

第 3 章

管理数据表

内容摘要

在数据库中表是一个非常重要的、用来组织和存储数据的数据库对象,对数据库的各类操作实际上是对表的操作。本章主要介绍表的概念、SQL Server 2005 数据库中的数据类型,学习掌握创建表、管理表的方法。

本章学习任务

- (1) 掌握表的概念。
- (2) 掌握 SQL Server 中的数据类型。
- (3) 掌握创建表的方法。
- (4) 掌握管理表的方法。

3.1 表 的 概 念

关系型数据库中的表是由行和列组成的二维表格结构,每个表代表一种实体,每一行代表实体中的一个记录,每一列代表实体的一个属性。如表 3-1 所示,每一行代表一条记录,表示每一名学生(实体),而每列则代表记录中的一个域(属性),分别表示学生的详细资料,如学号、姓名、出生日期等。

表 3-1 关系型表结构

学号	姓名	性别	出生日期	入学时间
0891B001	王晓红	女	1988/12/25	2008/9/2
0891B002	李刚	男	1988/5/8	2008/9/2

3.2 SQL Server 中的数据类型

在创建表时,要确定表中有哪些列(或字段)及定义每个列所对应的数据类型,数据类型能够指定某列信息能够保存的类型,如数值、日期等,以及数据在计算机中的存储格式、存储长度等属性。SQL Server 2005 提供了多种数据类型,常用的数据类型如下。

1. 字符数据类型

字符数据类型用于存储汉字、数字、英文字母、标点及各类符号,输入时用英文的单引号将其括起,包括 3 种类型,分别是 char, varchar 和 text。

- char(n)用于存储定长的字符串, n 的取值范围是 $1\sim 8000$,若输入的数据信息字节长度不足 n 个字节,则补足存储,若输入的数据信息字节长度超过 n 个字节,则截断存储。
- varchar(n)用于存储变长的字符串, n 的取值范围是 $1\sim 8000$,若输入的数据信息字节长度不足 n 个字节,则按数据信息的实际长度存储,若输入的数据信息字节长度超过 n 个字节,则截断后存储。
- text 可以存储 $2^{31}-1$ 个字节的数据。

2. Unicode 字符数据类型

Unicode 字符数据类型采用两个字节编码每个字符,当需要的字符不能全部存储在单字节字符数据内时,就需要使用 Unicode 字符数据,但是由于采用两个字节编码一个字符,所以最大存储范围是 4000 个字符。包括 3 种类型,分别是 nchar, nvarchar 和 ntext。用法分别与字符数据类型的用法相对应。

- nchar(n)用于存储定长的 Unicode 字符数据, n 的取值范围是 $1\sim 4000$ 。
- nvarchar(n)用于存储变长的 Unicode 字符数据, n 的取值范围是 $1\sim 4000$ 。
- ntext 用于存储 $2^{30}-1$ 个字节的 Unicode 字符数据。

3. 日期时间数据类型

日期时间数据类型用于存储日期和时间的数据,有两种类型,分别是 datetime 和 smalldatetime。

- datetime 用于存储 1753 年 1 月 1 日到 9999 年 12 月 31 日的日期和时间数据,精确度为百分之三秒。
- smalldatetime 用于存储 1900 年 1 月 1 日到 2079 年 6 月 6 日的日期和时间数据,精确度为分钟。

如果没有提供日期,则将 1900 年 1 月 1 日作为系统默认值;如果没有提供时间,则将 00:00:00.000 设为默认值。在输入日期时间型数据时,字符串中可以使用斜杠(/)、连字符(-)、句点(.)作为分隔符将月、日、年分开。

4. 整型

整型用于存储整数,可以分为 4 种类型,分别为 bigint, int, smallint 和 tinyint。

- tinyint 存储范围 $0\sim 255$ 之间的整数,所占存储空间为 1 个字节。
- smallint 存储范围是 $-2^{15}\sim 2^{15}-1$ 之间的整数,所占存储空间为 2 个字节。
- int 存储范围是 $-2^{31}\sim 2^{31}-1$ 之间的整数,所占存储空间为 4 个字节。
- bigint 和 int 相同,存储范围是 $-2^{31}\sim 2^{31}-1$ 之间的整数,所占存储空间为 4 个字节。

5. 精确数值类型

精确数值类型要求准确地指定小数位和精度。主要包括两种数据类型 decimal 和 numeric。

- Decimal[(p[,s])]: 使用最大精度时,有效值范围是 $-10^{38}+1 \sim 10^{38}-1$ 。
- numeric[(p[,s])]: 用法与 Decimal[(p[,s])]相同。

p 是从 1 到最大精度之间取值,最大精度为 38,s 是可以存储小数的最大位数,s 的取值范围是 0 到 p,默认小数位数是 0。

6. 近似(浮点)数值类型

近似数值类型用于存储浮点数,包括 float 和 real 两种数据类型。

- float(n): 用于存储 $-1.79E+308 \sim 1.79E+308$ 之间的数据,其中 n 是精度,n 的取值范围是 1~53 之间的整数。
- real: 用于存储 $-3.40E+38 \sim 3.40E+38$ 之间的浮点数据,所占存储空间为 4 个字节。

注意: 在 where 语句中尽量不用数据类型是 float 或 real 的列。

7. 货币类型

货币类型有两种,分别是 money 和 smallmoney。

- Money: 存储 $-2^{63} \sim 2^{63}-1$ 之间的数据,精确度为货币单位的千分之十,所占存储空间为 8 字节。
- Smallmoney: 存储 $-214748.3648 \sim +214748.3647$ 之间的数据,精确度为货币单位的千分之十,所占存储空间为 4 个字节。

8. 二进制数据类型

二进制数据类型用于存储图像等数据,包括 3 种类型,分别是 binary, varbinary 和 image。

- Binary[(n)]用于存储定长的二进制数据,n 的取值范围是 1~8000,所占存储空间为 n+4 个字节,如果输入的数据信息不足 n+4 个字节,则补足存储。若输入的数据信息超过 n+4 个字节,则截断后存储。
- varBinary[(n)] 用于存储变长的二进制数据,n 的取值范围是 1~8000,若输入的数据信息不足 n+4 个字节则按实际长度存储,若输入的数据信息超过 n+4 个字节则截断后存储。
- image 存储范围是不超过 $2^{31}-1$ 个字节的二进制数据。

9. bit 型

Bit 数据类型只能存储 0 或 1 或 null,如果输入了除 0 和 1 以外的任何数据,系统都会将其作为 1 来处理,存储所占空间为 1 比特。但是如果一个表中仅有一个 bit 类型的列,那么也占用整个字节,一个字节最多可以存储 8 个 bit 列。

3.3 创建数据表

前面我们曾经创建了一个名为 student 的学生信息管理数据库,下面将在此数据库下创建表。创建表有两种方法,可以通过 SSMS 中的对象资源管理器以图形化的方式创建,或者使用 T-SQL 语句创建。无论使用哪种方法,都要求用户具有创建表的权限,默认状态下,系统管理员和数据库的所有者具有创建表的权限。

在同一个表中,列(字段)名具有唯一性,每个列只能有一种数据类型。各列还可以包含其他属性,如为空性、默认值和标识属性等。

1. 允许空(NULL)

在设计数据表时,可以把表的某一列指定为 NULL 属性,这表示向表中插入数据时允许省略该列的值。如果设置列为空,决定该列在表中是允许有空值的。在 SQL Server 2005 中列的默认属性是允许为空。

提示

注意,空值 NULL 并不等于 0、空白字符或零长度字符串“ ”。

2. 默认值(DEFAULT)

当向数据表中插入数据时,如果定义了字段的默认值属性,则如果该字段未被插入数据时,则该字段取默认值。

提示

列的 NULL、DEFAULT 属性、插入新的一行该列的实际值的对应关系如表 3-2 所示。

表 3-2 对应关系表

列的属性	无输入、定义 DEFAULT 属性	无输入、没定义 DEFAULT 属性	输入空值
允许空值 NULL	默认值	NULL	NULL
不允许空值 NOT NULL	默认值	提示错误	提示错误

3. 标识属性(identity)

将列的属性设置为标识属性,可以使表在该列自动生成数字,其在表中唯一标识一行。设置标识属性时,同时可以指定一个初始值和增量(默认都为 1)。SQL Server 2005 默认状态下对标识列的设置为隐式插入标识列的值,即当插入新记录到含有标识列的数据表时,用户无须提供标识列的值,系统将为标识列自动生成值。

对标识列的隐式或显式插入值的语法设置为：

```
SET IDENTITY_INSERT 表 ON|OFF
```

其中：

ON：为允许显式插入标识列值，即由用户提供标识列的列表和标识列值。

OFF：为允许显式插入标识列值，即用户不需要提供标识列的列表和标识列的值，系统将自动产生标识列值。

3.3.1 使用 SSMS 对象资源管理器创建表

创建表一般要经过定义表结构、设置约束和添加数据 3 个步骤，其中设置约束可以在定义表结构时建立，也可以在定义完成后再建立。

定义表结构：给表的每一列取字段名，并确定每一列的数据类型、数据长度、列数据是否允许为空等。字段名可以是汉字或由符合标识符定义的字符构成。

设置约束：设置约束是为了限制该列输入值的取值范围，以保证输入数据的正确性和一致性。

添加数据：表结构建立完成之后，就可以向表中输入数据了。

下面在学生管理数据库 student 中使用 SSMS 对象资源管理器创建学生基本信息表，学生基本信息表结构如表 3-3 所示。

表 3-3 学生基本信息表结构

字段名	类型	大小	约束	备注
学号	Char	8	主键	
姓名	Char	10	不允许为空	
性别	Char	2	默认为‘男’	
出生年月	datetime			
籍贯	varchar	20		
家庭住址	varchar	60		
电话	Char	15		
E_MAIL	varchar	50		
所属班级	char	8		

实现步骤如下。

(1) 在对象资源管理器中，右击 student 数据库的“表”节点，再单击“新建表”，如图 3-1 所示。

(2) 在出现的表设计窗口中，输入列名，选择数据类型，并选择各列是否允许为空值，如图 3-2 所示。

(3) 在“文件”菜单中，选择“保存表名”，或直接单击工具栏中的“保存”按钮进行保存。

(4) 在“选择名称”对话框中，为该表输入名称“学生基本信息表”，再单击“确定”按钮。



图 3-1 新建表快捷菜单

图 3-2 学生基本信息表设计窗口

3.3.2 使用 T-SQL 语句创建表结构

可以使用 T-SQL 语句中的 CREATE TABLE 语句创建表结构,其语法格式如下:

```
CREATE TABLE [database_name].[owner].[owner.]table_name
( {column_definition}
  | [ { PRIMARY KEY | UNIQUE }[, ... N] ]
)

< column_definition>::= { column_name data_type}
                        [ [ DEFAULT constraint_expression]
                          | [ IDENTITY [(seed,increment) ] ]
                        ]
                        [<column_constraint>][, ... n]
```

参数说明:

- database_name: 指定创建的表所在的数据库,在当前数据库创建表时该项可以省略。
- owner: 指定表的拥有者,如果表的拥有者为当前用户,该项可以省略。
- table_name: 指定新建表的名称。
- column_definition: 为表中字段的定义表达式。
- column_name: 为表中的字段名。
- data_type: 为字段的数据类型。
- PRIMARY KEY: 为主键约束关键字。
- UNIQUE: 为唯一约束关键字。

- DEFAULT: 为默认约束关键字。
- IDENTITY: 为自动编号标识。
- Seed: 为自动标识的开始值,默认为 1。
- Increment: 为自动编号的步长或增量,默认为 1。

提示

注意,SQL 语句不区分大小写,但为了清晰,一般都用大写表示系统保留字,用小写表示用户自定义的名称。一条语句可以写在多行上,但不能多条语句写在一行上。



实践案例

【示范案例 1】

使用 T-SQL 语句在学生管理数据库中创建学生基本信息表 stud_info,学生基本信息表结构如表 3-3 所示。

实现的 T-SQL 语句:

```
USE student
GO
CREATE TABLE stud_info
(
    学号 char(8) PRIMARY KEY,
    姓名 char(10) NOT NULL,
    性别 char(2) DEFAULT '男',
    出生年月 datetime,
    籍贯 varchar(20),
    家庭住址 varchar(60),
    电话 char(15),
    E_MAIL varchar(50),
    所属班级 char(8)
)
GO
```

【同步任务 1】

在学生管理数据库中用 T-SQL 语句创建班级信息表 class,班级表结构如表 3-4 所示。

表 3-4 班级表 class 的表结构

字段描述	字段名	类型	大小	约束	备 注
班级代码	Class_id	Char	8	主键	
班级名称	Class_name	varchar	20	不允许为空	
系部代码	Depart_id	Char	2		
备注	memory	varchar	50		

【示范案例 2】

在学生管理数据库中用 T-SQL 语句创建一个学生成绩表 stud_score,表结构如表 3-5

所示,其中包含标识列和计算列。

表 3-5 学生成绩表 stud_score 的表结构

字段描述	字段名	类型	大小	约束	备 注
序号	Scoreid	int		主键	标识列(1,1)
学号	Stud_no	char	8	非空	
数学	Math	int		默认为 0	
语文	Chinese	int		默认为 0	
计算机	Computer	int		默认为 0	
总分	total_score	int			数学+语文+计算机

实现的 T-SQL 语句:

```
CREATE TABLE stud_score
(
    Scoreid int IDENTITY(1,1) PRIMARY KEY,
    Stud_no char(8) NOT NULL,
    Math int DEFAULT 0,
    Chinese int DEFAULT 0,
    Computer int DEFAULT 0,
    total_score AS Math + Chinese + computer
)
GO
```

提示

设置列的标识属性时,应注意以下几项。

- 该列的数据类型必须是 decimal, int, numeric, smallint, bigint, tinyint 中的一种,才可以设置标识属性。
- 标识列不允许为空值,也不能包含默认属性。
- 每个表中只允许有一个标识列,并且不可以修改。

提示

设置计算列时,应注意以下几项。

- 在计算列中不能添加如 primary key、unique、foreign key、default 等约束条件。
- 不能对计算列进行赋值。

【同步任务 2】

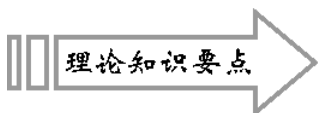
分别使用 SSMS 对象资源管理器和 T-SQL 语句在产品数据库中创建一个订货表 orders,表结构如表 3-6 所示。

表 3-6 订货表 orders 的表结构

字段描述	字段名	类型	大小	约束	备 注
订货序号	Order_id	int		主键	标识列(1,5)
产品编号	Prod_id	Char	8	非空	

续表					
字段描述	字段名	类型	大小	约束	备 注
单价	unitprice	money		默认为 0	
订货数量	quantity	int		允许空	
总金额	Total_price			允许空	单价×订货数量
备注	memory	varchar	100	允许空	

3.4 管理数据表



当表建好以后,在使用过程中可能会根据需要对表的结构进行更改,比如添加字段、删除字段、修改字段及表名或者对列的属性进行更改等,现在以更改学生基本信息表为例讲解更改表结构的方法。

3.4.1 修改表结构

1. 添加字段

1) 使用 SSMS 对象资源管理器向表中添加字段

例如,使用 SSMS 对象资源管理器修改学生基本信息表结构,增加字段“QQ 号码”,数据类型为 varchar,长度为 15,可为空;增加“入学时间”字段,时间/日期类型,默认时间为 2006-9-1。其实现步骤如下。

① 在对象资源管理器中,右击 student 数据库的“学生基本信息表”,从弹出的快捷菜单中选择“修改”命令,如图 3-3 所示。

② 在表的设计窗口中增加字段“QQ 号码”,varchar(15),可为空;增加“入学时间”字段,datetime 类型,默认时间为 2006-9-1,如图 3-4 所示。

③ 单击工具栏中的“保存”按钮即可。

2) 使用 T-SQL 语句向表中添加字段

使用 ALTER TABLE 语句可以向表中增加字段,其语法格式为:

```
ALTER TABLE table_name
ADD column_definition
```

参数说明:

- table_name: 要修改的表的名称。
- column_definition: 希望增加的列的定义表达式,即字段名和数据类型等。

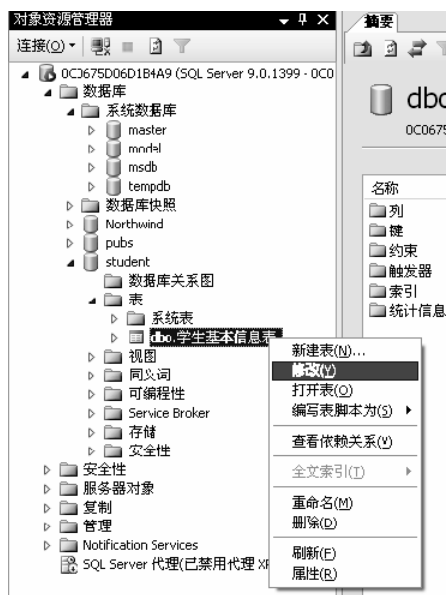


图 3-3 修改表的快捷菜单



图 3-4 学生基本信息表修改窗口

2. 修改字段

1) 使用 SSMS 对象资源管理器在表中修改字段

例如,使用 SSMS 对象资源管理器修改学生基本信息表结构,修改字段“家庭住址”的长度为 100。其实现步骤如下。

① 在对象资源管理器中,右击数据库的学生基本信息,再单击“修改”按钮,可打开表设计器窗口。

② 在表设计器中修改字段“家庭住址”,将其长度更改为 100。

③ 单击工具栏中的“保存”按钮即可。

2) 使用 T-SQL 语句在表中修改字段

使用 ALTER TABLE 语句可以修改表中的字段,其语法格式为:

```
ALTER TABLE table_name
ALTER COLUMN column_definition
```

参数说明:

- table_name: 要修改的表的名称。
- column_definition: 要修改的列的定义表达式,即字段名和数据类型等。

3. 删除字段

1) 使用 SSMS 对象资源管理器在表中删除字段

例如,用 SSMS 对象资源管理器删除学生基本信息表的 QQ 号码字段。其实现步骤如下。

① 在对象资源管理器中,右击数据库的学生基本信息,再单击“修改”按钮,可打开表设

计器窗口。

② 在表设计器中单击字段“QQ 号码”，然后右击，从快捷菜单中选择“删除列”命令，如图 3-5 所示。

③ 单击工具栏中的“保存”按钮即可。

2) 使用 T-SQL 语句在表中删除字段

使用 ALTER TABLE 语句可以删除表中的字段，其语法格式为：

```
ALTER TABLE table_name
DROP COLUMN column_name
```

参数说明：

- table_name：要修改的表的名称。
- column_name：要删除的字段名。

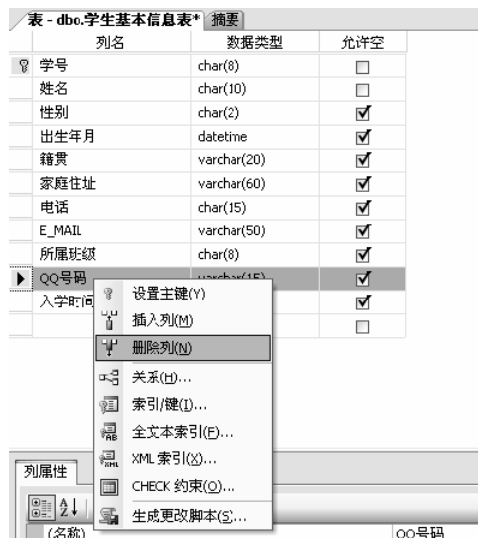


图 3-5 删除表中某列的快捷菜单

3.4.2 删除数据库中的表

对于不再需要的表，可以将其删除。一旦表被删除，则表的结构、表中的数据、约束、索引等数据库对象都将永久地被删除。删除表的操作可以通过图形化的 SSMS 管理器完成，也可以通过执行 DROP TABLE 语句完成。

1. 使用 SSMS 对象资源管理器删除表

例如，使用 SSMS 管理器删除学生基本信息表，其实现步骤如下。

(1) 在对象资源管理器中，右击数据库的“学生基本信息”，再单击“删除”按钮，可打开删除对象窗口。

(2) 在此窗口中可查看欲删除对象的依赖关系，如果确认要删除此表，单击“确定”按钮即可。

2. 使用 T-SQL 语句删除表

使用 DROP TABLE 语句可以删除数据库中的表，其语法格式为：

```
DROP TABLE table_name
```

参数说明：

table_name：要删除的表的名称。

3.4.3 更改表名

1. 使用 SSMS 对象资源管理器更改表名

例如，使用 SSMS 对象资源管理器更改学生基本信息表的表名，将其更新成为“学生信