

实验1.1

1 实验内容:

2 实验步骤:

实验 1.2

1 实验内容:

2 实验步骤:

邓语苏 计科2021级本科生

22920212204066

完成日期: 3.21

实验1.1

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

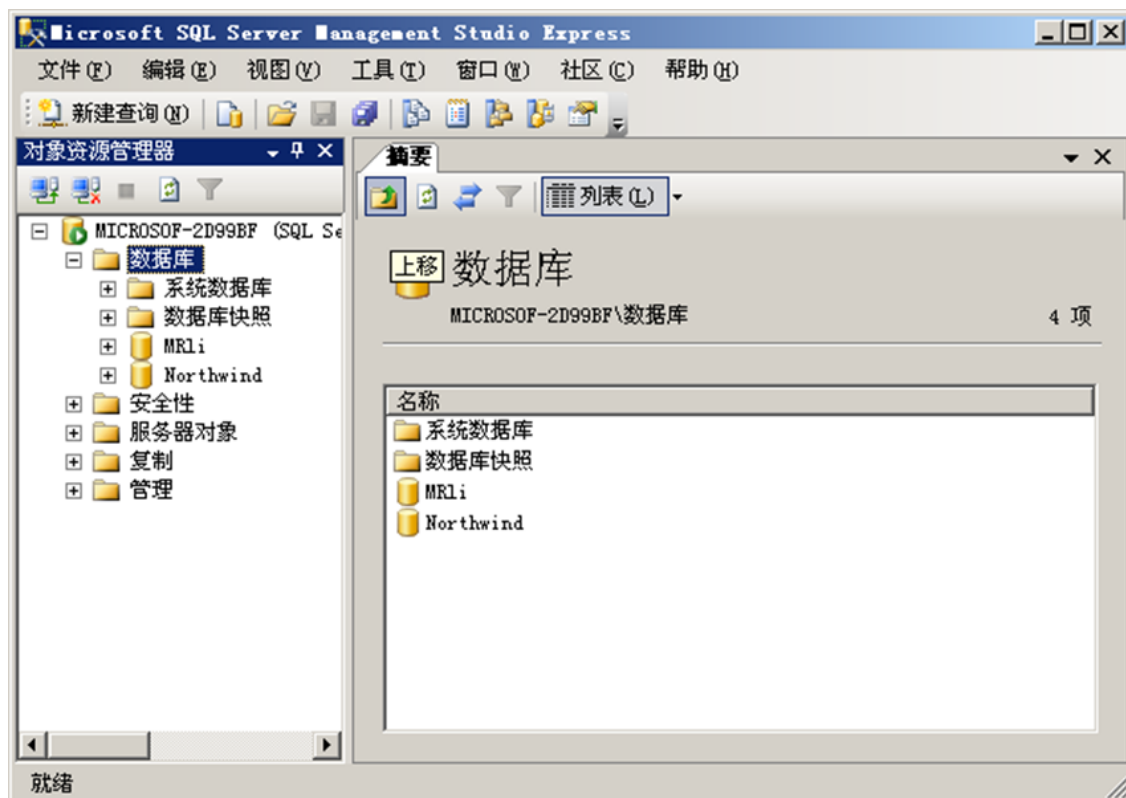
1 实验内容:

1. 使用SSMS(SQL Server Management Studio)加入实验数据库。
2. 使用SSMS可视化建立、修改和删除数据库、表。
3. 使用SSMS对数据库进行备份和恢复。
4. 使用SSMS对表进行查询、插入、修改、删除。

2 实验步骤:

1. 加入School数据库。

使用管理员权限打开SSMS



右键数据库点击附加，在弹出的对话框中选择硬盘上的School_Data.MDF文件。然后单击确定，便可以导入数据库。

2. 建立Test数据库。

新建查询，输入以下代码

```
1 create database Test
2 go
```

新建数据库

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)。

```
1 use School /*在School数据库下新建PERSON表*/
2 create table PERSON(
3     P# char(10) PRIMARY KEY,
4     Pname char(10),
5     Page int,
6     Ptype char(10)
7 )
```

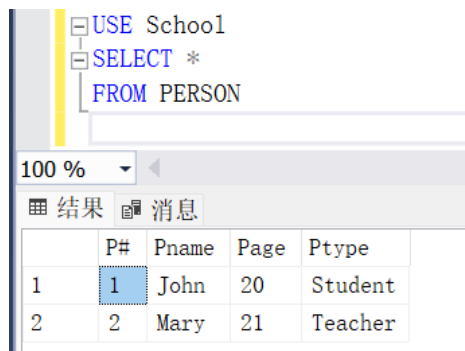
4. 用SSMS的查询功能(新建查询)对PERSON表进行查询、插入、修改、删除等操作：首先插入两条记录；修改第二条记录；删除第二条记录。

o 插入两条记录

```
1 insert into PERSON(P#,Pname,Page,Ptype)
  values(1,'John',20,'Student');
2 insert into PERSON(P#,Pname,Page,Ptype) values(2,'Mary',21,'Teacher');
```

o 查询表中数据

```
1 USE School
2 SELECT *
3 FROM PERSON
```



	P#	Pname	Page	Ptype
1	1	John	20	Student
2	2	Mary	21	Teacher

o 修改第二条记录

```
1 update PERSON set Pname='Johhhh'where P#=2
```

USE School
SELECT *
FROM PERSON

100 %
结果 消息
P# Pname Page Ptype
1 1 John 20 Student
2 2 Johhhh 21 Teacher

- 删除第二条记录

```
1 delete from PERSON where P#=2
```

USE School
delete from PERSON where P#=2
SELECT *
FROM PERSON

100 %
结果 消息
P# Pname Page Ptype
1 1 John 20 Student

5. 备份Test数据库。

```
1 backup database test to disk='E:\XMU\大三下\数据库\实验\实验1\test.bak'
2 go
```

backup database test to disk='E:\XMU\大三下\数据库\实验\实验1\test.bak'
go

.00 %
消息
已为数据库 'test', 文件 'test' (位于文件 1 上)处理了 576 页。
已为数据库 'test', 文件 'test_log' (位于文件 1 上)处理了 2 页。
BACKUP DATABASE 成功处理了 578 页, 花费 0.025 秒(180.468 MB/秒)。
完成时间: 2024-03-21T15:36:21.1841284+08:00

6. 删除表PERSON。

```
1 use school
2 drop table PERSON
3 go
```

7. 恢复Test数据库。

```
1 restore database test from disk ='E:\XMU\大三下\数据库\实验\实验1\test.bak'
```

8. 删除Test数据库。

```
1 drop database test
2 go
```

实验 1.2

1 实验内容：

1. 使用CREATE语句创建基本表。
2. 更改基本表的定义，增加列，删除列，修改列的数据类型。
3. 创建表的升降序索引。
4. 取消表、表的索引或表的约束。

2 实验步骤：

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。

创建新表前需运行以下代码删除PERSON、ROOM、PR表

```
1 drop table PERSON
2 drop table ROOM
```

```
1 create database hospital--新建一个名为hospital的数据库
2
3
4 use hospital
5 -- 创建人员表 PERSON
6 create table PERSON (
7     P# int primary key,
8     Pname varchar(50),
9     Page int check (Page > 18) -- 确保page大于18
10 );
11
12 -- 创建房间表 ROOM
13 create table ROOM (
14     R# int primary key,
15     Rname varchar(50),
16     Rarea float
17 );
18
19 -- 创建关系表 PR
20 create table PR (
21     P# int,
22     R# int,
23     Date datetime,
24     foreign key (P#) references PERSON(P#), -- 外键约束，确保p#在person表中
    存在
25     foreign key (R#) references ROOM(R#) -- 外键约束，确保r#在room表中存在
26 );
27
28
```

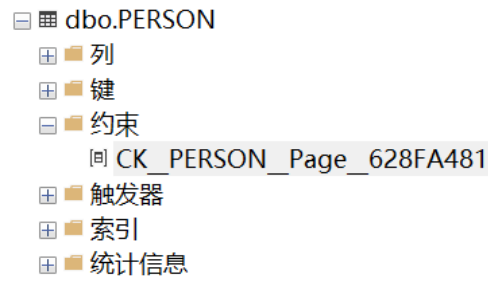
2. 更改表PERSON,增加属性Ptype(类型是CHAR,长度是10)，取消Page大于18的约束。把表ROOM中的属性Rname数据长度改成40。

- 增加属性Ptype(类型是CHAR,长度是10)

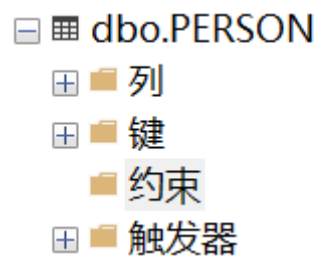
```
1 alter table PERSON
2 add Ptype char(10);
```

- 取消Page大于18的约束

找到PERSON表中的约束名

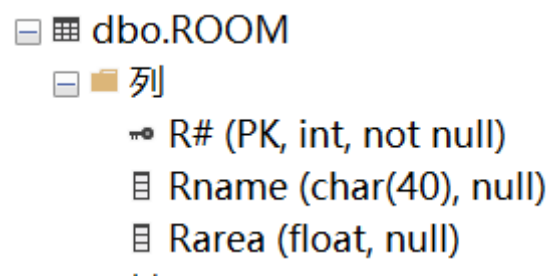


```
1 alter table PERSON
2 drop constraint CK_PERSON_Page_628FA481;
```



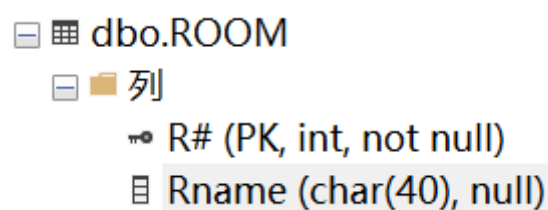
- 把表ROOM中的属性Rname数据长度改成40

```
1 use hospital
2 alter table ROOM
3 alter column Rname char(40);
```



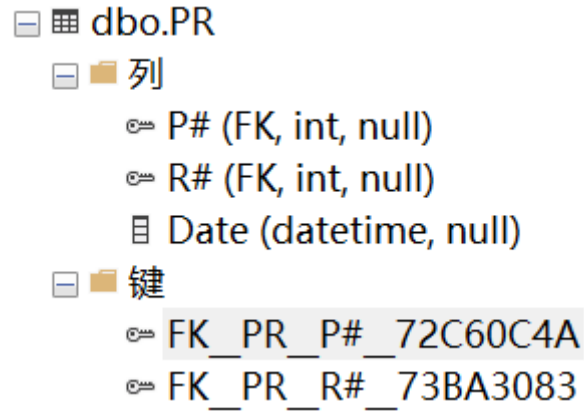
3. 删除表ROOM中的一个属性 Rarea。

```
1 alter table ROOM
2 drop column Rarea;
```



4. 取消表PR的外键。

打开PR表中键文件夹

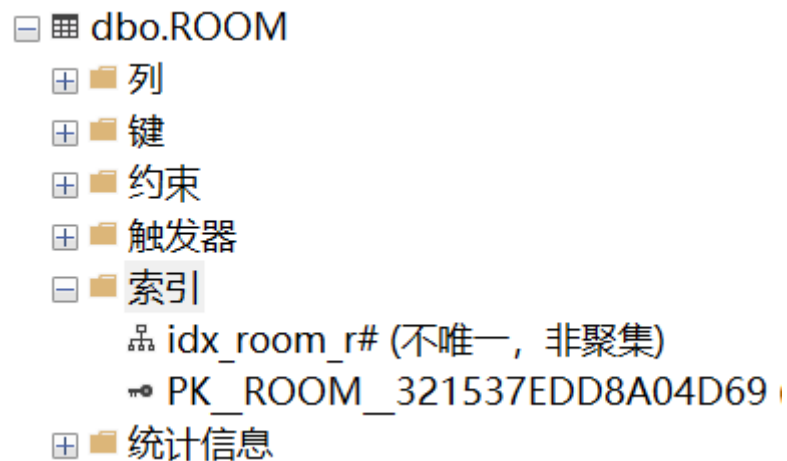


```
1 use hospital
2 alter table PR
3 drop constraint FK__PR__P#__72C60C4A;
4 alter table PR--两个alter
5 drop constraint FK__PR__R#__73BA3083;
```



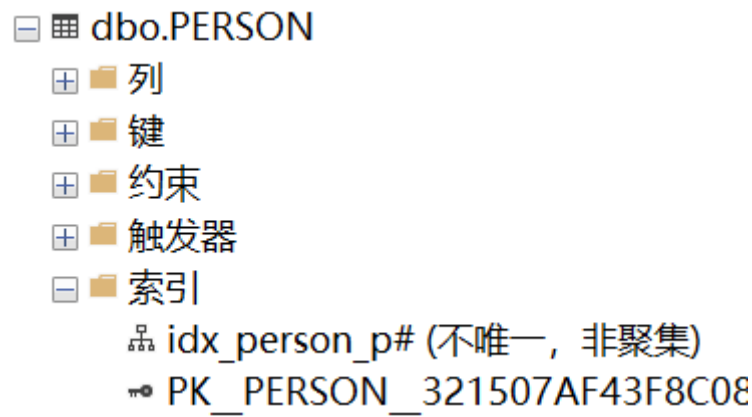
5. 为表ROOM创建按R#降序排列的索引。

```
1 create index idx_room_r# on ROOM (R# DESC)
```



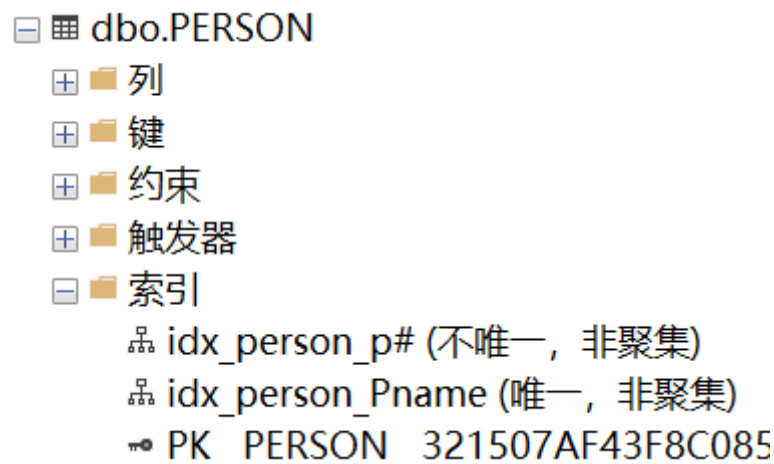
6. 为表PERSON创建按P#升序排列的索引。

```
1 create index idx_person_p# on PERSON(P# ASC)
```



7. 创建表PERSON的按Pname升序排列的唯一性索引。

```
1 create unique index idx_person_Pname on PERSON(Pname ASC)
```



8. 取消表PERSON的P#升序索引。

```
1 drop index idx_person_p# on PERSON
```

