

---

# 《数据库技术》

## 大作业

题目：对分易管理系统

姓名：章静怡

完成日期：2022 年 12 月 30 日

---

# 目 录

1	需求分析 .....	3
1.1	背景 .....	3
1.2	功能需求 .....	3
1.3	数据需求 .....	4
2	概念模型 .....	6
3	逻辑模型 .....	6
4	建库建表代码 .....	9

---

# 1 需求分析

## 1.1 背景

随着互联网技术的不断发展，越来越多的学习者开始使用教学辅助系统进行学习。虽然传统教学辅助平台已取得一定的效果，但是没有解决课堂教学中形式单一、师生互动欠缺等问题。另一方面，随着智能终端设备的普及，人们更加习惯使用手机来获取信息。同时，因为微信公众号的即时性、便捷性等优点，所以基于微信公众号的教辅系统不仅可以让老师更加了解课堂教学效果，及时发现课堂教学中的问题，还可以增进学生和教师之间的互动[1]。为了进一步提高课堂效率与活跃度，方便老师及时了解学生反馈的学习情况，满足数据存储和信息管理、处理等需求，本项目根据对分易管理系统的学生活动（包括出勤、课堂回答问题、课堂在线练习）和学生质量管理（包括成绩统计、专业课程安排）两个功能的相关需求设计了对分易管理系统数据库，对分易管理系统数据库有利于学生信息的存储、管理和利用，有助于教育行业统计分析各种数据，研究更加和谐有效的、师生共进的教育策略，也有助于探究课堂教育的新方式以及发展方向。

## 1.2 功能需求

对分易管理系统主要是方便老师在对分易平台上对学生数据进行观测、管理，包括存储、统计和查询；学生在对分易平台里及时了解自己的学习状况数据，包括查询。

老师需要了解对于不同学院不同专业的学生而言自己所教授课程的所属类型，学生课堂的出席情况、课堂回答次数、作业成绩、在线练习成绩、期末成绩、该门课程需制定的考核标准等。

学生则需要了解需要学习的专业课程，需要学习的课程的所属类型，课程所对应的考核标准，以及自己在专业课程的出席情况、课堂回答次数、作业成绩、在线练习成绩、期末成绩、排名等。

每个学院有不同专业和若干老师；每个专业有若干学生；每个老师可以教授一门课程或者多门课程；每门课程可能被多名学生选修；每个学生可能选修一门课程或多门课程；每门课程对于不同专业的学生而言可能属于不同类型的课程；每门课程对于不同专业的学生而言有不同的考核类型。

### 1.3 数据需求

本节为数据需求分析阶段，重点描述业务需要采集/管理的数据项及其数据依赖关系。为保证需求表达完整性，部分表在需求阶段允许出现冗余数据项（如名称字段），并在备注中标注“冗余、需在逻辑设计中优化”。后续逻辑结构设计阶段将基于范式原则对冗余字段进行拆分与消除，并补充主外键约束。

表 1 学院与专业信息表

数据依赖		专业号->专业名, 学院号, 学院名 学院号->学院名						
数据项名	数据类型	数据长度	小数位数	单位	约束	允许为空	备注	
专业号	字符串	10			唯一	否	业务主键	
专业名	字符串	20				否		
学院号	字符串	10				否		
学院名	字符串	20				否	冗余数据, 需在逻辑设计中优化	

表 2 学生信息表

数据依赖		学号->姓名, 性别, 班级, 专业号, 手机号						
数据项名	数据类型	数据长度	小数位数	单位	约束	允许为空	备注	
学号	字符串	15			唯一	否		
姓名	字符串	20				否		
性别	字符串	2				是		
班级	字符串	10				否		
专业号	字符串	10				否	所属专业	
手机号	字符串	11				是		

表 3 老师信息表

数据依赖		工号->姓名, 性别, 学院号, 手机号, 职称						
数据项名	数据类型	数据长度	小数位数	单位	约束	允许为空	备注	
工号	字符串	15			唯一	否		
姓名	字符串	20				否		
性别	字符串	2			男/女	是		
学院号	字符串	10				否	所属学院	
手机号	字符串	11				是		
职称	字符串	20				否		

表 4 课程信息表

数据依赖		课程号->课程名, 课时, 学分, 开课学期						
数据项名	数据类型	数据长度	小数位数	单位	约束	允许为空	备注	
课程号	字符串	10			唯一	否		
课程名	字符串	20				否		

课时	整数	2				否	
学分	浮点数	5	2			否	
开课学期	字符串	2				否	

表 5 课程类型信息表

<b>数据依赖</b>		专业号, 课程号->课程类型号, 课程类型名, 考核类型号, 考核类型名 课程类型号->课程类型名 考核类型号->考核类型名					
数据项名	数据类型	数据长度	小数位数	单位	约束	允许为空	备注
专业号	字符串	10				否	
课程号	字符串	10				否	
课程类型号	字符串	10				否	冗余数据, 将在逻辑设计中拆分
课程类型名	字符串	20				否	
考核类型号	字符串	10				否	冗余数据, 将在逻辑设计中拆分
考核类型名	字符串	20				否	

表 6 选课信息表

<b>数据依赖</b>		学号, 课程号->工号, 期末成绩, 在线练习成绩, 作业成绩, 出席情况, 课堂参与次数, 排名, 课程类型号, 考核类型号 期末成绩->排名					
数据项名	数据类型	数据长度	小数位数	单位	约束	允许为空	备注
学号	字符串	15			唯一	否	参照表 2 主码, 与课程号构成联合唯一
课程号	字符串	10				否	参照表 4 主码
工号	字符串	15				否	参照表 3 主码
期末成绩	浮点数	5	2			是	
作业成绩	浮点数	5	2			是	
出席情况	整数	2				是	
在线练习成绩	浮点数	5	2			是	
课堂回答次数	整数	2				是	
课程类型号	字符串	10				否	冗余需求项, 逻辑设计阶段通过配置表关联
考核类型号	字符串	10				否	冗余需求项, 逻辑设计阶段通过配置表关联
排名	字符串	10				是	派生属性, 由上述属性统计出来

## 2 概念模型

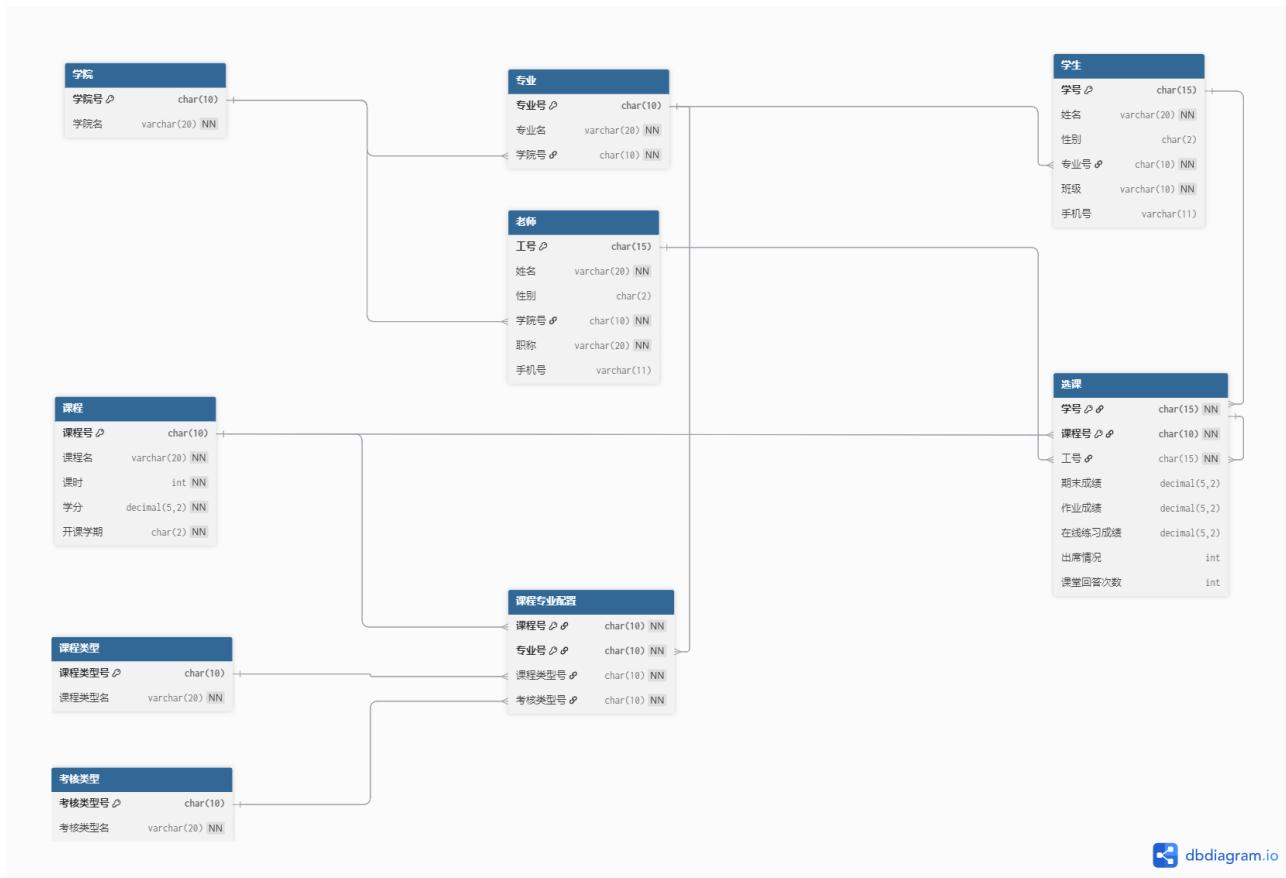


图 1 对分易管理系统 E-R 图

## 3 逻辑模型

根据数据库第三范式 (3NF) 要求，对 1.3 中的数据需求进行了以下优化：

- 将原“表 1”拆分为“学院表”和“专业表”，消除学院数据冗余。
- 将原“表 5”拆分为“课程类型字典”、“考核类型字典”和“课程专业配置表”，解耦类型定义与配置关系。
- “选课表”中移除了“排名”（改为视图计算）和“类型号”（通过配置表关联），并允许成绩字段为空。

### 1. 专业

属性名	说明	类型	长度	精度	默认值	允许空值	码
专业号		char	10			否	主码

专业名		varchar	20			否	
学院号		char	10			否	外码(学院. 学院号)

数据依赖: F = { 专业号 → 专业名, 专业号 → 学院号 }

## 2. 学院

属性名	说明	类型	长度	精度	默认值	允许空值	码
学院号		char	10			否	主码
学院名		varchar	20			否	

数据依赖: F={学院号→学院名}

## 3. 老师

属性名	说明	类型	长度	精度	默认值	允许空值	码
工号		char	15			否	主码
姓名		varchar	20			否	
性别		char	2			是	
学院号		char	10			否	外码(学院. 学院号)
职称		varchar	20			否	
手机号		varchar	11			是	

数据依赖: F={工号→姓名, 工号→性别, 工号→学院号, 工号→职称, 工号→手机号}

## 4. 学生

属性名	说明	类型	长度	精度	默认值	允许空值	码
学号		char	15			否	主码
姓名		varchar	20			否	
性别		char	2			是	
专业号		char	10			否	外码(专业. 专业号)
班级		varchar	10			否	
手机号		varchar	11			是	

数据依赖: F={学号→姓名, 学号→性别, 学号→专业号, 学号→班级, 学号→手机号}

## 5. 课程

属性名	说明	类型	长度	精度	默认值	允许空值	码
课程号		char	10			否	主码

课程名		varchar	20			否	
课时		int	2			否	
学分		float	5			否	
开课学期		char	2			否	

数据依赖: F={课程号→课程名, 课程号→课时, 课程号→学分, 课程号→开课学期}

## 6. 课程类型

属性名	说明	类型	长度	精度	默认值	允许空值	码
课程类型号		char	10			否	主码
课程类型名		varchar	20			否	

数据依赖: F={课程类型号→课程类型名}

## 7. 考核类型

属性名	说明	类型	长度	精度	默认值	允许空值	码
考核类型号		char	10			否	主码
考核类型名		varchar	20			否	

数据依赖: F={考核类型号→考核类型名}

## 8. 课程-专业配置表

属性名	说明	类型	长度	精度	默认值	允许空值	码
课程号		char	10			否	主码(联合) 外码(课程. 课程号)
专业号		char	10			否	主码(联合) 外码(专业. 专业号)
课程类型号		char	10			否	外码(课程类型. 课程类型号)
考核类型号		char	10			否	外码(考核类型. 考核类型号)

数据依赖: F = { (课程号, 专业号) → 课程类型号, (课程号, 专业号) → 考核类型号 }

## 9. 选课表

属性名	说明	类型	长度	精度	默认值	允许空值	码
学号		char	15			否	主码(联合) 外码(学生. 学号)
课程号		char	10			否	主码(联合) 外码(课程. 课程号)
工号		char	15			否	外码(老师. 工号)
期末成绩		float	5			是	
作业成绩		float	5			是	
出席情况		int	2			是	
在线练习成绩		float	5			是	
课堂回答次数		int	2			是	

数据依赖:  $F = \{(学号, 课程号) \rightarrow 工号, (学号, 课程号) \rightarrow 期末成绩, (学号, 课程号) \rightarrow 作业成绩, (学号, 课程号) \rightarrow 在线练习成绩, (学号, 课程号) \rightarrow 出席情况, (学号, 课程号) \rightarrow 课堂回答次数\}$

## 4 建库建表代码

```
/* ===== */
/* 第一部分: 建库与建表 (DDL) */
/* ===== */
```

```
DROP DATABASE IF EXISTS dfy_db;
CREATE DATABASE dfy_db DEFAULT CHARSET = utf8mb4;
USE dfy_db;
```

-- 1) 学院

```
CREATE TABLE `学院` (
  `学院号` CHAR(10) PRIMARY KEY,
  `学院名` VARCHAR(20) NOT NULL
) ENGINE=InnoDB;
```

-- 2) 专业

```
CREATE TABLE `专业` (
  `专业号` CHAR(10) PRIMARY KEY,
  `专业名` VARCHAR(20) NOT NULL,
```

```
-- 学院号` CHAR(10) NOT NULL,  
CONSTRAINT `fk_专业_学院`  
FOREIGN KEY (`学院号`) REFERENCES `学院`(`学院号`)  
ON UPDATE CASCADE ON DELETE RESTRICT  
) ENGINE=InnoDB;
```

-- 3) 老师

```
CREATE TABLE `老师` (  
`工号` CHAR(15) PRIMARY KEY,  
`姓名` VARCHAR(20) NOT NULL,  
`性别` CHAR(2),  
`学院号` CHAR(10) NOT NULL,  
`职称` VARCHAR(20) NOT NULL,  
`手机号` VARCHAR(11),  
CONSTRAINT `fk_老师_学院`  
FOREIGN KEY (`学院号`) REFERENCES `学院`(`学院号`)  
ON UPDATE CASCADE ON DELETE RESTRICT  
) ENGINE=InnoDB;
```

-- 4) 学生

```
CREATE TABLE `学生` (  
`学号` CHAR(15) PRIMARY KEY,  
`姓名` VARCHAR(20) NOT NULL,  
`性别` CHAR(2),  
`专业号` CHAR(10) NOT NULL,  
`班级` VARCHAR(10) NOT NULL,  
`手机号` VARCHAR(11),  
CONSTRAINT `fk_学生_专业`  
FOREIGN KEY (`专业号`) REFERENCES `专业`(`专业号`)  
ON UPDATE CASCADE ON DELETE RESTRICT  
) ENGINE=InnoDB;
```

-- 5) 课程

```
CREATE TABLE `课程` (  
`课程号` CHAR(10) PRIMARY KEY,  
`课程名` VARCHAR(20) NOT NULL,  
`课时` INT NOT NULL,  
`学分` DECIMAL(5, 2) NOT NULL,  
`开课学期` CHAR(2) NOT NULL  
) ENGINE=InnoDB;
```

-- 6) 课程类型

```
CREATE TABLE `课程类型` (  
`课程类型号` CHAR(10) PRIMARY KEY,  
`课程类型名` VARCHAR(20) NOT NULL
```

---

```
) ENGINE=InnoDB;
```

```
-- 7) 考核类型
```

```
CREATE TABLE `考核类型` (
    `考核类型号` CHAR(10) PRIMARY KEY,
    `考核类型名` VARCHAR(20) NOT NULL
) ENGINE=InnoDB;
```

```
-- 8) 课程-专业配置
```

```
CREATE TABLE `课程专业配置` (
    `课程号` CHAR(10) NOT NULL,
    `专业号` CHAR(10) NOT NULL,
    `课程类型号` CHAR(10) NOT NULL,
    `考核类型号` CHAR(10) NOT NULL,
    PRIMARY KEY (`课程号`, `专业号`),
    CONSTRAINT `fk_配置_课程`
        FOREIGN KEY (`课程号`) REFERENCES `课程`(`课程号`) ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT `fk_配置_专业`
        FOREIGN KEY (`专业号`) REFERENCES `专业`(`专业号`) ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT `fk_配置_课程类型`
        FOREIGN KEY (`课程类型号`) REFERENCES `课程类型`(`课程类型号`) ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT `fk_配置_考核类型`
        FOREIGN KEY (`考核类型号`) REFERENCES `考核类型`(`考核类型号`) ON UPDATE CASCADE ON DELETE RESTRICT
) ENGINE=InnoDB;
```

```
-- 9) 选课 (Enrollment)
```

```
CREATE TABLE `选课` (
    `学号` CHAR(15) NOT NULL,
    `课程号` CHAR(10) NOT NULL,
    `工号` CHAR(15) NOT NULL,
    `期末成绩` DECIMAL(5, 2) DEFAULT NULL,
    `作业成绩` DECIMAL(5, 2) DEFAULT NULL,
    `在线练习成绩` DECIMAL(5, 2) DEFAULT NULL,
    `出席情况` INT DEFAULT NULL,
    `课堂回答次数` INT DEFAULT NULL,
    PRIMARY KEY (`学号`, `课程号`),
    CONSTRAINT `fk_选课_学生`
        FOREIGN KEY (`学号`) REFERENCES `学生`(`学号`) ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT `fk_选课_课程`
        FOREIGN KEY (`课程号`) REFERENCES `课程`(`课程号`) ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT `fk_选课_老师`
        FOREIGN KEY (`工号`) REFERENCES `老师`(`工号`) ON UPDATE CASCADE ON DELETE RESTRICT
) ENGINE=InnoDB;
```

---

```
) ENGINE=InnoDB;
```

```
/* ===== */
/* 第二部分：模拟数据插入（DML） */
/* ===== */
```

```
-- 1. 插入学院
```

```
INSERT INTO `学院` VALUES
('XY01', '计算机学院'),
('XY02', '数理学院');
```

```
-- 2. 插入专业
```

```
INSERT INTO `专业` VALUES
('ZY01', '软件工程', 'XY01'),
('ZY02', '数据科学', 'XY01');
```

```
-- 3. 插入老师
```

```
INSERT INTO `老师` VALUES
('T001', '张教授', '男', 'XY01', '教授', '13800000001'),
('T002', '李讲师', '女', 'XY01', '讲师', '13800000002'),
('T003', '王教授', '男', 'XY02', '教授', '13800000003');
```

```
-- 4. 插入学生
```

```
INSERT INTO `学生` VALUES
('S2023001', '小明', '男', 'ZY01', '2301 班', '13900000001'),
('S2023002', '小红', '女', 'ZY01', '2301 班', '13900000002'),
('S2023003', '小刚', '男', 'ZY02', '2302 班', '13900000003');
```

```
-- 5. 插入课程
```

```
INSERT INTO `课程` VALUES
('KC01', '数据库原理', 64, 4.00, '3'),
('KC02', 'Java 程序设计', 48, 3.00, '3'),
('KC03', '高等数学', 80, 5.00, '1');
```

```
-- 6. 插入课程类型
```

```
INSERT INTO `课程类型` VALUES
('LX01', '专业必修'),
('LX02', '通识选修');
```

```
-- 7. 插入考核类型
```

```
INSERT INTO `考核类型` VALUES
('KH01', '闭卷考试'),
('KH02', '大作业');
```

---

-- 8. 插入配置

```
INSERT INTO `课程专业配置` VALUES  
('KC01', 'ZY01', 'LX01', 'KH01'),  
('KC02', 'ZY01', 'LX01', 'KH02');
```

-- 9. 插入选课记录

```
INSERT INTO `选课`  
(`学号`, `课程号`, `工号`, `期末成绩`, `作业成绩`, `在线练习成绩`, `出席情况`, `课堂回答次数`)  
VALUES ('S2023001', 'KC01', 'T001', 85.50, 90.00, 95.00, 10, 5);
```

```
INSERT INTO `选课`(`学号`, `课程号`, `工号`)  
VALUES ('S2023002', 'KC01', 'T001');
```

```
INSERT INTO `选课`  
(`学号`, `课程号`, `工号`, `期末成绩`, `作业成绩`, `在线练习成绩`, `出席情况`, `课堂回答次数`)  
VALUES ('S2023001', 'KC02', 'T002', 78.00, 88.00, 80.00, 8, 2);
```