

清华大学餐厅服务管理系统

设计文档

王世因

2016011246

目录

1 项目的总体说明和需求分析	4
1.1 总体说明	4
1.2 需求分析	4
1.3 场景设计	4
1.3.1 点餐	4
1.3.2 美食发现	5
1.4 展望功能	5
2 数据库概念设计 (E-R 模型)	6
2.1 实体集 (Entity Set)	6
2.1.1 顾客 (Customer)	6
2.1.2 服务员 (Waiter)	7
2.1.3 厨师 (Cook)	7
2.1.4 菜品 (Dish)	8
2.1.5 订单 (Order)	9
2.1.6 评论 (Comment)	9
2.2 联系集 (Relationship)	10
2.3 模型总览	11
3 数据库逻辑设计 (关系模式)	11
3.1 应用的转换规则	11
3.2 关系模式清单	12
3.2.1 顾客表 Customer: 由顾客实体集转化而来	12
3.2.2 服务员表 Waiter: 由服务员实体集转化而来	12
3.2.3 厨师表 Cook: 由厨师实体集转化而来	12
3.2.4 菜品表 Dish: 由菜品实体集转化而来	13
3.2.5 订单表 Order: 由订单实体集转化而来	13
3.2.6 评论表 Comment: 由评论实体集转化而来	13
3.2.7 餐饮制作表 CookFood: 由多对多关系 “制作 CookFood” 转化而来	14
3.2.8 餐饮订单表 Include: 由多对多关系 “包含 Include” 转化而来	14
3.2.9 菜品打分表 Mention: 由多对多关系 “提及 Mention” 转化而来	14
4 关系模式规范化处理	14
4.1 第一范式 (1NF)	14
4.2 第二范式 (2NF)	15
4.3 第三范式 (3NF)	15
4.4 Boyce-Codd 范式 (BCNF)	15

1 项目的总体说明和需求分析

1.1 总体说明

清华大学有十多个食堂，包含宿舍区附近的紫荆园、桃李园、玉树园和芝兰园，以及教学楼边上的听涛园、清芬园和观畴园，家属区内部的南园、寓园、家园和澜园，还有安家小厨和拾年咖啡等私人经营的用餐地点。然而根据我平时的了解，同学们对于食堂的了解存在一定的局限性，很多毕业的同学都没有品尝过所有的食堂，这真是一种遗憾。所以我想设计一款数据库应用，让大家能够记录各个食堂的美食，丰富大家的校园生活。

同时，根据大作业的要求，本项目还应当提供实体餐厅店内的点餐服务。管理员需要进行菜单和餐桌基本信息进行维护，注册顾客可以查询菜单和餐桌信息并进行预订。服务员需要能够对某餐桌实现点餐服务，用餐完毕之后根据实际情况结账。厨房人员能够浏览和统计点餐情况，菜品做好之后需要通知服务员进行上菜。

1.2 需求分析

本程序的主要客户人群是用餐的顾客们，我将根据他们的需求来进行设计。其中这种用户又分为正常用餐顾客、打卡用餐顾客和线上美食探索家三类。用餐的顾客们是我们的程序的主要流量来源，也是程序价值的体现群体，我们应当对他们提供免费服务，并鼓励用户参与其中贡献内容价值。

另一个客户人群是清华大学的餐厅们，根据大作业说明文档，他们需要对菜单进行维护。餐厅的服务员需要被提供不同于顾客的访问接口进行点餐、结账服务。厨房需要浏览和统计点餐情况，并给服务员发送菜品做好的通知。通过我的观察，这部分用户希望这个程序能够有很好的稳定性，对程序提供商的依赖度更强，可以对他们收费。

根据数据库系统的设计角度，本系统有 4 类用户：用餐顾客、服务员、厨师、系统管理员。

顾客可以浏览菜品的信息，在服务员的帮助下下单，之后由厨师根据订单制作菜品。顾客还可以根据自己的消费体验，对菜品进行评价。

服务员可以帮助用户点菜，并针对用户的订单服务。

厨师针对用户的订单服务。

系统管理员主要职责是维护其他用户的注册信息，并不断更新菜品的信息。

1.3 场景设计

本系统的系统边界在以下的两个场景，其他的功能会在下一个小节中做出展望。

1.3.1 点餐

在观畴园三层、桃李园三层，我们可以享受到点餐的服务，然而因为管理混乱原因，偶尔会出现点餐混乱、菜品忘记做的情况。因此我们的项目将试图改进顾客的用餐体验，通过合理设计的数据库机制充分发挥食堂服务员和厨师的能力，带给顾客一个满意的服务。点餐的流程分为一下午几步：

- 【顾客】进入食堂，通过程序接口访问数据库，浏览菜单

- 【服务员】 / 【顾客】 根据顾客的需求进行下单，把点餐信息上传到数据库
- 【厨师】 接收数据库中的点餐信息并浏览待做菜品的数量及需求
- 【厨师】 做好后，更改数据库中的订单状态，并通知服务员
- 【服务员】 接收到订单信息的变动信息，去厨房取餐，端给顾客
- 【顾客】 用餐，之后结账
- 【服务员】 根据系统内提供的收费信息对顾客收费和找零
- 【顾客】 填写对菜品的反馈评价

1.3.2 美食发现

做本项目的初衷是让大家了解我们的校园，因此，即使不实地到访清华大学的餐厅，顾客们也应该有一个围观各个餐厅精美菜品的机会，这也是对外展示清华大学魅力的窗口。

这部分功能是我设计的重点，但是流程并没有点餐那么复杂，而是可以被拆解为两个独立的功能。

第一个功能是允许用户远程浏览各个餐厅的菜品，网红式经营食堂，打出各个食堂的特色，培养潜在用户，丰富校园的餐饮文化。

第二个功能是允许用户上传对于菜品的评价，这有利于顾客之间分享清华大学餐厅菜品的评价，帮助厨师进一步完善菜品，有利于本项目程序积累有价值的内容。

1.4 展望功能

在本项目以外，我们还可以在未来增加以下这些功能，进一步完善程序，增加各个用户的体验。

- 用户友好的界面设计。因为咱们的课程重点在后端的数据库部分，而且是单人大作业时间不是很充足，所以和其他的项目一样，本项目对于前段的表达可能不是很充足。
- 智能排序算法。针对菜品的显示排序可以考虑根据顾客的用户画像进行优化，这有利于增强用户体验，增加平台价值。
- 添加打折、促销、优惠券功能。在计算价格的时候，我们假设一单的价格就是所有菜品价格的和。然而在实际中，餐厅可以推出一些优惠券来吸引用户消费。这就需要增加其他更复杂的实体和关系，在本单人项目中没有机会涉及，可以作为以后改进的一点。
- 随笔分享功能。现有的设计是让用户针对菜品进行评价，用户的体验需要和某一个菜品相关。在积累了足够的用户后，我们可以把此程序打成一个美食为主的泛地点社交平台，让大家可以分享一些校园生活的体验。
- 顾客之间的关注和分享。在本项目中，我们把用户与用户看成独立的点，并没有任何信息上的直接交集。如果增添用户之间的关系的的话，我们可以让一个顾客关注另一个顾客，从而获得关注好友的点评信息，增强平台的用户粘性。

2 数据库概念设计 (E-R 模型)

2.1 实体集 (Entity Set)

客观存在并可相互区别的事物称为实体，现实世界由各种各样的实体组成。实体可以是具体的人、事、物，也可以是抽象的概念或联系。同型实体的集合称为实体集。在数据库的 E-R 模型中，我们需要首先定义出场景中的实体集，进而决定数据库的结构设计。

2.1.1 顾客 (Customer)

表 1: 顾客实体集属性设计

属性名称	属性描述	典型取值	说明
customerNo	顾客编号 (主码)	20131209000000000032	系统自动生成的流水号，长度 20 位
customName	顾客姓名	顾卿	顾客姓名，最多 10 字
birthday	顾客生日	1996-9-11	顾客的出生年月日
phone	顾客手机号	18000000000	顾客的手机电话，长度 11 位
email	顾客电子邮件	wagqqk@mails.tsinghua.edu.cn	顾客的电子邮件地址

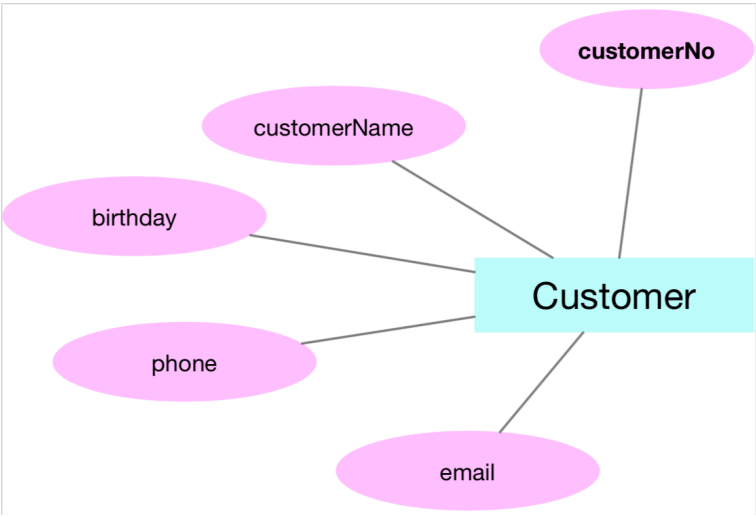


图 1: 顾客实体集

2.1.2 服务员 (Waiter)

表 2: 服务员实体集属性设计

属性名称	属性描述	典型取值	说明
<u>waiterNo</u>	服务员编号 (主码)	beijing023	服务员的工作证号, 长度 10 位
waiterName	服务员姓名	李娟	服务员的中文名字, 上限 10 字
date	服务员入职时间	2010-3-17	服务员的入职时间

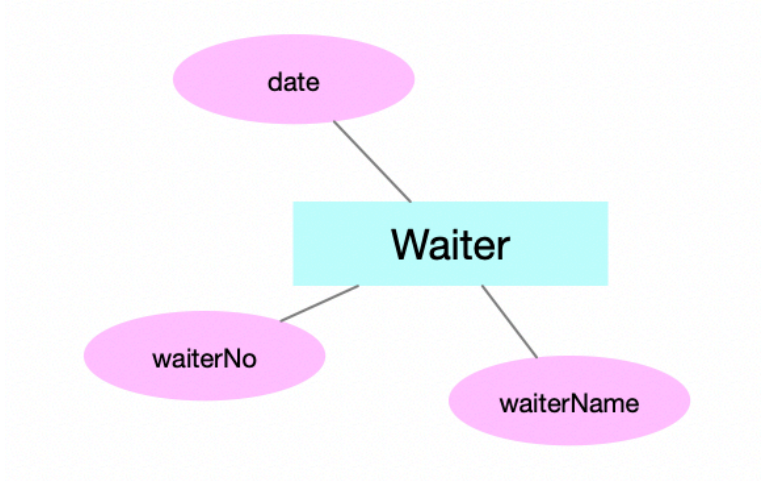


图 2: 服务员实体集

2.1.3 厨师 (Cook)

表 3: 厨师实体集属性设计

属性名称	属性描述	典型取值	说明
<u>cookNo</u>	厨师编号 (主码)	qingf11029	厨师的工作证号, 长度 10 位
cookName	厨师姓名	张大宝	厨师的中文名字, 上限 10 字
date	厨师入职时间	2015-2-19	厨师进入食堂工作的时间

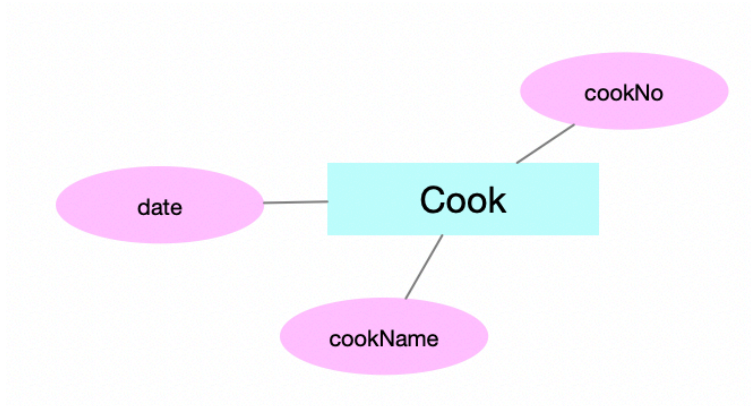


图 3: 厨师实体集

2.1.4 菜品 (Dish)

表 4: c 实体集属性设计

属性名称	属性描述	典型取值	说明
dishNo	菜品编号（主码）	sichuancai0000000876	菜品的唯一的 id 标示，长度 20 位
dishName	菜品姓名	麻婆豆腐	菜品的中文名字，上限 20 字
dishPrice	菜品价格	48	菜品的价格
dishDescription	菜品简介	经典的四川菜……	菜品介绍，上限 1000 字
photo	菜品的图片		每个菜品会配一张图

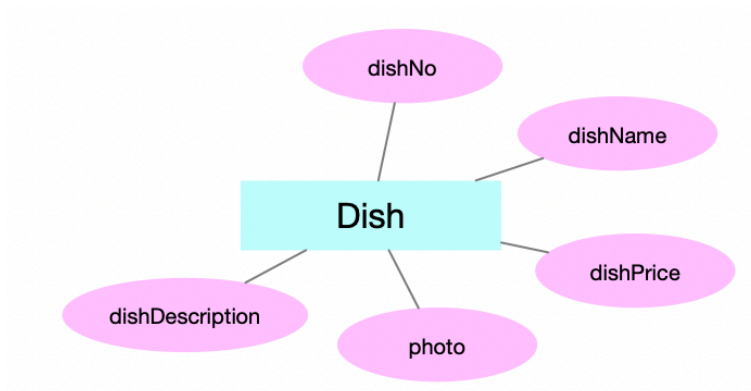


图 4: 菜品实体集

2.1.5 订单 (Order)

表 5: 订单实体集属性设计

属性名称	属性描述	典型取值	说明
<u>orderNo</u>	订单编号 (主码)	20181114115700000001	系统根据时间自动生成的流水号, 长度 20 位
time	下单时间	2018-11-14 11:57:00	用户下单的时间, 至少精确到分钟, 一般精确到秒
totalPrice	订单金额	130	订单的花费, 单位是元
status	订单状态	UN	订单在流程中所处的状态

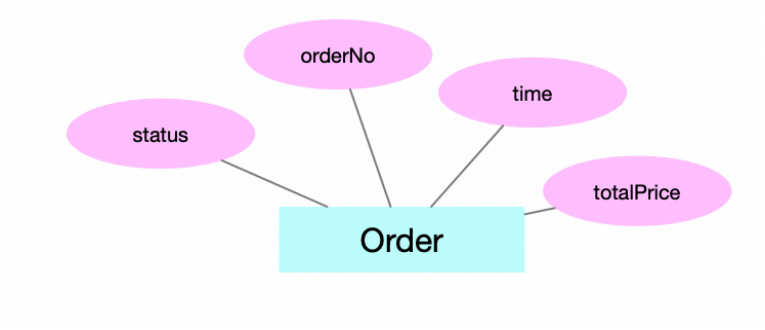


图 5: 订单实体集

订单的状态有 IN (刚初始化, 未完成点菜)、UN (点菜后给服务员审核)、IC (服务员审核后, 厨房开始做)、FI (厨房做完, 等待服务员领取)、DL (服务员给顾客端上桌, 用户进餐)、FI (用户买单, 结束订单)。用户点完后, 只有服务员和厨师有权限更改订单的状态。

2.1.6 评论 (Comment)

表 6: 评论实体集属性设计

属性名称	属性描述	典型取值	说明
<u>commentNo</u>	评论编号 (主码)	20181114235959000003	系统根据时间自动生成的流水号, 长度 20 位
content	评论内容		用户对菜品的评价, 应该是富文本格式

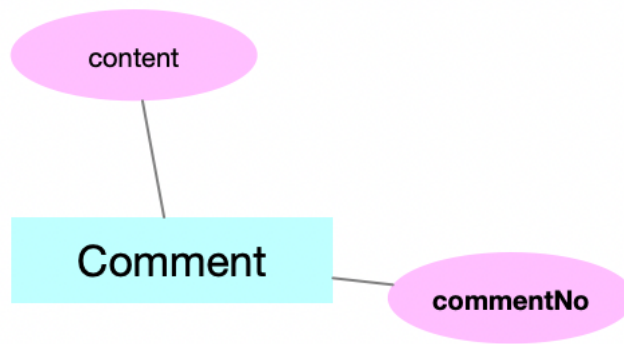


图 6: 评论实体集

2.2 联系集 (Relationship)

顾客和订单的“点菜 (PlaceOrder)”联系集，一对一联系。

顾客和评论的“写 (Write)”联系集，一对多联系。

服务员和订单的“维护 (Maintain)”联系集，一对多联系。

厨师和订单的“制作 (CookFood)”联系集，多对多联系。

订单和菜品的“包含 (Include)”联系集，一对多联系。

菜品和评论的“提及 (Mention)”联系集，多对多联系。

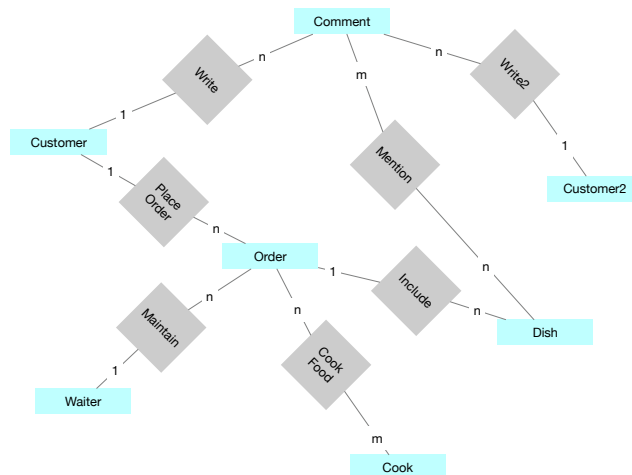


图 7: 餐厅服务管理系统关系图实例

2.3 模型总览

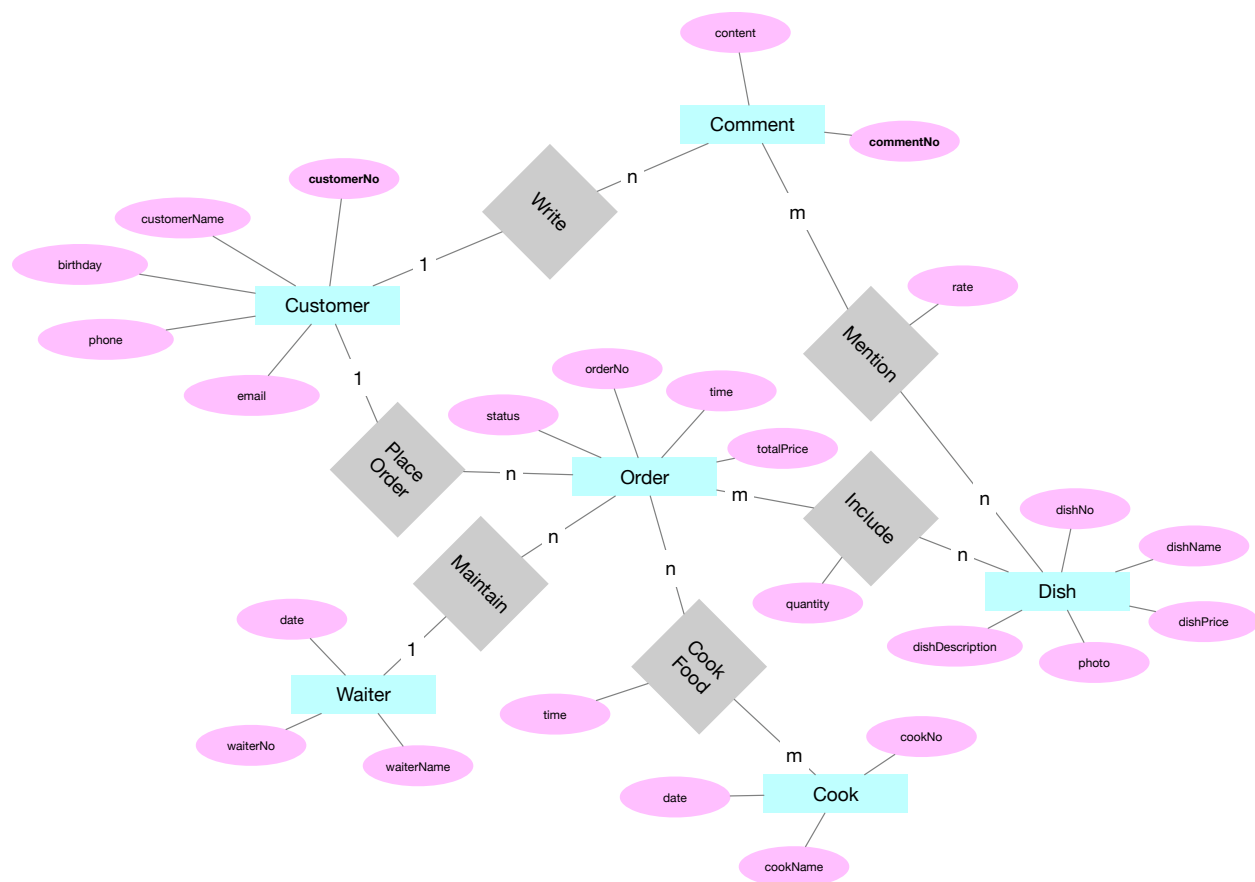


图 8: 餐厅服务管理系统 E-R 图

3 数据库逻辑设计（关系模式）

3.1 应用的转换规则

- 每个实体型转换为一个关系模式，实体的属性就是关系的属性，实体的键就是关系的键。键用下划线标出。
- 若实体之间联系是 1:n (包括 1:1)，在“n”端关系模式中加入“1”端实体的主键，作为其外键，“联系”本身的属性也进入“n”端关系模式。
- 若实体的联系是 m:n，则将联系转换为一个关系模式。“联系”两端实体的主键以及“联系”本身的属性转换为该关系的属性。该关系的主键为各实体主键的组合。

3.2 关系模式清单

3.2.1 顾客表 Customer：由顾客实体集转化而来

表 7: 顾客表 Customer

属性名称	数据类型	属性描述
<u>customerNo</u>	char(20)	顾客编号（主码）
customName	varchar2(10)	顾客姓名
birthday	date	顾客生日
phone	number(11, 0)	顾客手机号
email	varchar2(50)	顾客电子邮件

3.2.2 服务员表 Waiter：由服务员实体集转化而来

表 8: 服务员表 Waiter

属性名称	数据类型	属性描述
<u>waiterNo</u>	char(10)	服务员编号（主码）
waiterName	varchar2(10)	服务员姓名
date	date	服务员入职时间

3.2.3 厨师表 Cook：由厨师实体集转化而来

表 9: 厨师表 Cook

属性名称	数据类型	属性描述
<u>cookNo</u>	char(10)	厨师编号（主码）
cookName	varchar2(10)	厨师姓名
date	date	厨师入职时间

3.2.4 菜品表 Dish：由菜品实体集转化而来

表 10: 菜品表 Dish

属性名称	数据类型	属性描述
<u>dishNo</u>	char(20)	菜品编号（主码）
dishName	varchar2(20)	菜品名字
dishPrice	number(5)	菜品价格
dishDescription	varchar2(1000)	菜品简介
photo	blob	菜品的图片

3.2.5 订单表 Order：由订单实体集转化而来

表 11: 订单表 Order

属性名称	数据类型	属性描述
<u>orderNo</u>	char(20)	订单编号（主码）
customerNo	char(20)	客户编号（外码）
waiterNo	char(10)	服务员编号（外码）
time	timestamp	下单时间
totalPrice	number(5)	订单金额
status	char(2)	订单状态

3.2.6 评论表 Comment：由评论实体集转化而来

表 12: 评论表 Comment

属性名称	数据类型	属性描述
<u>commentNo</u>	char(20)	评论编号（主码）
customerNo	char(20)	客户编号（外码）
content	clob	评论内容

3.2.7 餐饮制作表 CookFood：由多对多关系“制作 CookFood”转化而来

表 13: 餐饮制作表 CookFood

属性名称	数据类型	属性描述
<u>dishNo</u>	char(10)	菜品编号（主码）
<u>cookNo</u>	char(10)	厨师编号（主码）
time	timestamp	菜品制作时间

3.2.8 餐饮订单表 Include：由多对多关系“包含 Include”转化而来

表 14: 餐饮订单表 Include

属性名称	数据类型	属性描述
典型取值		
<u>dishNo</u>	char(10)	菜品编号（主码）
<u>orderNo</u>	char(20)	订单编号（主码）
quantity	number(3,1)	定制分量

3.2.9 菜品打分表 Mention：由多对多关系“提及 Mention”转化而来

表 15: 菜品打分表 Mention

属性名称	数据类型	属性描述
<u>dishNo</u>	char(10)	菜品编号（主码）
<u>commentNo</u>	char(20)	评论编号（主码）
rate	number(1,0)	菜品打分

4 关系模式规范化处理

4.1 第一范式（1NF）

若关系模式 R 的每个属性都是不可再分的数据项，则称 R 属于第一范式。
显然这个系统中，每个属性都不可再分，所以满足第一范式。

4.2 第二范式 (2NF)

在满足第一范式的实体中，如果有复合候选码（多个属性共同构成候选码），那么所有非主属性必须依赖于全部的候选码，而不能只是依赖于部分的候选码属性。

在这个系统中，所有的非主属性都完全函数依赖于主属性，所以满足第二范式。

4.3 第三范式 (3NF)

在满足第二范式的实体中，非主属性不能依赖于另一个非主属性，即所有的非主属性应该直接依赖于全部的主属性（即必须完全依赖，这是 2NF 的要求），并且彼此之间无相互依赖关系（即不能存在传递依赖，这是 3NF 的要求）。

在这个系统中，所有的非主属性不依赖于另一个非主属性，而是直接依赖于主属性，所以满足第三范式。

4.4 Boyce-Codd 范式 (BCNF)

BCNF 要求：所有非主属性都完全函数依赖于每个候选码；所有主属性都完全函数依赖于每个不包含它的候选码；没有任何属性完全函数依赖于非候选码的任何一组属性。

在这个系统中，任何属性（包括主属性和非主属性）都不存在对候选码的部分依赖和传递依赖，也不存在主属性之间的传递依赖，所以满足 BCNF 范式。