

《数据库原理》大作业报告		
题目： 学生学籍管理系统 设计	小组成员学号姓名	日期
	2021300119 周翔宇 2021380108 LEONARDO 2021302492 徐旭阳	2023/11/25

一、需求分析

1、功能概述

一个简单的学籍管理系统，系统功能分为面向学生和面向管理员两个部分。

面向学生，学生仅有查阅自己课程对应成绩信息的权限，以及对自己的学籍信息进行修改的权限；

面向管理员，可以查询、增加、删除、修改所有学生的相关学籍与成绩信息；按照班级、科目查阅对应学生的统计信息。

学生功能：

修改：及时纠正学籍数据录入错误，如性别、户口、政治面貌等信息。

查询：获取详细的学生成绩信息

管理员功能：

学生信息增加/修改：增加所有学生的学籍信息，修改所有学生的学籍信息。

学生成绩录入/修改：设定学号和课程号后，录入/修改学生该课程的考试分数。

统计表查看：统计某个班级某学年某学期的某门课程的考试情况并排名；

按班级/学科查看成绩信息：按照学生所在班级或学生所选科目，查看学生的成绩信息。

2、开发和运行环境

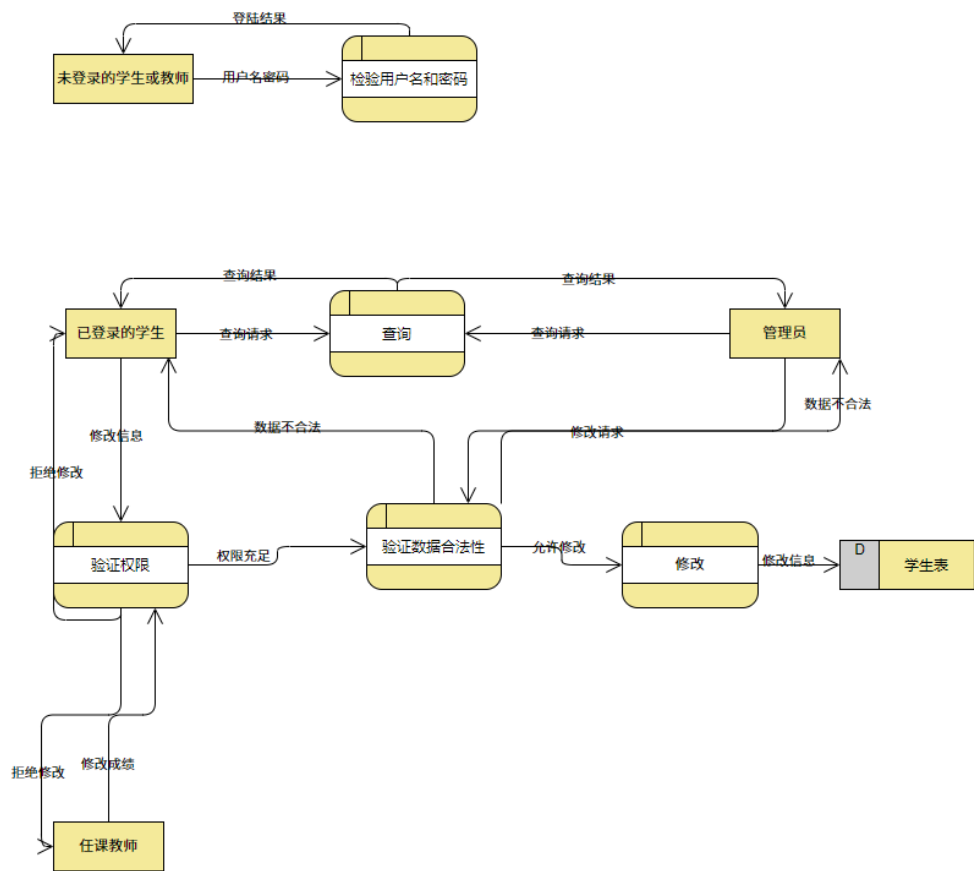
开发环境：

前端 vue.js 后端 express.js 数据库版本 mysql 8.0

运行环境：

Windows 环境、node.js、MySQL8.0

3、数据流图



二、 数据字典

数据项：

1、 学生表：

数据项名	说明	数据类型	取值范围	约束	备注
sno	学生的唯一标识	char	8 位	主键	学号
sname	学生的姓名	char	10 位	NOT NULL	姓名
sgender	学生的性别	char	2 位	NOT NULL	性别
syear	学生所在年级	year	YYYY	NOT NULL	年级
sbirth	学生的出生日期	date	YYYY-MM-DD	NULL	出生日期
spolitic	标识学生的政治面貌	char	10 位	NULL	政治面貌
sethnic	标识学生的名族	char	10 位	NULL	民族
sclno	记录学生的班级	char	8 位	NOT NULL	班级号

2、 班级表

数据名	说明	数据类型	取值范围	约束	备注
-----	----	------	------	----	----

clno	班级的唯一标识	char	8 位	主键	班级号
clyear	班级的注册日期	year	YYYY	NOT NULL	入学日期
clmno	班级专业的专业号	char	8 位	NOT NULL	专业号

3、专业表

数据名	说明	数据类型	取值范围	约束	备注
mno	专业的唯一标识	char	8 位	主键	专业号
mname	专业的名称	char	20 位	NOT NULL	专业名
mfno	专业开设学院号	char	8 位	NOT NULL	学院号

4、学院表

数据名	说明	数据类型	取值范围	约束	备注
fno	学院的唯一标识	char	8 位	主键	学院号
fname	学院名称	char	20 位	NOT NULL	学院名

5、课程表

数据名	说明	数据类型	取值范围	约束	备注
cno	课程的唯一标识	char	8 位	主键	课程号
cname	课程名称	char	20 位	NOT NULL	课程名
ccredit	修读课程可获得的学分	int		NOT NULL	学分
ctype	课程类型	char	10 位	NOT NULL	课程类型

6、成绩表

数据名	说明	数据类型	取值范围	约束	备注
sno	学生的学号	char	8 位	主键	学号
cno	学生所选课程的课程号	char	8 位	主键	课程号
score	学生所选课程获得的成绩	int	32 位	NOT NULL	成绩

7、登陆信息表

数据名	说明	数据类型	取值范围	约束	备注
username	登陆用的用户名	char	20位	主键 NOT NULL	用户名
password	用户名对应密码	char	64位	NOT NULL	密码

数据结构

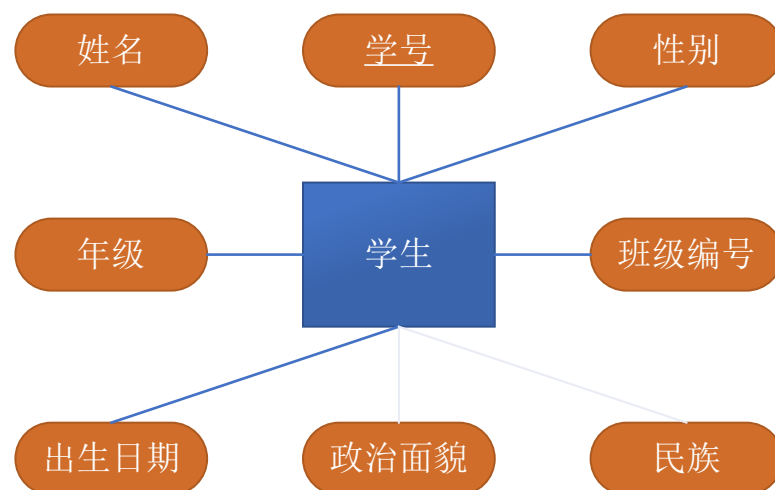
数据结构名	含义	组成成分
学生表	记录学生的学籍信息	学号，姓名，性别，民族，年级，出生日期，政治面貌，班级号
班级表	记录当前存在的班级	班号，注册日期，对应的专业
专业表	记录学校存在的专业	专业号，专业名，专业开设的学院号
学院表	记录学校存在的学院	学院号，学院名
课程表	记录全校开课情况	课程号，课程名，对应学分，课程类型
成绩表	记录学生修读的课程获得的成绩	学号，课程号，成绩
登录信息表	记录注册到系统中的用户名密码	用户名，密码

三、 概念结构设计

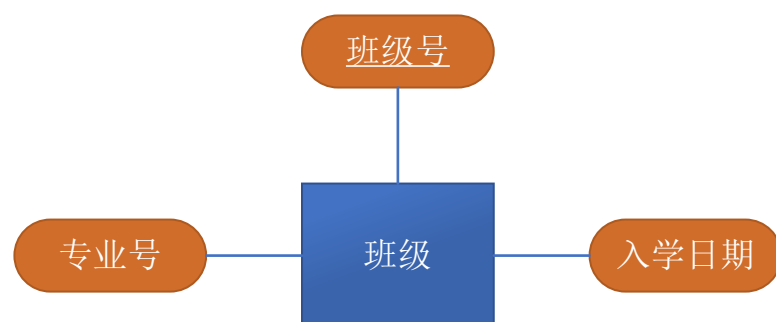
1、ER 图

● 实体：

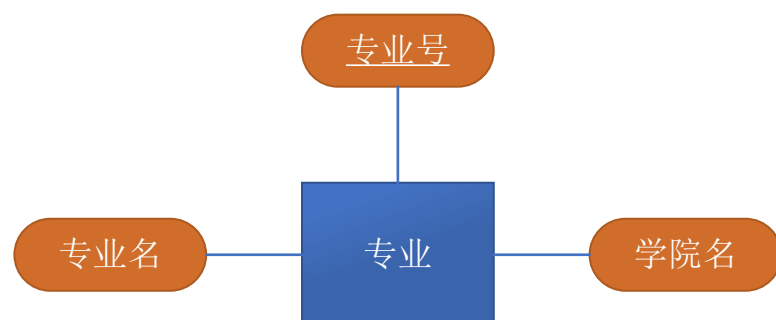
◇ 学生



◇ 班级



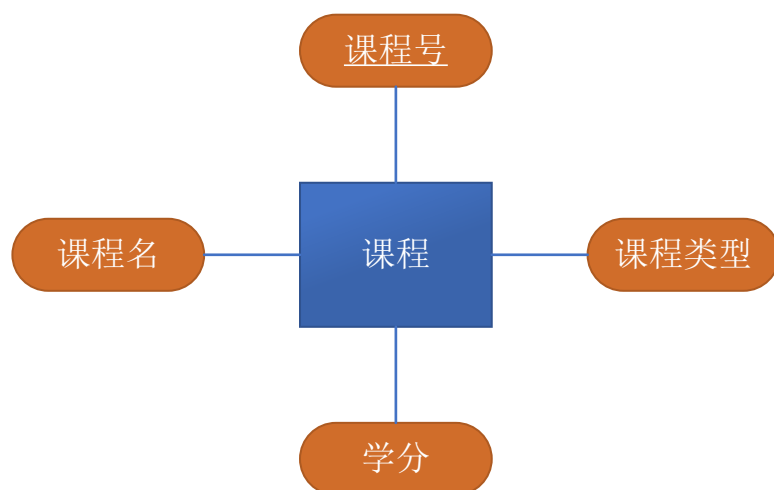
◇ 专业



◇ 学院



◇ 课程



联系:

◇ 学院与专业的联系为 1:n 型



◇ 专业与班级的联系为 1:n 型



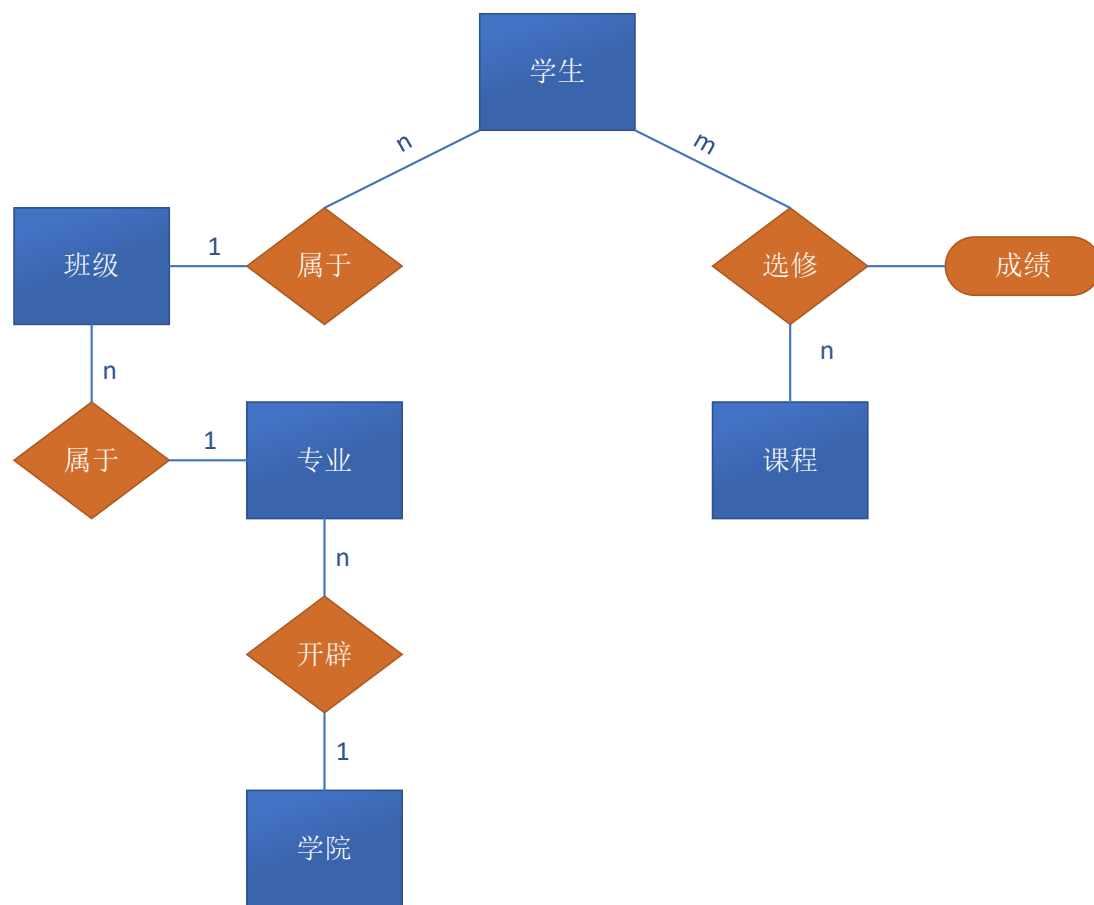
◇ 班级与学生的联系为 1:n 型



◇ 课程与选修的联系为 n:m 型



整体 E-R 图：



四、 逻辑设计

1、关系模式

学生(学号, 姓名, 性别, 年级, 出生日期, 政治面貌, 民族, 班级号)；

班级(班级号, 入学日期, 专业号)

专业(专业号, 专业名, 学院号)

学院(学院号, 学院名)

课程(课程号, 课程名, 学分, 课程类型)

成绩表(学号, 课程号, 成绩)

2、函数依赖：

在学生关系模式中：

学号→姓名, 学号→性别, 学号→年级, 学号→出生日期, 学号→政治面貌, 学号→民族, 学号→班级号

在班级关系模式中：

班级号→入学日期, 班级号→专业号

在专业关系模式中：

专业号→专业名，专业号→学院号

在学院关系模式总：

学院号→学院名

在课程关系模式中：

课程号→课程名，课程号→学分，课程号→课程类型

成绩表关系模式中：

(学号，课程号)→成绩

关系模式分析：以学生为例分析，学生(学号，姓名，性别，年级，出生日期，政治面貌，民族，班级号)该关系的码是学号。所有属性都依赖于主码，因此不存在非主属性和主属性对码的部分函数依赖。所有的非主属性都只依赖于主码，因此不存在非主属性对码的传递函数依赖。每一个函数依赖的决定因素都包含码，因此满足 BCNF。同理，其他关系均满足 BCNF。

3、完整性约束

实体完整性：下划线为主键

参照完整性：

学生表的班级号是外键，参照班级表的班级号

班级表的专业号是外键，参照专业表的专业号

专业表的学院号是外键，参照学院表的学院号

用户自定义完整性（check 约束）：

在学生表中，性别的取值为男和女，年级和出生日期不允许未来的日期，政治面貌和民族也限定在特定的取值范围。

在班级表中，入学日期不允许未来的日期。

在课程号中，学分的取值范围为[0, 6]。

在成基表中，成绩的取值范围为[0, 100]。

4、触发器

插入/更新学生信息的触发器，用于检查用户定义的完整性。

插入触发器的实现如下：


```

DELIMITER //
CREATE TRIGGER before_insert_student_check
BEFORE INSERT ON student
FOR EACH ROW
BEGIN
    IF NEW.syear > YEAR(CURDATE()) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = '年级不能是未来的日期';
    END IF;

    IF NEW.sbirth > CURDATE() THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = '出生日期不能是未来的日期';
    END IF;
END;
//
DELIMITER ;

```

插入/更新班级信息的触发器，用于检查用户定义的完整性。

插入触发器的实现如下：

```

DELIMITER //
CREATE TRIGGER before_insert_class_check
BEFORE INSERT ON class
FOR EACH ROW
BEGIN
    IF NEW.clyear > YEAR(CURDATE()) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = '入学日期不能是未来的日期';
    END IF;
END;
//
DELIMITER ;

```

5、存储过程

检查用户登录的有效性的存储过程

存储过程的实现如下：

```

DELIMITER //
CREATE PROCEDURE CheckLogin(IN p_username CHAR(20), IN p_password CHAR(64))
BEGIN
    IF EXISTS (SELECT * FROM login WHERE username = p_username AND password = p_password) THEN
        SELECT 1 AS result;
    ELSE
        SELECT 0 AS result;
    END IF;
END //
DELIMITER ;

```

6、视图设计

```
CREATE VIEW student_course_grade AS
SELECT s.sno AS sno, s.sname AS sname, c.cno AS cno, c.cname AS cname, s.sclno AS clno, c.ccredit AS ccredit, g.score AS score
FROM grade g
INNER JOIN student s ON g.sno = s.sno
INNER JOIN course c ON g.cno = c.cno;
```

五、物理设计

索引设计：从需求分析中得知，不存在根据非主属性进行查询的操作。由于 Mysql 会自动为所有表的主键进行索引，并且 MySQL 4.1.2 以后的版本在建立外键时会自动创建索引，所以物理结构设计中没有显式创建索引。

六、遇到的问题及解决办法

- 1. Node.js 的 mysql 模块在和 mysql 8.0 服务器通信时提示身份验证方式不支持，解决方法是修改管理员密码，使用 mysql_native_password 验证方式，或者换用更新的 mysql 模块。
- 2. 项目采用前后端分离架构，前端 Vite 调试服务器和后端 API 服务器监听的端口不同，导致浏览器访问后端 API 时因跨域导致请求失败。解决方法是在 Vue 项目中配置代理，浏览器向 Vite 调试服务器请求 API，调试服务器转而后端服务器通信。
- 3. 标准登录流程是使用 cookie+session 或者 token 机制，由于登录流程的编写不是本次大作业的重点，因此后端简单采用用户名密码验证，每次请求 API 都需要带用户名和密码。

七、部分设计结果展示

学籍管理	学号	姓名	性别	年级	班别	出生日期	政治面貌	民族	操作
学生统计	40000001	张三	男	2023	30000001	2006-1-1	党员	汉族	保存删除
	40000002	李四	女	2023	30000002	2004-2-2	团员	满族	保存删除
成绩管理	40000003	王五	男	2022	30000003	2003-3-3	群众	回族	保存删除
学期成绩统计	40000004	赵六	女	2022	30000004	2002-4-4	无党派人士	其他	保存删除
	40000005	小红	男	2021	30000005	2001-5-5	党员	汉族	保存删除
分段成绩统计	40000006	小明	女	2021	30000006	2000-6-6	团员	满族	保存删除
	40000007	小李	男	2020	30000007	1999-7-7	群众	回族	保存删除
学生成绩统计	40000008	小刘	女	2020	30000008	1998-8-8	无党派人士	其他	保存删除
	40000009	丽丽	男	2019	30000009	1997-9-9	党员	汉族	保存删除
退出	40000010	小华	女	2019	30000010	1996-10-10	团员	满族	保存删除
	40000011	小强	男	2018	30000011	1995-11-11	群众	回族	保存删除
	40000012	世世	女	2018	30000012	1994-12-12	无党派人士	其他	保存删除

学籍管理	学号	姓名	课程号	课程名称	学分	成绩	操作
学生统计	40000001	张三	50000001	高等数学	4	85	保存删除
	40000002	李四	50000002	英语语法与写作	3	72	保存删除
成绩管理	40000003	王五	50000003	计算机编程基础	3	94	保存删除
学期成绩统计	40000004	赵六	50000004	线性代数	4	68	保存删除
	40000005	小红	50000005	数据库原理	3	90	保存删除
分段成绩统计	40000006	小明	50000006	数据结构与算法	4	78	保存删除
	40000007	小李	50000007	离散数学	3	89	保存删除
学生成绩统计	40000008	小刘	50000008	大学物理	4	75	保存删除
	40000009	丽丽	50000009	电子科学与技术	3	92	保存删除
退出	40000010	小华	50000010	计算机网络	4	86	保存删除
	40000011	小强	50000011	软件工程	4	77	保存删除
	40000012	世世	50000012	人工智能基础	3	95	保存删除
	40000013	大明	50000013	操作系统原理	4	80	保存删除