**南京信息工程大学**

**应用技术学院**

数据库系统

课程设计报告

设计题目： 某高校学生选课管理系统

姓 名 韩开元

学 号 201833100035

姓 名 成凯

学 号 201833050025

专 业 计算机科学与技术

指导教师 孙菁

2022 年 11 月 29 日

目 录

[1绪论 1](#_Toc280214414)

[1.1设计背景 1](#_Toc132018227)

[1.2设计概述 1](#_Toc479733838)

[1.3开发系统技术的选用与介绍 1](#_Toc1233004428)

[1.4本系统解决的问题 2](#_Toc2035711493)

[2 系统需求分析 2](#_Toc493598847)

[2.1 安全性需求 2](#_Toc186493168)

[2.2 系统需求 2](#_Toc1212033603)

[2.3 功能需求 3](#_Toc1766373826)

[2.4 可行性分析 3](#_Toc630957454)

[2.5 系统运行环境 3](#_Toc227680492)

[2.6 数据流图 3](#_Toc1957653737)

[2.7 数据字典 3](#_Toc689402072)

[3 系统总体设计 4](#_Toc1106348539)

[3.1 系统功能结构设计 4](#_Toc1486479247)

[3.2 系统功能流程设计 4](#_Toc1579438778)

[3.3 数据库设计 6](#_Toc582181279)

[4 概念逻辑设计 7](#_Toc785260421)

[4.1 E-R图 8](#_Toc1584884932)

[5系统实现 10](#_Toc1921378383)

[5.1 创建表 10](#_Toc894883142)

[5.2信息录入 12](#_Toc1472987653)

[5.3创建触发器 13](#_Toc312001355)

[5.4 查询某学年没有被选修的课程 14](#_Toc1799191158)

[5.5 查询指定学期学生的学分获取情况 14](#_Toc288559099)

[5.6 查询各个教师任课的学生数量 15](#_Toc794701967)

[6 总结 15](#_Toc1355158676)

[6.1 团队分工 15](#_Toc2087791097)

[7 参考文献 16](#_Toc91079545)

1绪论

1.1设计背景

随着无纸化办公的 普追实现，信息的自动处理以及网络是的信息交互方式己经被人们广泛应用，让计算机来管理学生信息是现在各个高校的基木工作，也慢慢演变成高校教学管理工作必不可少的一部分，网上选课方式相比传统更加的节约资源，增加了学生选课的自主权，也减少了教师不必要的工作量。

学生选课系统作为一种现代化的教学技术，越来越受到人们的重视，是学校很重要的一部分。学生选课系统就是为了管理好选课信息而设计的。学生选课系统使选课变得更加规范化、系统化、程序化，避免了重复，提高信息的正确性、办事效幸的高速化，能够淮确、及时、有效的查询和修改学生选课情况。

随着计算机应用地日益普及，教务管理自动化是提高工作效率地有效途径，随着各高校学分制地实施，实现信息化管理学生选课势在必行。学生选课管理系统是教案管理地重要组成部分，其开发主要包括后台、数据库地建立和维护以及前端应用程序地开发两个方面，前者要求建立起数据地一致性、完整性和安全性好地数据库，而对于后者则要求应用程序功能完备、易使用及界面友好等特点。系统实现后，极大地方便对学生进行选课和选课后临时班级地安排管理，学生选课信息地管理。避免在安排课程信息地滞后，减少信息交流地烦琐过程及其带来地开销。促进高校教育地计算机信息化进程，提高学校地工作效率。对于系统地本身而言，应该具有较高地实用性、安全性。能够极大地满足学生选课，以及学校对选课信息地管理。

本文以高校学生选课管理系统开发过程为背景，进行数据库系统的设计与开发，采用自上而下，逐步细化，逐步完善的原则。基本功能全面，安全性好。

1.2设计概述

本次设计主要目的是实现一个学生选课系统，系统主要功能要求如下：

1.学生管理：包括学生信息管理、选课模块和学期学分查询模块（可以查询本课程的最高成绩、平均成绩、最低成绩和名次）。

2.教师管理：包括可以增加新的课程信息（课程号，课程名称、学分等信息），删除旧的课程信息，修改已有的课程信息；课程管理模块，可以实现期末成绩的录入。

3.统计管理：统计某个学期没有被选修过的课程，以便做后续的学习课程安排；统计指定学期的学生学分；查看各个任课教师所教学的学生人数等等。

1.3开发系统技术的选用与介绍

本系统采用MySQL（InnoDB引擎）数据库实现，并使用Navicat Premium作为DBMS可视化管理工具进行数据的管理操作以及数据模拟。

MySQL是一种关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型和大型网站的开发都选择 MySQL 作为网站数据库。

MySQL是一个小型的开源的关系型数据库管理系统，与其他大型数据库管理系统例如Oracle、DB2、SQL Server等相比，MySQL规模小，功能有限，但是它体积小、速度快、成本低，且它提供的功能对稍微复杂的应用已经够用，这些特性使得MySQL成为世界上最受欢迎的开放源代码数据库。

MySQL数据库的优势：

（1）MySQL是开放源代码的数据库，任何人都可以获得该数据库的源代码。

（2）MySQL能够实现跨平台操作，可以在Windows、UNIX、Linux和Mac OS等操作系统上运行。

（3）MySQL数据库是一款自由软件，大部分应用场景下都是免费使用

MySQL数据库的特点：

（1）MySQL数据库是用C和C++语言编写的，并且使用了多种编辑器进行测试，以保证源码的可移植性；

（2）支持多个操作系统例如：Windows、Linux、Mac OS等等；

（3）支持多线程，可以充分的利用CPU资源；

（4）为多种编程语言提供API，包括C语言，Java，PHP。Python语言等；

（5）MySQL优化了SQL算法，有效的提高了查询速度；

（6）MySQL内提供了用于管理，检查以及优化数据库操作的管理工具；

（7）它能够作为一个单独的应用程序应用在客户端服务器网络环境中，也可以作为一个库嵌入到其他的软件中并提供多种语言支持。

1.4本系统解决的问题

开发一套线上的学生选课系统，可让学生以及教师通过网络直接进行课程的开设以及选课操作，省时省力，提供更好的安全性以及可维护性，免去线下选课的不方便性。

系统实现后，极大的方便对学生进行选课和选课后临时班级的安排管理，学生选课信息的管理。避免在安排课程信息的滞后，减少信息交流的烦琐过程及其带来的开销。促进高校教育的计算机信息化进程，提高学校的工作效率。对于系统的本身而言，应该具有较高的实用性、安全性。能够极大的满足学生选课，以及学校对选课信息的管理。

2 系统需求分析

2.1 安全性需求

系统安全性要求体现在数据库安全性、信息安全性和系统平台的安全性等方面。安全性先通过视图机制，不同的用户只能访问系统授权的视图，这样可提供系统数据一定程度上的安全性，再通过分配权限、设置权限级别来区别对待不同操作者对数据库的操作来提高数据库的安全性：系统平台的安全性体现在操作系统的安全性、计算机系统的安全性和网络体系的安全性等方面。

系统完整性要求系统中数据的正确性以及相容性。可通过建立主、外键，使用 check约束，或者通过使用触发器和级联更新。

2.2 系统需求

用户的需求具体体现在选课信息和用户信息的提供、保存、更新和查询的方面。这就要求数据库的设计必须合理，使之能够充分满足各种信息的输入和输出，保证数据存储的可靠性，并且能够快速取出和存入。而前台显示部分，应具有人性化的界面，方便用户操作。因各个学校的实际情况不同，系统应该具有兼容性。例如：一些学校学生人数较多，同时登陆系统，系统承载的负荷就很大。系统需要同时处理很大的数据量，这时系统不会因此崩溃。此外，系统还应该具有较强的安全性，保证身份不同的用户，不能越权操作。非合法用户不能对数据进行操作。

2.3 功能需求

通过系统功能的分析，结合需求分析员在各大高校实地考查，调查的对象涵盖了，学校的教职工、在校师生。特别是对已经运行了与本系统同类产品的学校的师生使用选课管理系统心得体会进行了分析，总结出如下的需求信息：

（1）学生的需求：能进行选课，查看管理员发布的选课信息，自己的选课情况，本人的基本信息，课程的成绩；

（2）教师的需求：能查看自己的个人信息，及所授课的班级的所有学生的本门课程的成绩信息，并能进行增加和修改；

（3）管理员的需求：对学生选课情况进行管理，包括发布选课信息，对学生的选课情况进行查看。管理员还可以对授课老师的信息进行管理。

2.4 可行性分析

随着互联网的蓬勃发展，越来越多的机构将科研管理和Internet融合到一起，以便各高校管理。在这样的社会背景下，基于Windows和SQL server 2019，运用Java语言开发的学生管理系统。科学技术不断提高，计算机科学日渐程顺，因此技术可行。目标群体为高校的老师、管理员、学生，该系统将三者有机的结合在一起，有效的提高管理水平和效率。系统界面友好，操作简单，系统要求配置不高，应社会需求，有较好的社会前景。

2.5 系统运行环境

本系统测试运行环境为MacOS 13，开发平台为**MySQL**和**Navicat Premium**。

2.6 数据流图

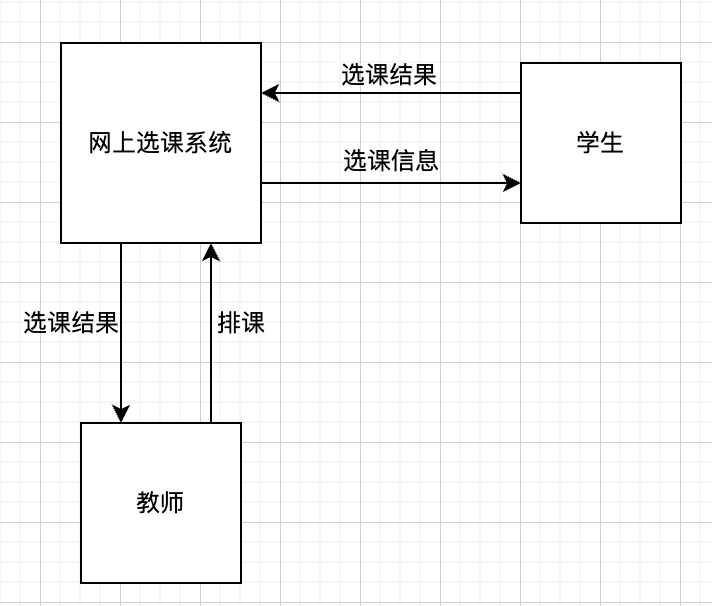
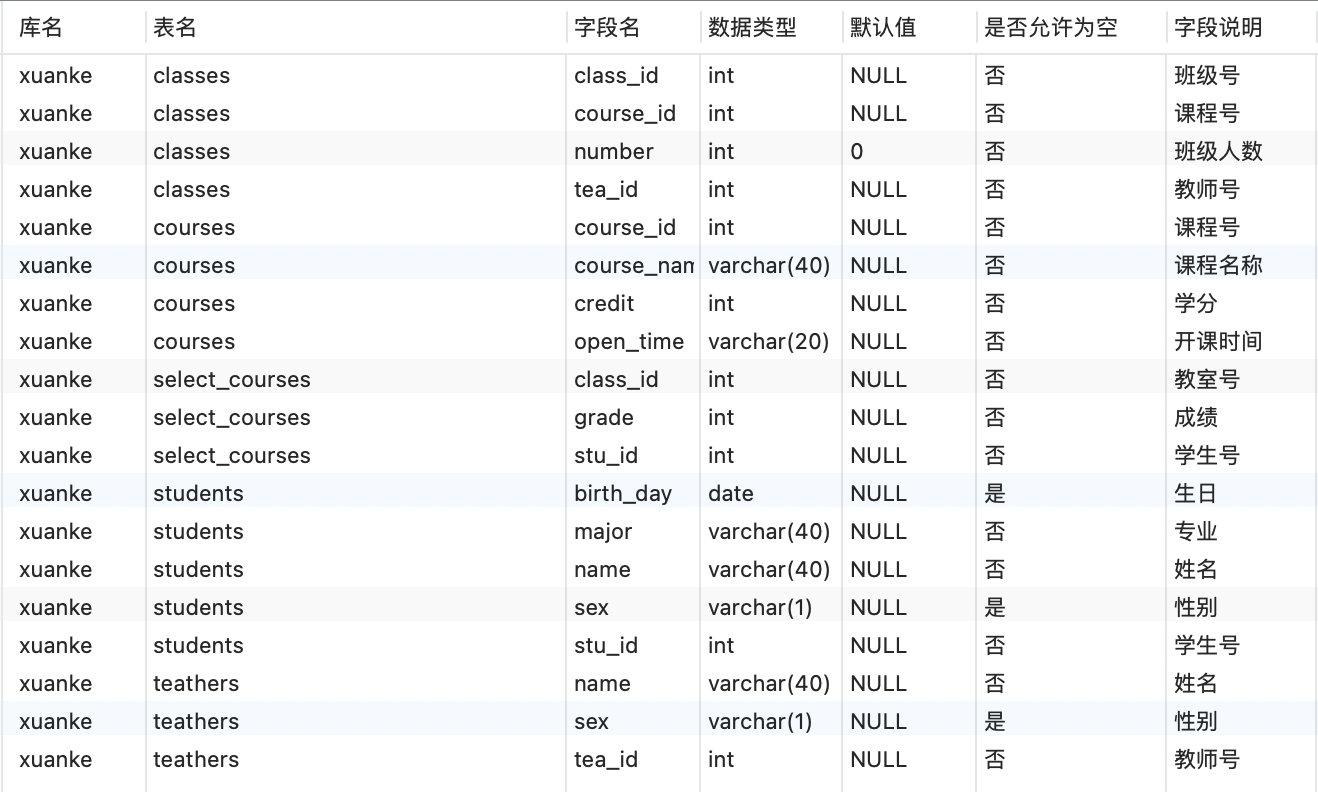


图2-1 数据流图

2.7 数据字典



3 系统总体设计

根据需求分析，总共可分为四个模块，老师模块、管理员模块、学生模块。然后根据选课流程和题目要求，安排每个模块工作内容和工作顺序，指定系统的工作流程。

3.1 系统功能结构设计

通过上述功能需求分析，因为存在多种用户类型，所以需要增加用户登录的功能。老师有对课程进行增删改查的功能，管理员统计课程和选课情况，学生功能为选课和查询信息。系统功能模块划分如图3-1所示。

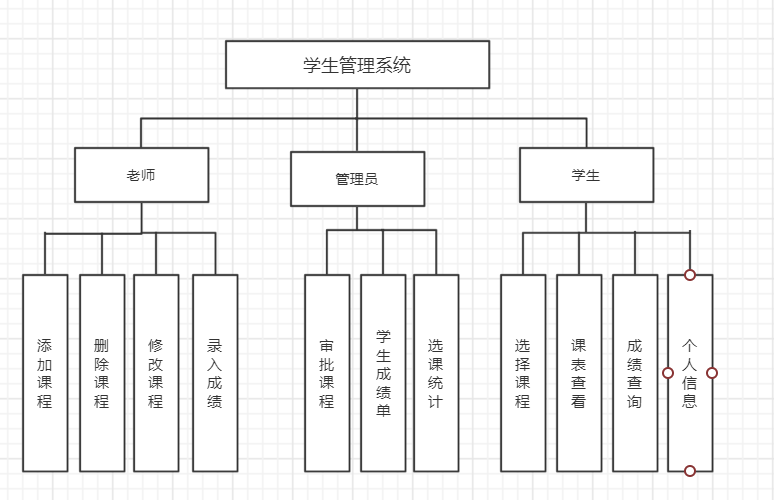


图3-1 系统功能结构图

3.2 系统功能流程设计

3.2.1删除课程功能

删除课程功能流程图如图3-2所示。

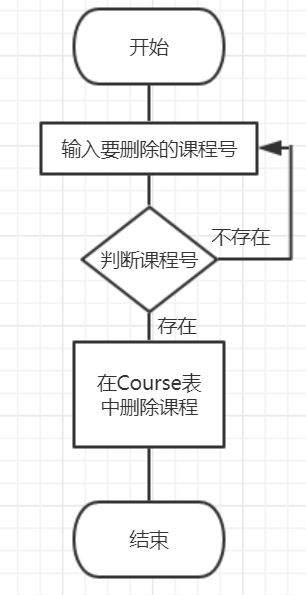


图3-2 删除课程功能流程图

3.2.2修改课程功能

修改课程功能流程图如图3-3所示。

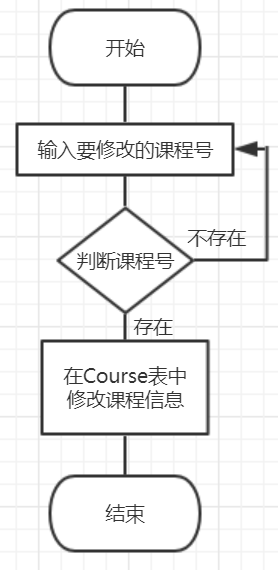


图3-3 修改课程功能流程图

3.2.3系统选课功能

系统选课流程如图3-4所示。

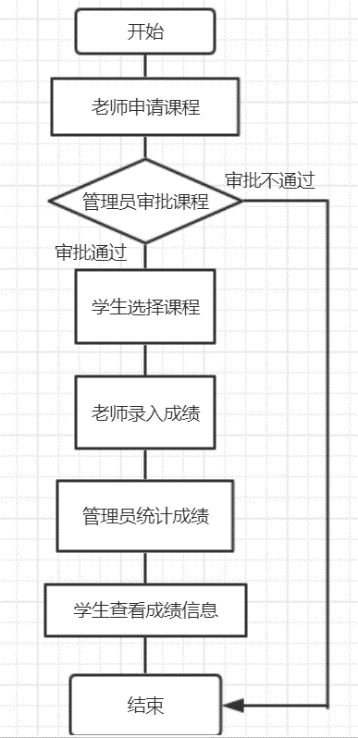


图3-4 系统选课流程图

3.3 数据库设计

设计一个选课数据库，能够实现学生自主选课功能，并能实现学生信息、课程信息、学生选课信息、学生成绩信息等的创建、更新、删除、修改，数据库完整性检查、冲突检查、查询优化以及数据库角色、权限的分配，实现数据库的备份等事务处理以保证数据库的完整性、一致性、安全性、可靠性。

（1）students（学生表）

students（stu\_id，name，birth\_day，sex, major），stu\_id为主键，唯一且非空。其他四个普通属性。

students表如表3-1所示。

表3-1 students表结构说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 非空 | 主键 | 含义 |
| stu\_id | int(10) | YES | YES | 学号 |
| name | varchar(40) | YES | NO | 姓名 |
| birth\_day | date | NO | NO | 生日 |
| sex | varchar(40) | NO | NO | 性别 |
| major | varchar(40) | YES | NO | 专业 |

1. teathers（教师表）

teathers表如表3-2所示。

表3-2 teathers表结构说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 非空 | 主键 | 含义 |
| tea\_id | int(10) | YES | YES | 教师号 |
| name | varchar(40) | YES | NO | 姓名 |
| sex | varchar(10) | NO | NO | 性别 |

1. courses（课程表）

courses表如表3-3所示。

表3-3 courses表结构说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 非空 | 主键 | 含义 |
| course\_id | int(10) | YES | YES | 课程号 |
| course\_name | varchar(40) | YES | NO | 课程名 |
| credit | int | YES | NO | 学分 |
| open\_time | varchar(20) | YES | NO | 开课时间 |

1. classes（临时班级表）

一名学生可以选修多门课程，一门课程可被多个学生选修，学生先修课程的实质是选临时班级。每个临时班级对应一门课程和一个老师，每门课程有一个成绩，且选修课程名不能重复；一名教师可以任教多门课程，一门课程可由多个老师教授。

classes表如表3-4所示。

表3-4 classes表结构说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 非空 | 主键 | 含义 |
| class\_id | int(10) | YES | YES | 班级号 |
| course\_id | int(10) | YES | NO | 课程号 |
| tea\_id | int(10) | YES | NO | 教师号 |
| number | int(10) | YES | NO | 班级人数 |

1. select\_courses（选课表）

select\_courses表如表3-5所示。

表3-5 select\_courses表结构说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 非空 | 主键 | 含义 |
| stu\_id | int(10) | YES | YES | 学生号 |
| class\_id | int(10) | YES | NO | 班级号 |
| grade | int(10) | YES | NO | 成绩 |

4 概念逻辑设计

4.1 E-R图

根据系统的需求分析和关系模式画出如下的E-R图：

1. 学生E-R图

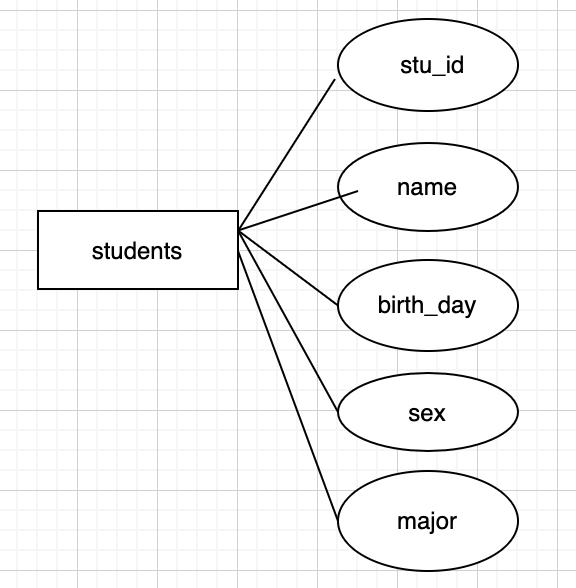


图4-1 学生E-R图

1. 教师E-R图

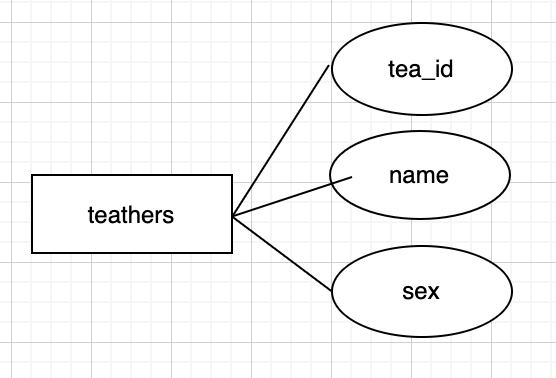


图4-2 教师E-R图

1. 课程E-R图

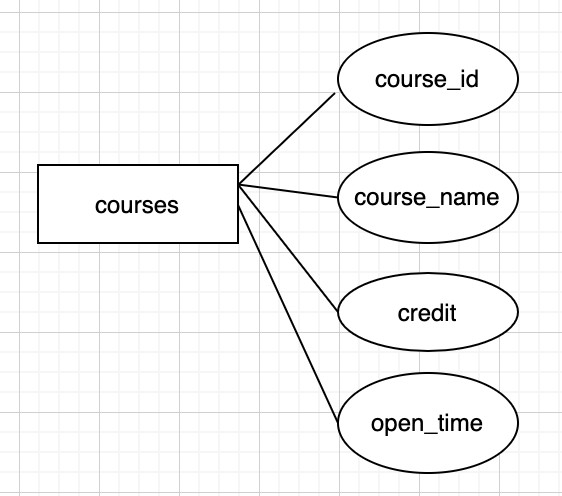


图4-3 课程E-R图

1. 班级E-R图

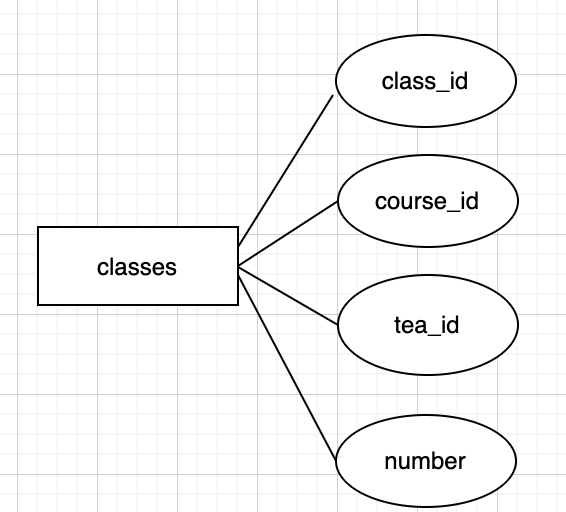


图4-4 班级E-R图

1. 选课E-R图

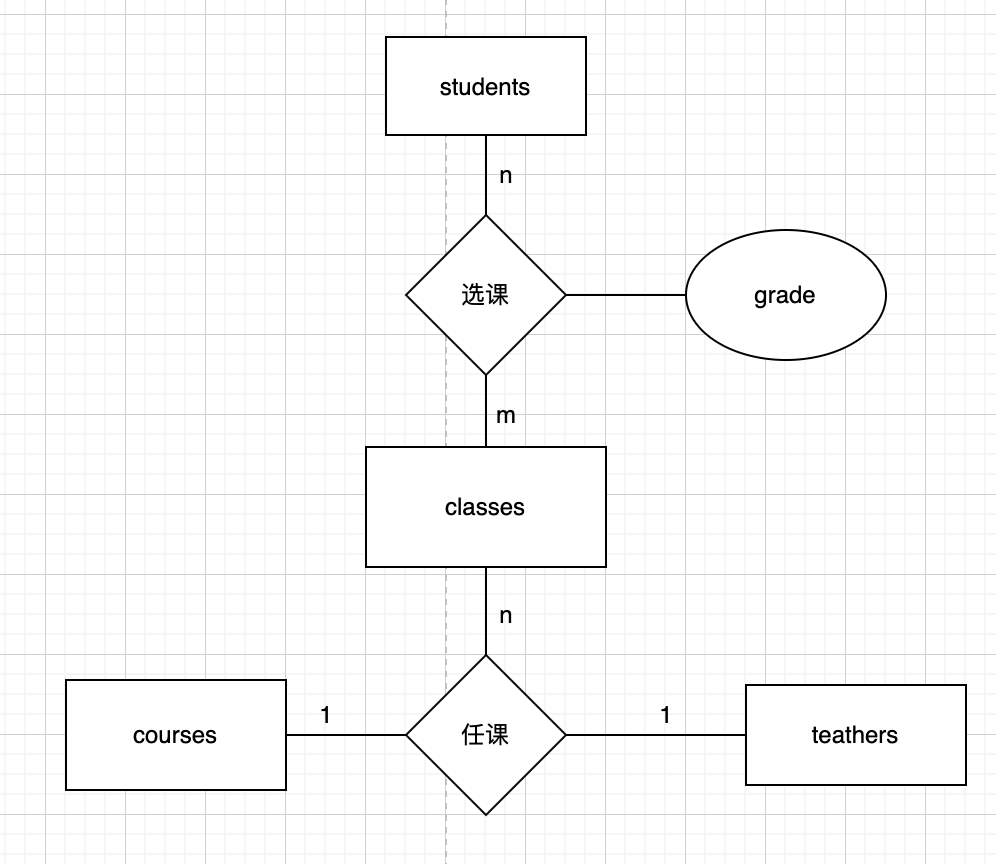


图4-5 选课E-R图

5系统实现

5.1 创建表

使用SQL语句分别创建学生表、教师表、课程表、班级表共四个实体表，以及一个选课关系表，编写的SQL语句如下：

1. 创建学生表：

-- student table

CREATE TABLE students(

stu\_id INT PRIMARY KEY,

name VARCHAR(40),

birth\_day DATE,

sex VARCHAR(1),

major VARCHAR(40)

);

1. 创建教师表：

-- teacher table

CREATE TABLE teathers(

tea\_id INT PRIMARY KEY,

name VARCHAR(40),

sex VARCHAR(1)

);

1. 创建课程表

-- courses table

CREATE TABLE courses(

course\_id INT PRIMARY KEY,

course\_name VARCHAR(40),

credit INT,

open\_time VARCHAR(20)

);

1. 创建班级表

-- temp class table

CREATE TABLE classes(

class\_id INT PRIMARY KEY,

course\_id INT,

tea\_id INT,

number INT DEFAULT 0,

FOREIGN KEY (course\_id) REFERENCES courses(course\_id) ON DELETE CASCADE,

FOREIGN KEY (tea\_id) REFERENCES teathers(tea\_id) ON DELETE CASCADE

);

1. 创建选课关系表

-- 建立多对多关系表

CREATE TABLE select\_courses(

stu\_id INT,

class\_id INT,

grade INT,

PRIMARY KEY (stu\_id, class\_id),

FOREIGN KEY (stu\_id) REFERENCES students(stu\_id) ON DELETE CASCADE,

FOREIGN KEY (class\_id) REFERENCES classes(class\_id) ON DELETE CASCADE

);

5.2信息录入

-- 插入除select\_courses表以外的数据

-- insert data into students

INSERT INTO students VALUES(101, '邓紫棋', '1991-08-16', '女', '音乐');

INSERT INTO students VALUES(102, '小红', '1999-01-03', '男', '软件工程');

INSERT INTO students VALUES(103, '小王', '1995-9-18', '男', '计算机科学与技术');

INSERT INTO students VALUES(104, '小李', '1998-08-30', '女', '计算机科学与技术');

INSERT INTO students VALUES(105, '小刘', '1999-4-21', '男', '艺术');

-- insert data into teachers

INSERT INTO teathers VALUES(201, '冯启明','男');

INSERT INTO teathers VALUES(202, '赵宇舟','男');

INSERT INTO teathers VALUES(203, '仲淼','男');

INSERT INTO teathers VALUES(204, '张欢欣','男');

INSERT INTO teathers VALUES(205, '刘芳雨','女');

-- insert data into courses

-- 2021-2022-2 表示：2021-2022学年的第2学期

INSERT INTO courses VALUES(3001, '数据库设计', 2, '2021-2022-1');

INSERT INTO courses VALUES(3002, 'C语言', 3, '2021-2022-1');

INSERT INTO courses VALUES(3003, 'Java程序设计', 3, '2021-2022-2');

INSERT INTO courses VALUES(3004, '声学', 2, '2021-2022-2');

INSERT INTO courses VALUES(3005, '操作系统', 3, '2020-2021-1');

INSERT INTO courses VALUES(3006, '美术艺术', 2, '2020-2021-1');

-- insert data into classes

-- 作为临时表，管理多个学生和多个老师的选课，这里插入初始数据暂时先不插入班级人数属性，等后面添加触发器后自动更新

INSERT INTO classes(class\_id, course\_id, tea\_id) VALUES(6001, 3001, 201);

INSERT INTO classes(class\_id, course\_id, tea\_id) VALUES(6002, 3001, 204);

INSERT INTO classes(class\_id, course\_id, tea\_id) VALUES(6003, 3002, 201);

INSERT INTO classes(class\_id, course\_id, tea\_id) VALUES(6004, 3002, 202);

INSERT INTO classes(class\_id, course\_id, tea\_id) VALUES(6005, 3003, 203);

INSERT INTO classes(class\_id, course\_id, tea\_id) VALUES(6006, 3004, 205);

INSERT INTO classes(class\_id, course\_id, tea\_id) VALUES(6007, 3005, 202);

INSERT INTO classes(class\_id, course\_id, tea\_id) VALUES(6008, 3005, 204);

INSERT INTO classes(class\_id, course\_id, tea\_id) VALUES(6009, 3006, 201);

INSERT INTO classes(class\_id, course\_id, tea\_id) VALUES(6010, 3006, 202);

5.3创建触发器

-- 建立改变班级人数的辅助触发器

CREATE TRIGGER class\_number\_insert AFTER INSERT

ON select\_courses

FOR EACH ROW

BEGIN

IF NEW.class\_id IN (SELECT class\_id FROM classes) THEN

UPDATE classes SET number = number + 1

WHERE class\_id = NEW.class\_id;

END IF;

END;

-- 建立改变班级人数的辅助触发器

CREATE TRIGGER class\_number\_delete AFTER DELETE

ON select\_courses

FOR EACH ROW

BEGIN

IF OLD.class\_id IN (SELECT class\_id FROM classes) THEN

UPDATE classes SET number = number - 1

WHERE class\_id = OLD.class\_id;

END IF;

END;

-- CORE

-- 建立触发器，实现学生不能选择课程号相同的课程。

CREATE TRIGGER non\_repetition BEFORE INSERT

ON select\_courses

FOR EACH ROW

BEGIN

IF (SELECT course\_id

FROM classes

WHERE NEW.class\_id = class\_id)

IN

(SELECT classes.course\_id

FROM select\_courses

JOIN classes ON classes.class\_id = select\_courses.class\_id

WHERE stu\_id = NEW.stu\_id)

THEN

SIGNAL SQLSTATE 'HX000' SET MESSAGE\_TEXT = 'course was selected';

END IF;

END;

5.4 查询某学年没有被选修的课程

-- 创建视图，查询2006－2007学年没有被选修的课程

-- 这里“未选修”对应到表上就是，课程人数为0。2006-2007学年即2006-2007-1和2006-2007-2两个学期。

CREATE VIEW not\_selected\_2006\_2007

AS

SELECT class\_id, course\_name, name

FROM classes

JOIN courses c on classes.course\_id = c.course\_id

JOIN teathers t on classes.tea\_id = t.tea\_id

WHERE number = 0 AND open\_time in ('2006-2007-1','2006-2007-2');

5.5 查询指定学期学生的学分获取情况

-- 创建存储过程，查看指定学期的学生学分

CREATE

PROCEDURE get\_total\_credit(IN term VARCHAR(20))

BEGIN

SELECT students.stu\_id, students.name, SUM(courses.credit)

FROM select\_courses

JOIN classes ON classes.class\_id = select\_courses.class\_id

JOIN students ON select\_courses.stu\_id = students.stu\_id

JOIN courses ON classes.course\_id = courses.course\_id

WHERE courses.open\_time = term

GROUP BY students.stu\_id, name;

END;

5.6 查询各个教师任课的学生数量

-- 创建存储过程，各教师任课的学生人数

CREATE PROCEDURE num\_of\_students()

BEGIN

SELECT t.name, c.course\_name, SUM(number) num\_of\_stu

FROM classes

JOIN courses c on classes.course\_id = c.course\_id

JOIN teathers t on classes.tea\_id = t.tea\_id

GROUP BY c.course\_id, t.tea\_id;

END;

6 总结

通过这次课程设计，我们把已经就快遗忘地数据库知识重新复习一遍，这也让我们认识到了经常总结之前学到知识地重要性。这次看似简单地课程设计，将我们身上地很多不足之处都暴露了出来，也让我们认识到了自己想要提高自己需要往哪方面继续努力。这次课程设计是我们团队一起完成地，大家都有明确地分工，当然在这个过程中我们遇到了很多问题，并不是一个人去埋头苦干，而是提出来大家一起帮忙想办法去解决决问题地过程中，我们不断发现错误，不断改正，不断领悟，不断获取，终于顺利完成课程设计。此次课设让大家都有了一定地进步，更让我们明白了团队地重要性。课程设计诚然是一门专业课，给我们很多专业知识以及专业技能上地提升，同时又是一门讲道课，一门辩思课，给了我们很多思想，给了我们莫大地提升空间。因为能力有限，系统还有很多不足之处，如：程序中错误处理和输入控制不够全面，可能导致输入无效引发错误。但在整个开发过程中我们学到了很多地知识，增强了动手能力，积累了解决各种难题地经验。同时也让我们发现了自己地许多不足之处，有待在将来地工作中进一步地学习和完善，掌握和运用更多地知识。我们对学生选课系统有了较深刻地认识，也具备了一定地实际开发经验。认识到开发系统首先要明确用户地需求，然后进行系统分析，其次构建系统模型、建立完善地数据库，接着编写代码，最后测试系统。最后，课设让我们明白了许多，团队地不可替代性，知识地重要性，以及，对于不明白地问题，要积极地动脑、动手、当然还要及时地请教懂问题地人。

6.1 团队分工

|  |  |
| --- | --- |
| 成凯 | 数据库设计、参与课设文档编写、数据流图与E-R图绘制 |
| 韩开元 | 编写信息管理查询、参与课设文档编写 |

7 参考文献

[1]王小玲，刘卫国. 数据库应用基础教程，北京：中国铁道出版社，2008

[2]施伯乐，丁宝康，汪卫•数据库系统教程第3版，北京：高等教育出版社，2008

[3]李伟波，软件工程，武汉：武汉大学出版社，2006

[4]郑人杰 实用软件工程 北京：清华大学出版社，2003

[5]张海藩，软件工程导论 北京：清华大学出版社，2008