

# 实验一：创建库表

## 一、实验内容

- 1 熟悉SQL Server的软件环境，了解SQL Server的功能
- 2 查看示例数据库对象。
- 3 创建STUDENT数据库，根据学生信息、课程信息和选修信息设计学生关系、课程关系、选课关系，并创建对应的基本表student、course、SC

## 二、实验步骤

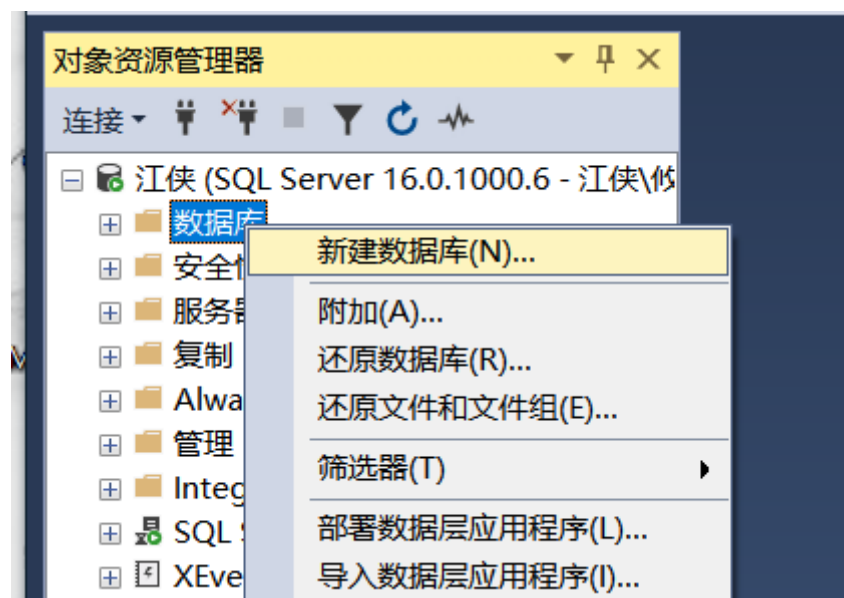
- \1. 登录SQL Server数据库管理系统
- \2. 熟悉SQL Server的操作环境，熟悉SQL Server支持的数据对象
- \3. 根据需求设计学生数据库student，并创建
- \4. 根据需求设计学生数据库的基本表student、course、SC；
- \5. 基本表录入数据；

## 三、思考题

- 1、SQL Server 数据库管理系统有哪些版本？你了解最新的SQL Server产品吗？还有哪些著名的大型数据库管理系统软件？
- 2、SQL Server 数据库管理系统的主要功能有那些？

## 1. 创建数据库 STUDENT

点击数据库，右击

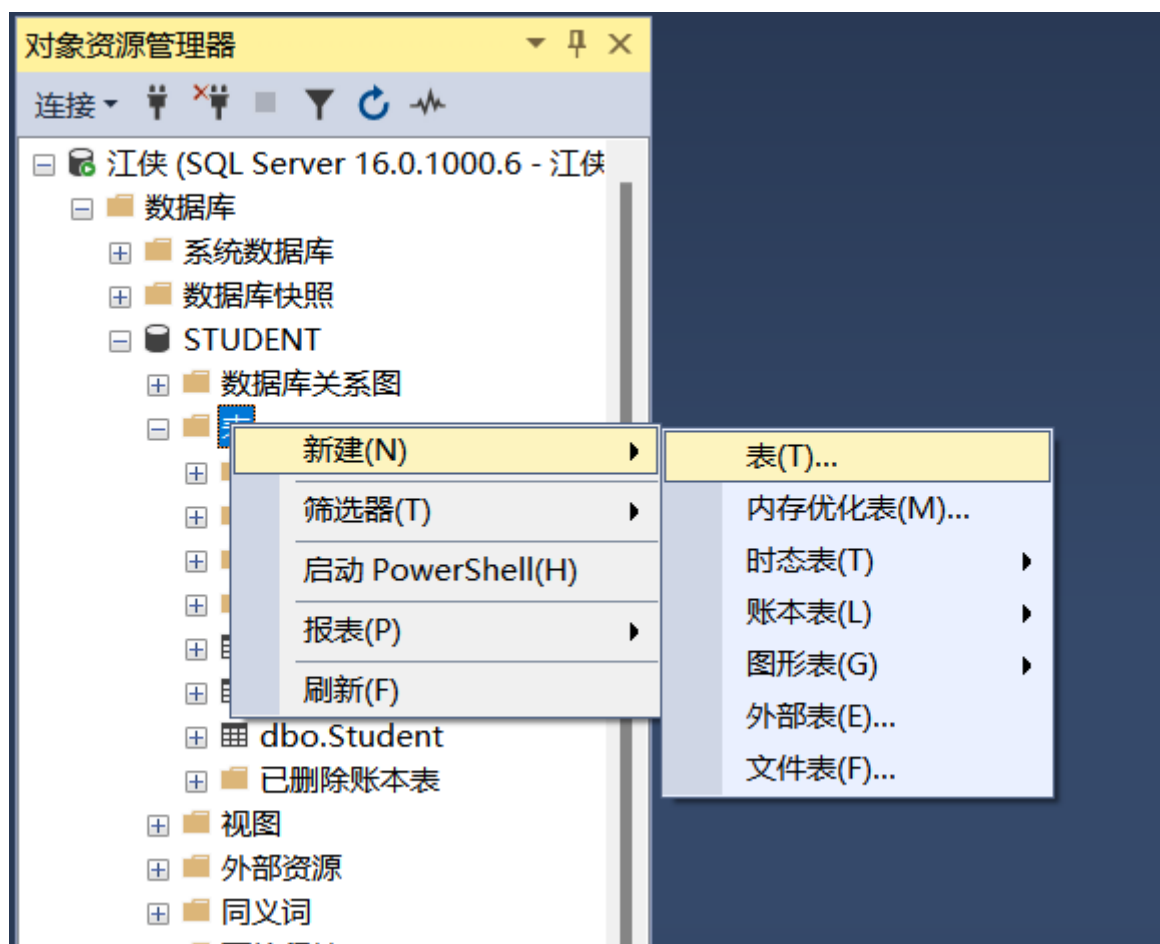


新建数据库，输入库名



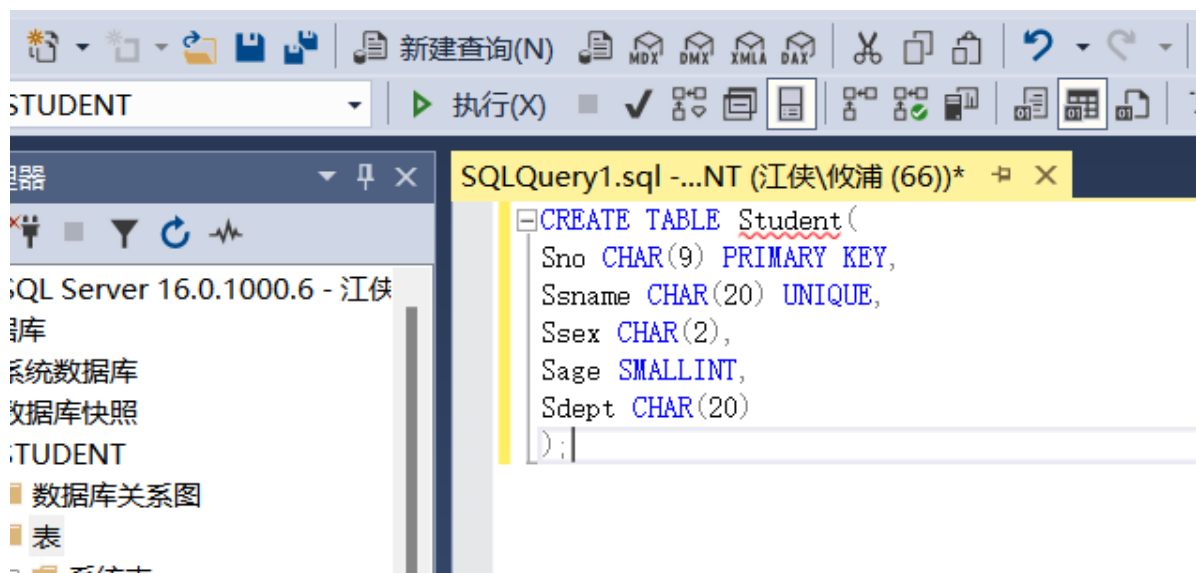
## 2. 创建Student表

通过库直接建表：



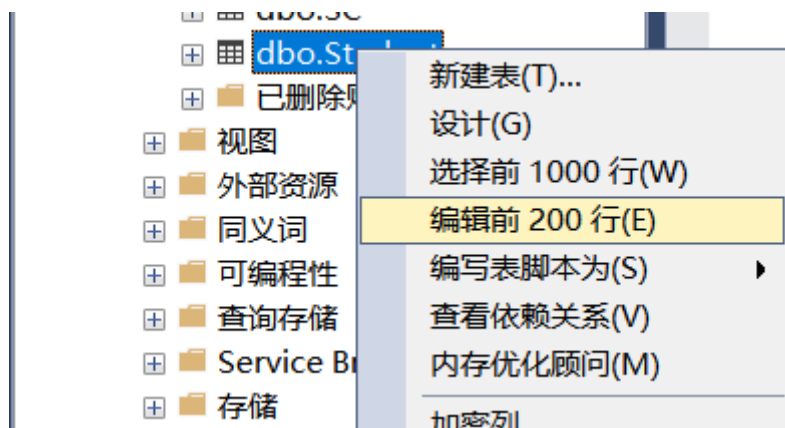
或者可以通过命令行：

点击新建查询，输入以下内容



```
CREATE TABLE Student(
Sno CHAR(9) PRIMARY KEY,
Sname CHAR(20) UNIQUE,
Ssex CHAR(2),
Sage SMALLINT,
Sdept CHAR(20)
);
```

输入数据，右键对应的表，编辑前200行：

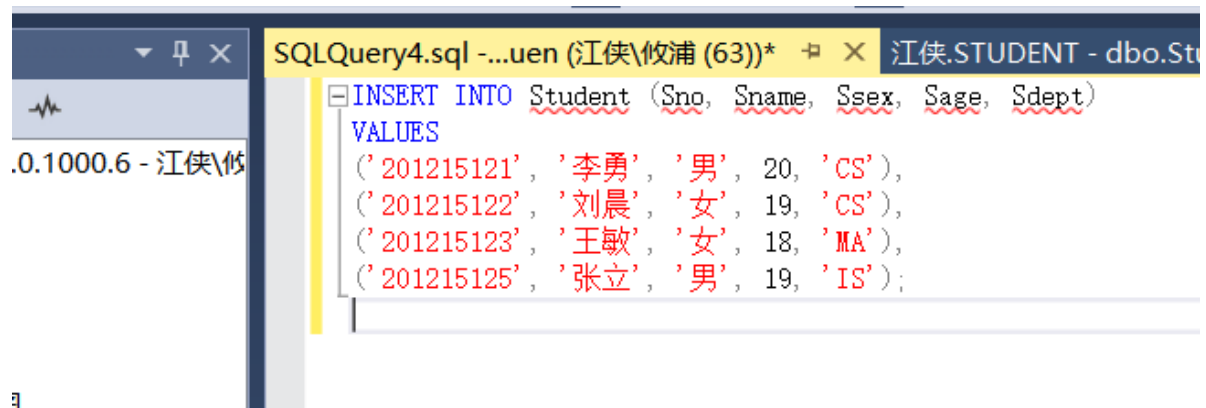


按行输入数据：

	Sno	Sname	Ssex	Sage	Sdept
▶	201215121	李勇	男	20	CS
	201215122	刘晨	女	19	CS
	201215123	王敏	女	18	MA
	201215125	张立	男	19	IS
*	NULL	NULL	NULL	NULL	NULL

或者命令行输入数据：

```
INSERT INTO Student (Sno, Sname, Ssex, Sage, Sdept)
VALUES
('201215121', '李勇', '男', 20, 'CS'),
('201215122', '刘晨', '女', 19, 'CS'),
('201215123', '王敏', '女', 18, 'MA'),
('201215125', '张立', '男', 19, 'IS');
```



### 3. 建立Course表

用命令行:

```
CREATE TABLE Course (
    Cno VARCHAR(4) PRIMARY KEY,
    Cname VARCHAR(40) NOT NULL,
    Cpno VARCHAR(4),
    Ccredit SMALLINT
);
```

然后再使用命令行:

```
USE STUDENT
GO
ALTER TABLE Course
ADD CONSTRAINT FK_Sn FOREIGN KEY(Cpno) REFERENCES Course(Cno)
```

这一步是在 `Course` 表中添加一个外键约束，`Cpno` 列现在被定义为引用 `Course` 表中 `Cno` 列的外键。这意味着 `Course.Cpno` 中的值必须在 `Course.Cno` 列中存在，确保了数据的完整性。

`Cpno` 是某个课程的前置课程编号，`Cno` 是课程编号，这个外键约束可以确保前置课程必须存在于 `Course` 表中。

输入数据:

```
INSERT INTO Course (Cno, Cname, Cpno, Ccredit)
VALUES
('1', '数据库', NULL, 5),
('2', '数学', NULL, 2),
('3', '信息系统', '1', 4),
('4', '操作系统', '6', 3),
('5', '数据结构', '7', 4),
('6', '数据处理', NULL, 2),
('7', 'PASCAL语言', '6', 4);
```

## 4. 建立SC表

使用命令行：

```
CREATE TABLE SC(
Sno CHAR(9),
Cno VARCHAR(4),
Grade SMALLINT,
PRIMARY KEY(Sno,Cno)
);
```

然后添加约束：

```
USE STUDENT;
GO
-- 添加外键约束 FK_Sno, 引用 Student(Sno)
ALTER TABLE SC
ADD CONSTRAINT FK_Sno FOREIGN KEY(Sno) REFERENCES Student(Sno);

-- 添加外键约束 FK_Cno, 引用 Course(Cno)
ALTER TABLE SC
ADD CONSTRAINT FK_Cno FOREIGN KEY(Cno) REFERENCES Course(Cno);
```

- `REFERENCES Student(Sno)`: 表示 `SC` 表中的 `Sno` 外键引用了 `Student` 表中的 `Sno` 列，要求 `SC` 表中插入的每个 `Sno` 值必须在 `Student` 表中存在。换句话说，如果在 `Student` 表中没有对应的 `Sno`，就无法将该值插入到 `SC` 表。
- `REFERENCES Student(Cno)`: 确保 `SC` 表中的 `Cno`（课程编号）必须是 `Course` 表中的有效课程编号。

输入数据：

```
INSERT INTO SC (Sno, Cno, Grade)
VALUES
('201215122', '2', 90),
('201215121', '1', 92),
('201215121', '2', 85),
('201215121', '3', 88),
('201215122', '3', 80);
```

(time: 2024.10.23)

## 实验二：SQL语言使用

## 一、实验内容

第一部分：

1 使用上次实验创建的数据库和表，利用企业管理器或查询分析器向基本表中插入实验数据。

第二部分：

2 查询全体学生学号和姓名

```
SELECT Sno, Sname  
FROM Student;
```

![[{64363A93-6A67-4C19-8EB7-27105217C37F}.png]]

3 查询全体学生学号、姓名、性别、年龄和所在系

```
SELECT Sno, Sname, Ssex, Sage, Sdept  
FROM Student;
```

![[{1160CD8E-3E27-45B5-9484-7304DE0234B9}.png]]

4 查询全体学生选课情况，即学号、课程号、成绩

```
SELECT Sno, Cno, Grade  
FROM SC;
```

![[{94D43EA0-A5EC-45BD-844E-FE4B0879CDC6}.png]]

5 显示所有选课学生的学号，并取掉重复行

```
SELECT DISTINCT Sno  
FROM SC;
```

![[{81F99A78-9F79-43D3-A807-1489C31CECD7}.png]]

6 查询成绩大于80分的学生的学号、课程号、成绩

```
SELECT Sno, Cno, Grade  
FROM SC  
WHERE Grade > 80;
```

![[{042BEA41-C1E6-4D2F-A8FD-D4DF495CD4AA}.png]]

7 查询计算机系全体学生的学号

```
SELECT Sno  
FROM Student  
WHERE Sdept = 'CS';
```

![[{03358811-F8EC-4ACD-BD53-B93BFB472F54}.png]]

8 查询计算机系年龄在18岁以上的学生的学号、姓名

```
SELECT Sno, Sname
FROM Student
WHERE Sdept = 'CS' AND Sage > 18;
```

![[{666C1C97-DAB8-4B89-8315-F99BE6B319D0}.png]]

9 查询选修了数据库课程或数据结构课程学生的学号、成绩

```
SELECT Sno, Grade
FROM SC
WHERE Cno = '1' OR Cno = '5';
```

![[{DC453D28-6DCB-465E-A9F0-368AB43DA2AE}.png]]

10 查询选修了C1课程的学生学号和成绩，查询结果按成绩降序排序

```
SELECT Sno, Grade
FROM SC
WHERE Cno = '1'
ORDER BY Grade DESC;
```

![[{9CB2FC4C-1E94-4561-8200-F23696161B58}.png]]

11 查询每个学生的学号、姓名、选修的课程名、成绩

```
SELECT Student.Sno, Student.Sname, Course.Cname, SC.Grade
FROM Student
JOIN SC ON Student.Sno = SC.Sno
JOIN Course ON SC.Cno = Course.Cno;
```

![[{8CF7170B-3ABE-4868-BFF4-02AC360A670B}.png]]

12 查询与'李勇'在同一个系学习的学生学号、姓名

```
SELECT Sno, Sname
FROM Student
WHERE Sdept = (SELECT Sdept FROM Student WHERE Sname = '李勇');
```

![[{E03C5914-17D5-4BC2-89C8-CFF88C1041C7}.png]]

13 查询选修了3门以上课程的学生学号

```
SELECT Sno
FROM SC
GROUP BY Sno //根据学号进行分组
HAVING COUNT(Cno) > 3; //筛选学号中满足特定条件的组
```

![[{000DCE9B-6209-49C8-9AF1-B81A7F42D5C3}.png]]

14 查询没有选修课程学生的基本信息

```
SELECT *
FROM Student
WHERE Sno NOT IN (SELECT Sno FROM SC);
```

![[{DFE2ED73-7DE8-478F-928A-E60C8FA17284}.png]]

15 找出平均成绩在85分以上的学生，结果按系分组，并按平均成绩的升序排列

```
SELECT Student.Sdept, Student.Sno, AVG(SC.Grade) AS AvgGrade
FROM Student
JOIN SC ON Student.Sno = SC.Sno
GROUP BY Student.Sdept, Student.Sno
//根据学生的系别（`Sdept`）和学号（`Sno`）对结果进行分组。
HAVING AVG(SC.Grade) > 85
ORDER BY AvgGrade ASC;
```

//AVG(SC.Grade) AS AvgGrade：计算每个学生的平均成绩，并将结果命名为 AvgGrade。

- 使用 JOIN 关键字将 Student 表和 SC 表连接在一起。
- ON Student.Sno = SC.Sno：指定连接条件，即通过学号（Sno）将两张表关联起来。这样可以获取每个学生的成绩信息。

![[{0D99D9FF-78AE-4C23-92C6-5FE0A8F23CFE}.png]]

16查询选修课程平均成绩在80分以上的学生学号

```
SELECT Sno
FROM SC
GROUP BY Sno
HAVING AVG(Grade) > 80;
```

![[{D3B2114E-28F3-4FF6-81F0-5761A529DB08}.png]]

17查询数据结构课程的先行课的先行课课程名

```
SELECT C2.Cname AS PrePreCourse
FROM Course C1 //指定数据源为 `Course` 表，并给它一个别名 `C1`。这个别名用于后续的引用
JOIN Course C2 ON C1.Cpno = C2.Cno
WHERE C1.Cno = '5';
```

![[{E475417D-86F6-4AA5-B7A3-EDFDC744F2F0}.png]]

18 查询同时选修《数据库》和《数据结构》两门课程的学生姓名

```
SELECT S1.Sname //择学生的姓名（`Sname`），这个字段来自于 `Student` 表，别名为 `S1`。
FROM SC AS SC1
JOIN SC AS SC2 ON SC1.Sno = SC2.Sno
JOIN Student AS S1 ON SC1.Sno = S1.Sno
WHERE SC1.Cno = '1' AND SC2.Cno = '5';
```

无

19 查询每个系的平均年龄，并建立视图

```
CREATE VIEW AvgAgeByDept AS
SELECT Sdept, AVG(Sage) AS AvgAge
FROM Student
GROUP BY Sdept;
```

```
SELECT * FROM AvgAgeByDept;
```



![[{CD21CA47-9013-41B6-9B67-0CA481614E91}.png]]

20 建立课程的先修课程的视图

```
CREATE VIEW PrerequisiteCourses AS
SELECT C1.Cno AS CourseNo, C1.Cname AS CourseName, C2.Cname AS PrerequisiteName
FROM Course C1
LEFT JOIN Course C2 ON C1.Cpno = C2.Cno;
```

```
SELECT * FROM PrerequisiteCourses;
```

![[{BF2C2249-EDB4-4CAB-922E-1A0C4190DF24}.png]]

## 二、思考题

- 1、对于实验任务的完成，SQL Server 提供了哪些方法或工具？
- 2、请考虑SQL Server 的企业管理器和查询分析器中用SQL语句两种方法完成任务的场合和优缺点。