

# 2022-2023 学年冬季学期数据库基础与应用 实验报告

李昀哲 20123101

计算机工程与科学学院

## 目录

2022-2023 学年冬季学期数据库基础与应用实验报告 .....	1
一、 SQL 实验 .....	2
1. 数据库表 .....	2
2. 实验和结果 .....	4
3. SQL 实验体会 .....	14
二、 选课管理系统实验 .....	15
1. 需求分析 .....	15
2. 数据表简介 .....	15
3. 教师端 .....	19
4. 选课管理系统实验体会 .....	28

# 一、SQL 实验

## 1. 数据库表

- 学生表 student: 学号, 姓名, 性别, 出生日期, 籍贯, 手机号码, 院系号;

表 1-1: student

student_id	name	sex	date_of_birth	native_place	mobile_phone	dept_id
1101	李明	男	1993-03-06	上海	13613005486	02
1102	刘晓明	男	1992-12-08	安徽	18913457890	01
1103	张颖	女	1993-01-05	江苏	18826490423	01
1104	刘晶晶	女	1994-11-06	上海	13331934111	01
1105	刘成刚	男	1991-06-07	上海	18015872567	01
1106	李二丽	女	1993-05-04	江苏	18107620945	01
1107	张晓峰	男	1992-08-16	浙江	13912341078	01
1108	王刚	男	1990-02-24	浙江	18721653583	01

- 院系表 department: 院系号, 名称, 地址, 联系电话;

表 1-2: department

dept_id	dept_name	address	phone_code
01	计算机学院	上大东校区三号楼	65347567
02	通讯学院	上大东校区二号楼	65341234
03	材料学院	上大东校区四号楼	65347890
06	Null	Null	Null

- 教师表 teacher: 工号, 姓名, 性别, 出生日期, 职称, 基本工资, 院系号;

表 1-3: teacher

staff_id	name	sex	date_of_birth	professional ranks	salary	dept_id
0101	陈迪茂	男	1973-03-06	副教授	3567. 00	01
0102	马小红	女	1972-12-08	讲师	2845.00	01
0201	张心颖	女	1960-01-05	教授	4200.00	02
0103	吴宝钢	男	1980-11-06	讲师	2554.00	01

- 课程表 course: 课号, 课名, 学分, 学时, 院系号; (默认学分 4, 学时 40)

表 1-4: course

course_id	course_name	credit	credit_hours	dept_id
08305001	离散数学	4	40	01
08305002	数据库原理	4	50	01
08305003	数据结构	4	50	01
08305004	系统结构	6	60	01
08301001	分子物理学	4	40	03
08302001	通信学	3	30	02

- 开课表 classes: 学期, 课号, 工号, 上课时间;

表 1-5: classes

semester	course_id	staff_id	class_time
2012-2013 秋季	08305001	0103	星期三 5-8
2012-2013 冬季	08305002	0101	星期三 1-4
2012-2013 冬季	08305002	0102	星期三 1-4
2012-2013 冬季	08305002	0103	星期三 1-4
2012-2013 冬季	08305003	0102	星期五 5-8
2013-2014 秋季	08305004	0101	星期二 1-4
2013-2014 秋季	08305001	0102	星期一 5-8
2013-2014 冬季	08302001	0201	星期一 5-8
2022-2023 冬季	08302001	0201	星期二 3-4
2022-2023 冬季	08305001	0102	星期一 1-2
2012-2013 冬季	08305004	0101	星期二 3-4
2012-2013 冬季	08305001	0101	星期一 9-11

- 选课表 course selection: 学号, 学期, 课号, 工号, 平时成绩, 考试成绩, 总评成绩;  
(成绩范围 1-100)

表 6: course selection

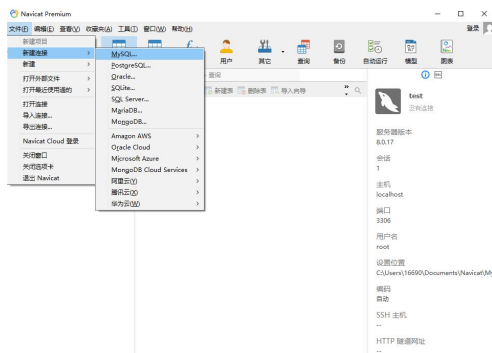
student_id	semester	course_id	staff_id	normal_score	test_score	total_score
1101	2012-2013 秋季	08305001	0103	60	60	60
1102	2012-2013 秋季	08305001	0103	87	87	87
1102	2012-2013 冬季	08305002	0101	82	82	82
1102	2013-2014 秋季	08305004	0101	null	null	null
1103	2012-2013 秋季	08305001	0103	56	56	56
1103	2012-2013 冬季	08305002	0102	75	75	75
1103	2012-2013 冬季	08305003	0102	84	84	84
1103	2013-2014 秋季	08305001	0102	null	null	null
1103	2013-2014 秋季	08305004	0101	null	null	null
1104	2012-2013 秋季	08305001	0103	74	74	74
1104	2013-2014 冬季	08302001	0201	null	null	null
1106	2012-2013 秋季	08305001	0103	85	85	85
1106	2012-2013 冬季	08305002	0103	66	66	66
1107	2012-2013 秋季	08305001	0103	90	90	90
1107	2012-2013 冬季	08305003	0102	79	79	79
1107	2013-2014 秋季	08305004	0101	null	null	null
1103	2013-2014 冬季	08301001	null	88	88	88
1103	2013-2014 冬季	08302001	0201	28	61	89
1103	2013-2014 冬季	08305004	0101	86	86	86
1107	2022-2023 冬季	08305001	0102	null	null	null
1102	2012-2013 冬季	08305004	0101	96	88	90
1101	2012-2013 冬季	08305004	0101	58	57	57

1103	2012-2013 冬季	08305004	0101	78	79	78
1104	2012-2013 冬季	08305004	0101	88	86	86
1105	2012-2013 冬季	08305002	0101	99	88	91
1105	2012-2013 秋季	08305001	0103	null	null	null
1108	2012-2013 秋季	08305001	0103	null	null	null

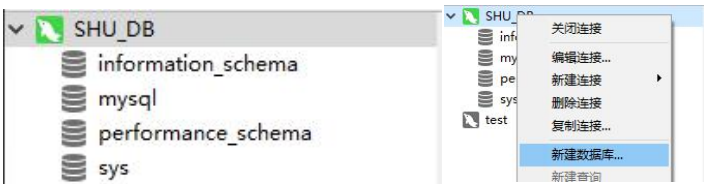
## 2. 实验和结果

### 2.1 新建数据库

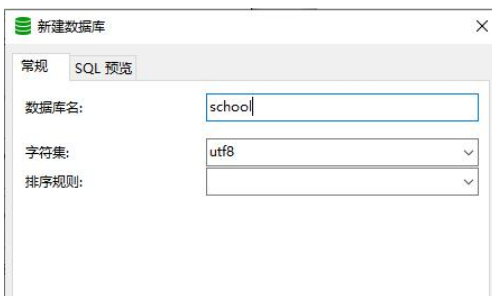
Navicat 下新建 Mysql 连接



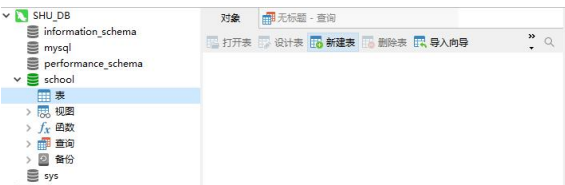
创建连接“SHU\_DB”后如图所示，同时新建数据库



新建 school 数据库

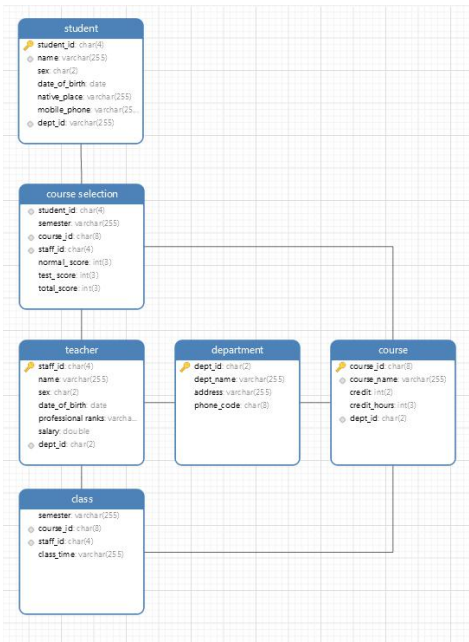


新建表，每张表根据要求编写 SQL 语言



```
create table student (  
  student_id char(4) not null primary key,  
  name varchar(255),  
  sex char(2),  
  date_of_birth date,  
  native_place varchar(255),  
  mobile_phone VARCHAR(255),  
  dept_id VARCHAR(255)  
);
```

各表关联如图所示：



2.2 SQL 实验及结果

1. 在学生表中建立索引 idx1:院系号升序，姓名降序；在课程表中建立索引 idx2：课名 SQL:

```
CREATE index idx1 ON student(dept_id asc, student.`name` DESC);  
CREATE index idx2 on course(course_name)
```

结果：

学生表中的索引

名	字段	索引类型	索引方法	注释
idx1	`dept_id`,`name`	NORMAL	BTREE	

课程表中的索引

字段	索引	外键	触发器	选项	注释	SQL 预览
名	字段	索引类型	索引方法	注释		
idx2	`course_name`	NORMAL	BTREE			

2. 查询 2011 年进校年龄大于 20 岁的男学生的学号与姓名。

SQL:

```
SELECT student.student_id, student.`name` from student where sex='男' and (2011
-YEAR(date_of_birth)) > 20
```

结果:

student_id	name	date_of_birth
1108	王刚	1990-02-24

---

3. 检索刘晓明不学的课程的课程号。

SQL:

```
select DISTINCT course_id
from course_selection
where course_id not in
(
SELECT DISTINCT course_id from course_selection join student
where course_selection.student_id = student.student_id and student.`name` = '刘晓明'
)
```

结果:

course_id
08301001
08302001
08305003

---

4. 检索马小红老师所授课程的学年, 学期, 课程号, 上课时间。

SQL:

```
SELECT semester, course_id, class_time
from classes join teacher
where classes.staff_id = teacher.staff_id
and teacher.`name` = '马小红'
```

结果:

semester	course_id	class_time
2012-2013冬季	08305002	星期三1-4
2012-2013冬季	08305003	星期五5-8
2013-2014秋季	08305001	星期一5-8
2022-2023冬季	08305001	星期一1-2

---

5. 查询计算机学院男生总评成绩及格、教授开设的课程的课程号、课名、开课教师姓名，按开课教师升序，课程号降序排序。

SQL:

```
SELECT DISTINCT course.course_id, course_name, teacher.`name`
from classes JOIN course join teacher
where course.course_id = classes.course_id
      and classes.staff_id = teacher.staff_id
      and teacher.staff_id in (
SELECT DISTINCT staff_id
from course_selection join student join department
where   course_selection.student_id = student.student_id
      and student.dept_id = department.dept_id
      and      student.sex = '男' and dept_name = '计算机学院'
      and total_score > 60
) ORDER BY teacher.`name` asc, course_id DESC
```

结果:

course_id	course_name	name
08305002	数据库原理	吴宝钢
08305001	离散数学	吴宝钢
08305004	系统结构	陈迪茂
08305002	数据库原理	陈迪茂
08305001	离散数学	陈迪茂
08305003	数据结构	马小红
08305002	数据库原理	马小红
08305001	离散数学	马小红

6. 检索学号比张颖同学大，年龄比张颖同学小的同学学号、姓名。

SQL:

```
SELECT student_id, student.`name`
from student
WHERE student_id >
( SELECT student_id from student where student.`name` = '张颖')
and   date_of_birth >
( SELECT date_of_birth from student where student.`name` = '张颖')
```

结果:

student_id	name
1104	刘晶晶
1106	李二丽

---

7. 检索同时选修了“08305001”和“08305002”的学生学号和姓名。

SQL:

```
SELECT DISTINCT x.student_id, student.`name`
from student, course_selection x, course_selection y
where student.student_id = x.student_id
      and x.student_id = y.student_id
      and x.course_id = '08305001'
      and y.course_id = '08305002'
```

结果:

student_id	name
1102	刘晓明
1103	张颖
1106	李二丽
1105	刘成刚

---

8. 验证在 1000 万个以上记录时在索引和不索引时的查询时间区别。

SQL:

```
-- 数据表创建 test5()

create table index_test(id int, major VARCHAR(255)); # 创建一个新表用于测试

DELIMITER //      # 使用 DELIMITER 关键字临时声明修改 SQL 语句的结束符为//

CREATE PROCEDURE test5(in args int)

begin

declare i int default 1;

start transaction;

while i <= args do

insert into index_test(id,major) value(i,concat("数据库-",i));

set i = i+ 1;

end while;

commit;

end //
```

---



```
DELIMITER ;
```

```
call test5(10000000)
```

```
-- 创建数据表
```

```
call test5(10000000)
```

```
select * from index_test where major='数据库-1'
```

```
create index major on index_test(major DESC)
```

```
select * from index_test where major='数据库-1'
```

结果：

创建表所需的时间

```
call test5(10000000)
> OK
> 时间: 293.404s
```

建立索引前

```
select * from index_test where major='数据库-1'
> OK
> 时间: 4.784s
```

建立索引后

```
select * from index_test where major='数据库-1'
> OK
> 时间: 0.001s
```

---

## 9. 查询每个学生选课情况（包括没有选修课程的学生）。

SQL:

```
select DISTINCT student.student_id, course_id
```

```
from student LEFT JOIN course_selection on student.student_id =
```

```
course_selection.student_id
```

```
ORDER BY student.student_id
```

结果：

student_id	course_id
1101	08305001
1101	08305004
1102	08305001
1102	08305002
1102	08305004
1103	08305001
1103	08305002
1103	08305003
1103	08305004
1103	08301001
1103	08302001
1104	08305001
1104	08302001
1104	08305004

---

10. 检索所有课程都选修的的学生的学号与姓名。

SQL:

```
-- exist 返回的是 true 或 false;没有一门课没有被该生选择

SELECT student.student_id, student.`name`

from student

where not EXISTS

(   SELECT * from course

    where not EXISTS

    (SELECT * from course_selection

     where student.student_id = course_selection.student_id

        and course_selection.course_id = course.course_id

    )

)
```

结果:

student_id	name
1103	张颖

---

11. 检索选修课程包含 1106 同学所学全部课程的学生学号和姓名。

SQL:

```
select student.student_id, student.`name`

from student

where student_id in

(   select DISTINCT student_id from course_selection

    where course_id in

    (select course_id from course_selection where student_id = '1106')

)
```

结果:

student_id	name
1105	刘成刚
1102	刘晓明
1104	刘晶晶
1107	张晓峰
1103	张颖
1106	李二丽
1108	王刚
1101	李明

---

12. 查询每门课程中分数最高的学生学号和学生姓名。

SQL:

```
select DISTINCT semester, course_id, student.student_id, student.name
from student, course_selection
where
    student.student_id = course_selection.student_id and
    course_selection.total_score in
    (select max(total_score)
    from course_selection
    GROUP BY course_id)
```

结果:

semester	course_id	student_id	name
2012-2013冬季	08305002	1105	刘成刚
2012-2013冬季	08305004	1102	刘晓明
2012-2013秋季	08305001	1107	张晓峰
2013-2014秋季	08305004	1107	张晓峰
2013-2014冬季	08305003	1103	张颖
2013-2014冬季	08301001	1103	张颖
2013-2014冬季	08302001	1103	张颖

---

13. 查询年龄小于本学院平均年龄，所有课程总评成绩都高于所选课程平均总评成绩的学生学号、姓名和平均总评成绩，按年龄排序。

SQL:

```
select *
from(select cs.student_id, s.`name`, avg(cs.total_score), date_of_birth
    from course_selection cs, student s
    where cs.student_id = s.student_id
    and s.date_of_birth > (SELECT avg(date_of_birth) from student s2 where
s2.dept_id=s.dept_id)
    and s.date_of_birth > all (SELECT avg(cs2.total_score) from course_selection cs2
where cs2.course_id = cs.course_id) GROUP BY cs.student_id
    having sum(course_id) = (SELECT sum(cs3.course_id) from course_selection cs3 where
cs3.student_id = cs.student_id)) as t
ORDER BY date_of_birth desc
```

结果:

---

student_id	name	avg(cs.total_score)	date_of_birth
1104	刘晶晶	53.3333	1994-11-06
1106	李二丽	75.5000	1993-05-04
1103	张颖	61.7778	1993-01-05

14. 建立计算机学院总评不及格成绩学生的视图，包括学生学号、姓名、性别、手机、所选课程和成绩。

SQL:

```
create view student_fail(student_id, name, sex, mobile_phone, course_id, grade) as (
    SELECT student.student_id, student.name, sex, mobile_phone, course_id, total_score
    from student, course_selection
    where student.student_id = course_selection.student_id and total_score < 60)
```

结果:

student_id	name	sex	mobile_phone	course_id	grade
1102	刘晓明	男	18913457890	08305004	0
1103	张颖	女	18826490423	08305001	56
1103	张颖	女	18826490423	08305001	0
1103	张颖	女	18826490423	08305004	0
1104	刘晶晶	女	13331934111	08302001	0
1101	李明	男	13613005486	08305004	57

15. 在 E 表中插入记录，把每个学生没学过的课程都插入到 E 表中，使得每个学生都选修每门课。

SQL:

```
insert into course_selection(student_id, semester, course_id, staff_id)
select s.student_id, c.semester, c.course_id, c.staff_id
from student s cross join classes c
where not exists(select * from course_selection cs where s.student_id = cs.student_id and
cs.course_id = c.course_id)
and c.semester = '2012-2013 秋季'
```

结果:

1102	2012-2013冬季	08305004
1101	2012-2013冬季	08305004
1103	2012-2013冬季	08305004
1104	2012-2013冬季	08305004
1105	2012-2013冬季	08305002
1105	2012-2013秋季	08305001
1108	2012-2013秋季	08305001

16. 求年龄大于所有女同学年龄的男学生姓名和年龄。

SQL:

```
select student.name, 2011-year(date_of_birth) 年龄 from student where sex = '男' and  
2011-year(date_of_birth) > (select max(2011-year(date_of_birth)) from student where sex='女')
```

结果:

name	年龄
刘晓明	19
刘成刚	20
张晓峰	19
王刚	21

---

17. 在选课表中修改 08305001 课程的平时成绩, 若成绩小于等于 75 分时提高 5%, 若成绩大于 75 分时提高 4%。

SQL:

```
update course_selection set normal_score = normal_score * 1.04 where normal_score > 75 and  
course_id='08305001'
```

```
update course_selection set normal_score = normal_score * 1.05 where normal_score <= 75 and  
course_id='08305001'
```

结果:

```
update course_selection set normal_score = normal_score * 1.04 where  
normal_score > 75 and course_id='08305001'  
> Affected rows: 3  
> 时间: 0.01s
```

```
update course_selection set normal_score = normal_score * 1.05 where  
normal_score <= 75 and course_id='08305001'  
> Affected rows: 3  
> 时间: 0.01s
```

---

18. 删除没有开课的学院。

SQL:

```
delete from department  
  
where dept_id not in (  
  
select dept_id from course_selection cs, course c  
  
where cs.course_id = c.course_id)
```

结果:

```
delete  
from department  
where dept_id not in (  
select dept_id  
from course_selection cs,  
course c  
where cs.course_id = c.course_id)  
> Affected rows: 1  
> 时间: 0.003s
```

---

dept_id	dept_name	address	phone_code
01	计算机学院	上大东校区三号楼	65347567
02	通讯学院	上大东校区二号楼	65341234
03	材料学院	上大东校区四号楼	65347890
06	(Null)	(Null)	(Null)

dept_id	dept_name	address	phone_code
01	计算机学院	上大东校区三号楼	65347567
02	通讯学院	上大东校区二号楼	65341234
03	材料学院	上大东校区四号楼	65347890

## 19. 查询优、良、中、及格、不及格学生人数。

SQL:

```
select sum(case when total_score < 60 then 1 else 0 end) as'不及格',
       sum(case when total_score >=60 and total_score <70 then 1 else 0 end) as'合格',
       sum(case when total_score >=70 and total_score <80 then 1 else 0 end) as'中',
       sum(case when total_score >=80 and total_score <90 then 1 else 0 end) as'良',
       sum(case when total_score >=90 and total_score <100 then 1 else 0 end) as'优秀'
from course_selection
```

结果:

不及格	合格	中	良	优秀
6	2	4	8	4

## 3. SQL 实验体会

SQL 实验的开始是在课堂大致讲解完理论基础后进行的，当时说实话对于整个理论还一知半解，因此通过实验逐步构建了整个知识框架，对于后续的学习起到了基石的作用。SQL 实验寒假前后各做了一遍，寒假前主要是将课堂所学用于实践，动手尝试 SQL 的写法和作用，以熟悉语法为主，因此在语句编写的过程中还是遇到了较大的困难，写一条 SQL 的耗时也很长，最后的结果往往也不尽如人意；但在寒假后，尤其是在“选课管理系统”实验进行中的时候，对于 SQL 的理解突然茅塞顿开，简单的增删改查随手就能写出，较为复杂的查询也能根据要求拆解，较为顺利、有逻辑地写出来，结果也能满足要求。

可以说 SQL 语句的熟练掌握是在“选课系统”实验中不断练习得来的，两个实验一定程度上也起到了相辅相成的效果。

SQL 语句是使用数据库的基础，就好像是练就某项武功的葵花宝典，反复使用、操练，才能不断在实践中发现自己的不足和这项技术的核心。实验过程虽然付出了很多时间，但获得的锻炼，无论是技术上还是心理上，都对未来的工作和学习生活非常有帮助。

## 二、选课管理系统实验

### 1. 需求分析

选课管理系统是高校课程管理的重要基础，本课程以选课管理系统作为数据库原理的实践平台。基于我们日常对于上海大学选课系统的使用，我们认为学生主要应用该平台选课、退课、查看课表等功能；教师主要进行开课、查看课程实时选课情况、录入和修改成绩、查看成绩统计等；管理员主要负责新增课程、管理学生、管理教师。

鉴于上述功能，我们将系统分为学生端、教师端和管理员端。以不同身份登录系统将展示不同页面同时获得不同权限，以提高系统安全性。在用户输入信息的验证方面，我们充分使用前端验证功能、创建触发器和存储功能保证完整性。同时进行了大量测试，尽可能地避免因设计不当导致的系统崩溃和错误。

技术路线上，本选课管理系统实验技术栈基于 Vue+SpringBoot+MySQL，编译工具使用 WebStorm2022.2.3 和 IntelliJ IDEA 2022.2.3。

### 2. 数据表简介

#### 2.1 概念设计

实体关系的分析上，每个班级=包含多个学生，由教师讲授、学生选择；同一个老师可能开设多个课程；需要考虑不同学年和学期；教师、学生、某门课程隶属于某个学院；学生、老师、课程、学院受管理员管理。由以上分析可以得到以下 ER 图。

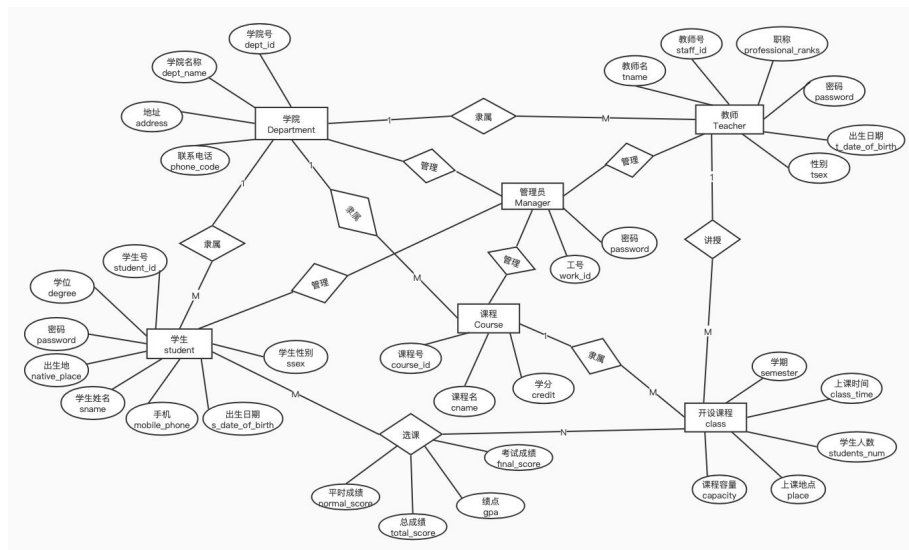


图 2-1 选课管理系统 ER 图

## 2.2 逻辑设计

数据表以实验一的 school 数据库为基础，包含 student, teacher, department, course, class, course\_selection 几张表，其中的字段在原有基础上根据实际需求进行了修改，各表结构介绍如下表所示：

将 E-R 图转换为以下关系模式：

- 学院（学员编号，学院名称，地址，联系电话）
- 教师（教师编号，教师姓名，教师性别，教师出生日期，教师职称，所属学院编号，用户密码）
- 班级（开课学期，课程编号，教师编号，上课时间，课程容量，上课地点，学生人数）
- 课程(课程编号，课程名称，课程学分，所属学院编号)
- 学生（学生学号，学生姓名，学生性别，学生出生日期，出生地，手机，所属学院编号，学位，用户密码）
- 开课情况(学生学号，开课学期，课程编号，教师编号，平时成绩，考试成绩，总成绩，绩点)

根据上述关系模式，数据字段类型的定义和约束如下所示：

表 2-1 Department

字段名	字段类型	约束条件
学院编号	dept_id: varchar	主键；非空；最大长度 20



学院名称	dept_name: varchar	非空；最大长度 20
地址	address: varchar	非空；最大长度 20
联系电话	phone_code: varchar	非空；最大长度 20

教师表字段中加入用户密码，用以登录，默认密码为 123，进入系统后用户可自行修改；

表 2-2 Teacher

字段名	字段类型	约束条件
教师编号	staff_id: varchar	<b>主键</b> ；非空；最大长度 20
教师姓名	tname: varchar	非空；最大长度 20
教师性别	tsex: varchar	非空；最大长度 20
教师出生日期	t_date_of_birth: date	非空
教师职称	professional_ranks	非空；最大长度 20
所属学院编号	dept_id: varchar	<b>外键</b> ；非空；最大长度 20
用户密码	password: varchar	非空；最大长度 20

开课表字段中加入课程容量和学生人数，用以为教师提供开课时的容量选择和实时统计某节课的选课人数；

表 2-3 Class

字段名	字段类型	约束条件
开课学期	semester: varchar	<b>联合主键</b> ；非空；最大长度 20
课程编号	course_id: varchar	<b>联合主键</b> ； <b>外键</b> ；非空；最大长度 20
教师编号	staff_id: varchar	<b>联合主键</b> ； <b>外键</b> ；非空；最大长度 20
上课时间	class_time:varchar	非空；最大长度 20
课程容量	capacity: int	非空；最大长度 11
上课地点	place: varchar	非空；最大长度 20
学生人数	students_num: int	非空；最大长度 11

表 2-4 Course

字段名	字段类型	约束条件
课程编号	course_id: varchar	<b>主键</b> ；非空；最大长度 20
课程名称	cname:varchar	非空；最大长度 20
课程学分	credit: int	非空；最大长度 11

所属学院编号	dept_id: varchar	外键；非空；最大长度 20
--------	------------------	---------------

学生表字段中新增学位和用户密码，用以管理员录入学生信息和登录，用户进入系统后可以自行修改密码；

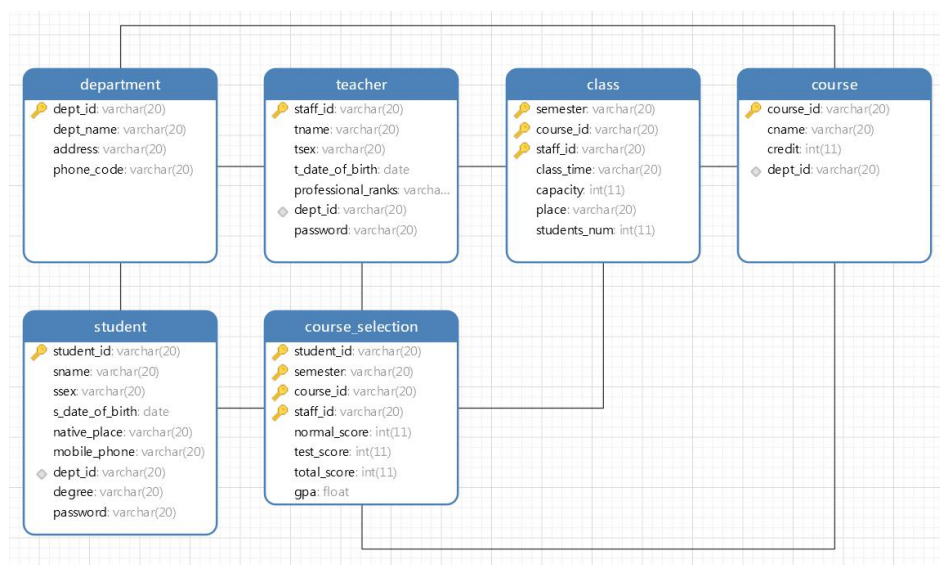
表 2-5 Student

字段名	字段类型	约束条件
学生学号	student_id: varchar	主键；非空；最大长度 20
学生姓名	sname: varchar	非空；最大长度 20
学生性别	ssex: varchar	非空；最大长度 20
学生出生日期	s_date_of_birth: date	非空
出生地	native_place: varchar	非空；最大长度 20
手机	mobile_phone: varchar	非空；最大长度 20
所属学院编号	dept_id: varchar	外键；非空；最大长度 20
学位	degree: varchar	非空；最大长度 20
用户密码	password: varchar	非空；最大长度 20

选课表字段中新增绩点字段，用以计算和显示学生基于总成绩的绩点；

表 2-6 Course\_selection

字段名	字段类型	约束条件
学生学号	student_id: varchar	联合主键；外键；非空；最大长度 20
开课学期	semester: varchar	联合主键；非空；最大长度 20
课程编号	course_id: varchar	联合主键；外键；非空；最大长度 20
教师编号	staff_id: varchar	联合主键；外键；非空；最大长度 20
平时成绩	normal_score: int	非空；最大长度 11
考试成绩	final_score: int	非空；最大长度 11
总成绩	total_score: int	非空；最大长度 11
绩点	gpa: varchar	非空；最大长度 20



## 3. 教师端

### 3.1 界面及功能

#### 3.1.1 登录

登录界面模仿上海大学本硕博一体化管理系统，添加选课卡通图图标和身份选择功能。用户输入学工号、密码和选择身份后，会调用数据库中相应的表进行验证，信息错误将会在页面顶部得到提示，正确则可进入后续学期选择页面。

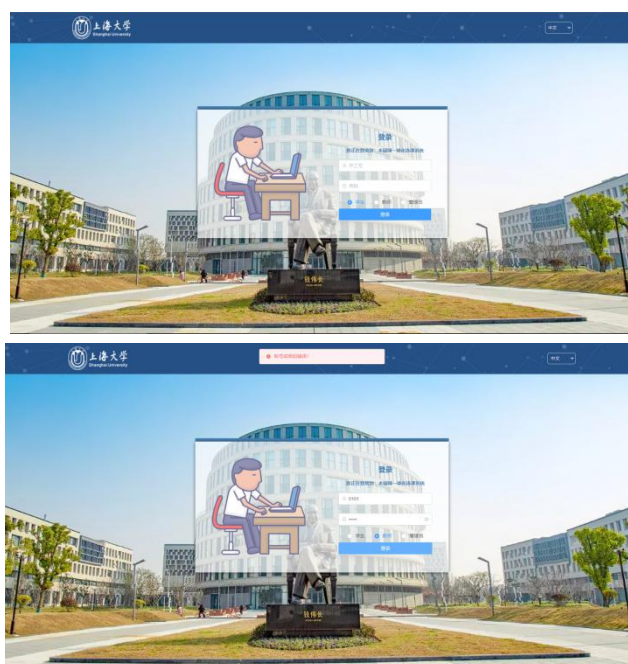


图 2-2 登陆界面

3.1.2 学期选择

学期选择界面同样模仿上海大学本硕博一体化选课系统,管理员拥有开放某个学期的权限,教师和学生仅能通过开放的学期进行选择。



图 2-3 学期选择界面

3.1.3 教师开课

教师开课页面可以根据当前学期展示该老师已经开设的课程数量,此处使用存储过程进行计算。同时还可以查看可以开设的课程,本项目中未设置教师限定开课的学院,即默认所有学院的课程教师都可以开设,对于该限制的改动,可以根据实际需求进行修改。

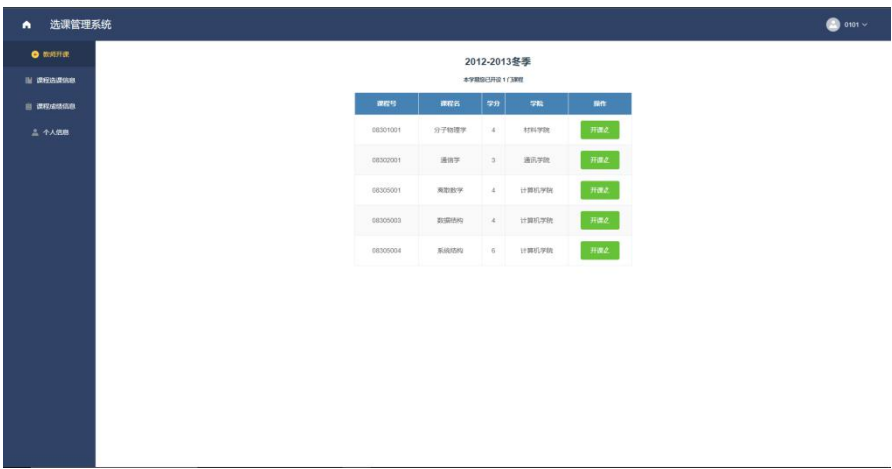


图 2-4 开课界面

点击某门课程的开课按钮后,即可在弹窗中进行课程开设,填写上课时间、课程容量和上课地点后即可完成开设。注意,该功能实际在管理员端较为合理,但鉴于任务分配和管理员端同学的实际技术情况,交由教师端完成:

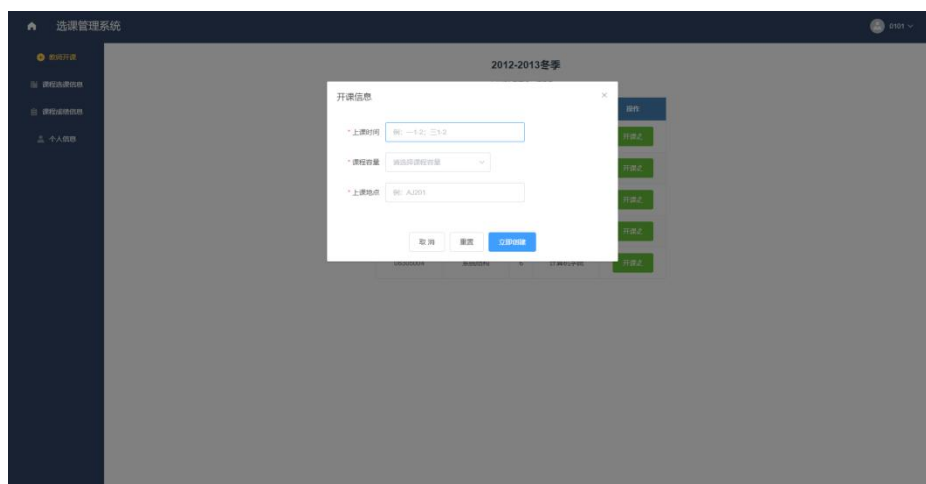


图 2-5 开课表单

对于未按要求填写的表单，将进行提示：



图 2-6 错误信息提示及正确填写

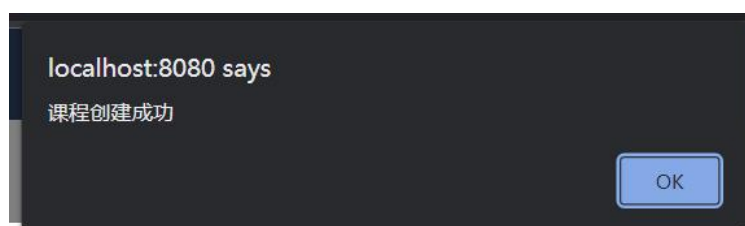
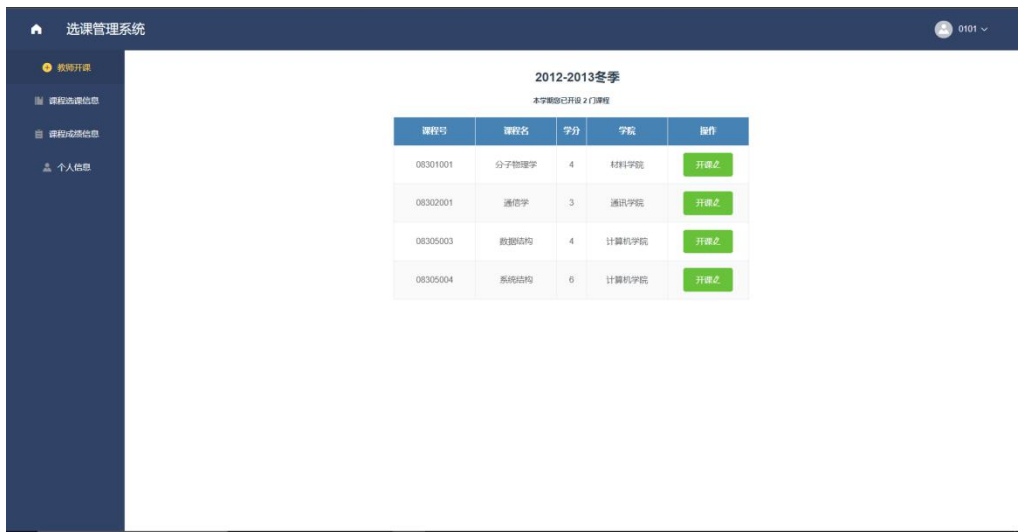


图 2-7 创建成功提示

成功创建课程后，开课页面的可开课表和已经开设的课程数量将会更新；



2012-2013冬季				
本学期已开设 2 门课程				
课程号	课程名	学分	学院	操作
08301001	分子物理学	4	材料学院	开课
08302001	通信学	3	通信学院	开课
08305003	数据结构	4	计算机学院	开课
08305004	系统结构	6	计算机学院	开课

图 2-8 成功开设“离散数学”后的开课页面

### 3.1.4 查看课程选课信息

此界面教师可以查看当前学期自己开课的情况，和实时选课人数，点击查看按钮可以查看选课学生的详细学院信息和姓名；



2012-2013冬季						
学期	课程号	课程名	上课时间	上课地点	当前选课人数	操作
2012-2013冬季	08305001	离散数学	二3-4	BJ203	2	查看
2012-2013冬季	08305002	数据库原理	三1-4	C304	1	查看

图 2-9 课程选课信息



图 2-10 学生选课信息详细查看

顶部的搜索框可以根据课程名和课程号进行模糊搜索以在教师开课较多的情况下，帮助教师迅速查找所需查看的课程；

2012-2013冬季						
<input type="text" value="1"/>		<input type="button" value="搜索"/>	<input type="text" value="请输入课程名"/>		<input type="button" value="搜索"/>	
学期	课程号	课程名	上课时间	上课地点	当前选课人数	操作
2012-2013冬季	08305001	离散数学	二3-4	BJ203	2	<input type="button" value="查看"/>

图 2-11 根据课程号模糊搜索

2012-2013冬季						
<input type="text" value="请输入课程号"/>		<input type="button" value="搜索"/>	<input type="text" value="数据库"/>		<input type="button" value="搜索"/>	
学期	课程号	课程名	上课时间	上课地点	当前选课人数	操作
2012-2013冬季	08305002	数据库原理	三1-4	C304	1	<input type="button" value="查看"/>

图 2-12 根据课程名模糊搜索

### 3.1.5 查看课程成绩信息

此界面教师可以查看当前学期自己开课的成绩统计信息，分为优秀、合格和不合格；顶部搜索框同 3.1.4 一样，可以根据课程号和课程名进行模糊搜索以快速定位课程。

选课管理系统						
2012-2013冬季						
<input type="text" value="请输入课程号"/> 搜索 <input type="text" value="请输入课程名"/> 搜索						
课程号	课程名	选课人数	优秀人数	合格人数	不合格人数	操作
08305001	离散数学	2	1	1	0	查看
08305002	数据库原理	1	0	1	0	查看

图 2-13 课程成绩信息

点击查看按钮可以查看该课程所有学生成绩信息并进行修改和录入；同时顶部会实时更新各个成绩段人数情况和均分，帮助教师得到及时的成绩统计；

选课管理系统						
2012-2013冬季						
<input type="text" value="请输入学号"/> 搜索 <input type="text" value="请输入姓名"/> 搜索						
课程号	课程名	4.0	3.7-3.9	3.0	均分	
08305001	离散数学	1	1	0	81.5	
学号	姓名	学院	平时成绩	期末成绩	总评	gpa
1101	李明	02	90	90	90	0.0
1107	张晓明	01	90	67	73	2.0

图 2-14 某课程成绩修改及录入界面

点击编辑按钮将会展示成绩录入及修改弹窗，为便于教师计算，在对平时成绩和期末成绩输入框内的数据进行改动后，点击计算按钮可以自动计算总分，如总分不合适，无需反复关闭和打开此弹窗，支持便捷修改。同时，提交成绩后将会通过触发器自动将成绩转换为绩点，绩点和成绩的对应情况如下表所示。



```
CREATE TRIGGER setGPA before update on course_selection for each row
BEGIN
IF new.total_score >= 90
THEN set new.gpa = 4.0;
ELSEIF new.total_score >= 85 and new.total_score < 90
THEN set new.gpa = 3.7;
ELSEIF new.total_score >= 80 and new.total_score < 85
THEN set new.gpa = 3.0;
ELSEIF new.total_score >= 75 and new.total_score < 80
THEN set new.gpa = 2.7;
ELSEIF new.total_score >= 70 and new.total_score < 75
THEN set new.gpa = 2.0;
ELSEIF new.total_score >= 60 and new.total_score < 70
THEN set new.gpa = 1.0;
ELSEIF new.total_score < 60
THEN set new.gpa = 0.0;
END IF;
END
```

图 2-15 成绩绩点对应触发器

表 2-7 触发器成绩绩点对应表

成绩	绩点
$100 \geq \text{score} \geq 90$	4.0
$90 > \text{score} \geq 85$	3.7
$85 > \text{score} \geq 80$	3.0
$80 > \text{score} \geq 75$	2.7
$75 > \text{score} \geq 70$	2.0
$70 > \text{score} \geq 60$	1.0
$\leq 60$	0.0

成绩的提交会通过触发器自动尽力将学生平时成绩提高以保证及格，该功能将在下一小节进行展示；

1101

平时成绩 90

期末成绩 93

总分 90

取消 计算 重置 提交

2012-2013冬季

请输入学号 搜索

请输入姓名 搜索

课程号	课程名	4.0	3.7-1.0	0.0	均分
08305001	离散数学	1	1	0	82.5

学号	姓名	学院	平时成绩	期末成绩	总评	gpa	操作
1101	李明	02	90	93	92	4.0	修改
1107	张丽峰	01	90	67	73	2.0	修改

图 2-16 成绩录入及修改

顶部搜索框可以根据学生学号和姓名进行模糊搜索以快速定位学生进行录入和修改；

2012-2013冬季

01 搜索

请输入姓名 搜索

课程号	课程名	4.0	3.7-1.0	0.0	均分
08305001	离散数学	1	1	0	82.5

学号	姓名	学院	平时成绩	期末成绩	总评	gpa	操作
1101	李明	02	90	93	92		修改

图 2-17 根据学号模糊搜索

2012-2013冬季

请输入学号

搜索

晓峰

搜索

课程号	课程名	4.0	3.7-1.0	0.0	均分
08305001	离散数学	1	1	0	82.5

学号	姓名	学院	平时成绩	期末成绩	总评	gpa	操作
1107	张晓峰	01	90	67	73		编辑

图 2-18 根据姓名模糊搜索

### 3.1.6 查看个人信息

个人信息界面未赋予教师太多的权限,仅限于教师基本信息的展示和个人用户密码的修改,通过前端技术,对于密码默认为不可见,可通过点击输入框的小眼睛进行查看:



图 2-19 个人信息界面



图 2-20 密码修改前后

### 3.2 有趣功能

**辅助拉分：**对于不及格的同学，系统会倾向增加该同学的平时分，对于考试成绩尚佳但平时成绩较低导致不及格的同学，我们认为他还是好同学，因此即使不及格，也会通过增长平时分的方式，将他强行及格。如下 1107 同学总成绩应为 57 分，但由于考试成绩较好，因此实际提交的成绩将会有所增加，但加分仅到 60 分即止。该功能通过触发器实现，实现代码见本节末尾。

1107

平时成绩

10

期末成绩

78

总分

57

取消

计算

重置

提交

1107	张晓峰	01	17	78	60	0.0	<div>编辑</div>
------	-----	----	----	----	----	-----	---------------

图 2-21 辅助拉分功能成功情况

但对于考试成绩不佳的同学，我们倾向于认为他是不认真的同学，因此即使平时成绩拉满，他同样不能及格；1101 同学该分数下并不能及格，但由于考试成绩过低，该辅助拉分功能也无能为力

1101

平时成绩

60

期末成绩

34

总分

41

取消

计算

重置

提交

1101	李明	02	100	34	54	0.0	<div>编辑</div>
------	----	----	-----	----	----	-----	---------------

图 2-22 辅助拉分功能失败情况

## 触发器实现

```
CREATE TRIGGER setTotalScore before update on course_selection for each row
BEGIN
declare outflag boolean DEFAULT(true);
WHILE new.total_score < 60 and new.normal_score != 100 DO
if new.normal_score + new.normal_score*0.05 <= 100
then set new.normal_score = new.normal_score + new.normal_score*0.05;
else set new.normal_score = 100;
end if;

set new.total_score = 0.3*new.normal_score + 0.7*new.test_score;
END WHILE;
end
```

图 2-23 辅助拉分触发器

## 4. 选课管理系统实验体会

本实验基于课堂所学数据库理论知识动手实践，将所学用于实践。实验中，对于原本书本上晦涩抽象的知识，如：数据库系统结构、结构化查询语言、实体联系模型、数据库设计、数据库管理等，有了更进一步的理解和体会。从实际需求出发，建立实体联系模型；经历概念设计、逻辑设计等数据库设计的必要阶段，对数据库设计、数据库系统结构加深了理解；实现功能过程中对 SQL 语言的增删改查功能更加熟悉，现在写来更加得心应手；最后学完第八章，又新增了触发器和存储过程使数据库的完整性得到保证。总的而言，本实验就像是一道综合题，将本学期所有知识串在了一起，在假期的实践中巩固了基础知识。

本实验贴近现实生活，上海大学本硕博选课管理系统自 2020 年入学以来，已经使用三年，本实验站在巨人肩膀上，模仿了部分体系成熟的功能和页面布局，对于教师部分由于并未实际使用，因此凭借自身理解、小组讨论、咨询老师等方式进行了调研以确定设计方案，设计中，不仅是对于技术的考验，更是对于沟通能力的培养。未来的学习和生活，永远不可能一人孤行，一定是以团队去实现，这方面的体验也在本实验中得到锻炼。

技术上，数据库是实现该系统的基础，但由于需要交付用户使用，界面和数据传递的技术也同样重要。假期中，通过小组讨论，我和刘沛根同学对于 Vue 和 SpringBoot 在其他项目中有过浅显了解和实践，且二者也是当下较为流行和实用的前后端框架，因此我们决定利用本实验对两个框架进行深入学习并完成实验，是一个双赢的方案。假期中我们不断交流学习资源和成果，交流踩过的坑，高效且高质量地完成了本次实验。

本实验使我学到技术，有了数据库应用的初体验，实现了自认为比较满意的选课管理系统，学习的过程中兼具做出成果的趣味，看着正常运行、页面布局优雅的网页，不由得心生快乐。可能学习也是如此，只要把自己的兴趣结合学业或工作，一切都会变得有趣，即使过程可能艰辛，但最终的成果一定会让自己觉得再辛苦也是值得的。无论如何，本实验的设置让我受益匪浅获益良多，希望未来还能遇到宋老师这样的老师和课程！