目 录

[一、需求分析 1](#_Toc61292514)

[1.1外卖点餐现状分析 1](#_Toc61292515)

[1.2顾客需求分析 1](#_Toc61292516)

[1.3管理员需求分析 1](#_Toc61292517)

[1.4性能需求分析 1](#_Toc61292518)

[1.5可行性分析 2](#_Toc61292519)

[二、概念结构设计 2](#_Toc61292520)

[2.1实体和属性 2](#_Toc61292521)

[2.2系统E-R图 2](#_Toc61292522)

[三、逻辑结构设计 3](#_Toc61292523)

[3.1关系设计 3](#_Toc61292524)

[3.2表的设计 3](#_Toc61292525)

[四、物理结构设计 5](#_Toc61292526)

[4.1设计表间关系 5](#_Toc61292527)

[4.2完整性设计 5](#_Toc61292528)

[五、数据库实施 6](#_Toc61292529)

[5.1建立数据库 6](#_Toc61292530)

[5.2建立表 7](#_Toc61292531)

[5.3数据载入 8](#_Toc61292532)

[5.4数据查询 10](#_Toc61292533)

[5.5创建存储过程、函数、触发器等 18](#_Toc61292534)

[六、数据库运行和维护 24](#_Toc61292535)

[6.1数据库安全性操作 24](#_Toc61292536)

[6.2数据库备份操作 25](#_Toc61292537)

[七、课设总结 26](#_Toc61292538)

一、需求分析

1.1外卖点餐现状分析

随着社会经济财富的不断上涨，人们在吃喝这个方面已经是变得越来越丰富了。毫无疑问，在吃喝这个方面的方式也是变得与之前截然不同。现在的人们特别是年轻人们普遍喜欢使用外卖点餐的方式来吃一顿,这种足不出户就能吃饱饭的方式方便了很多人,吃、喝的东西的种类也很丰富,在价格上也是处于可以接受的范围之内。这种外卖点餐的方式可以将菜品、订单、送餐等的信息记录在案,既可以看到菜品的图片,也能对菜量和菜品样式有直观的概念。用户在线上下完单后，商家就会接到订单信息然后开始准备做餐，与此同时另外一边的送餐骑手也会接到送餐的信息。商家做完餐后再由送餐骑手配送至用户手里。若用户在下单后突然有事也可以进行取消订单的操作。

这样的外卖点餐方式让菜品能图文并茂显示出来供用户选择,很多信息也能够实时更新和查看，这确实是节省了相关人员很多的时间和空间等。

1.2顾客需求分析

顾客模块的功能包括用户信息管理,美食选购,配送信息、美食评价四大功能。其中：

1. 在用户信息管理中可以对用户信息进行修改、还可以查看之前下单过的外卖情况；
2. 在选购美食的过程中可以通过加入购物车然后购买的形式；
3. 在配送信息这里可以选择外卖配送或者到店自取（到店自取看商家设置情况）；
4. 美食评价中顾客可以这对这次外卖点餐的商家服务、产品质量、配送服务等留下自己的评价，也可以选择不评价。

1.3管理员需求分析

管理员模块的功能包括菜品管理,订单管理,会员信息管理三大功能。其中：

1. 在菜品管理中可以对菜品信息进行添加、修改和查询操作；
2. 在订单管理中可以通过未确认、已确认、已下单三种形式进行管理、也可查看所有订单信息；
3. 会员信息管理中,可以添加会员信息和修改会员信息,进而方便顾客网上下单美食,并享有优惠等的一系列操作,这一做法人性化的操作既可以起到推广平台的作用,又减少了顾客的直接的抵触情绪。

1.4性能需求分析

该系统在性能功能上应达到如下需求:

1. 操作简单、界面友好:完全控件式的页面布局,使得菜品,配送,优惠等信息的录入工作更简便,很多选项包括商家信息、商家联系方式、配送进度等只需要点击相关界面就可以轻松完成；
2. 能自动跟踪出现的提示信息也让用户随时清楚自己的操作情况。对常见的类似平台的管理的各个方面:基本信息录入、浏览、删除、修改、搜索等方面都能大体实现,顾客对菜品的预定及支付方式；
3. 即时可见:对用户的订单信息的处理(包括录入、删除)将立即在主页的对应栏目显示出来,达到实时的功能；
4. 系统运行应该快速、稳定、高效和可靠；
5. 在结构上应具有很好的可扩展性,便于将来的功能扩展和维护。

1.5可行性分析

随着经济的快速发展,B2C电子商务越来越受到人们的关注。外卖点餐平台,已经出现在了中国大地的很多角落并发展得很迅速，现在看在特别是在公司、企业内部,团体外卖点餐已经发展得较为成熟，但这仍然有着发展空间，因为这种模式可以折射出很多行业很多企业很多地方也可以这么做。与此同时,随着全国人民的精神、物质和文化生活高度的提高,人们已经不在仅仅停留在吃饱的程度,不但要吃饱,还要吃好,而且口味也越来越挑刺,正因为外卖点餐平台的出现,迎合了这些人的口味,他们不但省去了自己做饭的麻烦,而且也能寻找和发现新的菜品,品尝不同风格的菜品,这也难免得以让外卖点餐平台在中国有了极大的发展空间。而当代大学生、白领工作人士作为社会中的重要群体,他们对饮食的要求都在不断提高,饮食在他们生活中起着重要作用。因此,主要面向大学生和白领人士的外卖点餐平台的设计有着良好的现实意义。

二、概念结构设计

2.1实体和属性

通过需求分析阶段的分析和结果，本系统所要设计的实体和属性如下。

1. 菜谱(菜谱编号，菜名，原价，优惠价，菜名介绍，创建时间)
2. 用户(用户编号，姓名，地址，手机号，注册时间)
3. 管理员(管理员编号，姓名，管理员密码，登录IP，最后登录时间)
4. 订单(订单编号，手机号，地址，总金额，下单时间，订单状态，订单改变时间，备注，会员编号)

2.2系统E-R图

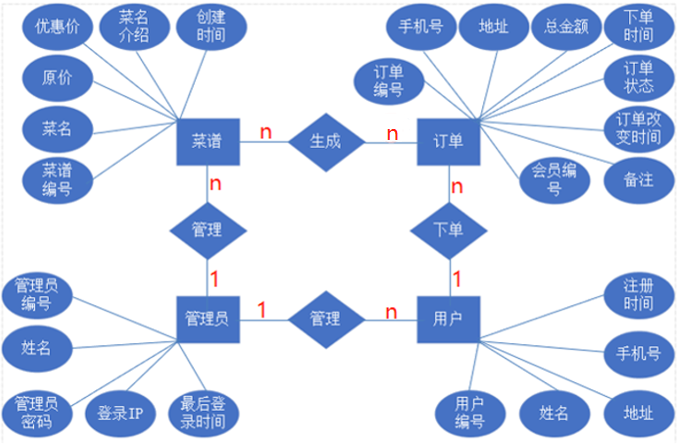
通过上面的介绍,我们已经基本认识到这个外卖点餐数据库的实体有菜谱、用户、管理员和订单。实体之间有着一个管理员可以有着拥有多个用户、多个菜谱的管理关系,因此管理员与用户之间和菜谱之间都是一对多的联系;一个用户只能属于管理员,因此用户与管理员是多对一的联系;一个订单只属于一个用户,但可以拥有多个菜谱,因此订单与菜谱的关系是一对多的联系。由此分析,我们可以得到该数据库的E-R图，如下图。

图2-1外卖点餐数据库-E-R图

三、逻辑结构设计

3.1关系设计

根据转换原则转换成5个关系模式，关系的主码用下划线标出。

1. 菜谱(菜谱编号，菜名，原价，优惠价，菜名介绍，创建时间)
2. 用户(用户编号，姓名，地址，手机号，注册时间)
3. 管理员(管理员编号，姓名，管理员密码，登录IP，最后登录时间)
4. 订单(订单编号，手机号，地址，总金额，下单时间，订单状态，订单改变时间，备注，会员编号)
5. 管理(管理员编号，菜谱编号，用户编号)
6. 下单(订单编号，用户编号)
7. 生成(菜谱编号，菜谱编号)

3.2表的设计

根据E-R图转换成基本表的思想,即每个实体转换成基本表,而1对多的联系根据实际需要可以转换或不转换为基本表,按照图2-1所示的E-R图,该系统的基本表应为5个,分别为：

1. 菜谱信息表
2. 用户信息表
3. 管理员信息表
4. 订单信息表
5. 订单项信息表

表3-1菜谱信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名** | **字段名** | **字段类型** | **主键** | **约束说明** |
| **菜谱信息表** | 菜谱编号 | int | 是 |  |
| 菜名 | char(10) |  | 不允许为空 |
| 原价 | float |  |  |
| 优惠价 | float |  |  |
| 菜名介绍 | char(50) |  |  |
| 创建时间 | datetime |  |  |

表3-2用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名** | **字段名** | **字段类型** | **主键** | **约束说明** |
| **用户信息表** | 用户编号 | int | 是 |  |
| 姓名 | char(10) |  |  |
| 地址 | char(50) |  |  |
| 手机号 | char(15) |  |  |
| 注册时间 | datetime |  |  |

表3-3管理员信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名** | **字段名** | **字段类型** | **主键** | **约束说明** |
| **管理员信息表** | 管理员编号 | int | 是 |  |
| 姓名 | char(10) |  |  |
| 管理员密码 | char(10) |  |  |
| 登录IP | char(20) |  |  |
| 最后登录时间 | datetime |  |  |

表3-4订单信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名** | **字段名** | **字段类型** | **主键** | **约束说明** |
| **订单信息表** | 订单编号 | char(15) | 是 |  |
| 手机号 | char(15) |  |  |
| 地址 | char(50) |  |  |
| 总金额 | float |  |  |
| 下单时间 | datetime |  |  |
| 订单状态 | char(10) |  |  |
| 订单改变时间 | datetime |  |  |
| 备注 | char(50) |  |  |
| 会员编号 | int |  | 参照用户信息表的用户编号，不允许为空 |

表3-5订单项信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名** | **字段名** | **字段类型** | **主键** | **约束说明** |
| **订单项信息表** | 订单项编号 | char(10) | 是 |  |
| 订单编号 | char(15) |  | 参照订单信息表的订单编号，不允许为空 |
| 姓名 | char(10) |  |  |
| 状态 | char(10) |  |  |

四、物理结构设计

4.1设计表间关系

在外卖点餐中,存在着参照关系和被参照关系。

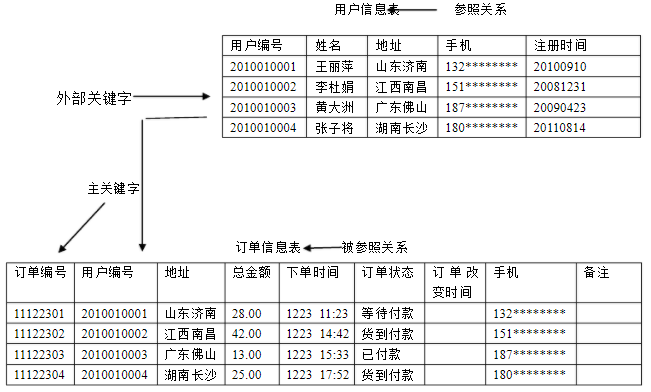


图4-1参照关系与被参照关系

4.2完整性设计

**1、主要字段完整性的字段名、完整性约束条件**

关系模型的完整性约束包括实体完整性、参照完整性和用户自定义完整性。主要字段完整性的字段名、完整性约束条件指的是实体完整性。其中实体完整性的规则如下:

1)实体完整性规则是针对关系而言的,而关系则对应一个现实世界中的实体集。

2)现实世界中的实体是可区分的,它们具有某种标识特征;相应地,关系中的元组也是可区分的,在关系中用主关键字作唯一性标识。

3)主关键字中的属性,即主属性不能取空值。

在外卖点餐中,管理员表(管理员编号,姓名,密码,登陆IP,最后登录时间)中管理员编号是主关键字,不可以取空值;用户表(用户编号,姓名,地址,手机号,注册时间)中,用户编号是主关键字,不可以取空值;订单表(订单编号,手机号,地址,总金额,下单时间,订单状态,订单改变时间,备注,会员编号)中,订单编号是主关键字,不可以取空值;菜谱表(菜谱编号,菜名,原价,优惠价,菜名介绍,创建时间)中,菜谱编号是主关键字,不可以取空值;订单项表(订单项编号,订单编号,姓名,状态)中,订单项编号是主关键字,不可以取空值。

**2、记录完整性约束及其约束条件**

除了实体完整性之外还有其他与应用密切相关的数据完整性约束,例如某个属性的值必须唯一,某个属性的取值必须在某个范围内,某些属性值之间应该满足一定的函数关系等。类似这些方面的约束不是关系数据模型本身所要求的,而是为了满足应用方面的语义要求提出来的,这些完整性需求需要用户来定义,所以称为用户自定义完整性。数据库管理系统需提供定义这些数据完整性的功能和手段,以便统一进行处理和检查,而不是由应用程序去实现这些功能。

在用户自定义完整性中最常见的是限定属性的取值范围,即对值域的约束,这包括说明属性的数据类型、精度、取值范围、是否允许空值等。在外卖点餐中,订单信息表中的订单编号不允许为空值,用户表中的用户编号为唯一值都是用了用户自定义完整性约束。

**3、参照完整性表**

现实世界中的实体之间存在着某种联系,而在关系模型中实体是用关系描述的,实体之间的联系也是用关系描述的,这样就自然存在着关系和关系之间的参照或引用。外卖点餐参照完整性如下：

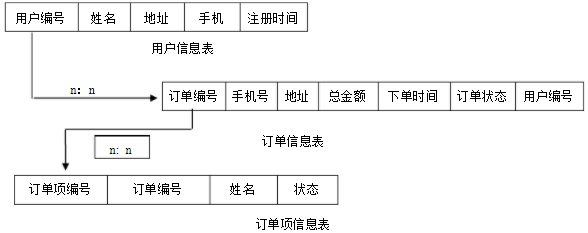


图4-2说明关联和参照关

五、数据库实施

5.1建立数据库

启动服务管理器,打开查询分析器,在调试程序窗口输入下列语句：

create database 外卖点餐数据库

on(

name=外卖点餐数据库\_data,

filename='D:\DATA\外卖点餐数据库\_data.mdf',

size=5mb,

maxsize=100mb,

filegrowth=1mb

)

log on(

name=外卖点餐数据库\_log,

filename='D:\DATA\外卖点餐数据库\_log.ldf',

size=5mb,

maxsize=50mb,

filegrowth=1%

)

5.2建立表

use 外卖点餐数据库

go

create table 菜谱信息表( --建立菜谱信息表

菜谱编号 int not null primary key ,

菜名 char(10) not null,

原价 float null,

优惠价 float null,

菜名介绍 char(50) null,

创建时间 datetime

)

use 外卖点餐数据库

go

create table 用户信息表( --建立用户信息表

用户编号 int not null primary key ,

姓名 char(10) null,

地址 char(50) null,

手机号 char(15) null,

注册时间 datetime null

)

use 外卖点餐数据库

go

create table 管理员信息表( --建立管理员信息表

管理员编号 int not null primary key ,

姓名 char(10) null,

管理员密码 char(10) null,

登录IP char(20) null,

最后登录时间 datetime null

)

use 外卖点餐数据库

go

create table 订单信息表( --建立订单信息表

订单编号 char(15) not null primary key ,

手机号 char(15) null,

地址 char(50) null,

总金额 float null,

下单时间 datetime null,

订单状态 char(10) null,

订单改变时间 datetime null,

备注 char(50) null,

会员编号 int not null foreign key references 用户信息表(用户编号)

)

use 外卖点餐数据库

go

create table 订单项信息表( --建立订单项信息表

订单项编号 char(10) not null primary key,

订单编号 char(15) not null foreign key references 订单信息表(订单编号),

姓名 char(10) null,

状态 char(10) null

)

5.3数据载入

在外卖点餐数据库中分别新建菜谱信息表、用户信息表、管理员信息表、订单信息表、订单项信息表,再陆续载入数据:

**1)插入菜谱信息表的数据:**

use 外卖点餐数据库

go

insert into 菜谱信息表 values

(001,'番茄炒蛋',10.5,8.8,'红黄搭配，美味又好看！','2021-01-04'),

(002,'鱼香肉丝',13.5,9.8,'条条分明，香飘几十里！','2021-01-04'),

(003,'土豆焖鸡',45.5,38.8,'土豆的松香味加上鸡肉的清香！','2021-01-04'),

(004,'深井烧鸭',68.5,58.8,'深井烧鸭，一口香脆满口香！','2021-01-04'),

(005,'清蒸排骨',25.5,18.8,'豆是下的排骨，简直香喷喷！','2021-01-04')

**2)插入用户信息表的数据:**

use 外卖点餐数据库

go

insert into 用户信息表 values

(2021001, '闲忆秋','广东广州','1122334455','2021-01-04'),

(2021002, '微芷瑶','广东肇庆','2233445566','2021-01-04'),

(2021003, '向 北','广东佛山','3344556677','2021-01-04'),

(2021004, 'Andeng','云南大理','4455667788','2021-01-04'),

(2021005, '阿里宝','浙江杭州','5566778899','2021-01-04')

**3)插入****管理员信息表的数据:**

use 外卖点餐数据库

go

insert into 管理员信息表 values

(101, '闲忆秋\_ad','admin101','10.3.5.101','2021-01-08'),

(102, '微芷瑶\_ad','admin102','10.3.5.102','2021-01-08'),

(103, '向 北\_ad','admin103','10.3.5.103','2021-01-08'),

(104, 'Andeng\_ad','admin104','10.3.5.104','2021-01-08'),

(105, '阿里宝\_ad','admin105','10.3.5.105','2021-01-08')

**4)插入订单信息表的数据:**

use 外卖点餐数据库

go

insert into 订单信息表 values

('20210104001','1122334455','广东广州',18.6,'11:30:20','等待付款','','2021001',''),

('20210104002','2233445566','广东肇庆',38.8,'11:31:21','已付款','','2021001',''),

('20210104003','3344556677','广东佛山',58.8,'11:32:22','已付款','','2021003',''),

('20210104004','4455667788','云南大理',57.6,'11:33:23','到店自取','','2021004',''),

('20210104005','5566778899','浙江杭州',97.6,'11:34:24','已付款','','2021005','')

这里插入订单信息表数据时会提示insert语句和外键FK\_\_订单信息表\_\_会员编号\_\_164452B1（外键名称可能不一样）有冲突，发生于用户信息表的用户编号列。然而在这里我们只需先删除这个位于订单信息表的外键约束即可插入。所以先执行下列语句：

alter table 订单信息表

drop constraint FK\_\_订单信息表\_\_会员编号\_\_164452B1

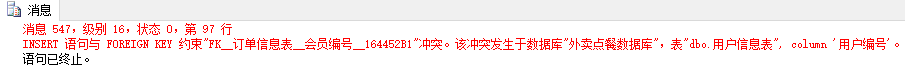


图5-3-1 插入冲突

**5)插入订单项信息表的数据:**

insert into 订单项信息表 values

('0104001','20210104001','闲忆秋','等待付款'),

('0104002','20210104002','微芷瑶','已付款'),

('0104003','20210104003','向 北','已付款'),

('0104004','20210104004','Andeng','到店自取'),

('0104005','20210104005','阿里宝','已付款')

5.4数据查询

**1)简单查询**

①投影查询

--a.查看外卖点餐数据库中菜谱信息表的菜名、优惠价、菜名介绍

use 外卖点餐数据库

go

select 菜名,优惠价,菜名介绍 from 菜谱信息表

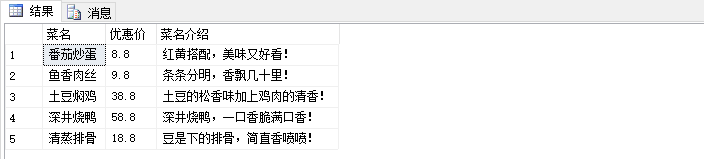


图5-4-1投影查询a

--b.查询外卖点餐数据库中订单信息表的所有信息

use 外卖点餐数据库

go

select \* from 订单信息表

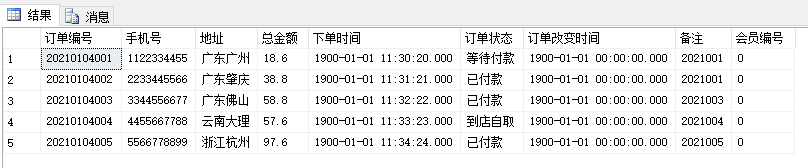


图5-4-2投影查询b

--c.查询外卖点餐数据库中管理员信息表的前三行信息

use 外卖点餐数据库

go

select top 3 \* from 管理员信息表



图5-4-3投影查询c

②选择查询\_关系

--查看外卖点餐数据库订单信息表中总金额大于等于50.0的订单编号、手机号、地址、订单状态。

use 外卖点餐数据库

go

select 订单编号,手机号,地址,订单状态 from 订单信息表

where 总金额>=50.0

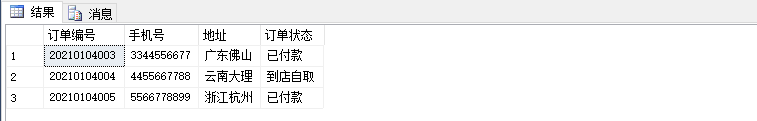


图5-4-4选择查询\_关系

③选择查询\_逻辑

--查看外卖点餐数据库用户信息表中注册时间是2021-01-04且地址是广东广州的用户信息。

use 外卖点餐数据库

go

select \* from 用户信息表

where 注册时间='2021-01-04' and 地址='广东广州'

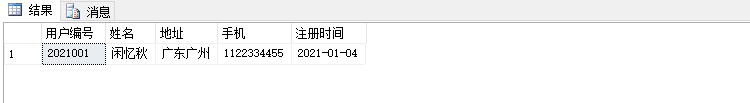


图5-4-5选择查询\_逻辑

④选择查询\_between

--查询外卖点餐数据库菜谱信息表中优惠价在10.0-40.0之间的菜谱编号,菜名,原价,优惠价。

use 外卖点餐数据库

go

select 菜谱编号,菜名,原价,优惠价 from 菜谱信息表

where 优惠价 between 10.0 and 40.0

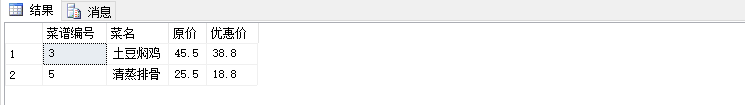


图5-4-6选择查询\_ between

⑤模糊查询\_like

--a.查看外卖点餐数据库订单信息表中订单状态包含“付款”的所有订单信息\_视图

use 外卖点餐数据库

go

create view 已付款或等待付款订单

as

select \* from 订单信息表

where 订单状态 like '%付款%'

--select \* from 已付款或等待付款订单

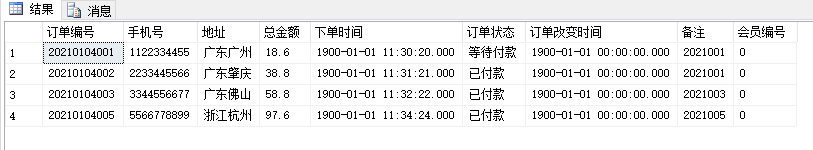


图5-4-7模糊查询a

--b.查看外卖点餐数据库用户信息表中手机以“2”开头的用户信息

use 外卖点餐数据库

go

select \* from 用户信息表

where 手机号 like '2%'

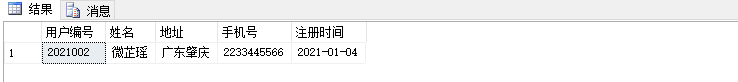


图5-4-8模糊查询b

⑥复合条件查询(and、or、not)

--查看外卖点餐数据库订单信息表中所有地址是广东和浙江并已付款的用户信息

use 外卖点餐数据库

go

select \* from 订单信息表

where 地址 like '%广东%' and 订单状态='已付款'

or 地址 like '%浙江%' and 订单状态='已付款'

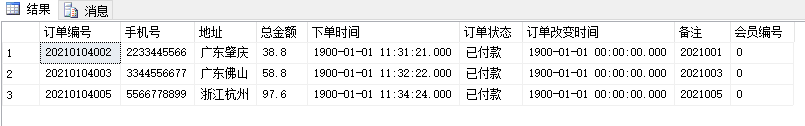


图5-4-9复合条件查询

⑦统计查询\_sum

--查看外卖点餐数据库订单信息表中全部营业额。\_视图

use 外卖点餐数据库

go

create view 营业额

as

select sum(总金额) 全部营业额 from 订单信息表

--select \* from 营业额



图5-4-10统计查询\_sum

⑧统计查询\_avg

--看外卖点餐数据库订单信息表中平均每单的金额。\_视图

use 外卖点餐数据库

go

create view 平均每单营业额

as

select avg(总金额) 平均每单的金额 from 订单信息表

--select \* from 平均每单营业额

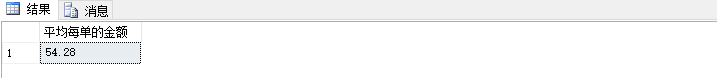


图5-4-11统计查询\_avg

⑨统计查询\_count

--看外卖点餐数据库用户信息表中用户总数。

use 外卖点餐数据库

go

select count(姓名) 用户总数 from 用户信息表



图5-4-12统计查询\_count

**2)多表查询**

①内连接查询

--查看外卖点餐数据库每个订单的订单编号,总金额,订单状态,用户编号,姓名,手机号,地址

use 外卖点餐数据库

go

select 订单编号,总金额,订单状态,用户编号,姓名,b.手机号,b.地址

from 订单信息表 a inner join 用户信息表 b

on a.手机号=b.手机号

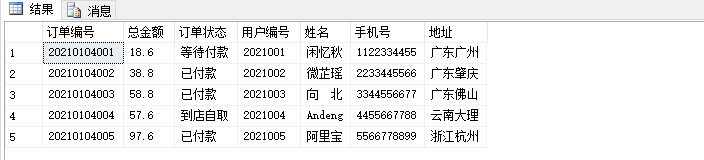


图5-4-13内连接查询

②左外连接查询

--查看外卖点餐数据库中每个用户及其下单的订单状态(含未下单的用户)

use 外卖点餐数据库

go

select a.\*,b.订单编号,b.总金额,b.订单状态

from 用户信息表 a left join 订单信息表 b

on a.手机号=b.手机号

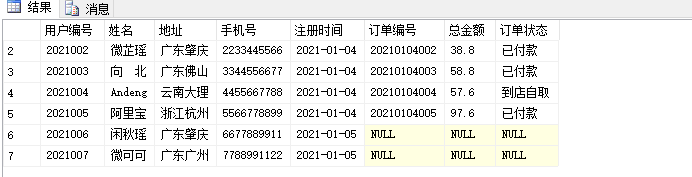


图5-4-14左外连接查询

注：为方便查询，在这里开始向用户信息表插入了一下记录：

use 外卖点餐数据库

go

insert into 用户信息表 values

(2021006, '闲秋瑶','广东肇庆','6677889911','2021-01-05'),

(2021007, '微可可','广东广州','7788991122','2021-01-05')

③右外连接查询

--查看外卖点餐数据库中每个订单及下单用户的注册时间情况(含未下单的用户)

use 外卖点餐数据库

go

select b.\*,a.用户编号,a.姓名,a.注册时间

from 订单信息表 b right join 用户信息表 a

on a.手机号=b.手机号

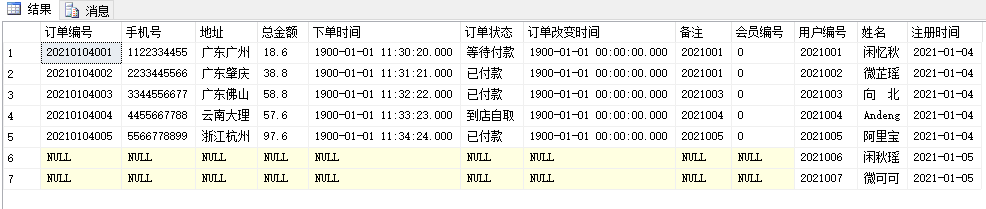


图5-4-15右外连接查询

④交叉连接查询(笛卡儿乘积)

--假如未知用户的手机号及地址，查看外卖点餐数据库中所有用户可能的下单情况

use 外卖点餐数据库

go

select a.\*,b.用户编号,b.姓名,b.注册时间

from 订单信息表 a cross join 用户信息表 b



图5-4-16交叉连接查询(笛卡儿乘积)

**3)子查询**

①无关子查询\_比较子查询

--查看外卖点餐数据库用户信息表中与姓名为微芷瑶同时注册的用户编号,姓名,手机号,注册时间

use 外卖点餐数据库

go

select 用户编号,姓名,手机号,注册时间 from 用户信息表

where 注册时间=(select 注册时间 from 用户信息表 where 姓名='微芷瑶')

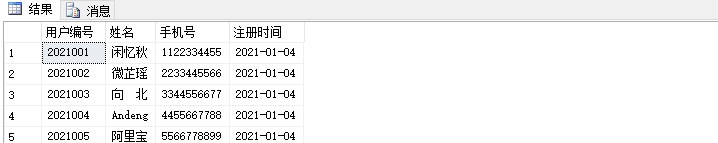


图5-4-17无关子查询\_比较子查询

②无关子查询\_in子查询

--查看外卖点餐数据库中订单状态是已付款的用户编号,姓名,手机号

use 外卖点餐数据库

go

select 用户编号,姓名,手机号 from 用户信息表

where 手机号 in(select 手机号 from 订单信息表

where 订单状态='已付款')

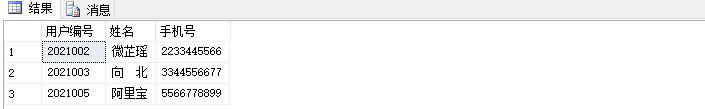


图5-4-18无关子查询\_in子查询

③相关子查询\_比较子查询

--查看外卖点餐数据库订单信息表中每单的总金额比平均每单的金额低的订单信息。

use 外卖点餐数据库

go

select \* from 订单信息表

where 总金额<(select avg(总金额) from 订单信息表

where 订单信息表.手机号=订单信息表.手机号)

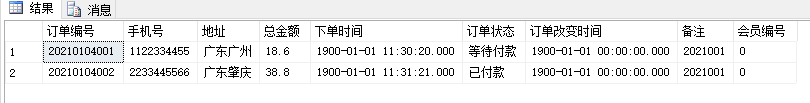


图5-4-19相关子查询\_比较子查询

④相关子查询\_exists查询

--查看外卖点餐数据库用户信息表中注册时间是2021-01-04的用户的订单信息

use 外卖点餐数据库

go

select \* from 订单信息表

where exists

(select \* from 用户信息表 where 用户编号=用户信息表.用户编号

and 注册时间='2021-01-04')

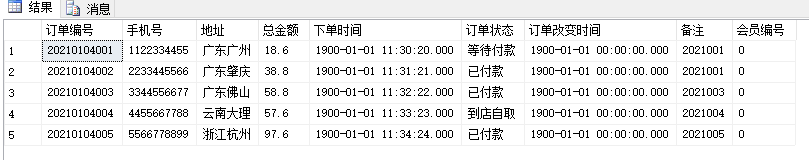


图5-4-20相关子查询\_exists查询

**4)其他查询**

①union联合查询

--查看外卖点餐数据库订单信息表中订单状态是已付款和等待付款的用户编号,姓名,地址,手机号

use 外卖点餐数据库

go

select 用户编号,姓名,用户信息表.地址,用户信息表.手机号,

订单编号,订单状态 from 用户信息表,订单信息表

where 订单状态='已付款' and 用户信息表.手机号=订单信息表.手机号

union

select 用户编号,姓名,用户信息表.地址,用户信息表.手机号,

订单编号,订单状态 from 用户信息表,订单信息表

where 订单状态='等待付款' and 用户信息表.手机号=订单信息表.手机号



图5-4-21 union联合查询

②对查询结果排序(asc升序，desc降序)

--查看外卖点餐数据库菜谱信息表中的菜谱编号,菜名,优惠价,菜名介绍，并按优惠价升序排列。

use 外卖点餐数据库

go

select 菜谱编号,菜名,优惠价,菜名介绍 from 菜谱信息表

order by 优惠价 asc

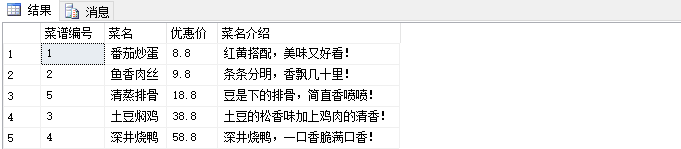


图5-4-22 对查询结果排序

5.5创建存储过程、函数、触发器等

①创建存储过程

--在外卖点餐数据库中创建带参数的存储过程，修改某个用户的手机号和地址

use 外卖点餐数据库

go

create procedure 修改用户信息 @用户编号 int,

@手机号 char(15),@地址 char(50)

as

update 用户信息表 set 手机号=@手机号,地址=@地址

where 用户编号=@用户编号

--exec 修改用户信息 '2021007','9988776655','广西桂林'

--select \* from 用户信息表

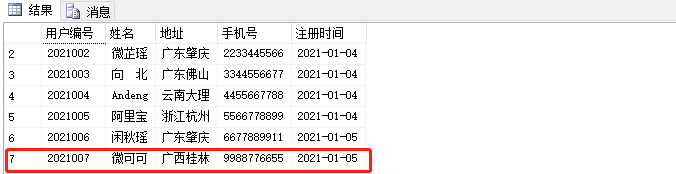


图5-5-1执行存储过程

②创建函数

--查看外卖点餐数据库订单信息表中某个订单状态所有订单的订单编号,用户编号,手机号,地址,总金额,订单状态。

use 外卖点餐数据库

go

create function 查看某个状态的订单(@状态 char(10)) returns table

as return

(select 订单编号,用户编号,订单信息表.手机号,订单信息表.地址,总金额,订单状态 from 订单信息表,用户信息表

where 订单状态=@状态 and 订单信息表.手机号=用户信息表.手机号)

--select \* from 查看某个状态的订单('等待付款')



图5-5-2执行函数

③创建触发器

--a.创建触发器,若修改了用户信息表，则修改订单信息表。

go

create trigger 更新用户信息

on 用户信息表

for update

as

begin

declare @新姓名 char(10),@新地址 char(50), @新手机号 char(15)

declare @旧姓名 char(10),@旧地址 char(50), @旧手机号 char(15)

select @旧姓名=姓名,@旧地址=地址,@旧手机号=手机号

from deleted --旧的用户信息

select @新姓名=姓名,@新地址=地址,@新手机号=手机号

from inserted --新的用户信息

update 订单信息表 set 地址=@新地址

where @旧地址=地址

update 订单信息表 set 手机号=@新手机号

where @旧手机号=手机号

end

--触发触发器

update 用户信息表

set 姓名='无名氏',地址='上海浦东',手机号='11110000'

where 姓名='阿里宝'

--查看新的用户信息

select \* from 用户信息表

select \* from 订单信息表

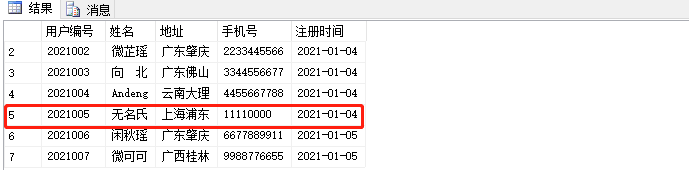


图5-5-3更新的用户信息表

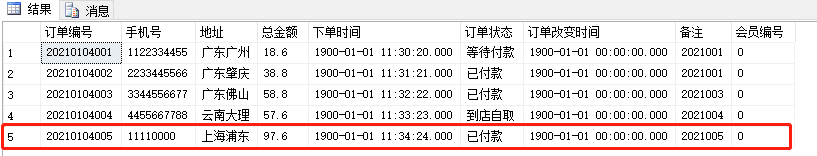


图5-5-4更新的订单信息表

--b.创建触发器，若插入新的订单，则订单项信息表中增加相应订单(inserted表)。

go

create trigger 新的订单

on 订单信息表

for insert

as

begin

declare @订单编号 char(15),@订单项编号 char(10),

@姓名 char(10),@状态 char(10)

select @订单编号=订单编号,@状态=订单状态 from inserted

insert into 订单项信息表 values('',@订单编号,'',@状态)

end

--触发触发器

insert into 订单信息表 values(20212021,198898999,'湖南长沙',

8.8,2021-01-06,'已付款',2021-01-06,'无',0001)

--查看更新的信息

select \* from 订单信息表

select \* from 订单项信息表

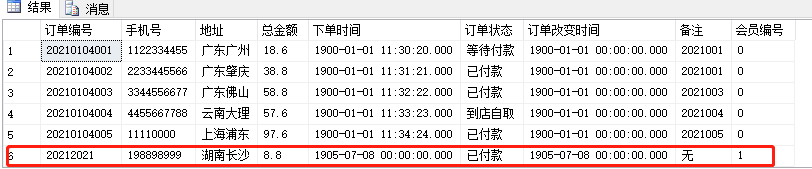


图5-5-5插入新订单的订单信息表

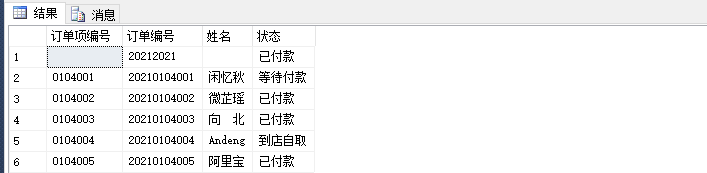


图5-5-6插入新订单的订单项信息表

\*注：这里只是简单的插入订单编号和状态，订单项编号和姓名未插入。

--c.创建触发器,若删除了订单，则修改全部营业额。

go

create trigger 取消订单

on 订单信息表

for update

as

begin

declare @更新营业额 float

select @更新营业额=总金额 from deleted

update 营业额 set 全部营业额=@更新营业额

end

--触发触发器

delete 订单项信息表 where 订单编号=20210104001

delete 订单信息表 where 订单编号=20210104001

--查看更新的信息

select \* from 订单信息表

select \* from 订单项信息表

select \* from 营业额

select 总金额 from 订单信息表

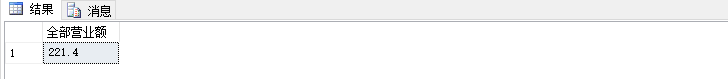


图5-5-7未删除订单前的营业额



图5-5-8被删除的订单



图5-5-9删除订单后的营业额

--d.创建触发器,若修改了菜谱信息表中的原价，则修改菜谱信息表中的优惠价。

go

create trigger 修改菜谱信息

on 菜谱信息表

for update

as

begin

declare @原价 float,@菜名 char(10)

select @原价=原价,@菜名=菜名 from inserted --更新原价

update 菜谱信息表 set 优惠价=@原价\*0.7 where 菜名=@菜名

end

--触发触发器

update 菜谱信息表 set 原价=50.0 where 菜名='土豆焖鸡'

--查看更新的信息

select \* from 菜谱信息表

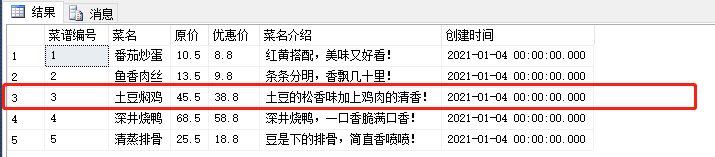


图5-5-10修改原价前的菜谱信息表

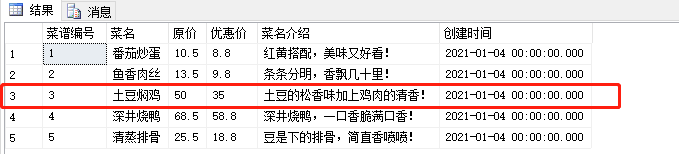


图5-5-11修改原价后的菜谱信息表

--e. 创建触发器,为外卖点餐数据库中的任一表创建一个DDL触发器limited来防止该表被修改或删除

use 外卖点餐数据库

go

create trigger 限制修改删除表 on database

for drop\_table,alter\_table

as

print '名为限制修改删除表的触发器不允许执行对该表的修改删除操作。'

rollback

--drop table 用户信息表

--alter table 菜谱信息表

--drop column 菜名

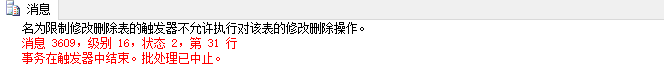


图5-5-12触发器-限制提醒

--f.创建触发器,为外卖点餐数据库中的菜谱信息表创建一个DML触发器，在插入和修改数据时，都会自动显示所有菜名的信息

use 外卖点餐数据库

go

create trigger 自动显示菜名 on 菜谱信息表

for insert,update

as

select \* from 菜谱信息表

--update 菜谱信息表

--set 菜名='番茄炒鸭蛋' where 菜名='番茄炒蛋'

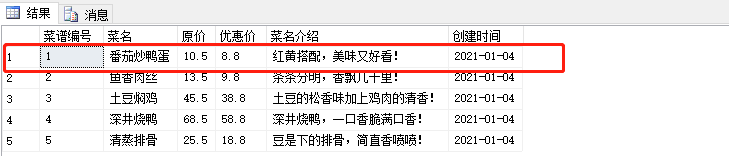


图5-5-13触发器-自动显示所有菜名

六、数据库运行和维护

数据库的运行和维护对于任何数据库管理系统来说都是至关重要的。其中包含了对数据库系统的安全性的操作，数据库的安全性是指保护数据库以防止因不合法的访问而造成数据的泄露或破坏。因此，要防止这样的情况发生，我们就必须建立对数据库的安全性机制。

与此同时，作为一名数据库管理员，无论其管理的网络的规模是大还是小，在日常的管理中除了维护网络平稳运行、及时排除网络故障、保护网络安全等工作以外，备份网络中关键数据也是其中的一个非常非常重要的工作环节。网络中的各种故障无非就分两种：软件故障和硬件故障。对于“硬件故障”可以通过维修或更换硬件设备得到及时解决；对于“软件故障”则可以通过重新安装或升级软件、重做网络或应用软件系统等方法及时解决，而且用此方法来解决网络故障大多需要一些基础的、关键的数据支持才能得以恢复正常。但是，网络中诸如此类的关键数据（特别是“应用软件系统”中的关键数据）的损坏或丢失，绝大部分是无法恢复和弥补的。所以在这里我们就必须进行对数据库的运行和维护做一些必要的操作。

6.1数据库安全性操作

1. 创建管理员角色

--创建管理员角色

begin transaction

use 外卖点餐数据库

go

create login 闲忆秋admin with password='admin102',

default\_database=外卖点餐数据库

create user 闲忆秋admin1 for login 闲忆秋admin

create role 忆秋

if @@Error=0

begin

grant select,alter,update on 菜谱信息表 to 忆秋 with grant option

grant select,alter,update on 用户信息表 to 忆秋 with grant option

grant select,alter,update on 订单信息表 to 忆秋 with grant option

grant select,alter,update on 订单项信息表 to

忆秋 with grant option

grant create view, create table to 忆秋1 with grant option

exec sp\_addrolemember'忆秋','闲忆秋admin'

print '创建管理员角色成功。'

commit

end

else

begin

print '创建管理员角色失败。'

rollback

end

1. 创建用户角色

--创建用户角色

begin transaction

use 外卖点餐数据库

go

create login 薇芷瑶admin with password='2233445566',

default\_database=外卖点餐数据库

create user 薇芷瑶admin for login 薇芷瑶admin

create role 芷瑶

if @@Error=0

begin

grant select on 菜谱信息表 to 芷瑶1 with grant option

exec sp\_addrolemember '芷瑶','薇芷瑶admin'

print '创建用户角色成功。'

commit

end

else

begin

print '创建用户角色失败。'

rollback

end

6.2数据库备份操作

1. 完整数据库备份——每周进行一次

--数据库的完整备份

backup database 外卖点餐数据库

to disk='D:\DATA\外卖点餐数据库.bak'

with format,

name= '外卖点餐数据库的完整备份'

1. 差异数据库备份——每天进行一次

--数据库的差异备份

backup database 外卖点餐数据库

to disk='D:\DATA\外卖点餐数据库差异备份.bak'

with differential