

实验 1、数据定义

1.1 实验目的

熟悉 SQL 的数据定义语言,能够熟练地使用 SQL 语句来创建和更改基本表,创建和取消索引。

1.2 实验内容

- 在 SSMS 环境中使用 SQL 语言创建数据库 chenht_University_Mis
- 使用 CREATE 语句创建基本表。
- 更改基本表的定义: 增加列, 删除列, 修改列的数据类型。
- 创建表的升、降序索引。
- 删除基本表的约束、基本表的索引或基本表。

1.3 实验步骤

1) 在 SSMS 环境中使用 SQL 语言创建数据库 chenht_University_Mis

2) 使用 SQL 语句创建关系数据库基本表:

学 生 表 chenht_Students(cht_Sno,cht_Sname,cht_Semail,cht_Scredit,cht_Sroom);

教师表 chenht_Teachers(cht_Tno,cht_Tname,cht_Temail,cht_Tsalary);

课程表 chenht_Courses(cht_Cno,cht_Cname,cht_Ccredit);

成绩表 chenht_Reports(cht_Sno,cht_Tno,cht_Cno,cht_Score);

其中: cht_Sno、cht_Tno、cht_Cno 分别是表 Chenht_Students、表 Chenht_Teachers、表 Chenht_Courses 的主键,具有唯一性约束,cht_Scredit 具有约束“大于等于 0”; Chenht_Reports 中的 cht_Sno,cht_Tno,cht_Cno 是外键,它们共同组成 Chenht_Reports 的主键。

3) 更改表 Chenht_Students: 增加属性 cht_Ssex(类型是 CHAR,长度为 2),取消 cht_Scredit “大于等于 0” 约束。把表 Chenht_Courses 中的属性 cht_Cname 的数据类型改成长度为 30。

4) 删除表 Chenht_Students 的一个属性 cht_Sroom。

5) 删除表 Chenht_Reports。

6) 为 Chenht_Courses 表创建按 cht_Cno 降序排列的索引。

7) 为 Chenht_Students 表创建按 cht_Sno 升序排列的索引。

8) 创建表 Chenht_Students 的按 cht_Sname 升序排列的唯一性索引。

9) 删除 Chenht_Students 表 cht_Sno 的升序索引。

1.4 实验结果

1) 在 SSMS 环境中使用 SQL 语言创建数据库 chenht_university_mis

启动 SQL Server Management Studio, 连接到服务器, 然后单击菜单栏中的

“新建查询”，在出现的任务页面中输入创建数据库的 SQL 命令，创建名为 chenht_university_mis 的数据库。

SQL 代码:

```
use master
go

/*
 * create database [chenht_student_mis]
 */
create database chenht_university_mis
on (
    name = chenht_university_mis_data,
    filename = 'C:\Users\t1542\Program\SQL\chenht_university_mis_
data.mdf',
    size = 10,
    maxsize = 50,
    filegrowth = 5
)
log on (
    name = chenht_university_mis_log,
    filename = 'C:\Users\t1542\Program\SQL\chenht_university_mis_
data.ldf',
    size = 5mb,
    maxsize = 25mb,
    filegrowth = 5mb
)
go

alter database chenht_university_mid collate Chinese_PRC_CI_AS
```

实验结果如图 1-1.

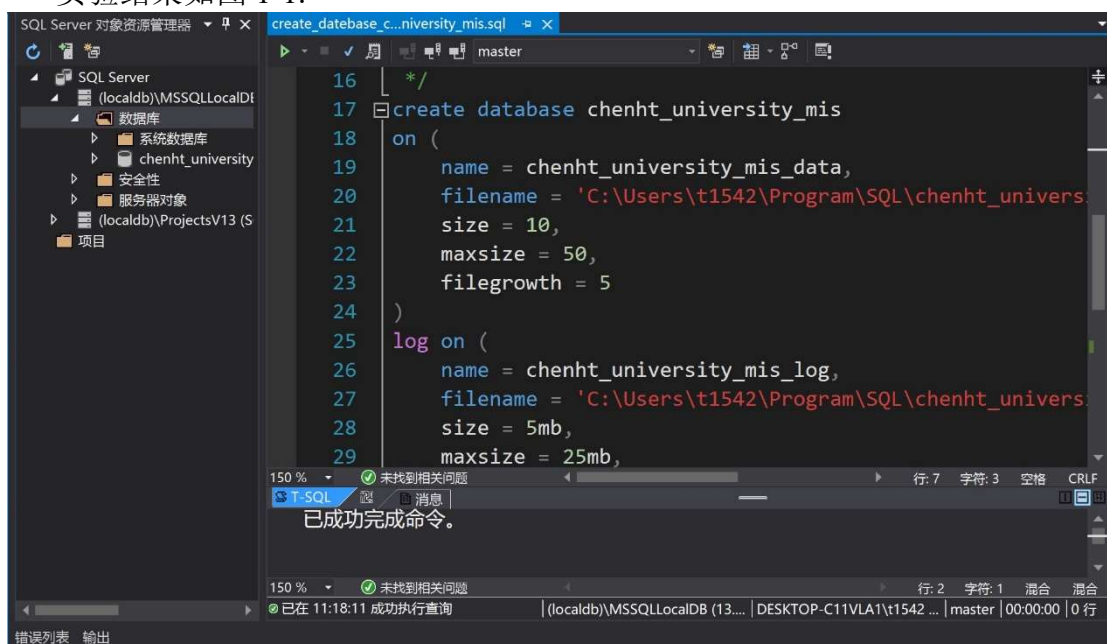


图 1-1 创建数据库 Chenht_University_Mis 的结果

2) 用 SQL 语言创建学生表

学 生 表 Chenht_Students(cht_Sno, cht_Sname, cht_Semail, cht_Scredit, cht_Sroom);

SQL 代码

```

/*
 * create table [chenht_Students]
 */
create table chenht_Students(
    cht_sno char(5) primary key,
    cht_sname char(10) not null,
    cht_semail char(50),
    cht_scredit int default(0)
    constraint cht_check_scredit check(cht_scredit>=0),
    cht_sroom char(5) not null,
)
.....

```

实验结果如图 1-2

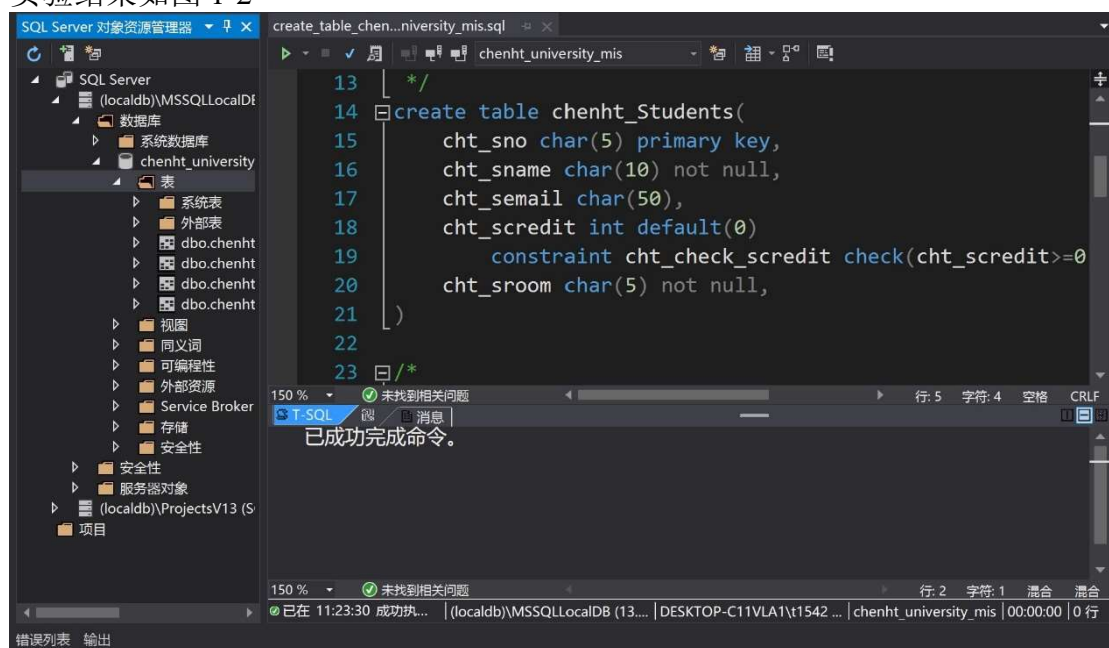


图 1-2 创建基本表 Chenht_Students 的命令和执行结果

3) 更改表 Chenht_Students: 增加属性 cht_Ssex(类型是 CHAR, 长度为 2), 取消 cht_Scredit “大于等于 0” 约束。把表 Chenht_Courses 中的属性 cht_Cname 的数据类型改成长度为 30。

SQL 代码

```

alter table chenht_Students
    add cht_ssex char(2) not null default('男')

alter table chenht_Students
    drop constraint cht_check_scredit

```

```
alter table chenht_Courses
    alter column cht_cname char(30) not null
```

实验结果如图 1-3

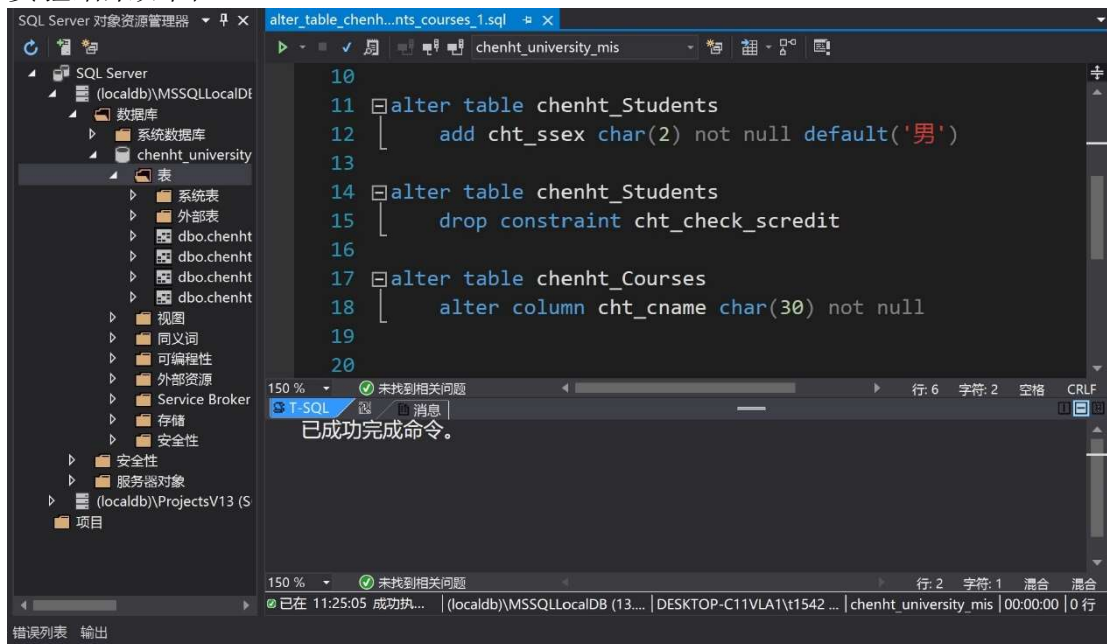


图 1-3 更改基本表 Students 和 Courses 的命令和执行结果

4) 删除表 Chenht_Students 的一个属性 cht_Sroom。

SQL 代码:

```
alter table chenht_Students
    drop column cht_sroom
```

实验结果如图 1-4:

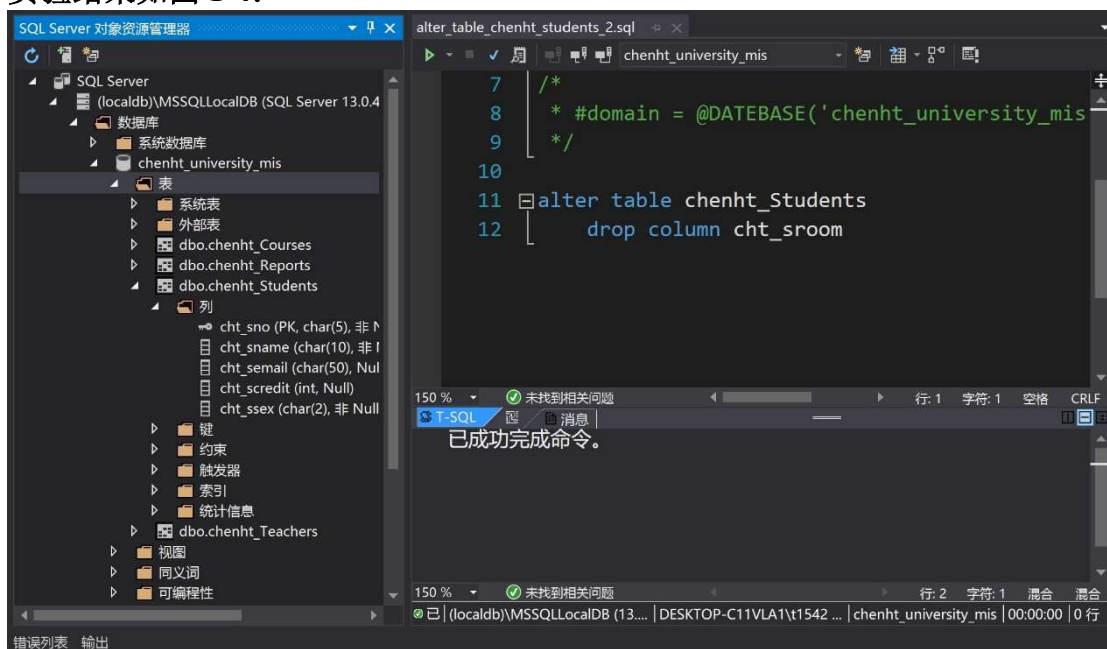


图 1-4 创建基本表 Chenht_Students 的属性 cht_Sroom 命令和执行结果

6) 为 Chenht_Courses 表创建按 cht_Cno 降序排列的索引。

SQL 代码:

```
create index cht_cno_desc on chenht_Courses
    (cht_cno desc)
```

实验结果如图 1-5

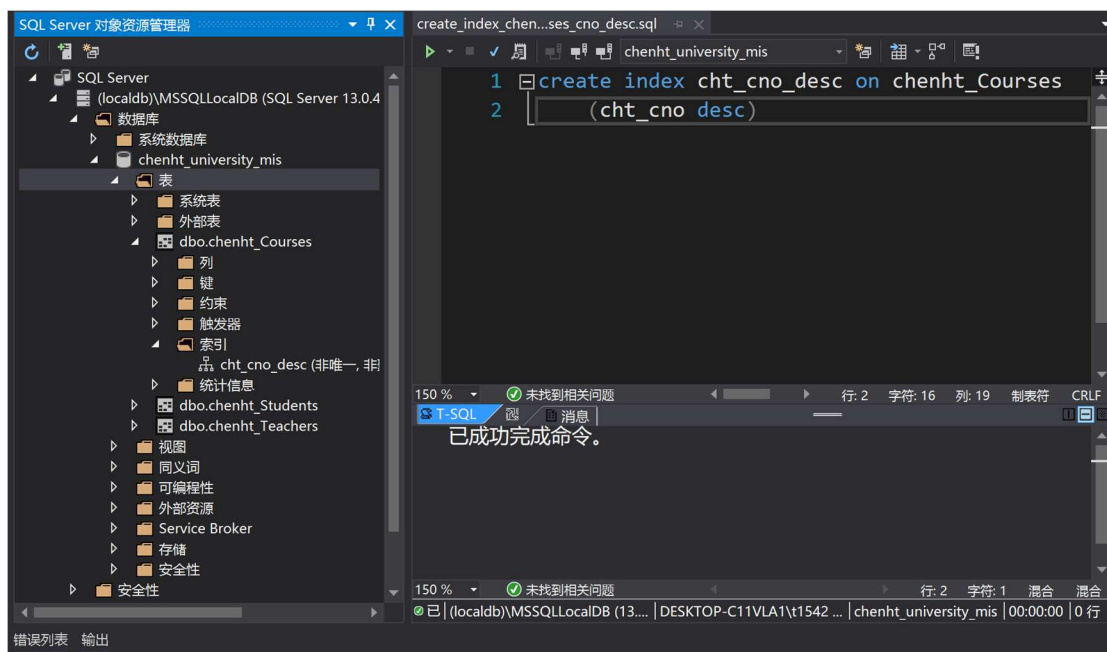


图 1-5 为 Chenht_Courses 表创建按 cht_Cno 降序排列的索引的命令和执行结果

9) 删除 Chenht_Students 表 cht_Sno 的升序索引。

SQL 代码

```
drop index chenht_Students.cht_sno_asc
```

实验结果如图 1-6

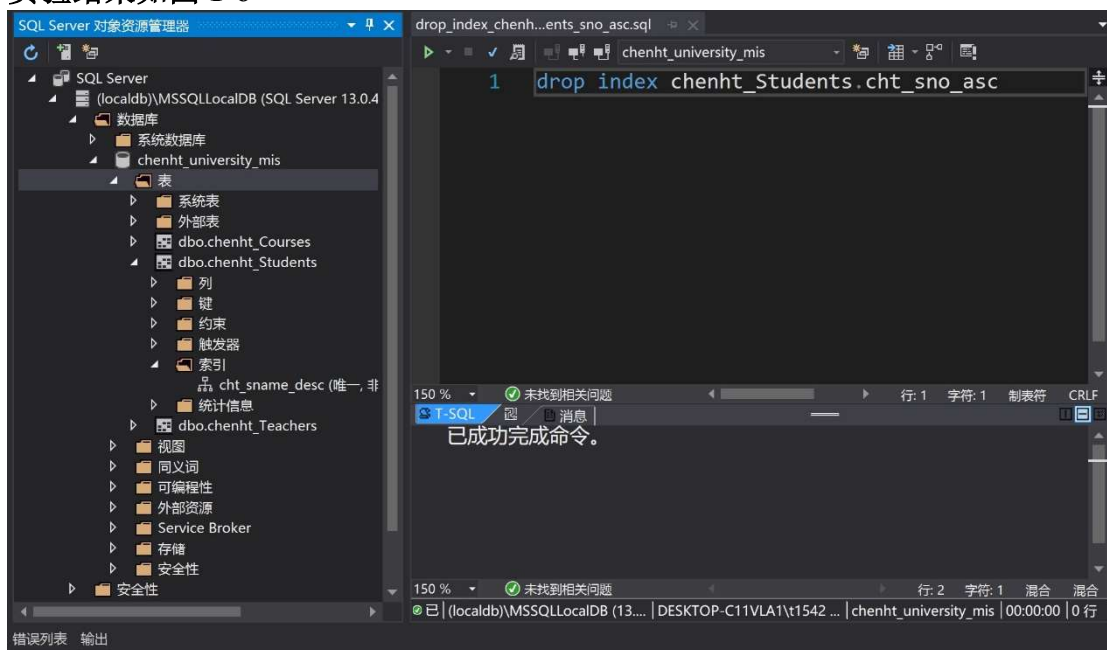


图 1-6 删除 Chenht_Students 表 cht_Sno 的升序索引的命令和执行结果

1.5 实验体会

1) 实验反思

- 在编写代码期间一不小心发生了拼写的错误，于是只能删除数据库重新再来。还好之前保存了完整的 SQL 代码，所以第二遍执行起来非常轻松了。
- 编写代码还不够快，有待改进。

2) 实验收获

- a. 学习了添加列的语法，发现其为 Add 而不是 Add Column，非常奇怪。
- b. 了解了普通索引，唯一索引和聚簇索引的联系与区别。