

北京林业大学

# 数据库原理与应用

# 创建数据表



# 创建数据表

**创建数据表有两种典型方法：**

一是通过Management Studio创建；

二是通过SQL命令创建。





## 用Management Studio创建数据表

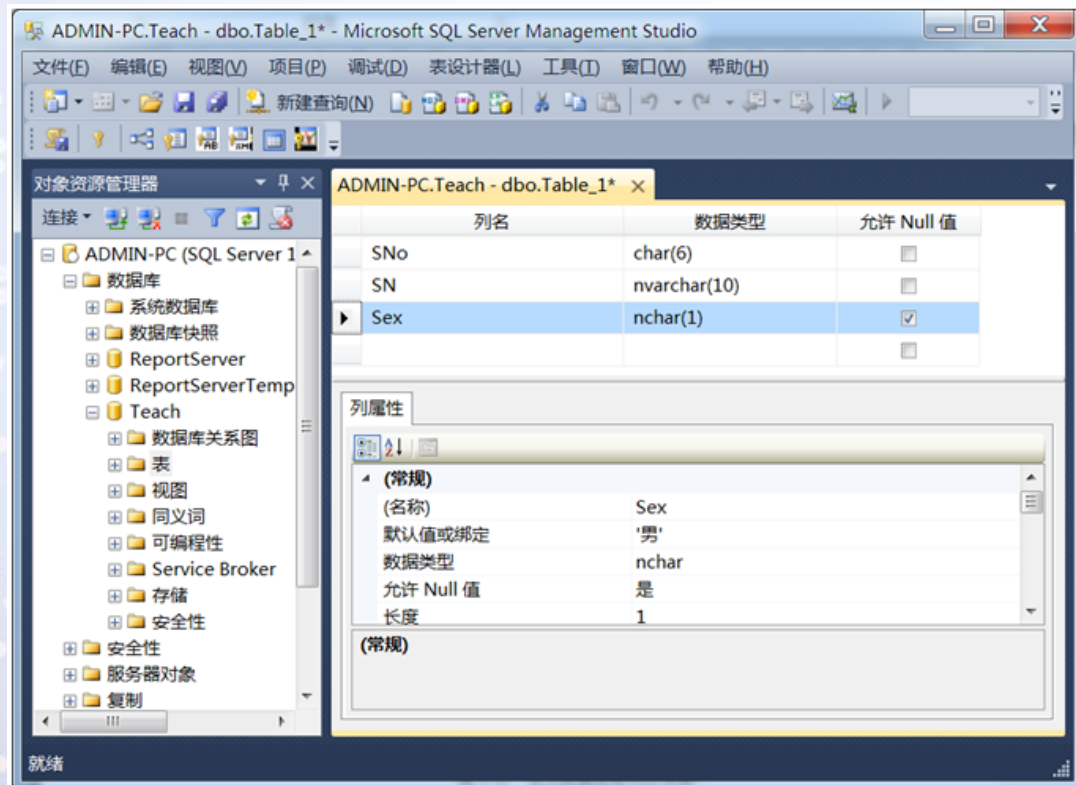
◆ 右键单击“对象资源管理器”中“数据库”节点下的“表”节点，从快捷菜单中选择“新建表”命令，会弹出定义数据表结构对话框。

◆ 其中，每一行用于定义数据表的一个字段，包括字段名、数据类型、长度、字段是否为NULL以及默认值等。



# 用Management Studio创建数据表

## ◆ 定义数据表结构对话框





# 用Management Studio创建数据表

- ◆ “列名”（即表中某个字段名）由用户命名，最长128字符，可包含中文、英文、下划线、#号、货币符号（¥）及@符号。同一表中不允许有重名的列。
- ◆ “数据类型”，定义字段可存放数据的类型。
- ◆ “允许空”，当对某个字段的“允许空”列上打勾时，表示该字段的值允许为NULL值。这样，在向数据表中输入数据时，如果没有给该字段输入数据，系统将自动取NULL值，否则，必须给该字段提供数据。
- ◆ “默认值”，表示该字段的默认值（即DEFAULT值）。如果规定了默认值，在向数据表中输入数据时，如果没有给该字段输入数据，系统自动将默认值写入该字段。





# 用SQL命令创建数据表

CREATE TABLE <表名>  
(<列定义>[{, <列定义>|<表约束>}])



<列名> <数据类型> [DEFAULT] [{<列约束>}]

- ◆ <表名>最多可有128个字符，如S、SC、C等，不允许重名。
- ◆ DEFAULT，若某字段设置有默认值，则当该字段未被输入数据时，以该默认值自动填入该字段。



## 用SQL命令创建数据表

```
CREATE TABLE <表名>  
(<列定义>[{ , <列定义> | <表约束> }])
```



```
<列名> <数据类型> [DEFAULT] [{<列约束>}]
```

[例] 用SQL命令建立一个学生表S。

```
CREATE TABLE S  
( SNo CHAR(6),  
  SN VARCHAR(10),  
  Sex NCHAR(1) DEFAULT '男' ,  
  Age INT,  
  Dept NVARCHAR(20))
```

缺省值为 “男”



# 定义数据表的约束



## 数据表的约束

在SQL Server中，对于数据表的约束分为列约束和表约束。



列约束是对某一个特定列的约束，包含在列定义中，直接跟在该列的其他定义之后，用空格分隔，不必指定列名。



表约束与列定义相互独立，不包括在列定义中，通常用于对多个列一起进行约束，与列定义用“,”分隔，定义表约束时必须指出要约束的列的名称。



# 定义数据表的约束

## 完整性约束的基本语法格式

[CONSTRAINT <约束名> ] <约束类型>



☐ NULL/NOT NULL

☐ UNIQUE

☐ PRIMARY KEY

☐ FOREIGN KEY

☐ CHECK





## NULL/NOT NULL约束

- ◆ NULL表示“不知道”、“不确定”或“没有数据”的意思
- ◆ 主键列不允许出现空值

[ CONSTRAINT <约束名> ][ NULL | NOT NULL ]

[例] 建立一个S表，对SNo字段进行NOT NULL约束。

```
CREATE TABLE S  
( SNo VARCHAR(6) CONSTRAINT S_CONS  
NOT NULL,
```

```
SN NVARCHAR(6),  
Sex NCHAR(2),  
Age INT,  
Dept NVARCHAR(20))
```

可省略约束名称：  
SNo VARCHAR(6) NOT NULL



## UNIQUE约束（唯一约束）

- ◆ 指明基本表在某一系列或多个列的组合上的取值必须唯一
- ◆ 在建立UNIQUE约束时，需要考虑以下几个因素：

- ◆ 使用UNIQUE约束的字段允许为NULL值。
- ◆ 一个表中可以允许有多个UNIQUE约束。
- ◆ UNIQUE约束用于强制在指定字段上创建一个UNIQUE索引，缺省为非聚集索引。



## UNIQUE约束 (唯一约束)

0

**UNIQUE用于定义列约束**

[CONSTRAINT <约束名>] UNIQUE

0

**UNIQUE用于定义表约束**

[CONSTRAINT <约束名>] UNIQUE  
(<列名>[{,<列名>}])





## UNIQUE约束 (唯一约束)

[例] 建立一个S表, 定义SN为唯一键。

```
CREATE TABLE S  
( SNo VARCHAR(6),  
  SN NVARCHAR(10) CONSTRAINT SN_UNIQ  
  UNIQUE,  
  Sex NCHAR(1),  
  Age INT,  
  Dept NVARCHAR(10) )
```

SN\_UNIQ可以省略  
SN NVARCHAR(10) UNIQUE,



## UNIQUE约束 (唯一约束)

**[例] 建立一个S表，定义SN+Sex为唯一键，此约束为表约束。**

```
CREATE TABLE S  
( SNo VARCHAR(6),  
  SN NVARCHAR(10) UNIQUE,  
  Sex NCHAR(1),  
  Age INT,  
  Dept NVARCHAR(20),  
  CONSTRAINT S_UNIQ UNIQUE(SN, Sex))
```



# PRIMARY KEY约束（主键约束）

## PRIMARY KEY约束（主键约束）

- ◆ 用于定义基本表的主键，起唯一标识作用

不能重复

不能为NULL

- ◆ PRIMARY KEY与UNIQUE 的区别：

在一个基本表中只能定义一个PRIMARY KEY约束，但可定义多个UNIQUE约束。

对于指定为PRIMARY KEY的一个列或多个列的组合，其中任何一个列都不能出现NULL值，而对于UNIQUE所约束的唯一键，则允许为NULL。

不能为同一个列或一组列，既定义UNIQUE约束，又定义PRIMARY KEY约束。





# PRIMARY KEY约束 (主键约束)



**PRIMARY KEY用于定义列约束**

CONSTRAINT <约束名> PRIMARY KEY



**PRIMARY KEY用于定义表约束**

[CONSTRAINT <约束名>] PRIMARY  
KEY (<列名>[{,<列名>}])



## PRIMARY KEY约束 (主键约束)

[例] 建立一个S表, 定义SNo为S的主键, 建立另外一个数据表C, 定义CNo为C的主键。

**定义数据表S:**

```
CREATE TABLE S  
( SNo VARCHAR(6) CONSTRAINT S_Prim PRIMARY KEY,  
  SN NVARCHAR(10) UNIQUE,  
  Sex NCHAR(1),  
  Age INT,  
  Dept NVARCHAR(20))
```

**定义数据表C:**

```
CREATE TABLE C  
( CNo VARCHAR(6) CONSTRAINT C_Prim PRIMARY KEY,  
  CN NVARCHAR(20),  
  CT INT)
```



## PRIMARY KEY约束 (主键约束)

[例] 建立一个SC表, 定义SNo+CNo为SC的主键。

```
CREATE TABLE SC  
( SNo VARCHAR(6) NOT NULL,  
  CNo VARCHAR(6) NOT NULL,  
  Score NUMERIC(4,1),  
  CONSTRAINT SC_Prim PRIMARY KEY(SNo,CNo))
```





# FOREIGN KEY约束 (外键约束)

主表

主键

引用

从表

外部键

[CONSTRAINT<约束名>] FOREIGN KEY  
REFERENCES

<主表名> (<列名> [{<列名>}])



## FOREIGN KEY约束 (外键约束)

**[例] 建立一个SC表, 定义SNo, CNo为SC的外部键。**

```
CREATE TABLE SC
( SNo VARCHAR(6) NOT NULL CONSTRAINT S_Fore
  FOREIGN KEY REFERENCES S(SNo),
  CNo VARCHAR(6) NOT NULL CONSTRAINT C_Fore
  FOREIGN KEY REFERENCES C(CNo),
  Score NUMERIC(4,1),
  CONSTRAINT S_C_Prim PRIMARY KEY (SNo,CNo))
```



# CHECK约束

## CHECK约束



CHECK约束用来检查字段值所允许的范围  
在建立CHECK约束时，需要考虑以下几个因素：

- ◆ 一个表中可以定义多个CHECK约束。
- ◆ 每个字段只能定义一个CHECK约束。
- ◆ 在多个字段上定义的CHECK约束必须为表约束。
- ◆ 当执行INSERT、UPDATE语句时，CHECK约束将验证数据。

[CONSTRAINT <约束名>] CHECK (<条件>)





## CHECK约束

**[例] 建立一个SC表, 定义Score的取值范围为0 ~ 100之间。**

```
CREATE TABLE SC  
( SNo VARCHAR(6),  
  CNo VARCHAR(6),  
  Score NUMERIC(4,1) CONSTRAINT Score_Chk  
  CHECK(Score>=0 AND Score <=100))
```

# 修改数据表



## 用Management Studio修改数据表的结构

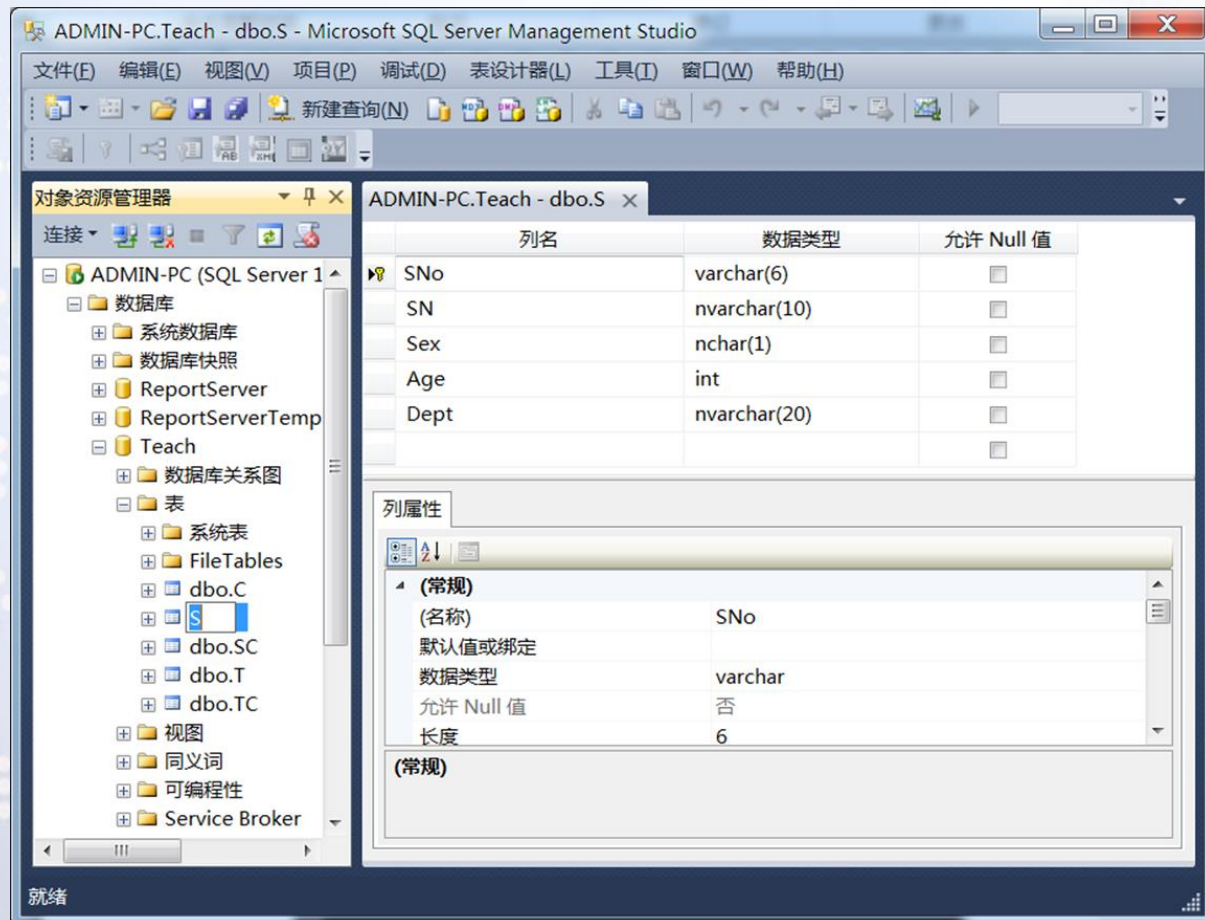
- ◆在Management Studio中的“对象资源管理器”窗口中，展开“数据库”节点。
- ◆右键单击要修改的数据表，从快捷菜单中选择“设计”命令，则会弹出修改数据表结构对话框。可以在此对话框中修改列的数据类型、名称等属性，添加或删除列，也可以指定表的主关键字约束。
- ◆修改完毕后，单击工具栏中的保存按钮，存盘退出。





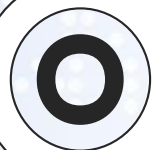
# 用Management Studio修改数据表的结构

## ◆ 修改数据表结构对话框





# 用SQL命令修改数据表结构



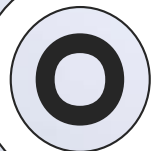
**ALTER TABLE** <表名>

**ADD** <列定义> | <完整性约束定义>



**ALTER TABLE** <表名>

**ALTER** COLUMN <列名> <数据类型>  
[NULL | NOT NULL]



**ALTER TABLE** <表名>

**DROP** CONSTRAINT <约束名>



## 用SQL命令修改数据表结构

**[例] 在S表中增加一个班号列和住址列。**

```
ALTER TABLE S  
ADD  
Class_No VARCHAR(6),  
Address NVARCHAR(20)
```

使用此方式增加的新列自动填充NULL值，所以不能为增加的新列指定NOT NULL约束。

**[例] 在SC表中增加完整性约束定义，使Score在0 ~ 100之间。**

```
ALTER TABLE SC  
ADD  
CONSTRAINT Score_Chk CHECK(Score  
BETWEEN 0 AND 100)
```





## 用SQL命令修改数据表结构

**[例] 把S表中的SN列加宽到12个字符。**

```
ALTER TABLE S  
ALTER COLUMN  
SN NVARCHAR(12)
```

- ◆ 不能改变列名;
- ◆ 不能将含有空值的列的定义修改为NOT NULL约束;
- ◆ 若列中已有数据, 则不能减少该列的宽度, 也不能改变其数据类型;
- ◆ 只能修改NULL/NOT NULL约束, 其他类型的约束在修改之前必须先将约束删除, 然后再重新添加修改过的约束定义。



## 用SQL命令修改数据表结构

**[例] 删除S表中的主键。**

```
ALTER TABLE S  
DROP CONSTRAINT S_Prim
```

# 删除数据表





## 删除基本表

○ 当某个基本表已不再使用时，可将其删除。

○ 删除后，表中的数据 and 所建的索引都被删除，建立在该表上的视图不会删除，系统将继续保留其定义，但已无法使用。

○ 如果重新恢复该表，这些视图可重新使用。



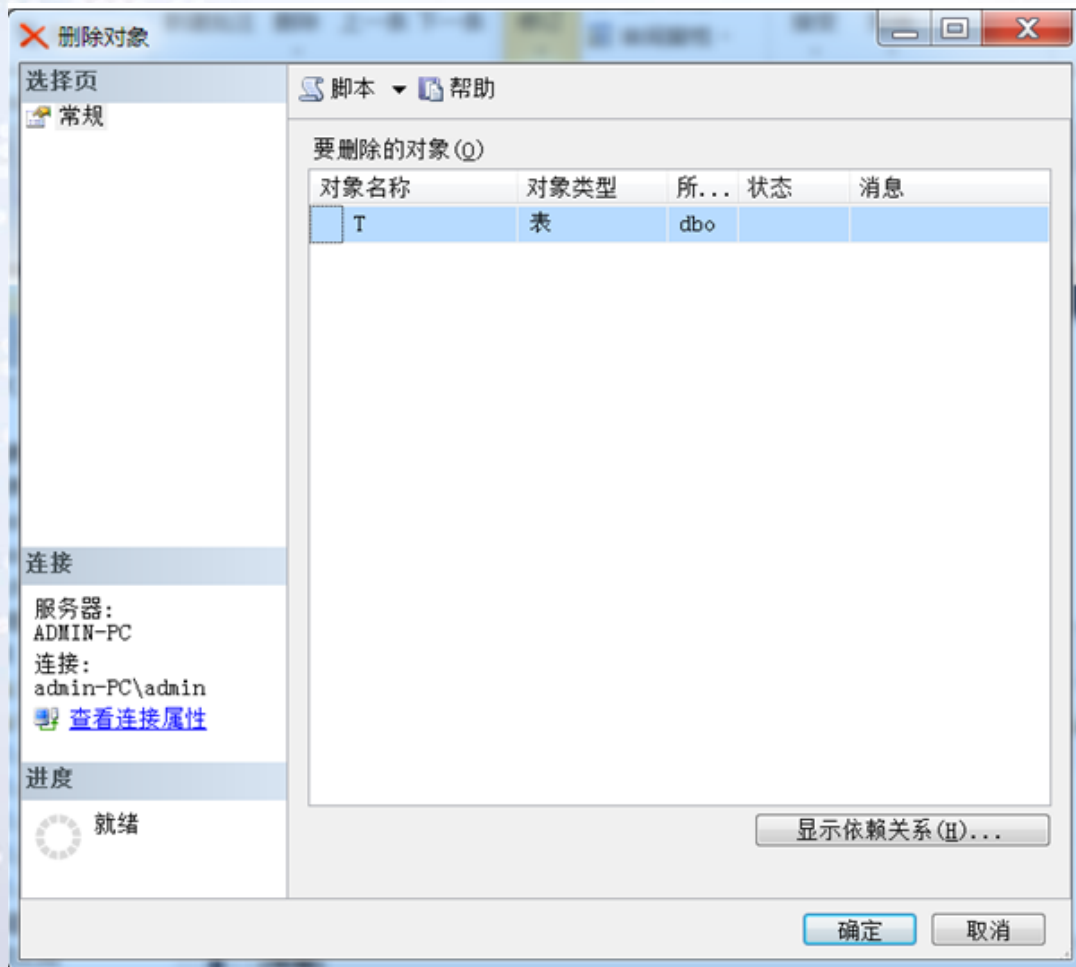
## 用Management Studio删除数据表

在Management Studio中，右键单击要删除的表，从快捷菜单中选择“删除”命令，会弹出“删除对象”对话框。

单击“显示依赖关系”按钮，即会弹出“依赖关系”对话框，其中列出了表所依靠的对象和依赖于表的对象，当有对象依赖于表时不能删除表。



# 用Management Studio删除数据表







# 用SQL命令删除数据表

DROP TABLE <表名>



只能删除自己建立的表，不能删除其他用户所建的表

# 查看数据表



## 查看数据表

### 查看数据表的属性

在Management Studio的“对象资源管理器”中展开“数据库”节点，选中相应的数据库，从中找到要查看的数据表。

右键单击该表，从快捷菜单中选择“属性”菜单项，则会弹出“表属性”对话框，从中可以看到表的详细属性信息，如表名、所有者、创建日期、文件组、记录行数、数据表中的字段名称、结构和类型等。





# 查看数据表

表属性 - S

选择页

- 常规
- 权限
- 更改跟踪
- 存储
- 扩展属性

脚本 帮助

当前连接参数

数据库	Teach
用户	admin-PC\admin
服务器	ADMIN-PC

复制

对表进行复制	False
--------	-------

说明

创建日期	2017/1/11 8:15
名称	S
系统对象	False
架构	dbo

选项

带引号的标识符	True
ANSI NULLs	True

连接

服务器: ADMIN-PC

连接: admin-PC\admin

[查看连接属性](#)

进度

就绪

名称

表的名称。

确定 取消



## 查看数据表

### 查看数据表中的数据

在Management Studio的“对象资源管理器”中，用右键单击要查看数据的表，从快捷菜单中选择“选择前1000行(W)”命令，则会显示表中的前1000条数据。



# 查看数据表

SQLQuery1.sql - ADMIN-PC.master (admin-PC\admin (56)) - Microsoft SQL Server Management Studio

文件(F) 编辑(E) 视图(V) 项目(P) 调试(D) 工具(T) 窗口(W) 帮助(H)

master

对象资源管理器

- ADMIN-PC (SQL Server 11.0.5058)
- 数据库
  - 系统数据库
  - 数据库快照
  - ReportServer
  - ReportServerTempDB
  - Teach
    - 数据库关系图
    - 表
      - 系统表
      - FileTables
      - dbo.C
      - dbo.S
      - dbo.SC
      - dbo.T
      - dbo.TC
    - 视图
    - 同义词
    - 可编程性
    - Service Broker

SQLQuery1.sql - ...in-PC\admin (56)

```
/****** Script for SelectTopNRows command from SSMS *****  
SELECT TOP 1000 [SNo]  
        , [SN]  
        , [Sex]  
        , [Age]  
        , [Dept]  
FROM [Teach]. [dbo]. [S]
```

100 %

结果 消息

	SNo	SN	Sex	Age	Dept
1	S1	赵亦	女	17	计算机
2	S2	钱尔	男	18	信息
3	S3	孙端	女	20	信息
4	S4	李思	男	21	自动化
5	S5	周武	男	19	计算机
6	S6	吴丽	女	20	自动化

查询已成功执行。 ADMIN-PC (11.0 SP2) admin-PC\admin (56) master 00:00:00 6行

就绪