成绩管理系统

项目数据库设计说明书

专 业： 计算机科学与技术

班 级： 2017级8班

姓 名： 张明伟

报告日期： 2018年7月

目录

[第一部分：课程设计内容 2](#_Toc530656880)

[第一章 绪论 2](#_Toc530656881)

[1课题简介 2](#_Toc530656882)

[2设计目的 2](#_Toc530656883)

[3设计内容 3](#_Toc530656884)

[第二章 需求分析 4](#_Toc530656885)

[1功能需求 4](#_Toc530656886)

[2 性能需求 5](#_Toc530656887)

[第三章 概念结构设计 6](#_Toc530656888)

[1局部E-R图 6](#_Toc530656889)

[2全局E-R图 7](#_Toc530656890)

[第四章 逻辑结构设计 8](#_Toc530656891)

[1关系模式 8](#_Toc530656892)

[2数据库结构的详细设计 8](#_Toc530656893)

[第五章 物理结构设计 12](#_Toc530656894)

[1建立索引 12](#_Toc530656895)

[2存储结构 12](#_Toc530656896)

[第六章 系统实现 12](#_Toc530656897)

[1 数据库的建立 12](#_Toc530656898)

[2 系统实现 16](#_Toc530656899)

[第三部分 分析与总结 17](#_Toc530656900)

[1数据库分析 17](#_Toc530656901)

[2课题总结 17](#_Toc530656902)

[3、项目展望 18](#_Toc530656903)

[4、心得与体会 18](#_Toc530656904)

[参考文献 19](#_Toc530656905)

# 第一部分：课程设计内容

## 第一章 绪论

### 1课题简介

随着现代科技的高速发展，设备和管理的现代化，在实际工作中如何提高工作效率成为一个很重要的问题。而建立管理信息系统是一个很好的解决办法。

一直以来人们使用传统人工的方式管理学生成绩，这种管理方式存在着许多缺点，譬如：效率低、保密性差，另外时间一长，将产生大量的文件和数据，这对于查找、更新和维护都带来了不少的困难。随着科学技术的不断提高，计算机科学日渐成熟，其强大的功能已为人们深刻认识，它已进入人类社会的各个领域并发挥着越来越重要的作用。学生成绩管理系统的出现轻而易举的解决了传统成绩管理的方法的缺点。伴随着人事管理信息系统的广泛应用，它已经成为了任何一所学校都不可缺少的组成部分，它的存在使得学生和老师们更容易了解自己和班级的成绩，易于进行查询调整。所以学生成绩管理系统能够为学生老师们提供充足的信息和快捷的查询手段。

本文按照数据库系统设计的基本步骤，采取了事先进行需求分析，然后进行数据库的概念设计和逻辑结构设计，最后进行数据库详细设计的方法，完成了一个学生成绩管理系统数据库系统的设计，可以实现：对课程的管理、学生信息管理、专业设置管理、选课情况管理。

### 2设计目的

应用对数据库系统原理的理论学习，通过上机实践的方式将理论知识与实践更好的结合起来，巩固所学知识。

数据库应用课程实践：实践和巩固在课堂教学中学习有关知识，熟练掌握对于给定结构的数据库的创建、基本操作、程序系统的建立和调试以及系统评价。

数据库原理软件设计实践：实践和巩固在课堂教学中学习的关于关系数据库原理的有关知识和数据库系统的建立方法，熟练掌握对于给定实际问题，为了建立一个关系数据库信息管理系统，必须得经过系统调研、需求分析、概念设计、逻辑设计、物理设计、系统调试、维护以及系统评价的一般过程。

### 3设计内容

运用基于E-R 模型的数据库设计方法和关系规范化理论做指导完成从系统的分析到设计直至系统的最终实现，开发学生成绩管理系统，完成学生成绩管理系统的全部功能， 对课程的管理、学生信息管理、专业设置管理、选课情况管理。

首先做好需求分析，并完成数据流图和数据字典。

其次做概念结构设计，利用实体联系的方法将用户需求抽象为信息结构，得到E-R 图。

再次就是逻辑结构设计，将E-R 图转换为计算机系统所支持的逻辑模型。

然后就是物理结构设计，将逻辑数据模型转换为计算机系统所支持的数据库物理结构。

最后就是数据库的实施与运行，把数据载入及运行数据库。

## 第二章 需求分析

### 1功能需求

通过对现行学生成绩管理系统的调查，明确了人事管理系统由课程的管理、学生信息管理、专业设置管理、选课情况管理组成。系统功能的主要描述如下：

（1）完成专业设置情况信息的输入、查询、修改、删除。具体的处理要求是：

输入专业设置基本情况。以直观的表格界面给出，操作者只负责输入数据，输入时必须禁止输入数据表中已有的专业编号。

修改专业设置基本信息。以和输入界面相类似的形式，给出要修改的专业设置基本情况，因此修改之前必须先给出修改的条件，查询到有该专业设置后才能修改。当修改专业编号时，如果该专业已经有学生，即在学生情况表中有该专业编号的学生时，则必须同时修改学生情况表中的记录。

删除专业设置基本情况。以和修改类似的方法先查询出要删除的专业的基本情况。如果学生情况表中有该专业的学生记录，则拒绝删除。

查询统计。根据给定的各种具体条件（专业编号、院名、系别、专业），可以以单个条件进行查询，也可以以若干单个条件组合成任意复杂的条件进行查询。在查询时必须同时给出相应的有关学生人数的统计信息。在查询统计时不允许修改任何信息。

设计一个菜单系统，能够将上述的所有需求联接起来构成一个完整的应用系统。

（2）完成学生基本情况信息的输入、查询、修改、删除。具体的处理要求是：

输入学生基本情况。以直观的表格界面给出，操作者只负责输入数据，输入时必须禁止输入数据表中已有的学号。

修改学生基本信息。以和输入界面相类似的形式，给出要修改的学生基本情况，因此修改之前必须先给出修改的条件，查询到有学生后才能修改。当修改学生的学号时，如果该学生已经选修了课程，即在学生选课表中有该学生的记录，则必须同时修改学生课表中的记录。

删除学生基本情况，以和修改类似的方法先查询出要删除的学生基本情况。查询到有该同学后才能删除和查询一样，若选课表中有该同学的选课记录，也必须同时删除。

查询统计。根据给定的各种具体条件（学号、姓名、性别、系别、专业、入学年份），可以以单个条件进行查询，也可以以若干单个条件组合成任意复杂的条件进行查询。当满足条件的同学有许多时，必须同时给出相应的统计信息，也应该能够逐个地显示每个同学的基本情况。在查询统计时不允许修改学生的任何信息。

设计一个菜单系统，能够将上述的所有需求联接起来构成一个完整的应用系统。

（3）完成课程管理的各种操作，具体的处理要求是：

输入课程的基本信息。要求是以直观的界面表格给出课程输入表格，操作者只负责输入，输入时必须禁止输入想同课程号的记录。

修改课程信息。以和输入界面相类似的形式给出要修改的课程信息，然后才能修改。因此，在修改之前必须先给出修改条件，查询到有相应的课程信息后才能修改。在修改时，如果学生选课表中有选修该门课程的学生记录，则当修改课程号时必须同时修改所有的学生选课表中课程号。

课程成绩的查询统计。应具有按课程号、课程名统计出选修改课程的人数、专业署、系别数以及平均成绩，最高成绩、最低成绩、及格通过率。

设计一个菜单系统，能够将上述各部分功能联起来构成一个完整的应用系统。

（4）完成学生选课的各种管理操作，具体的处理要求是：

学生选课记录的输入。以直观的界面显示学生选课表格，然后输入学生选课记录。学号必须是学生情况表中已有的学号，如果没有，禁止输入。同样，课程号必须时课程表中已有的课程号，如果没有，禁止输入。

学生选课记录的修改。以和输入界面相同的形式给出学生的选课记录。因此，修改之前必须给出修改条件，查询到有学生选课记录后才能修改。在此修改只能修改成绩和考试日期。

学生选课的查询与统计。根据给定的各种具体条件（学号、姓名、课程号、课程名、考试日期、系别、专业，重修），可以以单个条件进行查询统计，也可以以若干单个条件组合成任意复杂的条件进行查询统计。具体的统计内容是：最高分、最低分、平均成绩，各个分数段的门次/人次，不及格率。

设计一个菜单系统，能够将上述各功能联起来构成一个完整的应用系统。

### 2 性能需求

（1）并发访问数量在1000以内；

（2）各类操作系统响应时间控制在5s以内；

（4）要求具有良好的安全性和完整性控制。

## 第三章 概念结构设计

根据系统需求分析，可以得出成绩管理系统数据库的概念模型（信息模型），以下图是用E-R图表示的成绩管理系统的概念模型。最后，对设计出的数据模型进行规范化处理，使数据模型满足第三范式。

### 1局部E-R图

将成绩管理系统定名为“学生成绩管理系统”。 各实体的分E-R图，如图A、B、C、D所示：



**图A 专业设置表**



**图B 学生情况表**



**图C 课程表**



**图D 选课表**

### 2全局E-R图

全局E-R图，如图E所示：



**图E 各实体间的联系**

## 第四章 逻辑结构设计

### 1关系模式

**1.将E-R图转化为数据模型**

专业：（专业编号，院名，系别，专业名）

学生：（学号，姓名，性别，出生日期，入学日期，专业编号）

课程：（课程编号，课程名，学时数，学分）

选课：（学号，课程号，选修日期，考试日期，成绩，是否重修）

**2.优化数据模型**

专业设置表里，非主属性全部完全依赖于专业编号，属于第二范式。无传递依赖关系，属于第三范式。

学生情况表里，非主属性全部完全依赖于学号，属于第二范式。无传递依赖关系，属于第三范式。

课程表里，非主属性全部完全依赖于课程编号，属于第二范式。但存在传递关系，不属于第三范式。可以拆成（课程编号，课程名）（课程名，学时数，学分）。

学生选课表里，码是（学号，课程号，选修日期），非主属性都完全依赖于码，属于第二范式。但存在码推出成绩，成绩推出是否重修的传递关系。所以不属于第三范式。可以拆成（学号，课程号，选修日期，考试日期，成绩）（成绩，是否重修）

**3.设计用户子模式**

创建视图S1，S2

S1给老师看，存在以下属性

（专业编号，院名，系别，专业名，学号，姓名，性别，出生日期，入学日期，专业编号）

S2给学生看，存在以下属性

（学号，姓名，课程名，考试日期，成绩）

### 2数据库结构的详细设计

经过检查，此数据库的详细逻辑设计满足第3范式的要求。学生成绩管理系统中各表的属性设计情况如下：

SDEPT TABLE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 | 取值范围 |
| 专业编号 | 唯一标示一个专业的号码 | 数字类型 | 4 | 0000-9999 |
| 院名 | 学院的名称 | 字符类型 | 20 |  |
| 系别 | 系名 | 字符类型 | 20 |  |
| 专业 | 专业名字 | 字符类型 | 20 |  |

STUDENT TABLE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 | 取值范围 |
| 学号 | 唯一标示学生的号码 | 数字类型 | 8 | 00000000-99999999 |
| 姓名 | 学生名字 | 字符类型 | 20 |  |
| 性别 | 学生性别 | 字符类型 | 20 |  |
| 出生日期 | 出生日期 | 字符类型 | 20 |  |
| 入学日期 | 入学的日子 | 字符类型 | 20 |  |
| 专业编号 | 专业的编号 | 数字类型 | 4 | 0000-9999 |

COURSE TABLE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 | 取值范围 |
| 课程编号 | 唯一标示一门课程的编号 | 数字类型 | 2 | 00-99 |
| 课程名 | 一门课程的名字 | 字符类型 | 20 |  |
| 学时数 | 课程学习的时间 | 数字类型 | 2 | 00-99 |
| 学分 | 课程的学分 | 数字类型 | 2 | 00-99 |

SC TABLE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 | 取值范围 |
| 学号 | 唯一辨识学生的学号 | 数字类型 | 8 | 00000000-99999999 |
| 课程编号 | 唯一辨识课程的 | 数字类型 | 2 | 00-99 |
| 选修日期 | 唯一辨识选课的时间 | 字符类型 | 20 |  |
| 考试日期 | 考试的日期 | 字符类型 | 20 |  |
| 成绩 | 成绩 | 数字类型 | 4 | 0000-9999 |
| 是否重修 | 是否重修课 | 字符类型 | 4 |  |

## 第五章 物理结构设计

数据库物理设计阶段的任务是根据具体计算机系统(DBMS和硬件等)的特点，为给定的数据库系统确定合理的存储结构和存取方法。所谓的“合理”主要有两个含义：一个是要使设计出的物理数据库占用较少的存储空间，另一个对数据库的操作具有尽可能高的速度。主要体现在后者。

### 1建立索引

（1）对表 SDEPT 在属性列“专业编号”上建立非聚集索引，

（2）对表 STUDENT 在属性列“学号”上建立非聚集索引

（3）对表 COURSE 在属性列“课程编号”上建立非聚集索引

（4）对表 SC 在属性列“学号”上建立非聚集索引

在属性列“课程编号”上建立非聚集索引

### 2存储结构

确定数据库的存储结构主要指确定数据的存放位置和存储结构，包括确定关系、索引、日志、备份等的存储安排及存储结构，以及确定系统存储参数的配置。

将日志文件和数据库对象(表、索引等)分别放在不同的磁盘，可以改进系统的性能，提高系统的安全性。所以，系统应将日志文件和数据文件存放在不同磁盘上。

## 第六章 系统实现

### 1 数据库的建立

**用CREATE TABLE 命令定义数据库结构**

创建专业表

CREATE TABLE SDEPT(

SDNO NUMBER(20) PRIMARY KEY,

SDXYNAME VARCHAR(20),

SDXBNAME VARCHAR(20),

SDNAME VARCHAR(20)

)

创建学生表

CREATE TABLE STUDENT(

STNO NUMBER(20) PRIMARY KEY,

STNAME VARCHAR(20),

STSEX VARCHAR(20),

STBIRTHDAY VARCHAR(20),

STRXDAY VARCHAR(20),

SDNO NUMBER(20),

FOREIGN KEY(SDNO) REFERENCES SDEPT(SDNO)

)

创建课程表

CREATE TABLE COURSE(

CNO NUMBER(20) PRIMARY KEY,

CNAME VARCHAR(20),

CTIME NUMBER(2),

CREDIT NUMBER(2)

)

创建选课表

CREATE TABLE SC(

STNO NUMBER(20),

CNO NUMBER(20),

SCTIME VARCHAR(20),

TESTTIME VARCHAR(20),

SCORE NUMBER(4),

RESTUDY VARCHAR(4),

PRIMARY KEY(STNO,CNO,SCTIME),

FOREIGN KEY(STNO) REFERENCES STUDENT(STNO),

FOREIGN KEY(CNO) REFERENCES COURSE(CNO)

**)**

**用INSERT 命令数据装载**

专业

INSERT INTO SDEPT VALUES(001,'园林学院','林业','林业工程');

INSERT INTO SDEPT VALUES(002,'园林学院','林业','林学');

INSERT INTO SDEPT VALUES(003,'信息与计算机工程学院','计算机类','计算机科学与技术');

INSERT INTO SDEPT VALUES(005,'理学院','理学','数学与应用数学');

INSERT INTO SDEPT VALUES(004,'信息与计算机工程学院','计算机类','软件工程');

学生

INSERT INTO STUDENT VALUES(201901,'李一','男','1999-1','2017-9',003);

INSERT INTO STUDENT VALUES(201902,'李二','女','1999-2','2017-9',004);

INSERT INTO STUDENT VALUES(201903,'李三','男','1999-3','2017-9',001);

INSERT INTO STUDENT VALUES(201904,'李四','女','1999-4','2017-9',002);

INSERT INTO STUDENT VALUES(201905,'李五','男','1999-5','2017-9',005);

课程

INSERT INTO COURSE VALUES(201901,'数据库系统概论',48,2);

INSERT INTO COURSE VALUES(201902,'操作系统',52,3);

INSERT INTO COURSE VALUES(201903,'计算机网络',48,2);

INSERT INTO COURSE VALUES(201904,'计算机组成原理',36,2);

INSERT INTO COURSE VALUES(201905,'图像处理',48,2);

选课(在插入数据时，发现课程号和学号重合了，虽然对数据库本身没有什么影响，但是导致辨识度不高，在今后的数据库设计中需要注意)

INSERT INTO SC VALUES(201901,201901,'9-12','12-10',90,'否');

INSERT INTO SC VALUES(201902,201902,'9-12','12-10',96,'是');

INSERT INTO SC VALUES(201903,201903,'9-12','12-10',98,'否');

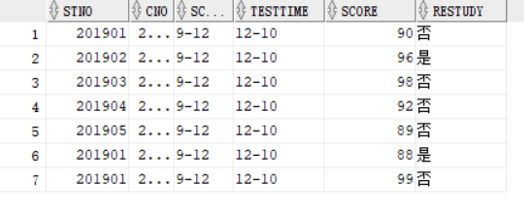
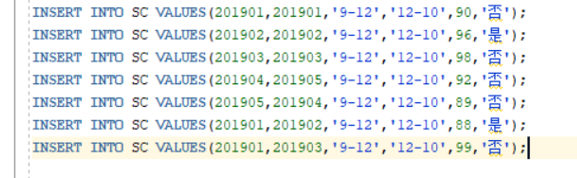
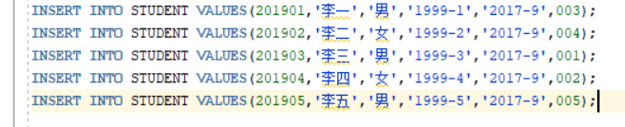
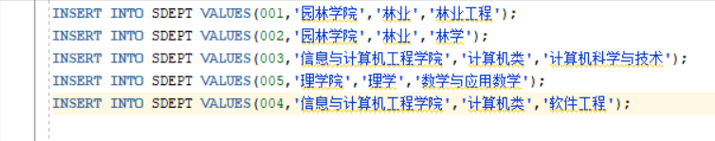
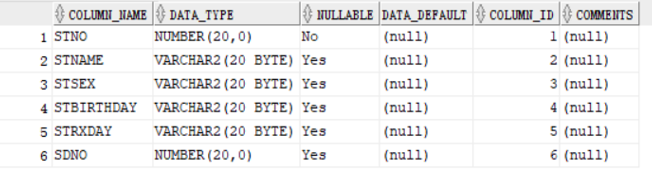
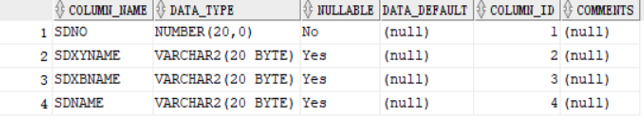
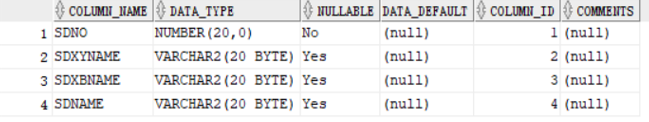
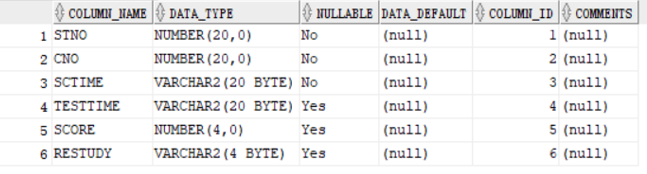
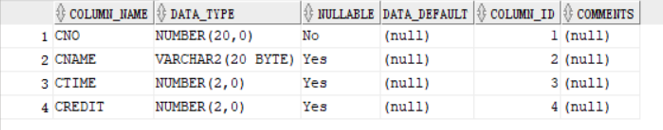
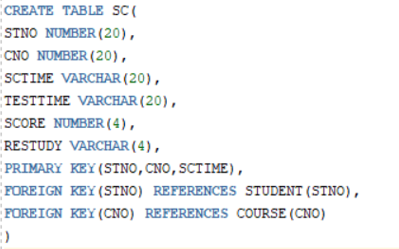
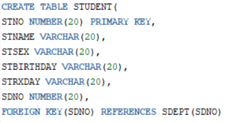
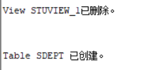
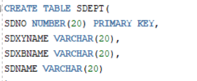
INSERT INTO SC VALUES(201904,201905,'9-12','12-10',92,'否');

INSERT INTO SC VALUES(201905,201904,'9-12','12-10',89,'否');

INSERT INTO SC VALUES(201901,201902,'9-12','12-10',88,'是');

INSERT INTO SC VALUES(201901,201903,'9-12','12-10',99,'否');

### 2 系统实现



系统运行情况及截图说明。

# 第三部分 分析与总结

经过努力，终于设计出一个达到程序设计要求的数据库。其功能完整，用户界面良好，但是也存在着某些缺陷。

## 1数据库分析

在本次校园外卖管理系统的数据库设计与实现过程中，我们小组进行了详细的需求分析、概念结构设计、逻辑结构设计和物理结构设计，最终完成了一个高效的数据库管理系统。通过详细的需求分析，我们确定了系统需要包括商家、菜品、用户、订单、骑手、配送、审批和后台管理人员等八个主要功能模块。

在需求分析阶段，我们明确了每个角色的功能需求和性能需求。例如，商家需要能够管理菜品信息，查看当前的订单信息，并能够与骑手和用户进行有效的沟通。用户需要能够浏览和搜索商家及菜品信息，在线下单，并查看订单的配送状态。骑手需要查看待配送的订单，更新配送状态，并能够与商家和用户联系。后台管理员则需要对系统中的数据进行审核和管理，确保系统的正常运行和数据的准确性。

在概念结构设计阶段，我们使用E-R图对这些模块进行了描述，将其转换为关系模式并进行详细设计。通过E-R图，我们清晰地定义了各个实体之间的关系，例如商家和菜品之间的关系，用户和订单之间的关系，订单和骑手之间的关系等。我们确保了各个实体和关系的完整性，避免了数据冗余和不一致性。

在逻辑结构设计阶段，我们将E-R图转换为关系模式，确定了每个表的属性和主键，定义了外键关系。例如，商家表包括商家ID、商家名称、联系方式、地址和状态等属性，菜品表包括菜品ID、商家ID、菜品名称、描述、价格、可用状态和图片等属性。我们还定义了订单表、骑手表、配送表、审批表和后台管理人员表的结构，并添加了必要的约束条件，以确保数据的完整性和一致性。

在物理结构设计阶段，我们创建了适当的索引来提高查询效率。例如，我们在订单表的用户ID、商家ID和骑手ID字段上创建了索引，以加快查询速度。我们还为每个表添加了必要的约束条件，如主键约束、外键约束和唯一约束，以确保数据的完整性和一致性。此外，我们对数据库进行了性能优化，确保其在高并发访问下仍能保持高效运行。。

## 2课题总结

在项目的完成过程中，我们小组成员分工合作，共同面对和解决了许多挑战。例如，在设计数据库结构时，我们仔细考虑了各个实体之间的关系，以及如何通过外键约束来维护这些关系的完整性。我们进行了多次头脑风暴和讨论，确定了最佳的设计方案。在数据库实现过程中，我们通过多次测试和调试，解决了数据插入、更新和查询中的各种问题。在这个过程中，我们不仅巩固了课堂上学到的数据库理论知识，还提高了实际操作和问题解决的能力。

在设计数据库结构时，我们遇到了一些挑战。例如，如何设计一个高效的查询结构，以满足系统的高并发访问需求。我们通过对数据库索引的合理设计，提高了查询速度和系统的整体性能。我们还遇到了数据一致性的问题，通过添加适当的约束条件，确保了数据的完整性和一致性。在数据插入和更新过程中，我们进行了多次测试和调试，确保系统能够正确处理各种边界情况和异常情况。

在完成项目的过程中，我们学到了许多新的知识。我们深入理解了关系数据库的设计原则和最佳实践，掌握了如何从需求分析到数据库实现的整个流程。我们学会了如何使用SQL进行复杂查询和数据操作，如何通过创建索引来优化数据库性能，以及如何处理数据的一致性和完整性问题。我们还学会了如何进行系统测试和调试，确保系统的稳定性和可靠性。

这一段时间的工作努力，最后的结果固重要，但更重要的是它让我们了解了数据库设计的一般过程。最后是软件的维护。在软件的使用过程中不可避免地会出现一些错误，因此维护便是必不可少的。这便是完成一个项目必经的过程。

## 3、项目展望

在项目拓展方面，我们认为该系统还可以进一步完善。例如，可以添加更详细的用户评价系统，让用户可以对商家和菜品进行评价和打分，从而帮助其他用户做出更好的选择。通过引入用户评价系统，可以提高用户的参与度和系统的整体用户体验。用户可以根据其他用户的评价，选择更加适合自己的商家和菜品，提高订单的成功率和满意度。

此外，我们还可以通过引入数据分析和机器学习技术，对用户的订单数据进行分析，提供个性化推荐服务。通过对用户历史订单数据的分析，可以了解用户的偏好和消费习惯，向用户推荐他们可能感兴趣的商家和菜品。这不仅可以提高用户的满意度，还可以增加商家的销售额，进一步提升系统的整体价值。

我们还可以引入实时位置跟踪和导航功能，帮助骑手更快地找到取餐和送餐的地址，提高配送效率。通过集成GPS和地图服务，骑手可以实时查看最佳的配送路线，避免交通拥堵和绕路，提高配送的及时性和准确性。这对于提高用户的满意度和系统的整体服务质量具有重要意义。

## 4、心得与体会

通过本次项目，我们不仅提高了专业技能，也增强了团队合作和解决问题的能力。我们学会了如何进行需求分析，如何设计高效的数据库结构，如何进行系统测试和调试，如何处理数据的一致性和完整性问题。我们还学会了如何进行团队合作，如何分工和配合，如何进行有效的沟通和协调。这些经验和技能对于我们未来的学习和工作具有重要的借鉴意义。

在项目的完成过程中，我们体会到了数据库设计和实现的复杂性和挑战性。我们深刻认识到，只有通过详细的需求分析和合理的设计，才能确保数据库系统的高效运行和数据的完整性。我们还认识到，系统的实现不仅仅是技术问题，还涉及到团队合作、沟通和协调等方面的问题。只有通过团队的共同努力，才能克服各种困难，成功完成项目。

总的来说，本次校园外卖管理系统的数据库设计与实现项目，让我们受益匪浅。我们不仅掌握了数据库设计和实现的基本方法和技巧，还提高了解决实际问题的能力。我们相信，通过不断的学习和实践，我们能够在未来的学习和工作中，取得更加优异的成绩。

# 参考文献

[1] 孟繁荣、闫秋艳等编著.《数据库原理与应用》.清华大学出版社.2010年

[2] 王珊、萨师煊编著.《数据库系统概论》.第四版.高等教育出版社，2007年

[3] 宋坤、李严、等编著.《SQL Server数据库开发实例解析》.机械工业出版社.2006年

[4] 周汉平编著.《数据库设计及其应用程序开发》.清华大学出版社.2010 年

[5] 徐兰芳.彭冰.吴永英编著.《数据库设计与实现》.上海交通大学出版社.2006年