

---

# 哈尔滨工业大学

## <<数据库系统>>

### 实验报告二

(2023 年度春季学期)

姓名:	徐柯炎
学号:	2021110683
学院:	计算学部
教师:	程思瑶

## 实验二

### 一、实验目的

在熟练掌握 MySQL 基本命令、SQL 语言以及用 C 语言编写 MySQL 操作程序的基础上，学习简单数据库系统的设计方法，包括数据库概要设计、逻辑设计。

### 二、实验环境

Windows 操作系统、MYSQL 关系数据库管理系统、pycharm2023。

### 三、实验过程及结果

#### 1. 概念数据库设计：

本次数据库实验我构建了一个学院管理系统，具体有以下几个实体：

- (1) 学院（学院名，学院主任）
- (2) 系（系名，系编号，学院名）
- (3) 系主任（工号，姓名，联系方式）
- (4) 学生（学号，姓名）
- (5) 教师（教师工号，姓名）
- (6) 班级（班级号）
- (7) 宿舍（宿舍编号，宿舍名）
- (8) 课程（编号，课程名）

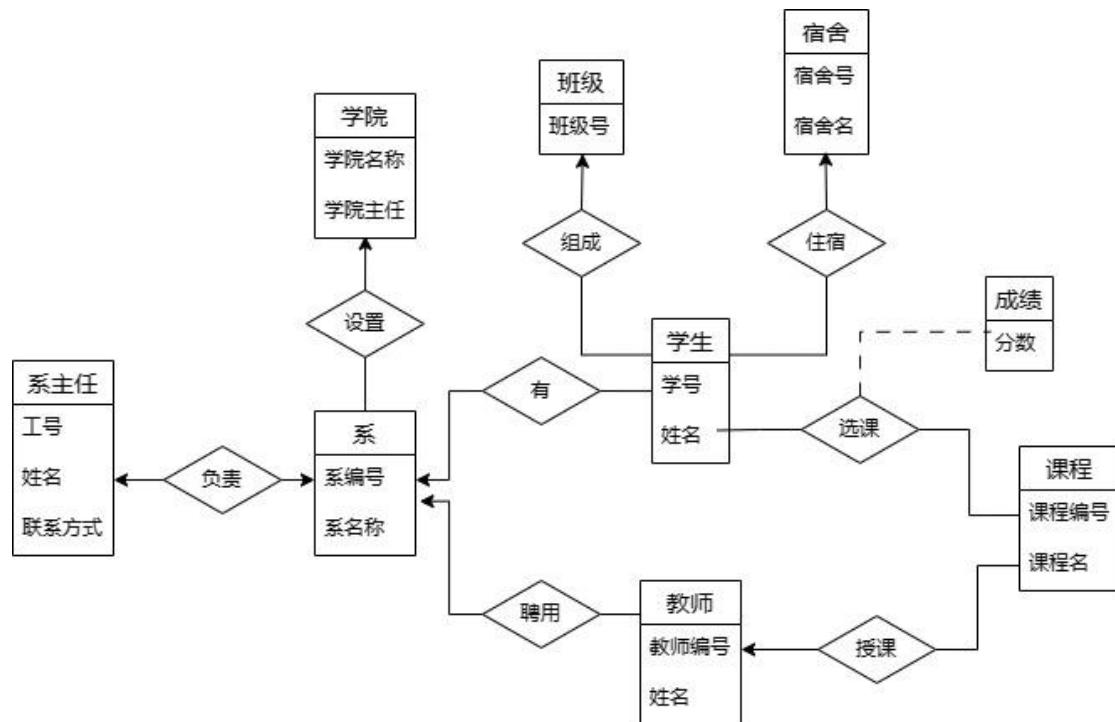
其中下划线表示该属性为此实体的主键。

在这几个实体之间存在以下联系：

- (1) 设置：一个系由一个学院设置，一个学院可以设置多个系；
- (2) 负责：一个系有且仅有一个系主任；
- (3) 有：一个学生属于一个系，一个系有多个学生；
- (4) 聘用：一个系聘用多个教师，一个教师由一个系设置；
- (5) 组成：一个班级由多个学生组成，一个学生属于一个班级；
- (6) 住宿：一个学生住在一个宿舍，一个宿舍有多个学生；
- (7) 选课：一个学生可以选多门课，一门课可有多多个学生选；
- (8) 授课：一个老师教授多门课，一门课只能有一个老师。

#### 2. ER 图绘制：

根据以上实体和联系可以画出以下 ER 图：



## 3. 逻辑数据库设计:

将所画的 ER 图转化为关系数据库模式, 可以得到如下几个关系:

1) 学院 college(coname, cdirector)

属性	含义	约束	数据类型
Coname	学院名称	非空, 主键	Varchar
Cdirector	学院主任	非空	Varchar

2) 系 department(deno, dename, coname)

属性	含义	约束	数据类型
Dename	系名	非空	Varchar
Deno	系编号	非空, 主键	Char
Coname	所属学院	非空, 外键	Varchar

其中 coname 参照 college.coname;

3) 系主任 director(dino, diname, contact, deno)

属性	含义	约束	数据类型
Dino	系主任编号	非空, 主键	Char
Diname	系主任名	非空	Varchar
Contact	联系方式	非空	Char
Deno	所属系	非空, 外键	Char

其中 deno 参照 department.deno;

4) 学生 student(sno, sname, deno, clno, dono)

属性	含义	约束	数据类型
----	----	----	------

Sno	学号	非空，主键	Char
Sname	姓名	非空	Varchar
Deno	所属系	非空，外键	Char
Clno	所属班级	非空，外键	Char
Dono	所属宿舍	非空，外键	Char

其中 deno 参照 department.deno, clno 参照 class.cln, dono 参照 dorm.dono;

#### 5) 教师 teacher(tno, tname, deno)

属性	含义	约束	数据类型
Tno	教师编号	非空，主键	Char
Tname	教师名称	非空	Varchar
Deno	所属系	非空，外键	Char

其中 deno 参照 department.deno;

#### 6) 班级 class(clno)

属性	含义	约束	数据类型
Clno	班级号	非空，主键	Char

#### 7) 宿舍 dorm(dono, doname)

属性	含义	约束	数据类型
Dono	宿舍编号	非空，主键	Char
Doname	宿舍名	非空	Varchar

#### 8) 课程 course(cno, cname, tno)

属性	含义	约束	数据类型
Cno	课程编号	非空，主键	Char
Cname	课程名	非空	Varchar
Tno	本课程教师	非空，外键	Char

其中 tno 参照 teacher.tno;

#### 9) 成绩 grade(cno, sno, gno)

属性	含义	约束	数据类型
Cno	课程编号	非空，主键，外键	Char
Sno	学生编号	非空，主键，外键	Char
Gno	成绩	非空	Int

其中 cno 参照 course.cno, sno 参照 student.sno。

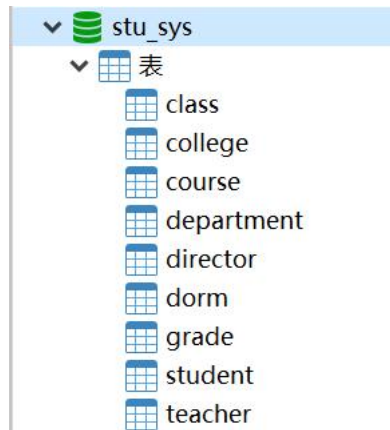
这些关系均满足 3NF 范式关系。

#### 4. 数据库具体实现:

本实验采用 ypython+mysql+pyqt5 可视化来进行实验。

##### (1) 在 mysql 中创建数据库

首先创建数据库并在数据库中添加上述关系，具体过程省略，如下图所示;



接着插入 txt 文本格式的数据：

名称	修改日期	类型
class.txt	2023/3/30 16:44	文本文档
college.txt	2023/3/30 16:44	文本文档
course.txt	2023/3/30 16:44	文本文档
department.txt	2023/3/30 16:44	文本文档
director.txt	2023/11/11 10:44	文本文档
dorm.txt	2023/3/30 16:44	文本文档
grade.txt	2023/3/30 16:44	文本文档
student.txt	2023/11/12 10:43	文本文档
teacher.txt	2023/11/12 10:46	文本文档

最终结果如下图所示：

对象	department @stu_sys (root) - 表	
开始事务	文本	筛选
排序	导入	导出
dename	deno	coname
计算机科学	301	Computing
软件工程	302	Computing
人工智能	303	Computing
信息安全	304	Computing
数据科学	305	Computing
物联网	306	Computing
生物信息	307	Computing
机械制造	308	Mechatronics
工业设计	309	Mechatronics
通信工程	310	Electronics
电子工程	311	Electronics
控制科学	312	Astronautics
航天航空	313	Astronautics

## (2) 可视化界面实现

由于本实验采用 `pyqt5` 进行可视化操作，因此会产生以下三种文件：

- 1) `Filename.py`: 可视化代码文件。
- 2) `Filename_.py`: 具体进行 `sql` 语句操作的文件；
- 3) `Filename.ui`: 可视化设计文件。

名称	修改日期	类型
.idea	2023/12/14 15:35	文件夹
_pycache_	2023/11/19 8:20	文件夹
add_student.py	2023/11/13 23:16	JetBrains PyCharm
add_student.ui	2023/11/13 23:16	UI 文件
add_student_.py	2023/11/13 23:33	JetBrains PyCharm
change.py	2023/11/13 23:17	JetBrains PyCharm
change.ui	2023/11/13 23:16	UI 文件
change_.py	2023/11/13 9:31	JetBrains PyCharm
drop_student.py	2023/11/13 9:21	JetBrains PyCharm
drop_student.ui	2023/11/12 23:09	UI 文件
drop_student_.py	2023/11/13 10:41	JetBrains PyCharm
index.py	2023/11/13 9:21	JetBrains PyCharm
index.ui	2023/11/12 23:24	UI 文件
index_.py	2023/11/13 15:06	JetBrains PyCharm
main_window.py	2023/11/13 15:23	JetBrains PyCharm
main_window.ui	2023/11/12 16:20	UI 文件

可视化界面如下图所示。



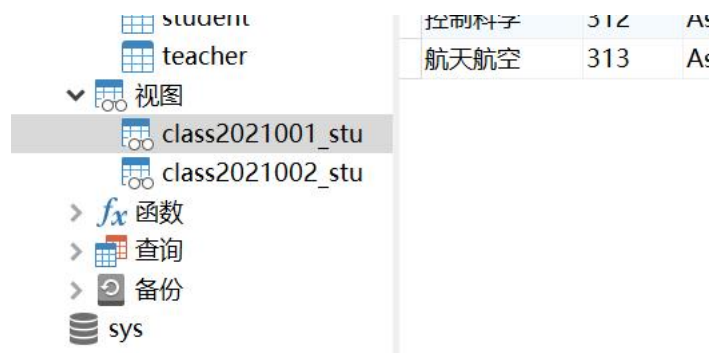
## (3) 检查点实现

- 1) 创建视图

本程序可以为某班级的所有学生创建视图，如下图所示：



视图创建结果如下：



## 2) 建立索引

本程序可以为学生的非键属性建立索引，如下图所示：

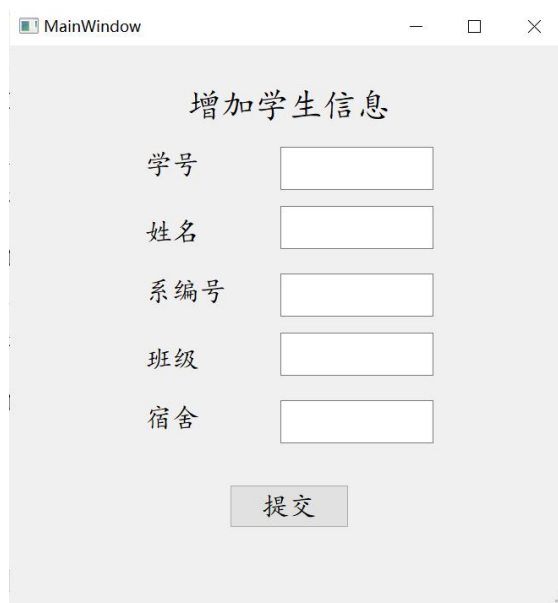


结果如下：

名	字段	索引类型
clno	`clno`	NORMAL
dono	`dono`	NORMAL
▶sname_index	`sname`	NORMAL
deno_index	`deno`	NORMAL

### 3) 插入

本程序可以为管理员提供插入学生信息的功能，如下图所示：



部分代码见下：

```
args = [sno, sname, deno, clno, dono]
if not sno or not sname or not deno or not clno or not dono:
    QMessageBox.warning(self, QWidget: '警告', p_str: '值不能为空')
    return
print(args)
sql = 'insert into student values(%s, %s, %s, %s, %s)'
cnt = cursor.execute(sql, args)
print(cnt)
QMessageBox.information(self, QWidget: '提示', p_str: '插入成功')
db.commit()
db.close()
```

### 4) 删除

本程序可以为管理员提供删除学生信息的功能，如下图所示：



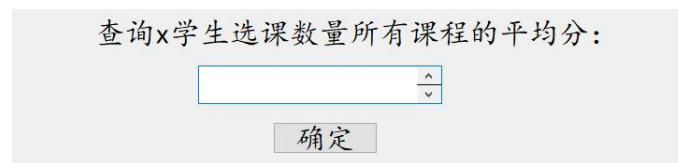


部分代码见下:

```
values = []
for i in range(len(key_all)):
    values.append(key_all[i][0])
# print(values)
if value not in values:
    QMessageBox.warning(self, QWidget: '警告', p_str: '数据库中不存在此值')
    return
sql = 'delete from student where ' + key + '=%s'
cnt = cursor.execute(sql, value)
QMessageBox.information(self, QWidget: '提示', p_str: f'删除成功, 删除了{cnt}条记录')
db.commit()
db.close()
```

##### 5) 连接查询

本程序可以为管理员提供查询学生选课数量以及平均分的功能，如下图所示：



代码见下:

```

def search_1(self):
    name = self.textEdit.toPlainText()
    print(name)
    if not name:
        QMessageBox.warning(self, QWidget: '警告', p_str: '请输入姓名')
    else:
        db = pymysql.connect(host='localhost', user='root', passwd='mysql', database='stu_sys')
        cursor = db.cursor()
        sql = 'select avg(gno), count(*) from student natural join grade where sname=%s'
        cnt = cursor.execute(sql, name)
        result = cursor.fetchall()
        if result[0][0] is None:
            QMessageBox.warning(self, QWidget: '警告', p_str: '查无此人')
            return
        print(result)
        str_avg = name + f'选了{result[0][1]}门课, 所有课程平均分为{round(result[0][0], 2)}'
        print(str_avg)
        QMessageBox.information(self, QWidget: '查询成功', str_avg)

```

#### 6) 嵌套查询

本程序可以为管理员提供查询宿舍学生信息的功能，如下图所示：

查询住在Axx宿舍的所有学生的信息：

代码见下：

```

def search_2(self):
    dorm = self.textEdit_3.toPlainText()
    print(dorm)
    if not dorm:
        QMessageBox.warning(self, QWidget: '警告', p_str: '请输入宿舍名')
    else:
        db = pymysql.connect(host='localhost', user='root', passwd='mysql', database='stu_sys')
        cursor = db.cursor()
        sql = ('select sno, sname, dename, clno, dono from student natural join '
              'department where dono in (select dono from dorm where noname=%s)')
        cnt = cursor.execute(sql, dorm)
        result = cursor.fetchall()
        head = ['学号', '姓名', '所在系名', '班号', '寝室号']
        if not result:
            QMessageBox.warning(self, QWidget: '警告', p_str: '查无此舍')
            return
        print(result)
        self.table_1 = Table(head, result)
        self.table_1.show()

```

#### 7) 分组查询

本程序可以为管理员提供如下功能，如下图所示：

查询平均分大于x分的课程：

代码见下：

```
def search_3(self):
    score = self.textEdit_2.toPlainText()
    print(score)
    if not score:
        QMessageBox.warning(self, QWidget: '警告', p_str: '请输入阈值')
    elif int(score) > 100 or int(score) < 0:
        QMessageBox.warning(self, QWidget: '警告', p_str: '请输入正确的阈值')
    else:
        db = pymysql.connect(host='localhost', user='root', passwd='mysql', database='stu_sys')
        cursor = db.cursor()
        sql = 'select cname, avg(gno) from course natural join grade group by cno having avg(gno)>%s'
        cnt = cursor.execute(sql, score)
        result = cursor.fetchall()
        head = ['课程名', '平均分']
        self.table_2 = Table(head, result)
        self.table_2.show()
        print(result)
```

#### 四、实验心得

本次实验我完整地设计了一个数据库，从概念数据库地设计，到逻辑数据库的转化，再到具体关系数据库的实现，帮助我对数据库有了更深层次的理解。在实验过程中，我熟练掌握了 ER 图的设计方法以及 ER 图的转化，对于 Sql 语句的运用更加熟练，包括创建数据库、创建表、创建视图、索引以及在表上的插入、更新、查询删除等操作，最后还学会了 pyqt5 的可视化界面设计，可谓是受益良多。