网上订餐系统

**数据库(顶层)设计说明**

**版本：1.0**

编写： 任聪豪

校对： 郭 峰、马亦骁

审核： 张 倩

批准： 李煜寒

**2019年7月**

**目 录**

[1引言 3](#_Toc12980960)

[1.1文档标识 3](#_Toc12980961)

[1.2数据库概述 3](#_Toc12980962)

[1.3文档概述 4](#_Toc12980963)

[2引用文件 4](#_Toc12980964)

[3数据库级设计决策 4](#_Toc12980965)

[3.1数据库管理系统 4](#_Toc12980966)

[3.2数据库备份与恢复 4](#_Toc12980967)

[4数据库详细设计 5](#_Toc12980968)

[4.1数据库概念设计 5](#_Toc12980969)

[4.2数据库逻辑设计 6](#_Toc12980970)

[4.2.1 数据库表的说明 6](#_Toc12980971)

[4.2.2 数据库表关系说明 6](#_Toc12980972)

[4.2.3 数据库表的属性说明 7](#_Toc12980973)

[4.3数据库物理设计 10](#_Toc12980974)

[4.3.1 建表SQL 10](#_Toc12980975)

[5需求的可追踪性 16](#_Toc12980976)

[6注释 16](#_Toc12980977)

# 1引言

## 1.1文档标识

中文名称：《数据库（顶层）设计说明》

英文名称：“Database Design Description（DBDD）”

文档版本：“1.0”

文档编号：“OR-DBDD-1.0(E)”

## 1.2数据库概述

随着互联网的发展以及人们生活节奏的加快，服务业相关产业逐渐进行信息化、便捷化，而餐饮作为服务业的重要部分，开发网上订餐系统有着广阔的前景。该网上订餐系统为客户提供了基于Web的菜品浏览、下单、支付、配送一条龙服务。该系统能够为客户以及餐饮商家提供便利、使商家能够更加快速与准确的分析客户喜好，提升利润。

本系统采用MySQL数据库，通过Java JDBC连接MySQL数据库进行数据持久化。本系统的数据库需要支持网上订餐系统中用户信息、订单信息、菜品信息的持久化。

|  |  |
| --- | --- |
| 数据库名称 | Restaurant |
| 数据库系统 | MySQL 8.0 |
| 数据库宿主环境 | Windows 10 |
| 客户端连接工具 | JDBC |
| 需求方 | 网上订餐系统 |
| 开发方 | 西北工业大学 |

## 1.3文档概述

本文档依据国家标准《GB/T 8567-2006计算机软件文档编制规范》制定，属于技术文档。本文档记录了网上订餐系统的数据库设计过程、数据库ER图以及数据库中各表的属性值及约束。本文档用于开发人员进行项目设计，同时也为后续的数据库维护在提供了良好的使用说明，可也作为未来版本升级时的重要参考资料。

# 2引用文件

《GB/T 8567-2006计算机软件文档编制规范》

《OR -SDP-1.0(E) 软件开发计划》

# 3数据库级设计决策

## 3.1数据库管理系统

|  |  |
| --- | --- |
| 名称 | 版本 |
| MySQL | mysql-8.0.13-winx64 |

## 3.2数据库备份与恢复

数据库备份采用直接拷贝数据库文件的方法，频率为每天备份一次。备份数据库时应在每天0：00关闭系统时备份，备份完毕后重启服务器。数据库恢复通过备份文件或更新日志来进行恢复。

# 4数据库详细设计

## 4.1数据库概念设计

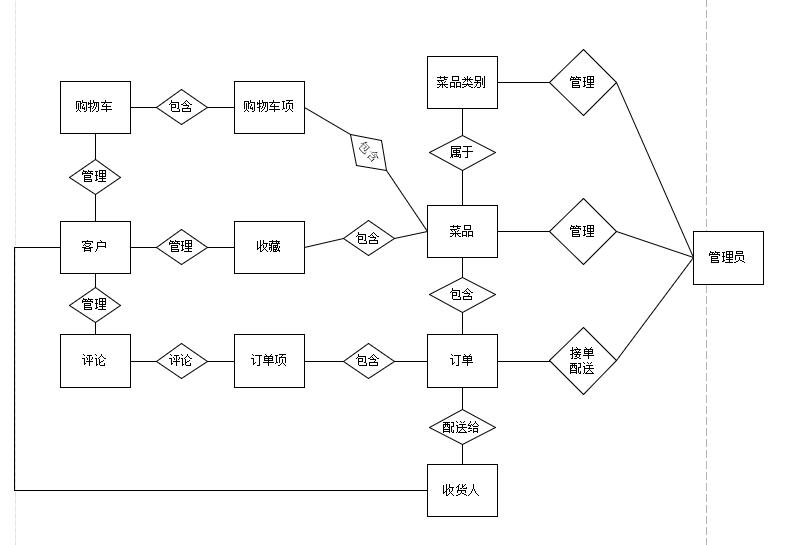


图 1 数据库ER图

## 4.2数据库逻辑设计

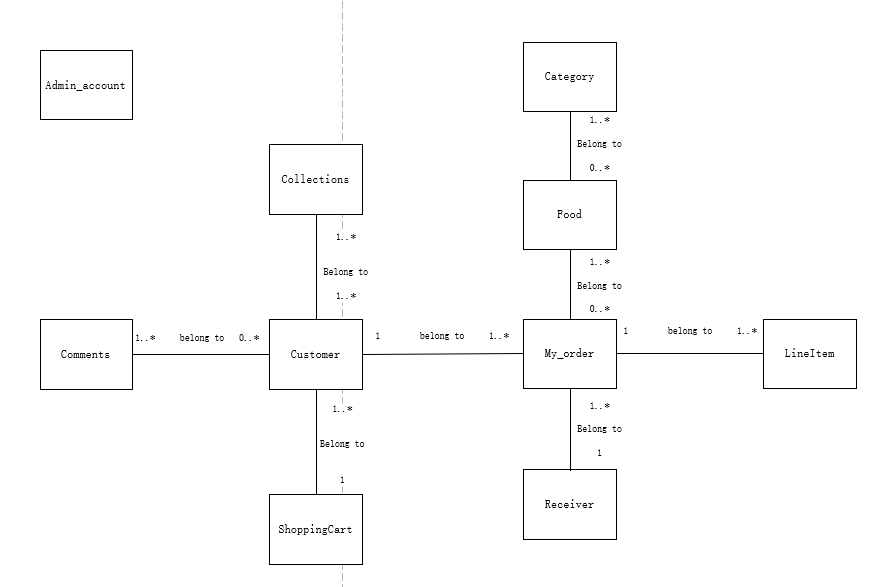


图 2 数据库ER图2

### 4.2.1 数据库表的说明

Admin\_account为存储管理员账户信息的实体类；Customer为存储客户账户信息的实体类；receiver为存储收货人信息的实体类； Food为存储食品信息的实体类；category为存储食品分类信息的实体类；food\_belong为存储食品及其所属类别的实体类；my\_order为存储客户订单信息的实体类；Lineitem存储Order中的食品数量、价格信息，ShoppingCart存储客户的购物车中的食品信息；comments为存储客户对食品进行的评论的实体类；collections为存储客户收藏食品的实体类。

### 4.2.2 数据库表关系说明

1. 一个Customer对应0到多个Order，一个Order对应一个Customer，因为一个客户可以没有订单，也可以有多个订单，一个Order必须对应一个Customer。
2. 一个Customer对应0到多条ShoppingCart信息，一条ShoppingCart信息对应一个Customer，因为一个客户的购物车中可以没有食品，也可以有多个食品，一条ShoppingCart信息必须对应一个Customer。
3. 一条ShoppingCart信息对应一条Food信息，一条Food信息可以对应0到多个ShoppingCart信息，因为每种食品都可以被多个客户添加购物车，而每条购物车中的食品信息只能对应一种食品。
4. 一个Order对应1到多个Lineitem，一条Lineitem只能对应一个Order，因为一个Order中至少有一种食品，可以有多种食品，而一条订单项只能存在于一个Order中。
5. 一条Lineitem信息对应一种Food，一种Food可以对应0到多个Lineitem，因为一种Food可以从未被添加到订单中，也可以被添加到多个订单中，一条订单中的食物信息只能对应一种食物。
6. 一种Food可以属于多个食品分类，一个食品分类中可以有0到多个食品，因为食品分类直接可能存在交集，并且在创建新的食品类别时该类可以没有具体的食品。
7. 一个Customer可以对多个菜品进行评论，一个菜品可以被多个客户评论
8. 一个Customer可以对多个菜品进行收藏，一个菜品可以被多个客户收藏

### 4.2.3 数据库表的属性说明

#### 4.2.3.1表admin\_account

|  |  |
| --- | --- |
| 属性 | 说明 |
| admin\_id | 主码，管理员的id（登录时使用） |
| ad\_tel | 管理员的电话号码 |
| password | 管理员的账户密码 |
| issuper | 管理员是否为超级管理员 |

#### 4.2.3.2表customer

|  |  |
| --- | --- |
| 属性 | 说明 |
| c\_id | 主码，客户登录时使用 |
| c\_tel | 客户的电话号码 |
| c\_password | 客户的账户密码 |
| email | 客户的电子邮件地址 |

#### 4.2.3.3表receiver

|  |  |
| --- | --- |
| 属性 | 说明 |
| r\_id | 主码，收货人的id |
| c\_id | 外键，客户的id |
| r\_name | 收货人的姓名 |
| address | 收货人的地址 |
| r\_tel | 收货人的电话号码 |
| isshow | 该收货人是否显示在客户的选择列表中 |

#### 4.2.3.4表food

|  |  |
| --- | --- |
| 属性 | 说明 |
| food\_id | 主码，食品的id |
| fd\_name | 食品的名称 |
| fd\_subname | 食品的别称或副标题 |
| fd\_image | 食品的图片路径 |
| fd\_price | 食品的价格 |
| fd\_description | 食品的描述 |
| isshow | 该菜品是否显示在菜品列表中 |

#### 4.2.3.5表category

|  |  |
| --- | --- |
| 属性 | 说明 |
| category\_id | 主码，食品类别的id |
| category\_name | 食品类别的名称 |

#### 4.2.3.6表food\_Belong

|  |  |
| --- | --- |
| 属性 | 说明 |
| food\_id | 主码，食品的id |
| category\_id | 主码，食品类别的id |

#### 4.2.3.7表my\_order

|  |  |
| --- | --- |
| 属性 | 说明 |
| order\_id | 主码，订单的id |
| c\_id | 下单的客户id |
| r\_id | 收货人的id |
| order\_date | 下单日期 |
| order\_state | 订单的状态，限定为“未付款”、“已付款”、“已接单”、“正在配送”、“已完成”、“已取消” |
| ship\_method | 订单的配送方式，限定为“到店自取”和“商家配送” |

#### 4.2.3.8表lineItem

|  |  |
| --- | --- |
| 属性 | 说明 |
| order\_id | 主码，订单的id |
| food\_id | 主码，食品的id |
| amount | 该种食品的数量 |
| total\_price | 该种食品购买时的总价格 |

#### 4.2.3.9表shoppingcart

|  |  |
| --- | --- |
| 属性 | 说明 |
| c\_id | 主码，客户的id |
| food\_id | 主码，食品的id |
| amount | 该种食品的数量 |

#### 4.2.3.10表comments

|  |  |
| --- | --- |
| 属性 | 说明 |
| order\_id | 主码，订单的id |
| food\_id | 主码，食品的id |
| comment | 顾客对该订单的该食品进行的评论 |
| comment\_date | 评论的日期 |

#### 4.2.3.11表collections

|  |  |
| --- | --- |
| 属性 | 说明 |
| c\_id | 主码，客户的id |
| food\_id | 主码，食品的id |

## 4.3数据库物理设计

### 4.3.1 建表SQL

DROP TABLE IF EXISTS collections;

DROP TABLE IF EXISTS comments;

DROP TABLE IF EXISTS shipment;

DROP TABLE IF EXISTS lineitem;

DROP TABLE IF EXISTS my\_order;

DROP TABLE IF EXISTS food\_belong;

DROP TABLE IF EXISTS category;

DROP TABLE IF EXISTS shoppingcart;

DROP TABLE IF EXISTS food;

DROP TABLE IF EXISTS receiver;

DROP TABLE IF EXISTS customer;

DROP TABLE IF EXISTS admin\_account;

CREATE TABLE admin\_account (

admin\_id varchar(100) PRIMARY KEY,

ad\_tel varchar(20),

password varchar(100),

issuper TINYINT

);

INSERT INTO admin\_account(admin\_id, ad\_tel, password, issuper) VALUES('admin','18000000000','123456', 1);

CREATE TABLE customer (

c\_id varchar(100) PRIMARY KEY,

c\_tel varchar(20),

c\_password varchar(100),

email varchar(30)

);

INSERT INTO customer VALUES('abc','18100000000','123456','aaa@163.com');

INSERT INTO customer VALUES('abcd','18000000000','123456','bbb@163.com');

CREATE TABLE receiver (

r\_id int PRIMARY KEY AUTO\_INCREMENT,

c\_id varchar(30),

r\_name varchar(20),

address varchar(100),

r\_tel varchar(20),

isshow TINYINT,

FOREIGN KEY (c\_id) REFERENCES customer (c\_id)

);

INSERT INTO receiver(c\_id,r\_name,address,r\_tel,isshow) VALUES('abc','张三','西安市碑林区友谊西路127号','18100000000',1);

INSERT INTO receiver(c\_id,r\_name,address,r\_tel,isshow) VALUES('abc','李四','西安市碑林区友谊西路127号','18200000000',1);

INSERT INTO receiver(c\_id,r\_name,address,r\_tel,isshow) VALUES('abc','王五','西安市碑林区友谊西路127号','18300000000',1);

CREATE TABLE food (

food\_id int PRIMARY KEY AUTO\_INCREMENT,

fd\_name varchar(40),

fd\_subname varchar(40),

fd\_image varchar(40),

fd\_price numeric(10, 2),

fd\_description varchar(10000),

isshow TINYINT

);

INSERT INTO food(fd\_name,fd\_subname,fd\_image,fd\_price,fd\_description,isshow) VALUES('鲜柠香煎银鳕鱼','头盘开胃菜','1.jpg','120','鳕鱼低脂肪、高蛋白，刺少，是老少皆宜的营养食品。鳕鱼具有高营养、低胆固醇、易于被人体吸收等优点。 鳕鱼鱼脂中含有球蛋白、白蛋白及磷的核蛋白，还含有儿童发育所必需的各种氨基酸，其比值和儿童的需要量非常相近，又容易被人消化吸收，还含有不饱和脂肪酸和钙、磷、铁、B族维生素等。',1);

INSERT INTO food(fd\_name,fd\_subname,fd\_image,fd\_price,fd\_description,isshow) VALUES('四季水果摩卡慕斯','副菜佳肴','2.jpg','198','龙虾一般分两种，一种是大西洋龙虾，以加拿大，美国波士顿最出名，特点是有一对很大的钳子；另一种是太平洋龙虾，从日本一直到我国南海到澳洲都有，它的钳子显然较细。那么出产澳大利亚的龙虾比较肥大有名，就经常叫这种龙虾为澳洲龙虾，应该讲澳大利亚不是唯一的原产地。龙虾一般分两种，一种是大西洋龙虾，以加拿大，美国波士顿最出名，特点是有一对很大的钳子；另一种是太平洋龙虾，从日本一直到我国南海到澳洲都有，它的钳子显然较细。那么出产澳大利亚的龙虾比较肥大有名，就经常叫这种龙虾为澳洲龙虾，应该讲澳大利亚不是唯一的原产地。',1);

INSERT INTO food(fd\_name,fd\_subname,fd\_image,fd\_price,fd\_description,isshow) VALUES('花椰菜生蔬菜沙拉','温婉靓汤','3.jpg','198','龙虾一般分两种，一种是大西洋龙虾，以加拿大，美国波士顿最出名，特点是有一对很大的钳子；另一种是太平洋龙虾，从日本一直到我国南海到澳洲都有，它的钳子显然较细。那么出产澳大利亚的龙虾比较肥大有名，就经常叫这种龙虾为澳洲龙虾，应该讲澳大利亚不是唯一的原产地。龙虾一般分两种，一种是大西洋龙虾，以加拿大，美国波士顿最出名，特点是有一对很大的钳子；另一种是太平洋龙虾，从日本一直到我国南海到澳洲都有，它的钳子显然较细。那么出产澳大利亚的龙虾比较肥大有名，就经常叫这种龙虾为澳洲龙虾，应该讲澳大利亚不是唯一的原产地。',1);

INSERT INTO food(fd\_name,fd\_subname,fd\_image,fd\_price,fd\_description,isshow) VALUES('南非龙虾配澳洲带子','经典西式大餐','4.jpg','208','龙虾一般分两种，一种是大西洋龙虾，以加拿大，美国波士顿最出名，特点是有一对很大的钳子；另一种是太平洋龙虾，从日本一直到我国南海到澳洲都有，它的钳子显然较细。那么出产澳大利亚的龙虾比较肥大有名，就经常叫这种龙虾为澳洲龙虾，应该讲澳大利亚不是唯一的原产地。龙虾一般分两种，一种是大西洋龙虾，以加拿大，美国波士顿最出名，特点是有一对很大的钳子；另一种是太平洋龙虾，从日本一直到我国南海到澳洲都有，它的钳子显然较细。那么出产澳大利亚的龙虾比较肥大有名，就经常叫这种龙虾为澳洲龙虾，应该讲澳大利亚不是唯一的原产地。',1);

INSERT INTO food(fd\_name,fd\_subname,fd\_image,fd\_price,fd\_description,isshow) VALUES('黑椒牛仔骨配黄油西兰花','充饥主菜','5.jpg','168','龙虾一般分两种，一种是大西洋龙虾，以加拿大，美国波士顿最出名，特点是有一对很大的钳子；另一种是太平洋龙虾，从日本一直到我国南海到澳洲都有，它的钳子显然较细。那么出产澳大利亚的龙虾比较肥大有名，就经常叫这种龙虾为澳洲龙虾，应该讲澳大利亚不是唯一的原产地。龙虾一般分两种，一种是大西洋龙虾，以加拿大，美国波士顿最出名，特点是有一对很大的钳子；另一种是太平洋龙虾，从日本一直到我国南海到澳洲都有，它的钳子显然较细。那么出产澳大利亚的龙虾比较肥大有名，就经常叫这种龙虾为澳洲龙虾，应该讲澳大利亚不是唯一的原产地。',1);

INSERT INTO food(fd\_name,fd\_subname,fd\_image,fd\_price,fd\_description,isshow) VALUES('奶豆腐配美味咖喱','甜点','6.jpg','148','龙虾一般分两种，一种是大西洋龙虾，以加拿大，美国波士顿最出名，特点是有一对很大的钳子；另一种是太平洋龙虾，从日本一直到我国南海到澳洲都有，它的钳子显然较细。那么出产澳大利亚的龙虾比较肥大有名，就经常叫这种龙虾为澳洲龙虾，应该讲澳大利亚不是唯一的原产地。龙虾一般分两种，一种是大西洋龙虾，以加拿大，美国波士顿最出名，特点是有一对很大的钳子；另一种是太平洋龙虾，从日本一直到我国南海到澳洲都有，它的钳子显然较细。那么出产澳大利亚的龙虾比较肥大有名，就经常叫这种龙虾为澳洲龙虾，应该讲澳大利亚不是唯一的原产地。',1);

INSERT INTO food(fd\_name,fd\_subname,fd\_image,fd\_price,fd\_description,isshow) VALUES('新西兰羊排配黑椒蘑菇汁','清爽蔬菜','7.jpg','108','龙虾一般分两种，一种是大西洋龙虾，以加拿大，美国波士顿最出名，特点是有一对很大的钳子；另一种是太平洋龙虾，从日本一直到我国南海到澳洲都有，它的钳子显然较细。那么出产澳大利亚的龙虾比较肥大有名，就经常叫这种龙虾为澳洲龙虾，应该讲澳大利亚不是唯一的原产地。龙虾一般分两种，一种是大西洋龙虾，以加拿大，美国波士顿最出名，特点是有一对很大的钳子；另一种是太平洋龙虾，从日本一直到我国南海到澳洲都有，它的钳子显然较细。那么出产澳大利亚的龙虾比较肥大有名，就经常叫这种龙虾为澳洲龙虾，应该讲澳大利亚不是唯一的原产地。',1);

CREATE TABLE category (

category\_id int PRIMARY KEY AUTO\_INCREMENT,

category\_name varchar(40)

);

INSERT INTO category(category\_name) VALUES('主菜');

INSERT INTO category(category\_name) VALUES('头盘');

INSERT INTO category(category\_name) VALUES('副菜');

INSERT INTO category(category\_name) VALUES('靓汤');

INSERT INTO category(category\_name) VALUES('蔬果');

CREATE TABLE food\_belong (

food\_id int,

category\_id int,

PRIMARY KEY (food\_id, category\_id),

FOREIGN KEY (food\_id) REFERENCES food (food\_id),

FOREIGN KEY (category\_id) REFERENCES category (category\_id)

);

INSERT INTO food\_belong(food\_id,category\_id) VALUES('5','1');

INSERT INTO food\_belong(food\_id,category\_id) VALUES('1','2');

INSERT INTO food\_belong(food\_id,category\_id) VALUES('2','3');

INSERT INTO food\_belong(food\_id,category\_id) VALUES('3','4');

INSERT INTO food\_belong(food\_id,category\_id) VALUES('4','1');

INSERT INTO food\_belong(food\_id,category\_id) VALUES('6','5');

INSERT INTO food\_belong(food\_id,category\_id) VALUES('7','5');

CREATE TABLE my\_order (

order\_id int PRIMARY KEY AUTO\_INCREMENT,

c\_id varchar(30),

r\_id int,

order\_date varchar(30),

order\_state varchar(10) CHECK (order\_state IN (

'未付款',

'已付款',

'正在配送',

'已接单',

'待评价',

'已完成',

'已取消'

)),

ship\_method varchar(10) CHECK (ship\_method IN ('商家配送', '到店自取')),

FOREIGN KEY (c\_id) REFERENCES customer (c\_id),

FOREIGN KEY (r\_id) REFERENCES receiver (r\_id)

);

INSERT INTO my\_order(c\_id,r\_id,order\_date,order\_state,ship\_method) VALUES('abc',1,'2018-11-11 18:00:00','已付款','商家配送');

INSERT INTO my\_order(c\_id,r\_id,order\_date,order\_state,ship\_method) VALUES('abc',2,'2018-11-12 18:00:00','未付款','商家配送');

INSERT INTO my\_order(c\_id,r\_id,order\_date,order\_state,ship\_method) VALUES('abc',1,'2018-11-13 18:00:00','已接单','到店自取');

INSERT INTO my\_order(c\_id,r\_id,order\_date,order\_state,ship\_method) VALUES('abcd',1,'2018-11-14 18:00:00','已付款','商家配送');

INSERT INTO my\_order(c\_id,r\_id,order\_date,order\_state,ship\_method) VALUES('abc',2,'2018-11-15 18:00:00','已取消','商家配送');

CREATE TABLE lineitem (

order\_id int,

food\_id int,

amount int,

total\_price numeric(10,2),

PRIMARY KEY (order\_id, food\_id),

FOREIGN KEY (order\_id) REFERENCES my\_order (order\_id),

FOREIGN KEY (food\_id) REFERENCES food (food\_id)

);

INSERT INTO lineitem(order\_id,food\_id,amount,total\_price) VALUES(1,2,2,300);

INSERT INTO lineitem(order\_id,food\_id,amount,total\_price) VALUES(1,3,3,600);

INSERT INTO lineitem(order\_id,food\_id,amount,total\_price) VALUES(2,1,1,200);

INSERT INTO lineitem(order\_id,food\_id,amount,total\_price) VALUES(3,1,4,500);

INSERT INTO lineitem(order\_id,food\_id,amount,total\_price) VALUES(3,5,5,700);

INSERT INTO lineitem(order\_id,food\_id,amount,total\_price) VALUES(4,4,1,300);

INSERT INTO lineitem(order\_id,food\_id,amount,total\_price) VALUES(5,5,3,400);

CREATE TABLE shoppingcart (

c\_id varchar(30),

food\_id int,

amount int,

PRIMARY KEY (c\_id, food\_id),

FOREIGN KEY (c\_id) REFERENCES customer (c\_id),

FOREIGN KEY (food\_id) REFERENCES food (food\_id)

);

INSERT INTO shoppingcart values('abc',1,2);

INSERT INTO shoppingcart values('abc',2,1);

INSERT INTO shoppingcart values('abc',3,1);

CREATE TABLE comments (

order\_id int,

food\_id int,

comment varchar(10000),

comment\_date varchar(30),

PRIMARY KEY (order\_id, food\_id),

FOREIGN KEY (order\_id) REFERENCES my\_order (order\_id),

FOREIGN KEY (food\_id) REFERENCES food (food\_id)

);

INSERT INTO comments(order\_id, food\_id, comment) values(1,2,'好吃!' );

CREATE TABLE collections (

c\_id varchar(30),

food\_id int,

PRIMARY KEY (c\_id, food\_id),

FOREIGN KEY (c\_id) REFERENCES customer (c\_id),

FOREIGN KEY (food\_id) REFERENCES food (food\_id)

);

INSERT INTO collections values('abc',1);

INSERT INTO collections values('abc',2);

INSERT INTO collections values('abcd',3);

# 5需求的可追踪性

本系统需要追踪系统的稳定性，防止出现特殊的意外。

# 6注释

MySQL：MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。