

# 餐厅管理系统数据库设计与实现

李艳琳 曾 丽

成都锦城学院计算机与软件学院 四川 成都 611731

【摘要】随着信息时代的来临，计算机技术不断发展，现如今计算机技术已经开始渗透进我们的日常生活中，为解决在餐厅日常管理过程中不断出现的问题，以及便于管理的理念，我设计了一个餐厅管理系统。在本系统数据库设计中，采用数据库表的创建以及查询的功能，在本餐厅系统数据库中用户可以查询到菜品，供应商等相关信息，主要功能就是便于用户管理餐厅。

【关键词】SQL 语言；管理；计算机技术

## 1 系统概述

近年来，随着经济不断发展，人们的生活水准不断提高，餐饮行业开始不断壮大，同行之间的竞争越来越激烈，新兴的餐饮模式层出不穷可是传统的餐饮行业还在使用人工管理的方法来管理餐厅。这样的运作方式大大降低了办事效率，在人工处理过程中还容易出现错误，可能出现记录的丢失以及错漏情况，原材料的购入以及入库都需要人工一个一个核对完成，并且客人点菜的时候都需要当面下单，由服务员将菜单送进厨房，菜品加工完成后再送给客人，最后的账单也需要人工来核对，最后完成订单。在此过程中，客人需要等待很久，而餐厅也需要投入很多，效益实在是太低。这里就迫切的需要现代技术来改善。作为计算机应用这一部分，可以有效解决这一问题，计算机查询方便，存储量大，计算速度快，成本低等特点正是目前我国餐饮行业最需要的。

餐饮行业是一个不容小觑的行业。只要由人类存在就需要餐饮行业，每个人都离不开吃，追求美食更是每个人的天性，人们在经济达到一定条件后就会开始追求更好的享受，餐饮就是其中一部分，由此社会越是向前发展就越是离不开餐饮业。

## 2 需求分析

### 2.1 系统需求分析

餐厅管理系统的关键就在于能否正确有效的管理好餐厅，可行性以及实用性就是需要考虑的方面。首先就是对于供应商的管理需要供应商提供包括商号，名字，报价，资质等信息之后才有资格为餐厅提供原材料，并且在提供原材料的时候也会自动生产订单方便交易之后的查询与核实。之后就是仓库，仓库会事先编号，在数

据库中会显示残酷的编号，存放的菜品以及菜品的余量是多少，在菜品余太少或者太多事时进行操做系统都会发出通知方便管理，菜品也可以由客人自己选着下单，之后由店员根据客人的桌号为客人进行服务，在服务完成之后还会为每一桌客人生成专属订单，其中就包括了客人当前消费以及为客人服务的店员的编号等信息。

### 2.2 数据需求管理

在对餐厅运行以及管理进行了多方考察以及各项评估之后得到了以下的数据项：

店员：店员属性包括员工号，姓名，电话。一个店员可以对多个供应商进行管理，为了方便管理，一个供应商只能由一个店员进行管理。一个店员还可以服务多个顾客。

供应商管理：在增加供应商时，应当提供的信息包括商号，供应商名称，电话，地址，法人姓名，资质等信息。一个供应商可以提供多种原料，一种原料也可以由多个供应商提供。

仓库原料管理：供应商提供的菜品将分门别类的暂时储存在仓库，系统将提供菜名，仓库编号，以及菜品余量。

菜品管理：菜品管理方面将显示菜名，售价，原料。在制作菜品时将根据原料在仓库原料中取得对应菜品进行制作，一道菜品由多种菜制成，一种菜也可以制作多种菜品。在取菜时还需提供重量。

顾客管理：顾客在来店提供姓名，电话方便店员为其服务，之后会生成属性性别以及桌号。

## 3 数据库结构设计

### 3.1 E-R 图设计

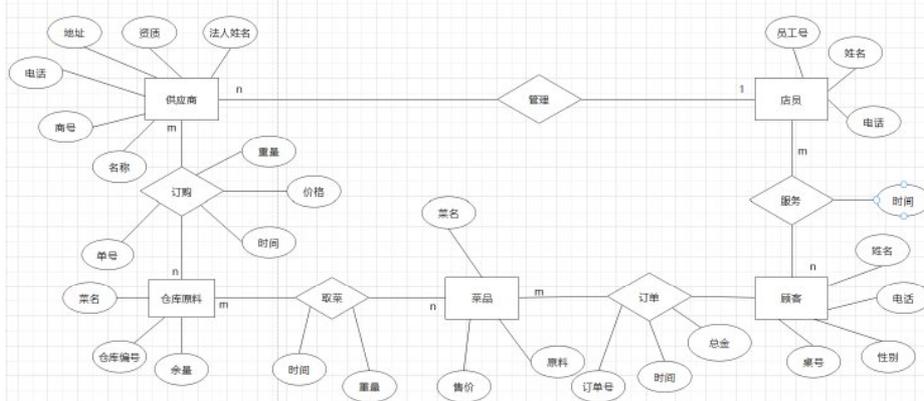


图1 餐厅系统ER图

### 3.2 关系模型

供应商（名称，商号，电话，地址，资质，法人姓名）

店员（员工号，姓名，电话）

仓库原料（菜名，仓库编号，余量）

菜品（菜名，售价，原料）

顾客（姓名，桌号，电话，性别）

订购（单号，商号，菜名，重量，价格，时间），

其中商号为外键。

取菜（菜名，原料，重量，时间），其中菜名，原料为外键

订单（桌号，订单号，总金额，时间）其中桌号为外键。

服务（员工号，桌号，时间）其中员工号，桌号为外键。

## 4 数据库的实现

### 4.1 数据库以及数据表的创建

在数据库中创建相关表，给定相应的名称以及约束，就可以得到相应的表，并在表与表之间创造联系。以下举出有关表的创建的代码实例：

首先是供应商表的创建，在这里将表名取为供应商，其中添加名称，商号，电话等基础信息，并将商号设置为主键，就得到以下代码：

```
create table 供应商
( 名称 nchar(10) not null ,
  商号 char(10) not null,
  电话 char(20) not null,
```

```
地址 nchar(20) not null,
  资质 nchar(10) not null,
  法人姓名 nchar(10) not null,
  primary key( 商号 )
)
```

上一个例子展现的是有关单独的表，没有和其他表有联系，接下来的例题将会牵扯到两个表，这个表用来记录供应商于用户之间的订购行为，将表名取为订购，并在其中输入单号，商号等有关列，其中单号为本表主键，而订购时必须要有商号一起才可以生成有关完整的订购行为，为此将供应商表中的主键商号设置为订购表中的外键，使得两表产生联系：

```
create table 订购
( 单号 char(10) not null,
  商号 char(10) not null,
  菜名 nchar(6) not null,
  重量 char(10) not null,
  价格 char(10) not null,
  时间 date,
  primary key( 单号 ),
  foreign key( 商号 ) references 供应商 ( 商号 )
)
```

### 4.2 关系图

将所有的表都创建完成后，由于表与表之间有外键作为联系就会生成以下关系图：

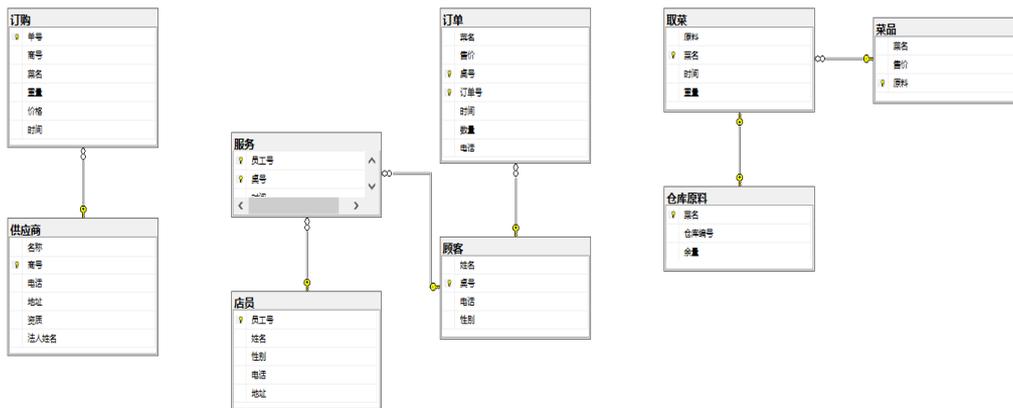


图 2 餐厅系统关系图

### 4.3 视图

在餐厅日常管理中，原材料的供应以及菜品的供应占了较大的一部分，为此，及时且合理的监管原料余量就非常重要。由此，我创建有以下功能的视图：在表中创建视图名字为麻婆豆腐余量查询的视图查询麻婆豆腐原料的余量。代码如下：

```
create view 麻婆豆腐余量查询
as
select 余量
from 仓库原料
where 菜名 in(select 原料 from 菜品 where 菜
```

名 = ' 麻婆豆腐' )

### 4.4 触发器

菜品在出售的时候，良好的性价比是关键，过于高昂的价格不仅使得菜品销售不易，还会使得菜品积压，导致亏本，在本功能中可以定义名字为菜品售价规定的触发器规定菜品的价格不要超过 40 元，超过则输出菜品价格不得超过 40 元：

```
create trigger 菜品售价规定
on 菜品 after insert
as
if (select 售价 from 菜品
```

```
where 售价 in (select 售价 from inserted)<40  
rollback  
print' 菜品价格不得超过 40 元'
```

注释：触发器内容可以根据餐厅的实际需求进行调整，以便于针对各种不同情况

#### 4.5 存储过程

(1) 在餐厅运行过程中，会有点菜这一环节，这时为顾客提供相关菜品的售价会更加方便顾客进行挑选，下单。在本存储过程中可以输入想要的菜名，系统就会为用户提供相应的菜品的价格：

```
create procedure 菜品价格查询  
@菜名 nchar(6)  
as  
select 菜名, 售价  
from 菜品  
where 菜名=@菜名
```

注释：在调用此存储过程时，需要给@菜名赋予一个常量，调用语句为：exec <存储过程名> <赋值>。如：查询青椒肉丝的价格时，就需要调用以下语句：exec 菜品价格查询 青椒肉丝。

(2) 为了更加深刻的了解餐厅的运行情况，需要对于供应商有更加深入的了解，供应商与餐厅合作最密切的就是菜品提供。在用户需要了解供应商提供菜品数量问题可采用如下语句来查询哪一个商号供应菜品最多：

```
create procedure 供应商菜品提供情况查询  
as  
select top 1 商号, count(单号) 菜品提供次数  
from 订购  
group by 商号  
order by count(单号) desc
```

在需要调用该语句时，可以使用如下调用语句：  
exec 供应商菜品提供情况查询

## 5 测试

### 5.1 输入测试

需要在表中添加信息时就可以使如下语句进行添加：

```
insert into 菜品(菜名, 售价, 原料)  
values('蒜苗回锅肉', '15', '蒜苗, 五花肉')  
insert into 菜品(菜名, 售价, 原料)  
values('青椒回锅肉', '15', '青椒, 五花肉')
```

### 5.2 修改和删除测试

这里的功能主要是方便用户对于表中的内容进行随意对增加或者减少，餐饮行业时时都在更新需要不断地改进。

例 1: 将紫菜蛋汤的价格改为 5

```
update 菜品 set 售价 = 5
```

```
where 菜名 = '紫菜蛋汤'
```

例 2: 将乐山钵钵鸡下架：

```
delete from 菜品  
where 菜名 = '乐山钵钵鸡'
```

## 5.4 功能测试部分

在用户需要知道员工业绩相关信息时可以使用如下语句，比如可以通过以下语句来查询姓名为小白的员工服务了哪些顾客：

```
select 桌号  
from 店员 inner join 服务  
on 店员. 员工号 = 服务. 员工号  
where 姓名 = '小白'
```

## 6 总结

总的来说在餐厅数据库管理系统之中，系统需要运行的环境时时都在变化，菜品以及人工的管理并不是一成不变的，为了保证服务器能够有效且稳定的运行，需要定期进行维护与更新就好比是很多高校的各大系统以及学生报名之类的系统时常崩溃就有服务器不稳定的原因，再来就是人流太大，在一个餐厅达到用餐高峰的时候难免会有系统崩坏的时候，但是系统一旦崩坏就会给餐厅带来很坏的影响，最好给餐厅做一个专用的服务器，避免崩坏，并且对数据库经常进行清理备份，保证数据库的运作正常，防止数据库数据丢失。在日常使用中时时进行反馈改进，这样子在这个大数据流行的时代才可以避免被淘汰。

如今信息时代已经来临，运用计算机技术解决问题会为我们带来巨大的便利。餐厅管理系统在目前来说还不是那么多普遍，但还是由一些餐厅正在使用，效果也确实不错，本文的重点就是把数据库相关知识与实际生活相互融合。

这个系统主要采用了 SQL 技术，将实际生活中繁杂的工作简化成一串串数字，将其展现在管理员面前，让用户能够有更直观的体验。

## 【参考文献】

- [1] 何玉洁. 数据库系统教程(第2版)[M]. 人民邮电出版社, 2015
- [2] 周志逵等. 数据库系统原理[M]. 清华大学出版社, 2010
- [3] 吴洁明, 方英兰等编著, 软件工程实例教程[M]. 清华大学出版社, 2010