第 5 章　SQL Server 2008 数据表和索引

在 SQL Server 2008 中，表是用来存储数据的对象。通常将同一类或者是相关数据存放在一个数据表中。例如，所有的学生信息存放在一个表中，而成绩信息都存放到另一个表中。用户可以通过多种方式访问数据库中的数据。掌握 SQL Server 2008 数据表的相关知识是非常重要的。本章主要包括数据表的基础知识、数据表定义的管理、表中数据的管理及索引的管理等知识。

5.1　数据表简介

为了更好地理解和掌握 SQL Server 2008 中数据表的设计与使用，本节首先简要介绍数据表及其相关的基础知识。

5.1.1　什么是数据表

数据表是 SQL Server 2008 的数据库对象，它存储着数据库的所有数据。在数据表中，数据以行和列的形式存储在规范化的二维表格中。SQL Server 2008 数据表主要由行和列构成。

- 列：用来保存对象的某一类属性。每列又称为一个字段，每列的标题称为字段名。
- 行：用来保存一条记录，是数据对象的一个实例，包括若干列信息项。

如图 5-1 显示的是教务管理系统数据库（Practice_JWGL）中的学生信息表的部分截图。该表包含行和列的信息，其中，行表示数据，即学生实体；列表示数据域，即学生实体的属性。

YANG-SQL-PC\M...GL - dbo.学生信息表

学号	姓名	性别	籍贯	班级编号
0511101020	李建平	男	四川	051110
0511101046	罗川伟	男	四川	051110
0516102073	李佩秋	女	云南	051610
0611101007	袁娜	女	陕西	061110
0612102055	杨多玲	女	重庆	061210
0612102061	李江涛	男	甘肃	061210
0702102020	谢涛	男	重庆	070210
0702102052	罗伟	男	甘肃	070210

行　列

图 5-1　学生信息表

5.1.2　数据完整性

定义表除了要定义每一列的名称、数据类型和长度之外，还可以为列定义一些其他属

性。这些属性也很重要，它们可以保证数据表数据的完整性和参照完整性。在 SQL Server 2008 中，用于保证以上两种完整性的相关技术有如下 3 种。

1．约束

约束是 SQL Server 2008 数据库中一种自动保持数据完整性的机制。约束可以定义在列上，也可以定义在表上。列约束只能对某一列起作用，表约束可以对表的多个列起作用。当表的多个列同时需要约束时，最好使用表约束。

SQL Server 2008 中使用的约束类主要有 PRIMARY KEY、FOREIGN KEY、UNIQUE、CHECK 和 NOT NULL。这些约束将在第 6 章中详细介绍。

2．规则（Rules）

规则是为了保持向后兼容性而保留下来的技术，它与一些 CHECK 约束的功能类似。CHECK 约束是首选的保证数据完整性的技术。CHECK 约束比规则更简明，并且一列可以使用多个约束，但是一列只能使用一个规则。另外，CHECK 约束可以在创建表的时候指定，而规则是一种分离定义的对象，然后限制到列上。

注意：在 SQL Server 未来的版本中，规则可能不再使用，所以用户应尽量避免使用规则。

3．默认值

默认值是指当用户在表中插入一行数据时，如果没有为某个（或某些）列指定值，这个（或这些）列就使用这个默认定义的值。默认值可以是计算结果为常量的任何值，例如常量、内置函数或数学表达式。

说明：若要应用默认值，可以通过在 CREATE TABLE 中使用 DEFAULT 关键字来创建默认值定义，这将为每一列分配一个常量表达式作为默认值。

5.1.3　SQL Server 2008 特殊表

SQL Server 2008 除了提供用户定义的标准数据表外，还提供了一些特殊用途的表：分区表、临时表和系统表。

1．分区表

当数据表很大时，可以把数据分割成很多单元，放在同一个数据库的规格文件组中。用户可以通过分区快速地访问和管理数据的某些部分子集而不是整个数据表，从而便于管理大表和索引。

2．临时表

SQL Server 2008 有两种临时表：局部临时表和全局临时表。局部临时表只是对一个数据库实例的一次连接中的创建者是可见的。当用户断开数据库的连接时，这些局部临时表就会被删除。全局临时表创建后对所有的用户和连接都是可见的，并且只有所有的用户都

断开与临时表相关的数据表时，全局临时表才会被删除。

3．系统表

系统表用来保存一些服务器配置信息数据，用户不能直接查看和修改这些系统表，只有通过专门的管理员连接才能查看和修改。不同版本的数据库系统的系统表一般不同。在升级数据库系统时，一些应用系统表的应用可能需要重写。在 SQL Server 2008 中，所有的系统表与基表都具有相同的逻辑结构，因此，用于检索和修改基表信息的 Transact-SQL 语句，同样可以用于检索和修改系统表中的信息。

有关系统表的详细信息，读者可以自行查看 SQL Server 2008 的帮助。下面简要介绍几个最重要的系统表。

- Sysobjects 表：SQL Server 的主系统表，出现在每个数据库中。在数据库中创建的每个对象（例如约束、默认值、日志、规则以及存储过程）都对应一行。
- Syscolumns 表：出现在 master 数据库和每个用户自定义的数据库中，对基表或者视图的每个列和存储过程中的每个参数都含有一行记录。
- Sysindexes 表：出现在 master 数据库和每个用户自定义的数据库中，它对每个索引和没有聚簇索引的每个表都含有一行记录，它还对包括文本/图像数据的每个表含有一行记录。
- Sysusers 表：出现在 master 数据库和每个用户自定义的数据库中，它对整个数据库中的每个 Windows NT 用户、WindowsNT 用户组、SQL Server 用户或者 SQL Server 角色含有一行记录。
- Sysdatabases 表：它对 SQL Server 系统上的每个系统数据库和用户自定义的数据库含有一行记录，只出现在 Master 数据库中。
- Sysdepends 表：它对表、视图和存储过程之间的每个依赖关系都含有一行记录本，出现在 Master 数据库和每个用户自定义的数据库中。
- Sysconstraints 表：对使用 CREATE TABLE 或者 ALTER TABLE 语句为数据库对象定义的每个完整性约束都含有一行记录，包含约束映射，映射到数据库中拥有该约束的对象。

注意：任何用户都不应直接更改系统表。如果使用 DDL 语句的 INSERT、UPDATE 和 DELETE 语句来修改系统表的信息，对整个系统是非常危险的，所以应该使用系统存储过程来代替。

5.1.4　规划数据表

在创建数据表之前，用户首先要做好表的设计工作，然后再创建表，最后输入数据。数据表的设计是非常重要的，若数据表设计得不合理，会十分浪费资源，并且有可能带来不可估量的损失。在设计数据表时，必须确定表的使用目的、数据的类型以及可以访问每个表的用户。在创建表及其对象之前，最好先确定表的下列特征：

- 表要包含的数据的类型。
- 表中的列数，每一列中数据的类型和长度（如果必要）。

- 哪些列允许空值。
- 是否需要使用以及何处使用约束、规则和默认值。
- 所需索引的类型，哪里需要索引，哪些列是主键，哪些是外键。

1．列的数据类型的选择

为每一列分配数据类型，是设计表的第一步。用户可以为列分配的数据类型有 SQL Server 2008 系统数据类型、系统数据类型的别名数据类型和用户自定义数据类型。

2．自动生成列的编号或标识符

每个表都可以通过系统生成的序列产生一个标识列。在同一个表中这些标识是不会重复的，但是对于不同的表，则是可以相同的。在 SQL Server 2008 中，是通过在表设计器中为该列设置“标识规范”属性来实现的。

3．需要数据计算的列

表中某列的值可能不会直接得到，而是需要通过某些列的计算得到。SQL Server 2008 支持直接的函数、数学表达式计算，但是不支持需要查询的计算。在没有说明的情况下，这些需要计算的列是虚列，物理表中不存在，每次查询时都需要重新计算。在 SQL Server 2008 中，可以设置参数，把这些需要计算的列保存在物理表中。

4．加强数据完整性

在数据表的设计中，还经常需要考虑使用什么样的技术来保证数据的完整性。用户要充分使用 SQL Server 2008 支持的约束、规则和默认值来保证数据的完整性。

以上介绍了设计数据表时需要考虑的一些因素。当数据表设计完成之后，可以采用多种方式创建数据表，如在 SQL Server Management Studio 中使用图形界面或者执行 Transact-SQL 语句来创建数据表。

5.1.5　系统数据类型

系统数据类型是 SQL Server 预先定义好的，可以直接使用。在实际使用中，SQL Server 会自动限制每个系统数据类型的值的范围。当插入数据库中的值超过了数据类型允许的范围，SQL Server 系统就会报错。SQL Server 2008 提供了 7 类，共 26 种系统数据类型，如图 5-2 所示。

1．精确数字

所谓精确数字，是指不带小数位的整数。

（1）bit 称为位数据类型，其数据有两种取值，即 0 和 1。SQL Server 在存储 bit 数据类型时做了优化。如果一个表中有 8 个或更少的 bit 列时，用 1 个字节存放。如果有 9～16 个 bit 列时，用 2 个字节存放。

在输入 0 以外的其他值时，系统均把它们当 1 看待。这种数据类型常作为逻辑变量使用，用来表示真、假或是、否等二值选择。

图 5-2　SQL Server 2008 系统数据类型

（2）tinyint：每个 tinyint 类型的数据占用 1 个字节的存储空间，它可以存储从 0～255 范围的所有正整数。

（3）smallint：每个 smallint 类型的数据占用 2 个字节的存储空间，其中，一个二进制位表示整数值的正负号，其他 15 个二进制位表示整数值的长度和大小。可以存储在-2^{15}（–32 768）～（2^{15}–1）（32 767）范围的所有正负整数。

（4）int（integer）：每个 int（或 integer）数据类型值存储在 4 个字节中，其中，一个二进制位表示整数值的正负号，其他 31 个二进制位表示整数值的长度和大小。int 数据类型可以存储在-2^{31}（–2 147 483 648）～（2^{31}–1）（2 147 483 647）范围的所有正负整数。

（5）bigint：用于存储从-2^{63}（–9 223 372 036 854 775 807）～（2^{63}–1）（–9 223 372 036 854 775 807）之间的所有正负整数，每个 bigint 类型的数据占用 8 个字节的存储空间。

（6）decimal 和 numeric：decimal 数据类型和 numeric 数据类型完全相同，分为两种是为了保持与 ANSI 标准兼容。

这两种数据类型可以提供小数所需的实际存储空间，可以用 2～17 个字节来存储（-10^{38}–1）～（10^{38}–1）之间的数值。也可以将其写为 decimal（p,s）的形式，其中，p 表示可供存储的数值的总位数，默认设置为 18。s 表示小数点后的位数，默认设置为 0。

注意：数值类型的总位数不包括小数，例如 decimal（10,5），表示共有 10 位数，其中整数 5 位，小数 5 位。

（7）money：用于存储货币值，存储在 money 数据类型中的数值以一个正数部分和一个小数部分存储在两个 4 字节的整型值中。存储范围为–922 337 213 685 477.5808～922 337 213 685 477.580 8，精度为货币单位的万分之一。

（8）smallmoney：与 money 数据类型类似，但范围比 money 数据类型小，其存储范围为–214 748.346 8～214 748.346 8。

当为 money 或 smallmoney 的表输入数据时，必须在有效位置前面加一个货币单位符号（如$或其他货币单位的记号）。

2．近似数字

近似数据类型用于存储十进制小数。近似数值的数据在 SQL Server 中采用只入不舍的方式进行存储。即当（且仅当）要舍入的数是一个非零数时，对其保留数字部分的最低有效位上的数值加 1，并进行必要的进位。

（1）float：可以精确到第 15 位小数，其范围为（–1.79E–308）～（1.79E+308）。如果不指定 float 数据类型的长度，它占用 8 个字节的存储空间。float 数据类型也可以写为 float（n）的形式，n 指定 float 数据的精度，n 为 1～15 之间的整数值。当 n 取 1～7 时，实际是定义了一个 real 类型的数据，系统用 4 个字节存储它；当 n 取 8～15 时，系统认为其是 float 类型，用 8 个字节存储它。

（2）real：每个 real 类型的数据占用 4 个字节的存储空间，可以存储正的或者负的十进制数值，最大可以有 7 位精确位数，它的存储范围在（–3.40E–38）～（3.40E+38）。

3．时间和日期

SQL Server 2008 支持的时间和日期数据类型主要有 6 种，分别是 datetime、smalldatetime、date、time、datetimeoffset、datetime2。下面主要讲解 datetime 和 smalldatetime 两种。

（1）datetime：用于存储日期和时间的结合体。它可以存储在公元 1753 年 1 月 1 日零时～公元 9999 年 12 月 31 日 23 时 59 分 59 秒之间的所有日期和时间，其精确度可达三百分之一秒，即 3.33 毫秒。

datetime 数据类型所占用的存储空间为 8 个字节，其中，前 4 个字节用于存储基于 1900 年 1 月 1 日之前或者之后日期数。数值分正负，负数存储的数值代表在基数日期之前的日期，正数表示基数日期之后的日期。时间以子夜后的毫秒存储在后面的 4 个字节中。

当存储 datetime 数据类型时，默认的格式是 MM DD YYYY hh:mm A.M./P.M。当插入数据或者在其他地方使用 datetime 类型时，需要用单引号把它括起来。默认的时间日期是 January1,1900 12:00 A.M。可以接受的输入格式有 Jan 4 1999、JAN 4 1999、January 4 1999、Jan 1999 4、1999 4 Jan 和 1999 Jan 4。Datetime 数据类型允许使用/、-和.作为不同时间单位间的分隔符。

（2）smalldatetime：与 datetime 数据类型类似，但其日期时间范围较小，它存储在 1900 年 1 月 1 日～2079 年 6 月 6 日之间的日期。Smalldatetime 数据类型使用 4 个字节存储数据，其中，SQL Server 2008 用两个字节存储日期 1900 年 1 月 1 日以后的天数，时间以子夜后的分钟数形式存储在另外两个字节中。smalldatetime 的精确度为 1 分钟。

4．字符串

字符数据类型是 SQL Server 中最常用的数据类型之一，它可以用来存储各种字母、数字符号和特殊符号。在使用字符数据类型时，需要在其前后加上英文单引号或者双引号。

（1）char：当用 char 数据类型存储数据时，每个字符和符号占用 1 个字节的存储空间。其定义形式为：

```
char(n)
```

其中，n 表示所有字符所占的存储空间，n 的取值为 1～8000。若不指定 n 值，系统默认 n 的值为 1。若输入数据的字符串长度小于 n，则系统自动在其后添加空格来填满设定好的空间。若输入的数据过长，将会截掉其超出部分。如果定义了一个 char 数据类型，而且允许该列为空，则该字段被当作 varchar 来处理。

（2）varchar：用 varchar 数据类型可以存储长达 8000 个字符的可变长度字符串，和 char 类型不同的是 varchar 类型的存储空间根据输入数据的实际长度而变化，其定义形式为：

```
varchar(n)
```

例如，定义 varchar(20)，则它对应的字段最多可以存储 20 个字节，如果数据的实际长度不到 20 个字节，系统不会在其后添加空格，因此使用 varchar 类型可以节省空间。

（3）text：用于存储大容量文本数据。当要存储的字符型数据非常巨大，char 和 varchar 已经不能满足其存储要求（大于 8000 字节）时，应该选择 text 数据类型。Text 数据类型的容量可以在 1～（2^{31}–1）（2 147 483 647）个字节范围之内，但实际应用时要根据硬盘的存储空间而定。

在定义 text 数据类型时，不需要指定数据长度，SQL Server 会根据数据的长度自动为其分配空间。

5．Unicode字符串

Unicode（统一字符编码标准）字符集标准，用于支持国际上的非英语语种。每个 Unicode 字符用两个字节为一个存储单位，所以 Unicode 数据类型所占用的存储空间是非 Unicode 数据类型的两倍。

（1）nchar：其定义形式为 nchar(n)，其中 n 表示所有字符所占的存储空间，n 的取值为 1～4000。

（2）nvarchar：其定义形式为 nvarchar(n) ，其中 n 表示所有字符所占的存储空间，n 的取值为 1～4000。

（3）ntext：用于存储大容量文本数据。其理论上的容量为 2^{30}–1（1 073 741 823）个字节。

6．二进制字符串

SQL Server 2008 支持的二进制字符串主要有以下 3 种：

- binary：其定义形式为 binary(n)，数据的存储长度是固定的，即 n+4 个字节，当输入的二进制数据长度小于 n 时，余下部分填充 0。二进制数据类型的最大长度（即 n 的最大值）为 8000，常用于存储图像等数据。
- varbinary：其定义形式为 varbinary(n)。数据的存储长度是变化的，它为实际所输入数据的长度加上 4 字节。其他含义同 binary。
  在输入二进制常量时，需在该常量前面加一个前缀 0x。
- image：用于存储照片、目录图片或者图画，其理论容量为 2^{31}–1（2 147 483 647）个字节。

7．其他数据类型

SQL Server 2008 支持的其他数据类型主要有以下几种：

- sql_variant：用于存储除文本、图形数据和 timestamp 类型数据外的其他任何合法的 SQL Server 数据。此数据类型极大地方便了 SQL Server 的开发工作。
- timestamp：也称作时间戳数据类型。timestamp 是一种自动记录时间的数据类型，主要用于在数据表中记录其数据的修改时间。它提供数据库范围内的唯一值，反映数据库中数据修改的相对顺序，相当于一个单调上升的计数器。
  当用 timestamp 定义的列在更新或者插入数据行时，此列的值会被自动更新，一个计数值将自动添加到此 timestamp 数据列中。
- uniqueidentifier：也称作唯一标识符数据类型。uniqueidentifier 用于存储一个 16 字节长的二进制数据类型。它是 SQL Server 根据计算机网络适配器地址和 CPU 时钟表产生的全局唯一标识符代码（Globally Unique Identifier，GUID），因此该数据类型可以保证在全球范围内不同的计算机所产生的标识符是唯一的。
- xml：可以在 SQL Server 数据库中存储 XML 文档和片段。XML 片段是缺少单个顶级元素的 XML 实例。可以创建 xml 类型的列和变量，并在其中存储 XML 实例。请注意，xml 数据类型实例的存储表示形式不能超过 2 GB。

可以选择将 XML 架构集合与 xml 数据类型的列、参数或变量相关联。集合中的架构用于验证和类型化 XML 实例。在这种情况下，XML 是类型化的。xml 数据类型和关联的方法有助于将 XML 集成到 SQL Server 的关系框架中。

8．特殊数据类型

SQL Server 2008 支持的特殊数据类型主要有以下两种：

- table：用于存储对表或者视图处理后的结果集。使用该数据类型可以利用变量存储一个表，从而使函数或过程返回查询结果更加方便、快捷。
- cursor：数据类型是唯一不能分配给表列的系统数据类型，它只能用于变量和存储过程参数。

注意：上面给出的两种特殊数据类型不能分配给表列。

5.1.6　用户自定义数据类型

SQL Server 2008 允许用户自定义数据类型，用户自定义数据类型是建立在 SQL Server 2008 系统数据类型基础上的。当用户定义一种数据类型时，需要指定该类型的名称、建立在其上的系统数据类型以及是否允许为空等。

SQL Server 2008 为用户提供了两种方法来创建自定义数据类型，即使用 SQL Server Management Studio 图形界面或者通过系统存储过程 sp_addtype。

1．使用SQL Server Management Studio创建用户定义数据类型

例如，在 SQL Server Management Studio 中，定义一个名称为 User_ID 的数据类型，

具体操作步骤如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在 SQL Server Management Studio 中的“对象资源管理器”组件窗口中，依次展开“数据库”|Practice_JWGL|“可编程性”|“类型”节点。右击“用户定义数据类型”，在弹出的快捷菜单中选择“新建用户定义数据类型”命令，打开“新建用户定义数据类型”对话框。

（3）在“架构”文本框中，直接输入或者使用旁边的浏览按钮来选择此数据类型所属的架构，此处采用默认值 dbo。在“名称”文本框中，输入自定义数据的名称 User_ID。在“数据类型”文本框中，选择 nvarchar 数据类型。在“长度”文本框输入 10。如果新建数据类型允许空值，选择“允许 NULL 值”复选框，此处采用默认值不为空。完成设置后的“用户定义数据类型”对话框，如图 5-3 所示。

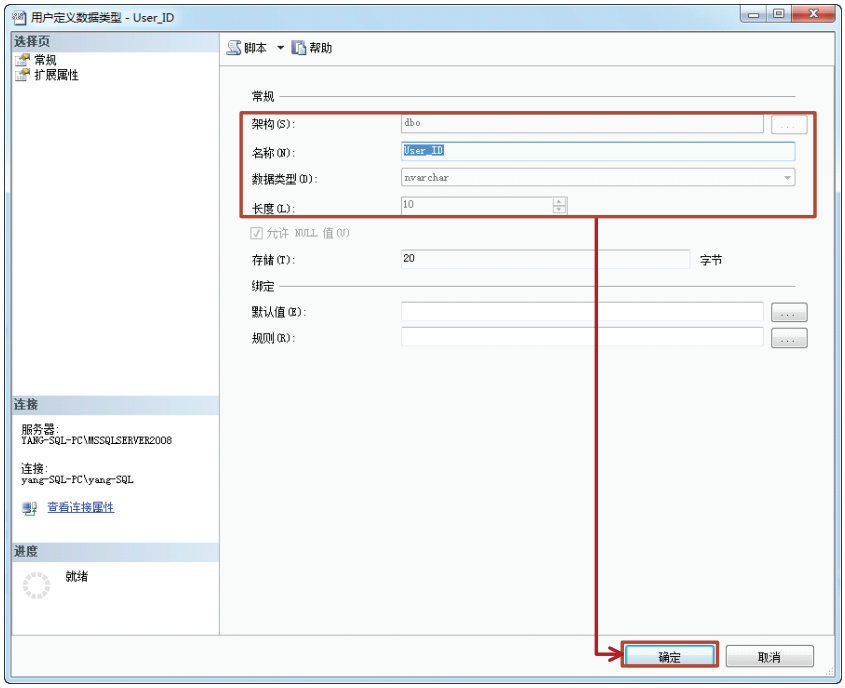

图 5-3　“用户定义数据类型”对话框

注意：默认值和规则不能在 SQL Server Management Studio 中完成创建，必须使用 Transact-SQL 语句来创建。

（4）单击“确定”按钮，即可完成 User_ID 用户定义数据类型的创建操作。

此时，即可在 SQL Server Management Studio 中的“对象资源管理器”组件窗口中的“数据库”|Practice_JWGL|“可编程性”|“类型”|“用户定义数据类型”节点下，看见一个新的数据类型 User_ID。以后，这个数据类型可以在其他地方任意地使用了。

2．利用系统存储过程创建用户自定义数据类型

SQL Server 2008 系统存储过程 sp_addtype 为用户提供了用 Transact-SQL 语句创建自定

义数据类型的途径，其语法形式如下：

```
sp_addtype [@typename=] type,
[@phystype=] system_data_type
[, [@nulltype=] 'null_type']
[, [@owner=] 'owner_name']
```

命令中各参数说明如下所示。

- type：指定用户定义的数据类型的名称。
- system_data_type：指定相应的系统提供的数据类型的名称及定义。注意，不能使用 timestamp 数据类型。当所使用系统数据类型有额外说明时，需要用单引号将其括起来，例如，'varchar(80)'。
- null_type：指定用户自定义数据类型的 null 属性，其值可以为 null、not null 或者 nonull。默认情况下，与系统默认的 null 属性相同。
- owner_name：指定用户自定义数据类型的所有者。用户自定义数据类型的名称在数据库中应该是唯一的，但不同名称的用户自定义数据类型可以有相同的类型定义。

【实例 5-1】自定义一个地址（address）数据类型，最多可以包含 20 个字符。Transact-SQL 代码如下：

```
EXEC sp_addtype address,'varchar(20)','not null'
GO
```

同样，删除用户自定义数据类型也有两种方法，即使用 SQL Server Management Studio 图形界面或者通过系统存储过程 sp_droptype。

3．使用SQL Server Management Studio删除用户定义数据类型

在 SQL Server Management Studio 图形界面中，右击“用户定义的数据类型”节点下的某个用户定义数据类型，在弹出的快捷菜单中选择“删除”命令，打开“删除对象”对话框，如图 5-4 所示。在该对话框中，可以单击“显示依赖关系”按钮查看数据库内是否有对象依赖此数据类型。如果没有，则可以单击“确定”按钮删除该数据类型。

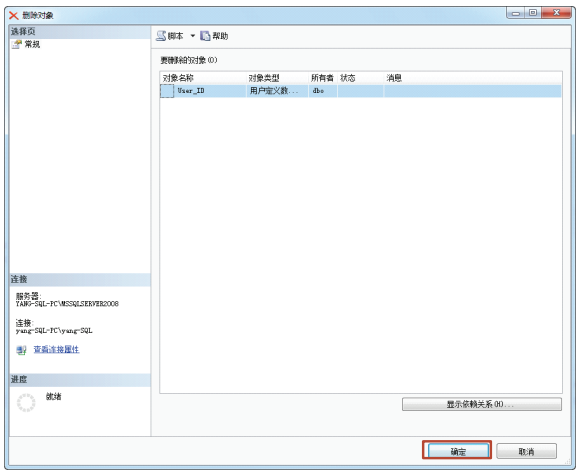

图 5-4　“删除对象”对话框

4．利用系统存储过程删除用户自定义数据类型

使用系统存储过程 sp_droptype 删除自定义数据类型的语法如下：

```
sp_droptype [@typename=] 'type'
```

【实例 5-2】删除自定义的地址（address）数据类型。Transact-SQL 代码如下：

```
EXEC sp_droptype address
GO
```

5.2　数据表定义

在设计完数据表后，接下来的任务就是创建数据表。用户可以使用 SQL Server Management Studio 图形化工具创建表，也可以使用 Transact-SQL 语句方便快速地创建数据表。

5.2.1　使用 SSMS 设计数据表

使用 SQL Server Management Studio 图形工具，可以完成 SQL Server 2008 数据表的建立、修改、查看和删除等绝大多数的工作，这是数据库管理员最常用的操作方式。

1．创建数据表

使用 SQL Server Management Studio 中的表设计器，用户可以方便地创建数据表，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL 节点。

（3）右击“表”节点，在弹出的快捷菜单中选择“新建表”命令，打开“表设计器”。

（4）输入列名，选择数据类型，设置是否允许空以及通过下方的“列属性”窗口来设置列的其他属性，如图 5-5 所示。

技巧：注意观察图 5-5 中右侧的“属性”窗口，该窗口显示了创建的数据表的一些属性信息。用户还可以通过该窗口，设置数据表所存储的位置等信息。如果该窗口没有显示，可以直接按 F4 键或者通过选择“视图”|“属性窗口”命令打开该窗口。

（5）重复第（4）步过程，可以为数据表添加任意多个数据列。

（6）单击工具栏上的按钮或者选择“文件”|“保存”命令，打开“选择名称”对话框。为数据表输入一个名称，如“学生信息”，单击“确定”按钮，即可完成数据表的创建操作。

用户可以在 SQL Server Management Studio 中的“对象资源管理器”窗口中，依次展开“数据库”|Practice_JWGL｜“表”节点，查看刚才建立的数据表。

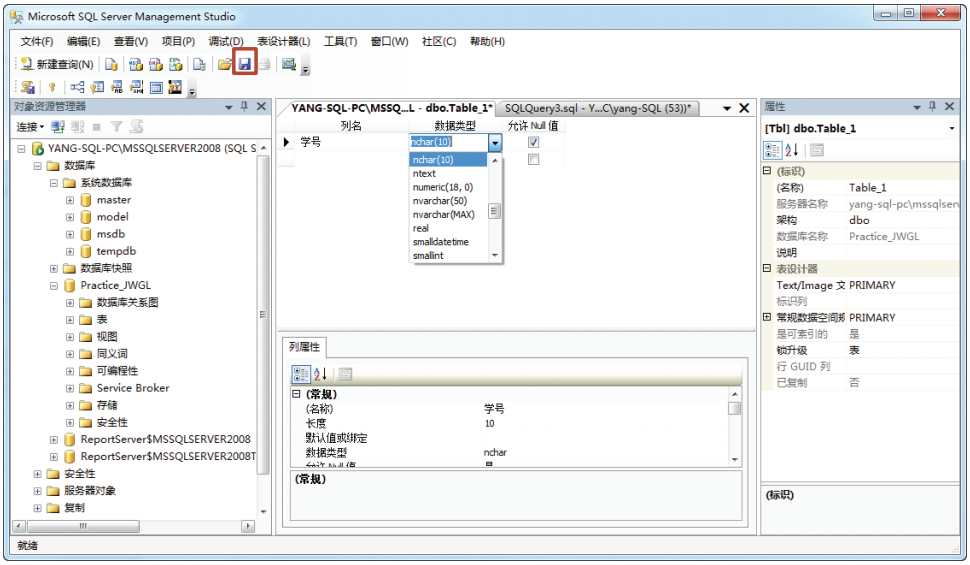

图 5-5　表设计器

2．修改数据表

数据库管理员可以根据需要，修改已经建立的数据表。对数据表的修改，主要包括修改列属性、添加和删除列及修改约束等选项。修改数据表同样可以在“表设计器”窗口中进行，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”节点。

（3）右击“学生信息”表，在弹出的快捷菜单中选择“设计”命令，打开“表设计器”窗口，如图 5-6 所示。

注意：若选择的是“编写表脚本为”|“CREATE 到”|“新查询编辑器窗口”命令，则打开“查询编辑器”窗口，如图 5-7 所示。

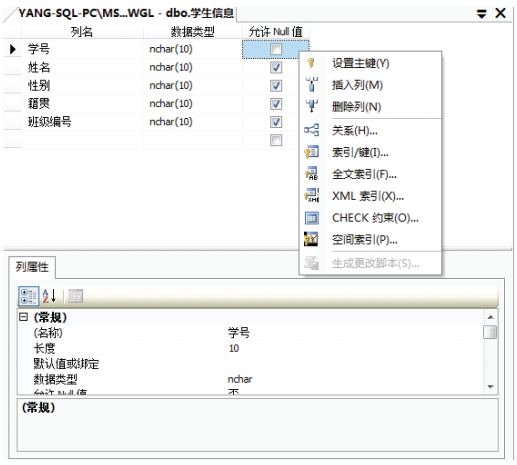

图 5-6　“表设计器”窗口

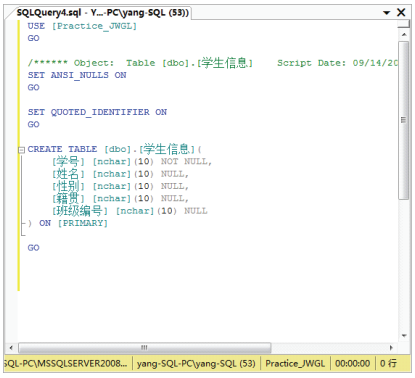

图 5-7　“查询编辑器”窗口

（4）用户可以在“表设计器”窗口中，对数据表的各个列及属性进行修改。

（5）修改完成后，单击工具栏上的按钮，保存对数据表的修改。

3．重命名数据表

在 SQL Server Management Studio 管理工具中，用户可以对创建后的数据表进行重命名。具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”节点。

（3）右击“学生信息”表，在弹出的快捷菜单中选择“重命名”命令。

（4）在数据表名的可编辑文本框中，输入新的数据表名称。

（5）按下 Enter 键或者单击其他数据表，即可完成数据表重命名操作。

4．查看数据表属性

在 SQL Server Management Studio 管理工具中，用户可以查看表的物理信息及常规设置等属性信息，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”节点。

（3）右击“学生信息”表，在弹出的快捷菜单中选择“属性”命令，打开“表属性”对话框，如图 5-8 所示。

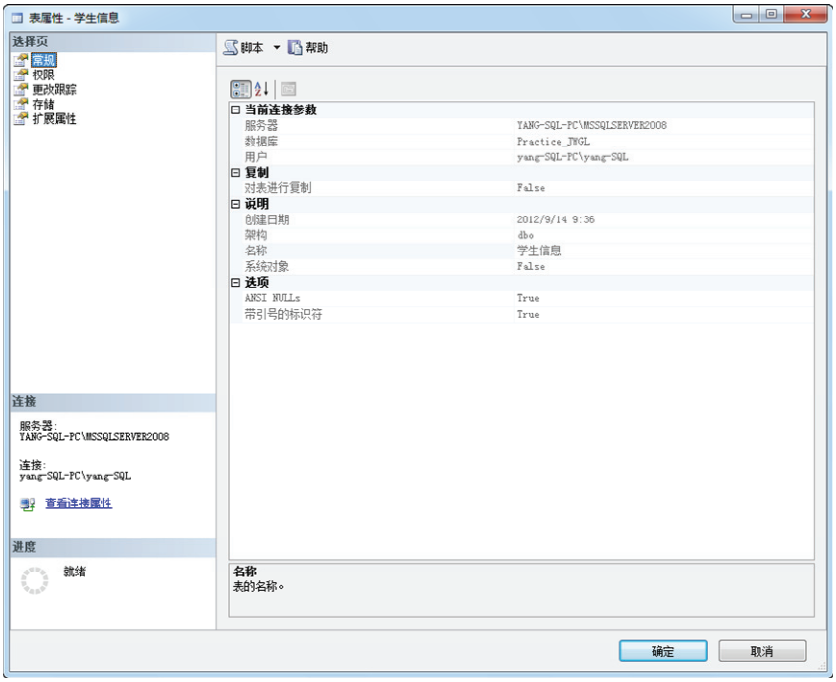

图 5-8　“表属性”对话框

（4）用户可以查看“存储”选项页，“存储”选项页下的条目提供了有关空间使用的详细情况。如“数据空间”显示了数据表在磁盘上说使用的空间数量；“索引空间”显示

了数据表索引空间在磁盘上的大小；“行计数”显示了数据表中数据行的数目。

（5）用户还可以在“权限”选项页中，对数据表的访问权限进行设置。

（6）单击“确定”按钮，完成数据表的查看操作。

5．删除数据表

用户可以删除不再需要或者设计错误的数据表。删除表后，该表的结构定义、数据、全文索引、约束和索引都将从数据库中删除，原来存储该表及其索引的空间可以用来存储其他数据表。使用 SQL Server Management Studio 删除数据表，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”节点。

（3）右击“学生信息”表，在弹出的快捷菜单中选择“删除”命令，打开“删除对象”对话框。

（4）单击“确定”按钮，即可完成删除数据表的操作。

6．编写表脚本

在 SQL Server Management Studio 管理工具中，用户可以依据图形化界面所设计后的数据表的属性，来自动生成对应的 Transact-SQL 脚本语句，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”节点。

（3）右击“学生信息”表，弹出快捷菜单，如图 5-9 所示。

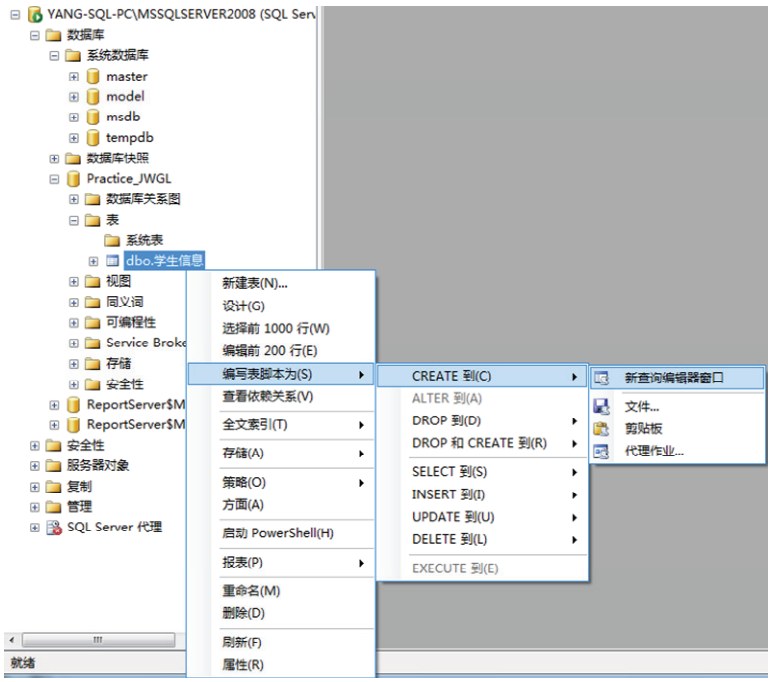

图 5-9　编写表脚本快捷菜单

（4）选择相应的快捷菜单命令，即可完成对应的编写表脚本操作。

5.2.2　使用 SSMS 维护数据表

数据库的系统管理员，日常工作中最基本的一项工作就是对数据表的维护。它包括修改数据表的结构、改变表的排列顺序以及数据的备份等工作。SQL Server Management Studio 提供了方便的图形化界面帮助用户完成这些工作。下面列举一些常见操作，说明其具体的操作步骤。

1．在数据表中插入新列

用户可能在使用数据表一段时间以后，根据业务需求的变化增加数据表的字段，以记录更加丰富的属性信息。使用 SQL Server Management Studio 工具，可以方便地在表中增加数据列，即修改数据表的结构，具体操作过程如下：

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”|“学生信息”|“列”节点。

（3）右击“列”节点，在弹出的快捷菜单中选择“新建列”命令，打开“表设计器”，并且光标定位在最后一行。

（4）输入要增加的字段的“列名”，选择“数据类型”，设置是否“允许空”以及通过下方的“列属性”窗口来设置新增列的其他属性。

（5）重复第（4）步过程，可以为数据表添加任意多个新列。

（6）单击工具栏上的按钮，即可完成插入新列的操作。

技巧：以上操作默认是在“表设计器”的最后依次添加新列。用户也可以根据需要在不同的位置插入新列，其方法是在“表设计器”中，右击某行，在弹出的快捷菜单中选择“插入列”命令，即可在当前行前插入一个新的数据列，也可以通过选择“表设计器”|“插入列”命令来实现。

2．改变数据表中列的排序规则

数据列的“排序规则”用于将“列值”与其他列的值进行比较的各类操作中。如果 SQL Server 数据库实例中的所有用户都是使用的同一语言，则应选择支持该语言的排序规则。例如，如果所有用户都使用的是英语，则应选择英语排序规则。如果 SQL Server 数据库实例中的用户使用不同的语言，则应挑选最能满足多个语种要求的排序规则。例如，如果用户大多数使用西欧语言，则应选用 Latin1_ General 排序顺序。使用 SQL Server Management Studio 工具，设置列的排序规则，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”节点。

（3）右击“学生信息”表，在弹出的快捷菜单中选择“设计”命令，打开“表设计器”。

（4）选择要设置“排序规则”的列，用户可以在该列的“列属性”视图中，查看到该列的排序规则，如图 5-10 所示。

（5）单击“排序规则”文本框右侧的省略号（…）按钮，打开“排序规则”对话框，

如图 5-11 所示。

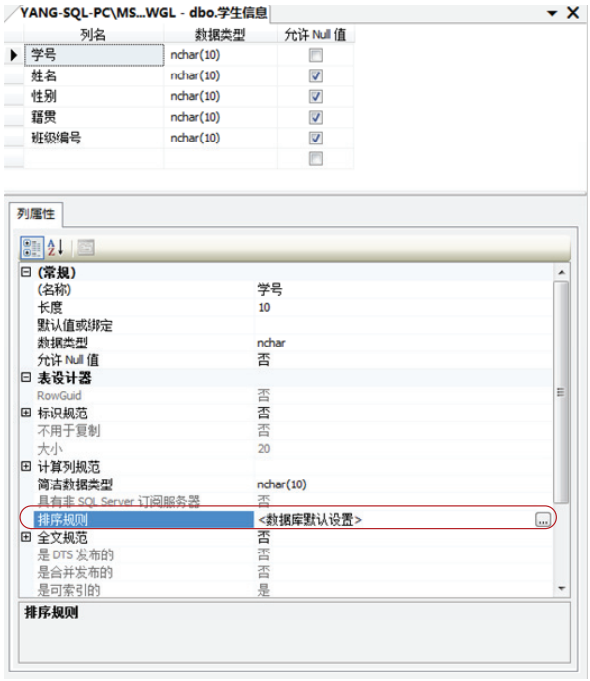

图 5-10 “列属性”视图

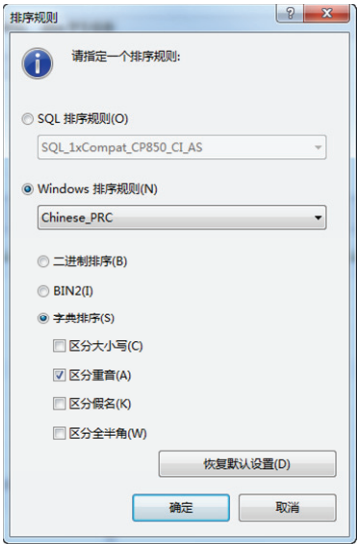

图 5-11 “排序规则”对话框

（6）为选定列设置好排序规则后，单击“确定”按钮，返回“表设计器”视图。

（7）单击工具栏上的按钮，完成“排序规则”的设置。

技巧：请注意改变列的“排序规则”与改变列的“排列顺序”两者之间的区别。前者是指某数据列内数据的排列顺序，而后者是指在表数据视图中，列与列之间的先后顺序。在“表设计器”视图中，可以任意拖动来调整列与列之间的“排列顺序”。

3．自动编号列和标识符列

每个表均可以创建一个系统生成的序号值的标识符列，该序号值以唯一方式标识表中的每一行。例如，当表中插入新行的时候，标识符列可以自动为应用程序生成唯一的客户回执编号。标识符列在其所定义的表中包含的值通常是唯一的。使用 SQL Server Management Studio 工具，可以定义某数据列为标识属性的列，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”节点。

（3）右击目标表，在弹出的快捷菜单中选择“设计”命令，打开“表设计器”，选择要设置为自动编号的数据列。

（4）在“列属性”视图中，展开“标识规范”属性。

（5）单击“是标识”下拉列表框，选择“是”选项，如图 5-12 所示。

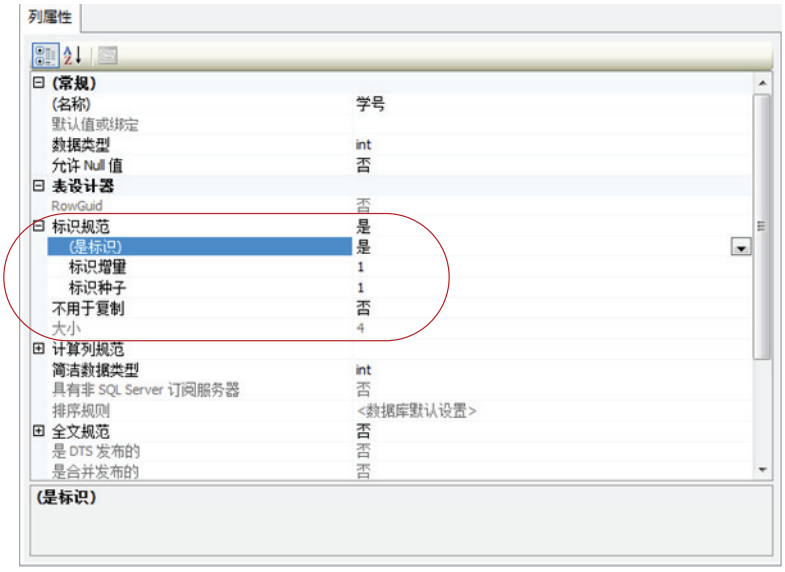

图 5-12　标识规范选项图

（6）在“标识种子”文本框中，输入起始值。此值将赋给表中的第一行，默认情况下将赋值为 1。

（7）在“标识增量”文本框中，输入增量值。此值是基于“标识种子”依次为每个后继行增加的增量，默认情况下将赋值为 1。

（8）单击“保存”按钮，即可完成自动编号列的设置。

注意：如果单击“保存”按钮无法保存，弹出如图 5-13 所示的错误提示，则需要更改或者启用“阻止保存要求重新创建表的更改”选项。方法是打开 SQL Server 2008，选择“工具”|“选项”|Designers|“表设计器和数据库设计器”，把“阻止保存要求重新创建表的更改”的勾选去掉然后单击“确定”按钮即可，如图 5-14 所示。

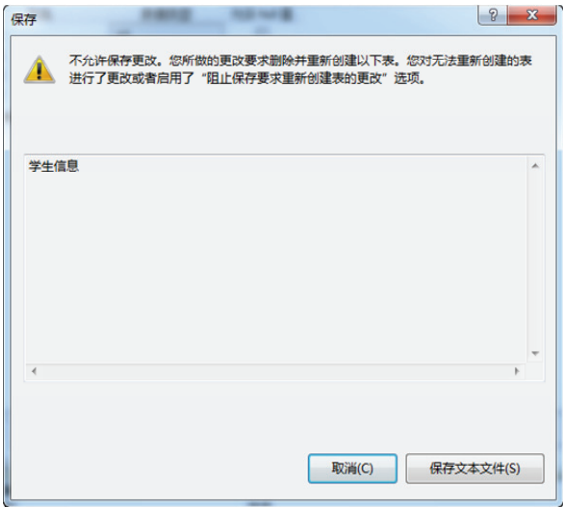

图 5-13　错误提示

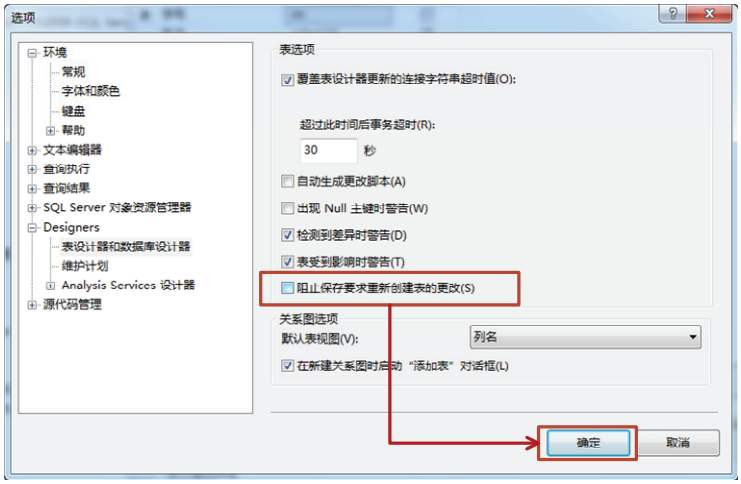

图 5-14　“标识规范”选项图

4．可计算的列

计算列由同一数据表中其他列的表达式计算得来。该表达式可以是非计算列的列名、常量、函数，也可以是用一个或多个运算符连接的这些元素的任意组合。例如，“教师工资”表中的“实发工资”列可以定义为“应发工资–扣除部分”。使用 SQL Server Management Studio 工具，指定可计算列，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”节点。

（3）右击“教师工资信息”表，在弹出的快捷菜单中选择“设计”命令，打开“表设计器”。选择“实发工资”数据列。

（4）在“列属性”视图中，展开“计算所得的列规范”属性。

（5）在“公式”文本框中输入“应发工资–扣除部分”。

（6）在“是持久的”下拉列表框中选择“是”或“否”，以指示该计算结果是否存入物理磁盘中。

（7）单击工具栏上的按钮，完成可计算的列的设置操作。

5．删除数据列

和插入新列一样，用户可以通过 SQL Server Management Studio 工具，使用以下两种方法来删除数据列。在“对象资源管理器”中删除数据列，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”|“学生信息”|“列”节点。

（3）右击要删除的列，在弹出的快捷菜单中选择“删除”命令，打开“删除对象”对话框。

（4）单击“确定”按钮，即可删除该数据列。

在“表设计器”中删除数据列，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”节点。

（3）右击“学生信息”表，在弹出的快捷菜单中选择“设计”命令，打开“表设计器”。

（4）右击要删除的数据列，在弹出的快捷菜单中选择“删除列”命令。也可以选择“表设计器”|“删除列”命令来实现。

（5）单击工具栏上的按钮，即可完成该数据列的删除操作。

5.2.3　使用 Transact-SQL 设计与维护数据表

除了使用 SQL Server Management Studio 工具之外，用户还可以使用 Transact-SQL 语句来设计与维护数据表。下面简要介绍如何使用 Transact-SQL 语句来创建和管理数据表。

1．使用CREATE TABLE语句创建表

Transact-SQL 语言提供了创建数据表的语句 CREATE TABLE，其语法格式如下：

```
CREATE TABLE [database_name.]table_name
(
   column_name data_type
   [ NULL|NOT NULL ]
   [ DEFAULT constant_expression ]
   [PRIMARY KEY | UNIQUE],
   ...n,--表示可以定义许多列
)
FOREIGN KEY(column_name1[ ,...n ])
REFERENCES referenced_table_name(ref_column_name [ ,...n ])
```

各参数说明如下。

- database_name：指定要创建数据表所属的数据库名称，如果省略，则默认为 Master 数据库。故在执行语句前一定要用 USE database_name 或在工具栏中选择相应的数据库，来确定创建数据表所在的数据库。
- table_name：要创建的新表的名称。
- column_name：要创建新列的名称。
- data_type：要创建新列的数据类型。
- NULL|NOT NULL：是否可以为空。
- DEFAULT constant_expression：设置字段默认值。
- PRIMARY KEY | UNIQUE：PRIMARY KEY 确定为主索引，UNIQUE 确定为唯一索引。
- column_name1：被确定为外键的列名。
- referenced_table_name：与之相关联的另外一个表。
- ref_column_name：与之相关联的另外一个表的主键。

【实例 5-3】在 Practice_JWGL 数据库中创建一个“学生信息”表，并确定学号为主键。Transact-SQL 代码如下：

```
--打开 Practice_JWGL 数据库
USE Practice_JWGL
GO
```

```
--创建学生信息数据表
CREATE TABLE [dbo].[学生信息]
(
    [学号] [nvarchar](10)NOT NULL PRIMARY KEY,
    [姓名] [nvarchar](10)NULL,
    [性别] [nvarchar](2)NULL,
    [籍贯] [nvarchar](20)NULL,
    [班级编号] [nvarchar](6)NULL,
)
GO
```

2．使用ALTER TABLE语句修改表

Transact-SQL 语言提供了修改数据表定义的语句 ALTER TABLE，其语法格式如下：

```
ALTER TABLE [database_name.]table_name
(
   ALTER COLUMN column_newdesp      --修改已经存在列的属性
   |[ADD new_column_desp]           --新增加一列
   |[DROP COLUMN column_name]       --删除列
)
```

各参数说明如下。

- database_name：指定要修改的数据表所属的数据库名称，如果省略，则默认为 Master 数据库。故在执行语句前一定要用 USE database_name 或在工具栏中选择相应的数据库，来确定创建数据表所在的数据库。
- table_name：要修改的表名称。
- column_newdesp：对列的属性的新定义。
- new_column_desp：要创建的新列的属性描述。
- column_name：要删除的列。

【实例 5-4】在 Practice_JWGL 数据库中创建一个学生信息表 Student，然后增加一个出生日期字段（出生日期，Datatime，Null），修改姓名的长度为 20，最后删除籍贯字段。Transact-SQL 代码如下：

```
--打开 Practice_JWGL 数据库
USE Practice_JWGL
GO

--创建 Student 表
CREATE TABLE [dbo].[Student]
(
    [学号] [nvarchar](10)NOT NULL PRIMARY KEY,
    [姓名] [nvarchar](10)NULL,
    [性别] [nvarchar](2)NULL,
    [籍贯] [nvarchar](20)NULL,
    [班级编号] [nvarchar](6)NULL,
)
GO

--增加字段
ALTER TABLE [dbo].[Student]
ADD 出生日期 datetime  NULL
```

```
GO

--修改字段属性
ALTER TABLE [dbo].[Student]
ALTER COLUMN 姓名 nvarchar(20)NULL
GO

--删除字段
ALTER TABLE [dbo].[Student]
DROP COLUMN 籍贯
GO
```

3．利用系统存储过程sp_rename重命名表和表中的数据列

系统存储过程 sp_rename 可以对数据表和数据表中的数据列进行重命名操作，其语法格式如下：

```
sp_rename 'old_name','new_name'
```

【实例 5-5】将前面创建的 Student 数据表，重命名为 Students。Transact-SQL 代码如下：

```
--打开 Practice_JWGL 数据库
USE Practice_JWGL
GO

--重命名数据库
EXEC sp_rename 'student','students'
GO
```

【实例 5-6】将前面创建的 Student 数据表中的“班级编号”数据列，重命名为“班号”。Transact-SQL 代码如下：

```
--打开 Practice_JWGL 数据库
USE Practice_JWGL
GO

--重命名字段
EXEC sp_rename 'student.班级编号','班号'
GO
```

注意：在以上两例中，都必须先打开 Practice_JWGL 数据库，否则会找不到对象。

4．使用DROP TABLE语句删除表

Transact-SQL 语句中删除表的语句是 DROP TABLE，其语法格式如下：

```
DROP TABLE [database_name.]table_name [,...n]
```

各参数说明如下。

- database_name：表所属的数据库名称。
- table_name：要删除的表的名称。
- n：表示在任何数据库中删除多个表。如果删除的表引用了另一个表的主键，则另

一个表也将被删除。

【实例 5-7】将前面创建的 Student 数据表，从 Practice_JWGL 数据中删除。Transact-SQL 代码如下：

```
--打开 Practice_JWGL 数据库
USE Practice_JWGL
GO

--删除 student 数据表
DROP TABLE student
GO
```

5．利用系统存储过程SP_help查看表的信息

系统存储过程 SP_help 可以提供指定数据库对象的信息，也可以提供系统或者用户定义的数据库类型的信息，其语法形式如下：

```
sp_help [[@objname=]name]
```

SP_help 存储过程只适用于当前数据库。其中，[@objname=]name 子句用于指定对象的名称。如果不指定，默认为当前数据库中的所有对象名称、对象所有者和对象的类型。

【实例 5-8】分别查看 Practice_JWGL 数据库中所有对象信息和学生信息表信息。Transact-SQL 代码如下：

```
USE Practice_JWGL
GO

--显示 Practice_JWGL 数据库的所有信息
EXEC sp_help
GO

--显示学生信息表的信息
EXEC sp_help 学生信息
GO
```

执行完毕后显示的结果如图 5-15 所示。从图 5-15 中下方可以看到有两个窗格。

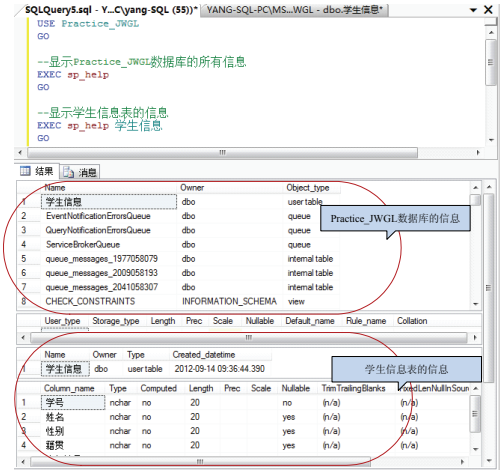

图 5-15 【实例 5-8】执行结果图

- 第 1 个窗格中显示的是 Practice_JWGL 数据库中所有对象信息。
- 第 2 个窗格中显示的是 Practice_JWGL 数据库中“学生信息”表的所有信息。

5.3　管理数据表中的数据

数据库管理的主要内容是数据，用户或管理员会经常操作、查询表和视图中的数据。SQL Server Management Studio 工具为用户提供了简便的图形化方式来实现这一功能，用户也可以在查询编辑器中使用 Transact-SQL 中的数据操纵语言（DML）来实现。

说明：本节主要介绍如何使用 SQL Server Management Studio 工具来实现表中数据的查询与维护功能，而关于 Transact-SQL 中的数据操纵语言（DML）部分将在第 13 章详细介绍。

5.3.1　查看数据

使用 SQL Server Management Studio 工具可以查看数据表中的所有记录信息，也可以只查看自己需要的数据。

1．查看数据表中的所有记录

用户可以使用 SQL Server Management Studio 工具，很方便地浏览数据表中的所有数据，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”节点。

（3）右击“学生信息”表，在弹出的快捷菜单中选择“打开表”命令。

（4）在“文档窗口”视图中，将显示“学生信息”表中的所有记录的详细信息，如图 5-16 所示。

结果　消息

	学号	姓名	性别	籍贯	班级编号
1	0511101020	李建平	男	四川	051110
2	0511101046	罗川伟	男	四川	051110
3	0516102073	李佩秋	女	云南	051610
4	0611101007	袁娜	女	陕西	061110
5	0612102055	杨多玲	女	重庆	061210
6	0612102061	李江涛	男	甘肃	061210
7	0702102020	谢涛	男	重庆	070210
8	0702102052	罗伟	男	甘肃	070210

图 5-16　浏览数据结果

2．查询数据表中的记录

SQL Server Management Studio 工具还提供了“视图”工具，用户可以查看指定条件的数据表记录。使用“视图”工具，查看表的指定记录的具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL 节点。

（3）右击“视图”，在弹出的快捷菜单中选择“新建视图”命令。

（4）依次单击工具栏上的“显示关系图窗格”、“显示条件窗格”、“显示 SQL 窗格”和“显示结果窗格”按钮，打开“查询设计器”视图。

- 显示关系图窗格：可视化图形的方式显示数据表、视图以及表间关系等数据对象。
- 显示条件窗格：对可以用的表、列、视图、别名、排序以及筛选等信息进行设置的操作界面。
- 显示 SQL 窗格。展示了通过操作界面处理而自动生成的 Transact-SQL 语句，用户也可以直接在该窗格里面编写 Transact-SQL 语句来实现查询功能。
- 显示结果窗格。用于以表格的形式显示视图的执行结果。

技巧：在每个窗格中右击空白处，都会弹出相应的快捷菜单，提供我们需要的功能。

（5）设置好输出信息、排序以及筛选条件的“查询学生信息”视图，如图 5-17 所示。

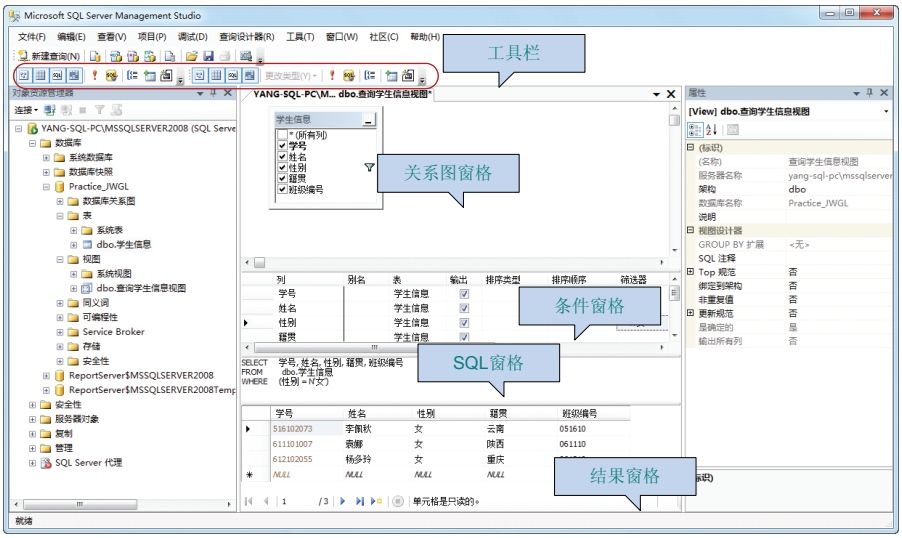

图 5-17 “查询设计器”视图

（6）单击工具栏上的 按钮或者选择“查询设计器”|“执行 SQL”命令，运行查询。在“结果窗格”中将显示指定条件的数据记录。

技巧：还可以通过“查询设计器”工具栏上的其他按钮，完成相应的操作。

5.3.2　更新数据

对数据表中数据的维护操作，是 SQL Server 2008 用户要掌握的基本操作。一般的维护过程包括查找表，根据查询条件查看记录，对表数据记录进行添加、修改和删除等。

1．添加新的记录

使用 SQL Server Management Studio 工具的图形界面，可以非常方便地向数据表增加记录，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”节点。

（3）右击“学生信息”表，在弹出的快捷菜单中选择“打开表”命令，打开“学生信息”表的“文档窗口”视图。

（4）在“文档窗口”视图中的最后一行，即标有“*”的数据行中，输入每个字段的值。

（5）输入完毕后，选择其他数据行，使当前记录的数据行失去焦点，即可完成添加新的记录。

2．修改现有记录

修改现有的数据表中的记录，也是数据库管理员的一项基本技能。在实际应用中，用户一般是通过客户端应用程序对表中的数据进行修改。这里仅介绍如何使用 SQL Server Management Studio 工具来修改表中的数据。

（1）在打开的表的“文档窗口”视图中，查找到要修改的数据记录。

（2）单击要修改的数据项，激活并修改。

（3）修改完毕后，单击其他数据行，使已修改的数据行失去焦点，SQL Server Management Studio 将自动提交，以完成修改操作。

3．删除记录

删除现有表中的数据记录和修改表中的数据行相同。在实际应用中，用户一般是通过客户端应用程序对表中的数据进行删除操作。这里仅介绍如何使用 SQL Server Management Studio 工具来删除表中的数据行。

（1）在打开的表的“文档窗口”视图中，查找到要修改的数据记录行。

（2）右击要删除的数据行，在弹出的快捷菜单中选择“删除”命令，弹出删除确认对话框。

（3）单击“是”按钮，即可完成数据行的删除操作。

技巧：可以配合按 Ctrl 键或 Shift 键，来实现多个记录的选择和删除等操作。

5.4　索　　引

索引是数据库中的重要数据对象，通过建立索引可以提高数据查询速度或者其他操作的效率。SQL Server 2008 数据库提供了多种索引类型。

5.4.1　什么是索引

索引可以创建在任意表和视图的列字段上，索引中包含键值，这些键值存储在一种数据结构（B-树）中，通过键值可以快速地找到与键值相关的数据记录。

SQL Server 提供了两种形式的索引，即聚集索引（Clustered）和非聚集索引（Nonclustered）。聚集索引根据键的值对行进行排序，所以每个表只能有一个聚集索引。非聚集索引不根据键值排序，索引数据结构与数据行是分开的。由于非聚集索引的表没有按顺序进行排列，所以查找速度明显低于带聚集索引的表。SQL Server 2008 提供的索引类型具体包括以下几种。

- 聚集索引：根据索引的键值，排序表中的数据并保存。
- 非聚集索引：索引的键值包含指向表中记录存储位置的指针，不对表中数据排序，只对键值排序。
- 唯一索引：保证索引中不含有相同的键值，聚集索引和非聚集索引都可以是唯一索引。
- 包含列的索引：一种非聚集索引，其中包含一些非键值的列，这些列对键值有辅助作用。
- 全文（full-text）索引：由 Microsoft 全文引擎（full-text engine）创建并管理的一种基于符号的函数（token-based functional）索引，支持快速地字符串中单词的查找。
- XML 索引：XML 数据列中的 XML 二进制大对象（BLOBs）。

索引的建立有利也有弊，建立索引可以提高查询速度，但过多地建立索引会占据很多的磁盘空间。所以在建立索引时，数据库管理员必须权衡利弊，考虑让索引带来的有利效果大于带来的弊病。

下列情况适合建立索引：

- 经常被查询搜索的列，如经常在 WHERE 子句中出现的列。
- 在 ORDER BY 子句中使用的列。
- 外键或主键列。
- 值唯一的列。

下列情况不适合建立索引：

- 在查询中很少被引用的列。
- 包含太多重复值的列。
- 数据类型为 bit、text、image 等的列不能建立索引。

5.4.2　维护索引

用户可以使用 SQL Server Management Studio 工具或者 Transact-SQL 语句创建和管理索引。下面介绍如何使用 SQL Server Management Studio 工具来创建和管理索引，Transact-SQL 语句部分将在第 6 章中具体介绍。使用 SQL Server Management Studio 工具为“学生信息”表创建一个索引，使其按照性别排序，具体操作过程如下。

（1）打开 SQL Server Management Studio 并连接到数据库引擎服务器。

（2）在“对象资源管理器”中，依次展开“数据库”|Practice_JWGL|“表”|“学生信息”表节点。

（3）右击“索引”节点，在弹出的快捷菜单中选择“新建索引”命令，打开“新建索引”对话框。

（4）在“索引名称”文本框中，输入“索引_性别”。

（5）在“索引类型”下拉列表框中，选择索引类型，这里，采用系统默认值“非聚集”。

（6）单击“添加”按钮，打开“选择列”对话框，并选择“性别”列，如图 5-18 所示。

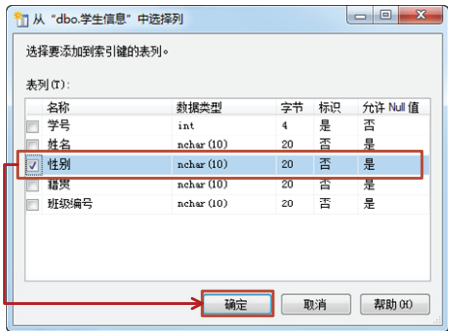

图 5-18　“选择列”对话框

（7）单击“确定”按钮，返回“新建索引”对话框，可以在其中设置索引键的属性，如图 5-19 所示。另外，用户还可以通过选择对话框左侧的“选项”选项页来设置索引的参数；通过“包含性列”选项页来为索引添加非键值辅助列；通过“存储”选项页来选择索引存储文件组等参数。

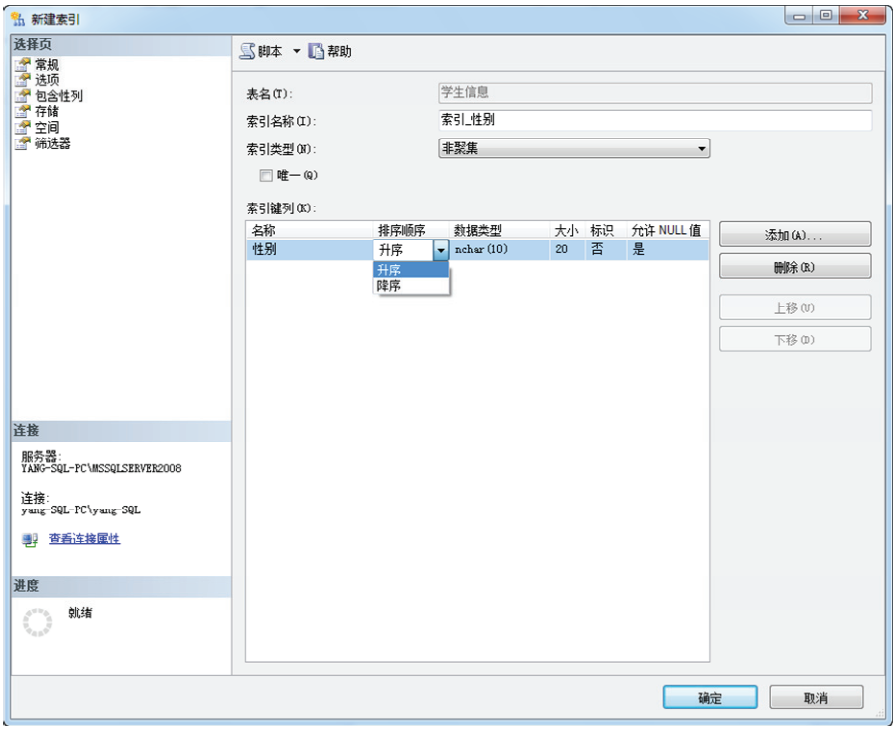

图 5-19　在“新建索引”对话框中设置索引键列的属性

（8）单击“确定”按钮，完成该索引的创建操作。

此时，可以在“学生信息”表的“索引”节点下，看到刚创建好的“索引_性别”索引。用户还可以右击该索引，在弹出的快捷菜单中选择相应的菜单命令，完成其他相应的操作（编写索引脚本为、禁用该索引、重命名该索引、查看该索引的属性和删除该索引等），如图 5-20 所示。

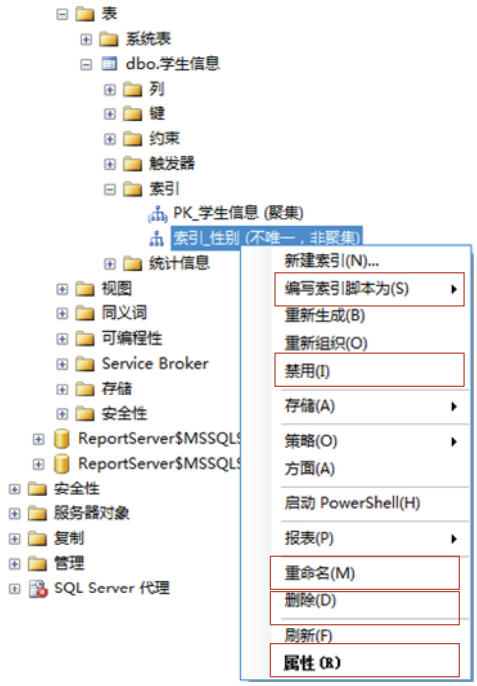

图 5-20　右击“索引_性别”索引后的快捷菜单

5.5　小　　结

数据表是 SQL Server 数据库中的一种重要的对象，存储数据库中所有的数据。SQL Server 中的数据表分为永久表和临时表，前者创建后一直存储在数据库文件中，直到删除，后者在用户退出或系统修复时被自动删除。数据表的创建、维护、管理可以利用图形工具向导，也可利用 SQL 语言代码。索引是一种特殊类型的数据对象，利用索引可提高数据表中数据访问的速度，同时还能够强制实施某些数据完整性，重点是用户要掌握 SQL Server 2008 数据表的基础知识以及表的常见操作。第 6 章将学习 SQL Server 2008 数据完整性的基础知识。

5.6　习　　题

1．数据表在结构上有什么特点？
2．创建一数据表的步骤是怎样的？
3．如何控制数据表中数据的完整性？
4．SQL Server 2008 中系统表有哪些？作用分别是什么？
5．SQL Server 2008 数据表中数据的类型有哪些？
6．创建一个名为 tongxunlu 的数据库，并在数据库中创建一个名为 personal 的表，其

中包括以下字段：编号（int，自动编号）、姓名（char(8)，NOT NULL）、性别（char(2)，NOT NULL）、出生日期（datetime）、联系方式（char(16)）、备注（text，NULL）。

7．通过建立查询向“个人信息”表中增加“家庭住址”字段（text(20)，null）。

8．从“个人信息”表中删除“性别”字段。

9．在数据库 school 中添加以下数据表：

- Students(sid nvchar(10),sname nvchar(12),email nvchar(20),grade nvchar(6))
- Teachers(tid nvchar(10),tname nvchar(12),email nvchar(20),salary decimal(7,2))
- Courses(cid char(8),cname nvchar(40),hour int(3))
- Choices(no int(6),sid nvchar(10),tid nvchar(10),cid char(8),score decimal(6,2))

10．在上述数据表中添加如下记录。

Students表

sid	sname	email	grade
0515101142	蔡雪松		05 工本 2 班
0514102002	纪川		05 科本 1 班
0511102052	温俊文		05 信本 2 班
0614103009	周忆杭		06 产本 1 班
0613204003	钟秋虹		06 城园 1 班
0615103026	刘文平		06 电本 1 班
0615101037	孟宇		06 工本 1 班
0616101039	张瑜		06 工商 1 班
0614102008	梁超		06 科本 1 班
0604101042	贺飞		06 食本 1 班
0616103013	万芳芳		06 市本 1 班
0616103023	杨勇		06 市本 1 班
0616103040	唐波		06 市本 1 班
0615102049	唐鸿波		06 水本 1 班
0507201081	马娟		06 英本 2 班
0716102021	杨小龙		07 财本 1 班
0716102033	侯小龙		07 财本 1 班
0716102055	肖创		07 财本 2 班
0716102088	沈兴建		07 财本 2 班

Teachers表

tid	tname	email	salary
XCC03800003	张三		1500.00
XCC03600013	李四		800.00
XCC03700060	王五		1250.00
XCC00700012	孙六		2010.00
XCC00700013	赵七		2500.00
XCC03700003	吴八		780.00

续表

tid	tname	email	salary
XCC02100006	刘九		4000.00
XCC02100007	龙十		3500.00
XCC00700008	岳一		640.00
XCC00800003	黄二		900.00
XCC00700015	葛十一		1200.00

Courses表

cid	cname	hour
11140080	操作系统原理	32
11141000	算法设计与分析	32
11141010	汇编语言程序设计	32
11141040	计算机通信与网络	40
11141070	Unix 操作系统	24
11141190	信息安全基础	32
11156030	C++程序设计	40
11141080	Java 程序设计	48
11140060	数据结构	32
11141120	UML 统一建模语言	32

Choices表

no	sid	tid	cid	score
1	0515101142	XCC03800003	11140080	
2	0514102002	XCC03600013	11141000	
3	0511102052	XCC03700060	11141010	
4	0614103009	XCC00700012	11141040	
5	0613204003	XCC00700013	11141070	
6	0615103026	XCC03700003	11141190	
7	0615101037	XCC02100006	11156030	
8	0616101039	XCC02100007	11141080	
9	0614102008	XCC00700008	11140060	
10	0604101042	XCC00800003	11141120	
11	0616103013	XCC00700015	11140080	
12	0616103023	XCC03800003	11141000	
13	0616103040	XCC03600013	11141010	
14	0615102049	XCC03700060	11141040	
15	0507201081	XCC00700012	11141070	
16	0716102021	XCC00700013	11141190	
17	0716102033	XCC03700003	11156030	
18	0716102055	XCC02100006	11141080	
19	0716102088	XCC02100007	11140060	
20	0616101039	XCC00700008	11141120	