2024—2025学年 第 一 学期

数据库系统及应用课程设计

目录

[一、需求分析 1](#_Toc11049588)

[1.1 问题描述 1](#_Toc11049589)

[1.2 系统需求分析 1](#_Toc11049590)

[二、概念结构设计 1](#_Toc11049591)

[2.1 概念结构设计工具（E-R模型） 1](#_Toc11049592)

[2.2 概念结构设计方法 1](#_Toc11049593)

[2.3 总体E-R图 2](#_Toc11049594)

[三、逻辑结构设计 2](#_Toc11049595)

[3.1关系数据模式设计 2](#_Toc11049596)

[3.2 视图的设计：即需定义什么样的视图表结构； 3](#_Toc11049597)

[3.3 优化 3](#_Toc11049598)

[3.4 设计关系模式的属性类型及约束 3](#_Toc11049599)

[四、数据库实施 4](#_Toc11049600)

[4.1 数据库的创建 4](#_Toc11049601)

[4.2 数据表的创建（可同时在创建时定义相关的约束） 4](#_Toc11049602)

[4.3 数据操作要求及实现 4](#_Toc11049603)

[4.3.1 插入数据操作 4](#_Toc11049604)

[4.3.2 删除数据操作 5](#_Toc11049605)

[4.3.3 修改数据操作 5](#_Toc11049606)

[4.3.4 数据查询操作 5](#_Toc11049607)

[4.4视图的创建 5](#_Toc11049608)

[4.5 存储过程的定义实现 5](#_Toc11049609)

[4.6 触发器的定义实现 6](#_Toc11049610)

[4.7 数据安全性 6](#_Toc11049611)

[4.7.1 创建登录用户 6](#_Toc11049612)

[4.7.2 创建数据库用户 6](#_Toc11049613)

[4.7.3 创建角色 7](#_Toc11049614)

[4.7.4 权限管理 7](#_Toc11049615)

[五、总结及体会 7](#_Toc11049616)

[六、主要参考文献 7](#_Toc11049617)

一、需求分析

# 1.1 问题描述

高校科研管理系统旨在帮助高校对科研项目、科研人员、科研成果及其相关数据进行高效的管理。通过该系统，可以实现科研项目的立项、进度跟踪、经费管理、成果登记等全流程管理，同时为科研人员提供一个高效的工作平台，以支持高校的科研工作有序开展。

# 1.2 系统需求分析

**1. 用户需求分析**

根据目标用户群体的不同，主要有以下几类用户：

* **科研人员**：可以查看自己参与的科研项目、成果信息，申请科研经费，更新科研进展。
* **项目管理**：负责项目的创建、任务分配、进度管理，跟踪

**2. 功能需求分析**

* 实现部门、职务、职称等基本信息的管理;
* 实现教师信息的管理;
* 实现可以科研项目的申报、审批管理;
* 实现科研项目的验收管理

二、概念结构设计

数据库的概念设计是在需求分析的基础上，利用与用户双方都能理解的形式，设计出数据库的概念模型。本课程设计要求采用E-R模型进行数据库的概念设计。

# 2.1 概念结构设计工具（E-R模型）

什么是概念结构设计？概念结构设计常用的工具是E-R模型。什么是E-R模型，如何设计E-R模型即用什么形状表示什么等。

**1、概念结构设计**是数据库设计的一个阶段，其主要任务是根据用户需求，构建一个与现实世界相对应的抽象数据模型。在这个阶段，设计人员会提炼出业务领域中各个实体及它们之间的关系，从而为后续的逻辑设计和物理设计奠定基础。概念结构设计的目标是将实际世界中的信息需求转化为一个易于理解、无歧义的模型。

**2、E-R模型简介**

**E-R模型（Entity-Relationship Model，实体-关系模型）是一种用于描述现实世界事物和事物之间关系的数据模型。E-R模型通过一组实体**（Entity）、**属性**（Attribute）和**关系**（Relationship）来表示数据。

* **实体（Entity）**：表示一个独立的、具有现实意义的对象或事物，通常是一个具体的物理事物或抽象概念。例如，学生、课程、书籍等。
* **属性（Attribute）**：表示实体的某一特性或描述。每个实体可以有多个属性，例如学生实体可能包含姓名、学号、年龄等属性。
* **关系（Relationship）**：表示实体之间的联系。例如，学生和课程之间的"选修"关系，教师和课程之间的"教授"关系。

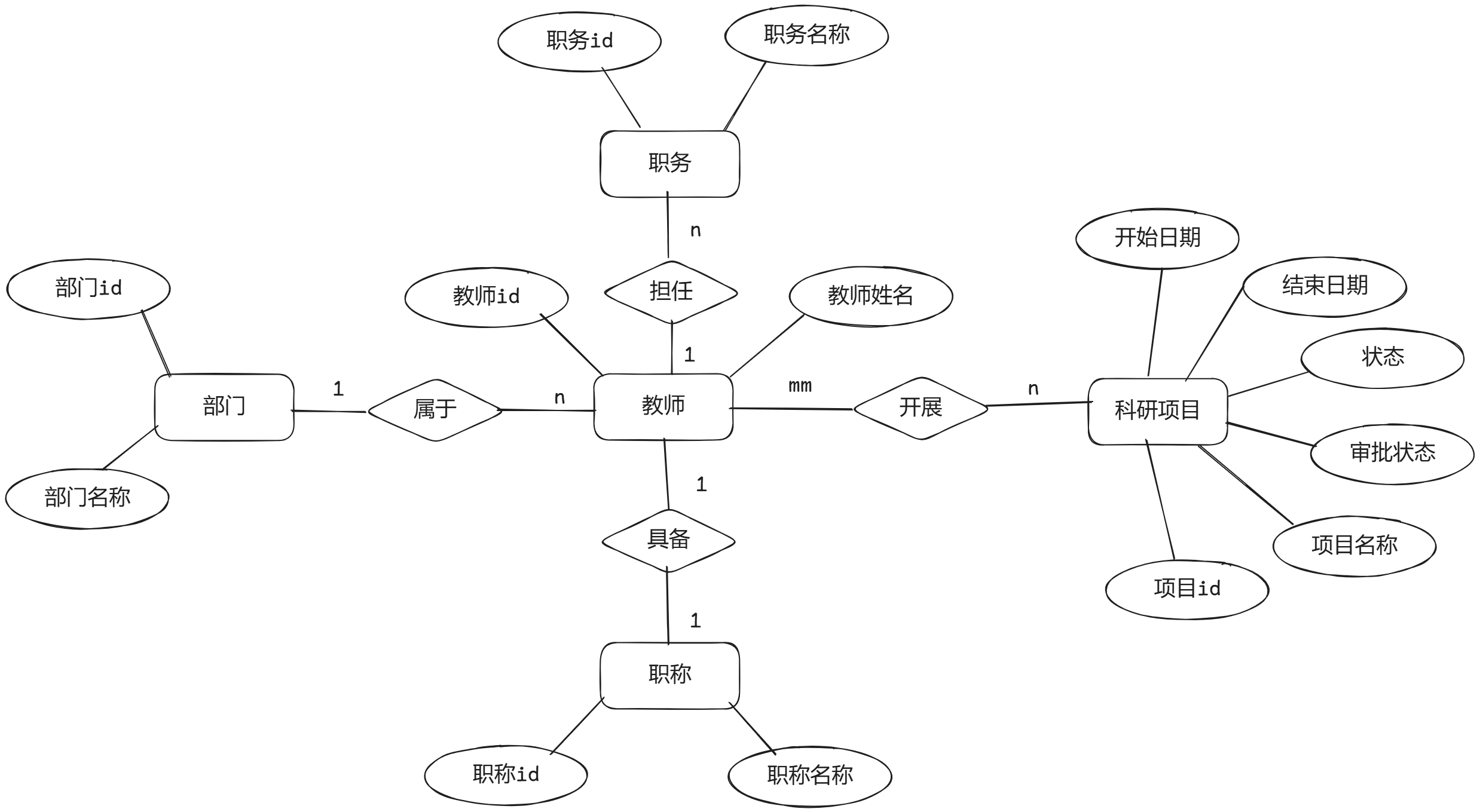
E-R模型的图形化表示通过使用特定的符号和形状来表达这些概念。

**2、E-R模型的符号和图形表示**

1. **实体（Entity）**
   * **形状**：用**矩形**表示。
   * **说明**：实体是现实世界中具有独立存在的对象，可以是具体的事物（如“学生”）或抽象的概念（如“课程”）。
2. **属性（Attribute）**
   * **形状**：用**椭圆形**表示，并通过直线与对应的实体连接。
   * **说明**：属性是描述实体特性的数据项。例如，学生实体的属性可以是姓名、学号、出生日期等。
3. **关系（Relationship）**
   * **形状**：用**菱形**表示，并通过直线与相关实体连接。
   * **说明**：关系描述了不同实体之间的关联。例如，学生与课程之间的关系可以是"选修"。
4. **主键（Primary Key）**
   * **形状**：通常用下划线标识。
   * **说明**：主键是实体的唯一标识符，确保每个实体的唯一性。
5. **多重性（Cardinality）**
   * **形状**：用数字标注，表示实体间关系的数量限制。
   * **说明**：例如，1对多（1:N），多对多（M:N），1对1（1:1）等。
6. **弱实体（Weak Entity）**
   * **形状**：用双矩形表示。
   * **说明**：弱实体依赖于其他实体，不能单独存在。它通常与强实体之间存在特定的关联。
7. **组合属性（Composite Attribute）**
   * **形状**：用嵌套的椭圆表示。
   * **说明**：组合属性由多个基本属性组成，例如地址可以包含街道、城市、邮政编码等。
8. **多值属性（Multivalued Attribute）**
   * **形状**：用双椭圆表示。
   * **说明**：多值属性可以有多个值，例如学生可以有多个电话号码。
9. **继承（Generalization/Specialization）**
   * **形状**：用三角形表示，并连接不同的实体。
   * **说明**：继承表示多个实体共同拥有的属性可以被抽象为一个超类，特定子类可以继承这个超类的属性。

# 2.3 总体E-R图

经过合并并消除冗余之后得到系统整体E-R如下图所示：



三、逻辑结构设计

# 3.1 设计关系模式的属性类型及约束

1. 部门

主键：部门ID

属性：部门ID, 部门名称

1. 职务

主键：职务ID

属性：职务ID, 职务名称

1. 职称

主键：职称ID

属性：职称ID, 职称名称

1. 教师

主键：教师ID

外键：

部门ID → 部门(部门ID)

职务ID → 职务(职务ID)

职称ID → 职称(职称ID)

属性：教师ID, 姓名, 部门ID, 职务ID, 职称ID

1. 科研项目

主键：项目ID

外键：

教师ID → 教师(教师ID)

属性：项目ID, 项目名称, 教师ID, 开始日期, 结束日期, 状态, 审批状态

四、数据库实施

# 4.1 数据库的创建

-- 1. 创建数据库

CREATE DATABASE gxky;

GO

USE gxky;

GO

# 4.2 数据表的创建（在创建表时定义相关的完整性约束）

-- 2. 创建基础信息表（部门、职务、职称）

CREATE TABLE Department (

DepartmentID INT PRIMARY KEY IDENTITY(1,1),

DepartmentName NVARCHAR(100) NOT NULL

);

CREATE TABLE Position (

PositionID INT PRIMARY KEY IDENTITY(1,1),

PositionName NVARCHAR(100) NOT NULL

);

CREATE TABLE Title (

TitleID INT PRIMARY KEY IDENTITY(1,1),

TitleName NVARCHAR(100) NOT NULL

);

-- 3. 创建教师信息表

CREATE TABLE Teacher (

TeacherID INT PRIMARY KEY IDENTITY(1,1),

Name NVARCHAR(100) NOT NULL,

DepartmentID INT FOREIGN KEY REFERENCES Department(DepartmentID),

PositionID INT FOREIGN KEY REFERENCES Position(PositionID),

TitleID INT FOREIGN KEY REFERENCES Title(TitleID)

);

-- 4. 创建科研项目表

CREATE TABLE ResearchProject (

ProjectID INT PRIMARY KEY IDENTITY(1,1),

ProjectName NVARCHAR(200) NOT NULL,

TeacherID INT FOREIGN KEY REFERENCES Teacher(TeacherID),

StartDate DATE NOT NULL,

EndDate DATE,

Status NVARCHAR(50) DEFAULT '未验收', -- 默认值为未验收

ApprovalStatus NVARCHAR(50) DEFAULT '未审批'

);

-- 5. 创建科研项目验收触发器

CREATE TRIGGER trg\_UpdateProjectStatus

ON ResearchProject

AFTER UPDATE

AS

BEGIN

IF UPDATE(Status)

BEGIN

UPDATE ResearchProject

SET Status = '验收通过'

WHERE Status = '验收中';

END

END;

GO

-- 6. 创建存储过程：统计各院系科研项目的申报和完成数量

CREATE PROCEDURE sp\_DeptProjectStats

AS

BEGIN

SELECT d.DepartmentName,

COUNT(CASE WHEN r.Status = '未验收' THEN 1 END) AS TotalSubmittedProjects,

COUNT(CASE WHEN r.Status = '验收通过' THEN 1 END) AS TotalCompletedProjects

FROM Department d

LEFT JOIN Teacher t ON d.DepartmentID = t.DepartmentID

LEFT JOIN ResearchProject r ON t.TeacherID = r.TeacherID

GROUP BY d.DepartmentName;

END;

GO

-- 7. 插入示例数据

INSERT INTO Department (DepartmentName) VALUES ('计算机科学学院'), ('机械工程学院'), ('电子信息学院');

INSERT INTO Position (PositionName) VALUES ('讲师'), ('副教授'), ('教授');

INSERT INTO Title (TitleName) VALUES ('助理研究员'), ('研究员');

INSERT INTO Teacher (Name, DepartmentID, PositionID, TitleID)

VALUES ('张三', 1, 1, 1), ('李四', 2, 2, 2), ('王五', 3, 3, 1);

INSERT INTO ResearchProject (ProjectName, TeacherID, StartDate, Status)

VALUES ('人工智能研究', 1, '2024-01-01', '未验收'),

('机械自动化项目', 2, '2023-06-01', '未验收'),

('物联网开发项目', 3, '2022-08-01', '验收中');

-- 8. 执行存储过程

EXEC sp\_DeptProjectStats;

# 4.3 数据操作要求及实现

4.3.1 插入数据操作

INSERT INTO Department (DepartmentName)

VALUES

('计算机科学学院'),

('机械工程学院'),

('电子信息学院'),

('土木工程学院'),

('化学工程学院'),

('物理学院'),

('数学学院'),

('生命科学学院'),

('经济管理学院'),

('艺术学院');

INSERT INTO Position (PositionName)

VALUES

('讲师'),

('副教授'),

('教授'),

('助理'),

('高级工程师'),

('主任'),

('副主任'),

('实验员'),

('助理教授'),

('研究员');

INSERT INTO Title (TitleName)

VALUES

('助理研究员'),

('研究员'),

('高级研究员'),

('学科带头人'),

('实验技术专家'),

('副研究员'),

('博士后'),

('访问学者'),

('兼职教授'),

('特聘教授');

INSERT INTO Teacher (Name, DepartmentID, PositionID, TitleID)

VALUES

('张三', 1, 1, 1),

('李四', 2, 2, 2),

('王五', 3, 3, 1),

('赵六', 4, 4, 3),

('孙七', 5, 5, 4),

('周八', 6, 6, 5),

('吴九', 7, 7, 6),

('郑十', 8, 8, 7),

('王小明', 9, 9, 8),

('李小华', 10, 10, 9);

INSERT INTO ResearchProject (ProjectName, TeacherID, StartDate, Status)

VALUES

('人工智能研究', 1, '2024-01-01', '未验收'),

('机械自动化项目', 2, '2023-06-01', '未验收'),

('物联网开发项目', 3, '2022-08-01', '验收中'),

('土木结构稳定性研究', 4, '2023-03-15', '未验收'),

('有机合成化学研究', 5, '2022-09-20', '未验收'),

('量子物理实验项目', 6, '2023-11-11', '验收中'),

('数学建模应用研究', 7, '2024-04-01', '未验收'),

('生物多样性保护研究', 8, '2023-05-10', '未验收'),

('经济发展与政策分析', 9, '2022-06-30', '未验收'),

('艺术设计创新项目', 10, '2024-02-15', '未验收');

* + 1. 删除数据操作：

-- 删除指定的科研项目：

DELETE FROM ResearchProject

WHERE ProjectID = 3; -- 删除ID为3的项目

-- 删除指定的教师信息：

DELETE FROM Teacher

WHERE TeacherID = 5; -- 删除ID为5的教师

* + 1. 修改数据操作

-- 修改科研项目的状态：

UPDATE ResearchProject

SET Status = '验收通过'

WHERE ProjectID = 2; -- 修改ID为2的项目状态为“验收通过”-- 修改教师的职务：

UPDATE Teacher

SET PositionID = 3

WHERE TeacherID = 4; -- 将ID为4的教师职务改为ID为3（教授）

* + 1. 数据查询操作

-- 查询所有“未验收”的科研项目：

SELECT ProjectID, ProjectName, StartDate, Status

FROM ResearchProject

WHERE Status = '未验收';

-- 查询各部门的教师数量：

SELECT d.DepartmentName, COUNT(t.TeacherID) AS 教师数量

FROM Department d

LEFT JOIN Teacher t ON d.DepartmentID = t.DepartmentID

GROUP BY d.DepartmentName;

-- 查询所有教师及其对应的项目名称：

SELECT t.Name AS 教师姓名, rp.ProjectName AS 科研项目

FROM Teacher t

LEFT JOIN ResearchProject rp ON t.TeacherID = rp.TeacherID;

# 4.4视图的创建

-- 8. 创建视图，显示教师及其所属部门的科研项目

CREATE VIEW vw\_TeacherProjects AS

SELECT t.TeacherID, t.Name AS 教师姓名, d.DepartmentName AS 所属部门, rp.ProjectName AS 科研项目名称, rp.Status AS 项目状态

FROM Teacher t

INNER JOIN Department d ON t.DepartmentID = d.DepartmentID

LEFT JOIN ResearchProject rp ON t.TeacherID = rp.TeacherID;

-- 9. 调用视图并打印结果

SELECT \* FROM vw\_TeacherProjects;



# 4.5 存储过程的定义实现

CREATE PROCEDURE sp\_DeptProjectStats

AS

BEGIN

SELECT d.DepartmentName,

COUNT(CASE WHEN r.Status = '未验收' THEN 1 END) AS TotalSubmittedProjects,

COUNT(CASE WHEN r.Status = '验收通过' THEN 1 END) AS TotalCompletedProjects

FROM Department d

LEFT JOIN Teacher t ON d.DepartmentID = t.DepartmentID

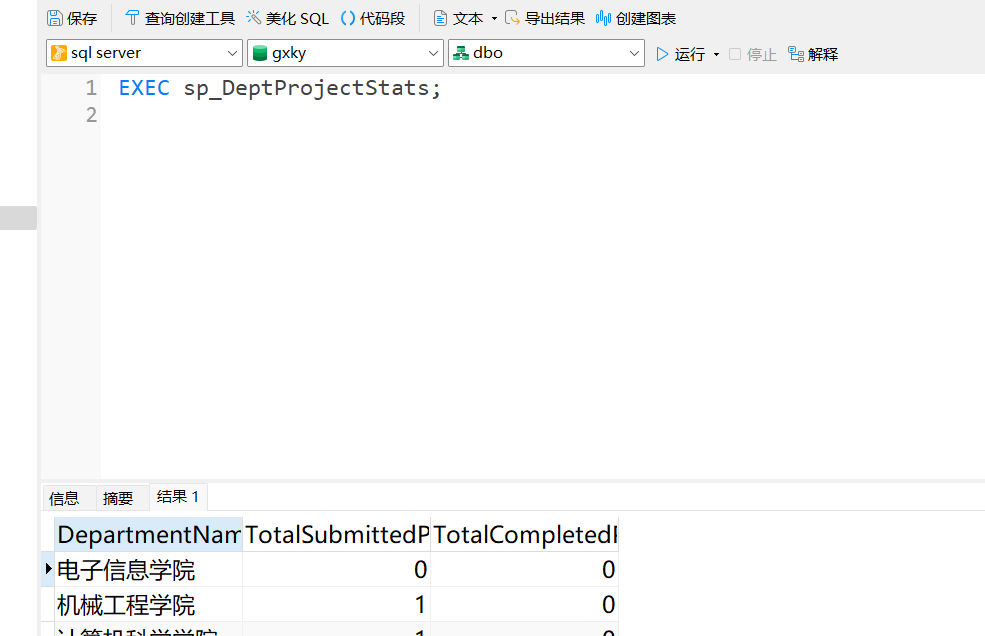
LEFT JOIN ResearchProject r ON t.TeacherID = r.TeacherID

GROUP BY d.DepartmentName;

END;

GO

EXEC sp\_DeptProjectStats;



# 4.6 触发器的定义实现

根据选题中创建触发器的要求，写出相应的SQL语句实现，可以把题目中的创建触发器的要求复制过来。（把题目要求中的创建触发器文字复制过来后可以把这段说明文字删除）

T-SQL语句：

-- 5. 创建科研项目验收触发器

CREATE TRIGGER trg\_UpdateProjectStatus

ON ResearchProject

AFTER UPDATE

AS

BEGIN

IF UPDATE(Status)

BEGIN

UPDATE ResearchProject

SET Status = '验收通过'

WHERE Status = '验收中';

END

END;

GO

# 数据安全性

**1、 身份验证 (Authentication)**

* + **Windows 身份验证**：利用操作系统的用户账户进行验证，适用于企业内部网络环境。
  + **SQL Server 身份验证**：使用SQL Server自身的用户名和密码进行验证，适合外部用户或混合环境。
  + **多因素认证 (MFA)**：结合多种验证方式（如密码、短信验证码、生物识别等），增强登录安全性。

**2、 网络安全性**

**防火墙配置**：限制对SQL Server实例的网络访问，只允许特定IP地址或端口连接。

* + **虚拟专用网络 (VPN)**：为远程用户创建安全的网络隧道，确保数据传输的安全性。

4.7.1 创建登录用户

创建一个“SQL SERVER验证模式”的登录账户，登录名为自己的S自己的学号，登录密码为“ABC123#”，默认数据库为自己课程设计所创建的数据库。 对应的SQL语句如下：

-- 创建SQL Server身份验证模式的登录账户

CREATE LOGIN S230084160143

WITH PASSWORD = 'ABC123#',

DEFAULT\_DATABASE = gxky,

CHECK\_EXPIRATION = OFF, -- 禁用密码过期检查

CHECK\_POLICY = OFF; -- 禁用Windows密码策略

4.7.2 创建数据库用户

在“\*\*\*”数据库中创建一个用户，用户名为自己姓名，并将它和登录名为S自己学号的登录用户关联在一起。对应的SQL语句如下：

-- 创建用户并将其与登录名 S230084160143 关联

CREATE USER [牛靖宇] FOR LOGIN [S230084160143];

GO

-- (可选) 授予用户特定权限或角色

-- 例如，授予 db\_datareader 和 db\_datawriter 角色

ALTER ROLE db\_datareader ADD MEMBER [牛靖宇];

ALTER ROLE db\_datawriter ADD MEMBER [牛靖宇];

GO

-- (可选) 授予更多具体的权限或自定义角色

-- 例如，授予对特定表的 SELECT 权限

GRANT SELECT ON OBJECT::[Teacher] TO [牛靖宇];

GO

4.7.3 创建角色

创建一个自定义角色“njy”，并将“姓名”的用户添加为其成员。对应的SQL语句如下：

USE [gxky];

GO

-- 创建自定义角色

CREATE ROLE njy;

GO

-- 将用户“姓名”添加到自定义角色“\*\*\*”

ALTER ROLE [njy] ADD MEMBER [牛靖宇];

GO

4.7.4 权限管理

为创建的角色分配数据库中各对象的操作权限。

1、对ResearchProject表拥有增加、查询权限。

USE [gxky];

GO

-- 确保角色 njy 已经存在

IF NOT EXISTS (SELECT \* FROM sys.database\_principals WHERE name = 'njy')

BEGIN

CREATE ROLE [njy];

END

GO

-- 1. 对 ResearchProject 表拥有增加、查询权限

GRANT INSERT, SELECT ON OBJECT::[ResearchProject] TO [njy];

GO

-- 2. 对 Department 表拥有查询权限

GRANT SELECT ON OBJECT::[Department] TO [njy];

GO

五、总结及体会

这是对数据库技术课程设计的总结，每个学生都应该客观地对自己的课程设计进行认真总结。大致包括以下内容：

通过本项目的设计与实现，我对数据库设计、SQL语法以及数据库管理有了更深入的理解。在项目中，我学习了如何创建数据库表、定义主键与外键约束，如何通过触发器实现数据的自动化更新，以及使用存储过程和视图实现复杂的数据查询和统计功能。整个项目的开发过程中，锻炼了我的数据分析和逻辑思维能力，同时提升了实际解决问题的能力。在未来的学习和工作中，我将更加注重数据库设计的规范性和高效性，持续提升自己的技能水平。

六、主要参考文献

[1]雷景生,叶文珺等.数据库原理及应用[M].清华大学出版社，2015.

[2]周爱武,汪海威等.数据库课程设计[M].机械工业出版社，2017.