第 3 章

创建与维护数据表

技能要求

1. 掌握 SQL Server 2012 数据库的常用数据类型；
2. 掌握使用对象资源管理器和 T-SQL 语句创建表及修改表的方法；
3. 掌握使用对象资源管理器和 T-SQL 语句向数据表中增加和删除记录的方法。

3.1 【实例 3】创建数据表

实例说明

本案例将介绍在 SQL Server 2012 中使用 SQL Server Management Studio 实现对表的创建操作，在【实例 2】创建的“教学管理”数据库中创建 4 个表。

(1) 学生(student)信息表：反映了学生个人基本信息，表结构如表 3-1 所示。

表 3-1 学生信息表(student)

编号	字 段 名 称	字段说明	类 型	是否允许空	长 度
1	student_id	学号	nchar(10)	否	10
2	student_name	姓名	nchar(10)	否	10
3	sex	性别	nchar(2)	否	2
4	birthday	出生日期	date	否	8
5	rxdate	入学日期	date	否	8
6	phone	电话	nchar(20)	否	20
7	grade	年级	nchar(10)	否	10
8	department	专业	nchar(20)	否	20
9	political	政治面貌	nchar(10)	是	10
10	native	籍贯	nchar(10)	是	10
11	nation	民族	nchar(10)	是	10

（2）课程(course)信息表：反映了学校的课程信息，表结构如表 3-2 所示。

表 3-2　课程信息表(course)

编号	字 段 名 称	字段说明	类　　型	是否允许空	长　　度
1	course_id	课程号	nchar(10)	否	10
2	course_name	课程名称	nchar(50)	否	10
3	credit	学分	int	否	3

（3）成绩表(score)：反映了学生考试成绩信息，表结构的效果如表 3-3 所示。

表 3-3　成绩信息表(score)

编号	字 段 名 称	字段说明	类　　型	是否允许空	长　　度
1	student_id	学号	nchar(10)	否	10
2	course_id	课程号	nchar(10)	否	10
3	score	成绩	int	否	3

（4）教师(teacher)授课表：反映了教师讲授课程信息，表结构如表 3-4 所示。

表 3-4　教师信息表(teacher)

编号	字 段 名 称	字段说明	类　　型	是否允许空	长　　度
1	teacher_id	教师号	nchar(10)	否	10
2	teacher_name	老师姓名	nchar(10)	否	10
3	course_id	课程号	nchar(10)	否	10
4	period	学时数	int	是	3
5	class	班级	nchar(20)	是	20
6	profession	职称	nchar(10)	是	10

实例操作

使用对象资源管理器创建表的操作步骤如下：

（1）在对象资源管理器中展开“教学管理”数据库，右击“表”节点，在弹出的快捷菜单中选择“新建表”命令，如图 3-1 所示。

（2）在弹出的“设计表”窗口中输入每一个字段的相关信息。参照表 3-1 定义所有的字段，其中 student_id 为学生信息(student 表)的主键，选中 student_id 字段，单击工具栏上的主键设置按钮，将字段 student_id 设置成学生信息的主键，如图 3-2 所示。

（3）在表中所有字段定义完成后，单击工具栏上的保存按钮，在弹出的“选择名称”对话框中输入创建的表名 student。

（4）最后，单击“确定”按钮，完成创建表的操作。右击“表”节点，在弹出的快捷菜单

中选择“刷新”命令，在对象资源管理器中数据库“教学管理”的表对象中就可以找到刚创建的学生信息表 student。

(5) 重复步骤(1)～(4)，依次创建课程(course)信息表、成绩表(score)和教师(teacher)授课表。

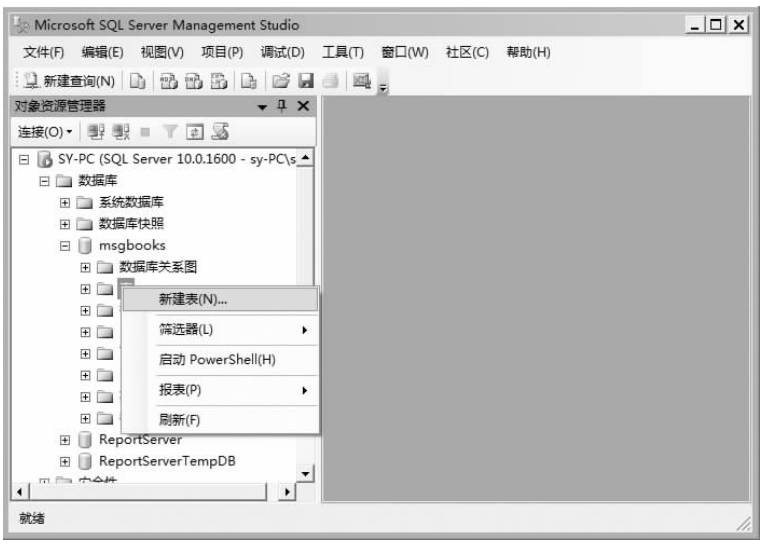

图 3-1 创建表

图 3-2 学生信息表(student 表)

知识学习

3.1.1　表（Table）

表是在日常工作和生活中经常使用的一种表示数据及其关系的形式，例如，表 3-5 就是一个学生基本信息表(student)。

表 3-5　学生信息表(student)

student_id	student_name	sex	birthday	rxdate	phone	grade	department	political	native	nation
2014711025	王明	女	1990-09-15	2014-09-01	138×××××678	14 秋	计算机网络	团员	北京	汉
2013711026	张红	女	1992-08-01	2012-09-01	135×××××234	12 秋	工商管理	团员	上海	汉
2012911002	李一冰	男	1991-06-26	2012-09-01	123×××××912	12 秋	计算机网络	党员	天津	回
2012911005	赵静	女	1991-02-24	2012-09-01	123×××××999	12 秋	工商管理	团员	山西	汉
2013711007	沈明明	男	1990-07-09	2013-03-01	135×××××234	13 春	法学	团员	北京	蒙古
2014911029	王丽丽	女	1991-12-04	2014-03-01	136×××××234	14 春	法学	党员	天津	回
2013711028	李伟	男	1989-12-28	2013-09-01	123×××××567	13 秋	计算机网络	党员	湖北	汉
2012711003	曾天昊	男	1991-03-27	2012-09-01	135×××××901	12 秋	工商管理	团员	山东	汉
2014911115	李杰	女	1990-05-08	2014-03-01	136×××××124	14 春	法学	团员	湖南	汉

1. 表的基本结构

(1) 表名

每个表都有一个名字，以标识该表。

(2) 字段

每个记录由若干个数据项(列)构成，构成记录的每个数据项就称为字段(Field)，每个字段都有其数据类型，即该字段的取值类型。

(3) 空值

空值(NULL)通常表示未知、不可用或将在以后添加的数据。

(4) 关键字

用于唯一标识实体的属性或属性集称为实体的关键字或码。在学生信息表中，student_id(学号)字段的值对表中所有记录唯一，这个字段可以作为关键字，将表中的不同记录区分开来。

2. 表示实体的表和表示联系的表

数据库不仅要反映数据本身的内容，而且要反映数据之间的联系。在关系数据库中，包含了反映实体信息的表和反映实体之间联系的表。例如，在“教学管理”数据库中，学生信息表表示了学生这一实体的信息，课程信息表示了学生可选修课程实体的信息，如表 3-6 所示。

表 3-6　课程信息表(course)

course_id	course_name	credit	course_id	course_name	credit
C001	C++ 程序设计	4	F002	国际经济法	3
C002	软件工程	3	F003	商法	4
C003	数据库基础与应用	3	F004	证据学	3
C004	Java 程序设计	4	G001	西方经济学	3
F001	中国法制史	4	G002	企业战略管理	4

此外,还需要一个表示学生实体与课程实体联系的表——成绩表,如表 3-7 所示。

表 3-7　成绩表(score)

student_id	course_id	score	student_id	course_id	score
2012711003	G001	89	2013711028	C004	85
2012711003	G002	90	2013711028	C002	83
2012711003	G004	43	2013911118	G001	92
2012911002	C001	95	2014911029	F001	79
2014711025	C002	91	2014911029	F002	56
2014711025	C003	79	2014911115	F001	86

3.1.2　数据类型

在 SQL Server 数据库中存储的数据都具有一定的数据类型,数据类型决定了数据的存储格式,以及数据所占用的空间,代表了各种不同的信息类型。SQL Server 2012 提供了一系列的系统数据类型,同时用户也可以根据需要在系统数据类型的基础上创建自己定义的数据类型。SQL Server 2012 提供的常用系统数据类型大致可以分为以下几类。

1. 字符数据类型

字符数据类型包括 varchar、char、nvarchar、nchar、text 及 ntext。这些数据类型用于存储字符数据。使用 varchar 数据类型会稍增加一些系统开销。通常的原则是,任何小于或等于 5 个字节的列应存储为 char 数据类型,而不是 varchar 数据类型。如果超过这个长度,使用 varchar 数据类型的好处将超过其额外开销。

nvarchar 数据类型和 nchar 数据类型的工作方式与对等的 varchar 数据类型和 char 数据类型相同,但这两种数据类型可以处理国际性的 Unicode 字符。它们需要一些额外开销,没有必要的话应避免使用 Unicode 列。

表 3-8 列出了这些类型,对其作了简单描述,并说明了要求的存储空间。

表 3-8　字符数据类型简表

数 据 类 型	描　　述	存 储 空 间
Char(n)	n 为 1～8000 字符之间	n 字节
Nchar(n)	n 为 1～4000 Unicode 字符之间	(2n 字节)＋2 字节额外开销
Ntext	最多为 $2^{30}-1$(1 073 741 823)Unicode 字符	每字符 2 字节
Nvarchar(max)	最多为 $2^{30}-1$(1 073 741 823)Unicode 字符	2×字符数＋2 字节额外开销
Text	最多为 $2^{31}-1$(2 147 483 647)字符	每字符 1 字节
Varchar(n)	n 为 1～8000 字符之间	每字符 1 字节＋2 字节额外开销
Varchar(max)	最多为 $2^{31}-1$(2 147 483 647)字符	每字符 1 字节＋2 字节额外开销

2. 精确数值数据类型

数值数据类型包括 bit、tinyint、smallint、int、bigint、numeric、decimal、money、float 以及 real。这些数据类型都用于存储不同类型的数字值，如表 3-9 所示。

表 3-9　精确数值数据类型简表

数 据 类 型	描　　述	存 储 空 间
bit	0、1 或 Null	1 字节(8 位)
tinyint	0～255 之间的整数	1 字节
smallint	－32 768～32 767 之间的整数	2 字节
int	－2 147 483 648～2 147 483 647 之间的整数	4 字节
bigint	－9 223 372 036 854 775 808～9 223 372 036 854 775 807 之间的整数	8 字节
numeric(p,s)或 decimal(p,s)	－1 038＋1～1 038－1 之间的数值	最多 17 字节
money	－922 337 203 685 477.580 8～922 337 203 685 477.580 7	8 字节
smallmoney	－214 748.3648～2 14 748.3647	4 字节

Decimal 和 numeric 等数值数据类型可存储小数点右边或左边的变长位数。p(precision)为总位数，s(scale)是小数点右边的位数。

3. 近似数值数据类型

这个数据类型分类包括 float 和 real，用于表示浮点数据。float(n)中的 n 是用于存储该数科学记数法尾数(mantissa)的位数。SQL Server 对此只使用两个值。如果指定位于 1～24 之间，SQL 就使用 24。如果指定 25～53 之间，SQL 就使用 53。当指定 float() 时(括号中为空)，默认为 53。

表 3-10 列出了近似数值数据类型，对其进行简单描述，并说明了要求的存储空间。

表 3-10　近似数值数据类型简表

数据类型	描　　述	存储空间
float[(n)]	−1.79E+308～−2.23E−308,0,2.23E−308～1.79E+308	8 字节
real()	−3.40E+38～−1.18E−38,0,1.18E−38～3.40E+38	4 字节

小提示

real 的同义词为 float(24)。

4. 二进制数据类型

如 varbinary、binary、varbinary(max)或 image 等二进制数据类型用于存储二进制数据,如图形文件、Word 文档或 MP3 文件。image 数据类型的首选替代数据类型是 varbinary(max),可保存最多 8KB 的二进制数据,其性能通常比 image 数据类型好。

SQL Server 2012 的新功能是可以在操作系统文件中通过 FileStream 存储选项存储 varbinary(max)对象。这个选项将数据存储为文件,同时不受 varbinary(max)的 2GB 大小的限制。表 3-11 列出了二进制数据类型,对其作了简单描述,并说明了要求的存储空间。

表 3-11　二进制数据类型简表

数据类型	描　　述	存储空间
Binary(n)	n 为 1～8000 之间十六进制数字	n 字节
Image	最多为 $2^{31}-1$(2 147 483 647)十六进制数位	每字符 1 字节
Varbinary(n)	n 为 1～8000 之间十六进制数字	每字符 1 字节+2 字节额外开销
Varbinary(max)	最多为 $2^{31}-1$(2 147 483 647)十六进制数字	每字符 1 字节+2 字节额外开销

5. 日期和时间数据类型

表 3-12 列出了日期/时间数据类型,对其进行简单描述,并说明了要求的存储空间。

表 3-12　日期和时间数据类型简表

数据类型	描　　述	存储空间
Date	0001 年 1 月 1 日～9999 年 12 月 31 日	3 字节
Datetime	1753 年 1 月 1 日～9999 年 12 月 31 日,精确到最近的 3.33 毫秒	8 字节
Datetime2(n)	0001 年 1 月 1 日～9999 年 12 月 31 日,0～7 之间的 N 指定小数秒	6～8 字节
Datetimeoffset(n)	9999 年 1 月 1 日～12 月 31 日 0～7 之间的 N 指定小数秒+/−偏移量	8～10 字节
SmalldateTime	1900 年 1 月 1 日～2079 年 6 月 6 日,精确到 1 分钟	4 字节
Time(n)	小时:分钟:秒.99999990～7 之间的 N 指定小数秒	3～5 字节

3.1.3　使用 T-SQL 语句创建表

在 SQL Server 中，创建表主要有两种方法，除了在“实例 3”中使用对象资源管理器，还可以使用 T-SQL 语句创建表。

在 SQL Server 中，除了可以使用对象资源管理器来创建表外，还可以使用 T-SQL 语言中的 CREATE TABLE 命令来创建表，其语法格式如下：

```
CREATE TABLE [ database_name . [ schema_name ] . | schema_name . ] table_name
(  {  <column_definition>            /* 列的定义 */
      | column_name AS computed_column_expression [PERSISTED [NOT NULL]]
                                                          /* 定义计算列 */
   }
   [ <table_constraint>] [ ,...,n ]    /* 指定表的约束 */
)
[ ON { partition_scheme_name ( partition_column_name ) | filegroup | "default" }
]
                                     /* 指定分区方案和存储表的文件组 */
[ { TEXTIMAGE_ON { filegroup | "default" } ]
                                    /* 指定存储 text、image 等类型数据的文件组 */
[ FILESTREAM_ON { partition_scheme_name | filegroup | "default" } ]
                                     /* 指定存储 FILESTREAM 数据的文件组 */
[ WITH ( <table_option>[ ,...,n ] ) ]   /* 指定表选项 */
[ ; ]
```

说明：

(1) database_name 是数据库名，schema_name 是新表所属架构的名称，table_name 是表名，表的标识按照对象命名规则。如果省略数据库名则默认在当前数据库中创建表，如果省略架构名，则默认是 dbo。

(2) 列的定义格式如下：

```
<column_definition>::=
column_name data_type                          /* 指定列名、类型 */
    [ FILESTREAM ]                             /* 指定 FILESTREAM 属性 */
    [ COLLATE collation_name ]                 /* 指定排序规则 */
    [ NULL | NOT NULL ]                        /* 指定是否为空 */
    [ [ CONSTRAINT constraint_name ]
       DEFAULT constant_expression ]           /* 指定默认值 */
    | [ IDENTITY [ ( seed ,increment ) ] [ NOT FOR REPLICATION ]
                                               /* 指定列为标识列 */
    ]
    [ ROWGUIDCOL ]                             /* 指定列为全局标识符列 */
    [ <column_constraint>[ ,...,n ] ]          /* 指定列的约束 */
    [ SPARSE ]
```

① FILESTREAM：FILESTREAM 是 SQL Server 2012 引进的一项新特性，允许以独立文件的形式存放大对象数据，而不是像以往一样将所有数据都保存到数据文件中。FILESTREAM 存储以 varbinary(MAX)列的形式实现，在该列中数据以 BLOB 的形式

存储在文件系统中，大小仅受文件系统容量大小的限制。若要将指定列使用FILESTREAM存储在文件系统中，可以对 varbinary（MAX）数据类型的列指定FILESTREAM属性。这样数据库引擎会将该列的所有数据存储在文件系统，而不是数据库文件中。FILESTREAM数据必须存储在FILESTREAM文件组中。完成以上步骤后数据库实例即启用了FILESTREAM，接下来就可以创建FILESTREAM文件组和具有FILESTREAM数据列的表。在创建了FILESTREAM数据列后，访问的方法与访问一般的varbinary(MAX)列的方式相同。

② IDENTITY：指出该列为标识符列，为该列提供一个唯一的、递增的值。seed是标识字段的起始值，默认值为1，increment是标识增量，默认值为1。如果为IDENTITY属性指定了NOT FOR REPLICATION选项，则复制代理执行插入时，标识列中的值将不会增加。

③ ROWGUIDCOL：表示新列是行的全局唯一标识符列，ROWGUIDCOL属性只能指派给uniqueidentifier列。该属性并不强制列中所存储值的唯一性，也不会为插入到表中的新行自动生成值。

④ SPARSE：指定列为稀疏列。稀疏列是对NULL值采用优化的存储方式的普通列。稀疏列减少了NULL值的空间需求，但代价是检索非NULL值的开销增加。不能将稀疏列指定为NOT NULL。

⑤ ＜column_constraint＞：列的完整性约束，指定主键、替代键、外键等。如指定该列为主键使用PRIMARY KEY关键字。

（3） column _ name AS computed _ column _ expression [PERSISTED [NOT NULL]]：用于定义计算字段，计算字段是由同一表中的其他字段通过表达式计算得到。其中，column_name为计算字段的列名，computed_column_expression是表其他字段的表达式，表达式可以是非计算字段的字段名、常量、函数、变量，也可以是一个或多个运算符连接的上述元素的任意组合。系统不将计算列中的数据进行物理存储，该列只是一个虚拟列。如果需要将该列的数据物理化，需要使用PERSISTED关键字。

相关案例 1

在“图书管理”数据库中创建一个学生信息表(reader表)，表中各字段属性如表3-13所示。

表 3-13　学生信息表(reader)

编号	字段名称	字段说明	类　型	是否允许空	长　度
1	reader_id	学生编号	int	否	
2	reader_name	学生姓名	varchar	否	10
3	sex	性别	char	否	2
4	birthday	出生日期	datetime	否	
5	phone	电话	varchar	是	15
6	department	部门或单位	varchar	是	100

步骤如下。

(1) 在查询分析器的编辑窗口中输入如下语句：

```
USE 图书管理          /* 将数据库“图书管理”指定为当前数据库 */
GO
CREATE TABLE reader
(  reader_id int NOT NULL PRIMARY KEY,
   reader_name varchar(10) NOT NULL,
   sex char(2) NOT NULL,
   birthday date NOT NULL,
   phone varchar(15) NULL,
   department varchar(100) NULL,
)
```

(2) 单击工具栏中的“执行”按钮，或按 F5 键，就会在“图书管理”数据库中生成 reader 表了。

相关案例 2

在教学管理数据库中创建一个借书信息表(borrow 表)，表中各字段属性如表 3-14 所示。

表 3-14　借书信息表（borrow）

编号	字 段 名 称	字 段 说 明	类　　型	是否允许空	长　度
1	borrow_id	借书号	int	否	
2	reader_id	学生编号	int	否	
3	student_id	学生编号	varchar	否	20
4	borrow_time	借阅时间	datetime	否	
5	borrow_number	借书数量	smallint	否	
6	due_time	到期时间	datetime	否	
7	return_time	返还时间	datetime	是	
8	fine	罚款	decimal	是	18,2

步骤如下。

(1) 在查询分析器的编辑窗口中输入如下语句：

```
USE 图书管理
GO
CREATE TABLE borrow
(  borrow_id int NOT NULL PRIMARY KEY,
   reader_id int NOT NULL,
   student_id varchar(20) NOT NULL,
   borrow_time datetime NOT NULL,
   borrow_number smallint NOT NULL,
```