**《数据库系统原理》实验报告**

**实验题目：**

**姓名：** **实验日期： 年 月 日**

**实验内容及完成情况：**（可续页）

### 实验4 数据库的完整性设计

**实验学时：1学时**

**实验类型：验证**

**实验要求：必做**

**一、实验目的**

熟练掌握使用Navicat和Transact-SQL语言两种方法创建、修改和删除表。掌握管理表的有关系统存储过程。

**二、实验内容**

1、在数据库hub（已有学生基本情况表student)中，利用Navicat和SQL语句创建以下表格：

1）表名为teacher（教师基本情况表），表中各个属性的定义如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 含义 | 数据类型 | 长度 | 能否取空值 | 备注 |
| Tno | 工号 | int |  | no | 主码 |
| Tname | 姓名 | varchar | 20 | no |  |
| gender | 性别 | char | 1 | no |  |
| age | 年龄 | tinyint |  | no |  |
| institute | 院系 | varchar | 40 | no |  |
| professional\_title | 职称 | varchar | 20 | no |  |

SQL语句：

CREATE TABLE teacher (

Tno INT NOT NULL PRIMARY KEY, -- 工号，主码

Tname VARCHAR(20) NOT NULL, -- 姓名

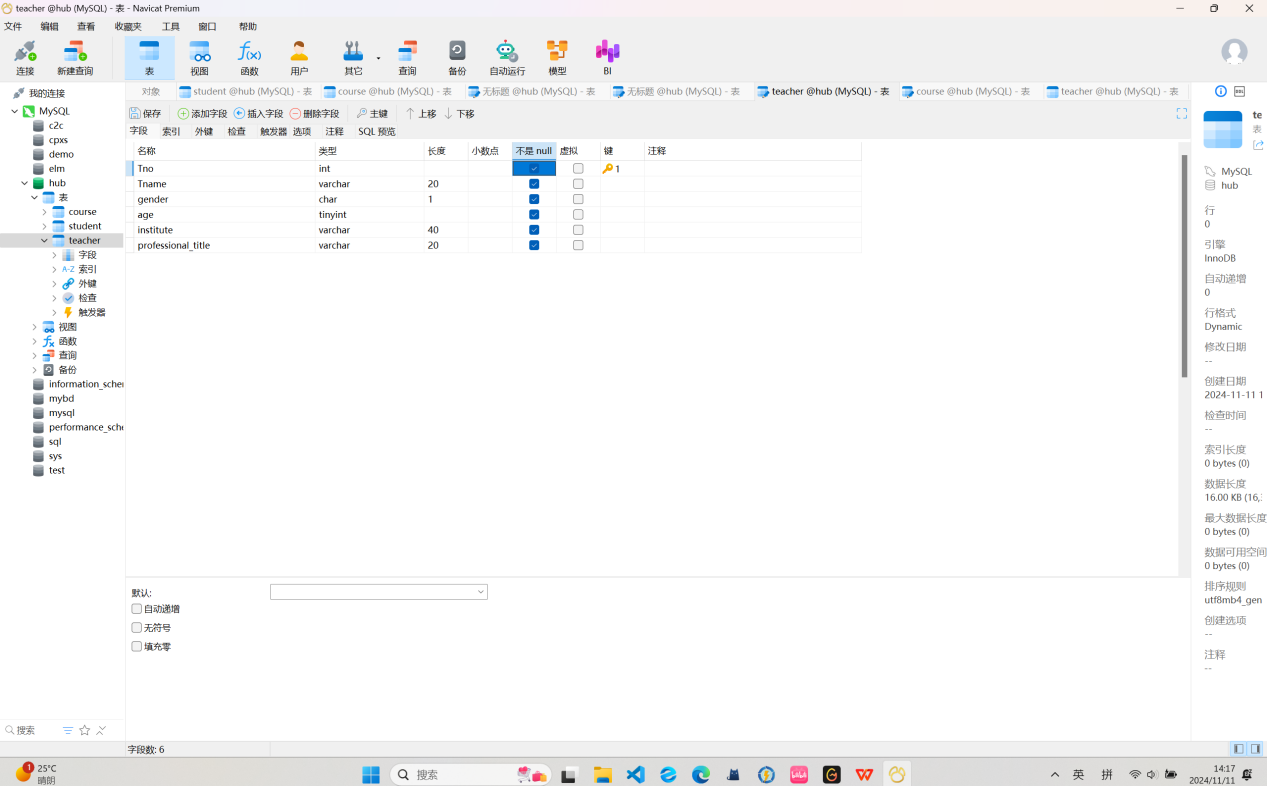
gender CHAR(1) NOT NULL, -- 性别

age TINYINT NOT NULL, -- 年龄

institute VARCHAR(40) NOT NULL, -- 院系

professional\_title VARCHAR(20) NOT NULL -- 职称

);



1. 表名为course（课程基本情况表），表中各个属性的定义如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 含义 | 数据类型 | 长度 | 能否取空值 | 备注 |
| Cno | 课程号 | int |  | no | 主码，自增 |
| Cname | 课程名 | varchar | 40 | no |  |
| credit | 学分 | float | （255，1） | no |  |

SQL语句：

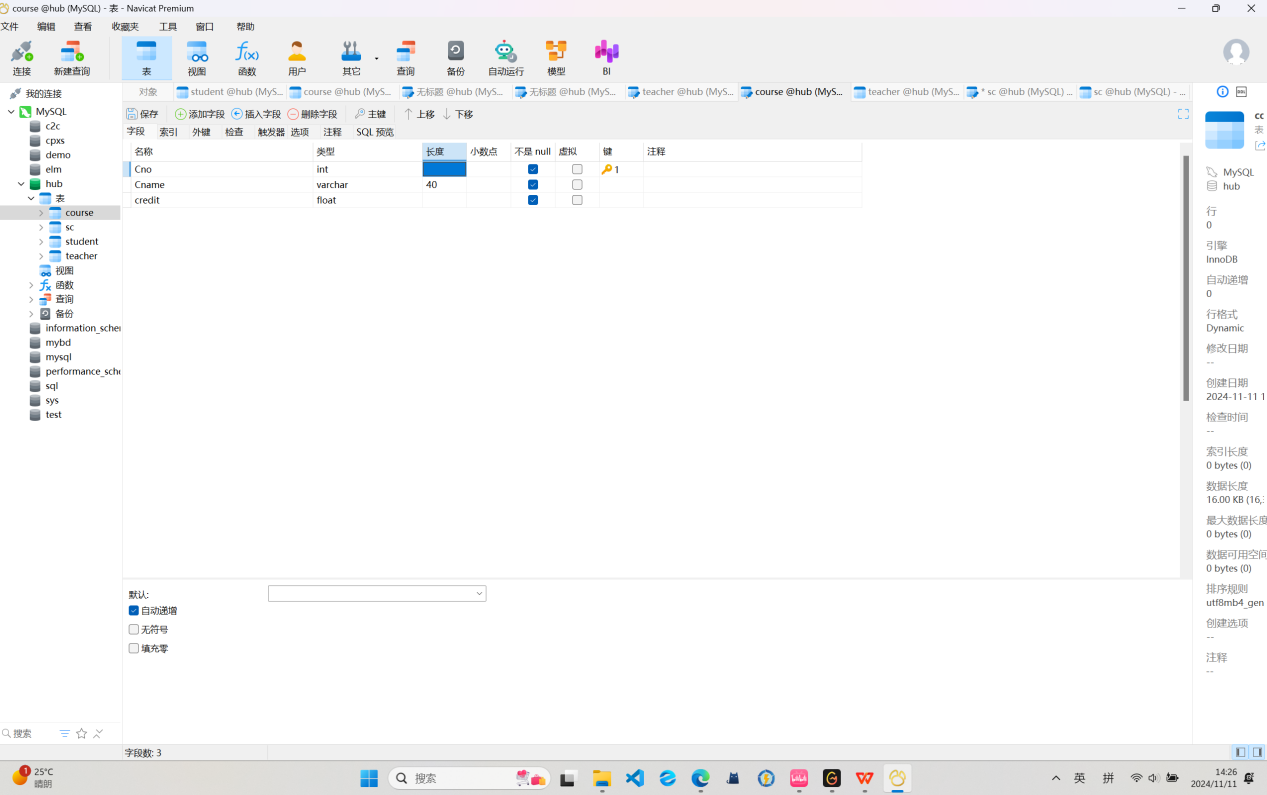
CREATE TABLE course (

Cno INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, -- 课程号，主码，自增

Cname VARCHAR(40) NOT NULL, -- 课程名

credit FLOAT(255, 1) NOT NULL -- 学分，保留一位小数

);



3）表名为sc（学生选课情况表），表中各个属性的定义如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 含义 | 数据类型 | 长度 | 能否取空值 | 备注 |
| Sno | 学号 | int |  | no | 主码，外键 |
| Cno | 课程号 | Int |  | no | 主码，外键 |
| grade | 分数 | float | （255，2） | yes |  |

SQL语句：

CREATE TABLE sc (

Sno INT NOT NULL, -- 学号，主码，外键

Cno INT NOT NULL, -- 课程号，主码，外键

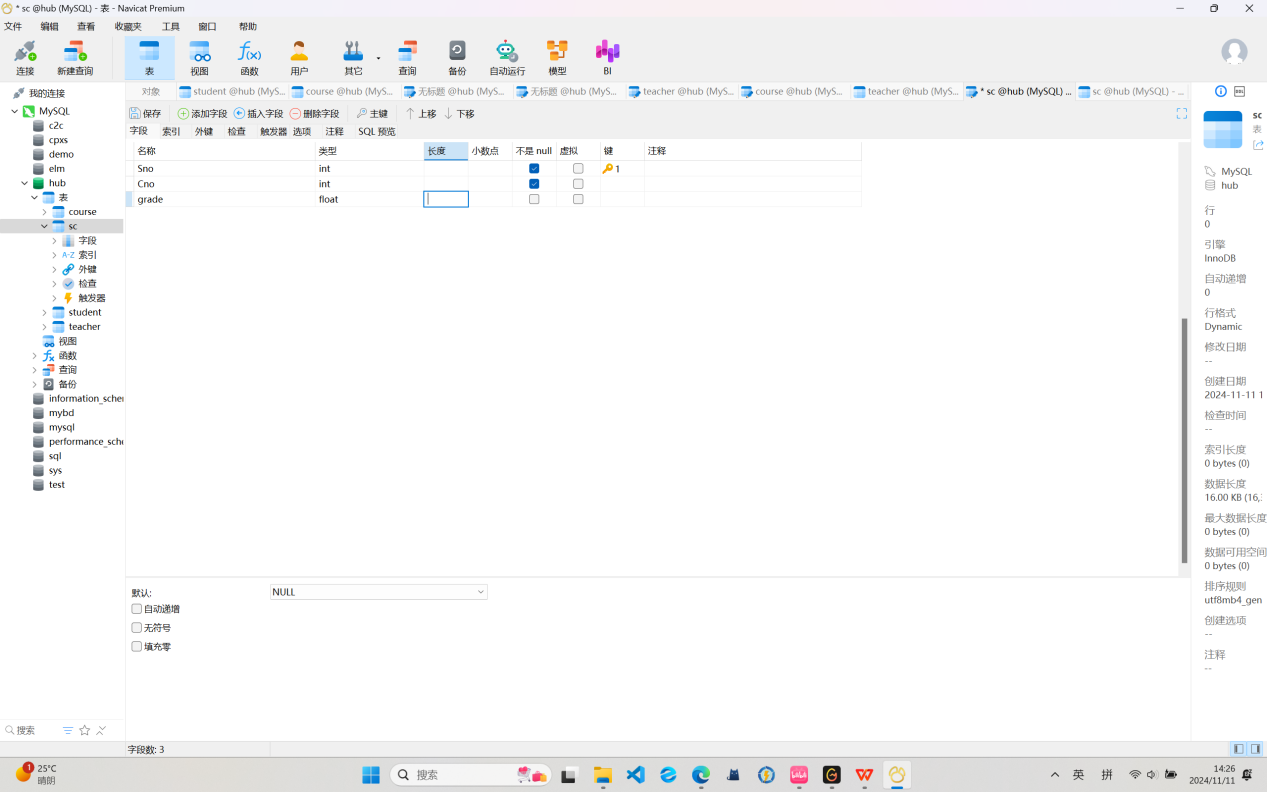
grade FLOAT(255, 2), -- 分数，保留两位小数，可为空

PRIMARY KEY (Sno, Cno), -- 复合主键

FOREIGN KEY (Sno) REFERENCES student(Sno), -- 学号外键引用学生表的Sno

FOREIGN KEY (Cno) REFERENCES course(Cno) -- 课程号外键引用课程表的Cno

);



4）表名为tc（教师教授课程情况表），表中各属性的定义如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 含义 | 数据类型 | 长度 | 能否取空值 | 备注 |
| Tno | 学号 | int |  | no | 主码，外键 |
| Cno | 课程号 | int |  | no | 主码，外键 |

SQL语句：

CREATE TABLE tc (

Tno INT NOT NULL, -- 工号，主码，外键

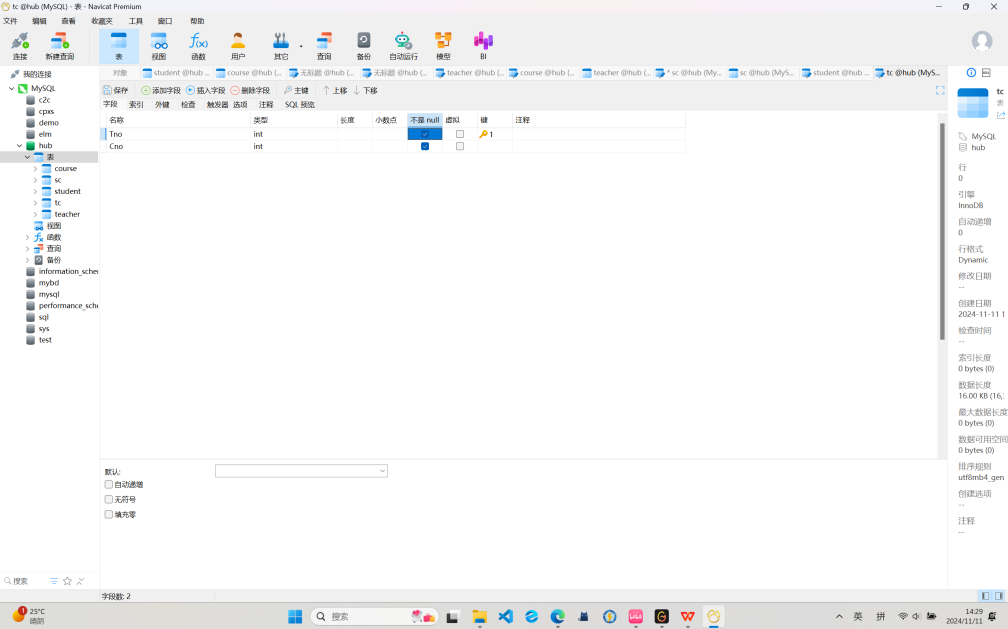
Cno INT NOT NULL, -- 课程号，主码，外键

PRIMARY KEY (Tno, Cno), -- 复合主键

FOREIGN KEY (Tno) REFERENCES teacher(Tno), -- 工号外键引用教师表的Tno

FOREIGN KEY (Cno) REFERENCES course(Cno) -- 课程号外键引用课程表的Cno

);



1. 在teacher表中定义Tno为主键。

ALTER TABLE teacher

ADD PRIMARY KEY (Tno);

1. 在sc表中定义Sno和Cno联合构成主键。

ALTER TABLE sc

ADD PRIMARY KEY (Sno, Cno);

4、在tc表中定义Tno和Cno联合构成主键。

ALTER TABLE tc

ADD PRIMARY KEY (Tno, Cno);

5、定义course表中的credit列默认值为0。

ALTER TABLE course

MODIFY credit FLOAT(255, 1) NOT NULL DEFAULT 0;

6、定义teacher表中的gender列的CHECK约束“男”或“女”。

ALTER TABLE teacher

ADD CONSTRAINT chk\_gender CHECK (gender IN ('男', '女'));

7、定义course表中的Cno列为自增。

ALTER TABLE course

MODIFY Cno INT NOT NULL AUTO\_INCREMENT;

7、在student表与sc表之间定义外键Sno。

ALTER TABLE sc

ADD CONSTRAINT fk\_sc\_student FOREIGN KEY (Sno) REFERENCES student(Sno);

8、在course表与sc表之间定义外键Cno。

ALTER TABLE sc

ADD CONSTRAINT fk\_sc\_course FOREIGN KEY (Cno) REFERENCES course(Cno);

9、在teacher表与tc表之间定义外键Tno。

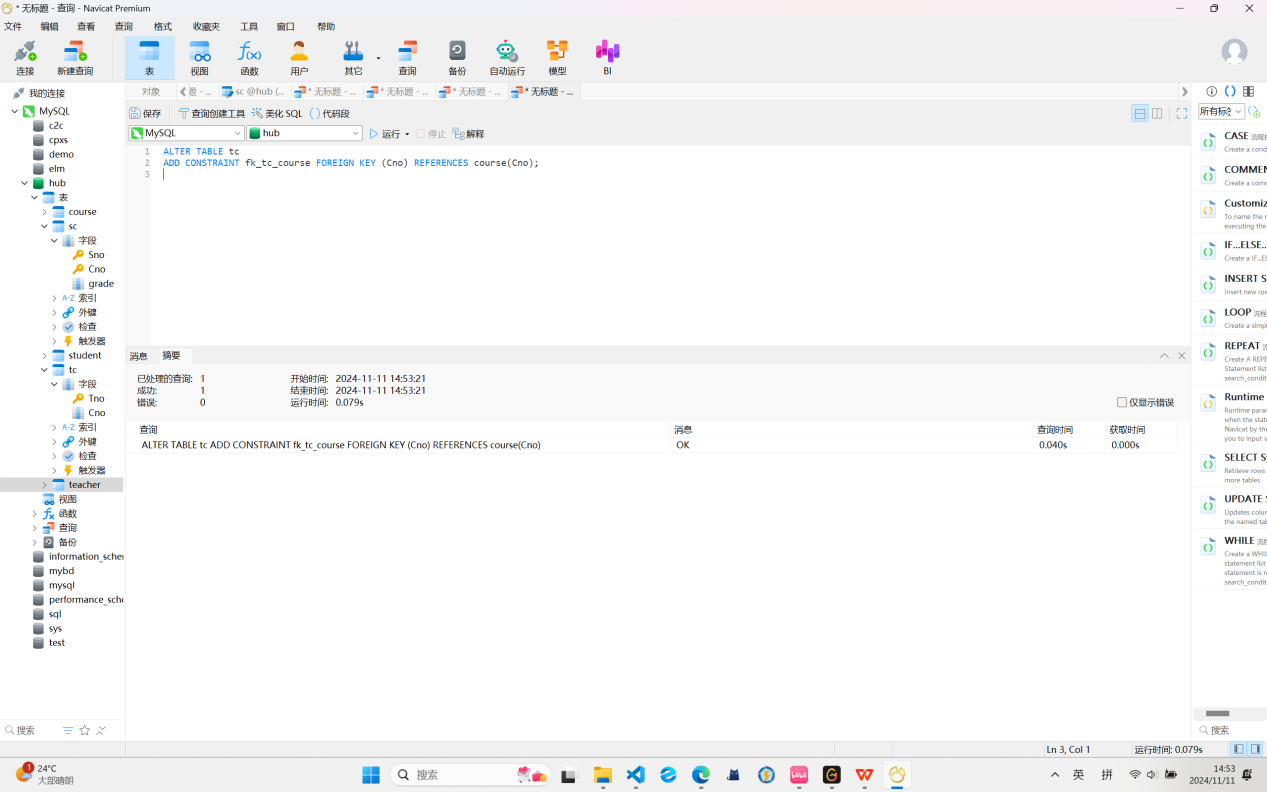
ALTER TABLE tc

ADD CONSTRAINT fk\_tc\_teacher FOREIGN KEY (Tno) REFERENCES teacher(Tno);

1. 在course表与tc表之间定义外键Cno。

ALTER TABLE tc

ADD CONSTRAINT fk\_tc\_course FOREIGN KEY (Cno) REFERENCES course(Cno);



**实验总结：**

本实验通过使用 Navicat 和 SQL 语句创建了 teacher、course、sc、tc 四个表，并设置了主键、外键、自增列、默认值及 CHECK 约束。通过对表结构和数据完整性约束的设置，使各表之间的数据关联性和完整性得到了保障。实验巩固了数据库设计的基本技能，提升了对表管理和约束定义的理解。

**教师评语及成绩**：