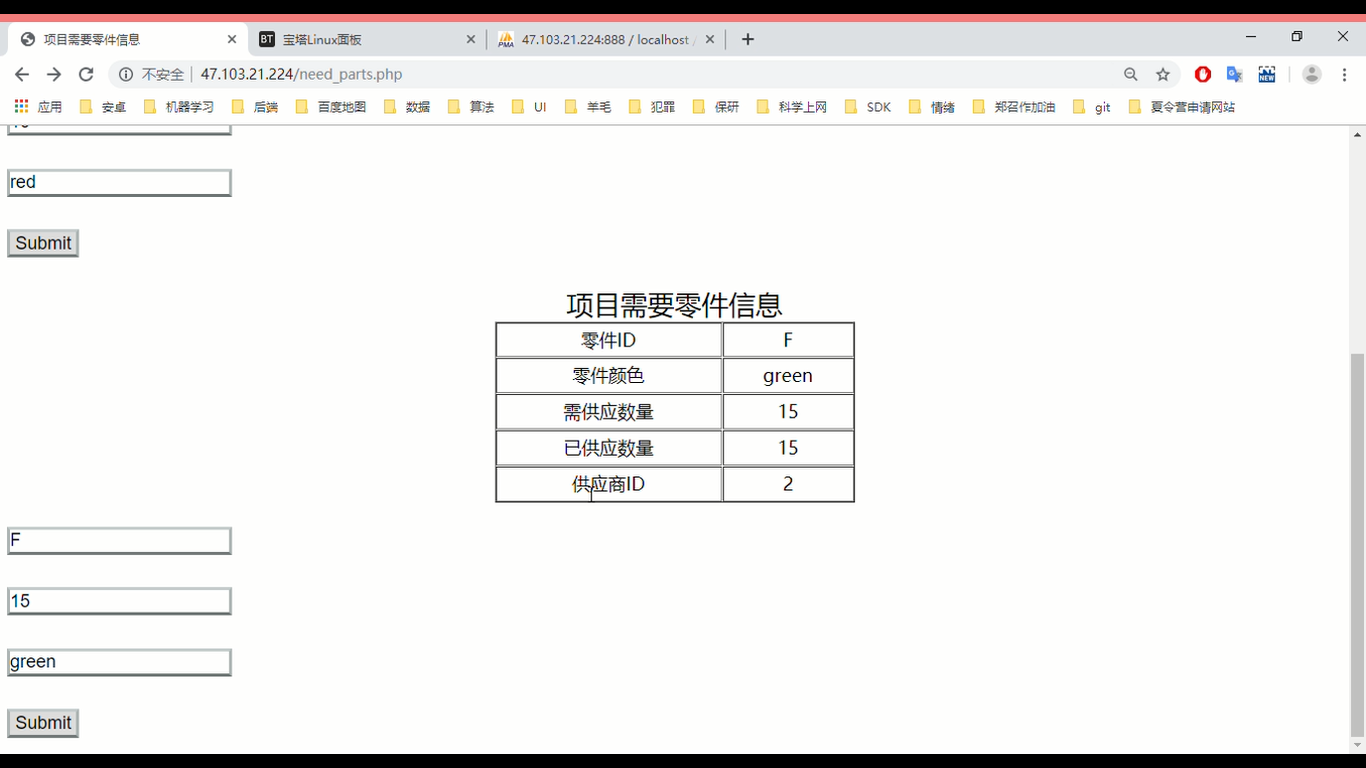


图4-16 返回零件采购界面2

## §4.4 供应商供应情况模块

功能描述：采购部门的部门经理可以对供应商的信息进行组合查询，例如找出在北京的可供应红色F型零件的供应商。

详细思路：实现过程中，我需要从前端获取到采购部门经理的输入，比如零件类型、零件数量、零件颜色，服务器接受到采购部门经理的输入后，后端通过传入的参数编写数据库查询语言，然后在后端执行数据库语言来实现supplier表和supply表的更新与检索。

文件介绍：

1、purchase.html：输入待查询的零件信息

2、search\_sup\_page.php：返回零件的查询结果

SQL语句：

1、对供应商的信息进行组合查询：

select \*

from supply natural join supplier

where sup\_place= $sup\_place and color = $parts\_color and parts\_id= $parts\_id.

实验截图：



图4-17 待查询的零件信息1

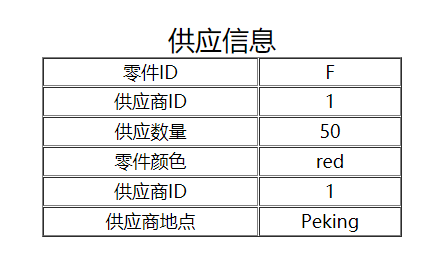


图4-18 零件的查询结果1



图4-19 待查询的零件信息2



图4-20 零件的查询结果2

## §4.5 项目阶段查看与修改模块

功能描述：项目负责人在输入项目ID、用户ID和密码，验证成功之后可以进入项目阶段，并修改项目阶段。当一个项目阶段完成之后，会将其加入到历史项目中。

文件介绍：

1、p\_info.html：输入项目ID

2、p\_search.php：根据项目ID返回项目信息、阶段修改页面

3、change\_phase.php：阶段修改成功页面

SQL语句：

1、根据项目ID查询项目信息：

select \* from need\_parts where project\_id = $p\_id

2、更新项目状态：

update project set phase = $phase where project\_id =$p\_id

3、将一个工程插入到历史过程中：

insert into history\_project values($p\_id , now())

实验截图：



图4-21 登陆负责人界面



图4-22 项目信息和零件信息

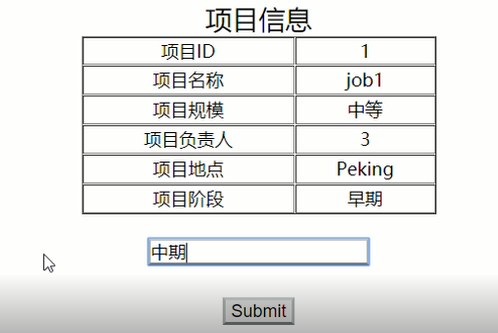


图4-23 项目阶段修改



图4-24项目阶段修改提示

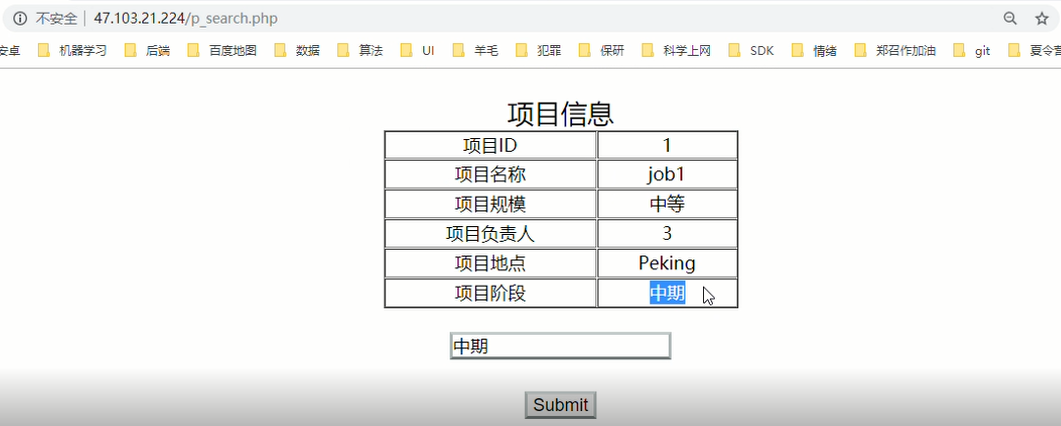


图4-25项目阶段修改成功

# 第五章 结束语

## §5.1实习过程中遇到的问题以及解决方法。

**遇到问题**：为了实现审计功能，添加了数据库恢复过程，而在恢复数据库时，提示尚未备份日志尾部，所以无法完成恢复。

**原因分析**：这是因为在线还原的数据库在最后备份后又产生了新的日志，所以按照默认设置的备份选项，系统将提示备份日志尾部以免造成事务中断。

**解决方法**：如果需要备份尾部日志则进行备份。如果不需要，则可以在还原数据库的选项卡中选择“覆盖现有数据库”。

**遇到问题**：使用service nginx start 启动服务时，nginx: [emerg] unknown log format "main" in /etc/nginx/nginx.conf:26

**原因分析**：开启了main 格式的日志信息，但是找不到日志的格式main ，一般来说都是打开了这个选项：access\_log /var/log/nginx/access.log main;。

解决方法：在access\_log /var/log/nginx/access.log main; 的上面有个日志格式main ，默认是注释的，把这三行取消注释就OK了

**遇到问题**：MySQL 1215 Cannot add foreign key constraint 错误

**错误原因**：外键数据格式和相对于的另一个表的主键格式不一样导致的

解决方案：把两个键的域类型改成相同的格式即可

**遇到问题**：由于窗体间降低了联系性，所以，当涉及到窗体间传递数据或者变量时，又是难题。

**解决方案**：数据不涉及安全保密是通过GET请求进行传输，涉及安全保密时通过POST来发送请求来增大联系，或直接通过内部函数响应来实现。

## §5.2实习的体会

通过这次课程设计，让我们更加深刻了解课堂上关于数据库系统的相关知识，并对以往对知识的疏忽得以补充，实现过程中遇到一些模糊的专业用语，比如说弱实体集以及函数依赖，在进行逻辑结构设计中，需要确定数据依赖，并对各个关系模式之间的数据依赖进行极小化处理，消除冗余的联系，在进行此过程前，我一直搞不懂函数依赖闭包如何求解，但是这些问题经过这次课程设计，都一一得以解决，我们相信这本书中还有很多我们未搞清楚的问题，但是这次的课程设计帮助我们打牢了基础知识，为我们以后的学习和工作打下了严实的基础。

虽然这次课程是那么短暂的两天时间，但是我们感觉到这一天我们的所学是非常值得的，这次任务原则上是实现，其实就是一次大的作业，是让我对课堂知识的巩固和对从前编程技术的熟悉和应用，使用MySQL数据库来进行数据的查询和更新，使用POST来请求获取零件、工程相关的信息。实现过程中，我做事的耐心和仔细程度得以提高，并且在和同学互相交流过程中，加大了我对数据库学习的深度。课程设计是培训学生运用本专业所学的理论知识和专业知识来分析解决实际问题的重要教学环节，是对三年所学知识的复习和巩固。因此，我们必须认真、谨慎、踏实、一步一步的完成设计。

课程设计是一个重要的教学环节，通过课程设计使我们了解到一些实际与理论之间的差异。通过课程设计不仅可以巩固专业知识，为以后的工作打下了坚实的基础，而其还可以培养和熟练使用资料，运用工具书的能力，把我们所学的课本知识与实践结合起来，起到温故而知新的作用。课程设计诚然是一门专业课，给我们很多专业知识以及专业技能上的提升。同时，设计实现过程中让我们感触很深。使我们对抽象的理论有了具体的认识。在课程设计过程中。我们要比较系统的了解数据库和数据库应用程序中的每一个环节，包括从需求分析、概念结构设计、逻辑结构设计原则，本次设计综合三年所学的专业课程，以《数据库系统概念》的指导思想为中心，参照有关资料，有计划有头绪、有逻辑地把这次设计搞好！

总之，这次课程设计使我们收获很多、学会很多、比以往更有耐心很多。感谢学校及老师给我们这次课程设计的机会，最真挚的感谢我们的章丽平老师，在设计过程中，老师精心的辅导和不厌其烦地的态度才使得我们以顺利的完成这次设计，她那无私的奉献的精神照耀着我们对学习的热爱，同时也增加我们对知识的追求和欲望度。