

专题3

数据表与表数据操作

3.1 数据类型

3.2 表结构设计

3.3 创建数据表

3.4 管理数据表

3.5 表数据操作

3.6 数据完整性

- 关系模型中的一个关系对应数据库中的一个数据表。
- 表是最主要的数据库对象，用来存放数据库中的数据。
- 每个数据库包含了多个表。

00 数据表

由行和列组成。
列也称为字段。
行称为记录。

学生				
学号	姓名	性别	出生日期	总学分
0603170801	徐文	男	1999-1-2	30
0603180107	王刚	男	1998-9-12	45
0902170215	李莉	女	1998-3-7	48
0902180323	刘明	女	1999-10-20	32

列名或字段名

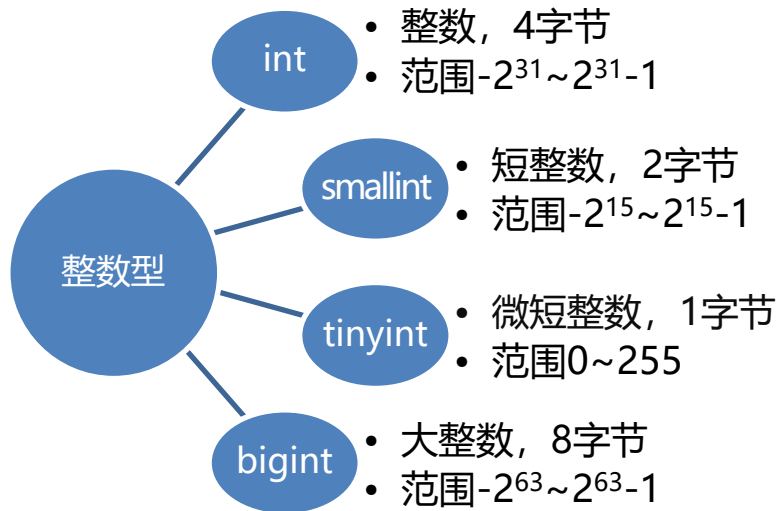
3.1

数据类型

- ➡ 数值型
- ➡ 字符型
- ➡ 二进制型
- ➡ 日期时间型

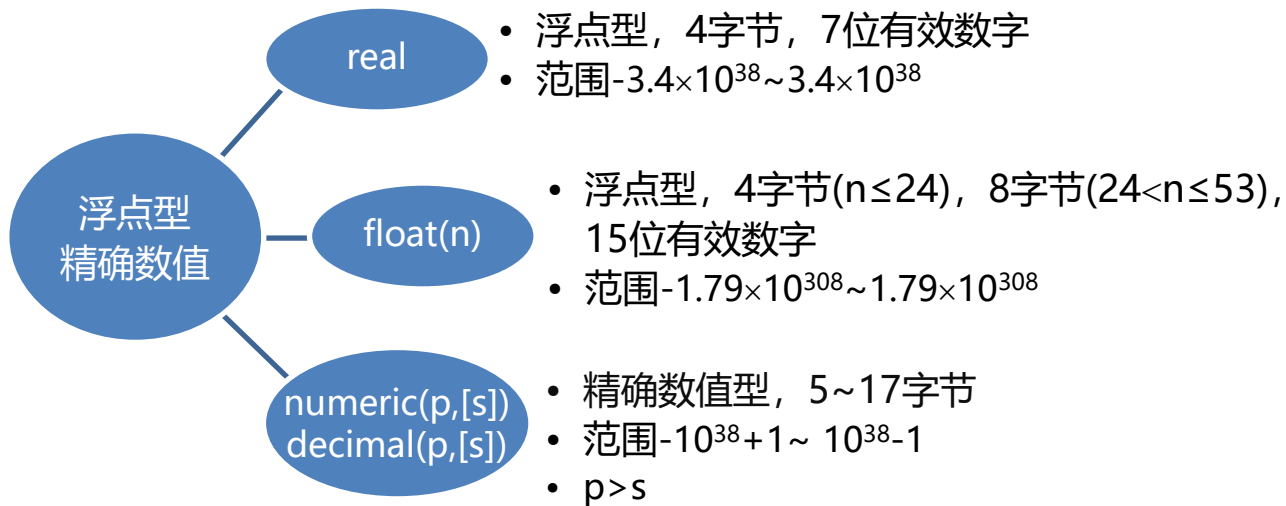
01 数值型

数值型数据类型包括整数型、浮点型与精确数值型。



01 数值型

数值型数据类型包括整数型、浮点型与精确数值型。



02 字符型

字符型数据用于存储字母、数字符号和其他符号构成的字符串，包括定长型、变长型、文本型。输入时，串中的字符需用单引号括起来。

char(n)

n个字节，n在1~8000，默认1

例如：char(6) 'abce'

a	b	c	e		
---	---	---	---	--	--

varchar(n)

n与char同，表示最大长度

例如：varchar(6) 'abc'

3	a	b	c
---	---	---	---

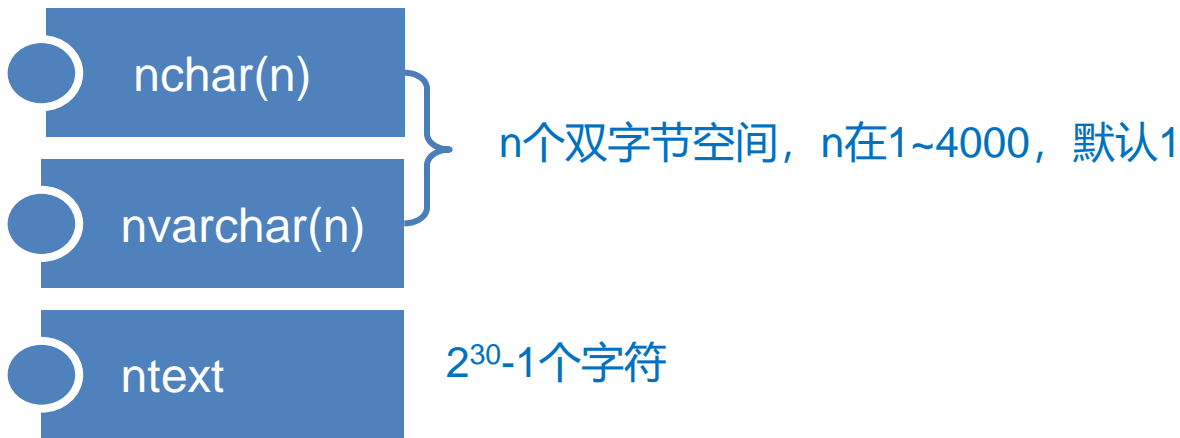
text

$2^{31}-1$ 个字符，存储长度为实际字符数

02 字符型

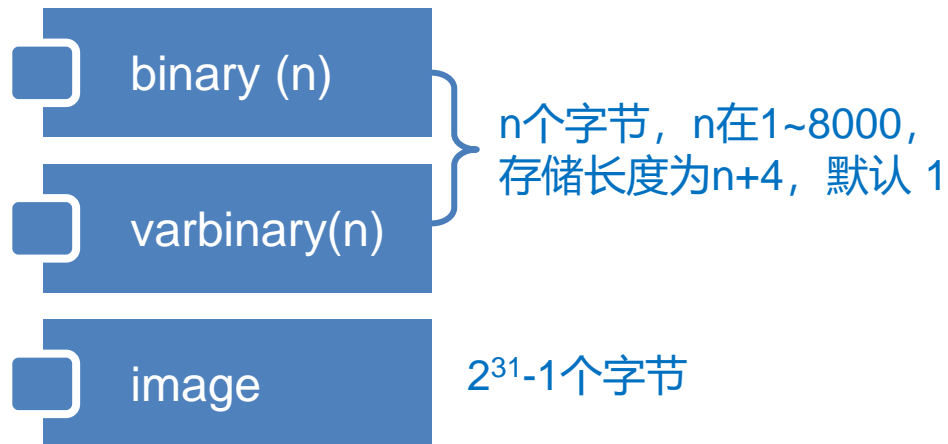
unicode字符型

unicode是“统一字符编码标准”，使用双字节给字符编码。unicode型数据类型用于unicode型数据，也包括定长型、变长型、文本型。







03 二进制型

二进制数据类型表示的是位数据流，包括定长型、变长型、图像型。
输入时十六进制数字串前需加前缀0x。



04 日期时间型

日期时间型用于存储日期和时间信息。输入时，日期时间值需用单引号括起来。

 datetime	8字节日期时间
 smalldatetime	4字节日期时间
 date	3字节日期
 time	5字节时间

05 其他类型

SQL Server提供了一些其他数据类型，下表列出部分数据类型。

类型名称	符号标识	描 述
货币型	money smallmoney	8/4字节，专门用于处理货币，十进制表示货币值
位型	bit	1字节，只存储0和1，相当于其他语言中的逻辑型数据
日期时间	datetime2 datetimeoffset	datetime数据类型的扩展 和datetime2一样，带有时区偏移量
时间戳型	timestamp	8字节，系统修改其值
游标型	cursor	用于创建游标变量或存储过程参数
唯一标识符	uniqueidentifier	16字节二进制数据，系统产生标识值
table	table	用于存储结果集的数据类型
sql_variant	sql_variant	可存储SQL Server的各种数据类型(text、ntext、image、timestamp、sql_variant数据除外)值的数据类型
xml	xml	用于保存xml文档和片段的数据类型

05 小 结

- ❖ 数据类型用于指定对象可存储的数据的类型，是字段、存储过程参数和变量等对象的数据特征。
- ❖ 数据类型决定了数据在数据库中存储空间的大小、取值范围和存储格式。
- ❖ 常用的数据类型有数值型、字符型、二进制型、日期时间型。

专题3

数据表与表数据操作

3.1 数据类型

3.2 表结构设计

3.3 创建数据表

3.4 管理数据表

3.5 表数据操作

3.6 数据完整性

3.2

表结构设计

- ➡ 表结构
- ➡ 设计表结构

01 表结构

数据表的第一行即为表结构，包含定义一个表的字段、类型、主键、外键、索引等元素。

学生				
主键	学号	姓名	性别	出生日期
	0603170801	徐文	男	1999-1-2
	0603180107	王刚	男	1998-9-12
	0902170215	李莉	女	1998-3-7
	0902180323	刘明	女	1999-10-20

日期型数据
3字节

02 设计表结构

- (1) 确定表名和字段名。名称要与用途相符，简略、直观、见名知意。
- (2) 确定字段的数据类型。
- (3) 确定字段的主键、约束。

StInfo

字段名	数据类型	长度	是否为空	默认值	说明
stid	char	10	×		主键
stname	char	10	×		
stsex	char	2	√	男	
birthdate	date	默认值	√		
telephone	varchar	15	√		
total	int	默认值	√	0	
remarks	varchar	300	√		

02 设计表结构

选课表和课程表的表结构可以采用与学生表相同的方法进行设计。

SCInfo (选课表)

字段名	数据类型	长度	是否为空	默认值	说明
stid	char	10	×		主键
cno	char	10	×		主键
score	int	默认值	√		范围0~100

CInfo (课程表)

字段名	数据类型	长度	是否为空	默认值	说明
cno	char	10	×		主键, 课程编号
cname	varchar	30	×		课程名称
ctype	char	4	√		课程类别
ccredit	int	默认值	×	0	学分
semester	tinyint	1	√	1	范围1~8,开课学期

03 小结

- ◆ 表结构设计是确定名称、数据类型、主键、约束的过程
- ◆ 表名和字段名尽量简单、见名知意；
- ◆ 定义字段的数据类型时充分考虑存放数据的类型、表现特征、需求等因素；
- ◆ 选择能唯一标识一个记录的字段作为主键。

专题3

数据表与表数据操作

3.1 数据类型

3.2 表结构设计

3.3 创建数据表

3.4 管理数据表

3.5 表数据操作

3.6 数据完整性

3.3

创建数据表

- ➡ 界面方式创建表结构
- ➡ 命令方式创建表结构

01 界面方式创建表结构

□ 使用“对象资源管理器”的表设计器创建表

例1 通过“对象资源管理器”在sems数据库中创建StInfo表。

步骤如下：

在对象资源管理器中，选择sems数据库的“表”节点，在快捷菜单中单击“新建”-“表”，打开“表设计器”

(1) 输入字段名，设置类型和字节数；

(2) 设置主键；

(3) 单击“保存”，输入StInfo表名，保存表结构。

StInfo

字段名	数据类型	长度	是否为空	说明
学号	char	10	×	主键
姓名	varchar	20	×	
性别	char	2	√	
出生日期	datetime	默认值	√	

02 命令方式创建表结构

创建数据表使用CREATE TABLE语句，语法格式：

```
CREATE TABLE [数据库名.owner.|owner.] 数据表名  
(  
    <列字义>  
    [, ... n]  
    [<表约束>]  
) .....
```

其中，<列定义>格式：

```
列名 数据类型  
[ NOT NULL ]  
[ AS 计算字段的字段值表达式 ]  
[ DEFAULT 常量表达式 ]  
[ <列约束> .....]
```

02 命令方式创建表结构

例2 在sems数据库中创建student表。

```
USE sems
GO
CREATE TABLE student
( sno      char(7) NOT NULL,
  sname    char(10),
  birthday date,
  sex      char(2)
)
```

student

字段名	数据类型	长度	是否为空
sno	char	7	×
sname	char	10	√
birthday	date	默认值	√
sex	char	2	√

02 命令方式创建表结构

例3 创建一个带计算列的表avg，包含课程的课程号、总成绩、选课人数、平均成绩字段。

```
CREATE TABLE avg
( 课程号  char(7) NOT NULL,
  总成绩  real    NOT NULL,
  人数    int     NOT NULL,
  平均成绩 AS 总成绩/人数
)
```

avg

字段名	数据类型	长度	是否为空	说明
课程号	char	7	×	
总成绩	real	默认值	×	
人数	int	默认值	×	
平均成绩	同总成绩	默认值	√	计算列

02 命令方式创建表结构

例3 创建一个带计算列的表avg，包含课程的课程号、总成绩、选课人数、平均成绩字段。

```
CREATE TABLE #avg  
( 课程号  char(7) NOT NULL,  
  总成绩  real    NOT NULL,  
  人数     int     NOT NULL,  
  平均成绩 AS 总成绩/人数  
)
```

#avg表示
本地临时表

注意：如果在表名前添加 “#” ，如#avg，该表为临时表。

- “#” 表示本地临时表
- “##” 表示全局临时表，可以由所有用户使用。临时表在断开与数据库连接时，会被服务器删除。

03 小 结

- 通过表设计器确定字段名、数据类型，非空、默认值、主键等属性。
- CREATE TABLE语句创建表结构。
- 刚创建的表只有表结构，没有数据。

专题3

数据表与表数据操作

3.1 数据类型

3.2 表结构设计

3.3 创建数据表

3.4 管理数据表

3.5 表数据操作

3.6 数据完整性

3.4

管理数据表

- ➡ 界面方式管理数据表
- ➡ 命令方式管理数据表

01 界面方式管理数据表

(1) 修改表结构

修改表结构可以增加列、删除列、修改已有列的属性。

例1 在已有student数据表中，增加列 score numeric(5,2)， sex长度设为2。

操作步骤如下：

- 在对象资源管理器中，选择 student表的快捷菜单中的“设计”，打开“表设计器”；
- 设置sex的长度为2；
- 在列表的末尾输入score，数据类型设为numeric(5,2)。
- 单击保存按钮，完成修改。

student

字段名	数据类型	是否为空
sno	char(7)	×
sname	char(10)	√
birthday	date	√
sex	char(1)	√

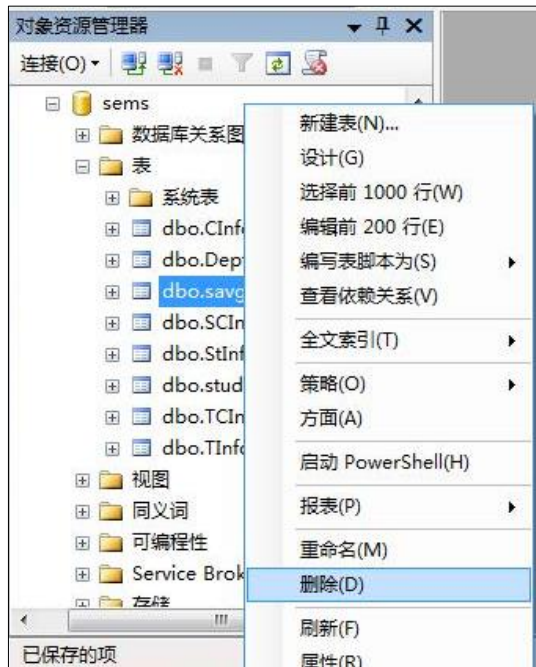
student

字段名	数据类型	是否为空
sno	char(7)	×
sname	char(10)	√
birthday	date	√
sex	char(2)	√
score	numeric (5,2)	√

(2) 删除数据表
对象资源管理器中操作。

例2 删除savg数据表。

注意：删除一个表时，该表的定义、表中的所有数据，以及表的索引、触发器、约束等均被删除；但不能删除系统表和外键约束所参照的表。



02 命令方式管理数据表

(1) 使用ALTER TABLE命令修改表结构

语法格式如下：

```
ALTER TABLE [数据库名.owner.|owner.] 数据表名
{
    ALTER COLUMN 列名 数据类型
    | ADD < 列定义 >
    | ADD 计算列名 AS 表达式 [ PERSISTED ]
    | DROP { COLUMN 列名 | 约束名 }
    .....
    [ ,...n ]
}
```

02 命令方式管理数据表

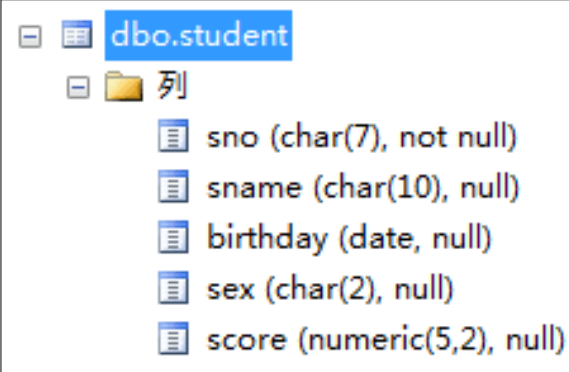
(1) 使用ALTER TABLE命令修改表结构

例3 在例1的student表中增加tel字段。

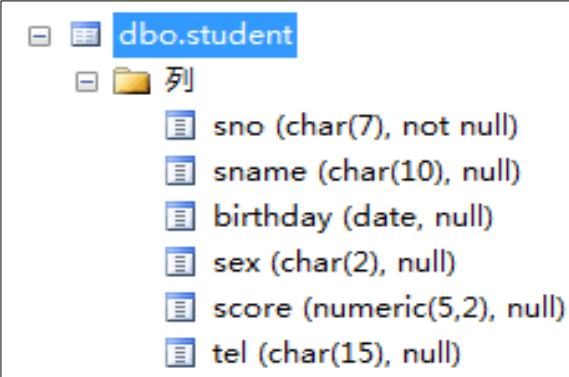
在查询窗口输入以下语句并执行：

```
ALTER TABLE student  
ADD tel char(15)
```

注意：添加列时，原表中不能有同名的列存在，否则语句运行时会出错。



dbo.student	
列	
sno	(char(7), not null)
sname	(char(10), null)
birthday	(date, null)
sex	(char(2), null)
score	(numeric(5,2), null)



dbo.student	
列	
sno	(char(7), not null)
sname	(char(10), null)
birthday	(date, null)
sex	(char(2), null)
score	(numeric(5,2), null)
tel	(char(15), null)

02 命令方式管理数据表

(1) 使用ALTER TABLE命令修改表结构

例4 将student表中sname列的长度为改为16;
将birthday列的数据类型改为smalldatetime。

在查询窗口输入以下语句并执行:

```
ALTER TABLE student
```

```
    ALTER COLUMN sname char(16)
```

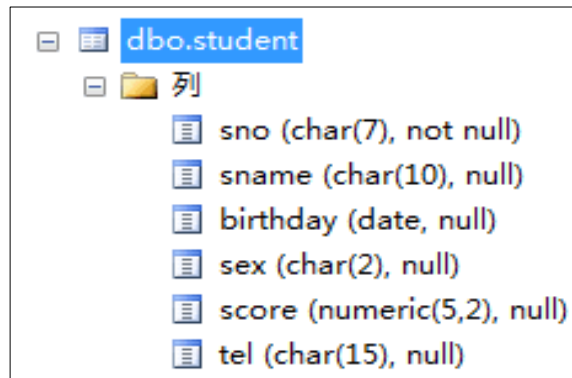
```
GO
```

```
ALTER TABLE student
```

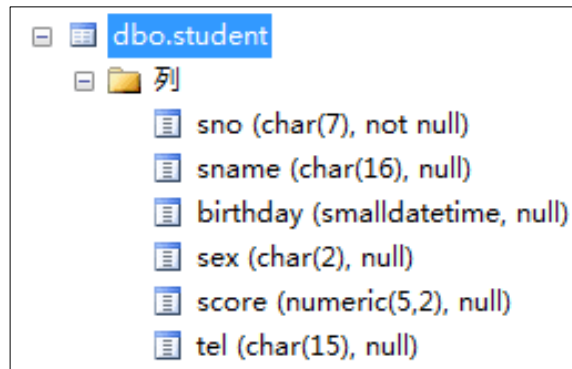
```
    ALTER COLUMN birthday smalldatetime
```

```
GO
```

注意: 使用ALTER COLUMN子句时, 一次只能修改一个列的属性。



dbo.student	
列	
sno	(char(7), not null)
sname	(char(10), null)
birthday	(date, null)
sex	(char(2), null)
score	(numeric(5,2), null)
tel	(char(15), null)



dbo.student	
列	
sno	(char(7), not null)
sname	(char(16), null)
birthday	(smalldatetime, null)
sex	(char(2), null)
score	(numeric(5,2), null)
tel	(char(15), null)

02 命令方式管理数据表

(1) 使用ALTER TABLE命令修改表结构

例5 删除student表中birthday字段。

使用以下语句：

```
ALTER TABLE student  
    DROP COLUMN birthday
```

注意：在ALTER TABLE 语句中，一次只能包含ALTER COLUMN、ADD、DROP子句中的一项。

02 命令方式管理数据表

(2) 删除数据表

使用DROP TABLE命令，语法格式如下：

```
DROP TABLE [数据库名.owner.|owner.] 数据表名 [ ,...n ]
```

例6 删除当前数据库已存在book表和sale表。

使用以下语句：

```
DROP TABLE  book, sale
```

03 小 结

- 在表设计器中可以修改列名、数据类型，以及非空、默认值、主键等属性；
- 修改表结构可以使用ALTER TABLE语句。ADD、ALTER COLUMN、DROP 子句分别用于添加、修改、删除列。
- DROP TABLE语句用于删除数据表。

专题3

数据表与表数据操作

3.1 数据类型

3.2 表结构设计

3.3 创建数据表

3.4 管理数据表

3.5 表数据操作

3.6 数据完整性

3.5

表数据操作

- ➡ 界面方式操作表数据
- ➡ INSERT语句插入记录
- ➡ UPDATE语句修改记录
- ➡ DELETE/TRUNCATE TABLE语句删除记录

01 界面方式操作表数据

使用表数据窗口。

例1 对已有的student表进行以下操作：

- 添加下表记录；
- 将第3条记录的sname字段值修改为“金铭”；
- 删除第2条记录。

student

sno	sname	birthday	sex
0617108	徐文	1999-9-8	男
0617211	曾莉娟		
2016206	金萍	1998-11-12	女

02 命令方式操作表数据

(1) 插入记录

使用INSERT语句，语法格式如下：

```
INSERT [INTO] 表名 [(字段列表)]  
VALUES (数据列表) [ ,...n ]
```


02 命令方式操作表数据

(1) 插入记录

例2 向sems数据库的表CInfo插入以下一行数据:

9720046, 多媒体技术与应用, 选修, 3, NULL

在新建查询窗口中输入以下语句:

```
USE sems
```

```
GO
```

```
INSERT INTO CInfo
```

```
VALUES ('9720046', '多媒体技术与应用', '选修', 3, NULL)
```

或

```
INSERT CInfo (CNo,CName,CCredit)
```

```
VALUES ('9720046','多媒体技术与应用',3)
```

注意: 必须是允许NULL列插入时才能为NULL

02 命令方式操作表数据

(1) 插入记录

例3 向CInfo表插入以下数据行：

9720047, 大数据及应用, 2

9720048, 移动应用开发, 2

使用以下语句：

```
INSERT CInfo (CNo,CName,CCredit)
```

```
VALUES ('9720047', '大数据及应用', 2) ,  
      ('9720048', '移动应用开发', 2)
```

CNo	CName	CType	CCredit	Hours
1805012	大学英语	必修	6	96
.....				
9720045	Web开发技术	选修	3	48
9720046	多媒体技术与应用	NULL	3	NULL
9720047	大数据及应用	NULL	2	NULL
9720048	移动应用开发	NULL	2	NULL

02 命令方式操作表数据

(2) 修改记录

使用UPDATE语句，语法格式如下：

UPDATE 表名

SET { 列名 = 表达式 [, ...n] }

[FROM { <表源> } [, ...n]]

[WHERE <查询条件>]

02 命令方式操作表数据

(2) 修改记录

例4 将当前数据库的CInfo表中课程号为“9720047”的课程的CType值修改为“必修”，同时将CCredit+1。

使用以下语句：

```
UPDATE CInfo  
SET CType='必修' ,  
    CCredit=CCredit+1  
WHERE CNO ='9720047'
```

CNo	CName	CType	CCredit	Hours
1805012	大学英语	必修	6	96
.....				
9720045	Web开发技术	选修	3	48
9720046	多媒体技术与应用	NULL	3	NULL
9720047	大数据及应用	必修	3	NULL
9720048	移动应用开发	NULL	2	NULL

02 命令方式操作表数据

(3) 删除记录

可以使用DELETE语句或TRUNCATE TABLE语句来实现。

DELETE删除表中符合条件的记录，语句格式如下：

```
DELETE [ FROM ] 表名  
[ WHERE <查询条件> ]
```

02 命令方式操作表数据

(3) 删除记录

例5 删除当前数据库的CInfo表中CType值为NULL的记录。

使用以下语句：

```
DELETE FROM CInfo  
WHERE CType IS NULL  
      CType = NULL  
          x
```

02 命令方式操作表数据

(3) 删除记录

可以使用DELETE语句或TRUNCATE TABLE语句来实现。

TRUNCATE TABLE删除表中所有记录，语句格式如下：

TRUNCATE TABLE 表名

例6 删除当前数据库的Cinfo1和Cinfo2表中的所有记录。

在新查询窗口中输入以下语句并执行：

```
TRUNCATE TABLE Cinfo1
```

```
DELETE Cinfo2
```

```
SELECT * FROM Cinfo1
```

```
SELECT * FROM Cinfo2
```

```
DROP TABLE Cinfo3
```

```
SELECT * FROM Cinfo3
```

03 小 结

- ◆ 在表数据窗口中可以对数据进行插入、删除、修改操作。
- ◆ INSERT语句用于添加记录;
UPDATE语句用于修改记录;
DELETE、TRUNCATE TABLE语句
用于删除数据表。

专题3

数据表与表数据操作

3.1 数据类型

3.2 表结构设计

3.3 创建数据表

3.4 管理数据表

3.5 表数据操作

3.6 数据完整性

3.6

数据完整性

- ➔ 完整性约束
- ➔ 实体完整性
- ➔ 域完整性
- ➔ 参照完整性

01 完整性约束

(1) 完整性

数据库中的数据在逻辑上的一致性和准确性。

学生表

学号	姓名	性别	出生日期
0603170801	徐文	男	1999-1-2
0603180107	王刚	人×	1998-9-12
0902170215	李莉	女	1998-3-7
0902180323	刘明	女	1999-10-20

选课表

学号	课程号	分数
0603170801	9710011	67
0603180102	9710021	88
0603180107	9710011	92
0603180107	9710021	78

例如，学生表中性别字段取值只能为“男”或“女”；同样，选课表中“0603180102”的学号没有出现在学生表中，表明学校没有这个学生，不能有成绩绩，不符合数据的完整性。

01 完整性约束

(2) 约束

约束定义关于列中允许值的规则，是强制完整性的标准机制。

约束可以作用的对象：

- 表
- 记录
- 字段

(2) 约束

约束可分为6种类型：

- 主键约束 (PRIMARY KEY)
 - 唯一约束 (UNIQUE)
 - 检查约束 (CHECK)
 - 默认约束 (DEFAULT)
 - 非空约束 (NOT NULL)
 - 外键约束 (FOREIGN KEY) → 参照完整性
- 实体完整性
- 域完整性或

02 实体完整性

又称为行完整性，就是要求每个数据表都必须有主键，其值不能为空，且能唯一地标识对应的记录。

实体完整性的实现：

- 主键约束
- 唯一约束

02 实体完整性

例1 对student表的“身份证号”列定义UNIQUE 约束。

(1) 定义唯一约束的方法：

- 在student的“表设计器”窗口中，选择sID快捷菜单的“索引/键”，打开“索引/键”对话框，单击“添加”按钮
- 在“标识”区域的名称栏中输入IX_sID，在“常规”区域的“类型”栏中选择“唯一键”，“列”栏中选择“sID 升序”，“是唯一的”选择“是”。
- 保存设置。

(2) 删除约束在“索引/键”对话框中选择约束，单击“删除”按钮即可完成。

student

字段名	数据类型	约束	说明
sno	char(10)	主键	学号
sname	varchar(20)		姓名
sex	char(2)		性别
birthday	date		出生日期
sID	char(18)	唯一	身份证号

02 实体完整性

主键约束与唯一约束的区别：

约束名	个数	为空	索引
主键约束	1个	不能为空	聚集索引
唯一约束	多个	可以为空	非聚集索引

02 实体完整性

命令方式创建或删除约束。

创建表语法格式：

CREATE TABLE 表名

(
 {<列名> <数据类型> [**<列约束>**]} [, ... n]
 [{**<表约束>**} [, ... n]]
)

列约束是对某一个特定列的约束，包含在列定义中

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

02 实体完整性

命令方式创建或删除约束。

列约束的语法格式：

[CONSTRAINT <约束名>] <约束类型>

PRIMARY KEY

UNIQUE

CHECK (<条件表达式>)

DEFAULT <值>

NOT NULL

FOREIGN KEY (外键) REFERENCES(参照列)

02 实体完整性

命令方式创建或删除约束。

表约束的语法格式：

[CONSTRAINT <约束名>] <约束类型>

PRIMARY KEY (列名)

UNIQUE (列名)

CHECK (<条件表达式>)

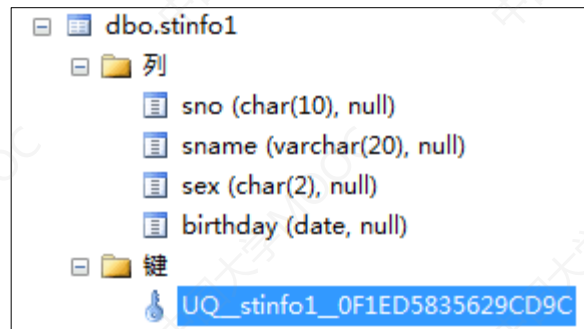
FOREIGN KEY (外键) REFERENCES (参照列)

02 实体完整性

例2 创建stinfo1，对sname字段定义唯一约束。

使用以下语句：

```
CREATE TABLE stinfo1  
( sno char(10),  
  sname varchar(20) UNIQUE ,  
  sex char(2),  
  birthday date  
)
```



02 实体完整性

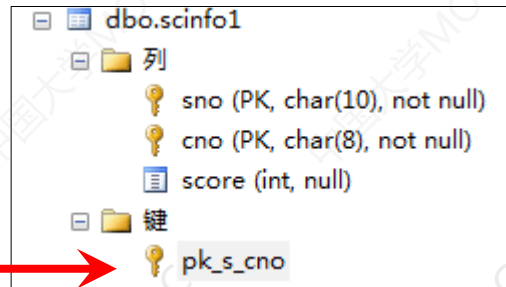
例3 修改scinfo1表，定义sno与cno属性组为表的主键。

使用以下语句：

```
ALTER TABLE scinfo1
```

```
ADD CONSTRAINT pk_s_cno PRIMARY KEY (sno,cno)
```

ADD子句用于
添加约束



02 实体完整性

例4 删除stinfo1表中的唯一约束uq_name。

使用以下语句：

```
ALTER TABLE stinfo1
```

```
DROP CONSTRAINT uq_name
```



DROP子句用于删除约束

03 域完整性

也称为列完整性，是对数据表中字段属性的约束，检验给定列输入的有效性，是用户自定义完整性的主要内容。

域完整性的实现：

- 数据类型
- 非空约束
- 默认约束
- 检查约束

```
CREATE TABLE stinfo1  
( sno char(10) NOT NULL ,  
  sname varchar(20) ,  
  sex char(2) DEFAULT '男' ,  
  birthday date  
)
```

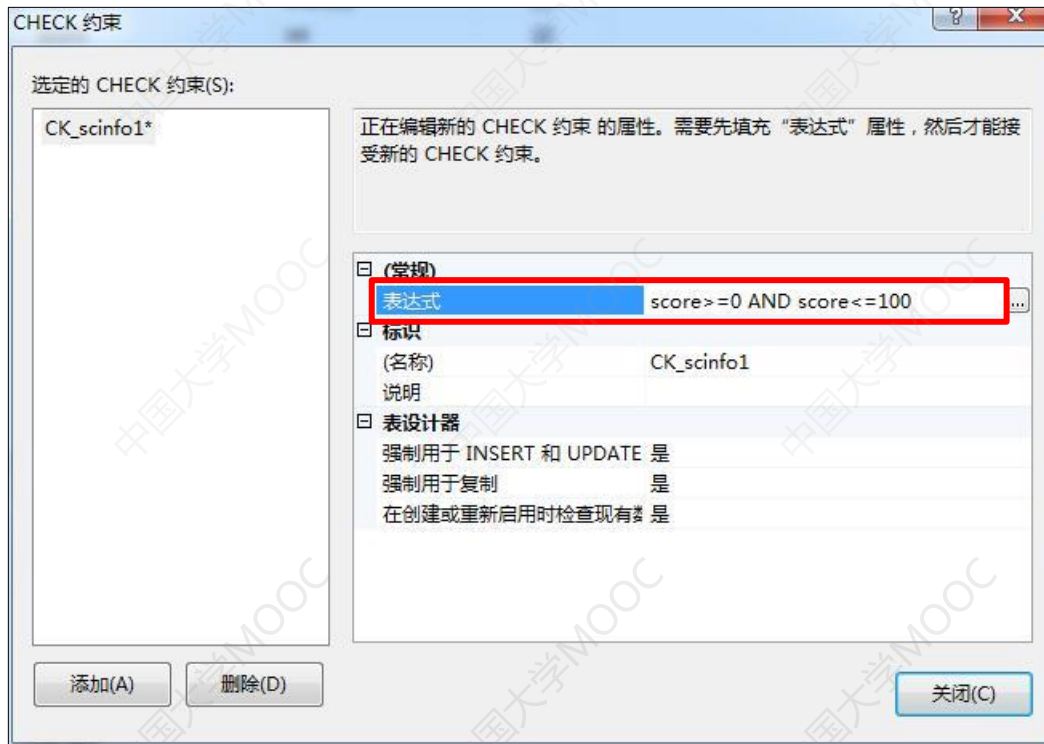
以上语句中定义sno字段NOT NULL，定义sex字段的默认值为“男”，这样就限制可以数据输入的类型、格式或可能的取值范围。

03 域完整性

例5 设置scinfo1表的score字段的取值范围，以保证学生每门功课的成绩范围为0~100分。

方法如下：

- 单击对象资源管理器的scinfo1表“约束”的快捷菜单中的“新建约束”命令。
- 在打开的“CHECK约束”对话框中，已添加一个名为CK_scinfo1的CHECK约束，在“常规”区域的“表达式”栏中输入“score>=0 AND score<=100”约束表达式。
- 单击“关闭”完成约束设置。

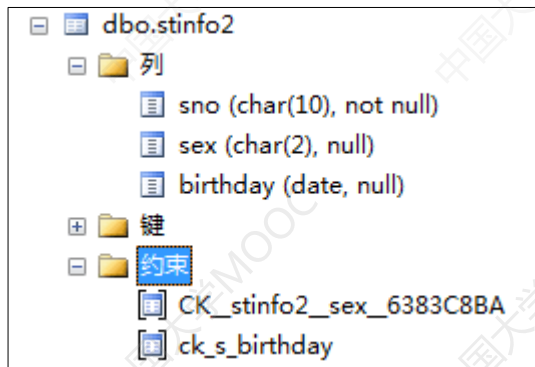


03 域完整性

例6 创建stinfo2表，只考虑学号、性别、出生日期三列，性别只能包含“男”或“女”，出生日期必须大于1998年1月1日，并命名约束为ck_s_birthday。

使用以下语句：

```
CREATE TABLE stinfo2  
( sno char(10) NOT NULL ,  
  sex char(2) CHECK (sex IN ('男', '女')) ,  
  birthday date CONSTRAINT ck_s_birthday CHECK (birthday>'1998-01-01')  
)
```



04 参照完整性

又称为引用完整性，用于保证主表中的数据与从表中数据的一致性。

参照完整性通过定义外键与主键之间的对应关系来实现：

- FOREIGN KEY 定义从表的外键
- PRIMARY KEY定义主表的主键
或者UNIQUE定义主表中的唯一键

04 参照完整性

例7 实现stinfo表和scinfo表的参照完整性。

stid	stname	sex	birthdate
0603170801	徐文	男	1999-01-02
0603180107	王刚	男	1998-09-12
0902170215	李莉	女	1998-03-07
0902180323	刘明	女	1999-10-20

stid	cno	score
0603170801	9710011	67
0603180101	9710021	88
0603180107	9710011	92
0603180107	9710021	78

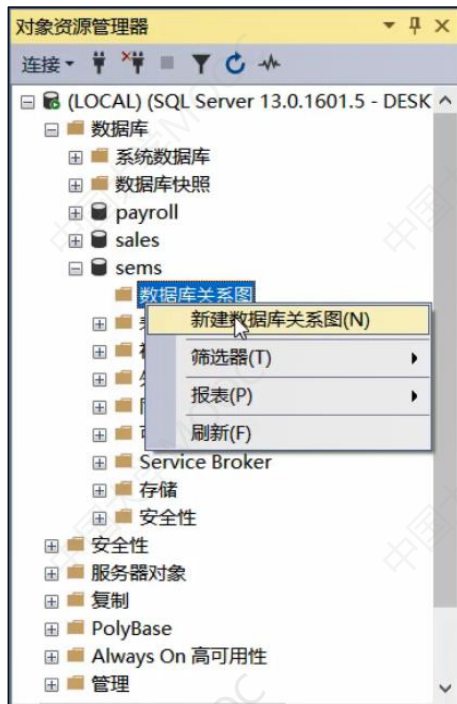
分析：对于学生表的每一个学号，在选课表中都有相关的课程成绩记录，stinfo表为主表，其stid定义为主键，scinfo表为从表，其stid定义为外键。从而建立主表与从表之间的联系，确保两表的stid字段值是一致的。

04 参照完整性

例7 实现stinfo表和scinfo的参照完整性。

操作步骤如下：

- 在对象资源管理器的sems数据库节点下，选择“数据库关系图”，单击快捷菜单的“新建数据库关系图”，打开“添加表”窗口，添加stinfo和scinfo表。
- 在“数据库关系图”设计窗口中，鼠标拖动主表主键到scinfo的stid上。打开“表和列”对话框，输入关系名，设置主键表和列名、外键表列名。两次单击“确定”按钮，出现两表连接形式，钥匙方为主键，另一方为外键。



04 参照完整性

例8 创建教师表tinfo，要求tinfo表中所有的院系编号都出现在dinfo表中，假设已经在dinfo中创建dno为主键。

tinfo

字段名	数据类型	约束	说明
tid	char(10)	主键	教号
tname	varchar(20)		姓名
tsex	char(2)		性别
dno	char(2)	外键	院系编号

dinfo

字段名	数据类型	约束	说明
dno	char(2)	主键	院系编号
dname	varchar(30)		院系名称
dean	char(10)		负责人

04 参照完整性

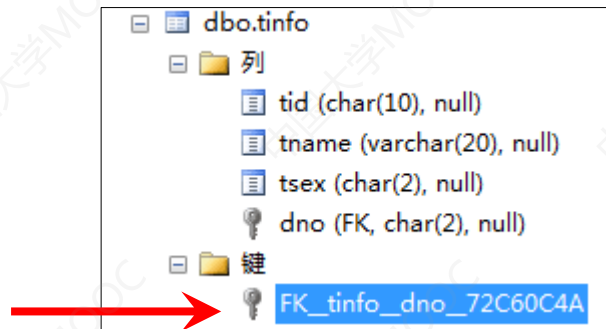
例8 创建教师表tinfo，要求tinfo表中所有的院系编号都出现在dinfo表中，假设已经在dinfo中创建dno为主键。

使用以下语句：

```
CREATE TABLE tinfo
```

```
( tid char(10),  
  tname varchar(20),  
  tsex char(2),
```

```
  dno char(2) FOREIGN KEY(dno) REFERENCES dinfo(dno)  
)
```



04 参照完整性

例9 创建成绩表grade，要求grade表中所有的学号stid都出现在stinfo表中，课程编号cno都出现在cinfo表中，假设cno和stid分别是cinfo、stinfo的主键。

stinfo

字段名	数据类型	约束	说明
stid	char(10)	主键	学号
stname	varchar(20)		姓名

cinfo

字段名	数据类型	约束	说明
cno	char(10)	主键	课程编号
cname	varchar(30)		课程名称

grade

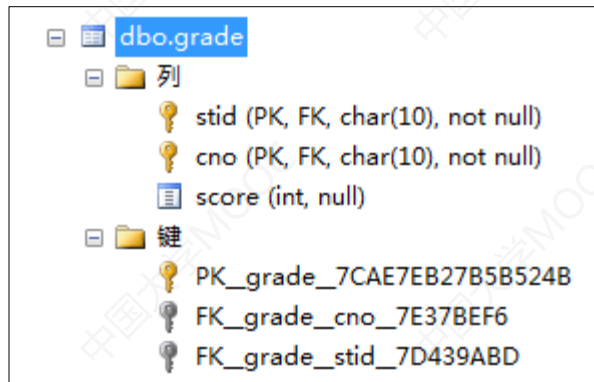
字段名	数据类型	约束	说明
stid	char(10)	主键	学号
cno	char(10)	主键	课程编号
score	int		成绩

04 参照完整性

例9 创建成绩表grade，要求grade表中所有的课程编号cno都出现在cinfo表中，学生学号stdid都出现在stinfo表中，假设cno和stdid分别是cinfo、stinfo的主键。

使用以下语句：

```
CREATE TABLE grade  
( std char(10),  
  cno char(10),  
  score int,  
  PRIMARY KEY(std,cno),  
  FOREIGN KEY(std) REFERENCES stinfo(std),  
  FOREIGN KEY(cno) REFERENCES cinfo(cno)  
)
```



05 本专题总结

- 设计表结构，就是确定字段、数据类型、存储长度、主键、外键等要素。
- 使用表设计器，或者使用CREATE TABLE、ALTER TABLE、DROP TABLE语句创建和维护表。
- 表数据的操作包括插入、删除、修改数据，使用界面方式，或者INSERT、UPDATE、DELETE、TRUNCATE语句进行操作。
- 数据完整性就是保证数据库中数据的有效性、一致性，SQL Server通过约束来实现数据完整性，约束包括主键约束、非空约束、默认约束、唯一约束、CHECK约束、外键约束。