

实验一 环境介绍与安装使用 Sql2019、数据定义

实验目的：熟悉 SQL SERVER2019 的操作，实践表的基本操作

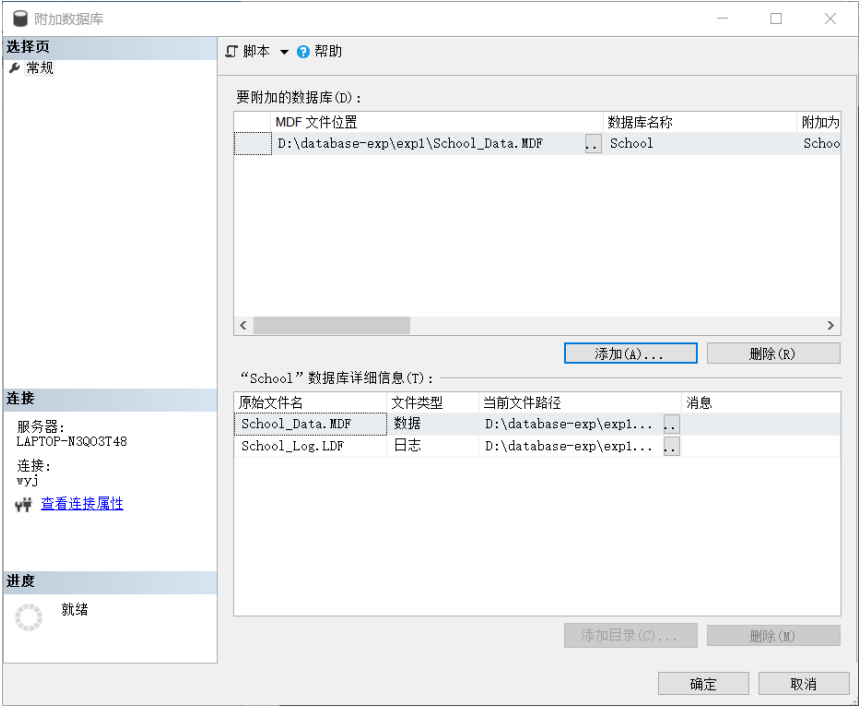
实验环境：WINDOWS10、SQL SERVER 2019

实验内容：

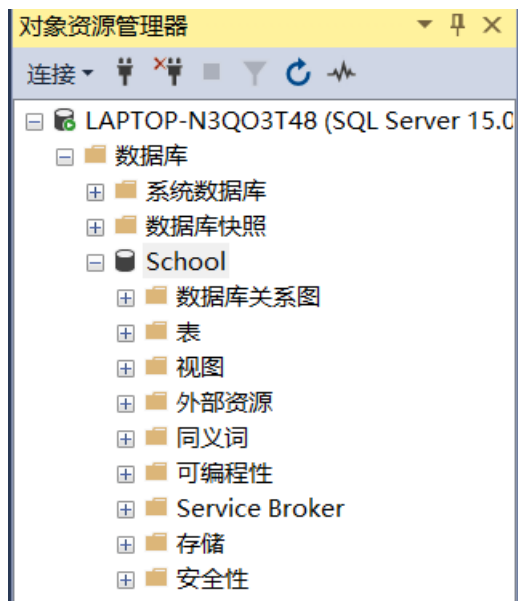
1.1 使用 SQL Server 工具(Microsoft SQL Server Management Studio Express) 管理数据库

(1) 加入 School 数据库。

在“数据库”上单击右键，选择“附加”，选择硬盘上的 School_Data.MDF 文件，然后单击确定，导入数据库。



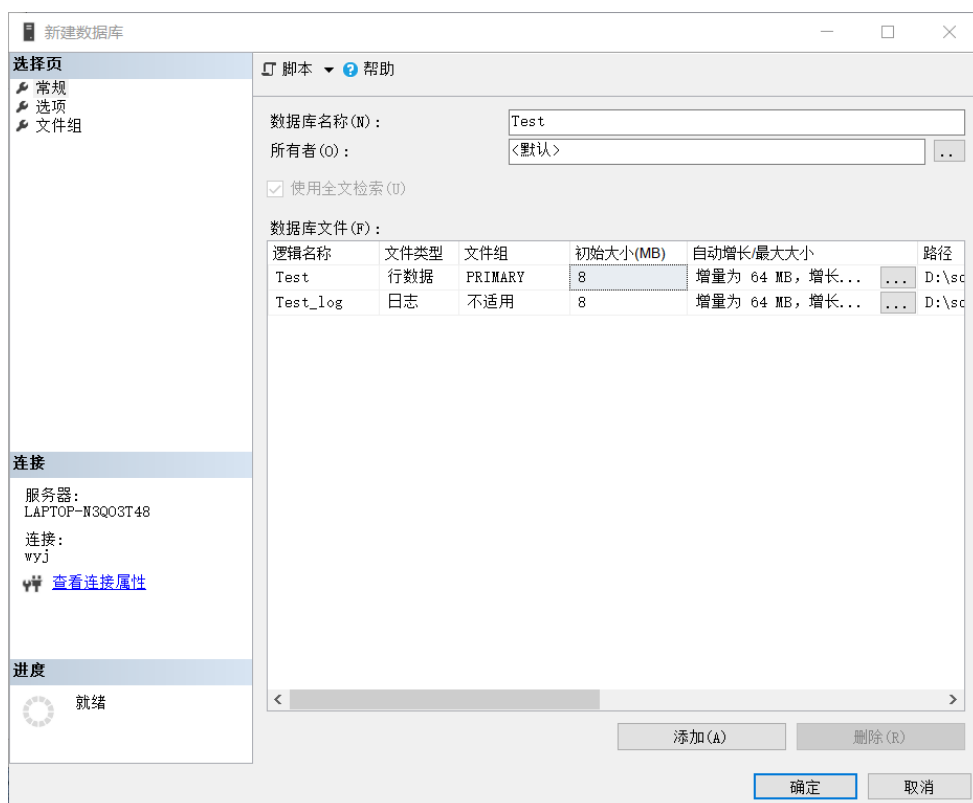
导入的数据库如下所示：



(2) 建立 Test 数据库。

在“数据库”上单击右键，选择“新建数据库”，输入新建数据库的名称“Test”。

然后单击“确定”，就可以新建数据库。



新建的数据库如下所示：



(3) 在数据库中建立人员表 PERSON(P#, Pname, Page)。更改表设置 P#为主键，增加属性 Ptype(类型是 CHAR, 长度是 10)。

在新建的数据库“Test”下的菜单项“表”上单击右键，选择“新建”，“表”，输入新建列的名称、数据类型。

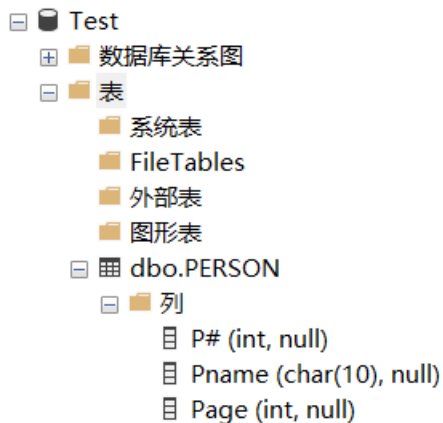
LAPTOP-N3QO3T48...t - dbo.Table_1* X			
列名	数据类型	允许 Null 值	
P#	int	<input checked="" type="checkbox"/>	
Pname	char(10)	<input checked="" type="checkbox"/>	
Page	int	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

保存后，在弹出的对话框输入表名“PERSON”，单击确定，就可以新建表。

选择名称
 ?
×

输入表名称(E):

新建的表如下所示：



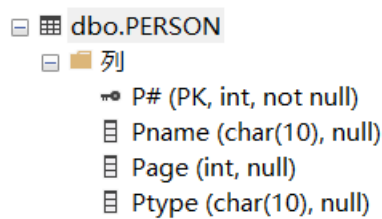
在表“dbo. PERSON”下展开“列”菜单，在“P#(int,null)”上单击右键选择“修改”。在弹出的对话框中进行修改，在“P#”一列单击右键，选择“设置主键”，保存后，便可以完成对表的修改。

LAPTOP-N3QO3T4...t - dbo.PERSON			
	列名	数据类型	允许 Null 值
🔑	P#	int	<input type="checkbox"/>
	Pname	char(10)	<input checked="" type="checkbox"/>
	Page	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

在表“dbo. PERSON”下 菜单“列”上单击右键，选择“新建列”。输入新建列的名称、数据类型，保存后完成对表的修改。

LAPTOP-N3QO3T4...t - dbo.PERSON*			
	列名	数据类型	允许 Null 值
🔑	P#	int	<input type="checkbox"/>
	Pname	char(10)	<input checked="" type="checkbox"/>
	Page	int	<input checked="" type="checkbox"/>
	Ptype	char(10)	<input checked="" type="checkbox"/>
▶			<input type="checkbox"/>

修改后的表如下所示：



(4) 用 SSMS 的查询功能(新建查询)对 PERSON 表进行查询、插入、修改、删除

等操作：首先插入两条记录；修改第二条记录；删除第二条记录。

插入两条记录：

右键单击“Test”，点击“新建查询”，输入下列语句，选中编写的代码，点击运行。

```
INSERT
    INTO PERSON
    VALUES (1, '李勇', 20, 'student');
INSERT
    INTO PERSON
    VALUES (2, '刘晨', 40, 'teacher');
```

运行结果如下：

LAPTOP-N3Q03T4...t - dbo.PERSON X SQLQuery1.sql - L...48.7				
	P#	Pname	Page	Ptype
▶	1	李勇	20	student
	2	刘晨	40	teacher
*	NULL	NULL	NULL	NULL

修改第二条记录：

代码如下

```
UPDATE PERSON
    SET Page=35
    WHERE P#=2;
```

运行结果如下

	P#	Pname	Page	Ptype
▶	1	李勇	20	student
	2	刘晨	35	teacher
*	NULL	NULL	NULL	NULL

删除第二条记录：

代码如下

```
DELETE
    FROM PERSON
    WHERE P#=2;
```

运行结果如下

	P#	Pname	Page	Ptype
▶	1	李勇	20	student
*	NULL	NULL	NULL	NULL

(5) 备份 Test 数据库。

在数据库“Test”上单击右键，选择“任务->备份”，点击确定，完成备份。

备份数据库 - Test

选择页

- 常规
- 介质选项
- 备份选项

脚本 帮助

源

数据库(T): Test

恢复模式(M): 完整

备份类型(K): 完整

☐ 仅复制备份(Y)

备份组件:

☒ 数据库(B)

☐ 文件和文件组(G):

目标

备份到(U): 磁盘

D:\sql\MSSQL15.MSSQLSERVER\MSSQL\Backup\Test.bak

添加(D) ...

删除(R)

内容(C)

确定 取消

连接

服务器:
LAPTOP-N3Q03T48

连接:
wyj

查看连接属性

进度

就绪

Microsoft SQL Server Management Studio

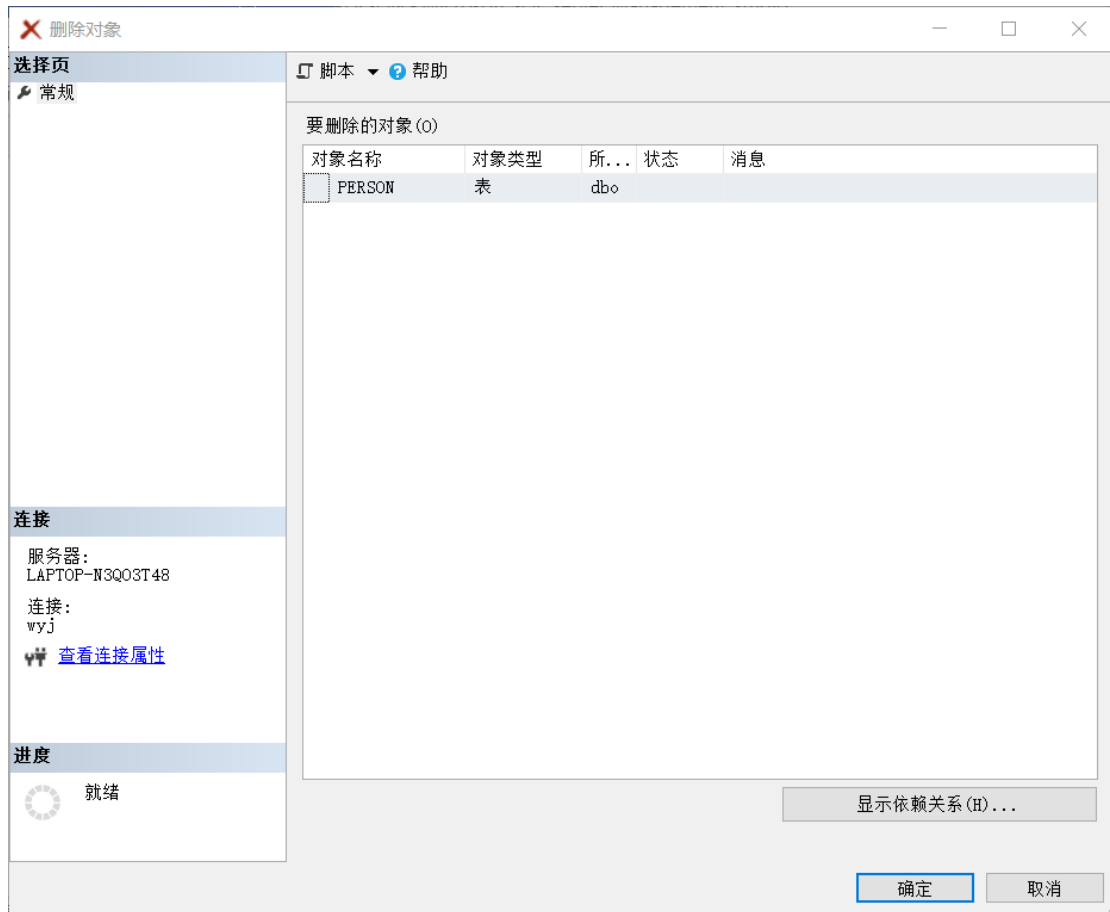
对数据库“Test”的备份已成功完成。

复制消息

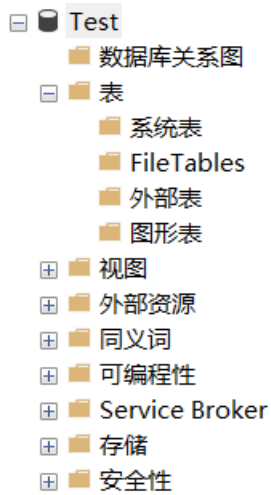
确定

(6) 删除表 PERSON。

在表“dbo. PERSON”上单击右键，选择“删除”。在弹出的对话框单击“确定”，完成对表的删除。

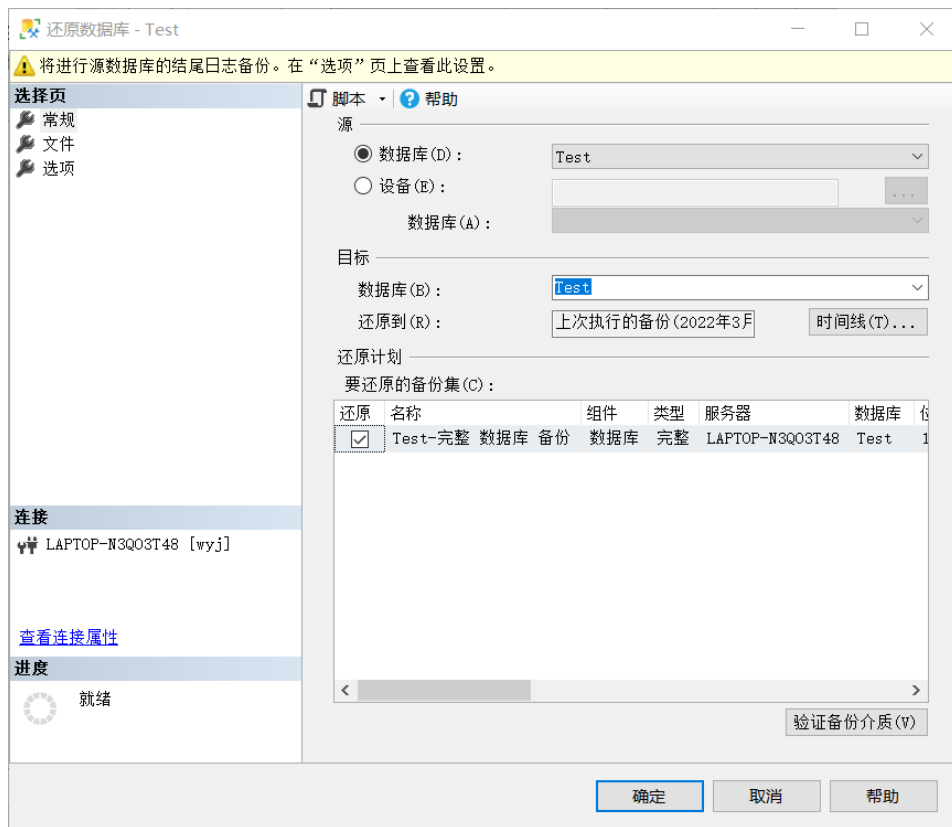


删除后：

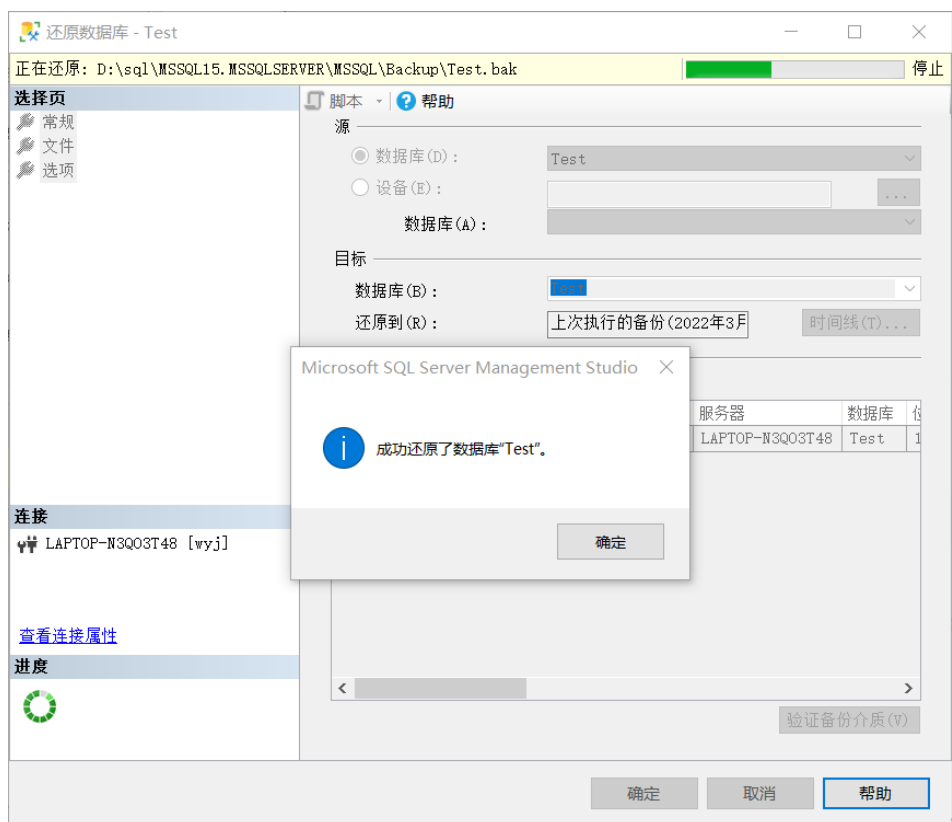


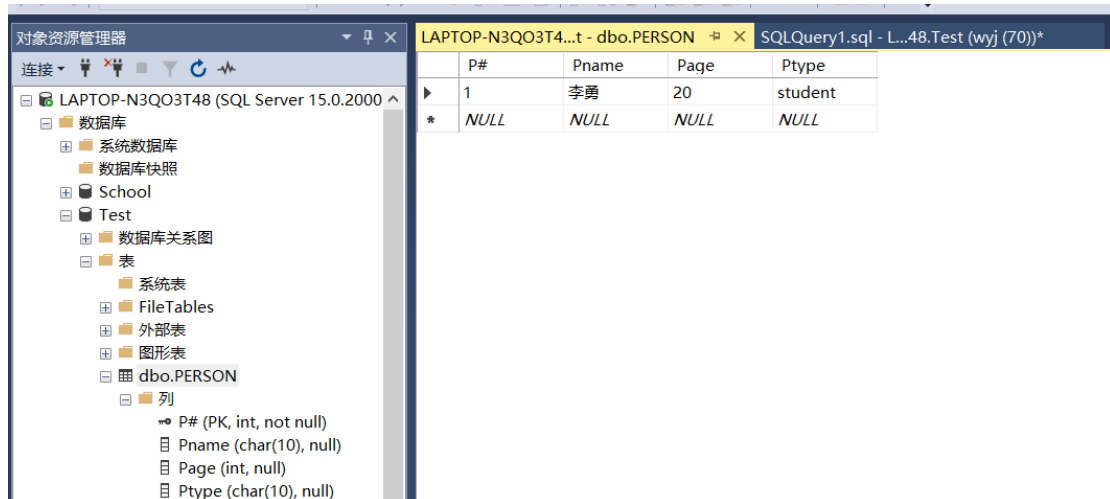
(7) 恢复 Test 数据库。

在数据库“Test”上单击右键，选择“任务->还原->数据库”。在弹出的对话框单击“确认”，完成对数据库的恢复。



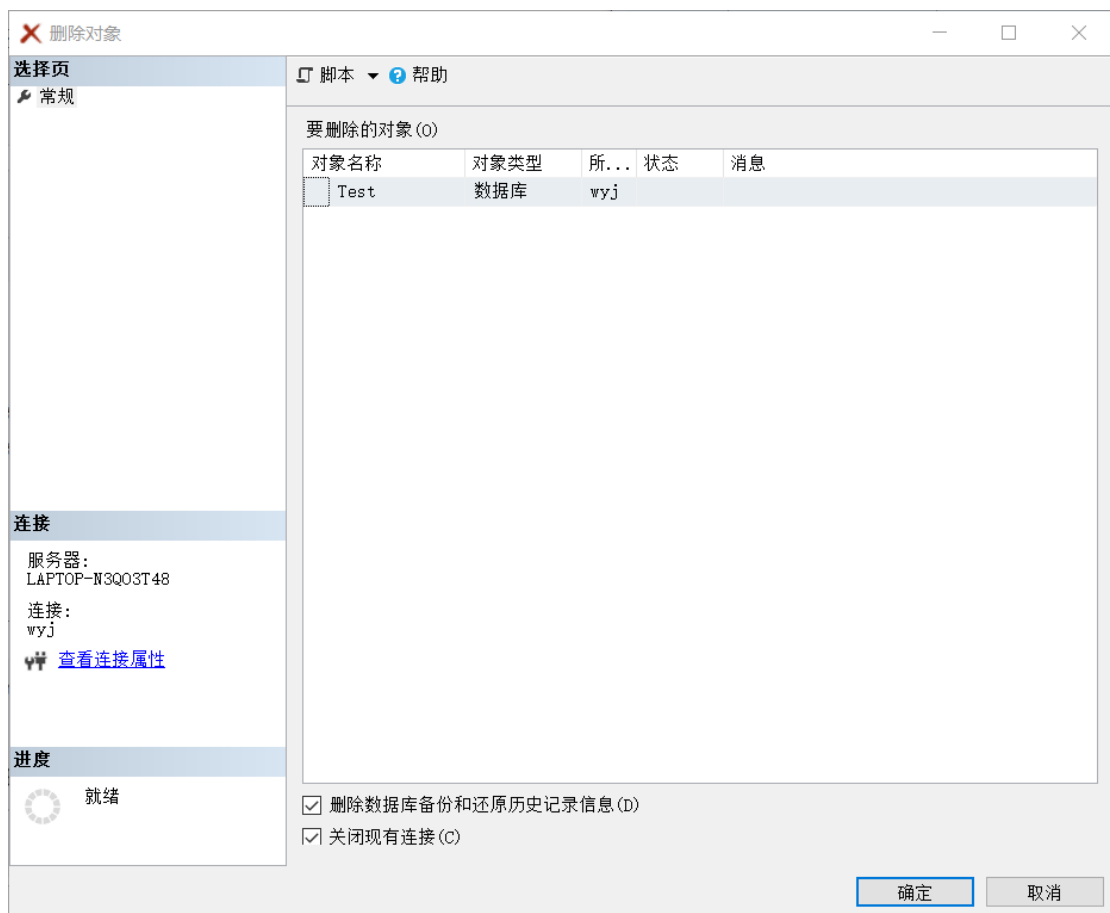
结果如下：



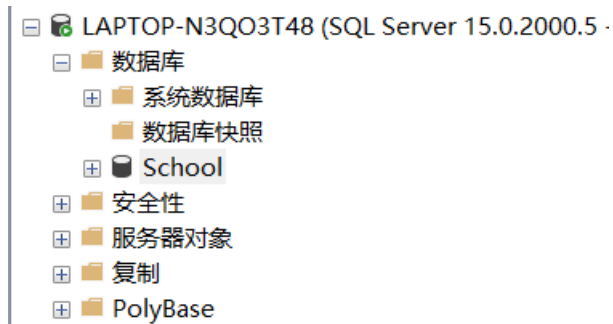


(8) 删除 Test 数据库。

在数据库“Test”上单击右键，选择“删除”。在弹出的对话框单击“确认”，完成对数据库的删除。



效果如下：



1.2 数据定义

(1) 使用 SQL 语句创建关系数据库表：人员表 PERSON(P#, Pname, Page)、房间表 ROOM(R# Rname, Rarea)、表 PR(P#, R#, Date)。其中：P#是表 PERSON 的主键，具有唯一性的约束；Page 具有约束“大于 18”；R#是表 ROOM 的主键，具有唯一性约束。表 PR 中的 P# R#是外键。

代码如下：

```
CREATE DATABASE TEST
```

```
USE TEST
```

```
CREATE TABLE PERSON(
```

```
    P# CHAR(10) PRIMARY KEY,
```

```
    Pname CHAR(10),
```

```
    Page INT,
```

```
    CHECK (Page > 18)
```

```
);
```

```
CREATE TABLE ROOM(
```

```
    R# CHAR(10) PRIMARY KEY,
```

```
    Rname CHAR(10),
```

```
    Rarea CHAR(10)
```

```
);
```

```
CREATE TABLE PR(
```

```
    P# CHAR(10),
```

```
    R# CHAR(10),
```

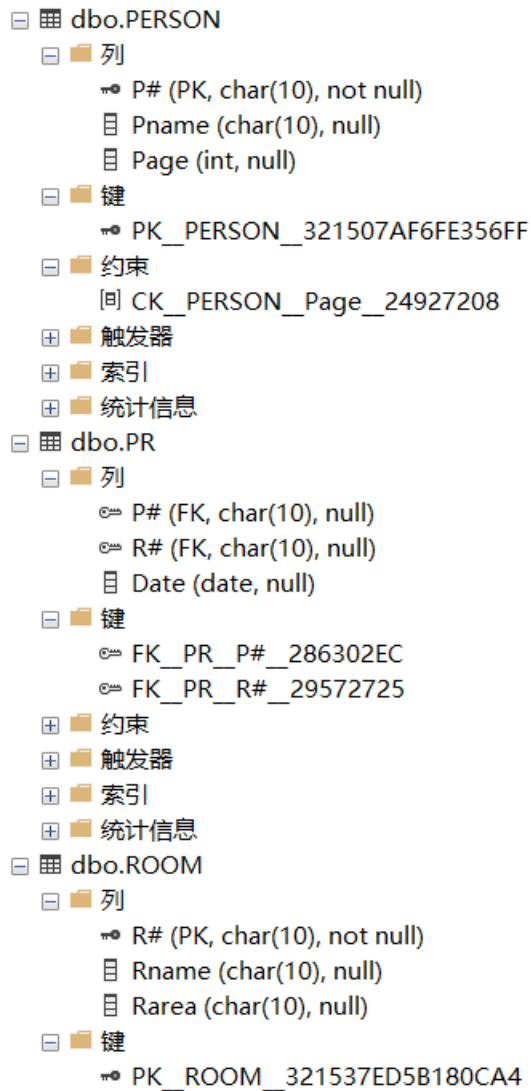
```
    Date DATE,
```

```
    FOREIGN KEY (P#) REFERENCES PERSON(P#),
```

```
    FOREIGN KEY (R#) REFERENCES ROOM(R#)
```

```
);
```

结果如下：

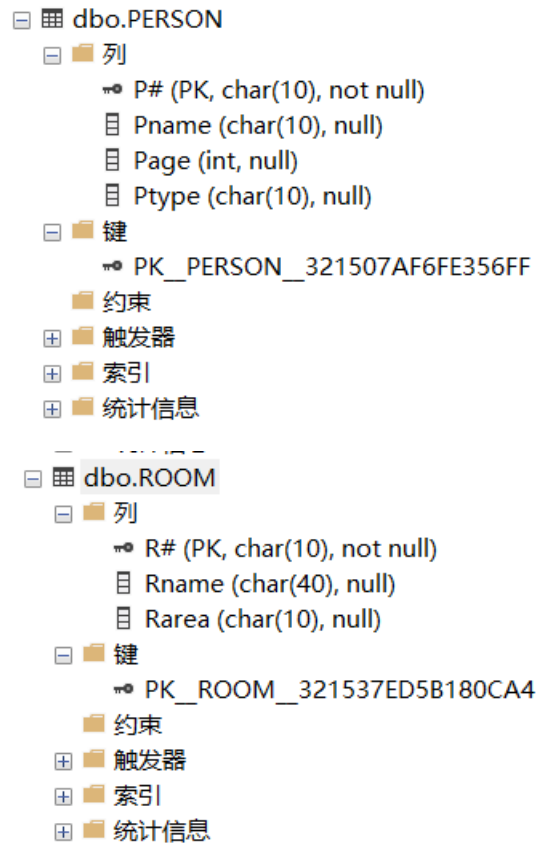


(2) 更改表 PERSON, 增加属性 Ptype (类型是 CHAR, 长度是 10), 取消 Page 大于 18 的约束。把表 ROOM 中的属性 Rname 数据长度改成 40。

代码如下:

```
ALTER TABLE PERSON ADD Ptype CHAR(10);  
ALTER TABLE PERSON DROP CONSTRAINT CK_PERSON_Page_24927208;  
ALTER TABLE ROOM ALTER COLUMN Rname CHAR(40);
```

结果如下:



(3) 删除表 ROOM 中的一个属性 Rarea。

代码如下：

```
ALTER TABLE ROOM DROP COLUMN Rarea;
```

结果如下：



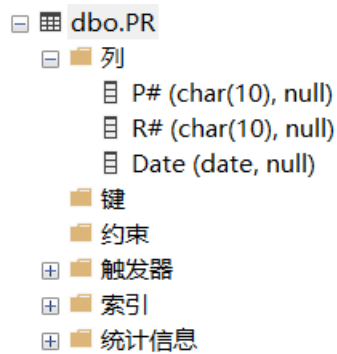
(4) 取消表 PR 的外键。

代码如下：

```
ALTER TABLE PR DROP CONSTRAINT FK__PR__P#__286302EC;
```

```
ALTER TABLE PR DROP CONSTRAINT FK__PR__R#__29572725;
```

结果如下：

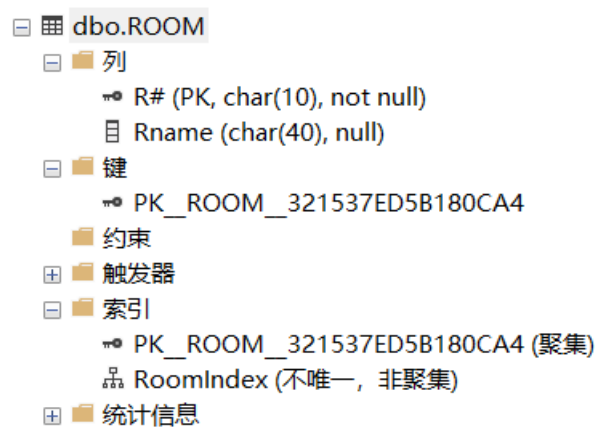


(5) 为表 ROOM 创建按 R#降序排列的索引。

代码如下：

```
CREATE INDEX RoomIndex ON ROOM(R# DESC);
```

结果如下：

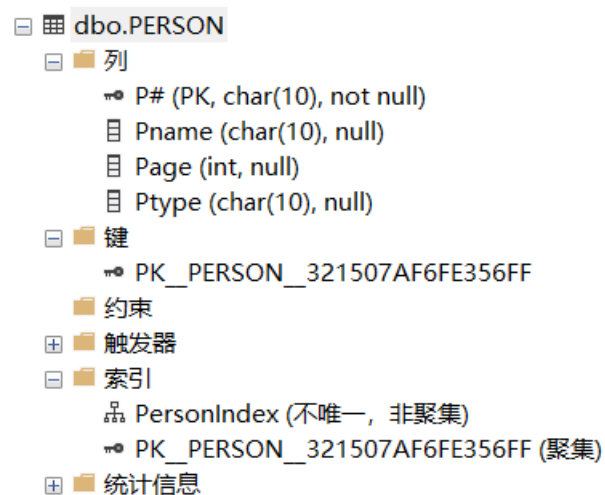


(6) 为表 PERSON 创建按 P#升序排列的索引。

代码如下：

```
CREATE INDEX PersonIndex ON PERSON(P# ASC);
```

结果如下：



(7) 创建表 PERSON 的按 Pname 升序排列的唯一性索引。

代码如下：

```
CREATE UNIQUE INDEX PersonName ON PERSON(Pname ASC);
```

结果如下：

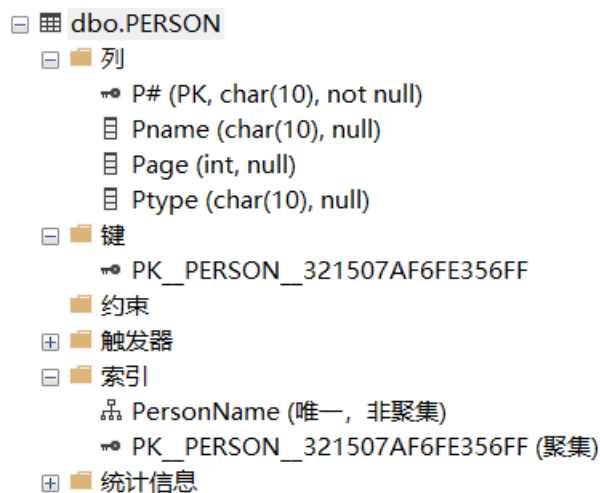


(8) 取消表 PERSON 的 P#升序索引。

代码如下：

```
DROP INDEX PERSON.PersonIndex;
```

结果如下：



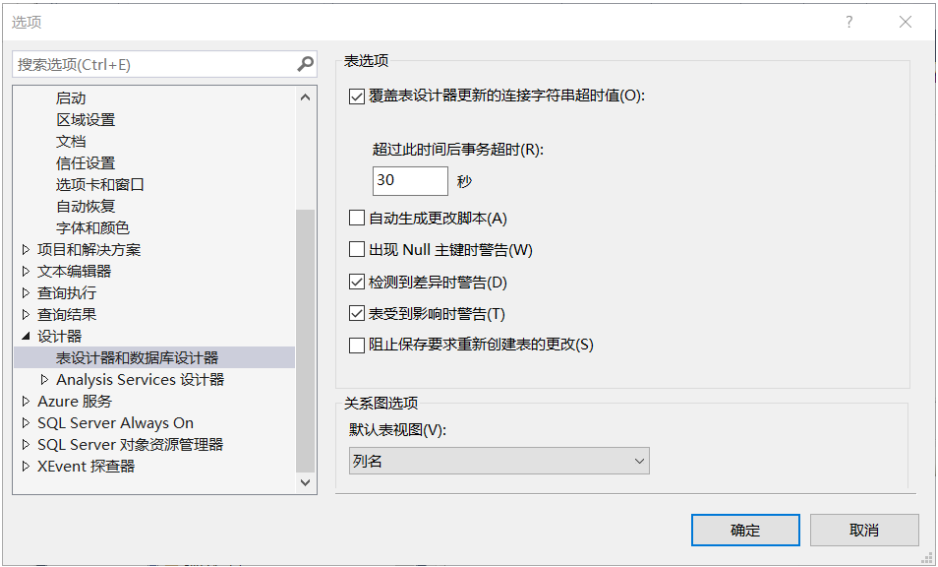
实验总结：

通过本次实验，我熟悉了 SQL SERVER2019 的使用方法和常用操作，由于是第一次安装和操作 SQL，在实验过程中出现了几个错误，在查阅资料后都一一解决了，遇到的问题和解决方法如下：

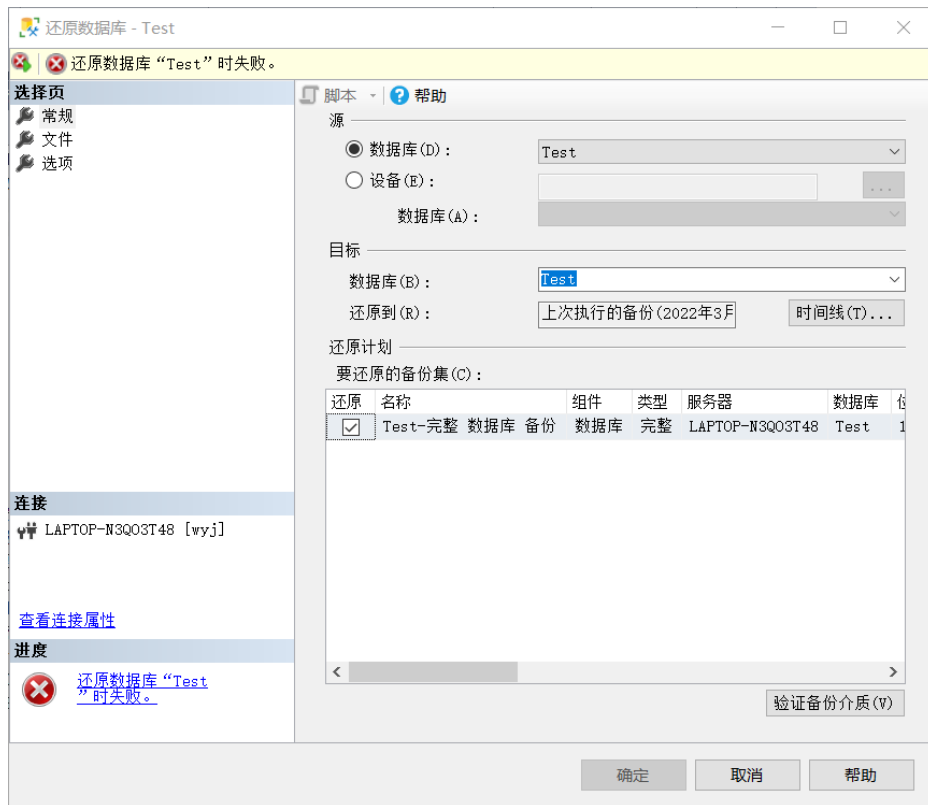
1. 问题一：在做实验 1.1 第 3 个问题，无法保存修改后的 PERSON 表，错误情况如下所示：



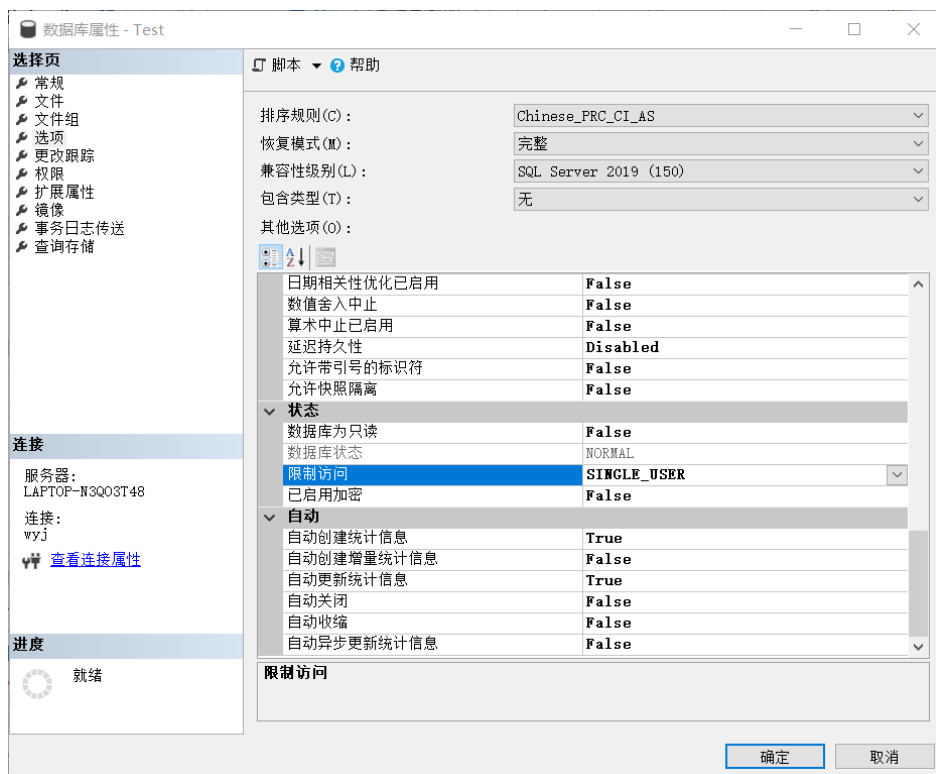
问题一解决方法：点击“工具一>选项”打开选项对话框，左侧目录树选择“设计器”取消勾选右侧“阻止保存要求重新创建表的更改”即可。

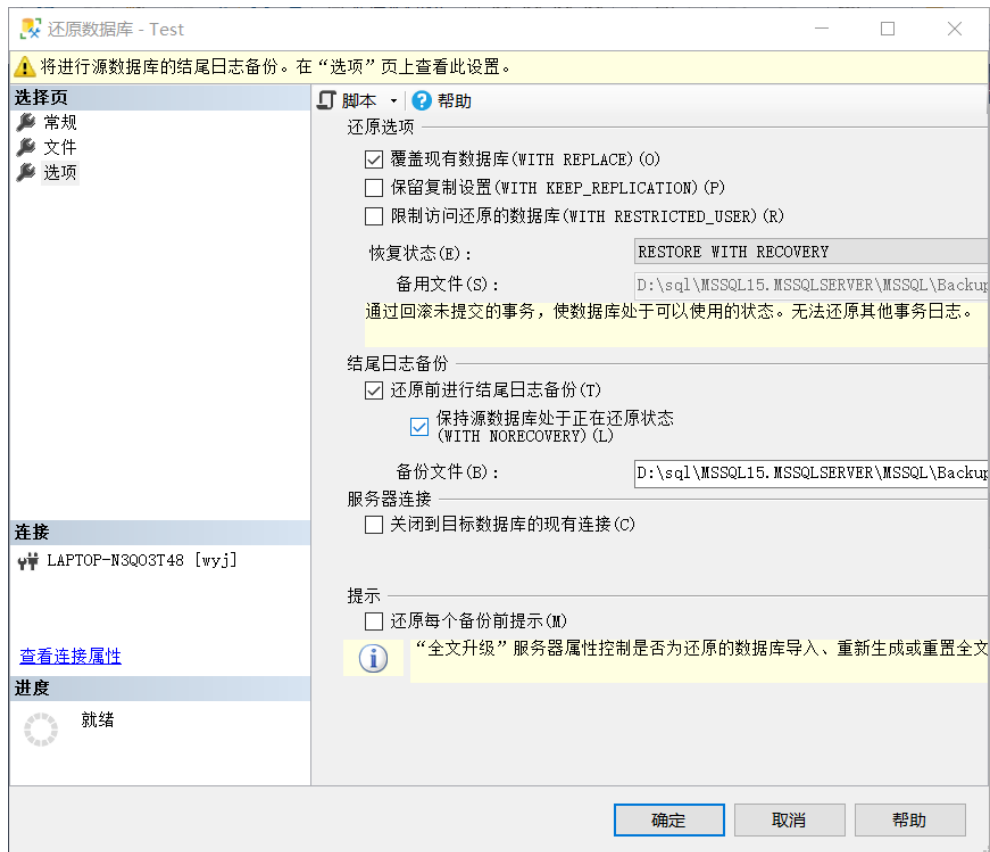


2. 问题二：在做实验 1.1 第 7 个问题，恢复 TEST 数据库的时候出现错误。如下所示：

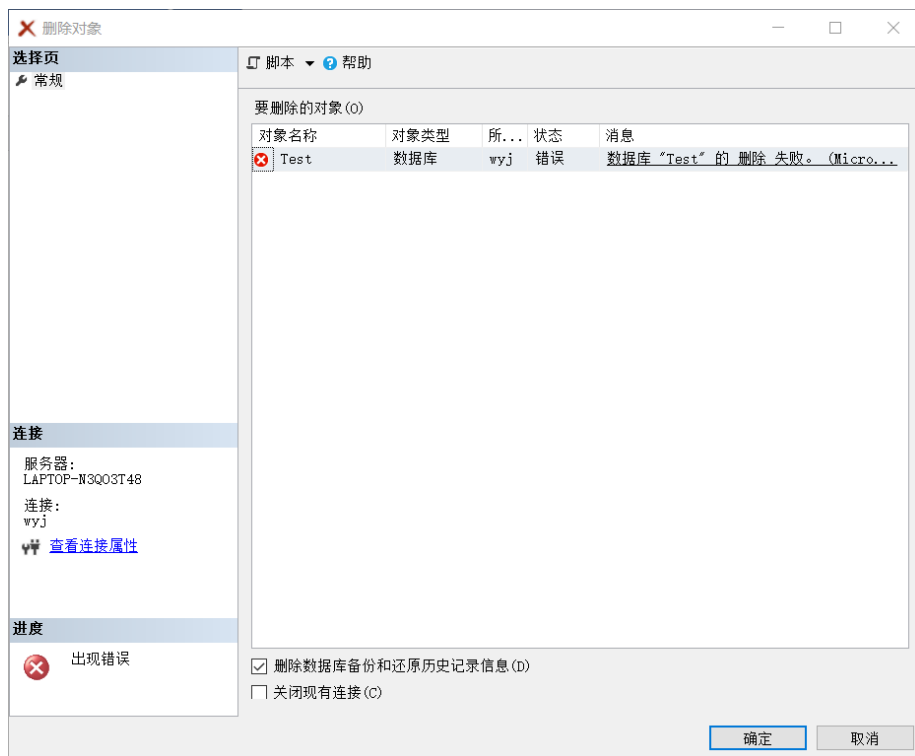


问题二解决方法: 设置数据库属性中的“状态”下的“限制访问”为“SINGLE_USER”，并且勾选还原数据库选项中的“覆盖现有数据库”。

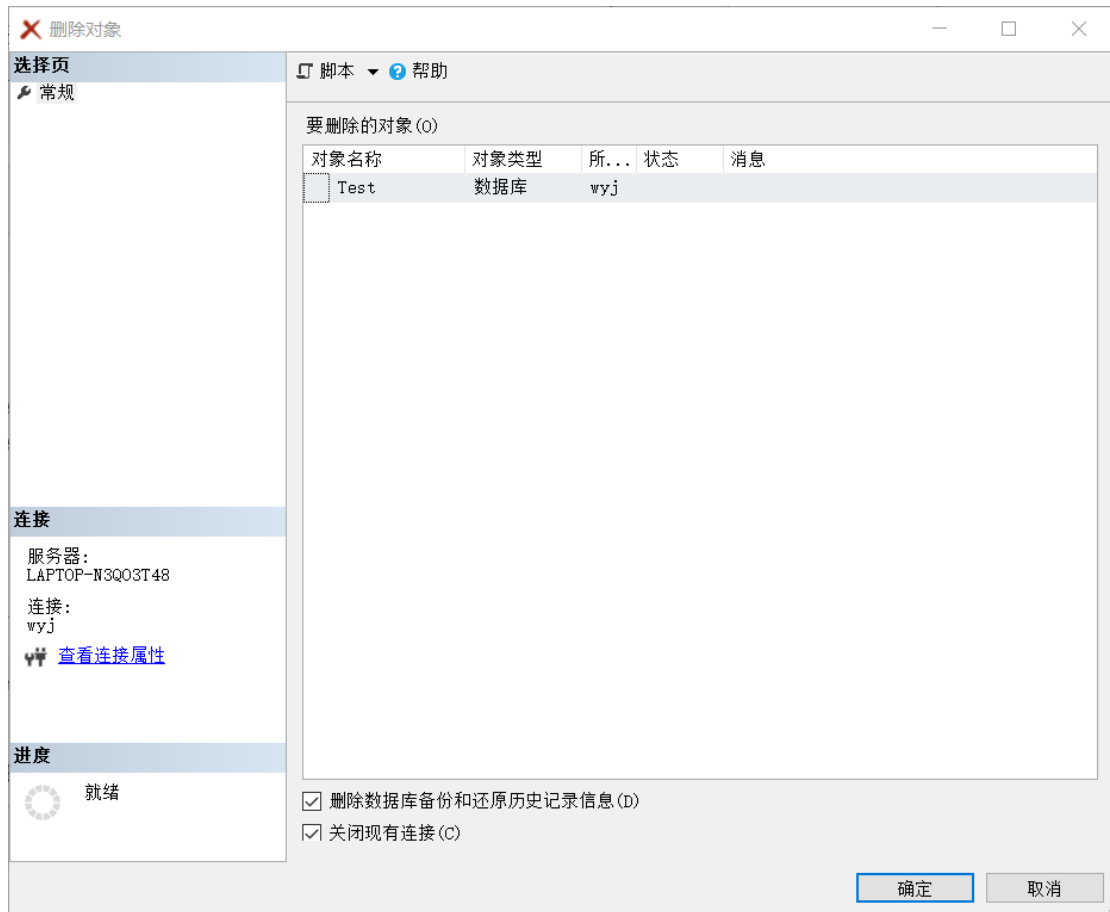




3. 问题三：在做实验 1.1 第 8 个问题时，删除 TEST 数据库失败，如下图所示：



问题三解决方法：在删除时要勾选“关闭现有连接”，即可成功删除。如下图所示：



综上，完成这次实验后，对数据库定义中的各种操作有了具象的理解，为更好地完成以后的数据库实验奠定了基础。