**安徽科技学院**

**软件类实验报告**

|  |  |
| --- | --- |
| 课程名称： | 数据库原理与应用课程设计 |
| 学 号： | 27002200209 |
| 姓 名： | 乔梁 |
| 班 级： | 计算机科学与技术204班 |
| 指导教师： | 李文才 |
| 开课学期： | 2022-2023-1 |
| 学 院： | 信息与网络工程学院 |

安徽科技学院教务处

## 对用户需求分析；

#### 一、收集信息

与该系统有关人员进行交流、座谈，充分了解用户需求，理解数据库需要完成的任务。

#### 二、标识实体（Entity）

标识数据库要管理的关键对象或实体，实体一般是名词。

#### 三、标识每个实体的属性（Attribute）

为后面分享ER图和规范化理论打下基础。

#### 四、标识实体之间的关系（RelationShip）

理清关系后才可以更好的设计数据表和ER图。

## 掌握规范化理论的使用方法；

使用范式实现数据库设计理论的规范化，不规范的表设计可能会出现诸多问题。

①、信息重复

②、更新异常

③、插入异常：无法正确表示信息

④、删除异常：丢失有效信息

一、第一范式的目标是确保每列的原子性

如果每列都是不可再分的最小数据单元，也称为最小的原子单元，则满足第一范式（1NF）

**一遍来说根据实际做出合适的切分即可，举例的话对地名要分割的开，即保持原子性，不可再分。**

二、第二范式要求每张表只描述一件事情（2NF）

**用自己的话来说第二范式就是属性之间不存在部分函数依赖，即表中其他数据元素都依赖主键。说的一件事情就是每张表仅依赖一个主键。**

三、如果一个关系满足2NF，并且除了主键以外的其他列都不传递依赖于主键列，则满足第三范式（3NF）

**非主键列必须直接依赖于主键列，不能存在传递依赖。即不能存在类似非主键列A依赖于主键列B，主键列B又依赖于主键A。**

**是二范式且非主属性对主属性没有传递函数依赖。**

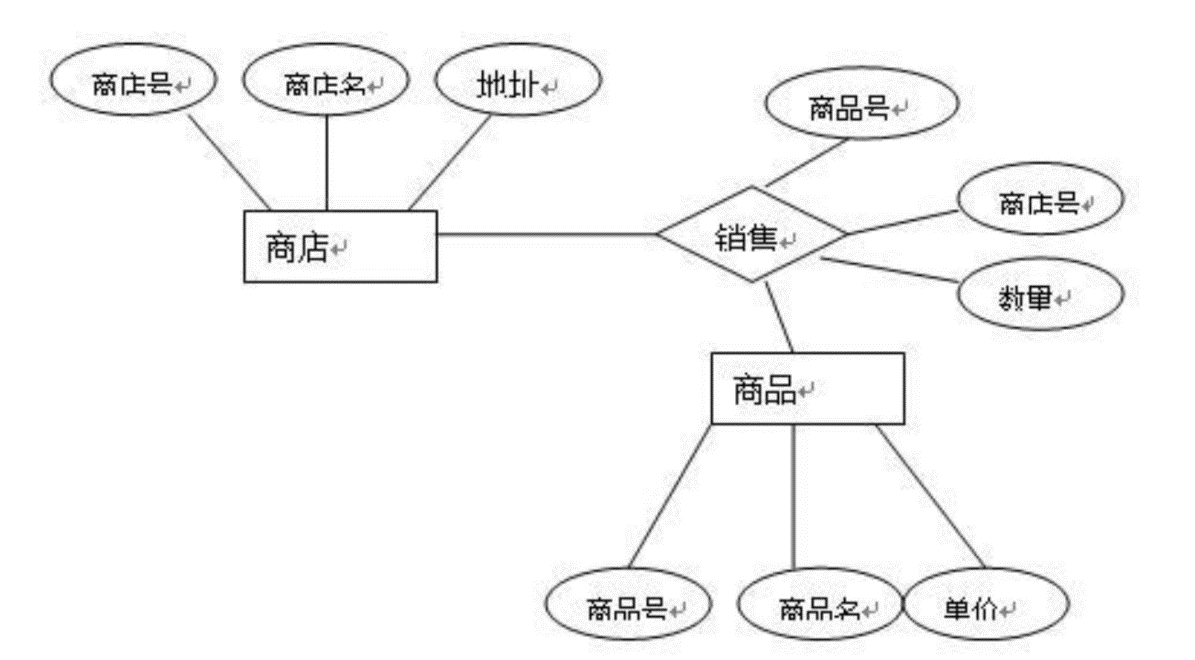
**四、BCNF范式**

**所有属性关系的左侧都是候选码。**

## 掌握E-R图的设计方法

一、E-R图的符号说明

|  |  |
| --- | --- |
| **符号** | **含义** |
|  | 实体，一般是名词 |
|  | 属性，一般是名词 |
|  | 关系，一般是动词 |



## 掌握数据表逻辑结构的设计；

## **第一步:将各个实体的名字转换为各个关系模式的名字**

## **第二步:实体的属性就是关系的属性，实体的码就是关系的码**

## **第三步:实体间联系的转换**

## **1对1联系:在任意一方加入对方的主码并设为其外码，并加入联系本身的属性**

## **1对多联系:将|方的主码加入n方作为外码，并同时将联系的属性加入n方**

## **n对m联系:将联系本身转换为一个关系模式，将联系双方的主码加入其中设为码，并将联系的属性也加入其中**

## 掌握数据表物理结构的设计方法；

数据库的物理设计以逻辑设计的结果作为输入，结合具体DBMS的特点与存储设备特性

进行设计，对于给定的逻辑数据模型，选取一个最适合应用环境的物理结构。

数据库的物理设计分为两个部分，首先是确定数据库的物理结构，在关系数据库中主要指数据的存取方法和存储结构。其次是对所设计的物理结构进行评价，评价的重点是系统的时间和空间效率。如果评价结果满足原设计要求， 则可以进入到物理实施阶段，否则需要

重新设计或修改物理结构，有时甚至要返回到逻辑设计阶段修改数据模型。

## 掌握数据字典的使用的方法；

数据字典( data dictionary )是Oracle 数据库的一个重要组成部分,这是一组

用于记录数据库信息的只读( read-only )表。

数据字典里存有用户信息、用户的权限信息、所有数据对象信息、表的约束条件、统计分析数据库的视图等。我们不能手工修改数据字典里的信息。当用户在对数据库中的数据进行操作时遇到困难就可以访问数据字典来查看详细的信息。

字典中信息包含数据库中所有方案对象定义，分配的空间，当前使用的空间，列的默认值，完整性约束，数据库用户名，授予权限的角色，监控信息和其他概要。

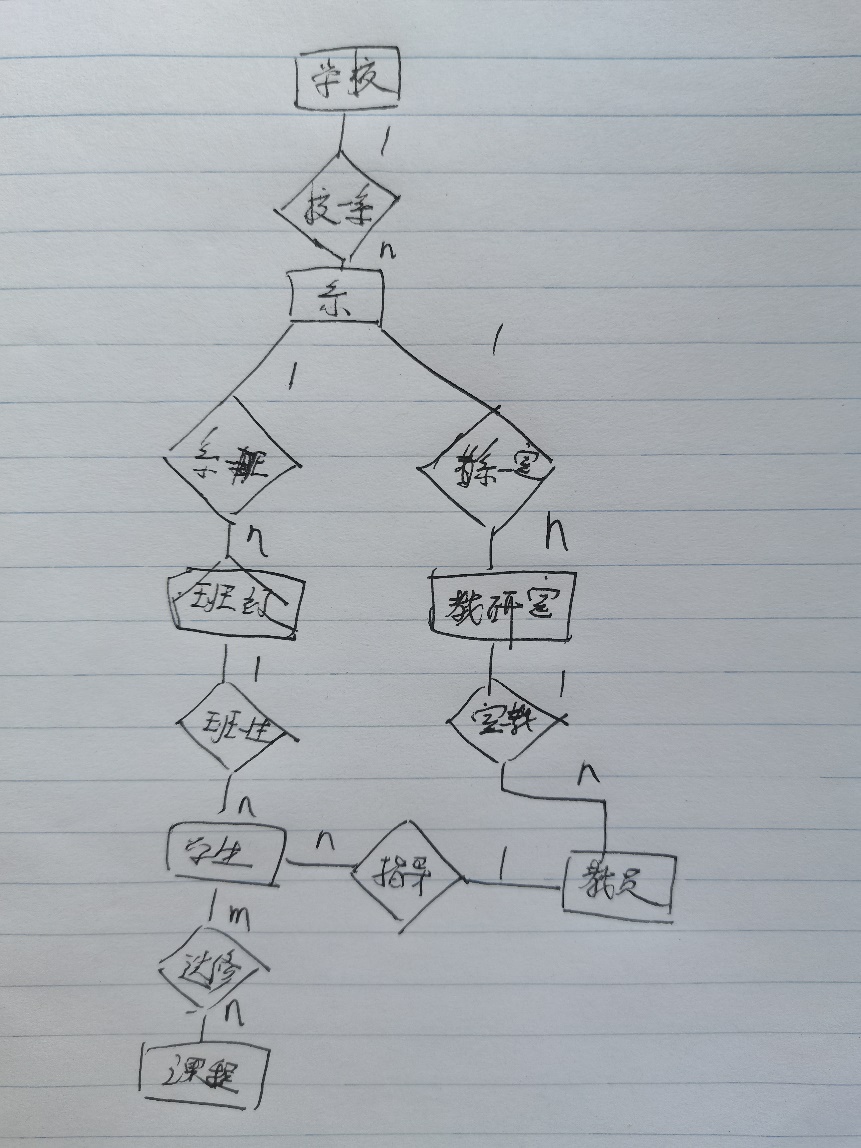
## 锻炼解决实际问题的能力。

实际数据库设计

7.学校中有若干系,每个系有若干班级和教研室, 每个教研室有若干教员,其中有的教授和副教授每人各带若干研究生,每个班有若干学生,每个学生选修若干课程,每门课可由若

干学生选修。

首先在需求分享和关系考虑后可以画出ER图。



系(系编号,系名,学校名)

班级(班级编号,班级名,系编号)

教研室(教研室编号,教研室，系编号)

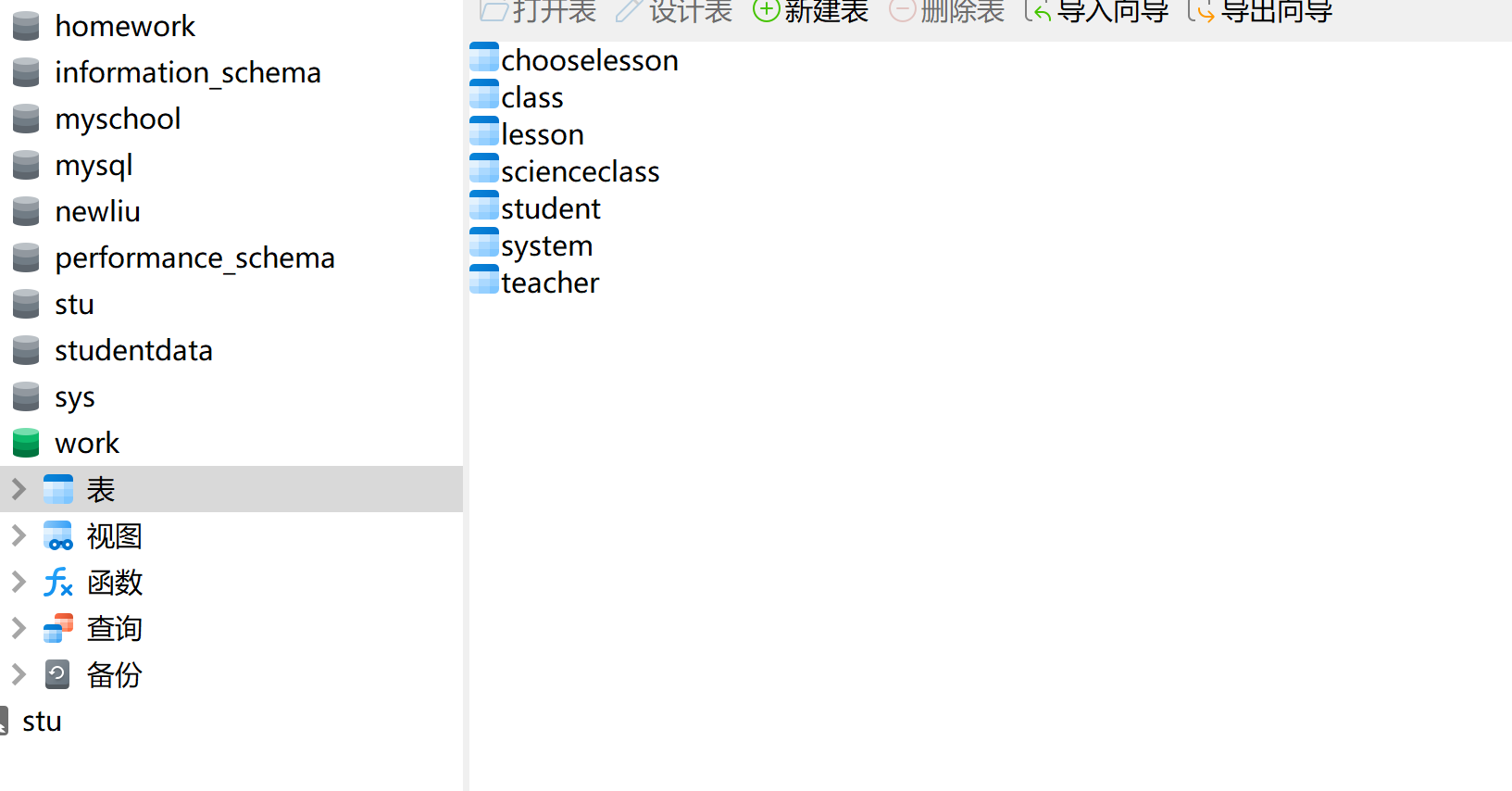
学生(学号,姓名,学历,班级编号,导师职工号)

课程(课程编号,课程名)

教员(职工号,姓名,职称,教研室编号)

选课(学号,课程编号,成绩)

通过Navicat创建表。



**安徽科技学院**

**软件类实验报告**

|  |  |
| --- | --- |
| 课程名称： | 数据库原理与应用课程设计 |
| 学 号： | 27002200209 |
| 姓 名： | 乔梁 |
| 班 级： | 计算机科学与技术204班 |
| 指导教师： | 李文才 |
| 开课学期： | 2022-2023-1 |
| 学 院： | 信息与网络工程学院 |

安徽科技学院教务处

## 以提供的数据库为例。

## 掌握数据库和数据表的设计方法；

在模式中定义表

一个模式包含多种基本表，有三种方式在模式中定义基本表。

(1)创建表时指出模式

CREATE TABLE模式名.表名

(列定义语句，

完整性约束语句

);

(2)创建模式时直接定义表

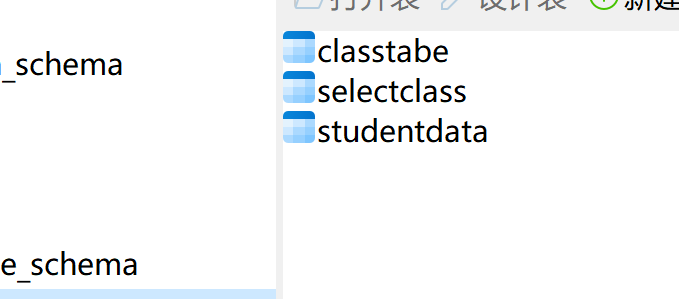
CREATE SCHEMA模式名AUTHORIZATION用户名

CREATE TABLE表名

(列定义语句，

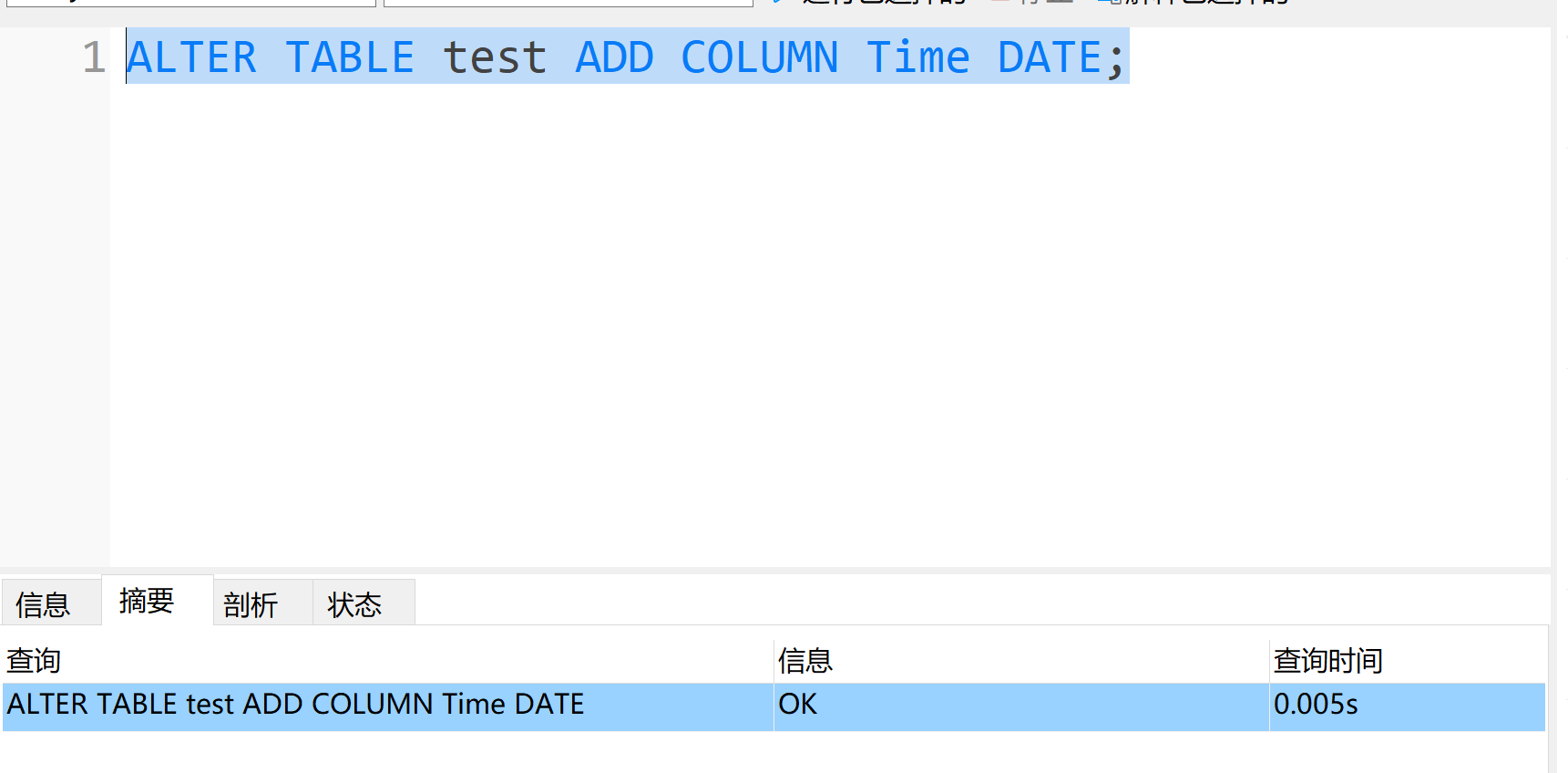
完整性约束语句）；

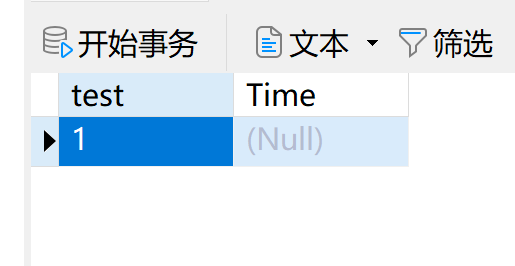
正常使用的时候用图形界面建表就可以了…



## 掌握表结构的修改方法；

**ALTER TABLE test ADD COLUMN Time DATE;**

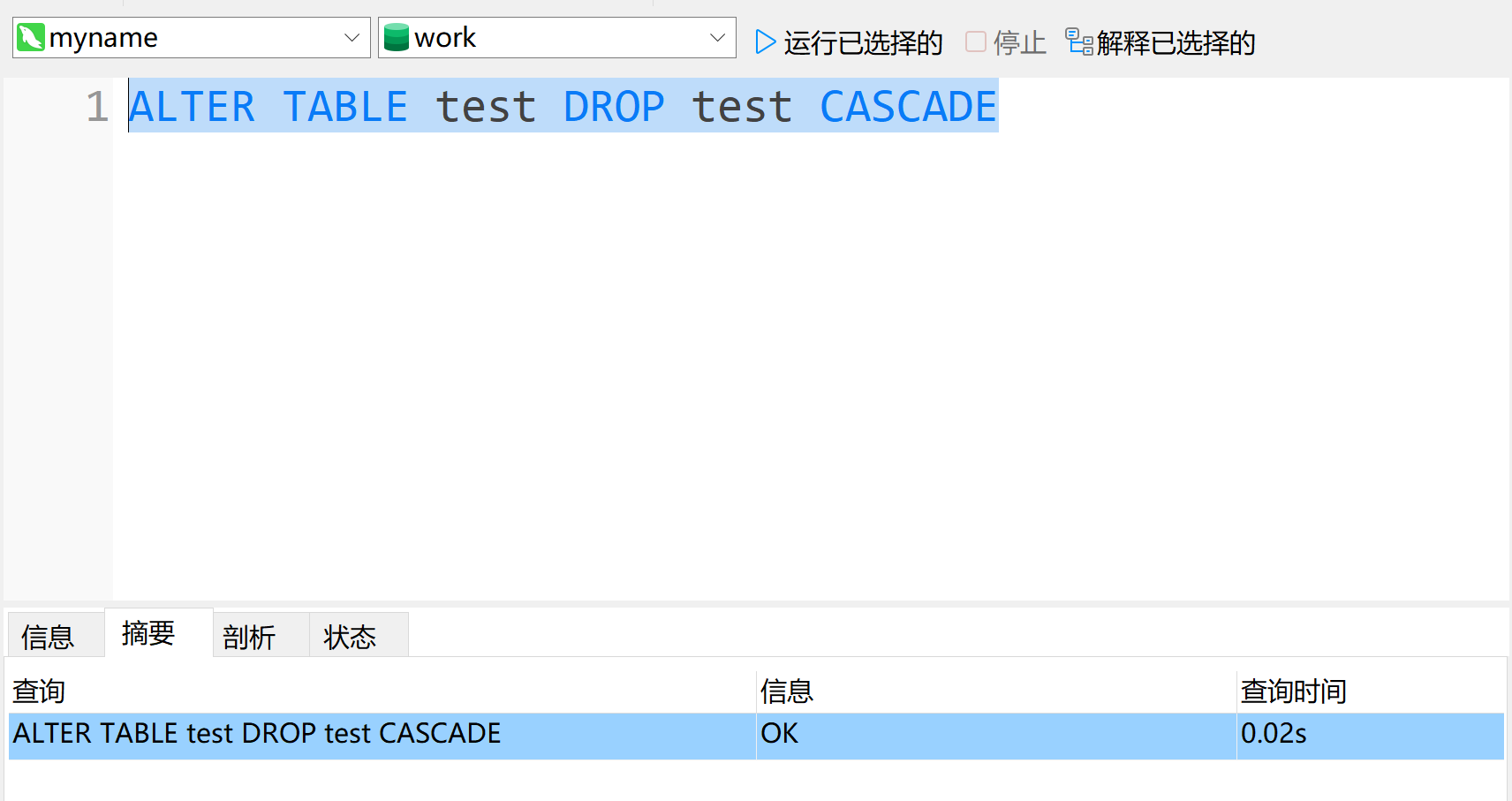
****

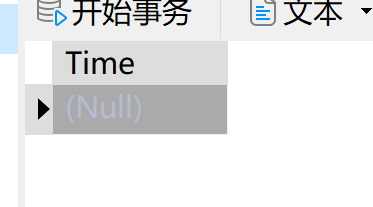
****

**向test表中添加Time列，类型为时间型。**

**ALTER TABLE表名DROP列名CASCADE; //级联:引用了该列的其他对象(例如视图)一起删除**

**ALTER TABLE test DROP test CASCADE**

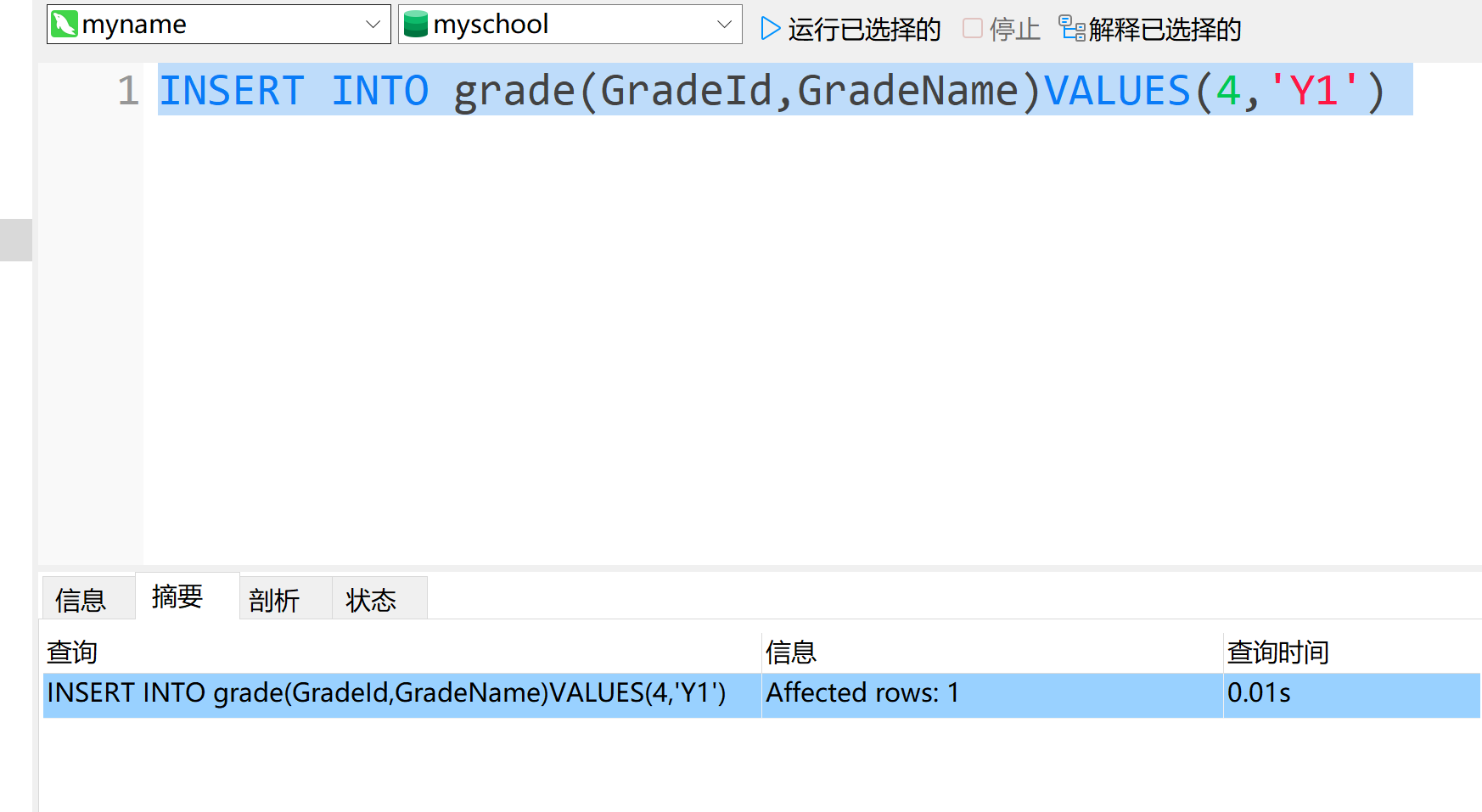
****

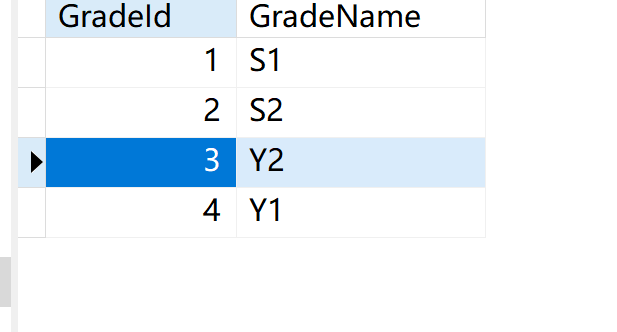
****

**ALTER TABLE表名DROP列名RESTRICT; //限制:若该列被其他对象引用，则拒绝删除**

## 掌握记录的增删改查方法

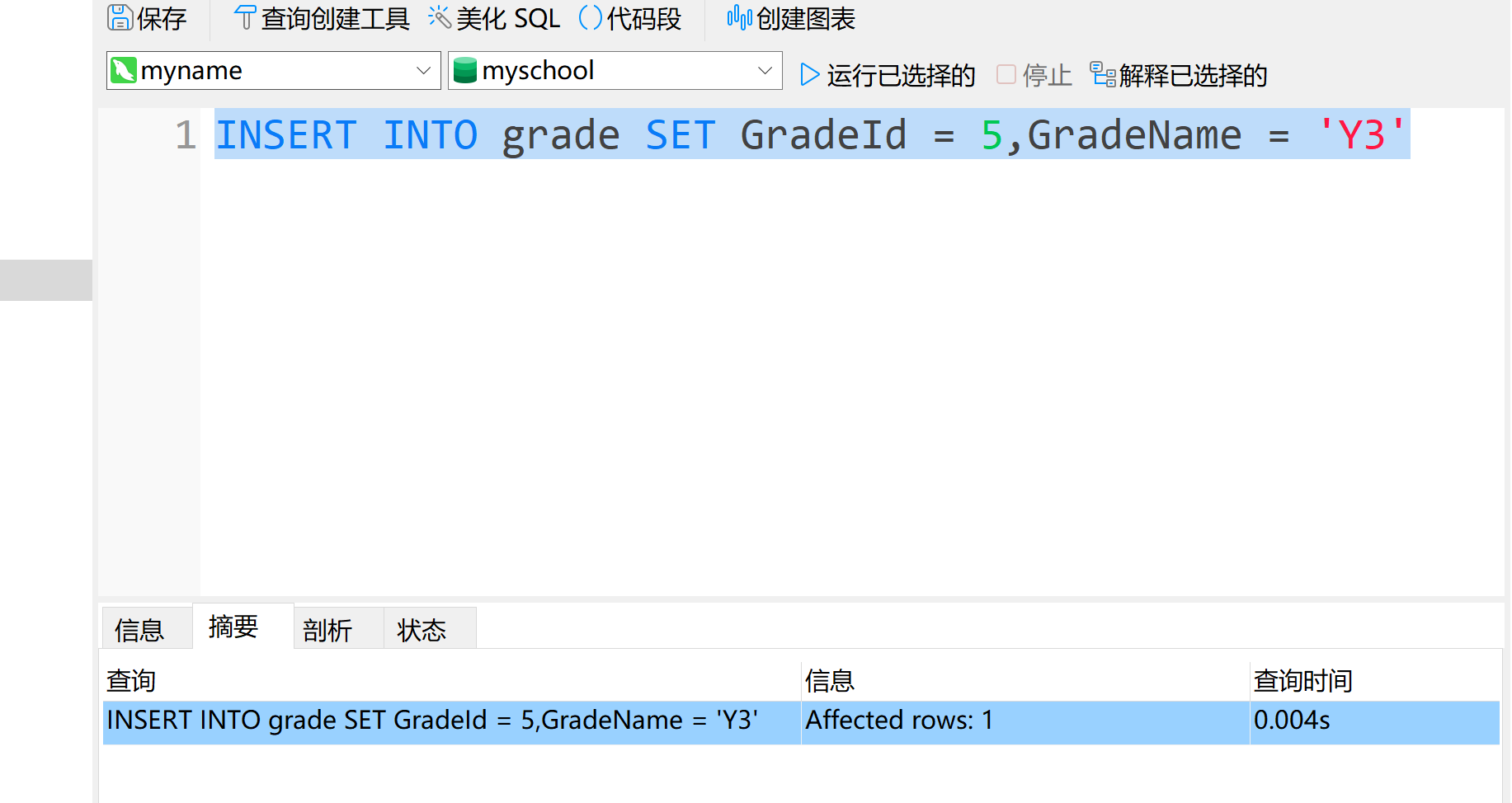
INSERT INTO grade(GradeId,GradeName)VALUES(4,'Y1')

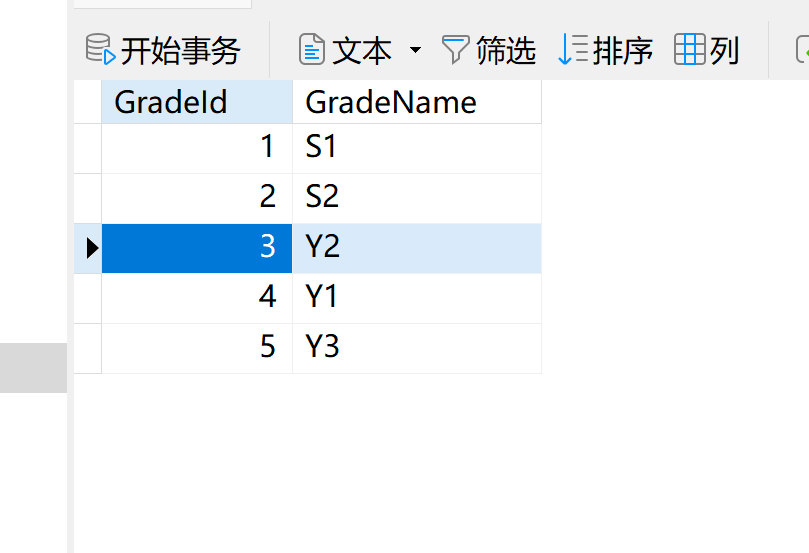




根据补充，另外还可以：

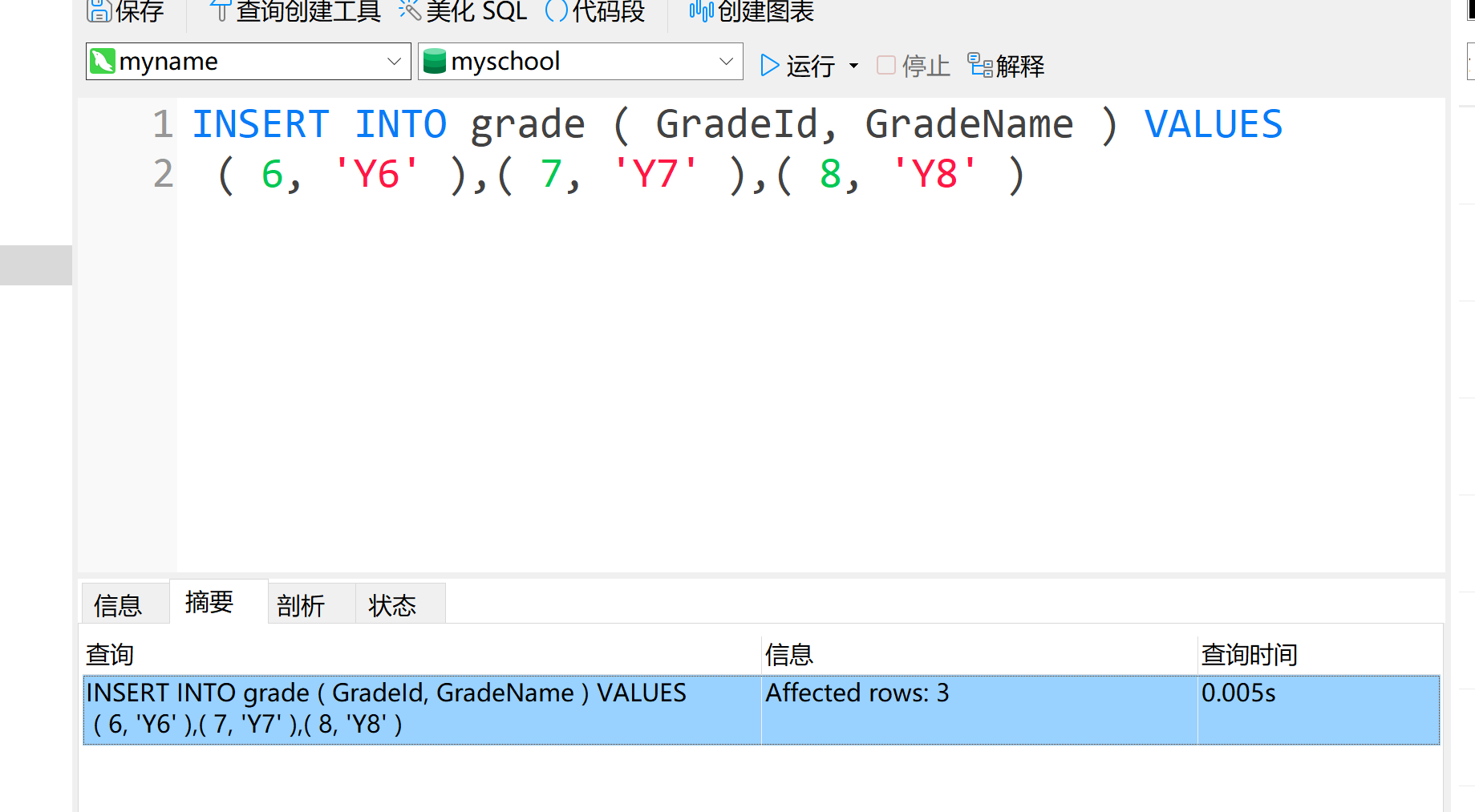
INSERT INTO grade SET GradeId = 5,GradeName = 'Y3'

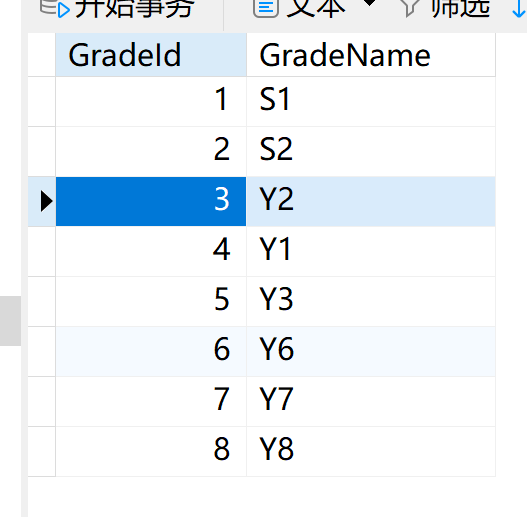




同样还有一次插入多行：

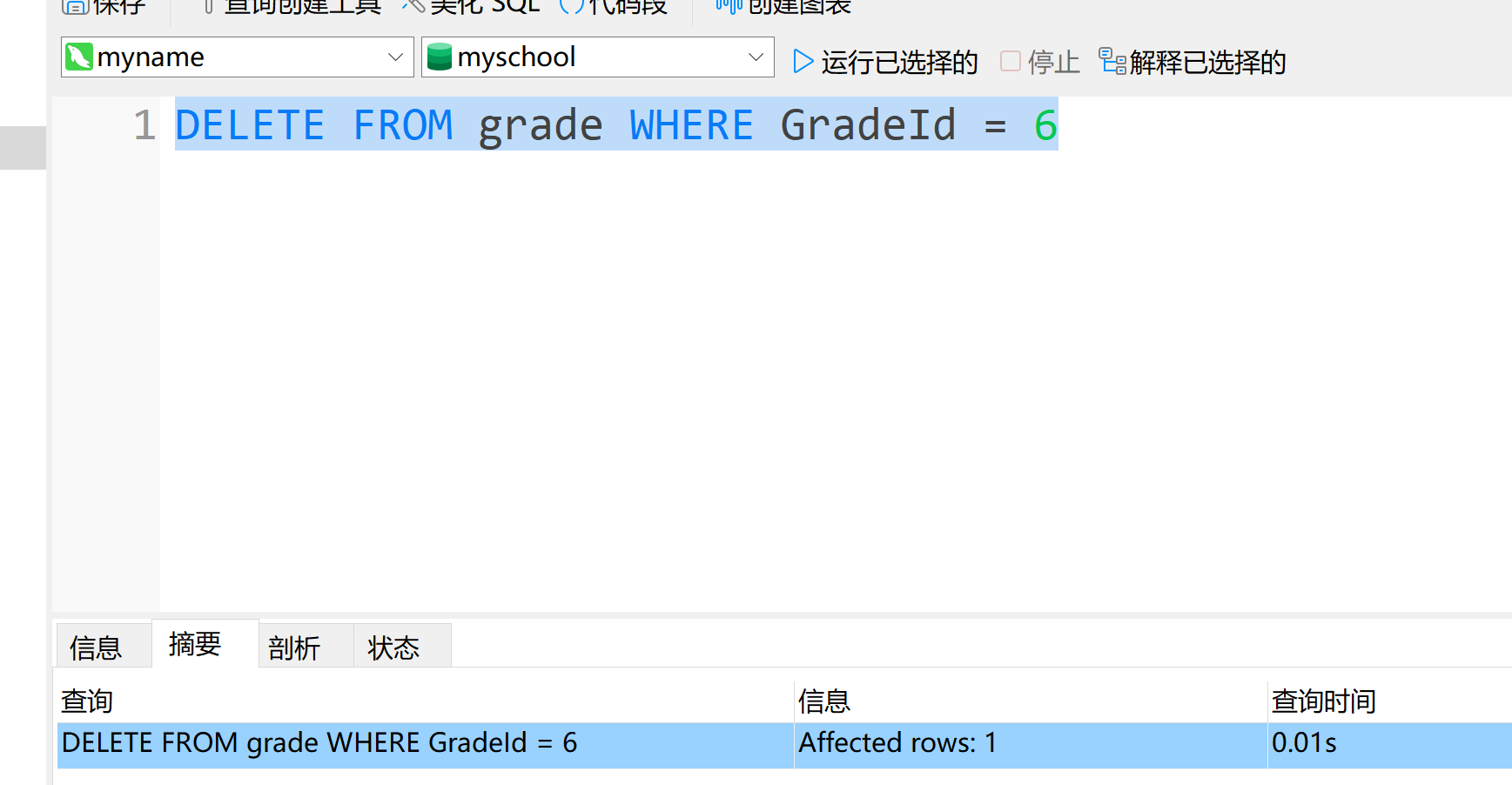
INSERT INTO grade ( GradeId, GradeName ) VALUES

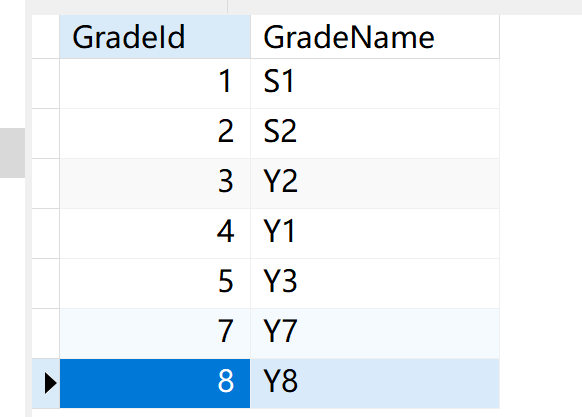
( 6, 'Y6' ),( 7, 'Y7' ),( 8, 'Y8' )



删除指定数据

DELETE FROM grade WHERE GradeId = 6





## 掌握存储过程的设计和使用方法；

没有参数的存储过程设计

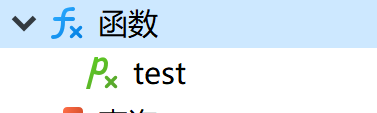
CREATE PROCEDURE test ()

BEGIN

INSERT INTO grade ( GradeId, GradeName ) VALUES ( 10, 'Y10' ),( 8, 'Y8' );

END





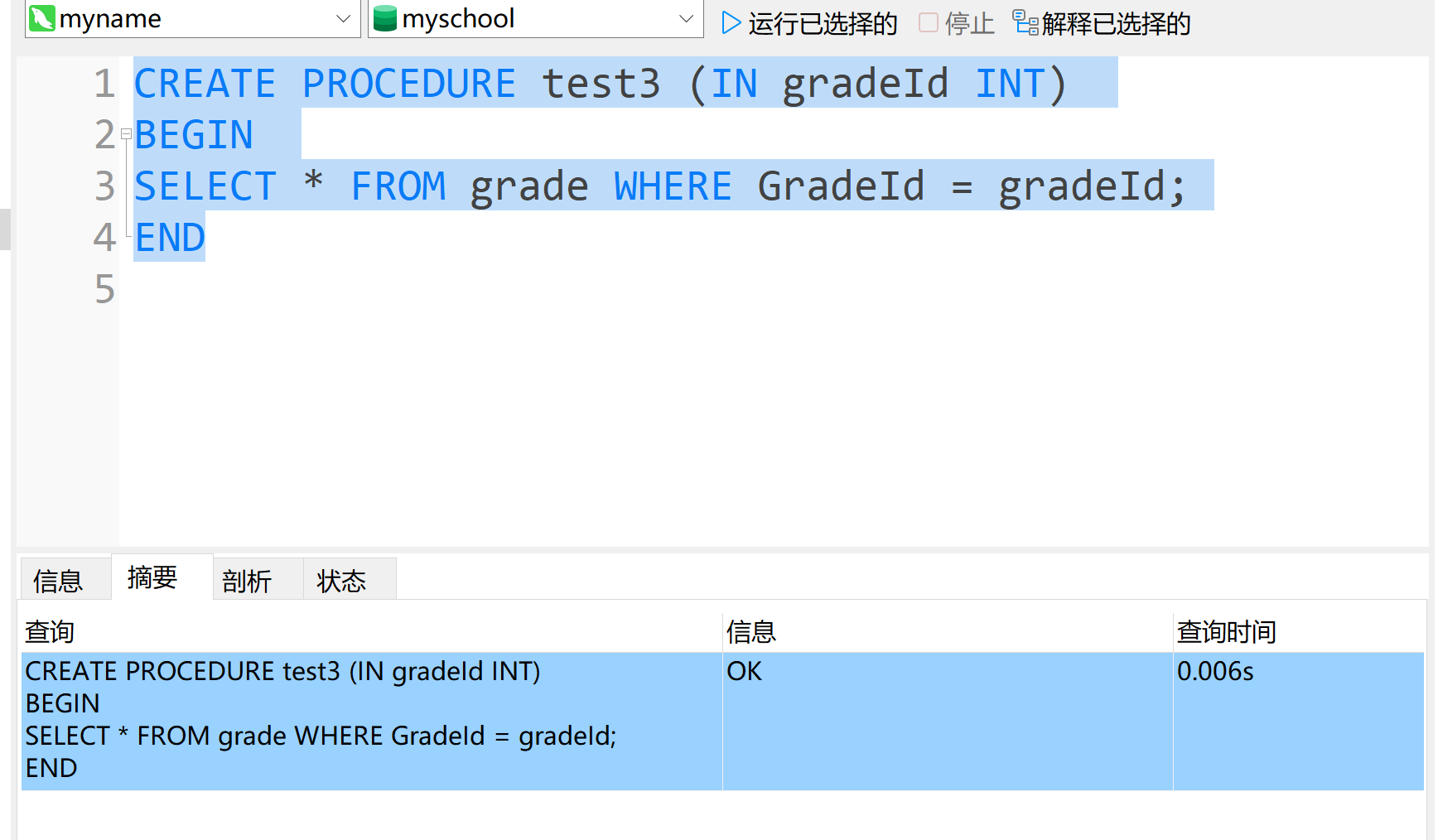
带IN参数的存储过程:

CREATE PROCEDURE test3 (IN gradeId INT)

BEGIN

SELECT \* FROM grade WHERE GradeId = gradeId;

END



尝试调用这个TEST3



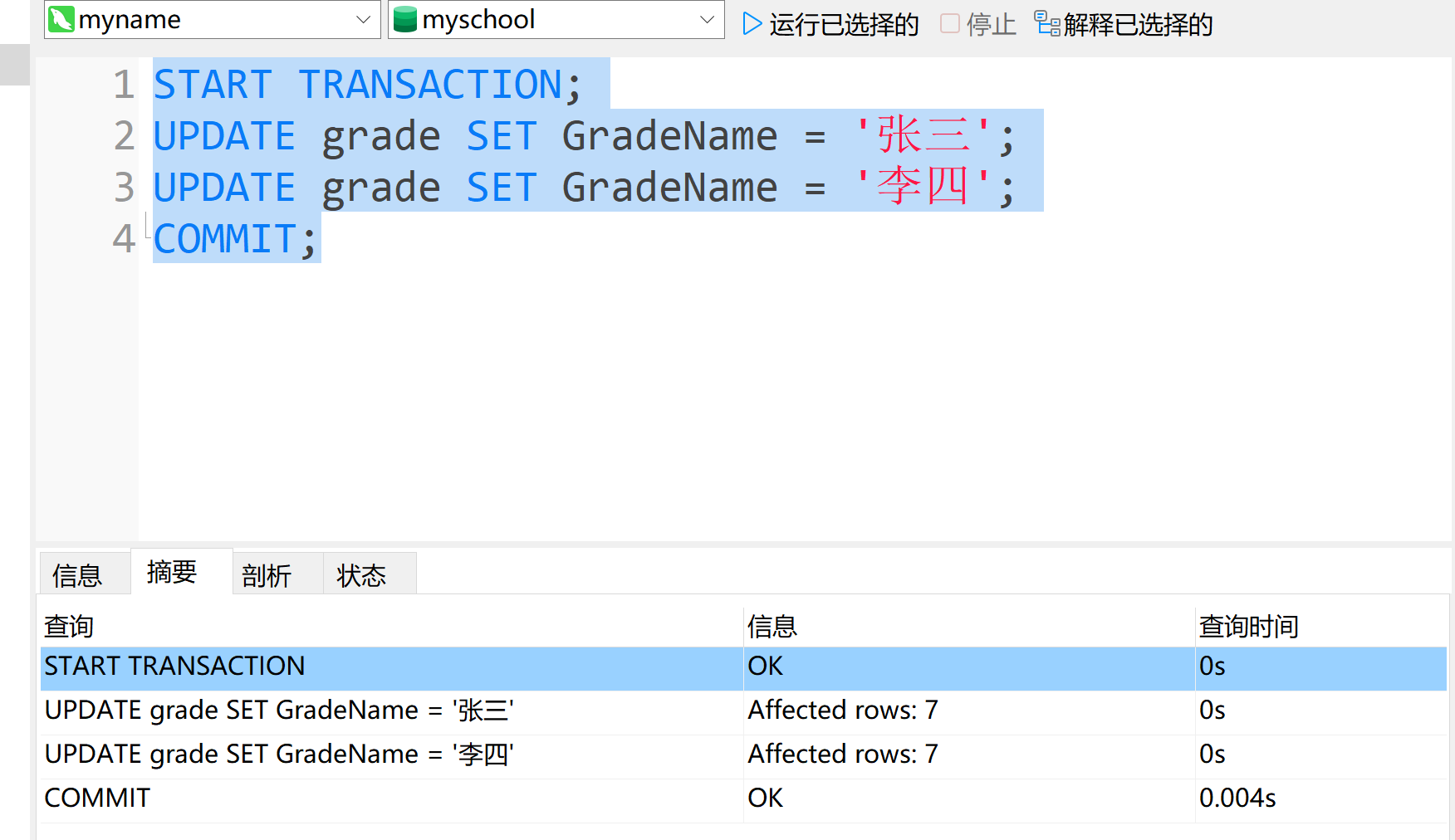
## 掌握事务的提交和回滚方法。

多条查询语句可以作为事务。

START TRANSACTION;

UPDATE grade SET GradeName = '张三';

UPDATE grade SET GradeName = '李四';

COMMIT;

修改最后一句可以做到回滚

START TRANSACTION;

UPDATE grade SET GradeName = '张三';

UPDATE grade SET GradeName = '李四';

ROLLBACK;

