

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

项目4 创建和管理数据库表

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【能力目标】

- 理解数据类型和表的基本概念
- 学会使用Management Studio创建表、修改表或删除表
- 学会使用Transact-SQL语句创建表
- 能显示表结构、修改表和删除表
- 插入表数据、删除表数据

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【项目描述】

在项目3中创建的XS数据库中创建三个表，学生档案表XSDA、课程信息表KCXX、学生成绩表XSCJ，并按照附录A录入表中数据。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【项目分析】

在学生数据库XS建立起来以后，还是无法实现具体数据的录入、查询等操作，原因是数据库中还没有建立用户自定义的数据表。只有建立了数据表，才能实现上述的操作。所以接下来就要按照项目2的设计，在学生数据库XS中建立学生档案表（XSDA）、课程信息表（KCXX）、学生成绩表（XSCJ）。该项目主要介绍如何在数据库中实现对数据表的各种操作。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【任务设置】

任务1		创建表
任务2		管理数据表
任务3		插入删除表中的数据
实训四		创建数据库表并录入表数据

项目4 创建和管理数据库表

任务1

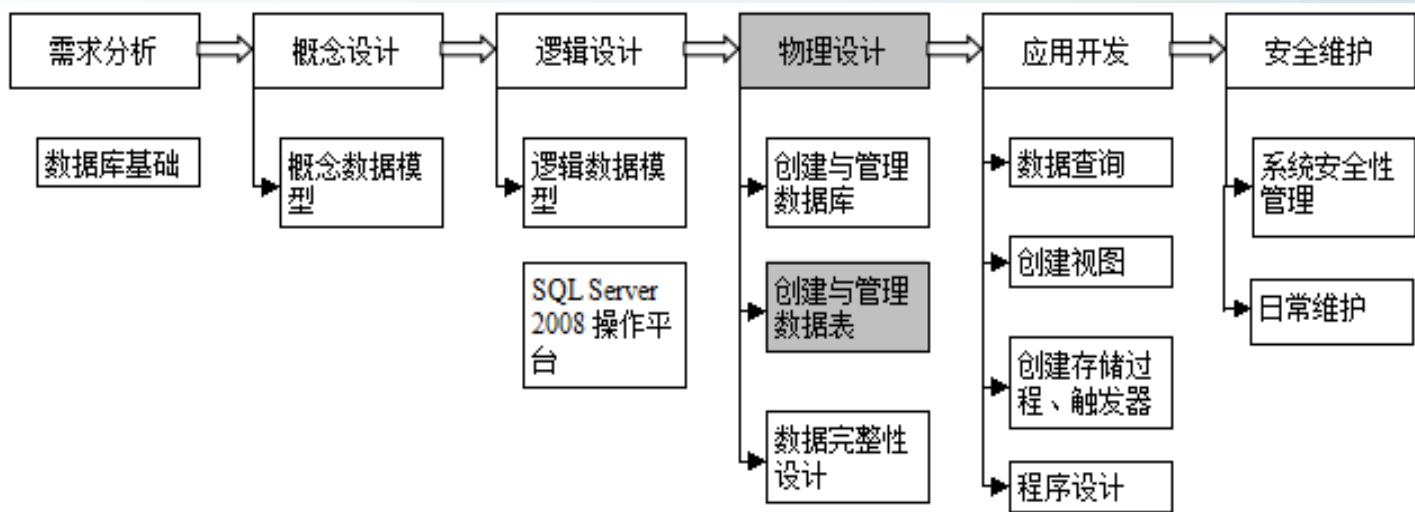
任务2

任务3

实训

【项目定位】

数据库系统开发



项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务1 创建表

【任务目标】

- 理解表的概念
- 灵活运用常用数据类型
- 学会使用对象资源管理器创建表
- 学会使用T-SQL语句创建表

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【任务描述】

根据提供的以下三个表的表结构，在数据库XS中分别使用SSMS和T-SQL语句建立这些表。

字段名	类型	长度	是否允许为空值	说明
学号	char	6	Not null	主键
姓名	char	8	Not null	
性别	char	2	Not null	男，女
系名	char	10	Not null	
出生日期	smalldatetime	4	Not null	
民族	char	4	Not null	
总学分	tinyint	1	Not null	
备注	Text	16		

表4-1 学生档案表（表名XSDA）结构

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

表4-2 课程信息表（表名KCXX）结构

字段名	类型	长度	是否允许为空值	说明
课程编号	char	3	Not null	主键
课程名称	char	20	Not null	
开课学期	tinyint	1	Not null	只能为1-6
学时	tinyint	1	Not null	
学分	tinyint	1	Not null	

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

表4-3 学生成绩表（表名XSCJ）结构

字段名	类型	长度	是否允许为空值	说明
学号	char	6	Not null	主键
课程编号	char	3	Not null	主键
成绩	tinyint	1		

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【任务分析】

该任务要求用两种方法创建三个表结构，实际上是创建了六个表。因为表不能重名，所以只能选择一种方法创建XSDA、KCXX、XSCJ，而另外一种方法创建这三个表必须用新表名，或者创建后删除表，再用新方法重新创建。因此，使用SSMS创建的三个表分别是XSDA_1、KCXX_1、XSCJ_1，使用T-SQL语句建立这些表分别是XSDA、KCXX、XSCJ，表结构分别如上图。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务1-1 数据表的概念

表是数据库的基本单位，它是一个二维表，表由行和列组成，如图4-4所示。每行代表唯一的一条记录，是组织数据的单位，通常称为表数据。每一行代表一名学生，各列分别表示学生的详细资料，如学号、姓名、性别、系名、出生日期、民族等。每列代表记录中的一个域，用来描述数据的属性，通常称为表结构，如姓名等，每个字段可以理解为字段变量，可以定义数据类型、大小等信息。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

表4-4 学生信息表

学号	姓名	性别	系名	出生日期	民族
200801	王红	女	信息	1986-02-14	汉
200802	刘林	男	信息	1986-05-20	汉
200803	曹红雷	男	信息	1985-09-24	汉
200804	方平	女	信息	1987-08-11	回
200805	李伟强	男	信息	1985-11-14	汉

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

SQL Server是一个关系数据库，它使用上述的由行和列组成的二维表来表示实体及其联系。SQL Server中的每个表都有一个名字，以标识该表。例如上表的名字是学生信息表。下面我们说明一些与表有关的名词：

(1) **表结构**。每个数据库包含了若干个表。每个表具有一定的结构，称之为“表型”。所谓表型是指组成表的名称及数据类型，也就是日常表格的“栏目信息”。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

(2) **记录**。每个表包含了若干行数据，它们是表的“值”，表中的一行称为一个记录，因此，表是记录的有限集合。

(3) **字段**。每个记录由若干个数据项构成，将构成记录的每个数据项称为字段。

(4) **关键字**。在学生信息表中，若不加以限制，每个记录的姓名、性别、系名、出生日期和民族这5个字段的值都有可能相同，但是学号字段的值对表中所有记录来说一定不同，学号是关键字，也就是说通过“学号”字段可以将表中的不同记录区分开来。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务1-2 数据类型

在设计数据库时，除了要确定它包括哪些表外，还要确定每个表中包含哪些列以及每列的数据类型等。**数据类型**就是定义每个列所能存放的数据值和存储格式。例如，如果某一列只能用于存放姓名，则可以定义该列的数据类型为字符型。同理，如果某列要存储数字，则可以定义该列的数据类型为数字型数据。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

列的数据类型可以是SQL Server 2008提供的**系统数据类型**，也可以是**用户自定义数据类型**。

1

系统数据类型

列的数据类型可以是SQL Server 2008提供的系统数据类型，也可以是用户自定义数据类型。这里介绍系统数据类型。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

表4-5 SQL Server 2008数据类型

数据类型	范围	存储
精确数字		
<u>bigint</u>	-2^{63} (-9,223,372,036,854,775,808) ~ $2^{63}-1$ (9,223,372,036,854,775,807)	8 字节
int	-2^{31} (-2,147,483,648) ~ $2^{31}-1$ (2,147,483,647)	4 字节
<u>smallint</u>	-2^{15} (-32,768) ~ $2^{15}-1$ (32,767)	2 字节
<u>tinyint</u>	0 ~ 255	1 字节
bit	1、0 或 NULL	不定
decimal	$-10^{38}+1 \sim 10^{38}-1$	5~17 字节
numeric		
money	-922,337,203,685,477.5808 ~ 922,337,203,685,477.5807	8 字节

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

数据类型	范围	存储
<u>smallmoney</u>	-214,748.3648~214,748.3647	4 字节
近似数字		
float	-1.79E + 308 ~ -2.23E - 308、0 以及 2.23E - 308 ~ 1.79E + 308	4~8 字节
real	-3.40E + 38 ~ -1.18E - 38、0 以及 1.18E - 38 ~ 3.40E + 38	4 字节
日期和时间		
<u>datetime</u>	1753 年 1 月 1 日到 9999 年 12 月 31 日 (精确到 3.33 毫秒)	8 字节
<u>smalldatetime</u>	1900 年 1 月 1 日到 2079 年 6 月 6 日 (精确到 1 分钟)	4 字节
字符串		
char	固定长度, 非 Unicode 字符数据, 长度为 n 个字节, n 的取值范围为 1~8,000	n 个字节
<u>varchar</u>	可变长度, 非 Unicode 字符数据, n 的取值范围为 1~8,000	输入数据的实际长度加 2 个字节
text	服务器代码页中长度可变的非 Unicode 数据, 最大长度为 $2^{31}-1$ (2,147,483,647) 个字符	$\leq 2,147,483,647$ 字节

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

Unicode 字符串		
<u>nchar</u>	固定长度的 Unicode 字符数据, n 值必须在 1 到 4,000 之间 (含)	2×n 个字节
<u>nvarchar</u>	可变长度 Unicode 字符数据, n 值在 1 到 4,000 之间 (含)	输入字符个数的两倍 + 2 个字节
<u>ntext</u>	长度可变的 Unicode 数据, 最大长度为 $2^{30} - 1$ (1,073,741,823) 个字符	输入字符个数的两倍
二进制字符串		
<u>binary</u>	长度为 n 字节的固定长度二进制数据, 其中 n 是从 1 到 8,000 的值	n 字节
<u>varbinary</u>	可变长度二进制数据, n 可以取从 1 到 8,000 的值	所输入数据的实际长度 + 2 个字节
<u>image</u>	长度可变的二进制数据, 从 0 到 $2^{31}-1$ (2,147,483,647) 个字节	不定

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

其他数据类型		
cursor	游标的引用	
sql_variant	用于存储 SQL Server 2008 支持的各种数据类型（不包括 text、ntext、image、timestamp 和 sql_variant）值的数据类型	
Table	一种特殊的数据类型，存储供以后处理的结果集	
Timestamp	数据库范围的唯一数字，每次更新行时也进行更新	
uniqueidentifier	全局唯一标识符（GUID）	
xml	存储 XML 数据的数据类型，可以在列中或者 xml 类型的变量中存储 xml 实例	

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

常用数据类型介绍：

bit类型。如果表中的列为 **8 bit 或更少**，则这些列作为 **1 个字节存储**。如果列为 9 到 16 bit，则这些列作为 2 个字节存储，以此类推。

char与varchar类型。如果列数据项的大小一致，则使用char。如果列数据项的大小差异相当大，则使用varchar。如果列数据项大小相差很大，而且大小可能超过8000字节，请使用varchar(max)。

binary与varbinary类型。如果列数据项的大小一致，则使用binary。如果列数据项的大小差异相当大，则使用varbinary。当列数据条目超出8000字节时，请使用varbinary(max)。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

二进制数据类型。二进制数据由十六进制数表示（例如，十进制数245等于十六进制数F5）。

image类型。image数据列可以用来存储超过8KB的可变长度的二进制数据，如Microsoft Word文档、Microsoft Excel电子表格、包含位图的图像、图形交换格式(GIF)文件和联合图像专家组(JPEG)文件。

text类型。text数据类型的列可用于存储大于8KB的ASCII字符。例如，由于HTML文档均由ASCII字符组成且一般长于8KB，所以这些文档可以以text数据类型存储在SQL Server中。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

nchar、nvarchar和ntext类型。此三种数据类型均为unicode数据类型。当一个列使用unicode类型时，该列可以存储多个字符集中的字符，此时数据以nchar、nvarchar和ntext数据类型存储。在unicode标准中，包括了以各种字符集定义的全部字符，如中文、日文字符集等。使用unicode数据类型，占用的存储空间是使用非unicode数据类型所占用存储空间的两倍。当一个列的长度可能变化时，应该使用nvarchar字符类型，这时最多可以存储4000个字符。当列的长度固定不变时，应该使用nchar字符类型，同样，这时最多可以存储4000个字符。当使用ntext数据类型时，该列可以存储多于4000个字符。

例如，字符型字符串“abcdABCD我们学习”作为unicode数据共12个字符占24字节。若存储为nchar(10)，则仅可保存“abcdABCD我们”；若存储为nchar(20)，则保存了“abcdABCD我们学习”占40字节；若存储为nvarchar(20)，则保存了“abcdABCD我们学习”占24字节。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

decimal 和 numeric类型。用于定义**精确数据**的数据类型。这种数据所占的存储空间根据该数据的小数点后面的位数来确定。

float和real类型。用于定义**近似小数数据**的数据类型。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

在SQL Server 2008中，对数据类型的设置与SQL Server 2005相比又有所改变。

(1) 用varchar(max)代替text。varchar(n)的最大长度为8000，但是varchar(max)则可以存储多达2GB的数据，其作用相当于SQL Server 2000中的text。

(2) 用nvarchar(max)代替text，用binary(max)代替image。

(3) 为XML数据选择xml类型。在SQL Server 2008中，为XML数据添加了相应的数据类型，显然存储XML数据的列不需要用varchar(max)或nvarchar(max)，而应当用xml数据类型，以利用T-SQL中专门针对xml数据列的新命令，以及针对xml列的索引。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

(4) 新增sql_variant数据类型。该类型能够保存除了text、ntext、image和timestamp以外任何SQL Server数据类型的数据。因此，通过使用sql_variant数据类型，可以在一个单独的字段、参数或变量中存储不同数据类型的数据值。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

2 用户自定义数据类型

用户定义的数据类型基于在Microsoft SQL Server中提供的数据类型。当在几个表中存储同一种数据类型的数据，并且为保证这些列有相同的数据类型、长度和为空性时，可以使用用户定义的数据类型。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

创建用户自定义数据类型时必须提供以下三个参数：数据类型名称、新数据类型所依据的系统数据类型、为空值（若为空值未定义，则系统将依据数据库或连接的ANSI NULL默认设置进行指派）。

在SQL Sever 2008中，创建用户自定义数据类型有两种方法：一是使用SQL Server Management Studio，二是使用T-SQL语句。下面分别介绍。

(1) 创建用户自定义的数据类型

(2) 删除用户自定义的数据类型

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务1-3 空值

空值（NULL）不等于零、空白或零长度的字符串。NULL值意味着没有输入，通常表明值是未知的或未定义的。例如学生成绩表（表名XSCJ）表中成绩列为空值时，并不表示该课程没有成绩或者成绩为0，而是指成绩未知或者尚未设定。

如果某一列向一个表中插入数据行时，没有给允许为NULL值得列提供值，SQL Server则自动将其赋值为NULL。

如果某一列不允许为空值，用户在想表中插入数据时必须为该列提供一个值，否则插入会失败。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

在设计表时，“允许空”的特性决定该列在表中是否允许为空值。

下面是空值的一些使用方法。

(1) 若要在SQL语句中测试某列是否为空值，可以在where字句中使用IS NULL或IS NOT NULL语句。

(2) 在查询窗口中查看查询结果时，空值在结果集内显示为（NULL）。

(3) 如果包含空值列，某些计算（如求平均值）可能得到不可预期的结果，所以在执行计算时要根据需要清除空值，或者根据需要对空值进行相应替换。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

(4) 如果数据中可能包含空值，那么在SQL语句中应尽量清除空值或将空值转换成其他值。

(5) **任何两个空值均不相等**。比如两个空值或将空值与任何其他数据相比较均返回未知。但如果数据库的ANSI_NULLS选项配置为关，则空值之间的比较（如NULL=NULL）等于TRUE。空值与任何其他数据类型之间的比较等于FALSE。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

建议：由于空值会导致查询和更新时使事情复杂，所以为了减少SQL语句的复杂性，**建议尽量不要允许使用空值**。如学生成绩表（表名XSCJ）表中成绩列可以设置它不允许空值，为其创建一个默认值（关于默认值，后面将详细介绍）为“-1”，这样对于没有确定的成绩就会取值“-1”，而不是空值。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务1-4 创建数据表

创建表的实质就是定义表结构及约束等属性，本任务主要介绍表结构的定义，而约束等属性将在后面专门介绍。在创建表之前，先要设计表，即确定表的名称、所包含的各列名、列的数据类型和长度、是否为空值、是否使用约束等。这些属性构成表结构。

在SQL Server 2008中我们可以使用SQL Server Management Studio和T-SQL语句两种方式创建表。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

1 使用SSMS创建数据表

下面以创建XS数据库中**学生档案表**（表名XSDA）为例介绍一下使用SQL Server Management Studio创建表的过程。

创建表前应该先确定**学生档案表**的名字和结构，表名为XSDA，表结构如表4-6所示。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

表4-6 XSDA表结构

字段名	类型	长度	是否允许为空值	说明
学号	char	6	not null	主键
姓名	char	8	not null	
性别	char	2	not null	
系名	char	10	not null	
出生日期	smalldatetime	4	not null	
民族	char	4	not null	
总学分	tinyint	1	not null	
备注	text	16		

项目4 创建和管理数据库表

任务1

任务2

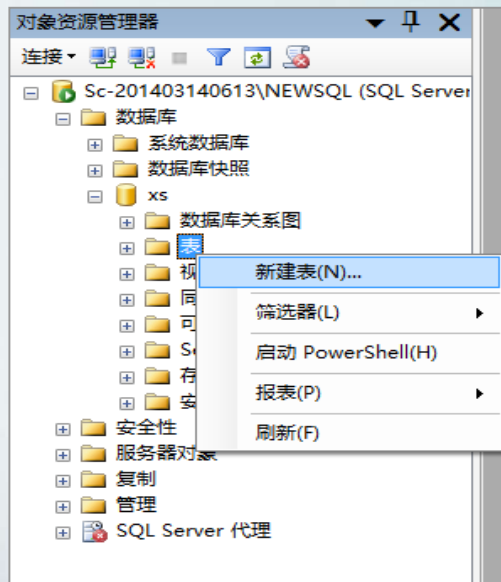
任务3

实训

然后就可以使用SQL Server Management Studio创建XSDA表了，操作步骤如下：

(1) 打开SQL Server Management Studio，在【对象资源管理器】中选择要创建表的XS数据库。

(2) 展开XS节点，右击【表】节点，在弹出的快捷菜单中选择【新建表】命令，如图4-3所示。



4-3在快捷菜单中选择“新建表”命令

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

(3) 在出现的表设计器窗口中定义表结构，即逐个定义表中的列(字段)，确定各字段的名称(列名)、数据类型、长度以及是否允许取空值等，如图4-4所示。

(4) 单击工具栏上的【保存】图标，保存新建的数据表。

(5) 在出现的【选择名称】对话框中，输入数据表的名称，例如XSDA，单击【确定】按钮，如图4-5所示。这时，可在右侧的【对象资源管理器】窗口中见到新建的XSDA数据表。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

Sc-201403140613\NEWSQL.xls - dbo.xsda			
	列名	数据类型	允许 Null 值
	学号	char(6)	☐
	姓名	char(8)	☐
	性别	char(2)	☐
	系名	char(10)	☐
	出生日期	smalldatetime	☐
	民族	char(4)	☐
	总学分	tinyint	☐
	备注	text	☒
			☐

图4-4 定义表中的列

选择名称 ? x

输入表名称(E):

XSDA

确定 取消

图4-5 “选择名称”对话框

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

2 用T-SQL语句创建数据表

使用T-SQL语句的CREATE TABLE可以创建表。

语法格式：

```
CREATE TABLE table_name
```

```
( {column_name data_type | IDENTITY (seed, increment) | NOT NULL | NULL} )
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

说明：

- `table_name`：新创建表的名称。表名必须符合标识符规则。
- `column_name`：是表中的列名。列名必须符合标识符规则，并且在表内唯一。
- `data_type`：指定列的数据类型。可以是系统数据类型或用户定义数据类型。
- `IDENTITY(seed, increment)`：指出该列为**标识列**。必须同时指定**种子和增量**，或者二者都不指定。如果二者都未指定，则取默认值（1, 1）。
- `NOT NULL|NULL`：指出该列中是否允许空值。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

CREATE TABLE语句的完整语法格式为：

CREATE TABLE

【 database_name. 【 owner 】 . | owner. 】 table_name

({ < column_definition >

 | column_name AS computed_column_expression

 | < table_constraint > ::= 【 CONSTRAINT constraint_name 】 }

 | 【 { PRIMARY KEY | UNIQUE } 【 ,...n 】

)

【 ON { filegroup | DEFAULT } 】

【 TEXTIMAGE_ON { filegroup | DEFAULT } 】

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

- **database_name**: 是要在其中创建表的数据库名称; **owner**: 表的所有者, 默认所有者为 **dbo**; **database_name** 必须是现有数据库的名称。如果不指定数据库, **database_name** 默认为当前数据库。数据库中的 **owner.table_name** 组合必须唯一。

- **column_definition**: 列定义, 其构成为:

< column_definition > ::= { column_name data_type }

【 COLLATE < collation_name > 】

【 【 DEFAULT constant_expression 】

| 【 IDENTITY 【 (seed , increment) 【 NOT FOR REPLICATION 】 】 】

】

【 ROWGUIDCOL 】

【 < column_constraint > 】 【 ...n 】

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

- `computed_column_expression`: 定义计算列值的表达式。**计算列**是物理上并不存储在表中的虚拟列。计算列由同一表中的其它列通过表达式计算得到。表达式可以是非计算列的列名、常量、函数、变量，也可以是用一个或多个运算符连接的上述元素的任意组合。表达式不能为子查询。
- `table_constraint`: 为表定义的各种约束，将在本书的后面具体讲述。
- `ON { filegroup | DEFAULT }`: 指定存储表的文件组。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

[例4-3] 创建名为jobs的表

```
USE xs
```

```
Go
```

```
CREATE TABLE jobs
```

```
(  
    工号 smallint identity(10000,1) primary key,  
    姓名 char(8) not null,  
    工种 char(12) null  
)
```

```
go
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

[例4-4] 创建名为students_T的表

```
Use xs
```

```
Go
```

```
CREATE TABLE students_T  
(  
    number int not null,  
    name varchar(10) not null,  
    sex char(2) null,  
    birthday datetime null,  
    hometown varchar(30) null,  
    telephone_no varchar(12) null,  
    address varchar(30) null,  
    others varchar(50) null  
)  
go
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务1-5 T-SQL 设置联合主键

主键是数据库表的一个重要属性，建立主键可以避免表中存在完全相同的记录，也就是说主键在一张表中的记录值是唯一的。

建立主键有两种方法：一种是在数据库提供的GUI环境中建立，另一种是通过SQL语句执行建立，下面分别介绍。

1. 在数据库对象资源管理器中建立。

输入表信息后按Ctrl键同时选中多行，然后点上面的主键按钮就行了。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

2. 通过SQL语句执行建立。又分两种，一是在建表语句中直接写，二是建表之后更改表结构。

在建表语句中直接写：

Create Table 表名

(字段名1 Int Not Null,

字段名2 nvarchar(13) Not Null Primary Key (字段名1, 字段名2),

字段名3.....

字段名N.....)

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

建表之后更改表结构：

```
CREATE TABLE 表名 ( 字段名1 Int Not Null,  
                    字段名2 nvarchar(13) Not Null,  
                    字段名3.....  
                    字段名N.....)
```

GO

```
ALTER TABLE 表名 WITH NOCHECK ADD  
CONSTRAINT 【PK_表名】 PRIMARY KEY  NONCLUSTERED  
(  
    【字段名1】 ,  
    【字段名2】  
)
```

GO

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务1-6 完成综合任务

1. 使用SSMS创建XSDA_1表

打开SQL Server Management Studio，在【对象资源管理器】中选择要创建表的XS数据库。展开XS节点，右击【表】->【新建表】->输入各字段的名称(列名)、数据类型、长度以及是否允许取空值->【保存】->【选择名称】->输入XSDA_1->【确定】。

可在左侧的【对象资源管理器】窗口中见到新建的XSDA_1数据表。

2. 使用SSMS创建KCXX_1、XSCJ_1表

方法与创建XSDA_1累述，不再累述。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

3. 使用T-SQL语句创建XSDA表，性别的默认值为男

```
USE XS
```

```
GO
```

```
CREATE TABLE XSDA (
```

```
    学号 char(6) NOT NULL PRIMARY KEY,
```

```
    姓名 char(8) NOT NULL ,
```

```
    性别 char(2) NOT NULL DEFAULT('男'),
```

```
    系名 char(10) NOT NULL ,
```

```
    出生日期 smalldatetime NOT NULL ,
```

```
    民族 char(4) NOT NULL ,
```

```
    总学分 tinyint NOT NULL ,
```

```
    备注 text NULL
```

```
)
```

```
GO
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

4. 使用T-SQL语句创建KCXX表

```
CREATE TABLE KCXX
```

```
(
```

```
    课程编号 char(3) NOT NULL PRIMARY KEY,
```

```
    课程名称 char(20) NOT NULL ,
```

```
    开学日期 tinyint NOT NULL ,
```

```
    学时 tinyint NOT NULL ,
```

```
    学分 tinyint NOT NULL
```

```
)
```

```
GO
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

5. 使用T-SQL语句创建XSCJ表

```
use xs
```

```
go
```

```
CREATE TABLE XSCJ (
```

```
    学号 char(6) NOT NULL ,
```

```
    课程编号 char(3) NOT NULL PRIMARY KEY(学号, 课程编号),
```

```
    成绩 tinyint NOT NULL
```

```
)
```

```
GO
```

运行结果如图4-6所示。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

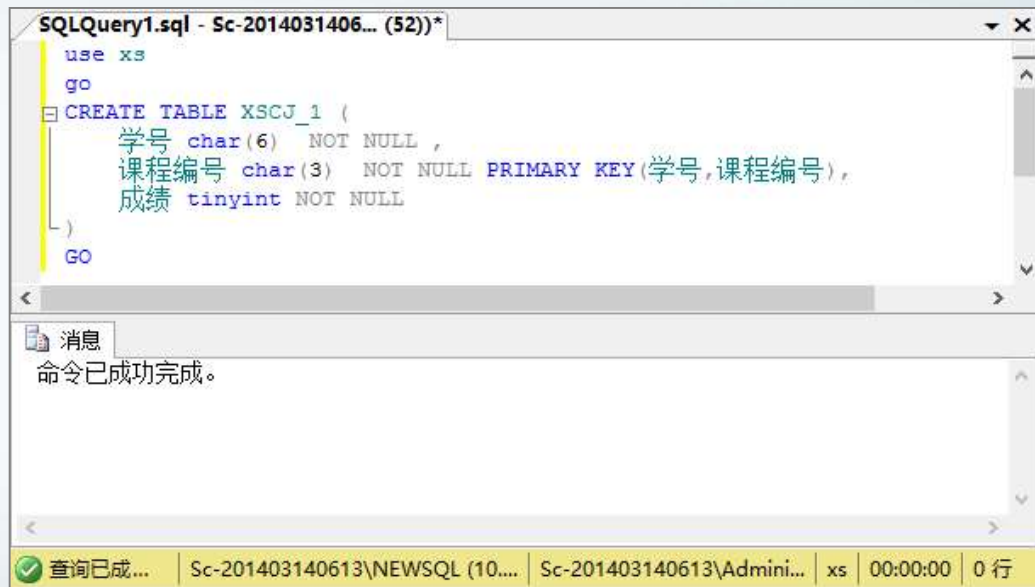


图4-6 设置联合主键

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务2 管理表

【任务目标】

- ◆ 学会在SSMS显示表结构
- ◆ 学会使用T-SQL显示表结构
- ◆ 灵活修改表结构
- ◆ 会重命名表
- ◆ 会删除没用的表

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【任务描述】

按照下面的要求修改、查看表结构。

1. 使用SSMS显示KCXX表结构
2. 用T-SQL课程信息表（表名KCXX）增加字段授课教师，数据类型为char(10)，考试时间，数据类型为datetime。
3. 用T-SQL课程信息表（表名KCXX）新增字段授课教师的字段名修改为teacher，数据类型为char(20)。
4. 用T-SQL课程信息表删除teacher字段。
5. 用T-SQL显示课程信息表（表名KCXX）的表结构。
6. 用SSMS将任务2-3中对XSDA表所做的修改恢复到求改前的状态。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【任务分析】

该任务中需要对表结构完成显示字段、增加字段、修改字段属性等操作。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务2-1 显示表结构

1 使用SSMS显示表结构

展开数据库，右击要打开表结构的表，
选择“**设计**”，如图4-7所示。

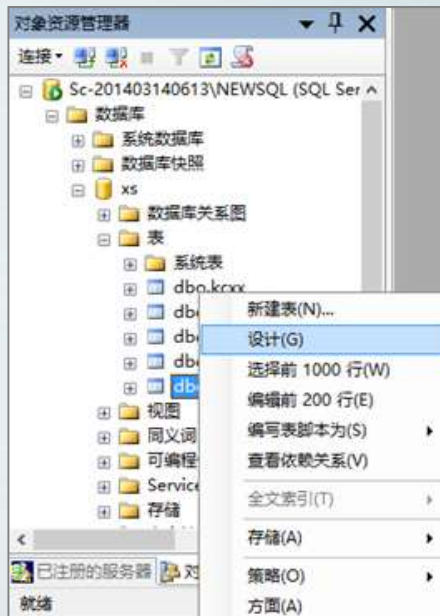


图4-7 打开表结构

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

2 使用T-SQL显示表结构

使用系统存储过程sp_help表名可以显示创建表的时间、表的所有者以及表中各列的定义等信息。

查看表结构：

```
Exec sp_help xsda
```

```
go
```

查看所有数据库对象

```
Exec sp_help
```

```
go
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务2-2 使用SSMS修改数据表

数据表创建以后，在使用过程中可能需要对原先定义的表的结构、约束等属性进行修改。表的修改与表的创建一样，可以通过SQL Server Management Studio和T-SQL语句两种方法来进行。

对一个已存在的表可以进行如下修改操作：

- (1) 更改表名
- (2) 增加列
- (3) 删除列
- (4) 修改已有列的属性（列名、数据类型、是否允许空值）

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

1 增加列

当原来创建的表中需要增加项目时，就要向表中增加列。

2 删除列

在SQL Server中被删除的列是不可恢复的，因此在删除列之前要慎重考虑。并且，在删除一个列以前，必须保证基于该列的所有索引和约束都已被删除。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

3 修改已有列的属性

在SQL Server中可以修改表结构，如更改列名、列的数据类型、长度和是否允许空等属性。建议当表中有记录后，不要轻易修改表的结构，特别是修改列的数据类型，以免产生错误。其操作过程是在SQL Server Management Studio中展开需进行操作的表，在XSDA表上单击右键，在弹出的快捷菜单上选择“设计”，如图4-7所示，然后选择要修改的列的列名，就可以修改列的属性了。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务2-3 使用T-SQL语句修改数据表

使用T-SQL语句的ALTER TABLE可以完成对表的修改。

语法格式：

```
ALTER TABLE table_name
```

```
{ 【 ALTER COLUMN column_name
```

```
{ new_data_type 【 ( precision 【 , scale 】 ) 】
```

```
【 NULL | NOT NULL 】
```

```
】 }
```

```
| ADD{ 【 < column_definition > 】 } 【 ,...n 】
```

```
| DROP{ 【 CONSTRAINT 】 constraint_name | COLUMN column_name } 【 ,...n 】
```

```
}
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

说明：

- ◆ ALTER COLUMN：用于说明修改表中指定列的属性，要修改的列名由column_name给出。
- ◆ new_data_type：指出要更改的列的新数据类型。
- ◆ precision：是指定数据类型的精度。scale：是指定数据类型的小数位数。
- ◆ ADD：指定要添加一个或多个列定义、计算列定义或者表约束。
- ◆ DROP：指定从表中删除约束或列。constraint_name：指定被删除的约束名；COLUMN column_name参数中指定的是被删除的列名。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

[例4-7] 在XSCJ表中增加一个新列——学分。

```
use xs
```

```
go
```

```
alter table xscj
```

```
add 学分 tinyint null
```

```
go
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

[例4-8] 在XSCJ表中删除名为“学分”的列。

```
use xs
```

```
go
```

```
alter table xscj
```

```
drop column 学分
```

```
go
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

[例4-9]将XSDA表中的“姓名”列的长度由原来的8改为10。

```
use xs
```

```
go
```

```
alter table xsda
```

```
    alter column 姓名 char(10)
```

```
go
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

[例4-10]在XSDA表名为“出生日期”列的数据类型由原来的smalldatetime改为datetime。

```
use xs
```

```
go
```

```
alter table xsda
```

```
    alter column 出生日期 datetime
```

```
go
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

[例4-11] 在XSDA表中名为“出生日期”的列名改为birthday。

```
use xs
```

```
go
```

```
sp_rename 'xsda.出生日期','birthday','column'
```

```
go
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务2-4 使用SSMS删除数据表

当数据库中的某些表失去作用时，可以删除表，以释放数据库空间，节省存储介质。删除表时，表的结构定义、表中所有的数据以及表的索引、触发器、约束等均被**永久**地从数据库中删除。**如果**要删除通过FOREIGN KEY和UNIQUE或 PRIMARY KEY约束相关的表，必须首先删除具有FOREIGN KEY约束的表。

我们可以使用SQL Server Management Studio和T-SQL语句两种方式删除表。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务2-5 使用T-SQL语句删除数据表

使用T-SQL语句中的DROP TABLE可以删除表。

语法格式：

```
DROP TABLE table_name
```

说明：

table_name为要删除的表。

【例4-13】删除数据库XS中的KCXX表。

```
USE XS
```

```
DROP TABLE KCXX
```

```
GO
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务2-6 使用T-SQL语句重命名数据表

【例4-14】把XSDA表更名为学生档案。

在查询窗口中执行如下SQL语句：

```
USE XS
```

```
EXEC sp_rename 'XSDA','学生档案'
```

```
GO
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务2-7 完成综合任务

1. 使用SSMS显示KCXX表结构

展开数据库->右击KCXX->设计

2. 用T-SQL课程信息表（表名KCXX）增加字段授课教师，数据类型为char(10)，考试时间，数据类型为datetime。

```
ALTER TABLE KCXX
```

```
ADD 授课教师 char(10) NULL, 考试时间 datetime NULL
```

```
GO
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

3. 用T-SQL课程信息表（表名KCXX）新增字段授课教师的字段名修改为teacher，数据类型为char(20)。

```
sp_rename 'KCXX.授课教师','teacher','COLUMN'
```

```
GO
```

```
ALTER TABLE KCXX
```

```
    ALTER COLUMN teacher char(20)
```

```
GO
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

4. 用T-SQL课程信息表删除teacher字段。

```
ALTER TABLE XSCJ
```

```
DROP COLUMN teacher
```

```
GO
```

5. 用T-SQL显示课程信息表（表名KCXX）的表结构。

```
sp_help KCXX
```

```
go
```

6. 用SSMS将任务2-3中对XSDA表所做的修改恢复到修改前的状态。

展开数据库->右击KCXX->设计->相应修改

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务3 插入删除表数据

【任务目标】

- ◆ 学会使用对象资源管理器SSMS插入、删除修改表数据
- ◆ 学会使用T-SQL语句插入表数据
- ◆ 学会使用T-SQL语句删除表数据

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【任务描述】

为XS数据库三个表录入表数据。

1. 参照附录A，使用SSMS录入学生档案表（表名XSDA）数据样本前五五行数据
2. 参照附录A，使用T-SQL录入学生档案表（表名XSDA）数据样本剩余数据
3. 参照附录A，使用T-SQL录入课程信息表（表名KCXX）数据样本数据
4. 参照附录A，使用T-SQL录入学生成绩表（表名XSCJ）数据样本前五五行数据
5. 删除学生成绩表（表名XSCJ）中学号为200501的数据
6. 删除课程信息表（表名KCXX）中全部数据

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【任务分析】

对于初学者来说使用T-SQL插入数据，很容易出现各种错误，所以一定要按照要求——正确插入样表数据，在排错过程中将更深刻的理解表数据和表结构。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务3-1 使用SSMS插入、删除、更新表数据

创建表后，下一步要做的就是表中进行数据操作，对表中的数据操作包括表记录的插入、修改和删除。可以通过SQL Server Management Studio和T-SQL语句两种方式操作表数据。

向表中插入数据就是将**新记录添加到表尾**，可以向表中插入多条记录。

1. 使用SQL Server Management Studio向表中插入数据

操作方法如下：

(1) 展开数据库XS，展开表，在表XSDA上单击右键，弹出快捷菜单选择“编辑前200行”，如图4-12所示。

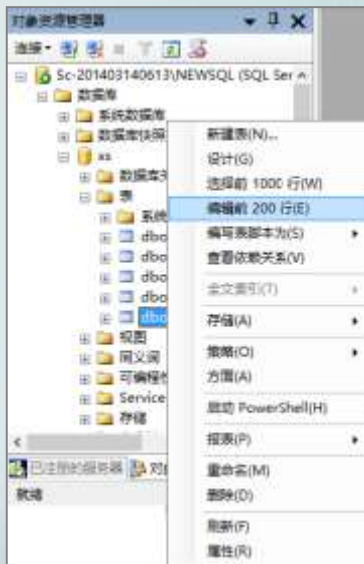


图4-12 打开表数据

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

(2) 进入打开的表XSDA数据窗口，输入数据，每输完一列的值，按Tab键，光标会自动跳到下一列，若输完最后一列数据，按回车键，则光标跳至下一行的第一列，直到输入完成，按关闭窗口按钮。如图4-13所示。



	学号	姓名	性别	系名	出生日期	民族
▶	200501	王红	女	信息	1986-02-14 00:...	汉
	200502	刘林	男	信息	1986-05-20 00:...	汉
	200503	曹红雷	男	信息	1985-09-24 00:...	汉
	200504	方平	女	信息	1987-08-11 00:...	回
	200505	李伟强	男	信息	1985-11-14 00:...	汉
	200506	周新民	男	信息	1986-01-20 00:...	回
	200507	王丽丽	女	信息	1987-06-03 00:...	汉
	200601	孙燕	女	管理	1987-05-20 00:...	汉
	200602	罗德敏	男	管理	1988-07-18 00:...	汉
	200603	孔祥林	男	管理	1986-05-20 00:...	汉
	200604	王华	女	管理	1987-04-16 00:...	汉
	200605	刘林	男	管理	1986-05-20 00:...	回
	200606	陈希	女	管理	1987-03-22 00:...	汉
*	NULL	NULL	NULL	NULL	NULL	NULL

图4-13 向XSDA表输入数据

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

2. 使用SQL Server Management Studio修改表中数据

使用SQL Server Management Studio修改XSDA表中数据的操作步骤：

- (1) 展开数据库XS，展开表，在表XSDA上单击右键，弹出快捷菜单选择“编辑前200行”，如图4-10所示。
- (2) 在弹出的查询窗口中，单击要修改的单元格后可以修改此处的表中数据，修改完成按关闭窗口按钮即可。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

3. 使用SQL Server Management Studio删除表中数据

展开表，右击表，在弹出的快捷菜单选择“编辑前200行”，在操作表窗口中定位要删除的数据行（可辅助Ctrl键或Shift键选中多行），单击鼠标右键，在弹出的快捷菜单中选择【删除】，如图4-14所示。在确认窗口单击【是】按钮，即可删除所选择的数据行。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训



学号	姓名	性别	系名	出生日期	民族
200501	王红	女	信息	1986-02-14 00:...	汉
200502	刘林	男	信息	1986-05-20 00:...	汉
200503	曹红雷	男	信息	1985-09-24 00:...	汉
200504	王平	女	信息	1987-08-11 00:...	回
200505	李伟强	男	信息	1985-11-14 00:...	汉
200506	周新民	男	信息	1986-01-20 00:...	回
200507	王丽丽	女	信息	1987-06-03 00:...	汉
200508	小燕	女	管理	1987-05-20 00:...	汉
200509	罗德敏	男	管理	1988-07-18 00:...	汉
200510	孔祥林	男	管理	1986-05-20 00:...	汉
200511	王华	女	管理	1987-04-16 00:...	汉
200605	刘林	男	管理	1986-05-20 00:...	回
200606	陈希	女	管理	1987-03-22 00:...	汉
NULL	NULL	NULL	NULL	NULL	NULL

图4-14选择、删除表中数据

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务3-2 使用T-SQL语句向表中插入数据

使用T-SQL语句的INSERT可以向表中插入数据，INSERT语句常用的格式有三种。语法格式一：

```
INSERT table_name
```

```
VALUES (constant1, constant2, ...)
```

该语句的功能是向由table_name指定的表中插入由VALUES指定的各列值的行。

注意：使用此方式向表中插入数据时，VALUES中给出的数据顺序和数据类型必须与表中列的顺序和数据类型一致，而且不可以省略部分列。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【例4-15】向XS数据库的表XSDA中插入如下一行。

200608 李忠诚 男 信息 1988-09-10 汉 60 null

可以使用如下的T-SQL语句：

```
USE XS
```

```
INSERT XSDA
```

```
VALUES ('200608', '李忠诚', '男', '信息', '1988-09-10', '汉', 60, null)
```

```
GO
```

马上查询一下，你会发现其中多了学号为“200608”这一行。

读者再尝试一下选择一些有默认值或者可以为空的列，在插入数据时省略这些列，也就是说采用默认值或者给允许为空的列赋空值。看下面的例子。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【例4-16】查看一下XS数据库的表XSDA的表结构，我们可知道性别可以使用默认值“男”，民族列可以使用默认值“汉”，备注列可以为空。如果我们将【例4-16】中的T-SQL语句改成下面这样：

```
USE XS
```

```
INSERT XSDA
```

```
VALUES ('200608', '李忠诚', '信息', '1988-09-10', 60)
```

```
GO
```

那么将无法实现预期的效果，并且在结果显示窗格中将会显示出错提示信息。如图4-15所示。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训



图4-15 结果窗格中错误提示

若想在INSERT语句中只给出部分列值，就需要用到INSERT语句的另一种格式了，将在后面的项目中讲解。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务3-3 使用T-SQL语句删除表记录

当表中某些数据不再需要时，要将其删除。有时录入数据出现错误，使用SSMS无法删除数据，也可以使用T-SQL删除表中记录。

语法格式：

DELETE 【FROM】

{table_name|view_name}

【WHERE <search_condition>】

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

说明：

- ◆ `table_name|view_name`：是要从其中删除行的表或视图的名称。其中，通过 `view_name` 来引用的视图必须可更新且正确引用一个基表。
- ◆ `WHERE <search_condition>`：指定用于限制删除行数的条件。如果没有提供 `WHERE` 子句，则 `DELETE` 删除表中的所有行。

【例4-17】将XSDA表中总学分小于54的行删除。

```
USE XS
DELETE FROM XSDA
WHERE 总学分<54
GO
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

【例4-18】将XSDA表中备注为空的行删除。

```
USE XS
```

```
DELETE FROM XSDA
```

```
WHERE 备注 IS NULL
```

```
GO
```

【例4-19】删除XSDA表中所有行。

```
USE XS
```

```
DELETE XSDA
```

```
GO
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

任务3-4 完成综合任务

1. 参照附录A，使用SSMS录入学生档案表（表名XSDA）数据样本前五五行数据
展开数据库XS，展开表XSDA上单击右键->编辑前200行->输入数据->Tab键->直到输入完成->关闭窗口。

2. 参照附录A，使用T-SQL录入学生档案表（表名XSDA）数据样本剩余数据

```
INSERT XSDA
```

```
VALUES('200506','周新民','男','信息','1986-01-20','回族',62,null)
```

```
INSERT XSDA
```

```
VALUES('200507','王丽丽','女','信息','1987-06-03','汉族',60,null)
```

```
GO
```

其它数据从略。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

3. 参照附录A，使用T-SQL录入课程信息表（表名KCXX）数据样本数据

```
INSERT KCXX
```

```
VALUES ('104', '计算机文化基础', 1, 60, 3, '2014-01-20')
```

```
INSERT XSDA
```

```
VALUES ('108', 'C语言程序设计', 1, 96, 5, '1987-06-03')
```

```
GO
```

其它数据从略。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

4. 参照附录A，使用T-SQL录入学生成绩表（表名XSCJ）数据样本前五行数据

```
INSERT XSCJ
```

```
VALUES ('200501', '104', 81)
```

```
INSERT XSDA
```

```
VALUES ('200501', '108', 77)
```

```
GO
```

其它数据从略。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

5. 删除学生成绩表（表名XSCJ）中学号为200501的数据

```
USE XS
```

```
DELETE FROM XSDA
```

```
WHERE 学号='200501'
```

```
GO
```

6. 删除课程信息表（表名KCXX）中全部数据

```
DELETE KCXX
```

```
GO
```

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

实训四 创建数据库表并录入表数据

本书实训项目都是围绕sale数据展开的，进销存系统通常包括客户资料、产品信息、进货记录、销售记录等。所以针对sale数据库，设计了表4-7~表4-10，并将在后续项目逐步完善。在sale数据库下创建一下各表输入数据。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

表4-7 Customer（客户表）

<u>CusNo(客户编号)</u> nvarchar(3) not null	<u>CusName(客户姓名)</u> nvarchar(10) not null	<u>Address(地址)</u> nvarchar(20)	<u>Tel(联系电话)</u> nvarchar(3)
001	杨婷	深圳	0755-22221111
002	陈萍	深圳	0755-22223333
003	李东	深圳	0755-22225555
004	叶合	广州	020-22227777
005	谭欣	广州	020-22229999

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

表4-8 Product（产品表）

<u>ProNo(产品编号)</u> nvarchar(5) not null	<u>ProName(产品名)</u> nvarchar(20) not null	Price(单价) Decimal(8,2) not null	<u>Stocoks(库存数量)</u> Decimal(8,0) not null
00001	电视	3000.00	800
00002	空调	2000.00	500
00003	床	1000.00	300
00004	餐桌	1500.00	200
00005	音响	5000.00	600
00006	沙发	6000.00	100

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

表4-9 ProIn（入库表）

<u>InputDate(入库日期)</u> <u>DateTime</u> not null	<u>ProNo(产品编号)</u> <u>nvarchar(5)</u> not null	<u>Quantity(入库数量价)</u> <u>Decimal(6,0)</u> not null
2006-1-1	00001	10
2006-1-1	00002	5
2006-1-2	00001	5
2006-1-2	00003	10
2006-1-3	00001	10
2006-2-1	00003	20
2006-2-2	00001	10
2006-2-3	00004	30
2006-3-3	00003	20

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

表4-10 ProOut（销售表）

<u>SaleDate(销售日期)</u> <u>DateTime</u> not null	<u>CusNo(客户编号)</u> <u>nvarchar(3)</u> not null	<u>ProNo(产品编号)</u> <u>nvarchar(5)</u> not null	<u>Quantity(入库数量价)</u> <u>Decimal(6,0)</u> not null
2006-1-1	001	00001	10
2006-1-2	001	00002	5
2006-1-3	002	00001	5
2006-2-1	002	00003	10
2006-2-2	001	00001	10
2006-2-3	001	00003	20
2006-3-2	003	00001	10
2006-3-2	003	00004	30
2006-3-3	002	00003	20

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

小结

本项目首先介绍了表的概念；介绍了SQL Server 2008系统数据类型；着重介绍了SQL Server Management Studio和T-SQL语句创建、修改和删除表数据的操作方法及语句格式；需要掌握的主要内容如下：

1. 数据表的概念：表是包含数据库中所有数据的数据库对象。与表有关的名词：表结构、记录、字段和关键字。
2. 创建表时指定列的数据类型：可以是SQL Server 2008提供的系统数据类型，也可以是用户自定义数据类型。

项目4 创建和管理数据库表

任务1

任务2

任务3

实训

3. 创建表就是定义表的结构，即确定表的名字、所包含的各列名、列的数据类型和长度、是否为空值等并使用SQL Server Management Studio或T-SQL语句实现之。数据表创建以后，在使用过程中可能需要对原先定义的表的结构属性进行修改。当数据库中的某些表失去作用时，可以删除表，以释放数据库空间，节省存储介质。创建表后，可以对表中数据进行操作，对表中的数据操作包括表记录的插入、修改和删除。