**《数据库系统原理》实验报告**

**实验题目：表的创建和管理**

**姓名： 郑德凯** **实验日期： 2024年 11月 5日**

**实验内容及完成情况：**

# 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

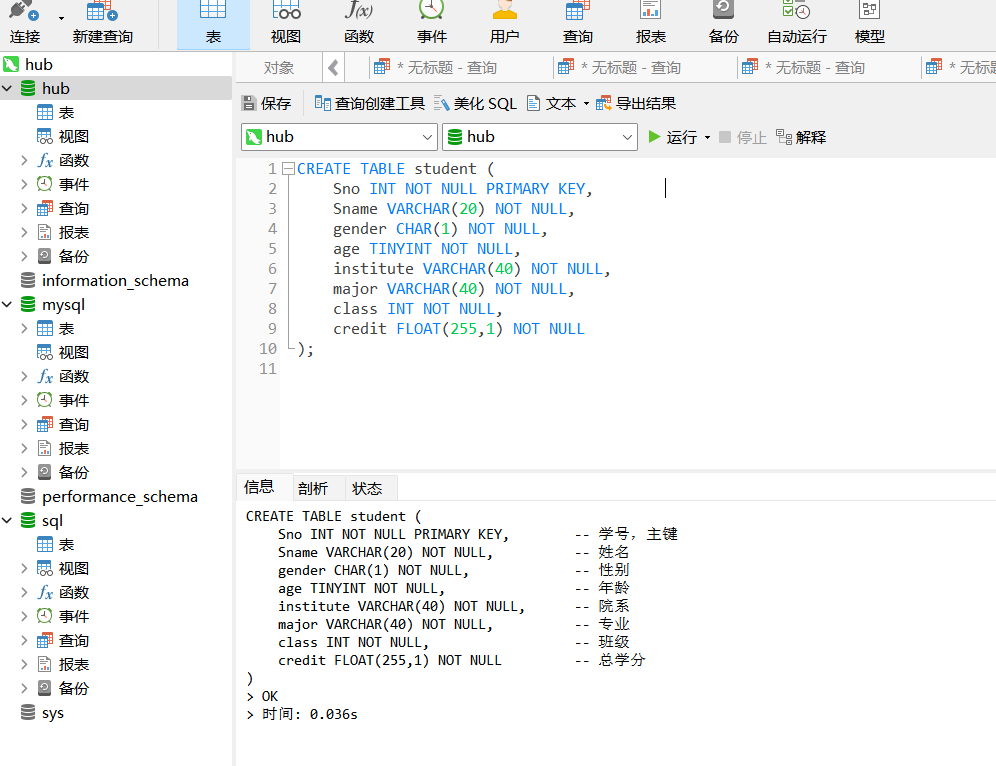
## 1、利用Navicat创建满足要求的数据库

1、在数据库hub中，利用Navicat创建以下表格：

1）表格名为student（学生基本情况表）；

2）表格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 |
| Sname | 姓名 | varchar | 20 | no |  |
| gender | 性别 | char | 1 | no |  |
| age | 年龄 | tinyint |  | no |  |
| institute | 院系 | varchar | 40 | no |  |
| major | 专业 | varchar | 40 | no |  |
| class | 班级 | int |  | no |  |
| credit | 总学分 | float | （255，1） | no |  |



## 2、按照以下步骤向表格xs中添加如下记录

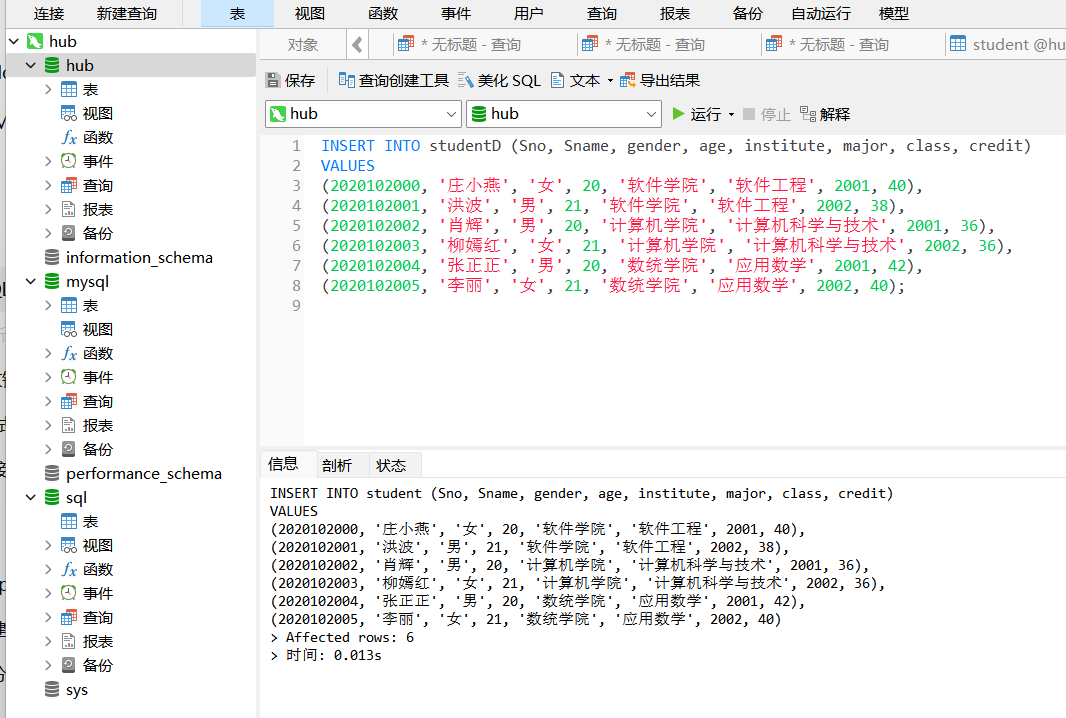
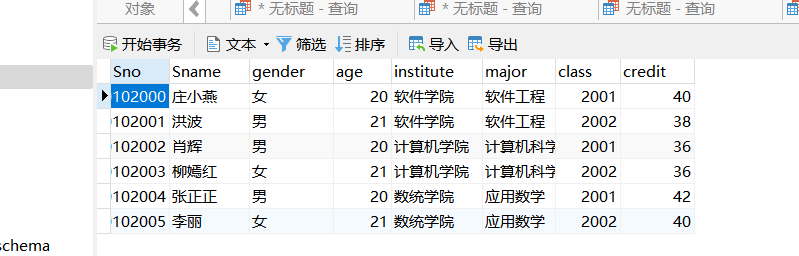
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sno  （学号） | Sname  （姓名） | gender  （性别） | age  （年龄） | institute  （院系） | major  （专业） | class  （班级） | credit  （总学分） |
| 2020102000 | 庄小燕 | 女 | 20 | 软件学院 | 软件工程 | 2001 | 40 |
| 2020102001 | 洪波 | 男 | 21 | 软件学院 | 软件工程 | 2002 | 38 |
| 2020102002 | 肖辉 | 男 | 20 | 计算机学院 | 计算机科学与技术 | 2001 | 36 |
| 2020102003 | 柳嫣红 | 女 | 21 | 计算机学院 | 计算机科学与技术 | 2002 | 36 |
| 2020102004 | 张正正 | 男 | 20 | 数统学院 | 应用数学 | 2001 | 42 |
| 2020102005 | 李丽 | 女 | 21 | 数统学院 | 应用数学 | 2002 | 40 |

1）打开Navicat；

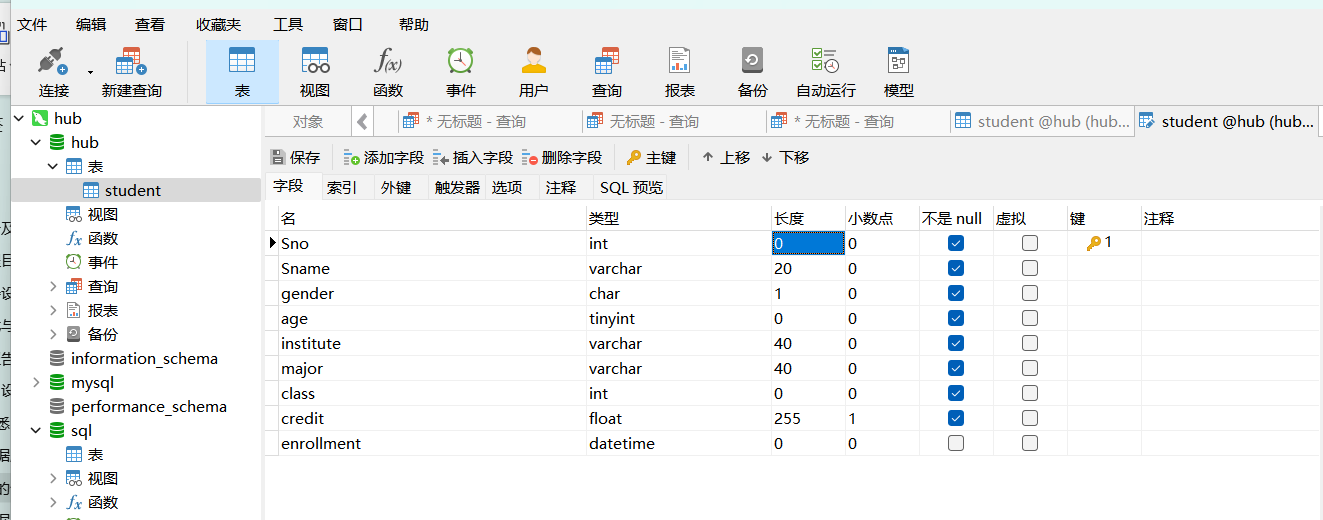
2）打开连接MySQL找到数据库hub中的表sdutents；

3）在表sdutent上右击鼠标，选择“打开表”；

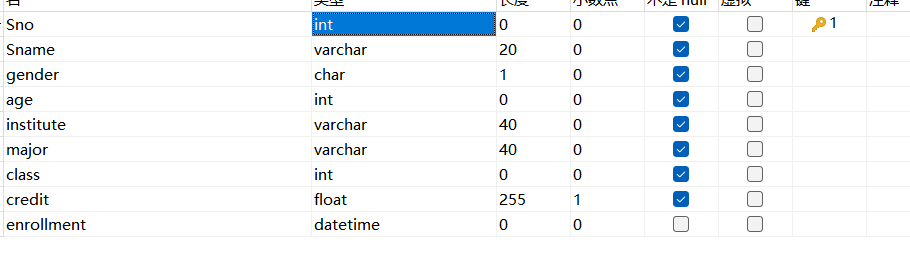
4）在接着出现的记录录入界面上添加记录。（方向键下增加新纪录）



## 向表students中增加“入学时间”属性列，其列名为enrollment，数据类型为datetime型



## 将表students中age（年龄）列的数据类型改为int型

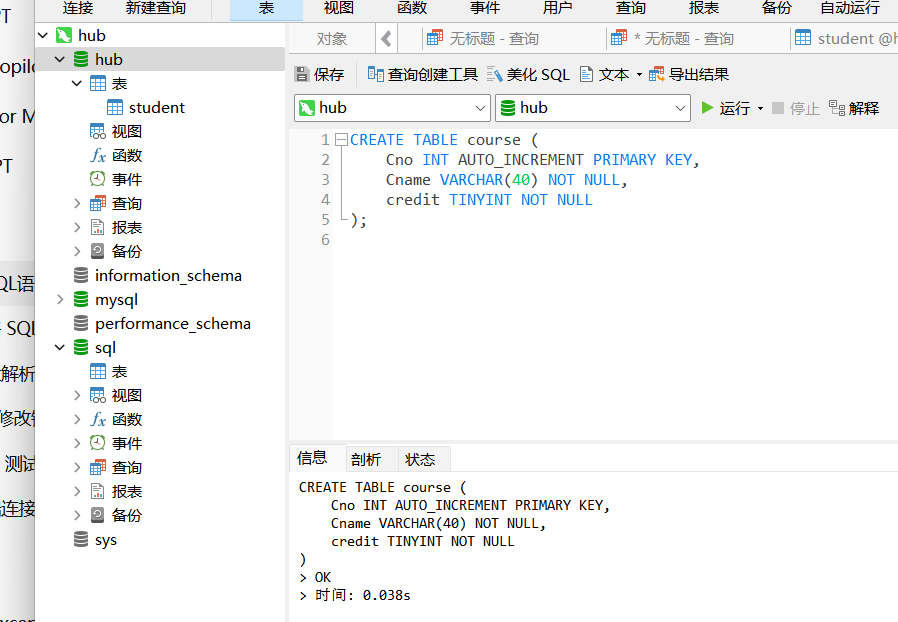


## 5、在数据库hub中，利用Transact-SQL语言创建以下表格：

1）表名为course（课程情况表）；

2）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 | 0 | no | 主码、自增 |
| Cname | 课程名 | varchar | 40 | no |  |
| credit | 学分 | tinyint |  | no |  |

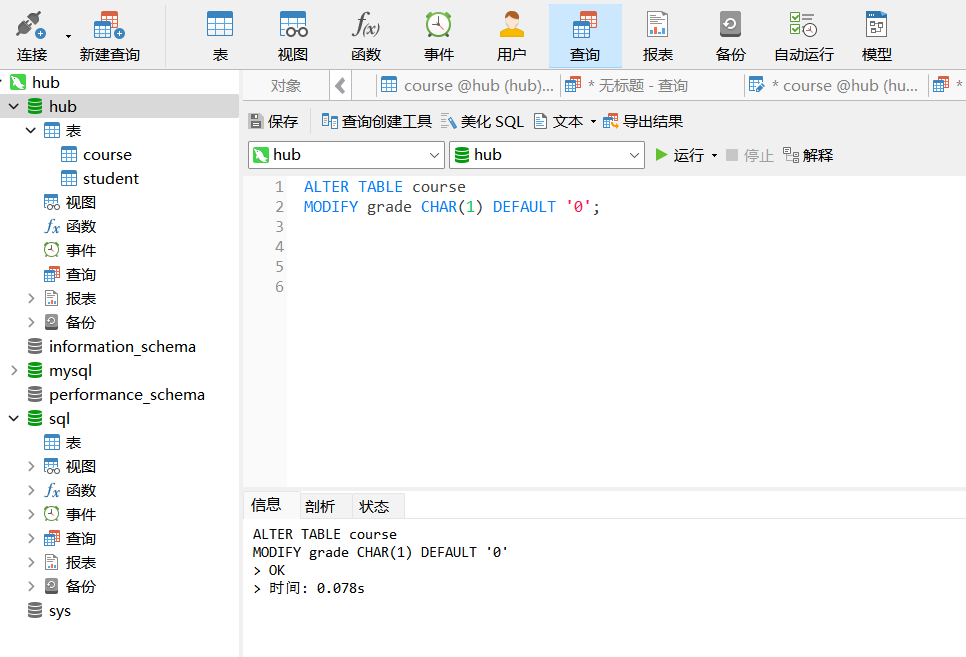


## 利用Transact-SQL语言修改course表

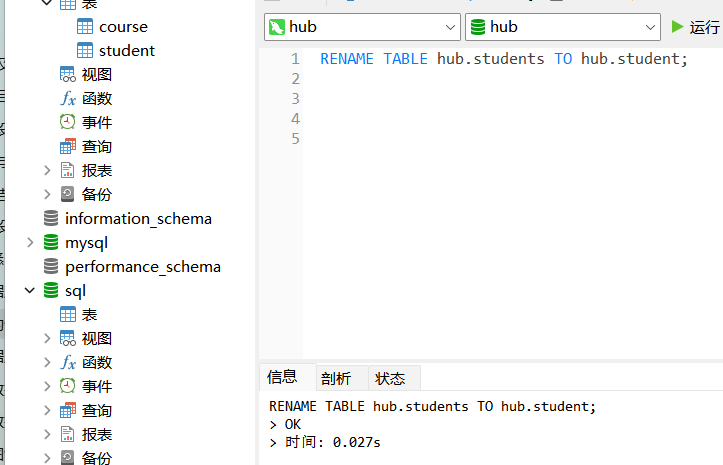
①增加“成绩”一列grade，int型，允许为空值，默认为0。



②修改grade列的类型为char。



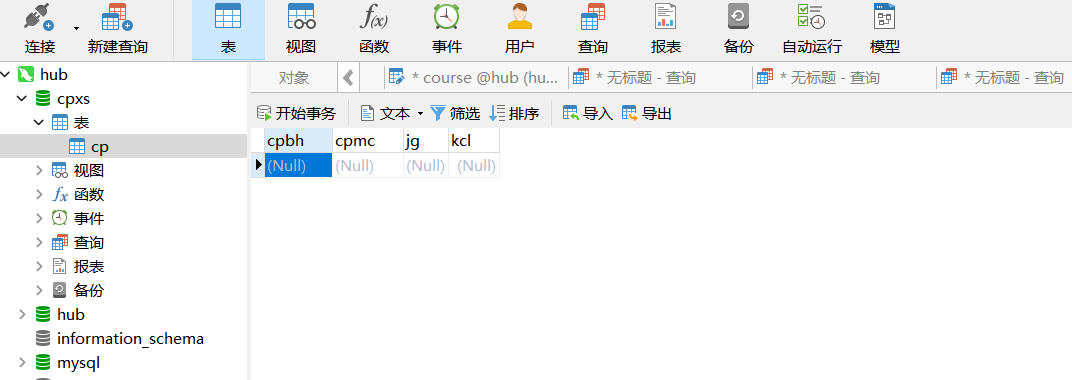
## 利用Transact-SQL将表students重命名为student



# 三、课后练习题

1、创建一个名为cpxs的数据库，在该数据库中分别通过Navicat和查询分析器创建一个名为cp的表（产品表），表格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 |
| cpbh | 产品编号 | char | 6 | no | 主码 |
| cpmc | 产品名称 | char | 30 | no |  |
| jg | 价格 | float |  | yes |  |
| kcl | 库存量 | int |  | yes |  |



2、思考以下问题：

1）如果要在一个已经存在的表格上删除一列并增加一列，能不能用一个ALTER TABLE语句来完成？

可以使用 一个 ALTER TABLE 语句 来同时删除一列和增加一列。

ALTER TABLE 语句允许你同时执行多个操作，包括删除列和增加列。可以将 DROP COLUMN 和 ADD 操作放在同一个 ALTER TABLE 语句中。

**实验总结：**

在本次实验中，我主要通过 Transact-SQL 语言对数据库进行操作，完成了以下任务：

创建数据库与表：首先，我在 SQL Server 中创建了一个名为 cpxs 的数据库，并在该数据库中创建了 cp 表。表中包含了产品编号（cpbh）、产品名称（cpmc）、价格（jg）和库存量（kcl）四个属性，并对 cpbh 设置为主键。通过 Navicat 和 查询分析器 两种方式实现了表的创建，了解了不同工具下的操作流程。

修改表结构：在创建了表之后，我尝试了修改表结构的操作。首先，我成功地使用 ALTER TABLE 语句增加了 grade 列，并为其设置了默认值 0。接着，我将 grade 列的数据类型从 INT 修改为 CHAR(1)，并设置默认值为字符 '0'。这些操作加深了我对 SQL 数据类型和列修改操作的理解。

删除列和添加新列：在后续的实验中，我通过 ALTER TABLE 语句实现了同时删除一列（如 kcl）并添加新列（如 description）。这不仅提高了我对 ALTER TABLE 语句的理解，也让我体会到在实际项目中，如何灵活操作表结构以应对需求变化。

**教师评语及成绩**：