

订餐管理系统

1、应用系统背景介绍

订餐管理系统是一个旨在简化订餐流程、提高效率的应用系统。该系统将依托餐饮单位的组织结构，为用户提供在线订餐服务。餐饮单位可以是各种规模的餐厅、饭店或其他提供外卖服务的场所。

该系统的主要业务是为用户提供在线订餐功能，用户可以通过系统浏览各个餐厅的菜单并下单。用户还可以管理个人信息、查看历史订单、收藏喜爱的餐厅等。对于餐厅管理人员来说，他们可以通过系统管理菜单、接收订单、处理订单、更新营业时间等。

这个系统的经营规模可以根据餐饮单位的规模而异，可以服务于小型餐厅到大型连锁餐饮集团。主要产品和服务包括用户端的订餐 App 或网站，以及餐厅端的订单管理系统。

总的来说，订餐管理系统旨在提供便利的订餐体验，同时帮助餐饮单位提高订单处理效率和管理水平。

2、需求描述

(1) 普通用户：

管理需求：能够注册账户、登录、编辑个人信息、查看历史订单、收藏喜爱的餐厅、在线浏览菜单、下单等。

信息需求：个人账户信息、订单状态、菜单信息等。

管理业务规则：普通用户可以进行浏览、下单、收藏餐厅等操作。

(2) 餐厅管理人员：

管理需求：管理菜品、接收订单、处理订单、更新营业时间、查看订单历史、管理餐厅信息等。

信息需求：订单信息、菜品信息、餐厅信息等。

管理业务规则：餐厅管理人员需要能够及时处理订单、更新菜品和营业时间，确保订单准确无误地送达给用户。

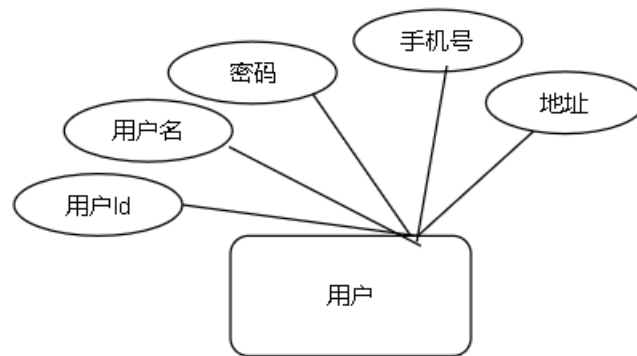
(3) 系统管理员：

管理需求：管理用户账户、管理餐厅信息、监控系统运行情况等。

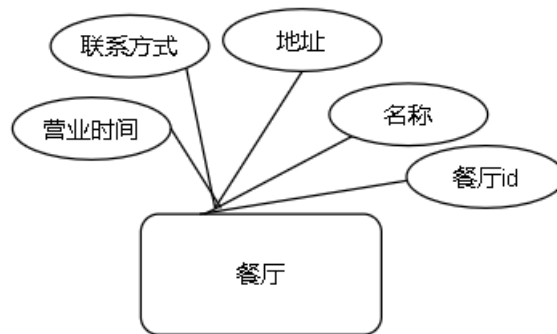
信息需求：用户信息、餐厅信息、订单信息等。

管理业务规则：系统管理员需要确保系统安全稳定运行，管理用户和餐厅信息，监控系统性能。

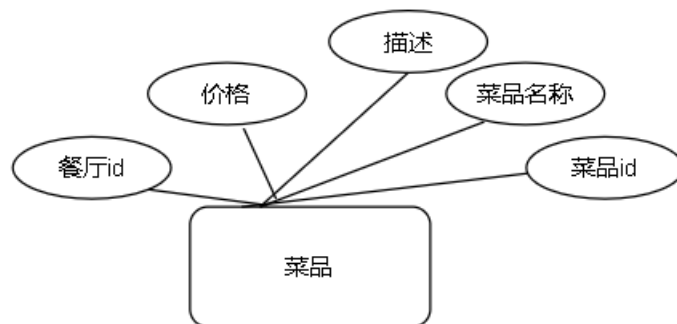
3、概念模型设计



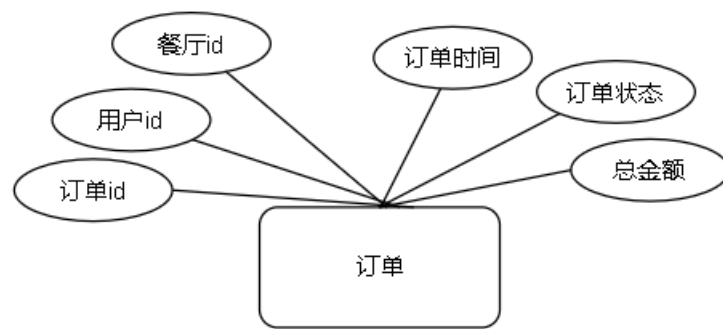
用户 ER 图



餐厅 ER 图



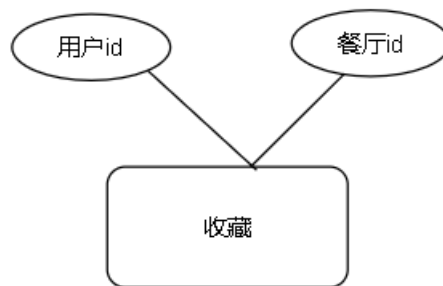
菜品 ER 图



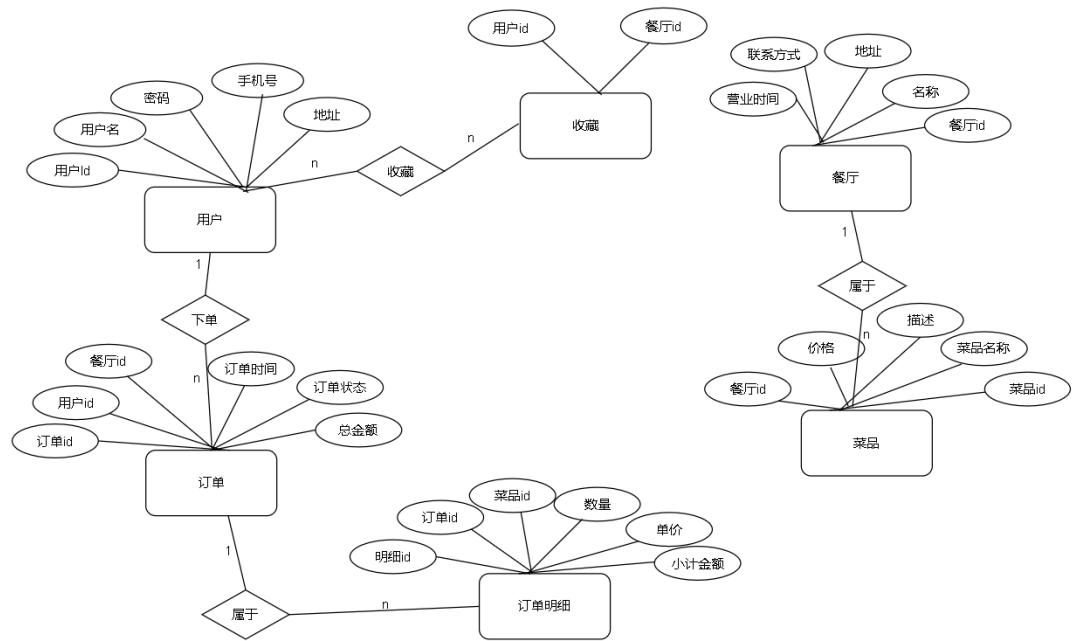
订单 ER 图



订单明细 ER 图



收藏 ER 图



整体 ER 图

用户和订单之间是一对多的关系，一个用户可以拥有多个订单，
 但一个订单只能属于一个用户。因此用户和订单之间是一对多的关系，
 用下单表示用户和订单之间的关系。

餐厅和菜品之间是一对多的关系，一个餐厅可以提供多个菜品，
 一个菜品只属于一个餐厅。因此餐厅和菜品之间是一对多的关系，用
 属于表示餐厅和菜品之间的关系。

订单和订单明细之间是一对多的关系，一个订单可以包含多个订
 单明细，一个订单明细只属于一个订单。因此订单和订单明细之间是
 一对多的关系，用属于表示订单和订单明细之间的关系。

用户和收藏表之间是多对多的关系，一个用户可以收藏多个餐厅，
 一个餐厅也可以被多个用户收藏。因此用户和收藏表之间是多对多的
 关系，用收藏表示用户和收藏表之间的关系。

4、数据库逻辑模型设计

用户表：主码 UserID

订单表：主码 OrderID，外码 UserID(来自用户表)

餐厅表：主码 RestaurantID

菜品表：主码 DishID，外码 RestaurantID（来自餐厅表）

订单明细表：主码 OrderDetailID，外码 OrderID(来自订单表)

收藏表：外码 UserID(来自用户表), 外码 RestaurantID（来自餐厅表）

5、数据库物理模型及完整性设计

(1) Users(用户表)

数据项名	数据类型	宽度及小数位	列级完整性约束	备注
UserID	int		Not Null Primary Key	主键 id
UserName	nvarchar	50	Not Null	用户名
Password	nvarchar	50	Not Null	密码
PhoneNumber	nvarchar	20		手机号
Address	nvarchar	100		地址

(2) Restaurants(餐厅表)

数据项名	数据类型	宽度及小数位	列级完整性约束	备注
RestaurantID	int		Not Null Primary Key	主键 id
RestaurantName	nvarchar	100	Not Null	餐厅名称
Address	nvarchar	100	Not Null	地址
ContactInfo	nvarchar	20	Not Null	联系方式
BusinessHours	nvarchar	50	Not Null	营业时间

(3) Dishes(菜品表)

数据项名	数据类型	宽度及小数位	列级完整性约束	备注
DishID	int		Not Null Primary Key	主键 id
DishName	nvarchar	100	Not Null	菜品名称
Description	nvarchar	255		描述
Price	decimal	10,2	Not Null	价格

RestaurantID	int		Not Null Foreign Key	对应餐厅表的 RestaurantID
--------------	-----	--	----------------------	------------------------

(4) Orders (订单表)

数据项名	数据类型	宽度及小数位	列级完整性约束	备注
OrderID	int		Not Null Primary Key	主键 id
OrderTime	datetime		Not Null	订单时间
OrderStatus	nvarchar	20	Not Null	订单状态
TotalAmount	decimal	10, 2	Not Null	订单总金额
UserID	int		Not Null Foreign Key	对应用户表的 UserID
RestaurantID	int		Not Null Foreign Key	对应餐厅表的 RestaurantID

(5) OrderDetails (订单明细表)

数据项名	数据类型	宽度及小数位	列级完整性约束	备注
OrderDetailID	int		Not Null Primary Key	主键 id
Quantity	Int		Not Null	数量
UnitPrice	decimal	10, 2	Not Null	单价
Subtotal	decimal	10, 2	Not Null	小计
OrderID	int		Not Null Foreign Key	对应订单表的 OrderID
DishID	int		Not Null Foreign Key	对应菜品表的 DishID

(6) UserFavorites (用户收藏表)

数据项名	数据类型	宽度及小数位	列级完整性约束	备注
UserID	int		Not Null Foreign Key	对应用户表的 UserID
RestaurantID	int		Not Null Foreign Key	对应餐厅表的 RestaurantID

6、数据定义语言

(1) 基本表及其完整性定义的 SQL Server 语句

-- 创建用户表

```
CREATE TABLE Users (
    UserID INT PRIMARY KEY IDENTITY(1, 1),
    UserName NVARCHAR(50),
    Password NVARCHAR(50),
    PhoneNumber NVARCHAR(20),
```

```
Address NVARCHAR(100)
);
-- 创建餐厅表
CREATE TABLE Restaurants (
RestaurantID INT PRIMARY KEY IDENTITY(1,1),
RestaurantName NVARCHAR(100),
Address NVARCHAR(100),
ContactInfo NVARCHAR(20),
BusinessHours NVARCHAR(50)
);
-- 创建菜品表
CREATE TABLE Dishes (
DishID INT PRIMARY KEY IDENTITY(1,1),
DishName NVARCHAR(100),
Description NVARCHAR(255),
Price DECIMAL(10, 2),
RestaurantID INT,
FOREIGN KEY (RestaurantID) REFERENCES Restaurants(RestaurantID)
);
-- 创建订单表
CREATE TABLE Orders (
OrderID INT PRIMARY KEY IDENTITY(1,1),
UserID INT,
RestaurantID INT,
OrderTime DATETIME,
OrderStatus NVARCHAR(20),
TotalAmount DECIMAL(10, 2),
FOREIGN KEY (UserID) REFERENCES Users(UserID),
FOREIGN KEY (RestaurantID) REFERENCES Restaurants(RestaurantID)
);
-- 创建订单明细表
CREATE TABLE OrderDetails (
OrderDetailID INT PRIMARY KEY IDENTITY(1,1),
```



```

    OrderID INT,
    DishID INT,
    Quantity INT,
    UnitPrice DECIMAL(10, 2),
    Subtotal DECIMAL(10, 2),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (DishID) REFERENCES Dishes(DishID)
);
-- 创建用户收藏表
CREATE TABLE UserFavorites (
    UserID INT,
    RestaurantID INT,
    FOREIGN KEY (UserID) REFERENCES Users(UserID),
    FOREIGN KEY (RestaurantID) REFERENCES Restaurants(RestaurantID)
);

```

(2) 基本表索引定义的 SQL Server 语句

--为 Users 表的 UserName 和 Password 创建索引

```

CREATE INDEX idx_UserName ON Users(UserName);
CREATE INDEX idx_Password ON Users(Password);

```

--为 Restaurants 表的 RestaurantName 创建索引

```

CREATE INDEX idx_RestaurantName ON Restaurants(RestaurantName);

```

--为 Dishes 表的 DishName 创建索引

```

CREATE INDEX idx_DishName ON Dishes(DishName);

```

--为 Orders 表的 OrderStatus 创建索引

```

CREATE INDEX idx_OrderStatus ON Orders(OrderStatus);

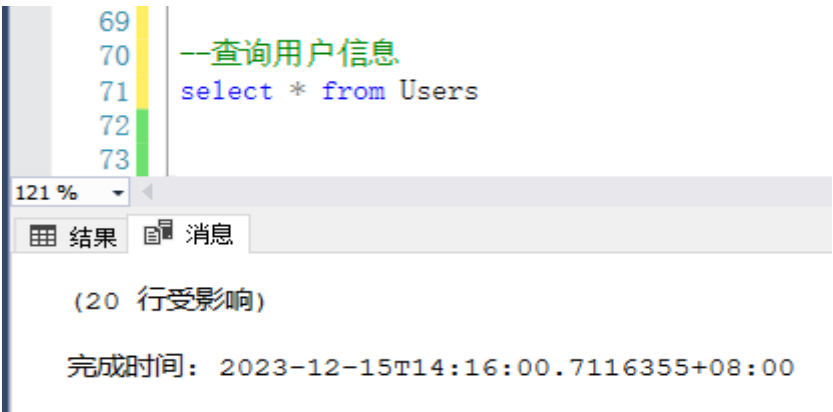
```

(3) 基本表中全部数据记录的列表截图



	UserID	UserName	Password	PhoneNumber	Address
1	1	曹操	123456	18825898745	上海市
2	2	刘备	123456	15689856985	上海市
3	3	孙权	123456	19985456523	上海市松江区
4	4	张飞	123456	15898698561	上海市
5	5	曹植	123456	18856985412	上海市
6	6	荀彧	123456	13478598564	上海市
7	7	贾诩	123456	15689854512	上海市
8	8	徐庶	123456	13745415263	上海市
9	9	郭嘉	123456	15787451256	上海市
10	10	司马懿	123456	15526358985	上海市
11	11	司马伦	123456	15874589623	上海市
12	12	司马昭	123456	18569856523	上海市
13	13	司马师	123456	18856523652	上海市
14	14	司马朗	123456	1565265895	上海市
15	15	诸葛亮	123456	15689874123	上海市
16	16	诸葛瑾	123456	15689745123	上海市
17	17	孙策	123456	15874859632	上海市
18	18	孙尚香	123456	15569874512	上海市
19	19	周瑜	123456	15789845623	上海市
20	20	鲁肃	123456	18856451252	上海市

用户表数据



(20 行受影响)
完成时间: 2023-12-15T14:16:00.7116355+08:00

运行时间

121 %	12	73	74	75	—查询餐厅表信息	select * from Restaurants
121 %	结果	消息				
	RestaurantID	RestaurantName	Address	ContactInfo	BusinessHours	
1	1	美味佳肴	上海	简介	09:00-22:00	
2	2	天堂餐厅	上海	简介	09:00-22:00	
3	3	食之源	上海	简介	09:00-22:00	
4	4	香香餐厅	上海	简介	09:00-22:00	
5	5	美食家	上海	简介	09:00-22:00	
6	6	心满意足	上海	简介	09:00-22:00	
7	7	美味家常	上海	简介	09:00-22:00	
8	8	大胃王	上海	简介	09:00-22:00	
9	9	食尚主义	上海	简介	09:00-22:00	
10	10	五星餐厅	上海	简介	09:00-22:00	
11	11	美食宫殿	上海	简介	09:00-22:00	
12	12	美食天堂	上海	简介	09:00-22:00	
13	13	食欲之旅	上海	简介	09:00-22:00	
14	14	食尚狂热	上海	简介	09:00-22:00	
15	15	美食之家	上海	简介	09:00-22:00	
16	16	食乐园	上海	简介	09:00-22:00	
17	17	食尚风情	上海	简介	09:00-22:00	
18	18	美食巡礼	上海	简介	09:00-22:00	
19	19	美食狂想曲	上海	简介	09:00-22:00	
20	20	食尚潮流	上海	简介	09:00-22:00	
21	21	食为天	上海	简介	09:00-22:00	

餐厅表数据

121 %	72	73	74	75	—查询餐厅表信息	select * from Restaurants
121 %	结果	消息				

(21 行受影响)

完成时间: 2023-12-15T14:30:36.6500652+08:00

运行时间

76

77

78

79

--查询菜品表数据

select * from Dishes

121 %

结果

消息

	DishID	DishName	Description	Price	RestaurantID
1	1	酸辣土豆丝	描述	10.00	1
2	2	西红柿炒蛋	描述	12.00	1
3	3	辣椒炒肉	描述	12.00	1
4	4	麻辣豆腐	描述	12.00	1
5	5	香干肉丝	描述	12.00	1
6	6	红烧鲫鱼	描述	22.00	2
7	7	干煸四季豆	描述	12.00	3
8	8	韭菜炒蛋	描述	12.00	1
9	9	拍黄瓜	描述	12.00	1
10	10	千张豆腐	描述	12.00	2
11	11	豆米萝卜干	描述	12.00	1
12	12	外婆菜	描述	15.00	3
13	13	清炒藕片	描述	12.00	1
14	14	辣子鸡丁	描述	19.00	1
15	15	宫保鸡丁	描述	12.00	4
16	16	清蒸鱼	描述	32.00	1
17	17	油菜虾仁	描述	22.00	1
18	18	红烧茄	描述	12.00	5
19	19	红烧排骨	描述	18.00	1
20	20	红烧鲤鱼	描述	12.00	1
21	21	凉拌西红柿	描述	12.00	6
22	22	凉拌豆角	描述	12.00	2
23	23	炒西瓜皮	描述	12.00	1
24	24	豆角炒黄瓜	描述	12.00	7
25	25	辣椒炒鸡蛋	描述	12.00	9

菜品表

76	--查询菜品表数据
77	select * from Dishes
78	
79	
121 %	
结果 消息	

(25 行受影响)

完成时间：2023-12-15T14:41:57.9809311+08:00

运行时间

79 --查询订单表数据
80 select * from Orders
81

121 %

结果 消息

	OrderID	UserID	RestaurantID	OrderTime	OrderStatus	TotalAmount
1	1	1	1	2023-12-15 08:22:56.000	预定	22.00
2	2	2	2	2023-12-16 12:20:22.000	预定	32.00
3	3	3	3	2023-12-16 13:22:22.000	配送	32.00
4	4	4	3	2023-12-16 13:28:22.000	配送	42.00
5	5	5	4	2023-12-16 13:22:22.000	完成	56.00
6	6	6	4	2023-12-16 14:22:22.000	预定	86.00
7	7	7	5	2023-12-16 13:22:22.000	配送	29.00
8	8	8	6	2023-12-16 13:22:22.000	完成	28.00
9	9	9	7	2023-12-16 14:22:22.000	预定	89.00
10	10	10	1	2023-12-15 13:22:22.000	完成	129.00
11	11	11	2	2023-12-16 13:22:22.000	完成	229.00
12	12	12	2	2023-12-16 19:22:22.000	预定	229.00
13	13	13	3	2023-12-14 13:22:22.000	完成	32.00
14	14	14	4	2023-12-16 14:22:22.000	预定	36.00
15	15	15	1	2023-12-12 13:22:22.000	完成	62.00
16	16	16	1	2023-12-16 13:22:22.000	完成	29.00
17	17	17	2	2023-12-16 18:22:22.000	预定	32.00
18	18	18	3	2023-12-12 13:22:22.000	预算	62.00
19	19	19	4	2023-12-16 13:22:22.000	预定	55.00
20	20	20	5	2023-12-16 13:22:22.000	预定	88.00

查询订单表数据

79 --查询订单表数据
80 select * from Orders
81

121 %

结果 消息

(20 行受影响)

完成时间: 2023-12-15T14:43:33.1568317+08:00

运行时间

```
82 | --查询订单明细表
83 | select * from OrderDetails
84 |
```

121 %

	OrderDetailID	OrderID	DishID	Quantity	UnitPrice	Subtotal
1	1	1	1	1	10.00	10.00
2	2	1	2	1	12.00	12.00
3	3	2	1	2	10.00	20.00
4	4	2	2	1	12.00	12.00
5	5	3	3	1	12.00	12.00
6	6	4	4	1	12.00	12.00
7	7	5	1	1	12.00	12.00
8	8	5	2	1	12.00	12.00
9	9	6	3	1	12.00	12.00
10	10	7	4	1	12.00	12.00
11	11	8	4	1	12.00	12.00
12	12	9	5	1	12.00	12.00
13	13	10	2	1	12.00	12.00
14	14	11	3	1	12.00	12.00
15	15	12	4	1	12.00	12.00
16	16	13	5	1	12.00	12.00
17	17	14	6	1	12.00	12.00
18	18	4	4	1	12.00	12.00
19	19	5	4	1	12.00	12.00
20	20	6	5	1	12.00	12.00

查询订单明细表数据

```
82 | --查询订单明细表
83 | select * from OrderDetails
84 |
```

121 %

	OrderDetailID	OrderID	DishID	Quantity	UnitPrice	Subtotal
--	---------------	---------	--------	----------	-----------	----------

(20 行受影响)

完成时间: 2023-12-15T14:44:50.6602289+08:00

查询时间

```
85 | --查询用户收藏表数据
86 | select * from UserFavorites
87 |
```

121 %

	UserID	RestaurantID
1	1	1
2	1	2
3	1	3
4	1	4
5	2	1
6	2	3
7	2	4
8	2	5
9	3	1
10	3	2
11	3	3
12	4	1
13	4	2
14	4	5
15	5	1
16	5	2
17	6	1
18	7	1
19	7	2
20	8	1
21	8	2
22	8	3
23	9	1
24	9	2

查询用户收藏数据

```
85 | --查询用户收藏表数据
86 | select * from UserFavorites
87 |
```

121 %

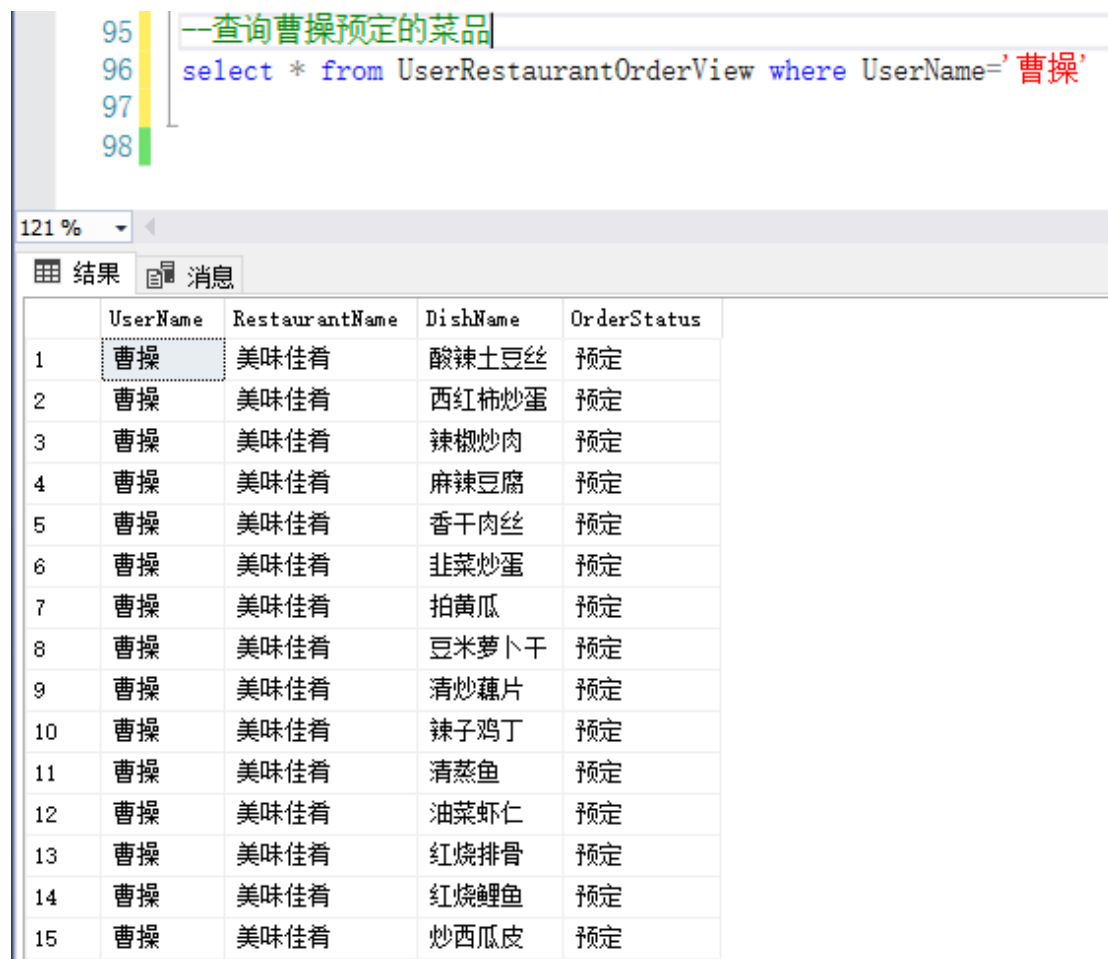
结果	消息
(24 行受影响)	
完成时间: 2023-12-15T14:46:26.0366135+08:00	

查询时间

(4) 视图的定义 Sql Server 语句

--创建视图查询用户姓名，餐厅名称，菜品名称，订单状态

```
CREATE VIEW UserRestaurantOrderView AS
SELECT Users.UserName, Restaurants.RestaurantName, Dishes.DishName,
Orders.OrderStatus
FROM Orders
JOIN Restaurants ON Orders.RestaurantID=Restaurants.RestaurantID
JOIN Dishes ON Dishes.RestaurantID = Orders.RestaurantID
JOIN Users ON Users.UserID = Orders.UserID;
```



The screenshot shows a SQL query window with the following text:

```
--查询曹操预定的菜品
select * from UserRestaurantOrderView where UserName='曹操'
```

Below the query window, the 'Results' tab is selected, displaying a table with 5 columns: 'UserName', 'RestaurantName', 'DishName', and 'OrderStatus'. The table contains 15 rows of data, all for the user '曹操' (Cao Cao) at '美味佳肴' (Mei Wei Jia Xiang) restaurant, with an 'OrderStatus' of '预定' (Reserved).

	UserName	RestaurantName	DishName	OrderStatus
1	曹操	美味佳肴	酸辣土豆丝	预定
2	曹操	美味佳肴	西红柿炒蛋	预定
3	曹操	美味佳肴	辣椒炒肉	预定
4	曹操	美味佳肴	麻辣豆腐	预定
5	曹操	美味佳肴	香干肉丝	预定
6	曹操	美味佳肴	韭菜炒蛋	预定
7	曹操	美味佳肴	拍黄瓜	预定
8	曹操	美味佳肴	豆米萝卜干	预定
9	曹操	美味佳肴	清炒藕片	预定
10	曹操	美味佳肴	辣子鸡丁	预定
11	曹操	美味佳肴	清蒸鱼	预定
12	曹操	美味佳肴	油菜虾仁	预定
13	曹操	美味佳肴	红烧排骨	预定
14	曹操	美味佳肴	红烧鲤鱼	预定
15	曹操	美味佳肴	炒西瓜皮	预定

7、查询操作

（1）反映管理需求且符合业务规则的查询访问操作请求表述

当用户需要查询订餐系统中的订单信息、菜品信息或者用户信息时，可以通过执行相应的 SQL 查询语句来实现。例如，用户可以查询某个用户的订单数量、查询订单状态为“完成”的订单、查询每个用户的订单总金额等。这些查询操作可以满足用户对订单数据的管理需求，并且符合业务规则。同时，用户还可以查询菜品表中的所有菜品名称和价格，查询每个餐厅的菜品数量等，以满足对菜品信息的管理需求。

（2）实现查询访问请求的 SQL Server 语句

--查询所有菜品名称和价格

```
SELECT DishName, Price FROM Dishes;
```

98 一查询所有菜品名称和价格

99 SELECT DishName, Price FROM Dishes;

121 %

结果 消息

	DishName	Price
1	酸辣土豆丝	10.00
2	西红柿炒蛋	12.00
3	辣椒炒肉	12.00
4	麻辣豆腐	12.00
5	香干肉丝	12.00
6	红烧鲫鱼	22.00
7	干煸四季豆	12.00
8	韭菜炒蛋	12.00
9	拍黄瓜	12.00
10	千张豆腐	12.00
11	豆米萝卜干	12.00
12	外婆菜	15.00
13	清炒藕片	12.00
14	辣子鸡丁	19.00
15	宫保鸡丁	12.00
16	清蒸鱼	32.00
17	油菜虾仁	22.00
18	红烧茄	12.00
19	红烧排骨	18.00
20	红烧鲤鱼	12.00
21	凉拌西红柿	12.00
22	凉拌豆角	12.00
23	炒西瓜皮	12.00
24	豆角炒黄瓜	12.00
25	辣椒炒鸡蛋	12.00

查询结果

查询所有订单状态为“完成”的订单

SELECT * FROM Orders WHERE OrderStatus = '完成';

101

102

103

104

一查询所有订单状态为“完成”的订单

SELECT * FROM Orders WHERE OrderStatus = '完成';

121 %

结果

消息

	OrderID	UserID	RestaurantID	OrderTime	OrderStatus	TotalAmount
1	5	5	4	2023-12-16 13:22:22.000	完成	56.00
2	8	8	6	2023-12-16 13:22:22.000	完成	28.00
3	10	10	1	2023-12-15 13:22:22.000	完成	129.00
4	11	11	2	2023-12-16 13:22:22.000	完成	229.00
5	13	13	3	2023-12-14 13:22:22.000	完成	32.00
6	15	15	1	2023-12-12 13:22:22.000	完成	62.00
7	16	16	1	2023-12-16 13:22:22.000	完成	29.00

查询结果

--查询用户“张飞”订购的菜品名称和对应的餐厅名称

```
SELECT Users.UserName, Dishes.DishName, Restaurants.RestaurantName
FROM Users
JOIN Orders ON Users.UserID = Orders.UserID
JOIN Dishes ON Orders.RestaurantID = Dishes.RestaurantID
JOIN Restaurants ON Dishes.RestaurantID = Restaurants.RestaurantID where
UserName='张飞';
```

121 %

结果

消息

	UserName	DishName	RestaurantName
1	张飞	干煸四季豆	食之源
2	张飞	外婆菜	食之源

查询结果

--查询每个用户的订单数量

```
SELECT Users.UserName, COUNT(Orders.OrderID) AS OrderCount
FROM Users
LEFT JOIN Orders ON Users.UserID = Orders.UserID
GROUP BY Users.UserName;
```

结果 消息		
	UserName	OrderCount
1	曹操	4
2	曹植	1
3	郭嘉	1
4	贾诩	1
5	刘备	5
6	鲁肃	0
7	司马朗	1
8	司马伦	1
9	司马师	0
10	司马懿	0
11	司马昭	1
12	孙策	1
13	孙权	1
14	孙尚香	0
15	徐庶	0
16	荀彧	0
17	张飞	1
18	周瑜	1
19	诸葛瑾	1
20	诸葛亮	0

查询结果

--查询订单金额最高的订单信息

```
SELECT * FROM Orders WHERE TotalAmount = (SELECT MAX(TotalAmount) FROM Orders);
```

结果 消息						
	OrderID	UserID	RestaurantID	OrderTime	OrderStatus	TotalAmount
1	11	11	2	2023-12-16 13:22:22.000	完成	229.00
2	12	12	2	2023-12-16 19:22:22.000	预定	229.00

查询结果

--查询订购了最贵菜品的用户信息

```
SELECT * FROM Users WHERE UserID IN ((SELECT DishID FROM Dishes WHERE Price = (SELECT MAX(Price) FROM Dishes)));
```

119
120
121
122

--查询订购了最贵菜品的用户信息
SELECT * FROM Users WHERE UserID IN ((SELECT DishID FROM Dishes WHERE Price = (SELECT MAX(Price) FROM Dishes)));

121 %

结果

消息

	UserID	UserName	Password	PhoneNunber	Address
1	16	诸葛瑾	123456	15689745123	上海市

查询结果

--查询每个餐厅的菜品数量

```
SELECT Restaurants.RestaurantName, COUNT(Dishes.DishID) AS DishCount
FROM Restaurants
LEFT JOIN Dishes ON Restaurants.RestaurantID = Dishes.RestaurantID
GROUP BY Restaurants.RestaurantName;
```

121 %

结果

消息

	RestaurantName	DishCount
1	大胃王	0
2	美食宫殿	0
3	美食家	1
4	美食狂想曲	0
5	美食天堂	0
6	美食巡礼	0
7	美食之家	0
8	美味佳肴	15
9	美味家常	1
10	食乐园	0
11	食尚潮流	0
12	食尚风情	0
13	食尚狂热	0
14	食尚主义	1
15	食为天	0
16	食欲之旅	0
17	食之源	2
18	天堂餐厅	3
19	五星餐厅	0
20	香香餐厅	1
21	心满意足	1

查询结果

--查询每个用户的订单总金额

```
SELECT Users.UserName, SUM(Orders.TotalAmount) AS TotalAmount
FROM Users
LEFT JOIN Orders ON Users.UserID = Orders.UserID
GROUP BY Users.UserName;
```

结果		
	UserName	TotalAmount
1	曹操	275.00
2	曹植	56.00
3	郭嘉	89.00
4	贾诩	29.00
5	刘备	266.00
6	鲁肃	NULL
7	司马朗	36.00
8	司马伦	229.00
9	司马师	NULL
10	司马懿	NULL
11	司马昭	229.00
12	孙策	32.00
13	孙权	32.00
14	孙尚香	NULL
15	徐庶	NULL
16	荀彧	NULL
17	张飞	42.00
18	周瑜	55.00
19	诸葛瑾	29.00
20	诸葛亮	NULL

查询结果

8、添加，删除，修改操作

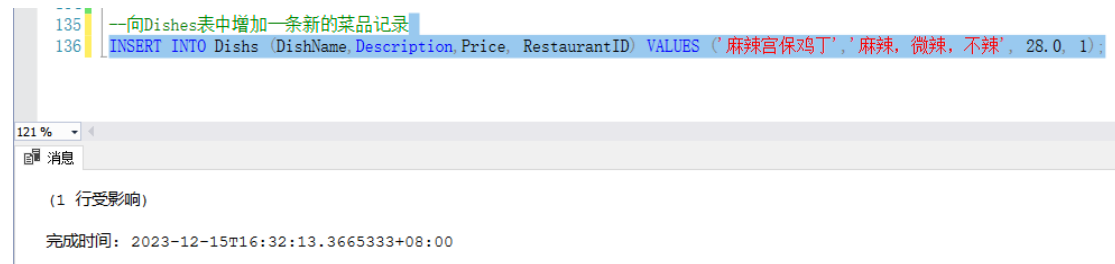
(1) 反映管理需求且符合业务规则的更新访问请求表述

当用户需要对订餐管理系统中的数据进行更新操作时，可以执行一系列的SQL更新语句来实现。例如，用户可以向菜品表中增加一条新的菜品记录，向用户表中增加一个新用户记录，或者删除订单详细表中特定的记录。这些更新操作可以满足用户对数据的管理需求，并且符合业务规则。此外，用户还可以修改菜品表中的菜品价格，修改用户表中的地址信息等，以满足对数据的更新需求。这些更新操作能够帮助用户更好地管理订餐系统中的数据。

(2) 实现更新访问请求的 SQL Server 语句

--向 Dishes 表中增加一条新的菜品记录

```
INSERT INTO Dishes (DishName,Description,Price, RestaurantID) VALUES ('麻辣宫保鸡丁','麻辣, 微辣, 不辣', 28.0, 1);
```



The screenshot shows a SQL query window with the following text:

```
--向Dishes表中增加一条新的菜品记录
INSERT INTO Dishes (DishName,Description,Price, RestaurantID) VALUES ('麻辣宫保鸡丁','麻辣, 微辣, 不辣', 28.0, 1);
```

Below the query window, the execution results are displayed:

121 %
消息
(1 行受影响)
完成时间: 2023-12-15T16:32:13.3665333+08:00

新增执行结果

--向 Users 表中增加一个新用户记录

```
INSERT INTO Users VALUES ('周润发', '123456', '13702365874', '上海静安区');
```



The screenshot shows a SQL query window with the following text:

```
--向Users表中增加一个新用户记录
INSERT INTO Users VALUES ('周润发', '123456', '13702365874', '上海静安区');
```

Below the query window, the execution results are displayed:

121 %
消息
(1 行受影响)
完成时间: 2023-12-15T16:39:47.2807159+08:00

新增执行结果

--从 OrderDetails 表中删除用户 OrderDetailID 为 5 的记录

```
DELETE FROM OrderDetails WHERE OrderDetailID = 5;
```

```
141 | --从OrderDetails表中删除用户OrderDetailID为5的记录
142 | DELETE FROM OrderDetails WHERE OrderDetailID = 5;
```

121 %

消息

(1 行受影响)

完成时间: 2023-12-15T16:46:31.7864632+08:00

删除运行结果

--删除收藏表中用户 id 为 1 的所有收藏信息

DELETE FROM UserFavorites WHERE UserID= 1;

```
144 | --删除收藏表中用户id为1的所有收藏信息
145 | DELETE FROM UserFavorites WHERE UserID= 1;
```

21 %

消息

(4 行受影响)

完成时间: 2023-12-15T16:50:10.3224777+08:00

删除运行结果

--将 Dishes 表中 ID 为 3 的菜品价格修改为 35.0

UPDATE Dishes SET Price = 35.0 WHERE DishID = 3;


```
147 | --将Dishes表中ID为3的菜品价格修改为35.0
148 | UPDATE Dishes SET Price = 35.0 WHERE DishID = 3;
```

121 %

消息

(1 行受影响)

完成时间: 2023-12-15T16:54:33.2998193+08:00

更新执行结果

--修改 Users 表中刘备的地址为“北京”

UPDATE Users SET Address='北京' WHERE UserName = '刘备';

```
150 | --修改Users表中刘备的地址为“北京”
151 | UPDATE Users SET Address='北京' WHERE UserName = '刘备';
```

21 %

消息

(1 行受影响)

完成时间: 2023-12-15T16:58:17.4786483+08:00

更新执行结果

--更新视图

```
CREATE OR ALTER VIEW UserRestaurantOrderView AS
SELECT TOP 10 Users.UserName, Restaurants.RestaurantName,
Dishes.DishName, Orders.OrderStatus
FROM Orders
JOIN Restaurants ON Orders.RestaurantID=Restaurants.RestaurantID
JOIN Dishes ON Dishes.RestaurantID = Orders.RestaurantID
JOIN Users ON Users.UserID = Orders.UserID;
```

9、触发器

(1) 业务规则描述

假设菜品的价格发生变化时,我们也希望订单明细对应的单价和小计总价也被修

改，这是一个常见的业务规则，我们希望通过触发器来实现这一流程。

（2）触发器控制逻辑的说明

触发器将会在菜品单价发生变化时，然后调用更新订单明细的价格和小计总价的操作。

（3）触发器定义的 SQL Server 语句

```
CREATE OR ALTER TRIGGER trg_update_unit_price_and_subtotal
ON Dishes
AFTER UPDATE
AS
BEGIN
    -- 更新订单详细表中对应菜品的单价和小计
    UPDATE OrderDetails
    SET UnitPrice = i.Price,
        Subtotal = i.Price * Quantity
    FROM OrderDetails od
    INNER JOIN inserted i ON od.DishID = i.DishID
    INNER JOIN deleted d ON od.DishID = d.DishID
    WHERE od.DishID = i.DishID
    AND od.UnitPrice = d.Price;
END;
```

（4）执行过程和结果

```
select *,GETDATE() as 系统运行时间 from Dishes where DishID=1
select *,GETDATE() as 系统运行时间 from OrderDetails where
DishID=1
```

```
update Dishes set Price=111 where DishID=1
```

```

181 select *,GETDATE() as 系统运行时间 from Dishes where DishID=1
182 select *,GETDATE() as 系统运行时间 from OrderDetails where DishID=1
183
184 update Dishes set Price=111 where DishID=1
185
186

```

109 %

结果 消息

	DishID	DishName	Description	Price	RestaurantID	系统运行时间
1	1	酸辣土豆丝	描述	10.00	1	2023-12-15 22:22:06.703

	OrderDetailID	OrderID	DishID	Quantity	UnitPrice	Subtotal	系统运行时间
1	1	1	1	1	10.00	10.00	2023-12-15 22:22:06.703
2	3	2	1	2	10.00	20.00	2023-12-15 22:22:06.703
3	7	5	1	1	12.00	12.00	2023-12-15 22:22:06.703

执行前

```

183
184 update Dishes set Price=111 where DishID=1
185
186

```

109 %

消息

(2 行受影响)

(1 行受影响)

执行后

```

180
181 select *,GETDATE() as 系统运行时间 from Dishes where DishID=1
182 select *,GETDATE() as 系统运行时间 from OrderDetails where DishID=1
183
184 update Dishes set Price=111 where DishID=1
185
186

```

109 %

结果 消息

	DishID	DishName	Description	Price	RestaurantID	系统运行时间
1	1	酸辣土豆丝	描述	111.00	1	2023-12-15 22:22:38.220

	OrderDetailID	OrderID	DishID	Quantity	UnitPrice	Subtotal	系统运行时间
1	1	1	1	1	111.00	111.00	2023-12-15 22:22:38.220
2	3	2	1	2	111.00	222.00	2023-12-15 22:22:38.220
3	7	5	1	1	12.00	12.00	2023-12-15 22:22:38.220

执行后

10、事务

(1) 业务规则描述

在这个场景中，我们的业务规则修改菜品表 (Dishs) 的单价 (Price) 和订单详细 (OrderDetails) 中对应菜品的单价 (UnitPrice)、小计 (Subtotal)。只有同时修改成功才算成功，若有一个失败则回滚，从而保持数据的一致性。

(2) 事务处理逻辑说明

当执行修改菜品表单价的操作时，我们需要确保以下操作作为一个事务进行处理：首先，执行菜品表单价的更新操作。然后，根据更新后的菜品单价，更新订单详细表中对应菜品的单价和小计。

(3) 事务定义的 SQL Server 语句

```
BEGIN TRANSACTION;
BEGIN TRY
    -- 更新菜品表的单价
    UPDATE Dishs
    SET Price = 10
    WHERE DishID=5;

    -- 更新订单详细表中对应菜品的单价和小计
    UPDATE OrderDetails
    SET UnitPrice = 10,
        Subtotal = 10* Quantity
    WHERE DishID = 5;

    COMMIT TRANSACTION; -- 提交事务
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION; -- 回滚事务
END CATCH;
```

(4) 事务测试过程及结果的屏幕截图

200	
207	select *,GETDATE() as 系统运行时间 from Dishes where DishID=5
208	select *,GETDATE() as 系统运行时间 from OrderDetails where DishID=5
209	
210	
211	
212	

39 %

结果 消息

	DishID	DishName	Description	Price	RestaurantID	系统运行时间
1	5	香干肉丝	描述	111.00	1	2023-12-16 09:18:48.883

	OrderDetailID	OrderID	DishID	Quantity	UnitPrice	Subtotal	系统运行时间
1	12	9	5	1	111.00	111.00	2023-12-16 09:18:48.883
2	16	13	5	1	111.00	111.00	2023-12-16 09:18:48.883
3	20	6	5	1	111.00	111.00	2023-12-16 09:18:48.883

事务执行前

185	
186	--事务
187	BEGIN TRANSACTION;
188	BEGIN TRY
189	-- 更新菜品表的单价
190	UPDATE Dishes
191	SET Price = 111
192	WHERE DishID=5;
193	
194	-- 根据更新后的菜品单价，更新订单详细表中对应菜品的单价和小计
195	UPDATE OrderDetails
196	SET UnitPrice = 111,
197	Subtotal = 111 * Quantity

109 %

消息

(3 行受影响)
(1 行受影响)
(3 行受影响)

事务执行后

206	--
207	select *,GETDATE() as 系统运行时间 from Dishes where DishID=5
208	select *,GETDATE() as 系统运行时间 from OrderDetails where DishID=5
209	
210	
211	
212	

109 %

结果

消息

	DishID	DishName	Description	Price	RestaurantID	系统运行时间
1	5	香干肉丝	描述	111.00	1	2023-12-16 09:19:21.947

	OrderDetailID	OrderID	DishID	Quantity	UnitPrice	Subtotal	系统运行时间
1	12	9	5	1	111.00	111.00	2023-12-16 09:19:21.947
2	16	13	5	1	111.00	111.00	2023-12-16 09:19:21.947
3	20	6	5	1	111.00	111.00	2023-12-16 09:19:21.947

事务执行后

11、总结

订餐管理系统包括了用户表、餐厅表、菜品表、订单表、订单明细表 and 用户收藏表。每个表都有相应的字段来存储相关信息，并且表与表之间建立了关联，保证了数据的一致性和完整性。此外，为了提高查询性能，对一些常用的字段创建了索引。

除了表的设计，系统还包括了视图，用于查询用户姓名、餐厅名称、菜品名称和订单状态，方便了对这些信息的查询。同时，为了保证数据的完整性，创建了一个触发器，当菜品表的单价发生变化时，订单详细表中对应菜品的单价和小计也会相应更新。

另外，系统还实现了事务处理，确保了在更新菜品表的单价后，订单详细表中对应菜品的单价和小计能够正确更新，从而保证了数据的一致性。

总的来说，这个订餐管理系统设计完善，实现了基本的订餐业务功能，并且在数据一致性和查询性能方面做了一些优化。