

SQL交互式实验

姓名	学号	班级	专业
王何佳	2023211603	2023211804	网络空间安全

一、实验概况

实验目的

熟悉通过SQL对数据库进行操作

实验工具

SQL Server/MySQL提供的交互查询工具

实验数据库

学生选课（自行补充扩展）

实验内容

数据定义

- 基本表的创建(3)、修改(1)和删除(1)
- 索引的创建(1)和删除(1)

数据操作

- 各类更新操作（插入数据(n)、修改数据(1)、删除数据(1)）
- 各类查询操作（单表查询(1)、连接查询(2)、嵌套查询(2)、集合查询(1)）

二、认识DBMS

1.安装

微软官网下载SQL: [SQL Server 下载](#) | [icrosoft](#)

SQL Server 2022 安装

数据库引擎配置

指定数据库引擎身份验证安全模式、管理员、数据目录、TempDB、最大并行度、内存限制和文件流设置。

全局规则

Microsoft 更新

产品更新

安装安装程序文件

安装规则

安装类型

版本

许可条款

适用于 SQL Server 的 Azure ...

功能选择

功能规则

实例配置

服务器配置

数据库引擎配置

功能配置规则

准备安装

安装进度

完成

服务器配置 数据目录 TempDB MaxDOP 内存 FILESTREAM

为数据库引擎指定身份验证模式和管理员。

身份验证模式

☐ Windows 身份验证模式(W)

☒ 混合模式(SQL Server 身份验证和 Windows 身份验证)(M)

为 SQL Server 系统管理员(sa)帐户指定密码。

输入密码(E): ●●●●●

确认密码(O): ●●●●●

指定 SQL Server 管理员

BASTANDERN\62477 (62477)

SQL Server 管理员对数据库引擎具有无限制的访问权限。

添加当前用户(C)

添加(A)...

删除(R)

< 上一步(B)

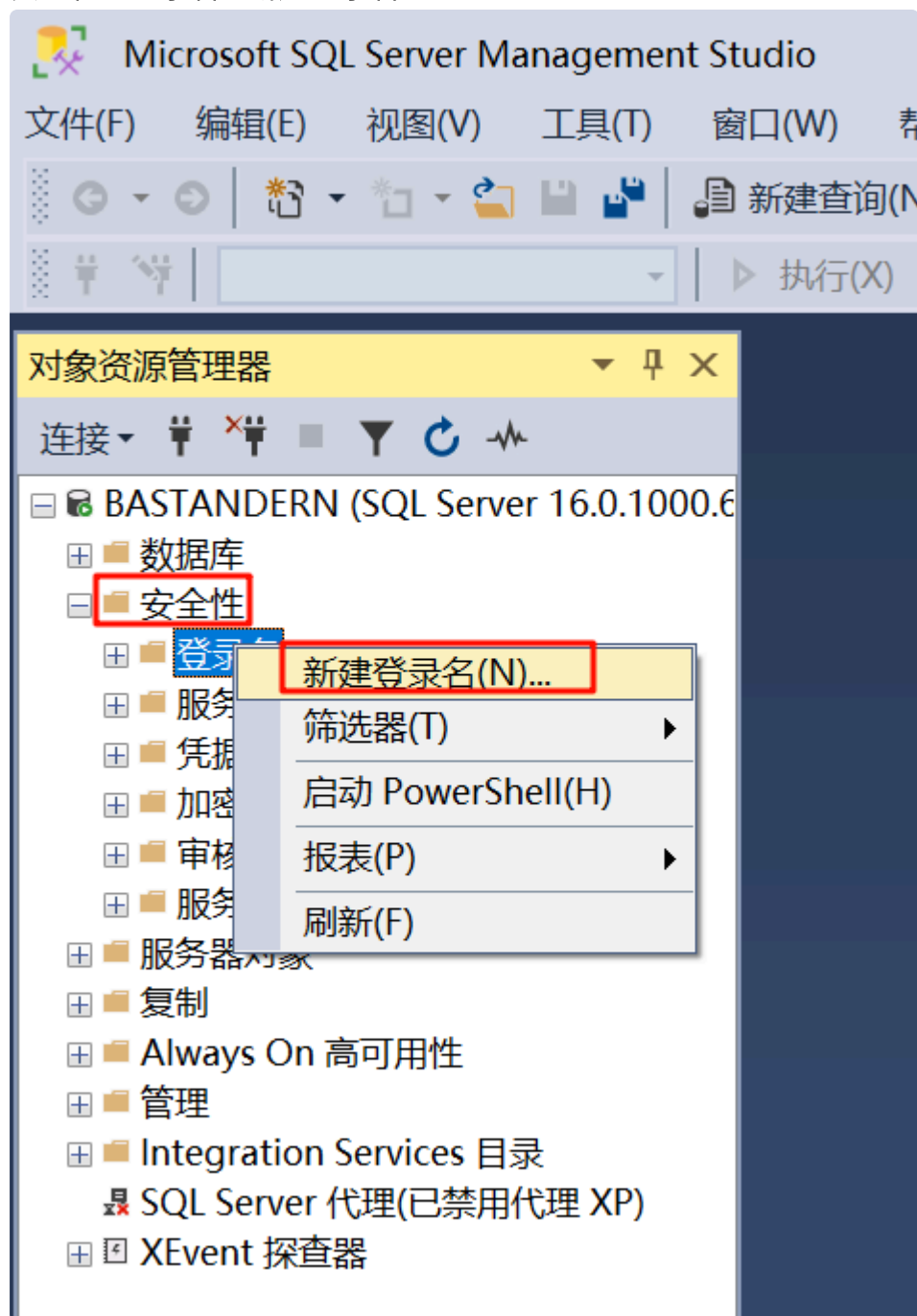
下一步(N) >

取消

下载SSMS: [下载 SQL Server Management Studio \(SSMS\)](#) | [Microsoft Learn](#)

2.创建用户、赋予权限

安全性→登录名→新建登录名



常规→选择SQL Server身份验证，输入密码。

想设置简单一点可以不勾选强制密码策略

登录名 - 新建

选择页

常规

服务器角色

用户映射

安全对象

状态

脚本

帮助

登录名(N):whj搜索(E)...

Windows 身份验证(W)

Microsoft Entra ID 身份验证

SQL Server 身份验证(S)

密码(P):

确认密码(C):

☐ 指定旧密码(I)

旧密码(O):

☐ 强制实施密码策略(F)

☐ 强制密码过期(X)

☐ 用户在下次登录时必须更改密码(U)

映射到证书(R)

映射到非对称密钥(T)

☐ 映射到凭据(M)

映射的凭据

凭据提供程序

添加(A)

删除(V)

默认数据库(D):master

默认语言(G):<默认>

确定

取消

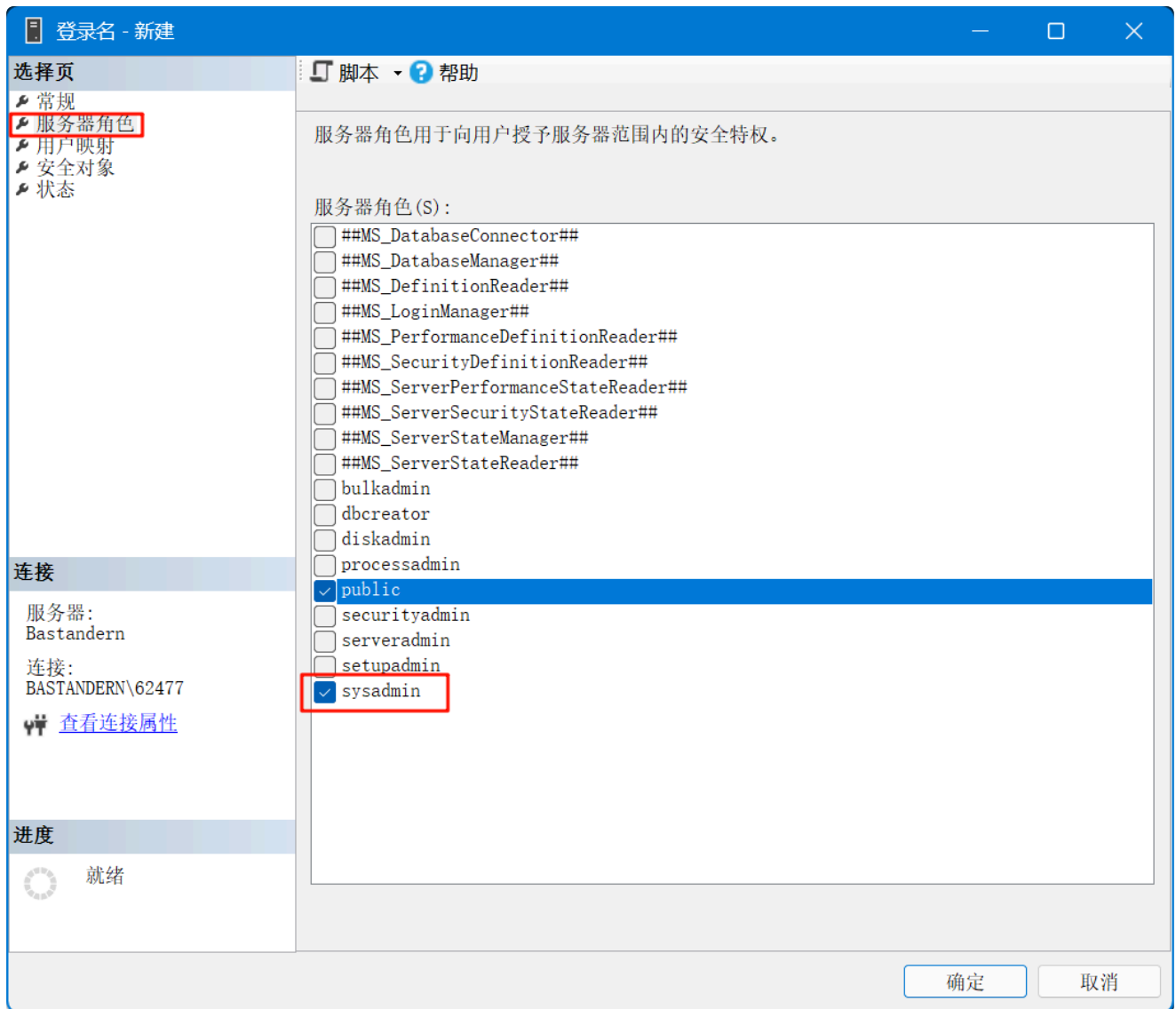
连接

服务器:
Bastandern
连接:
BASTANDERN\62477
[查看连接属性](#)

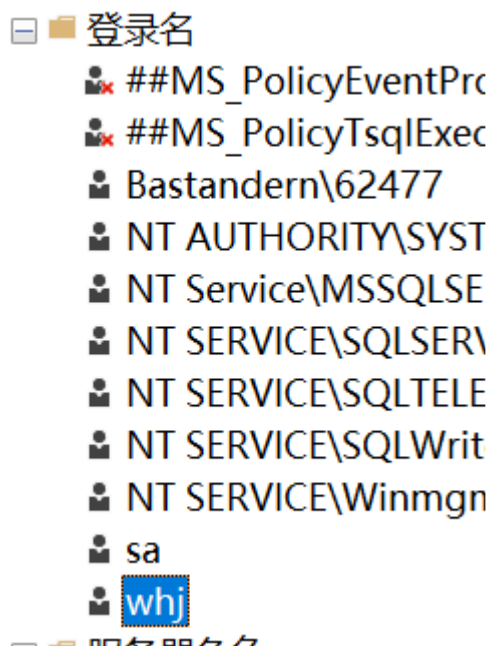
进度

就绪

服务器角色→勾选sysadmin授权

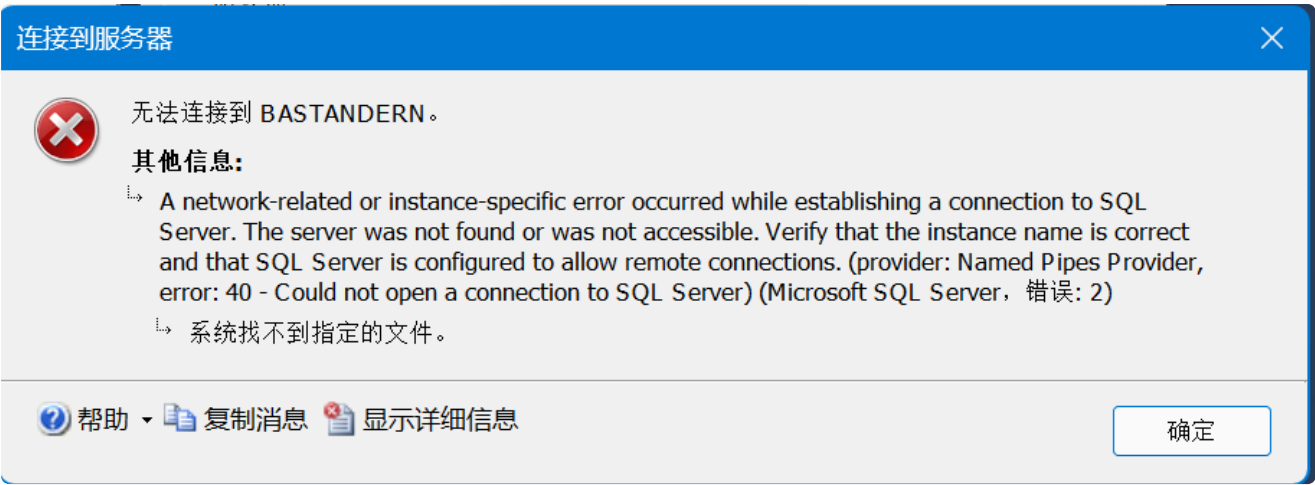


用户创建成功



3.服务的启动和停止

服务未开启时，打开SQL Server Management Studio 20无法连接

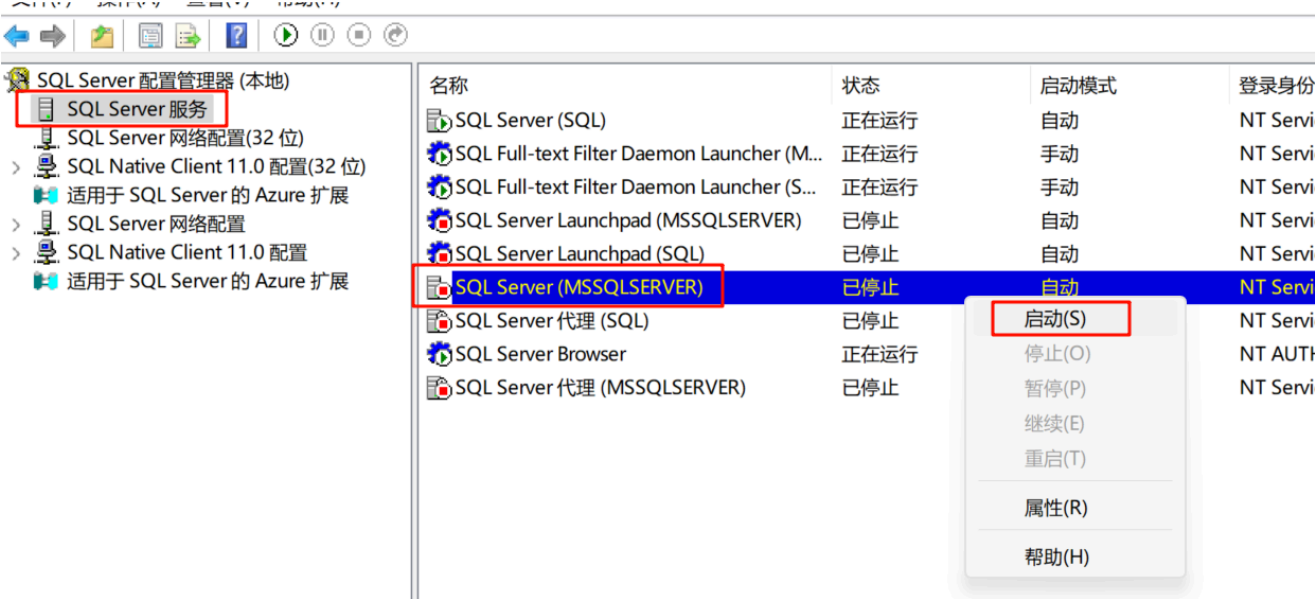


打开SQL Server 2022配置管理器

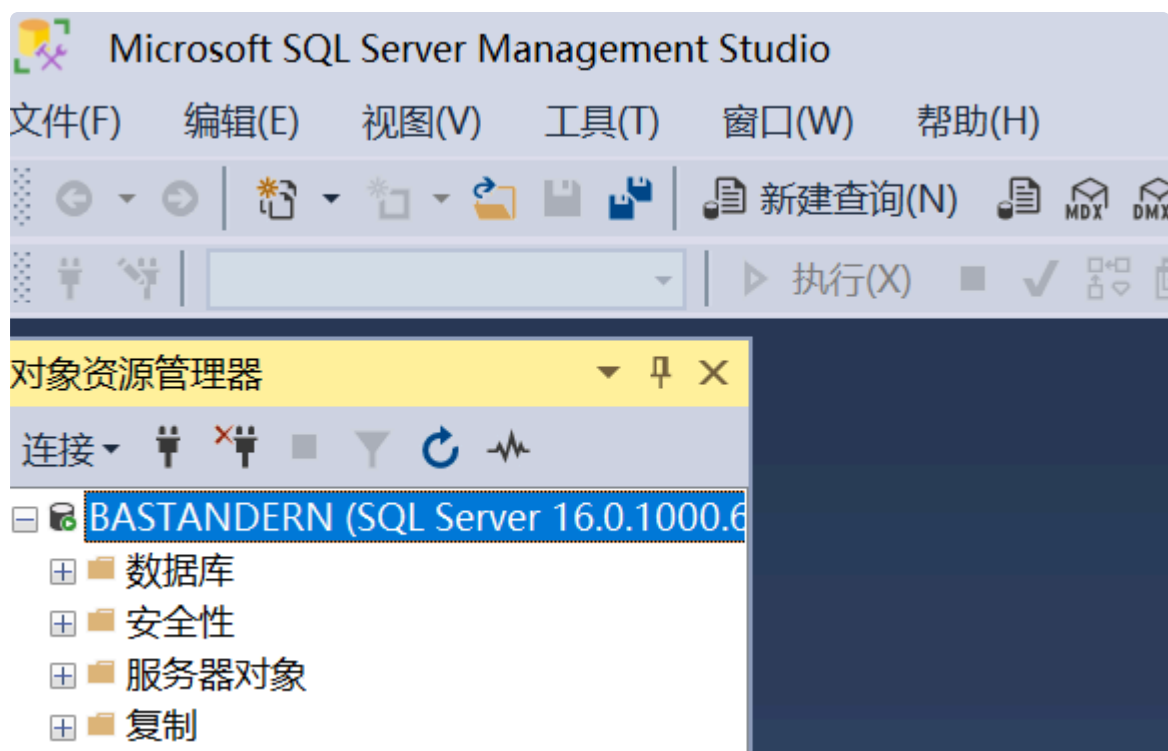


启动

SQL Server服务→选中SQL Server (MSSQLSERVER) →右键→启动



成功连接



停止

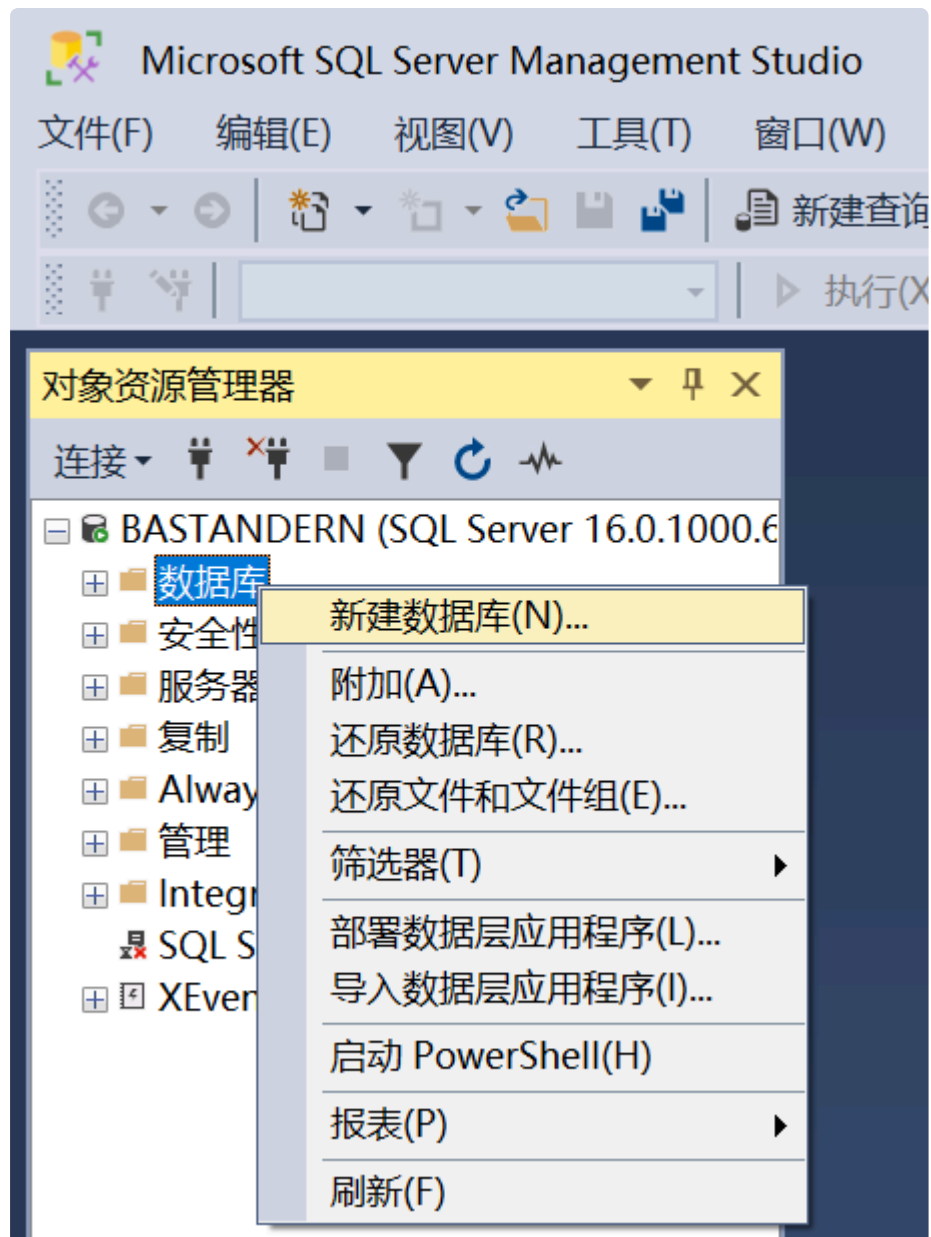
运行状态下，右键→停止



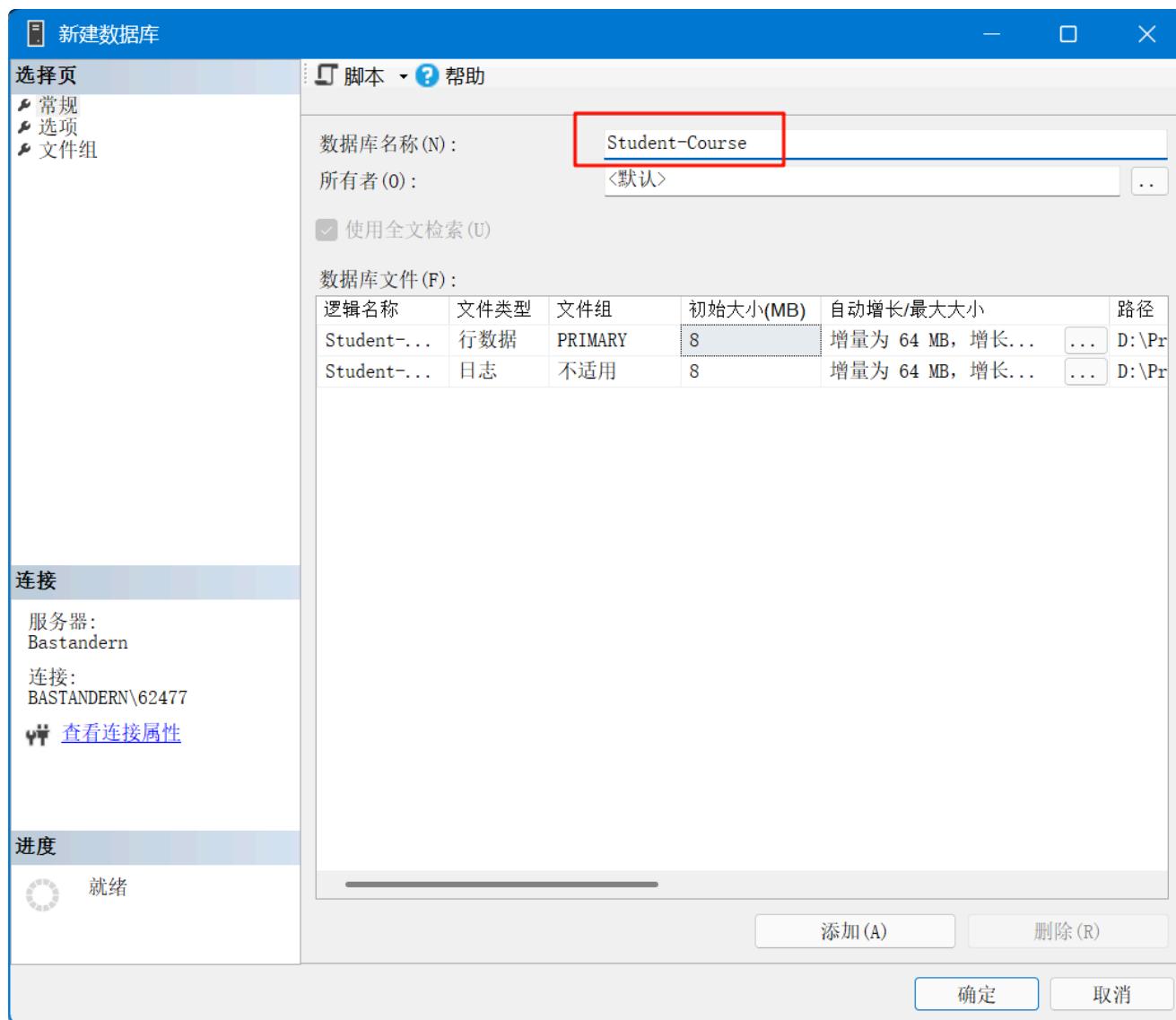
此时打开又弹出无法连接

4.创建数据库和表

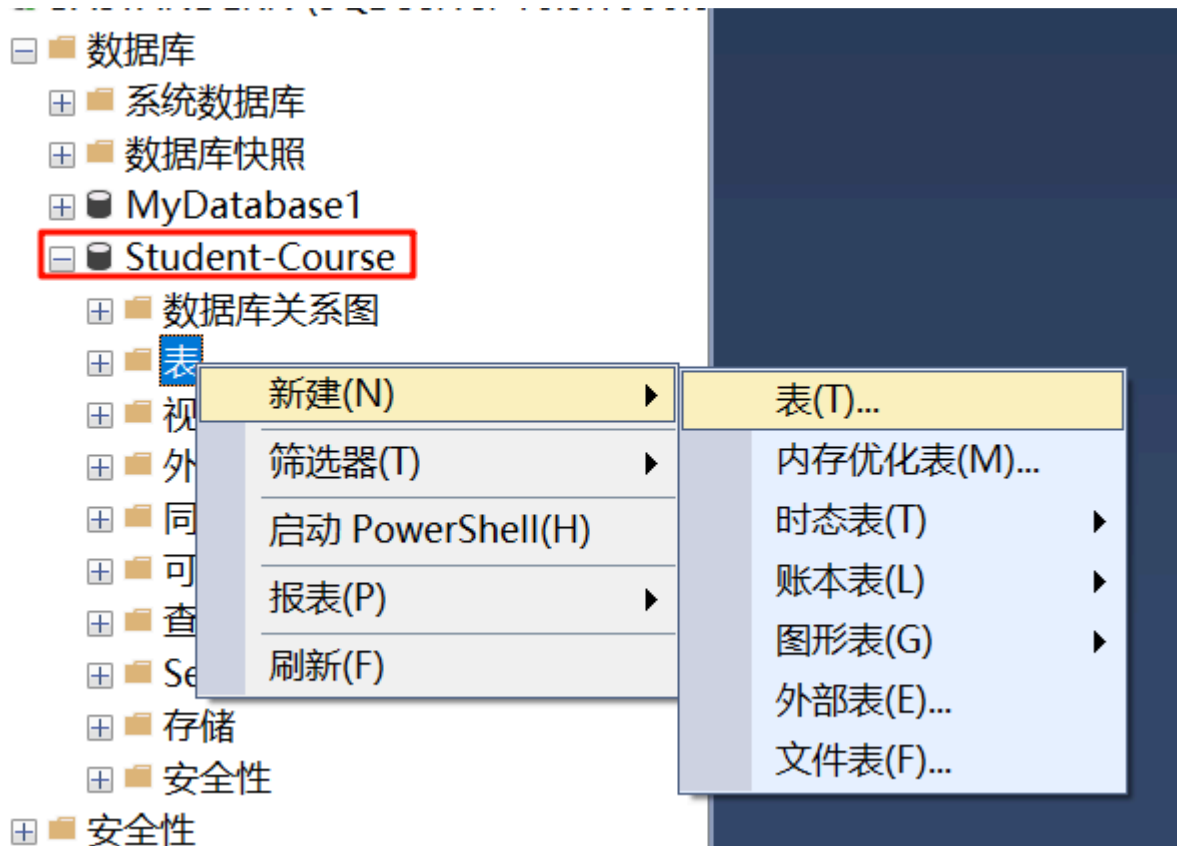
数据库→右键新建数据库



输入数据库名称



数据库→新建的数据库 (Student-Course) →表→新建→表



Student属性

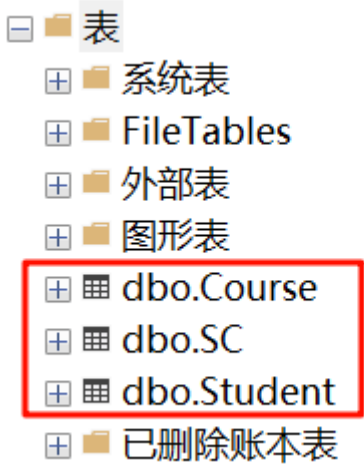
	列名	数据类型	允许 Null 值
	sno	char(10)	<input checked="" type="checkbox"/>
	sname	char(10)	<input checked="" type="checkbox"/>
	ssex	char(2)	<input checked="" type="checkbox"/>
	sage	int	<input checked="" type="checkbox"/>
	sdept	nchar(10)	<input checked="" type="checkbox"/>

Course属性

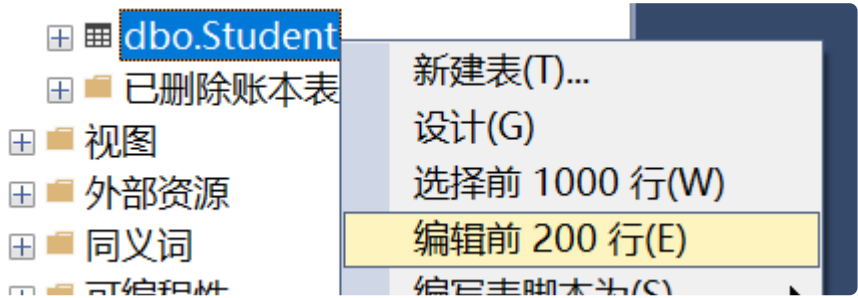
	列名	数据类型	允许 Null 值
	cno	int	<input checked="" type="checkbox"/>
	cname	nchar(10)	<input checked="" type="checkbox"/>
	cpno	int	<input checked="" type="checkbox"/>
	ccredit	int	<input checked="" type="checkbox"/>

SC属性

	列名	数据类型	允许 Null 值
	sno	char(10)	<input checked="" type="checkbox"/>
	cno	int	<input checked="" type="checkbox"/>
	grade	int	<input checked="" type="checkbox"/>



选择编辑前200行



写入数据

Student

	sno	sname	ssex	sage	sdept
	01	张三	男	19	网安
	02	李四	男	18	网安
	03	王五	女	19	电子
	04	赵六	女	20	网安

Course

	cno	cname	cpno	ccredit
	1	数据库	NULL	2
	2	计算机网...	NULL	3
	3	操作系统 ...	NULL	3
	4	逆向工程 ...	3	3
	5	信安数学 ...	NULL	2
	6	现代密码...	5	3

SC

	sno	cno	grade
	01	1	94
	03	2	89
	02	1	90
	04	4	90

5.问题与解决

问题：新建表后找不到

解决：右键→刷新



三、交互式SQL实验过程

创建数据库

新建数据库

选择页

- 常规
- 选项
- 文件组

脚本 帮助

数据库名称(N): S-C Management

所有者(O): <默认>

☒ 使用全文检索(U)

数据库文件(F):

逻辑名称	文件类型	文件组	初始大小(MB)	自动增长/最大大小	路径
S-C Mana...	行数据	PRIMARY	8	增量为 64 MB, 增长...	D:\Pr
S-C Mana...	日志	不适用	8	增量为 64 MB, 增长...	D:\Pr

数据定义

1.创建基本表

(1) 创建学生信息表Student

```
CREATE TABLE Student (  
    sno CHAR(10) NOT NULL PRIMARY KEY, --学号, 主键  
    sname CHAR(10) NOT NULL UNIQUE, --姓名, 唯一  
    ssex CHAR(2) NOT NULL CHECK (ssex IN ('男', '女')), --性别  
    sage INT NOT NULL, --年龄  
    sdept NCHAR(10) NOT NULL --所在系  
);
```

SQLQuery1.sql -...DERN\62477 (54))*

```
CREATE TABLE Student (  
    sno CHAR(10) NOT NULL PRIMARY KEY, --学号, 主键  
    sname CHAR(10) NOT NULL UNIQUE, --姓名, 唯一  
    ssex CHAR(2) NOT NULL CHECK (ssex IN ('男', '女')), --性别  
    sage INT NOT NULL, --年龄  
    sdept NCHAR(10) NOT NULL --所在系  
);
```

110 %

消息

命令已成功完成。

完成时间: 2025-05-11T00:05:17.0883757+08:00

(2) 创建课程信息表Course

```
CREATE TABLE Course (  
    cno INT NOT NULL PRIMARY KEY,    --课程号，主键  
    cname NCHAR(10) NOT NULL,        --课程名  
    cpno INT NULL,                    --先修课程，可以无  
    ccredit INT NOT NULL CHECK (ccredit > 0), --课程学分  
    FOREIGN KEY (cpno) REFERENCES Course(cno)  
    -- cpno必须引用已存在的cno  
);
```

SQLQuery1.sql -...DERN\62477 (54))*

```
CREATE TABLE Course (  
    cno INT NOT NULL PRIMARY KEY,    --课程号，主键  
    cname NCHAR(10) NOT NULL,        --课程名  
    cpno INT NULL,                    --先修课程，可以无  
    ccredit INT NOT NULL CHECK (ccredit > 0), --课程学分  
    FOREIGN KEY (cpno) REFERENCES Course(cno)  
    -- cpno必须引用已存在的cno  
);
```

110 %

消息

命令已成功完成。

完成时间: 2025-05-11T00:07:27.4695466+08:00

(3) 创建学生选课表SC

```
CREATE TABLE SC (  
    sno CHAR(10) NOT NULL,    --学号  
    cno INT NOT NULL,         --课程号  
    grade INT NULL CHECK (grade BETWEEN 0 AND 100),  
                                -- 假设成绩为百分制  
    PRIMARY KEY (sno, cno),    -- 联合主键  
    FOREIGN KEY (sno) REFERENCES Student(sno), -- 关联学生表  
    FOREIGN KEY (cno) REFERENCES Course(cno)   -- 关联课程表  
);
```

SQLQuery1.sql - ...DERN\62477 (54))* ✎ ✕

```
CREATE TABLE SC (  
    sno CHAR(10) NOT NULL,      --学号  
    cno INT NOT NULL,          --课程号  
    grade INT NULL CHECK (grade BETWEEN 0 AND 100),  
                                -- 假设成绩为百分制  
    PRIMARY KEY (sno, cno),      -- 联合主键  
    FOREIGN KEY (sno) REFERENCES Student(sno), -- 关联学生表  
    FOREIGN KEY (cno) REFERENCES Course(cno)   -- 关联课程表  
);
```

110 % ▾ ◀

消息

命令已成功完成。

完成时间: 2025-05-11T00:15:25.3229345+08:00

(4) 创建教师信息表Teacher

```
CREATE TABLE Teacher (  
    tno CHAR(10) NOT NULL PRIMARY KEY, -- 教师编号, 主键  
    tname CHAR(10) NOT NULL,          -- 教师姓名  
);
```

SQLQuery1.sql - ...DERN\62477 (54))* ✎ ✕

```
CREATE TABLE Teacher (  
    tno CHAR(10) NOT NULL PRIMARY KEY, -- 教师编号, 主键  
    tname CHAR(10) NOT NULL,          -- 教师姓名  
);
```

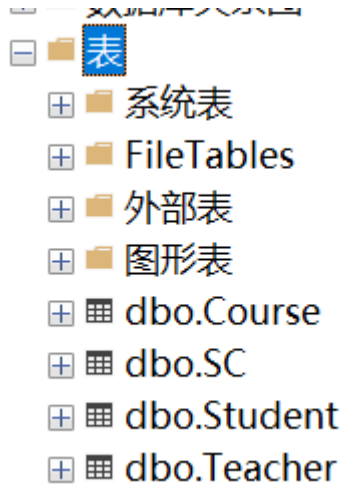
110 % ▾ ◀

消息

命令已成功完成。

完成时间: 2025-05-11T00:21:43.2374337+08:00

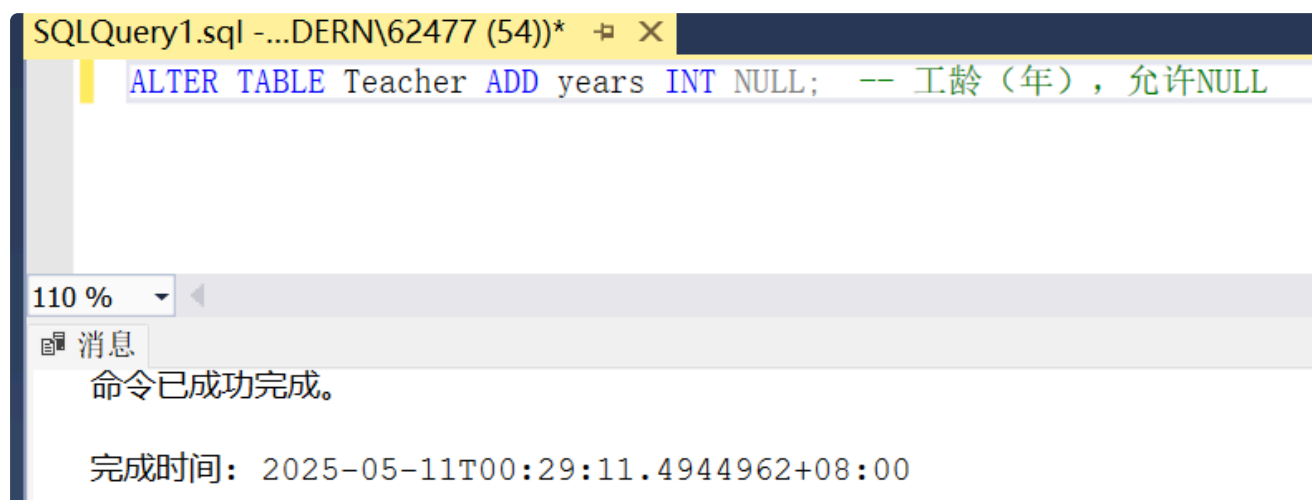
至此，已创建四个基本表。



2.修改基本表

在Teacher表中增加一列工龄years

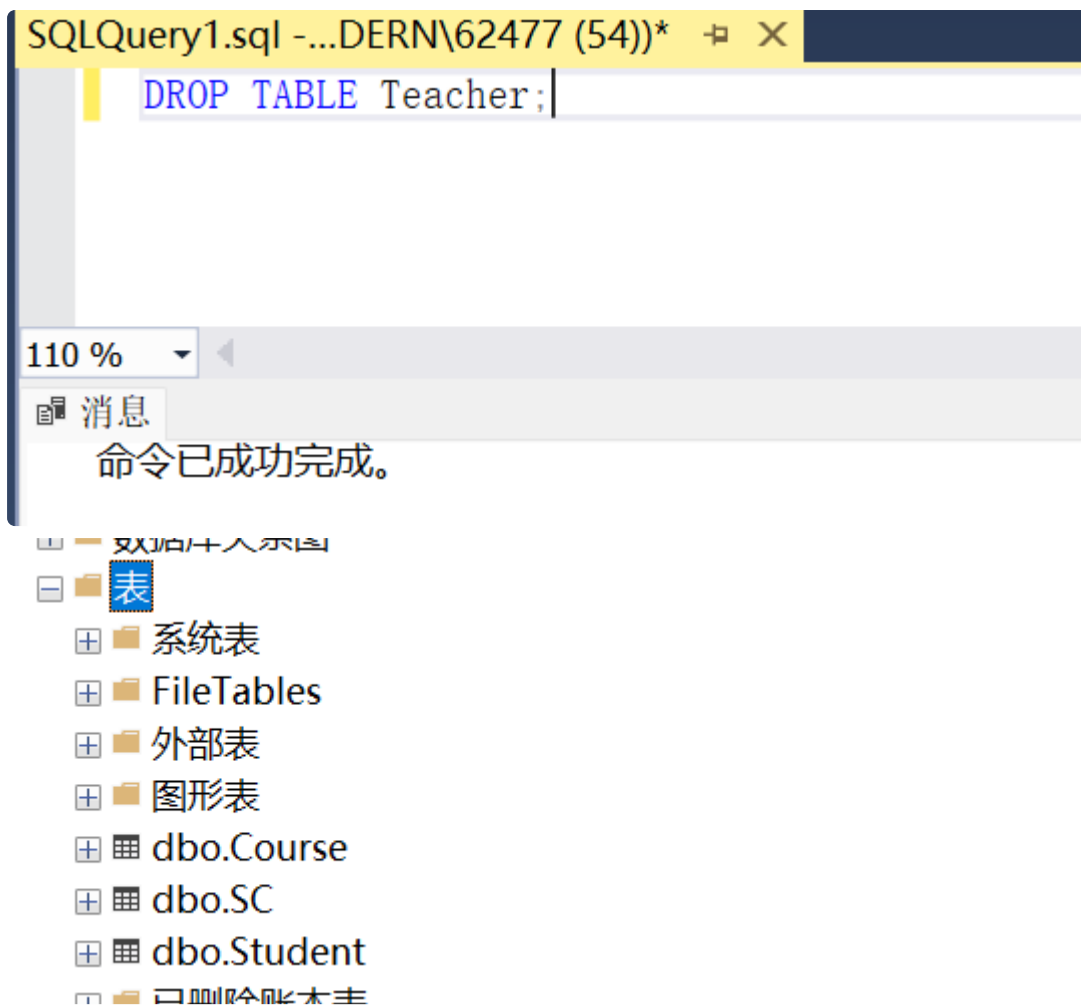
```
ALTER TABLE Teacher ADD years INT NULL; -- 工龄（年），允许NULL
```



3.删除基本表

删除Teacher表

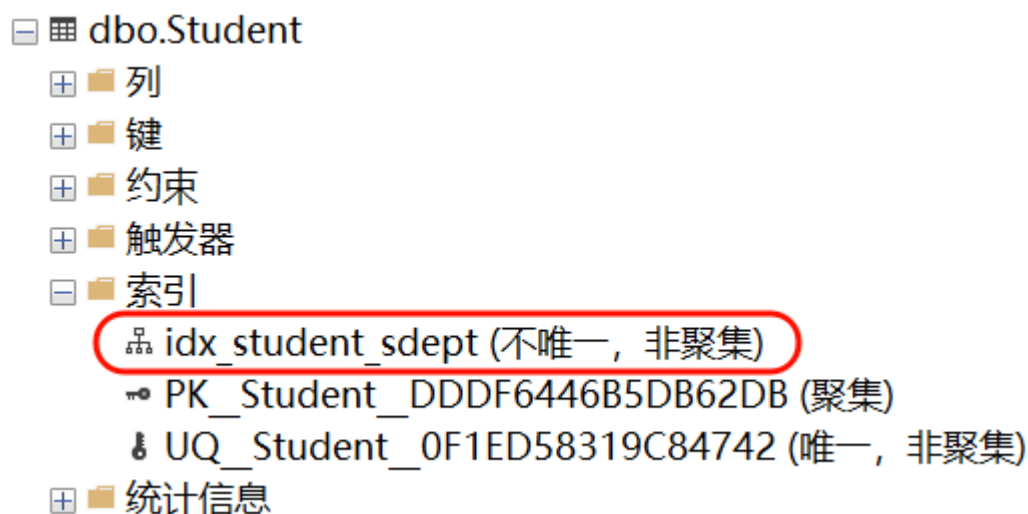
```
DROP TABLE Teacher;
```



4.创建索引

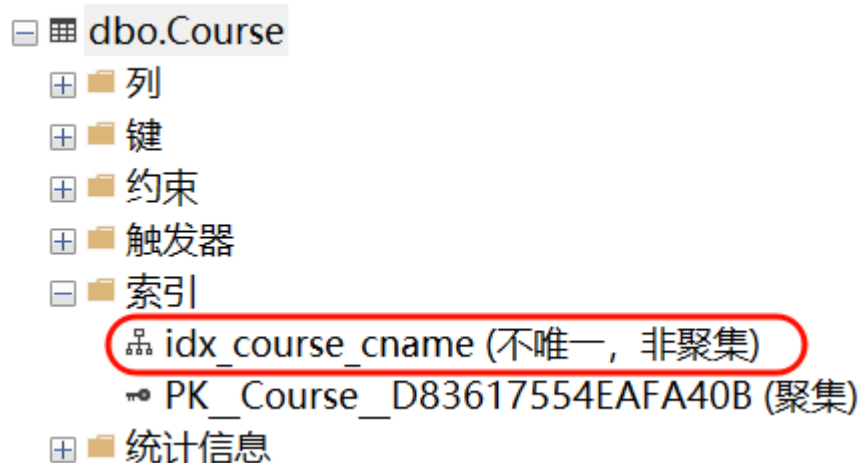
(1) 为学生表的sdept创建索引，用于按院系查询

```
CREATE INDEX idx_student_sdept ON Student(sdept);
```



(2) 为课程表的cname创建索引，用于按课程名查询

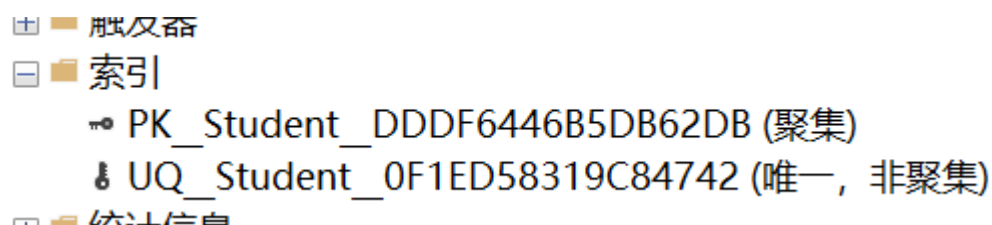
```
CREATE INDEX idx_course_cname ON Course(cname);
```

5.删除索引

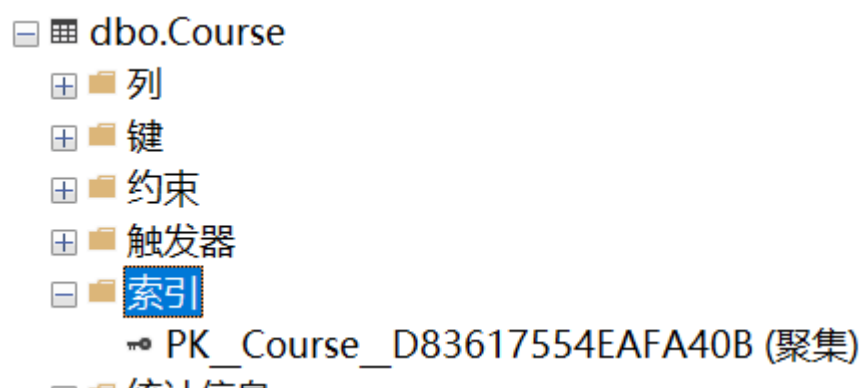
(1) 删除学生表的sdept索引

```
DROP INDEX idx_student_sdept ON Student;
```



(2) 删除课程表的cname索引

```
DROP INDEX idx_course_cname ON Course;
```



数据操作

更新操作

1.插入数据

为三个基本表插入数据

-- 插入学生数据至Student

```
INSERT INTO Student (sno, sname, ssex, sage, sdept) VALUES
('2023001', '张三', '男', 20, '计算机系'),
('2023002', '李四', '男', 21, '计算机系'),
('2023003', '王五', '女', 19, '数学系'),
('2023004', '赵六', '女', 22, '物理系'),
('2023005', '钱七', '男', 20, '数学系');
```

-- 插入课程数据至Course

```
INSERT INTO Course (cno, cname, cpno, ccredit) VALUES
(1, '数据库原理', NULL, 4), -- 无先修课程
(2, '数据结构', 1, 3), -- 先修数据库原理
(3, '高等数学', NULL, 5),
(4, '线性代数', 3, 4), -- 先修高等数学
(5, '程序设计', NULL, 3);
```

-- 插入选课成绩数据至SC

```
INSERT INTO SC (sno, cno, grade) VALUES
('2023001', 1, 85), -- 张三选修数据库原理，成绩85
('2023001', 2, 78), -- 张三选修数据结构，成绩78
('2023002', 1, 92), -- 李四选修数据库原理，成绩92
('2023002', 3, 88), -- 李四选修高等数学，成绩88
('2023003', 3, 95), -- 王五选修高等数学，成绩95
('2023003', 4, 90), -- 王五选修线性代数，成绩90
('2023004', 5, 76), -- 赵六选修程序设计，成绩76
('2023005', 3, 82), -- 钱七选修高等数学，成绩82
```

添加后查询

```
SELECT * FROM Student;  
SELECT * FROM Course;  
SELECT * FROM SC;
```

110 %

结果 消息

	sno	sname	ssex	sage	sdept
1	2023001	张三	男	20	计算机系
2	2023002	李四	男	21	计算机系
3	2023003	王五	女	19	数学系
4	2023004	赵六	女	22	物理系
5	2023005	钱七	男	20	数学系

	cno	cname	cpno	ccredit
1	1	数据库原理	NULL	4
2	2	数据结构	1	3
3	3	高等数学	NULL	5
4	4	线性代数	3	4
5	5	程序设计	NULL	3

	sno	cno	grade
1	2023001	1	85
2	2023001	2	78
3	2023002	1	92
4	2023002	3	88
5	2023003	3	95
6	2023003	4	90
7	2023004	5	76
8	2023005	3	82

2.修改数据

(1) 将学号为2023001的学生年龄改为21岁

```
UPDATE Student SET sage = 21 WHERE sno = '2023001';
```

结果		消息			
	sno	sname	ssex	sage	sdept
1	2023001	张三	男	20	计算机系
2	2023002	李四	男	21	计算机系
3	2023003	王五	女	19	数学系
4	2023004	赵六	女	22	物理系
5	2023005	钱七	男	20	数学系

结果		消息			
	sno	sname	ssex	sage	sdept
1	2023001	张三	男	21	计算机系
2	2023002	李四	男	21	计算机系
3	2023003	王五	女	19	数学系
4	2023004	赵六	女	22	物理系
5	2023005	钱七	男	20	数学系

(2) 将所有计算机系的学生转到人工智能系

```
UPDATE Student SET sdept = '人工智能系' WHERE sdept = '计算机系';
```

结果		消息			
	sno	sname	ssex	sage	sdept
1	2023001	张三	男	21	计算机系
2	2023002	李四	男	21	计算机系
3	2023003	王五	女	19	数学系
4	2023004	赵六	女	22	物理系
5	2023005	钱七	男	20	数学系

结果		消息			
	sno	sname	ssex	sage	sdept
1	2023001	张三	男	21	人工智能系
2	2023002	李四	男	21	人工智能系
3	2023003	王五	女	19	数学系
4	2023004	赵六	女	22	物理系
5	2023005	钱七	男	20	数学系

3.删除数据

删除学号为2023005的学生所有选课记录

```
DELETE FROM SC WHERE sno = '2023005';
```

	sno	cno	grade
1	2023001	1	85
2	2023001	2	78
3	2023002	1	92
4	2023002	3	88
5	2023003	3	95
6	2023003	4	90
7	2023004	5	76
8	2023005	3	82

	sno	cno	grade
1	2023001	1	85
2	2023001	2	78
3	2023002	1	92
4	2023002	3	88
5	2023003	3	95
6	2023003	4	90
7	2023004	5	76

查询操作

目前表的数据如下

结果		消息			
	sno	sname	ssex	sage	sdept
1	2023001	张三	男	21	人工智能系
2	2023002	李四	男	21	人工智能系
3	2023003	王五	女	19	数学系
4	2023004	赵六	女	22	物理系
5	2023005	钱七	男	20	数学系

	cno	cname	cpno	ccredit
1	1	数据库原理	NULL	4
2	2	数据结构	1	3
3	3	高等数学	NULL	5
4	4	线性代数	3	4
5	5	程序设计	NULL	3

	sno	cno	grade
1	2023001	1	85
2	2023001	2	78
3	2023002	1	92
4	2023002	3	88
5	2023003	3	95
6	2023003	4	90
7	2023004	5	76

1.单表查询

查询人工智能系所有学生信息

```
SELECT sno, sname, ssex, sage
FROM Student
WHERE sdept = '人工智能系';
```

```
SELECT sno, sname, ssex, sage
FROM Student
WHERE sdept = '人工智能系';
```

110 %

结果 消息

	sno	sname	ssex	sage
1	2023001	张三	男	21
2	2023002	李四	男	21

2.连接查询

(1) 查询所有学生及其选修课程的成绩（内连接）

```
SELECT S.sno, S.sname, C.cname, SC.grade
FROM Student S
JOIN SC ON S.sno = SC.sno
JOIN Course C ON SC.cno = C.cno;
```

```
SELECT S.sno, S.sname, C.cname, SC.grade
FROM Student S
JOIN SC ON S.sno = SC.sno
JOIN Course C ON SC.cno = C.cno;
```

110 %

结果 消息

	sno	sname	cname	grade
1	2023002	李四	数据库原理	92
2	2023002	李四	高等数学	88
3	2023003	王五	高等数学	95
4	2023003	王五	线性代数	90
5	2023001	张三	数据库原理	85
6	2023001	张三	数据结构	78
7	2023004	赵六	程序设计	76

(2) 查询所有课程及选修该课程的学生（左连接，显示无人选修的课程）

```
SELECT C.cname, S.sname, SC.grade
FROM Course C
LEFT JOIN SC ON C.cno = SC.cno
```

```
LEFT JOIN Student S ON SC.sno = S.sno;
```

```
SELECT C.cname, S.sname, SC.grade  
FROM Course C  
LEFT JOIN SC ON C.cno = SC.cno  
LEFT JOIN Student S ON SC.sno = S.sno;
```

110 %

结果 消息

	cname	sname	grade
1	数据库原理	张三	85
2	数据库原理	李四	92
3	数据结构	张三	78
4	高等数学	李四	88
5	高等数学	王五	95
6	线性代数	王五	90
7	程序设计	赵六	76

3.嵌套查询

(1) 查询选修了"数据结构"课程的学生信息

```
SELECT sno, sname, ssex, sage  
FROM Student  
WHERE sno IN (  
    SELECT sno  
    FROM SC  
    WHERE cno = (  
        SELECT cno  
        FROM Course  
        WHERE cname = '数据结构'  
    )  
);
```

```

SELECT sno, sname, ssex, sage
FROM Student
WHERE sno IN (
    SELECT sno
    FROM SC
    WHERE cno = (
        SELECT cno
        FROM Course
        WHERE cname = '数据结构'
    )
);

```

110 %

结果 消息

	sno	sname	ssex	sage
1	2023001	张三	男	21

(2) 查询选修了有先修课程的所有课程的学生

```

SELECT s.sno, s.sname
FROM Student s
WHERE s.sno IN (
    SELECT sc.sno
    FROM SC sc
    WHERE sc.cno IN (
        SELECT c.cno
        FROM Course c
        WHERE c.cpno IS NOT NULL
    )
);

```



```

SELECT s.sno, s.sname
FROM Student s
WHERE s.sno IN (
    SELECT sc.sno
    FROM SC sc
    WHERE sc.cno IN (
        SELECT c.cno
        FROM Course c
        WHERE c.cpno IS NOT NULL
    )
);

```

110 %

结果 消息

	sno	sname
1	2023001	张三
2	2023003	王五

4.集合查询

查询选修了课程但未选修"高等数学"的学生

```

SELECT sno FROM SC EXCEPT
SELECT sno FROM SC
WHERE cno = (SELECT cno FROM Course WHERE cname = '高等数学');

```

```

SELECT sno FROM SC EXCEPT
SELECT sno FROM SC
WHERE cno = (SELECT cno FROM Course WHERE cname = '高等数学');

```

110 %

结果 消息

	sno
1	2023001
2	2023004