建立角色：

CREATE TABLE 角色 (

角色ID INT PRIMARY KEY, -- 角色ID，主键

角色名称 VARCHAR(20) NOT NULL, -- 角色名称

角色描述 VARCHAR(150), -- 角色描述

角色权限 INT NOT NULL CHECK (角色权限 BETWEEN 0 AND 15) -- 角色权限，范围为 0 到 15，0 为最高权限

);

建立用户表：

CREATE TABLE 用户 (

用户ID INT PRIMARY KEY, -- 用户ID，主键

用户名 VARCHAR(20) NOT NULL, -- 用户名

更新时间 DATETIME, -- 更新时间

创建时间 DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP, -- 创建时间，默认为当前时间

角色ID INT, -- 角色ID，外键

邮箱 VARCHAR(50), -- 邮箱

手机号 VARCHAR(50), -- 手机号

姓名 VARCHAR(20), -- 姓名

密码 VARCHAR(20) NOT NULL, -- 密码

FOREIGN KEY (角色ID) REFERENCES 角色表(角色ID) -- 设置角色ID为外键，引用角色表的角色ID

);

创建学院表：

CREATE TABLE 学院 (

学院ID INT PRIMARY KEY,

学院名称 VARCHAR(50) NOT NULL

);

建立学科表，由于学科表与导师表中互相把对方的主键作为外键，在建立学科表时先不建立外键约束。

CREATE TABLE 学科 (

学科ID VARCHAR(15) PRIMARY KEY,

学科名称 VARCHAR(40),

学科等级 VARCHAR(10) CHECK (学科等级 IN ('一级学科', '二级学科')), -- 添加CHECK约束

学科概述 VARCHAR(500), -- 学科的一个描述

学科类型 VARCHAR(20), -- 学术型学科、专业学位型学科

学科研究方向 VARCHAR(100), -- 一级学科的上级学科ID为NULL

上级学科 VARCHAR(40), -- 上级学科的名称，允许为空

更新时间 DATETIME,

创建时间 DATETIME,

年度总招生指标 VARCHAR(20)，

增补指标 VARCHAR(20)，

);

再建立导师表：

CREATE TABLE 导师 (

导师ID INT PRIMARY KEY,

照片路径 VARCHAR(50),

用户ID INT,

更新时间 DATETIME,

创建时间 DATETIME,

是否具有招生资格 BIT,

简介 VARCHAR(1000),

职称 VARCHAR(50) CHECK (职称 IN ('副教授', '教授', '讲师')),

学院ID INT,

导师类别 VARCHAR(50),

审核状态 VARCHAR(20) CHECK (审核状态 IN ('已通过', '正在进行', '未通过')),

FOREIGN KEY (用户ID) REFERENCES 用户(用户ID),

FOREIGN KEY (学院ID) REFERENCES 学院(学院ID)

);

///

后续插入过程中 发现冗余

删除导是学生之间的关联

///

CREATE TABLE 招生目录 (

招生目录ID INT PRIMARY KEY,

招生年度 INT NOT NULL CHECK (招生年度 >= 1900 AND 招生年度 <= 2100), -- 约束确保年份在合理范围内

是否通过学校审核 BIT NOT NULL -- 使用 BIT 类型表示布尔值

);

建立招生目录条目表：

CREATE TABLE 招生目录条目 (

招生目录条目ID INT PRIMARY KEY,

年份 INT CHECK (年份 >= 1900 AND 年份 <= 2100), -- 年份范围约束

学院ID INT,

创建时间 DATETIME,

更新时间 DATETIME,

是否招收同等学力 BIT,

学科ID VARCHAR(15),

FOREIGN KEY (学院ID) REFERENCES 学院(学院ID),

FOREIGN KEY (学科ID) REFERENCES 学科(学科ID)

);

创建考试科目表：

CREATE TABLE 考试科目 (

考试科目ID INT PRIMARY KEY,

类别 VARCHAR(10) NOT NULL CHECK (类别 IN ('初试', '复试')), -- 可以是 "初试" 或 "复试"

名称 VARCHAR(30) NOT NULL,

考试时间 DATE NOT NULL,

考试地点 VARCHAR(50) NOT NULL

);

CREATE TABLE 总成绩 (

总成绩ID INT PRIMARY KEY,

类别 VARCHAR(10) NOT NULL,

总分数值 FLOAT NOT NULL,

更新时间 DATETIME,

创建时间 DATETIME,

考生ID INT,

FOREIGN KEY (考生ID) REFERENCES 考生(考生ID)

);

创建拥有表

CREATE TABLE 拥有 (

考生ID INT, -- 外键考生ID

总成绩ID INT, -- 总成绩ID

FOREIGN KEY (总成绩ID) REFERENCES 总成绩(总成绩ID), -- 外键引用总成绩表

FOREIGN KEY (考生ID) REFERENCES 考生(考生ID)

PRIMARY KEY (考生ID, 总成绩ID)

);

CREATE TABLE 复试志愿 (

志愿ID INT PRIMARY KEY,

导师ID INT,

考生ID INT,

学科ID VARCHAR(15),

志愿顺序 INT CHECK (志愿顺序 IN (1, 2, 3)), -- 志愿顺序只能在1、2、3中取值

是否接受方向调整 BIT, -- 使用BIT来表示BOOLEAN类型

更新时间 DATETIME,

创建时间 DATETIME,

FOREIGN KEY (导师ID) REFERENCES 导师(导师ID), -- 外键引用导师表

FOREIGN KEY (考生ID) REFERENCES 考生(考生ID), -- 外键引用考生表

FOREIGN KEY (学科ID) REFERENCES 学科(学科ID) -- 外键引用学科表

);

CREATE TABLE 复试结果 (

复试结果ID INT PRIMARY KEY,

考生ID INT,

更新时间 DATETIME,

创建时间 DATETIME,

最终导师ID INT,

录取状态 VARCHAR(20) CHECK (录取状态 IN ('已录取', '待审核', '未录取')), -- 录取状态限制为特定值

综合评价 VARCHAR(100),

FOREIGN KEY (考生ID) REFERENCES 考生(考生ID), -- 外键引用考生表

FOREIGN KEY (最终导师ID) REFERENCES 导师(导师ID) -- 外键引用导师表

);

创建具有表：

CREATE TABLE 具有 (

学科ID VARCHAR(15), -- 外键学科ID

招生目录条目ID INT, -- 外键招生目录ID

PRIMARY KEY (招生目录ID，学科ID), -- 将招生目录ID作为主键

FOREIGN KEY (学科ID) REFERENCES 学科(学科ID), -- 外键引用学科表

FOREIGN KEY (招生目录ID) REFERENCES 招生目录条目(招生目录条目ID) -- 外键引用招生目录表

);

CREATE TABLE 包含1 (

志愿ID INT, -- 外键复试志愿ID

学科ID VARCHAR(15), -- 外键学科ID

PRIMARY KEY (志愿ID), -- 将复试志愿ID作为主键

FOREIGN KEY (学科ID) REFERENCES 学科(学科ID), -- 外键引用学科表

FOREIGN KEY (志愿ID) REFERENCES 复试志愿(志愿ID)

);

CREATE TABLE 确定 (

复试结果ID INT, -- 外键复试结果ID

导师ID INT, -- 外键导师ID

PRIMARY KEY (导师ID，复试结果ID), -- 将导师ID作为主键

FOREIGN KEY (导师ID) REFERENCES 导师(导师ID), -- 外键引用导师表

FOREIGN KEY (复试结果ID) REFERENCES 复试结果(复试结果ID)

);

CREATE TABLE 对应1 (

导师ID INT, -- 外键导师ID

用户ID INT, -- 外键用户ID

PRIMARY KEY (导师ID), -- 将导师ID作为主键

FOREIGN KEY (用户ID) REFERENCES 用户(用户ID), -- 外键引用用户表

FOREIGN KEY (导师ID) REFERENCES 导师(导师ID)

);

CREATE TABLE 对应2 (

考生ID INT, -- 外键导师ID

用户ID INT, -- 外键用户ID

PRIMARY KEY (考生ID), -- 将导师ID作为主键

FOREIGN KEY (用户ID) REFERENCES 用户(用户ID), -- 外键引用用户表

FOREIGN KEY (考生ID) REFERENCES 考生(考生ID)

);

CREATE TABLE 获得 (

考生ID INT, -- 外键导师ID

复试结果ID INT, -- 外键用户ID

PRIMARY KEY (考生ID), -- 将导师ID作为主键

FOREIGN KEY (考生ID) REFERENCES 考生(考生ID), -- 外键引用用户表

FOREIGN KEY (复试结果ID) REFERENCES 复试结果(复试结果ID)

);

CREATE TABLE 从属 (

导师ID INT, -- 外键导师ID

学科ID VARCHAR(15), -- 外键用户ID

PRIMARY KEY (导师ID,学科ID), -- 将导师ID作为主键

FOREIGN KEY (导师ID) REFERENCES 导师(导师ID), -- 外键引用用户表

FOREIGN KEY (学科ID) REFERENCES 学科(学科ID)

);

CREATE TABLE 编制 (

学院ID INT, -- 外键导师ID

招生目录条目ID INT, -- 外键用户ID

PRIMARY KEY (学院ID), -- 将导师ID作为主键

FOREIGN KEY (学院ID) REFERENCES 学院(学院ID), -- 外键引用用户表

FOREIGN KEY (招生目录条目ID) REFERENCES 招生目录条目(招生目录条目ID)

);

CREATE TABLE 包含3 (

招生目录ID INT, -- 外键导师ID

招生目录条目ID INT, -- 外键用户ID

PRIMARY KEY (招生目录条目ID,生目录ID), -- 将导师ID作为主键

FOREIGN KEY (招生目录ID) REFERENCES 招生目录(招生目录ID), -- 外键引用用户表

FOREIGN KEY (招生目录条目ID) REFERENCES 招生目录条目(招生目录条目ID)

);

CREATE TABLE 作为 (

用户ID INT, -- 外键导师ID

角色ID INT, -- 外键用户ID

PRIMARY KEY (角色ID,用户ID), -- 将导师ID作为主键

FOREIGN KEY (用户ID) REFERENCES 用户(用户ID), -- 外键引用用户表

FOREIGN KEY (角色ID) REFERENCES 角色(角色ID)

);

CREATE TABLE 确定 (

复试结果ID INT, -- 外键复试结果ID

导师ID INT, -- 外键导师ID

PRIMARY KEY (导师ID，复试结果ID), -- 将复试结果ID作为主键

FOREIGN KEY (导师ID) REFERENCES 导师(导师ID), -- 外键引用导师表

FOREIGN KEY (复试结果ID) REFERENCES 复试结果(复试结果ID)

);

CREATE TABLE 拥有 (

考生ID INT, -- 外键考生ID

总成绩ID INT, -- 总成绩ID

类型 VARCHAR(20),

PRIMARY KEY (考生ID，总成绩ID) , -- 联合主键

FOREIGN KEY (总成绩ID) REFERENCES 总成绩(总成绩ID), -- 外键引用总成绩表

FOREIGN KEY (考生ID) REFERENCES 考生(考生ID)

);

CREATE TABLE 属于 (

导师ID INT, -- 外键，导师ID

学院ID INT, -- 外键，学院ID

PRIMARY KEY (导师ID, 学院ID), -- 设置联合主键

FOREIGN KEY (导师ID) REFERENCES 导师(导师ID), -- 外键引用导师表

FOREIGN KEY (学院ID) REFERENCES 学院(学院ID) -- 外键引用学院表

);

创建填报表：

CREATE TABLE 填报 (

考生ID INT, -- 外键：考生ID

志愿ID INT,

PRIMARY KEY (志愿ID), -- 主键

FOREIGN KEY (考生ID) REFERENCES 考生(考生ID),

FOREIGN KEY (志愿ID) REFERENCES 复试志愿(志愿ID)

);