|  |  |  |
| --- | --- | --- |
|  |  | |
| 数据库系统课程设计报告 | | |
| 小组名称：小组成员：汤嘉熙（组长）、付耀刚、周欣潼撰写时间：2024年12月17日文档版本：V1.0 | |  |

目录页

（该页面需要自动生成，生成后删除括号内文字）

目录

[数据库系统课程设计报告 1](#_Toc10379)

**[一、成员分工（观测点9.2） 3](#_Toc22183)**

**[二、需求分析（观测点2.4） 4](#_Toc8190)**

[1. 数据字典抽取： 4](#_Toc23591)

[5. 非功能性分析： 5](#_Toc25998)

**[三、系统概念结构设计（观测点2.4） 6](#_Toc15455)**

[1. 局部E-R图： 6](#_Toc6349)

**[四、系统的逻辑结构设计（观测点3.4） 10](#_Toc29462)**

[1. 关系模型：加粗为主键，下划线为外键，红色为实体间的关系 10](#_Toc7920)

[2. 范式级别： 10](#_Toc3826)

**[五、系统物理结构设计、持久层设计和关键业务代码（观测点3.4） 10](#_Toc14636)**

[1. 数据库管理系统基本信息： 11](#_Toc3314)

[2. DDL语句： 11](#_Toc5810)

[3. 物理结构设计： 16](#_Toc13546)

**[六、项目运维管理和优化（观测点11.1） 21](#_Toc1995)**

**[七、附件内容（观测点11.1） 23](#_Toc9913)**

一、成员分工（观测点9.2）

成员1：汤嘉熙（221002211）

职位：组长

任务分工：

完成xxxxxx

成员2：付耀刚（学号）

职位：组员

任务分工：

完成xxxxxx

成员3：周欣潼（学号）

职位：组员

任务分工：

完成xxxxxx

任务分工撰写要求：

* 详细描述个人任务，与其他人协作情况。
* 要分条清晰列举。
* 反面例子：负责\*\*\*业务的设计工作-设计内容含糊，没有标明具体哪个业务的哪块设计工作，也没有提及和其他人协作情况；负责ppt的设计和测试工作-没有完成数据库课程设计的培养目标，组内每个人都要有有效数据库设计任务。
* 小四，宋体，1.5倍行距。

二、需求分析（观测点2.4）

撰写要求：数据字典抽取 √

* 使用用例图描述各个业务。
* 将用例图转换成鲁棒图。
* 分析鲁棒图列举涉及的数据项。
* 数据项的格式为：数据项名称，数据项粗略类型（字符型、数字型、日期时间型，其他型）、数据项的简要描述。
* 描述与数据相关的非功能性分析，如安全性分析、完整性约束等。 √
* 用例图和鲁棒图需要有文字配套描述。
* 小四，宋体，1.5倍行距。

1. 数据字典抽取：

考生信息：考生的编号、姓名、电话、邮箱、生日、本科专业、本科学校、本科类型、生源地、简历、毕业时间、准考证号、身份证号、类别、志愿状态、研究生类型

导师信息：导师的编号、姓名、电话、邮箱、简介、职称、照片

学科信息：学科的编号、名称、类型、概述、等级、指标、上级学科

科目信息：科目的编号、名称、类别

学院信息：学院的编号、名称、官网

研究生管理秘书信息：研究生管理秘书的编号、名称

学科负责人信息：学科负责人的编号、名称

学科秘书信息：学科秘书的编号、名称

研究生主管领导信息：研究生主管领导的编号、名称

复试组信息：复试组的编号、名称

导师教学信息：导师名、学科名、学硕名额、专硕名额、直博名额

学科考核信息：学科名、科目名、考核类别

考生考试信息：考生名、科目名、成绩、评价、时间、地点、考试类别

考生选择导师信息：考生名、导师名、志愿优先级

考生报考学科信息：考生名、学科名、报考志愿顺序

1. 用例图xxxxx。如图1所示。
2. 鲁棒图xxxxxx。如图2所示。
3. 鲁棒图所涉及的数据项：如表1所示。

表1 鲁棒图涉及数据项

|  |  |  |
| --- | --- | --- |
| 名称 | 粗略类型 | 简要描述 |
|  |  |  |
|  |  |  |
|  |  |  |

1. 非功能性分析：

① 安全性分析：

* 用户认证与授权：

1. 每个用户都需要通过身份验证才能访问系统，使用用户名和密码或者更安全的多因素认证。
2. 根据用户的角色（考生、导师、学科秘书等），系统将授权不同的操作权限，例如导师可以查看和选择学生，学科秘书可以管理导师和学生的信息等。

* 数据加密：

1. 对考生的敏感信息（身份证号、电话、邮箱等）进行加密存储，确保即使数据泄露，信息也不会被轻易解读。

* 审计日志：

1. 系统应记录所有关键操作的审计日志，包括用户登录、信息修改、导师选择等，以便于追踪和审计。

* 防止SQL注入：

1. 在实现数据库操作时，使用参数化查询或ORM框架来防止SQL注入攻击，保护系统免受恶意攻击。

② 完整性约束：

* 数据完整性：

1. 在数据库设计中，使用外键来确保数据的引用完整性，例如考生信息与学科信息之间的关联。
2. 实现数据校验机制，如检查输入数据的格式和范围，确保数据的正确性。

* 事务管理：

1. 对于需要多个步骤完成的操作（如导师和学生的选择过程），使用数据库事务来保证操作的原子性。
2. 在出现错误时，能够回滚事务，保持数据的一致性。

③ 可维护性：

* 代码和文档：

1. 编写清晰、结构良好的代码，并提供详细的开发和用户文档。
2. 使用版本控制系统来管理代码变更。

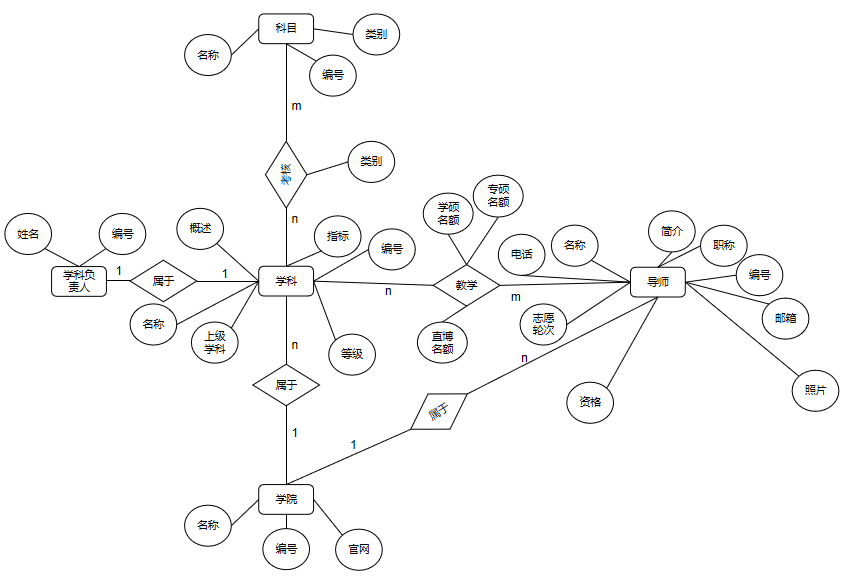
* 模块化设计：

1. 系统功能应模块化设计，如用户管理、成绩管理、导师选择等，以便于单独维护和升级。

三、系统概念结构设计（观测点2.4）

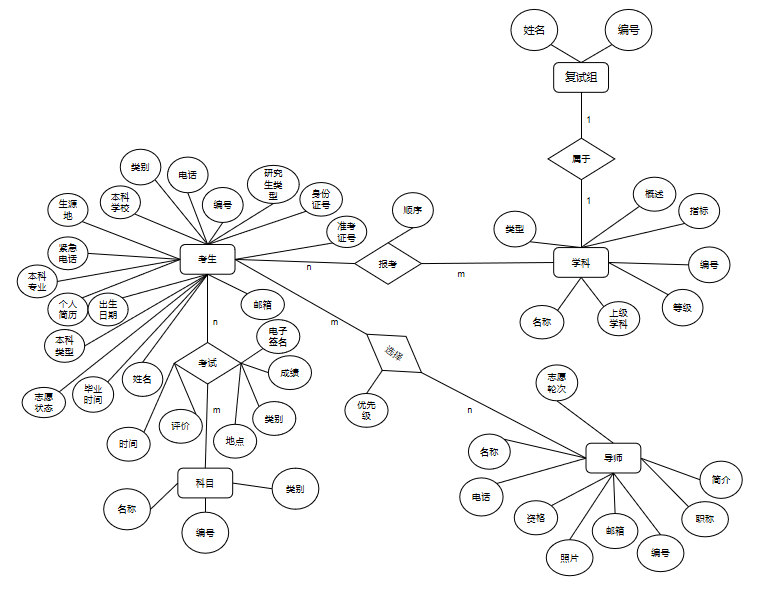
1. 局部E-R图：

① 业务1：招生目录包括不同的学院，每个学院下有不同的科目和导师，一个导师会教学多个学科，一个学科也会有多个导师教学。每个学科需要考核多个科目，每个科目在多个学科中都会被考核到。该业务局部E-R图如图x所示。



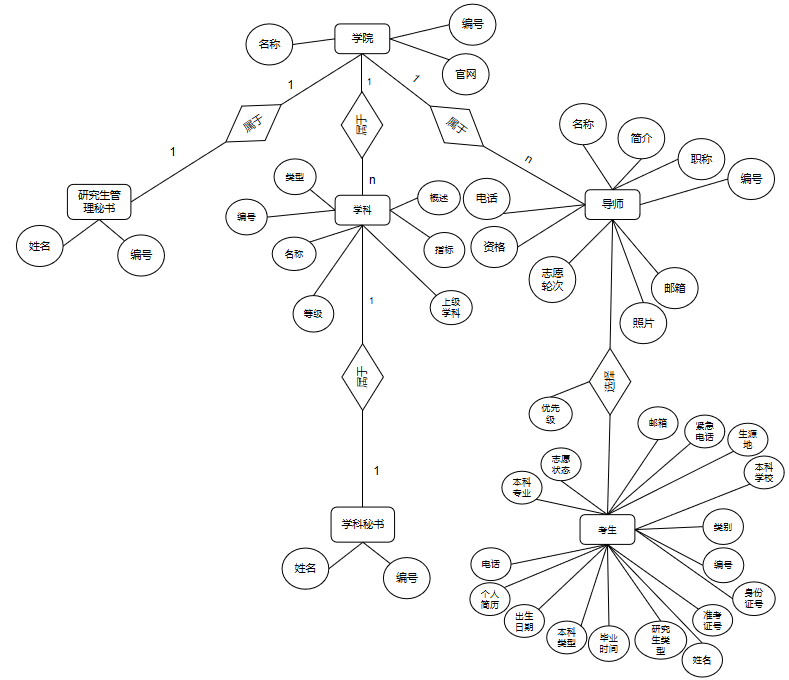
图x 业务1局部E-R图

② 业务2：考生需要填写自己的个人信息，每名考生可以报考多个学科（二级学科），每个学科也有多名考生报考。每名考生需要考试多个科目（有初试科目和复试科目的区别），每个科目也有多名考生考试。每名考生可以选择多个导师（志愿顺序），每个导师也有多名考生选择。该业务局部E-R图如图x所示。



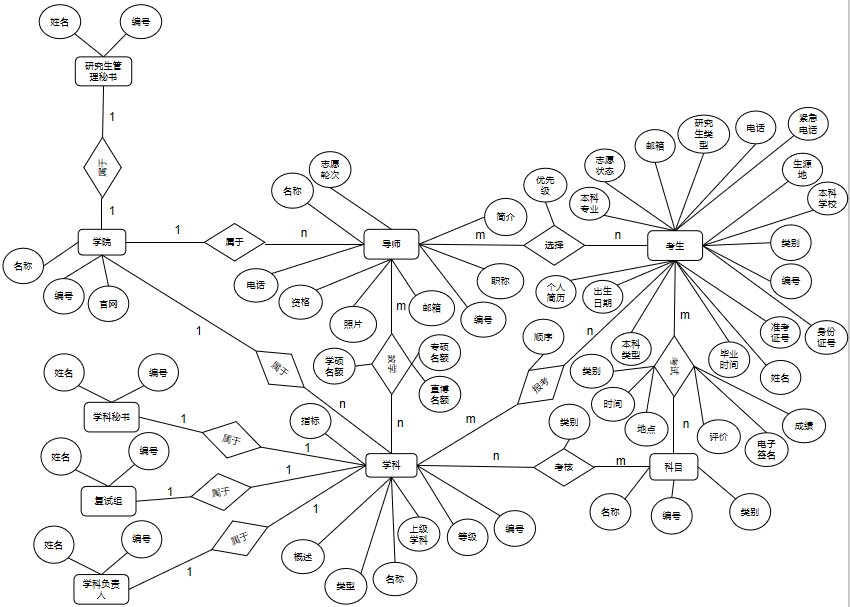
图x 业务2局部E-R图

③ 业务3：每个学院会包含多个研究生管理秘书和多个学科负责人，研究生管理秘书负责管理各学科负责人，而学科负责人负责管理该学科下的导师（分配名额、确定资格等）。研究生管理秘书还负责录入各考试的初复试成绩。该业务局部E-R图如图x所示。



图x 业务3局部E-R图

2. 全局E-R图：每个学院会包含多个研究生管理秘书、多个学科负责人、多个导师、多个学科。研究生管理秘书管理学科负责人，学科负责人管理导师。导师和学科之间有多对多的关系。考生需要考试指定的科目，考试结束后需要填写个人信息，由研究生管理秘书录入，并且根据考生填写的报考学科和选择的导师进行分配。全局E-R图如图x所示。



图x 全局E-R图

四、系统的逻辑结构设计（观测点3.4）

1. 关系模型：**加粗为主键**，下划线为外键，红色为实体间的关系

① 学科负责人（**编号**、姓名、学科编号）

② 导师（**编号**、姓名、电话、资格、图片、邮箱、职称、简介、志愿轮次、学院编号）

③ 选择（**导师编号、考生编号**、优先级）

④ 考生（**编号**、姓名、电话、紧急电话、邮箱、本科专业、本科学校、生源地、本科类型、个人简历、出生日期、毕业时间、准考证号、身份证号、类别、志愿状态、研究生类型）

⑤ 考试（**科目编号、考生编号**、类别、成绩、时间、地点、评价、电子签名）

⑥ 科目（**编号**、名称、类别）

⑦ 考核（**科目编号、学科编号**、类别）

⑧ 报考（**学生编号、学科编号**、顺序）

⑨ 学科（**编号**、名称、类型、指标、概述、上级学科、等级、学院编号）

⑩ 教学（**学科编号、导师编号**、学硕名额、专硕名额、直博名额）

⑪ 学院（**编号**、名称、官网）

⑫ 研究生管理秘书（**编号**、姓名、学院编号）

⑬ 复试组（**编号**、学科编号）

⑭ 学科秘书（**编号**、姓名、学科编号）

⑮ 研究生主管领导（**编号**、姓名）

⑯ 监督人员（**编号**、姓名）

⑰ 管理员（**编号**、姓名）

1. 范式级别：

各关系模型所有属性是原子的，且不存在部分依赖或传递依赖的情况。因此所有关系模式都满足3NF。

五、系统物理结构设计、持久层设计和关键业务代码（观测点3.4）

撰写要求：

* 提供使用的数据库管理系统基本信息，包括软件名称、版本号和其他软硬件环境信息等。 √
* 列出数据库每张表的DDL语句。 √
* 数据库物理结构设计需给出每个表的名称，表中各字段名称、类型、精度、约束条件 √
* 列出数据库中使用索引和视图的DDL语句。
* 完成系统实体的数据库持久层设计和实现工作
* 给出DAO基础实现代码。
* 编写数据库持久层的测试业务代码
* 围绕每个业务的关键需求，每个业务至少列出3个SQL语句，每个SQL语句提供至少2种实现，对比不同实现的差异。
* 自行根据系统业务需要，编写5条连接3个关系（含）以上的SQL语句
* 小四，宋体，1.5倍行距。

1. 数据库管理系统基本信息：

软件名称：MySQL

版本号：MySQL 8.0.39

操作系统：支持多种操作系统，包括Linux、Windows等

CPU架构：支持x86\_64架构

1. DDL语句：

-- Database dcdb

--学科负责人

CREATE TABLE DisciplineHead (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(50) NOT NULL,

discipline\_id INT NOT NULL,

graduate\_management\_secretary\_id INT NOT NULL,

FOREIGN KEY (discipline\_id) REFERENCES Discipline(id)

);

--学院

CREATE TABLE College (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

website VARCHAR(100) NOT NULL

);

--学科秘书

CREATE TABLE DisciplineSecretary (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(50) NOT NULL,

discipline\_id INT NOT NULL,

FOREIGN KEY (discipline\_id) REFERENCES Discipline(id)

);

--研究生管理秘书

CREATE TABLE GraduateManagementSecretary (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(50) NOT NULL,

college\_id INT NOT NULL,

FOREIGN KEY (college\_id) REFERENCES College(id)

);

--复试组

CREATE TABLE InterviewGroup (

id INT NOT NULL PRIMARY KEY,

discipline\_id INT NOT NULL,

FOREIGN KEY (discipline\_id) REFERENCES Discipline(id)

);

--学科

CREATE TABLE Discipline (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

type VARCHAR(50) NOT NULL,

quota INT,

overview TEXT NOT NULL,

parent\_discipline INT,

level INT NOT NULL,

college\_id INT NOT NULL,

FOREIGN KEY (college\_id) REFERENCES College(id)

);

--科目

CREATE TABLE Subject (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

type VARCHAR(50) NOT NULL

);

--导师

CREATE TABLE Teacher (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(50) NOT NULL,

phone VARCHAR(20) NOT NULL,

qualification VARCHAR(50),

image VARCHAR(255),

email VARCHAR(50) NOT NULL,

title VARCHAR(50) NOT NULL,

profile TEXT NOT NULL,

volunteer\_round INT NOT NULL,

college\_id INT NOT NULL,

FOREIGN KEY (college\_id) REFERENCES College(id)

);

--考生

CREATE TABLE Student (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(50) NOT NULL,

phone VARCHAR(20),

emergency\_phone VARCHAR(20),

email VARCHAR(50),

undergraduate\_major VARCHAR(100),

undergraduate\_school VARCHAR(100),

origin VARCHAR(100),

undergraduate\_type VARCHAR(50),

resume VARCHAR(255),

birth\_date DATE,

graduation\_date DATE,

exam\_ticket\_number VARCHAR(50),

id\_card\_number VARCHAR(50) NOT NULL,

category VARCHAR(50),

volunteer\_status VARCHAR(50),

graduate\_type VARCHAR(50)

);

--选择

CREATE TABLE Choice (

teacher\_id INT NOT NULL,

student\_id INT NOT NULL,

priority INT NOT NULL,

PRIMARY KEY (teacher\_id, student\_id),

FOREIGN KEY (teacher\_id) REFERENCES Teacher(id),

FOREIGN KEY (student\_id) REFERENCES Student(id)

);

--报考

CREATE TABLE Application (

student\_id INT NOT NULL,

discipline\_id INT NOT NULL,

preference\_order INT NOT NULL,

PRIMARY KEY (student\_id, discipline\_id),

FOREIGN KEY (student\_id) REFERENCES Student(id),

FOREIGN KEY (discipline\_id) REFERENCES Discipline(id)

);

--考试

CREATE TABLE Exam (

subject\_id INT NOT NULL,

student\_id INT NOT NULL,

type VARCHAR(50) NOT NULL,

score FLOAT NOT NULL,

time DATETIME,

location VARCHAR(255),

comment TEXT,

signature VARCHAR(255),

PRIMARY KEY (subject\_id, student\_id),

FOREIGN KEY (subject\_id) REFERENCES Subject(id),

FOREIGN KEY (student\_id) REFERENCES Student(id)

);

--考核

CREATE TABLE Assessment (

subject\_id INT NOT NULL,

discipline\_id INT NOT NULL,

type VARCHAR(50) NOT NULL,

PRIMARY KEY (subject\_id, discipline\_id),

FOREIGN KEY (subject\_id) REFERENCES Subject(id),

FOREIGN KEY (discipline\_id) REFERENCES Discipline(id)

);

--教学

CREATE TABLE Teaching (

discipline\_id INT NOT NULL,

teacher\_id INT NOT NULL,

academic\_quota INT NOT NULL,

professional\_quota INT NOT NULL,

direct\_phd\_quota INT NOT NULL,

PRIMARY KEY (discipline\_id, teacher\_id),

FOREIGN KEY (discipline\_id) REFERENCES Discipline(id),

FOREIGN KEY (teacher\_id) REFERENCES Teacher(id)

);

--研究生主管领导

CREATE TABLE GraduateManager (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(50) NOT NULL

);

--监督人员

CREATE TABLE Supervisor (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(50) NOT NULL

);

--管理员

CREATE TABLE Administrator (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(50) NOT NULL

);

--用户

CREATE TABLE User (

id INT NOT NULL,

user\_name VARCHAR(50) NOT NULL,

password VARCHAR(100) NOT NULL,

user\_identity ENUM NOT NULL,

PRIMARY KEY(id, user\_identity)

);

1. 物理结构设计：

① 学科负责人（DisciplineHead）

字段：

id（编号）：INT，主键，非空

name（姓名）：VARCHAR(50)，非空

discipline\_id（学科编号）：INT，外键，非空

② 学院（College）

字段：

id（编号）：INT，主键，非空

name（名称）：VARCHAR(100)，非空

website（官网）：VARCHAR(100)，非空

③ 学科秘书（DisciplineSecretary）

字段：

id（编号）：INT，主键，非空

name（姓名）：VARCHAR(50)，非空

discipline\_id（学科编号）：INT，外键，非空

④ 研究生管理秘书（GraduateManagementSecretary）

字段：

id（编号）：INT，主键，非空

name（姓名）：VARCHAR(50)，非空

college\_id（学院编号）：INT，外键，非空

⑤ 复试组（InterviewGroup）

字段：

id（编号）：INT，主键，非空

discipline\_id（学科编号）：INT，外键，非空

⑥ 学科（Discipline）

字段：

id（编号）：INT，主键，非空

name（名称）：VARCHAR(100)，非空

type（类型）：VARCHAR(50)，非空

quota（指标）：INT

overview（概述）：TEXT，非空

parent\_discipline（上级学科）：INT

level（等级）：INT，非空

college\_id（学院编号）：INT，外键，非空

⑦ 科目（Subject）

字段：

id（编号）：INT，主键，非空

name（名称）：VARCHAR(100)，非空

type（类别）：VARCHAR(50)，非空

⑧ 导师（Teacher）

字段：

id（编号）：INT，主键，非空

name（姓名）：VARCHAR(50)，非空

phone（电话）：VARCHAR(20)，非空

qualification（资格）：VARCHAR(50)

image（图片）：VARCHAR(255)

email（邮箱）：VARCHAR(50)，非空

title（职称）：VARCHAR(50)，非空

profile（简介）：TEXT，非空

volunteer\_round（志愿轮次）：INT，非空

college\_id（学院编号）：INT，外键，非空

⑨ 考生（Student）

字段：

id（编号）：INT，主键，非空

name（姓名）：VARCHAR(50)，非空

phone（电话）：VARCHAR(20)

emergency\_phone（紧急电话）：VARCHAR(20)

email（邮箱）：VARCHAR(50)

undergraduate\_major（本科专业）：VARCHAR(100)

undergraduate\_school（本科学校）：VARCHAR(100)

origin（生源地）：VARCHAR(100)

undergraduate\_type（本科类型）：VARCHAR(50)

resume（个人简历）：VARCHAR(255)

birth\_date（出生日期）：DATE

graduation\_date（毕业时间）：DATE

exam\_ticket\_number（准考证号）：VARCHAR(50)

id\_card\_number（身份证号）：VARCHAR(50)，非空

category（类别）：VARCHAR(50)

volunteer\_status（志愿状态）：VARCHAR(50)

graduate\_type（研究生类型）：VARCHAR(50)

⑩ 选择（Choice）

字段：

teacher\_id（导师编号）：INT，外键，非空

student\_id（考生编号）：INT，外键，非空

priority（优先级）：INT，非空

主键：(teacher\_id, student\_id)

⑪ 报考（Application）

字段：

student\_id（学生编号）：INT，外键，非空

discipline\_id（学科编号）：INT，外键，非空

preference\_order（顺序）：INT，非空

主键：(student\_id, discipline\_id)

⑫ 考试（Exam）

字段：

subject\_id（科目编号）：INT，外键，非空

student\_id（考生编号）：INT，外键，非空

type（类别）：VARCHAR(50)，非空

score（成绩）：FLOAT，非空

time（时间）：DATETIME

location（地点）：VARCHAR(255)

comment（评价）：TEXT

signature（电子签名）：VARCHAR(255)

主键：(subject\_id, student\_id)

⑬ 考核（Assessment）

字段：

subject\_id（科目编号）：INT，外键，非空

discipline\_id（学科编号）：INT，外键，非空

type（类别）：VARCHAR(50)，非空

主键：(subject\_id, discipline\_id)

⑭ 教学（Teaching）

字段：

discipline\_id（学科编号）：INT，外键，非空

teacher\_id（导师编号）：INT，外键，非空

academic\_quota（学硕名额）：INT，非空

professional\_quota（专硕名额）：INT，非空

direct\_phd\_quota（直博名额）：INT，非空

主键：(discipline\_id, teacher\_id)

⑮ 研究生主管领导（GraduateManager）

字段：

id（编号）：INT，主键，非空

name（姓名）：VARCHAR(50)，非空

⑯ 监督人员（Supervisor）

字段：

id（编号）：INT，主键，非空

name（姓名）：VARCHAR(50)，非空

⑰ 管理员（Administrator）

字段：

id（编号）：INT，主键，非空

name（姓名）：VARCHAR(50)，非空

⑱ 用户（User）

字段：

id（编号）：INT，主键，非空

user\_name（用户名）：VARCHAR(50)，非空

password（密码）：VARCHAR(100)，非空

user\_identity（用户身份）：ENUM，非空

1. 索引和视图：
2. 数据库持久层：
3. SQL语句：
4. 连接3个关系（含）以上的SQL语句：

六、项目运维管理和优化（观测点11.1）

撰写要求：

* 列举小组使用的工程管理和项目管理工具和实施过程，可提供必要的github截图或者小组讨论截图等佐证资料。
* 分析系统登录需要的安全策略以及决策控制机制
* 针对每个业务线设计至少3个实用的视图
* 围绕数据优化管理要求，给出至少4个视图以及4个索引设计方案并介绍方案的适用范围。
* 围绕数据库高级编程需要，每个业务给出至少1个存储过程和1个触发器实现方案并介绍适用范围.
* 围绕数据安全和数据备份角度，描述系统可采取的安全策略和风险控制策略。反例：数据安全照搬教材或网上的内容，没有与业务系统结合；撰写内容含糊，仅用1-2句话糊弄了事，可操作性较差。
* 总结项目管理过程中遇到的3个关键问题以及解决方案。
* 小四，宋体，1.5倍行距。

1. 项目管理：
2. 系统登录需要的安全策略以及决策控制机制：
3. 视图：

① 业务1：

② 业务2：

③ 业务3：

1. 视图与索引：
2. 存储过程与触发器：
3. 安全策略和风险控制策略：
4. 关键问题以及解决方案：

七、附件内容（观测点11.1）

提供内容：

* 根据课程设计产生的成果，以目录方式列举内容附件，建议采用附件编号和附件名称的组织方式，然后在提交材料中，按照附件目录内容命名附件中的文件。
* 附件需要至少包含全局UML类图和源代码。
* 附件还可包含其他表现工作量的材料。
* 附件中包含答辩的PPT以及答辩录屏。
* 附件中包含答辩过程中教师提问的反馈和解决方案。
* 提交报告前，删除报告中标注为黄色底色的模板信息。
* 小四，宋体，1.5倍行距。