第3章 创建数据库和创建表

本章要点：

- SQL Server 数据库的基本概念
- 以图形界面方式创建 SQL Server 数据库
- 以命令方式创建 SQL Server 数据库
- 表的基本概念
- 以命令方式创建 SQL Server 表
- 使用图形用户界面创建 SQL Server 表

使用 SQL Server 设计和实现信息系统，首先就要设计和实现数据的表示和存储，即创建数据库，数据库是 SQL Server 用于组织和管理数据的基本对象，SQL Server 能够支持多个数据库。表是最重要的数据库对象，它是由行和列构成的集合，用来存储数据。本章介绍 SQL Server 数据库的基本概念、以图形界面方式创建 SQL Server 数据库、以命令方式创建 SQL Server 数据库、表的基本概念、数据类型、以命令方式创建 SQL Server 表、以图形界面方式创建 SQL Server 表等内容。

3.1 SQL Server 数据库的基本概念

数据库是 SQL Server 存储和管理数据的基本对象，下面从逻辑数据库和物理数据库两个角度进行讨论。

3.1.1 逻辑数据库

从用户的观点看，组成数据库的逻辑成分称为数据库对象，SQL Server 数据库由存放数据的表以及支持这些数据的存储、检索、安全性和完整性的对象所组成。

1. 数据库对象

SQL Server 的数据库对象包括表(table)、视图(view)、索引(index)、存储过程(stored procedure)、触发器(trigger)等，下面介绍常用的数据库对象。

- 表：表是包含数据库中所有数据的数据库对象，由行和列构成，它是最重要的数据库对象。
- 视图：视图是由一个表或多个表导出的表，又称为虚拟表。
- 索引：为加快数据检索速度并可以保证数据唯一性的数据结构。
- 存储过程：为完成特定功能的 T-SQL 语句集合，编译后存放于服务器端的数据库中。

- 触发器：它是一种特殊的存储过程，当某个规定的事件发生时，该存储过程自动执行。

2. 系统数据库和用户数据库

SQL Server 的数据库有两类：一类是系统数据库；另一类是用户数据库。

(1) 系统数据库

SQL Server 在安装时创建 4 个系统数据库：master、model、msdb 和 tempdb。系统数据库存储有关 SQL Server 的系统信息，当系统数据库受到破坏时，SQL Server 将不能正常启动和工作。

- master 数据库：它是系统最重要的数据库，记录了 SQL Server 的系统信息，例如登录账号、系统配置、数据库位置及数据库错误信息等，用于控制用户数据库和 SQL Server 的运行。
- model 数据库：为创建数据库提供模板。
- msdb 数据库：该数据库是代理服务数据库，为调度信息、作业记录等提供存储空间。
- tempdb 数据库：它是一个临时数据库，为临时表和临时存储过程提供存储空间。

(2) 用户数据库

用户数据库是由用户创建的数据库，本书所创建的数据库都是用户数据库，用户数据库和系统数据库在结构上是相同的。

3. 完全限定名和部分限定名

在 T-SQL 中引用 SQL Server 对象对其进行查询、插入、修改、删除等操作，所使用的 T-SQL 语句需要给出对象的名称，用户可以使用完全限定名和部分限定名。

(1) 完全限定名

完全限定名是对象的全名，SQL Server 创建的每个对象都有唯一的完全限定名，它由 4 部分组成：服务器名、数据库名、数据库架构名和对象名，其格式如下：

```
server.database.scheme.object
```

例如，DELL-PC.stsco.dbo.student 即为一个完全限定名。

(2) 部分限定名

使用完全限定名往往很烦琐且没有必要，经常省略其中的某些部分。在对象全名的 4 部分中，前 3 部分均可被省略，当省略中间的部分时，圆点符“.”不可省略。这种只包含对象完全限定名中的一部分的对象名称为部分限定名。

在部分限定名中，未指出的部分使用以下默认值。

服务器：默认为本地服务器。

数据库：默认为当前数据库。

数据库架构名：默认为 dbo。

部分限定名格式如下：

```
server.database...object          /* 省略架构名 */
server..scheme.object             /* 省略数据库名 */
database.scheme.object            /* 省略服务器名 */
server...object                   /* 省略架构名和数据库名 */
```

```
scheme.object                    /* 省略服务器名和数据库名 */
object                           /* 省略服务器名、数据库名和架构名 */
```

例如，完全限定名 DELL-PC.stsco.dbo.student 的部分限定名如下：

```
DELL-PC.stsco..student
DELL-PC..dbo.student
stsco.dbo.student
DELL-PC..student
dbo.student
student
```

3.1.2 物理数据库

从系统的观点看，数据库是存储逻辑数据库的各种对象的实体，它们存放在计算机的存储介质中，从这个角度我们称数据库为物理数据库。SQL Server 的物理数据库架构包括页和区、数据库文件、数据库文件组等。

1. 页和区

页和区是 SQL Server 数据库的两个主要数据存储单位。

- 页：每页的大小是 8KB，1MB 的数据文件可以容纳 128 页，页是 SQL Server 中用于数据存储的最基本单位。
- 区：每连接的 8 页组成一个区，区的大小是 64KB，1MB 的数据库有 16 个区，区用于控制表和索引的存储。

2. 数据库文件

SQL Server 采用操作系统文件来存放数据库，使用的文件有主数据文件、辅助数据文件、日志文件三类。

(1) 主数据文件

主数据文件(Primary)用于存储数据，每个数据库必须有也只能有一个主文件，它的默认扩展名为.mdf。

(2) 辅助数据文件

辅助数据文件(Secondary)也用于存储数据，一个数据库中辅助数据文件可以创建多个，也可以没有，辅助数据文件的默认扩展名为.ndf。

(3) 日志文件

日志文件(Transaction Log)用于保存恢复数据库所需的事务日志信息。每个数据库至少有一个日志文件，也可以有多个，日志文件的扩展名为.ldf。

3. 数据库文件组

为了管理和分配数据将多个文件组织在一起，组成文件组，对它们进行整体管理，以提高表中数据的查询效率，SQL Server 提供了两类文件组：主文件组和用户定义文件组。

(1) 主文件组

主文件组包含主要数据文件和任何没有指派给其他文件组的文件，数据库的系统表均分配在主文件组中。

(2) 用户定义文件组

用户定义文件组包含所有使用 CREATE DATABASE 或 ALTER DATABASE 语句并用 FILEGROUP 关键字指定的文件组。

3.2 以图形界面方式创建 SQL Server 数据库

SQL Server 提供两种方式创建 SQL Server 数据库,一种方式是使用 SQL Server Management Studio 的图形用户界面创建 SQL Server 数据库;另一种方式是使用 T-SQL 语句创建 SQL Server 数据库。本节介绍以图形界面方式创建 SQL Server 数据库。

以图形界面方式创建 SQL Server 数据库包括创建数据库、修改数据库、删除数据库等内容,下面分别介绍。

1. 创建数据库

在使用数据库之前,首先需要创建数据库。在学生成绩管理系统中,我们以创建名称为 stsco 的学生成绩数据库为例,说明创建数据库的步骤。

【例 3.1】 使用 SQL Server Management Studio 创建 stsco 数据库。

创建 stsco 数据库的操作步骤如下。

(1) 单击"开始"按钮,选择 Microsoft SQL Server Tools 18→SQL Server Management Studio 18 命令,出现"连接到服务器"对话框,在"服务器名称"框中选择 DESKTOP-7O2OTMS,在"身份验证"框中选择"SQL Server 身份验证",在"登录名"框中选择 sa,在"密码"框中输入 123456,单击"连接"按钮,即可启动 SQL Server Management Studio,并连接到 SQL Server 服务器。

(2) 屏幕出现 SQL Server Management Studio 窗口,在左边"对象资源管理器"窗口中选中"数据库"节点,右击,在弹出的快捷菜单中选择"新建数据库"命令,如图 3.1 所示。

图 3.1 选择"新建数据库"命令

(3) 进入“新建数据库”窗口,在窗口的左上方有 3 个选项:“常规”选项、“选项”选项和“文件组”选项,其中“常规”选项首先出现。

在“数据库名称”文本框中输入创建的数据库名称 stsco,在“所有者”文本框使用系统默认值,系统自动在“数据库文件”列表中生成一个主数据文件 stsco.mdf 和一个日志文件 stsco_log.ldf,主数据文件 stsco.mdf 初始大小为 8MB,增量为 64MB,存放的路径为 C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\,日志文件 stsco_log.ldf 初始大小为 8MB,增量为 64MB,存放的路径与主数据文件的路径相同,如图 3.2 所示。

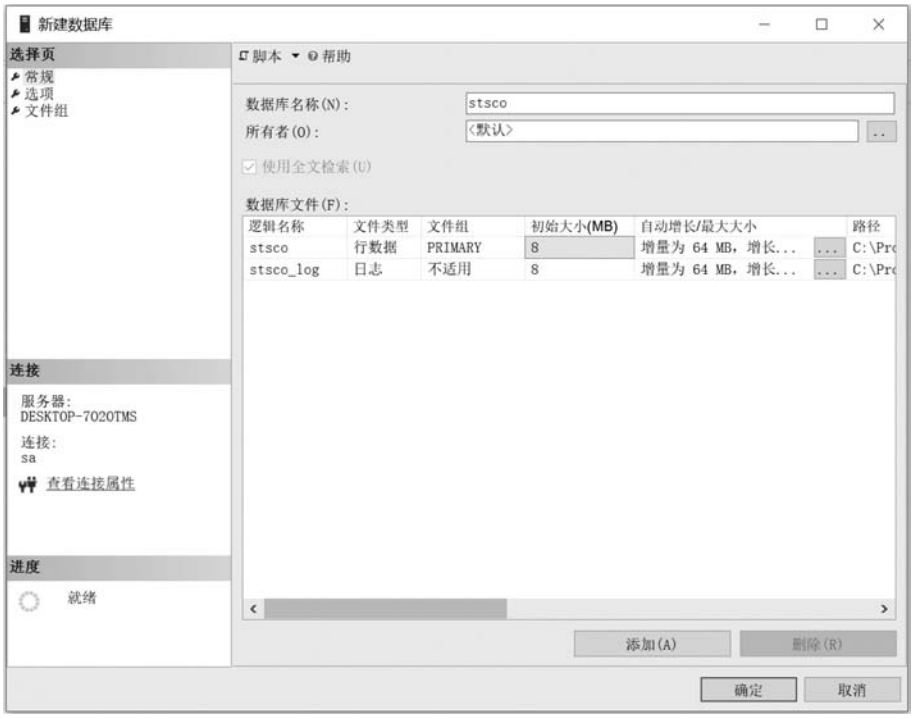

图 3.2 “新建数据库”窗口

这里只配置“常规”选项,其他选项采用系统默认设置。

(4) 单击“确定”按钮,stsco 数据库创建完成,在 C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\文件夹中增加了两个数据文件 stsco.mdf 和 stsco_log.ldf。

2. 修改数据库

在创建数据库后,用户可以根据需要对数据库进行以下修改。

- 增加或删除数据文件,改变数据文件的大小和增长方式。
- 增加或删除日志文件,改变日志文件的大小和增长方式。
- 增加或删除文件组。

【例 3.2】 在 abc 数据库(已创建)中增加数据文件 abcbk.ndf 和日志文件 abcbk_log.ldf。

操作步骤如下。

(1) 启动 SQL Server Management Studio,在左边“对象资源管理器”窗口中展开“数据库”节点,选中数据库 abc,右击,在弹出的快捷菜单中选择“属性”命令。

(2) 在弹出的“数据库属性-abc”窗口中,单击“选择页”中的“文件”选项,进入文件设置页面,如图 3.3 所示。通过本窗口可增加数据文件和日志文件。

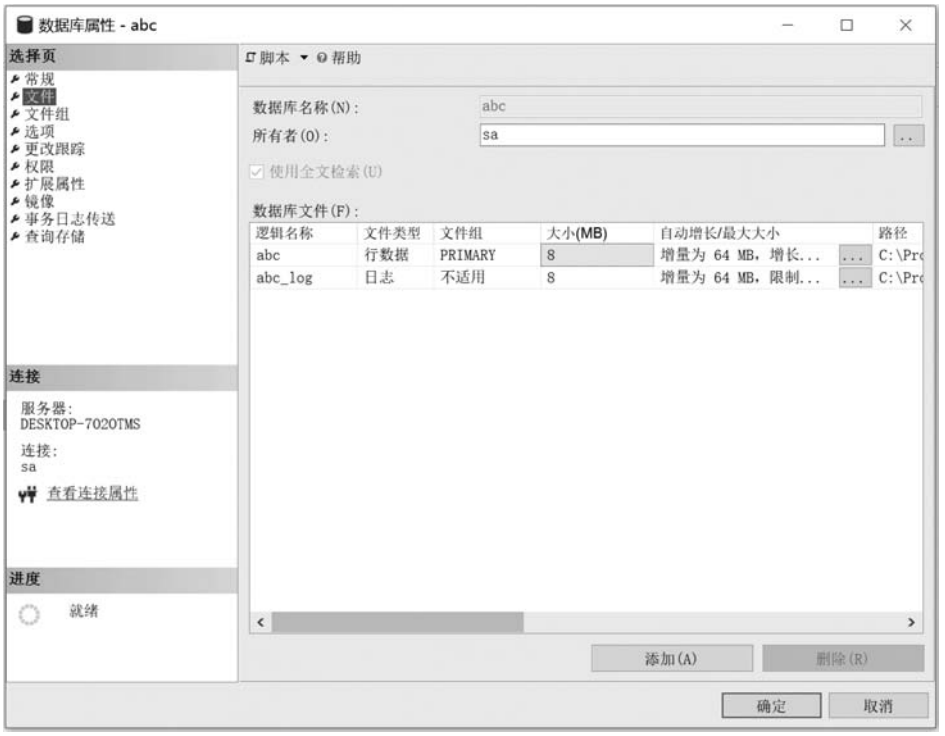

图 3.3 “文件”选项

(3) 增加数据文件。单击“添加”按钮,在“数据库文件”列表中出现一个新的文件位置,单击“逻辑名称”文本框并输入名称 abcbk,单击“大小”文本框,通过该框后的微调按钮将大小设置为 10MB,单击“自动增长”文本框中的“...”按钮,出现“更改 abcbk 的自动增长设置”对话框,将文件增长设置为 15MB,最大文件大小限制为 100MB,“文件类型”文本框、“文件组”文本框和“路径”文本框都选择默认值。

(4) 增加日志文件。单击“添加”按钮,在“数据库文件”列表中出现一个新的文件位置,单击“逻辑名称”文本框并输入名称 abcbk_log,单击“文件类型”文本框,在该框的下拉列表中选择“日志”,单击“大小”文本框,通过该框后的微调按钮将大小设置为 5MB,单击“自动增长”文本框中的“...”按钮,出现“更改 abcbk_log 的自动增长设置”对话框,将文件增长设置为 10%,最大文件大小无限制,“文件组”文本框和“路径”文本框都选择默认值,如图 3.4 所示,单击“确定”按钮。

操作完成后,在 C:\Program Files\Microsoft SQL Server\MSSQL15. MSSQLSERVER\MSSQL\DATA 文件夹中增加了辅助数据文件 abcbk.ndf 和日志文件 abcbk_log.ldf。

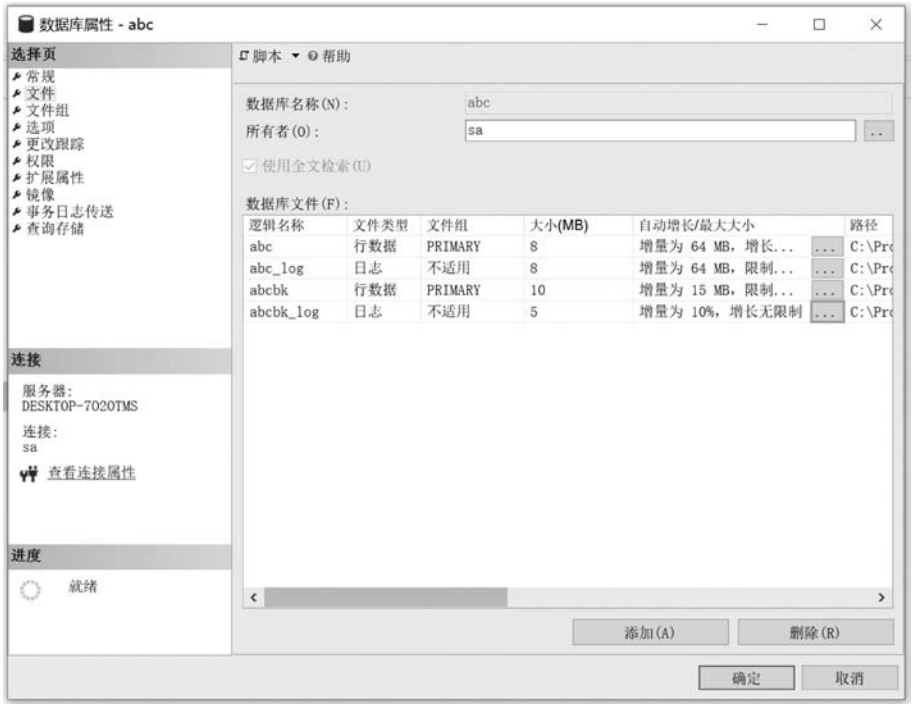

图 3.4　增加数据文件和日志文件

【例 3.3】 在 abc 数据库中删除数据文件和日志文件。

操作步骤如下。

(1) 启动 SQL Server Management Studio，在左边“对象资源管理器”窗口中展开“数据库”节点，选中数据库 abc，右击，在弹出的快捷菜单中选择“属性”命令。

(2) 出现“数据库属性-abc”窗口，单击“选择页”中的“文件”选项，进入文件设置页面，通过本窗口可删除数据文件和日志文件。

(3) 选择 abcbk 数据文件，单击“删除”按钮，该数据文件被删除。

(4) 选择 abcbk_log 日志文件，单击“删除”按钮，该日志文件被删除。

(5) 单击“确定”按钮，返回 SQL Server Management Studio 窗口。

3. 删除数据库

数据库运行后，需要消耗资源，往往会降低系统运行效率，通常可将不再需要的数据库进行删除，释放资源。删除数据库后，其文件及数据都会从服务器上的磁盘中删除，并永久删除，除非使用以前的备份，所以删除数据库应谨慎。

【例 3.4】 删除 abc 数据库。

操作步骤如下。

(1) 启动 SQL Server Management Studio，在左边“对象资源管理器”窗口中展开“数据库”节点，选中数据库 abc，右击，在弹出的快捷菜单中选择“删除”命令。

(2) 出现“删除对象”对话框，单击“确定”按钮，abc 数据库被删除。

3.3 以命令方式创建 SQL Server 数据库

3.2 节介绍了使用 SQL Server Management Studio 的图形用户界面创建 SQL Server 数据库,本节介绍使用 T-SQL 语句创建 SQL Server 数据库,与图形用户界面相比,使用 T-SQL 语句创建 SQL Server 数据库更为灵活、方便。

3.3.1 创建数据库

创建数据库使用 CREATE DATABASE 语句。

语法格式:

```
CREATE DATABASE database_name
[   [ON [filespec] ]
    [LOG ON [filespec] ]
]

< filespec >:: =
{(
 NAME = logical_file_name ,
 FILENAME = 'os_file_name'
 [, SIZE = size]
 [, MAXSIZE = {max_size | UNLIMITED }]
 [, FILEGROWTH = growth_increament [ KB | MB | GB | TB | % ]] )
}
```

说明:

- database_name: 创建的数据库名称,命名须唯一且符合 SQL Server 的命名规则,最多为 128 个字符。
- ON 子句: 指定数据库文件和文件组属性。
- LOG ON 子句: 指定日志文件属性。
- filespec: 指定数据文件的属性,给出文件的逻辑名、存储路径、大小及增长特性。
- NAME: 为 filespec 定义的文件指定逻辑文件名。
- FILENAME: 为 filespec 定义的文件指定操作系统文件名,指出定义物理文件时使用的路径和文件名。
- SIZE 子句: 指定 filespec 定义的文件的初始大小。
- MAXSIZE 子句: 指定 filespec 定义的文件的最大大小。
- FILEGROWTH 子句: 指定 filespec 定义的文件的增长增量。

当仅使用 CREATE DATABASE database_name 语句而不带参数时,创建的数据库大小将与 model 数据库的大小相等。

【例 3.5】 使用 T-SQL 语句,创建 test 数据库。

在 SQL Server 查询编辑器中输入以下语句。

```
CREATE DATABASE test
```

```
    ON
    (
        NAME = 'test',
        FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\test.mdf',
        SIZE = 12MB,
        MAXSIZE = 30MB,
        FILEGROWTH = 1MB
    )
    LOG ON
    (
        NAME = 'test_log',
        FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\test_
log.ldf',
        SIZE = 1MB,
        MAXSIZE = 10MB,
        FILEGROWTH = 10 %
    )
```

在查询编辑器窗口中单击“执行”按钮或按 F5 键，系统提示“命令已成功完成”。

【例 3.6】 创建 test2 数据库，其中主数据文件为 20MB，最大大小不限，按 1MB 增长；1 个日志文件，大小为 1MB，最大为 20MB，按 10%增长。

在 SQL Server 查询编辑器中输入以下语句。

```
CREATE DATABASE test2
    ON
    (
        NAME = 'test2',
        FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\test2.mdf',
        SIZE = 20MB,
        MAXSIZE = UNLIMITED,
        FILEGROWTH = 1MB
    )
    LOG ON
    (
        NAME = 'test2_log',
        FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\test2_
log.ldf',
        SIZE = 1MB,
        MAXSIZE = 20MB,
        FILEGROWTH = 10 %
    )
```

在查询编辑器窗口中单击“执行”按钮或按 F5 键，系统提示“命令已成功完成”。

【例 3.7】 创建一个具有两个文件组的数据库 test3。要求：主文件组包括文件 test3_dat1，文件初始大小为 15MB，最大为 45MB，按 4MB 增长；另一个文件组名为 test3gp，包括文件 test3_dat2，文件初始大小为 5MB，最大为 20MB，按 10%增长。

在 SQL Server 查询编辑器中输入以下语句。

```
CREATE DATABASE test3
    ON
```

```
    PRIMARY
    (
       NAME = 'test3_dat1',
       FILENAME = 'D:\data\ test3_dat1.mdf',
       SIZE = 15MB,
       MAXSIZE = 45MB,
       FILEGROWTH = 4MB
    ),
    FILEGROUP test3gp
    (
       NAME = 'test3_dat2',
       FILENAME = 'D:\data\ test3_dat2.ndf',
       SIZE = 5MB,
       MAXSIZE = 20MB,
       FILEGROWTH = 10 %
    )
```

在查询编辑器窗口中单击“执行”按钮或按 F5 键,系统提示“命令已成功完成”。

创建数据库后使用数据库时,可使用 USE 语句。

语法格式:

```
USE database_name
```

其中,database_name 是使用的数据库名称。

说明: USE 语句只在第一次打开数据库时使用,后续都作用在该数据库中。如果要使用另一数据库,需要重新使用 USE 语句。

3.3.2 修改数据库

修改数据库使用 ALTER DATABASE 语句。

语法格式:

```
ALTER DATABASE database
{ ADD FILE filespec
| ADD LOG FILE filespec
| REMOVE FILE logical_file_name
| MODIFY FILE filespec
| MODIFY NAME = new_dbname
}
```

说明:

- database:需要更改的数据库名称。
- ADD FILE 子句:指定要增加的数据文件。
- ADD LOG FILE 子句:指定要增加的日志文件。
- REMOVE FILE 子句:指定要删除的数据文件。
- MODIFY FILE 子句:指定要更改的文件属性。
- MODIFY NAME 子句:重命名数据库。

【例 3.8】 在 test2 数据库中，增加一个数据文件 test2add，大小为 10MB，最大为 50MB，按 5MB 增长。

```
ALTER DATABASE test2
   ADD FILE
   (
      NAME = 'test2add',
      FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\
test2add.ndf',
      SIZE = 10MB,
      MAXSIZE = 50MB,
      FILEGROWTH = 5MB
   )
```

3.3.3 删除数据库

删除数据库使用 DROP DATABASE 语句。

语法格式：

```
DROP DATABASE database_name
```

其中，database_name 是要删除的数据库名称。

【例 3.9】 使用 T-SQL 语句删除 test3 数据库。

```
DROP DATABASE test3
```

3.4 表的基本概念

在 SQL Server 中，每个数据库包括若干表，表用于存储数据。在建立数据库的过程中，最重要的一步就是创建表。下面介绍创建表要用到的两个基本概念：表和数据类型。

3.4.1 表和表结构

表是 SQL Server 中最基本的数据库对象，用于存储数据的一种逻辑结构，由行和列组成，又称为二维表。例如，在学生成绩管理系统中的学生表(student)，如表 3.1 所示。

表 3.1 学生表(student)

学号	姓名	性别	出生日期	专业	总学分
201001	罗俊杰	男	2000-06-15	通信	52
201002	韩红丽	女	1999-09-23	通信	49
201004	冯露	女	1999-08-07	通信	50
202001	徐桥	男	2000-02-25	计算机	52
202002	袁志敏	男	1999-12-04	计算机	48
202005	董莎	女	2000-04-19	计算机	50

(1) 表

表是数据库中存储数据的数据库对象,每个数据库包含了若干表,表由行和列组成。例如,表 3.1 的学生表由 6 行 6 列组成。

(2) 表结构

每个表具有一定的结构,表结构包含一组固定的列,列由数据类型、长度、允许 NULL 值等组成。

(3) 记录

每个表包含若干行数据,表中一行称为一个记录(Record)。表 3.1 的学生表有 6 个记录。

(4) 字段

表中每列称为字段(Field),每个记录由若干数据项(列)构成,构成记录的每个数据项就称为字段。表 3.1 的学生表有 6 个字段。

(5) 空值

空值(NULL)通常表示未知、不可用或将在以后添加的数据。

(6) 关键字

关键字用于唯一标识记录,如果表中记录的某一字段或字段组合能唯一标识记录,则该字段或字段组合称为候选关键字(Candidate Key)。如果一个表有多个候选关键字,则选定其中的一个为主关键字(Primary Key),又称为主键。表 3.1 的学生表的主键为“学号”。

3.4.2 数据类型

SQL Server 支持两类数据类型:系统数据类型和用户自定义数据类型。

1. 系统数据类型

创建数据库最重要的一步为创建其中的数据表,创建数据表必须定义表结构和设置列的数据类型、长度等。下面介绍 SQL Server 系统数据类型,包括整数型、精确数值型、浮点型、货币型、位型、字符型、Unicode 字符型、文本型、二进制型、日期时间类型、时间戳型、图像型、其他等,如表 3.2 所示。

表 3.2 SQL Server 系统数据类型

数据类型	符号标识
整数型	bigint,int,smallint,tinyint
精确数值型	decimal,numeric
浮点型	float,real
货币型	money,smallmoney
位型	bit
字符型	char,varchar,varchar(MAX)
Unicode 字符型	nchar,nvarchar,nvarchar(MAX)
文本型	text,ntext
二进制型	binary,varbinary,varbinary(MAX)
日期时间类型	datetime,smalldatetime,date,time,datetime2,datetimeoffset
时间戳型	timestamp
图像型	image
其他	cursor,sql_variant,table,uniqueidentifier,xml,hierarchyid

(1) 整数型

整数型包括 bigint、int、smallint 和 tinyint 4 类。

• bigint(大整数)

精度为 19 位,长度为 8 字节,数值范围为-2^{63}～$2^{63}-1$。

• int(整数)

精度为 10 位,长度为 4 字节,数值范围为-2^{31}～$2^{31}-1$。

• smallint(短整数)

精度为 10 位,长度为 2 字节,数值范围为-2^{15}～$2^{15}-1$。

• tinyint(微短整数)

精度为 3 位,长度为 1 字节,数值范围为 0～255。

(2) 精确数值型

精确数值型包括 decimal 和 numeric 两类,这两种数据类型在 SQL Server 中的功能是完全等价的。

精确数值型数据由整数部分和小数部分构成,可存储从$-10^{38}+1$ 到$10^{38}-1$ 的固定精度和小数位的数字数据,它的存储长度最少为 5 字节,最多为 17 字节。

精确数值型数据的格式是:

```
numeric | decimal(p[,s])
```

其中,p 为精度,s 为小数位数,s 的默认值为 0。

例如,指定某列为精确数值型,精度为 7,小数位数为 2,则为 decimal(7,2)。

(3) 浮点型

浮点型又称近似数值型,近似数值数据类型包括 float[(n)]和 real 两类,这两类通常都使用科学记数法表示数据。科学记数法的格式为:

```
尾数 E 阶数
```

其中,阶数必须为整数。

例如,4.804 E9、3.682-E6、7 8594E-8 等都是浮点型数据。

• real

精度为 7 位,长度为 4 字节,数值范围为$-3.40E+38$～$3.40E+38$。

• float[(n)]

当 n 为 1～24 时,精度为 7 位,长度为 4 字节,数值范围为$-3.40E+38$～$3.40E+38$。

当 n 为 25～53 时,精度为 15 位,长度为 8 字节,数值范围为$-1.79E+308$～$1.79E+308$。

(4) 货币型

处理货币的数据类型有 money 和 smallmoney,它们用十进制数表示货币值。

• money

精度为 19,小数位数为 4,长度为 8 字节,数值范围为-2^{63}～$2^{63}-1$。

• smallmoney

精度为 10,小数位数为 4,长度为 4 字节,数值范围为-2^{31}～$2^{31}-1$。

(5) 位型

SQL Server 中的位(bit)型数据只存储 0 和 1,长度为 1 字节,相当于其他语言中的逻

辑型数据。当一个表中有小于 8 位的 bit 列,将作为一字节存储,如果表中有 9～16 位 bit 列,将作为两字节存储,依此类推。

当为 bit 类型数据赋 0 时,其值为 0;而赋非 0 时,其值为 1。

字符串值 TRUE 和 FALSE 可以转换的 bit 值:TRUE 转换为 1,FALSE 转换为 0。

(6) 字符型

字符型数据用于存储字符串,字符串中可包括字母、数字和其他特殊符号。在输入字符串时,需将字符串中的符号用单引号或双引号括起来,如'def'、"Def＜Ghi"。

SQL Server 字符型包括两类:固定长度(char 字符数据类型) 字符数据类型、可变长度(varchar)字符数据类型。

• char[(n)]

固定长度字符数据类型,其中 n 定义字符型数据的长度,n 的取值为 1～8000,默认值为 1。若输入字符串长度小于 n 时,则系统自动在它的后面添加空格以达到长度 n。例如,某列的数据类型为 char(100),而输入的字符串为"NewYear2013",则存储的是字符 NewYear2013 和 89 个空格。若输入字符串长度大于 n,则截断超出的部分。当列值的字符数基本相同时,可采用数据类型 char[(n)]。

• varchar[(n)]

可变长度字符数据类型,其中 n 的规定与固定长度字符数据类型 char[(n)]中 n 完全相同,与 char[(n)]不同的是,varchar(n)数据类型的存储空间随列值的字符数而变化。例如,表中某列的数据类型为 varchar(100),而输入的字符串为"NewYear2013",则存储的字符 NewYear2013 的长度为 11 字节,其后不添加空格,因而 varchar(n)数据类型可以节省存储空间,特别是在列值的字符数显著不同时。

(7) Unicode 字符型

Unicode 是"统一字符编码标准",用于支持国际上非英语语种的字符数据的存储和处理。Unicode 字符型包括 nchar[(n)]和 nvarchar[(n)]两类。nchar[(n)]、nvarchar[(n)]和 char[(n)]、varchar(n)类似,只是前者使用 Unicode 字符集,后者使用 ASCII 字符集。

• nchar[(n)]

固定长度 Unicode 数据的数据类型,n 的取值为 1～4000,长度为 2n 字节,若输入的字符串长度不足 n,将以空白字符补足。

• nvarchar[(n)]

可变长度 Unicode 数据的数据类型,n 的取值为 1～4000,长度是所输入字符个数的两倍。

(8) 文本型

由于字符型数据的最大长度为 8000 个字符,当存储超出上述长度的字符数据(如较长的备注、日志等),即不能满足应用需求,此时需要文本型数据。

文本型包括 text 和 ntext 两类,分别对应 ASCII 字符和 Unicode 字符。

• text

最大长度为 $2^{31}-1$(2 147 483 647)个字符,存储字节数与实际字符个数相同。

• ntext

最大长度为 $2^{30}-1$(1 073 741 823)个 Unicode 字符,存储字节数是实际字符个数的 2 倍。

(9) 二进制型

二进制数据类型表示的是位数据流，包括 binary(固定长度)和 varbinary(可变长度)两种。

• binary[(n)]

固定长度的 n 字节二进制数据，n 的取值范围为 1～8000，默认值为 1。

binary(n)数据的存储长度为 n+4 字节。若输入的数据长度小于 n，则不足部分用 0 填充；若输入的数据长度大于 n，则多余部分被截断。

输入二进制值时，在数据前面要加上 0x，可以用的数字符号为 0～9、A～F(字母大小写均可)。例如，0xBE、0x5F0C 分别表示值 BE 和 5F0C。由于每字节的数最大为 FF，故 0x 格式的数据每两位占 1 字节，二进制数据有时也被称为十六进制数据。

• varbinary[(n)]

n 字节变长二进制数据，n 取值范围为 1～8000，默认值为 1。

varbinary(n)数据的存储长度为：实际输入数据长度+4 字节。

(10) 日期时间类型

日期时间类型数据用于存储日期和时间信息，共有 datetime、smalldatetime、date、time、datetime2、datetimeoffset 六种。

• datetime

datetime 类型可表示的日期范围从 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。

datetime 类型数据长度为 8 字节，日期和时间分别使用 4 字节存储。前 4 字节用于存储基于 1900 年 1 月 1 日之前或之后的天数，正数表示日期在 1900 年 1 月 1 日之后，负数则表示日期在 1900 年 1 月 1 日之前。后 4 字节用于存储距 12:00(24 小时制)的毫秒数。

默认的日期时间是 January 1,1900 12:00 A. M。可以接受的输入格式有 January 10 2012、Jan 10 2012、JAN 10 2012、January 10,2012 等。

• smalldatetime

smalldatetime 与 datetime 数据类型类似，但日期时间范围较小，表示从 1900 年 1 月 1 日到 2079 年 6 月 6 日的日期和时间，存储长度为 4 字节。

• date

date 类型可表示从公元元年 1 月 1 日到 9999 年 12 月 31 日的日期，表示形式与 datetime 数据类型的日期部分相同，只存储日期数据，不存储时间数据，存储长度为 3 字节。

• time

time 数据类型只存储时间数据，表示格式为 hh:mm:ss[.nnnnnnn]。hh 表示小时，范围为 0～23。mm 表示分钟，范围为 0～59。ss 表示秒数，范围为 0～59。n 是 0～7 位数字，范围为 0～9 999 999，表示秒的小数部分，即微秒数。所以 time 数据类型的取值范围为 00:00:00.000 000 0～23:59:59.999 999 9。time 类型的存储大小为 5 字节。另外，还可以自定义 time 类型微秒数的位数，例如 time(1)表示小数位数为 1，默认为 7。

• datetime2

新的 datetime2 数据类型和 datetime 类型一样，也用于存储日期和时间信息。但是 datetime2 类型取值范围更广，日期部分取值范围从公元元年 1 月 1 日到 9999 年 12 月 31

日,时间部分的取值范围从 00:00:00.000 000 0～23:59:59.999 999。另外,用户还可以自定义 datetime2 数据类型中微秒数的位数,例如 datetime(2)表示小数位数为 2。datetime2 类型的存储大小随着微秒数的位数(精度)而改变,精度小于 3 时为 6 字节,精度为 4 和 5 时为 7 字节,所有其他精度则需要 8 字节。

• datetimeoffset

datetimeoffset 数据类型也用于存储日期和时间信息,取值范围与 datetime2 类型相同。但 datetimeoffset 类型具有时区偏移量,此偏移量指定时间相对于协调世界时(UTC)偏移的小时和分钟数。datetimeoffset 的格式为 YYYY-MM-DD hh:mm:ss[.nnnnnnn] [{+|-}hh:mm],其中 hh 为时区偏移量中的小时数,范围为 00～14,mm 为时区偏移量中的额外分钟数,范围为 00～59。时区偏移量中必须包含“+”(加)或“-”(减)号。这两个符号表示是在 UTC 时间的基础上加上还是从中减去时区偏移量以得出本地时间。时区偏移量的有效范围为-14:00～+14:00。

(11) 时间戳型

反映系统对该记录修改的相对(相对于其他记录)顺序,标识符是 timestamp,timestamp 类型数据的值是二进制格式数据,其长度为 8 字节。

若创建表时定义一个列的数据类型为时间戳类型,那么每当对该表加入新行或修改已有行时,都由系统自动将一个计数器值加到该列,即将原来的时间戳值加上一个增量。

(12) 图像型

用于存储图片、照片等,标识符为 image,实际存储的是可变长度二进制数据,范围为 $0\sim2^{31}-1$(2 147 483 647) 字节。

(13) 其他

SQL Server 还提供了其他几种数据类型: cursor、sql_variant、table、uniqueidentifier、xml 和 hierarchyid。

• cursor

游标数据类型,用于创建游标变量或定义存储过程的输出参数。

• sql_variant

一种存储 SQL Server 支持的各种数据类型(除 text、ntext、image、timestamp 和 sql_variant 外)值的数据类型,sql_variant 的最大长度可达 8016 字节。

• table

用于存储结果集的数据类型,结果集可以供后续处理。

• uniqueidentifier

唯一标识符类型,系统将为这种类型的数据产生唯一标识值,它是一个 16 字节的二进制数据。

• xml

用来在数据库中保存 xml 文档和片段的一种类型,文件大小不能超过 2GB。

• hierarchyid

hierarchyid 数据类型是 SQL Server 新增加的一种长度可变的系统数据类型,可使用 hierarchyid 表示层次结构中的位置。

2. 用户自定义数据类型

在 SQL Server 中，除系统提供的数据类型外，用户还可以自定义数据类型。用户自定义数据类型根据基本数据类型进行定义，可将一个名称用于一个数据类型，能更好说明该对象中保存值的类型，方便用户使用。例如，student 表和 score 表都有 stid 列，该列应有相同的类型，即均为字符型值、长度为 6，不允许为空值，为了含义明确、使用方便，由用户定义一个数据类型，命名为 school_student_num，作为 student 表和 score 表的 stid 列的数据类型。

创建用户自定义数据类型应有以下三个属性。

- 新数据类型的名称。
- 新数据类型所依据的系统数据类型。
- 为空性。

(1) 创建用户自定义数据类型

使用 CREATE TYPE 语句来实现用户数据类型的定义。

语法格式：

```
CREATE TYPE [ schema_name. ] type_name
    FROM base_type [ ( precision [ , scale ] ) ]
    [ NULL | NOT NULL ]
[ ; ]
```

说明：

type_name 为指定用户自定义数据类型名称，base_type 为用户自定义数据类型所依据的系统数据类型。

【例 3.10】 使用 CREATE TYPE 命令创建用户自定义数据类型 school_student_num。

```
CREATE TYPE school_student_num
FROM char(6) NOT NULL
```

该语句创建了用户自定义数据类型 school_student_num。

(2) 删除用户自定义数据类型

使用 DROP TYPE 语句删除自定义数据类型。

语法格式：

```
DROP TYPE [ schema_name. ] type_name [ ; ]
```

例如，删除前面定义的类型 school_student_num 的语句为：

```
DROP TYPE school_student_num
```

3.4.3 表结构设计

创建表的核心是定义表结构及设置表和列的属性。创建表以前，首先要确定表名和表的属性，表所包含的列名、列的数据类型、长度、是否为空、是否主键等，这些属性构成表结构。

以学生成绩管理系统的 student(学生表)为例介绍表结构设计。

学生表 student 包含 stid、stname、stsex、stbirthday、speciality、tc 等列，其中，stid 列是

学生的学号,例如 201001 中 20 表示学生入学时间为 2020 年,10 表示学生的班级,01 表示学生的序号,所以 stid 列的数据类型选定长的字符型 char[(n)],n 的值为 6,不允许空;stname 列是学生的姓名,姓名一般不超过 4 个中文字符,所以选定长的字符型数据类型,n 的值为 8,不允许空;stsex 列是学生的性别,选字符型 char[(n)],n 的值为 2,不允许空;stbirthday 列是学生的出生日期,选 date 数据类型,不允许空;speciality 列是学生的专业,选定长的字符型数据类型,n 的值为 12,允许空;tc 列是学生的总学分,选整数型数据类型,不允许空。在 student 表中,只有 stid 列能唯一标识一个学生,所以将 stid 列设为主键。student 的表结构设计如表 3.3 所示。

表 3.3　student 的表结构

列　　名	数据类型	允许 NULL 值	是否主键	说明
stid	char(6)		主键	学号
stname	char(8)			姓名
stsex	char(2)			性别
stbirthday	date			出生日期
speciality	char(12)	√		专业
tc	int	√		总学分

3.5　以命令方式创建 SQL Server 表

可以采用 T-SQL 语句或图形用户界面创建 SQL Server 表,本节介绍使用 T-SQL 中的语句对表进行创建、修改和删除。

3.5.1　创建表

1. 使用 CREATE TABLE 语句创建表

语法格式:

```
CREATE TABLE [ database_name . [ schema_name ] . | schema_name . ] table_name
(
 {  <column_definition>
   | column_name AS computed_column_expression [PERSISTED [NOT NULL]]
 }
 [ <table_constraint> ] [ ,...n ]
)
[ ON { partition_scheme_name ( partition_column_name ) | filegroup | "default" } ]
 [ { TEXTIMAGE_ON { filegroup | "default" } ]
[ FILESTREAM_ON { partition_scheme_name | filegroup | "default" } ]
 [ WITH ( <table_option> [ ,...n ] ) ]
[ ; ]

<column_definition> ::=
column_name data_type
```

```
    [ FILESTREAM ]
    [ COLLATE collation_name ]
    [ NULL | NOT NULL ]
    [
       [ CONSTRAINT constraint_name ]
         DEFAULT constant_expression ]
       | [ IDENTITY [ ( seed ,increment ) ] [ NOT FOR REPLICATION ]
    ]
  [ ROWGUIDCOL ]
[ <column_constraint> [ ...n ] ]
      [ SPARSE ]
```

说明：

(1) database_name 是数据库名，schema_name 是表所属架构名，table_name 是表名。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) < column_definition >列定义：

- column_name 为列名，data_type 为列的数据类型。
- FILESTREAM 是 SQL Server 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。
- NULL | NOT NULL：确定列是否可取空值。
- DEFAULT constant_expression：为所在列指定默认值。
- IDENTITY：表示该列是标识符列。
- ROWGUIDCOL：表示新列是行的全局唯一标识符列。
- < column_constraint >：列的完整性约束，指定主键、外键等。
- SPARSE：指定列为稀疏列。

(3) column_name AS computed_column_expression[PERSISTED[NOT NULL]]：用于定义计算字段。

(4) < table_constraint >：表的完整性约束。

(5) ON 子句：filegroup|"default"指定存储表的文件组。

(6) TEXTIMAGE_ON{filegroup|"default"}：TEXTIMAGE_ON 指定存储 text、ntext、image、xml、varchar(MAX)、nvarchar(MAX)、varbinary(MAX)和 CLR 用户自定义类型数据的文件组。

(7) FILESTREAM_ON 子句：filegroup|"default"指定存储 FILESTREAM 数据的文件组。

【例 3.11】 使用 T-SQL 语句，在 stsco 数据库中创建 student 表、score 表。

在 stsco 数据库中创建 student 表的语句如下。

```
USE stsco
CREATE TABLE student
   (
      stid char(6) NOT NULL PRIMARY KEY,
      stname char(8) NOT NULL,
      stsex char(2) NOT NULL,
      stbirthday date NOT NULL,
```

```
        speciality char(12) NULL,
        tc int NULL
    )
GO
```

上面的 T-SQL 语句,首先指定 stsco 数据库为当前数据库,然后使用 CREATE TABLE 语句在 stsco 数据库中创建 student 表,student 表的表结构如图 3.5 所示。

上述语句中的 GO 命令不是 Transact-SQL 语句,它是由 SQL Server Management Studio 代码编辑器识别的命令。SQL Server 实用工具将 GO 解释为应该向 SQL Server 实例发送当前批 Transact-SQL 语句的信号。当前批语句由上一条 GO 命令后输入的所有语句组成,如果是第一条 GO 命令,则由会话或脚本开始后输入的所有语句组成。GO 命令和 Transact-SQL 语句不能在同一行中,但在 GO 命令行中可包含注释。

注意:SQL Server 应用程序可以将多个 Transact-SQL 语句作为一个批发送到 SQL Server 的实例来执行。然后,该批中的语句被编译成一个执行计划。程序员在 SQL Server 实用工具中执行特殊语句,或生成 Transact-SQL 语句的脚本在 SQL Server 实用工具中运行时,使用 GO 作为批结束的信号。

提示:由一条或多条 T-SQL 语句组成一个程序,通常以.sql 为扩展名存储,称为 sql 脚本。双击 sql 脚本文件,其 T-SQL 语句即出现在查询编辑器窗口内。查询编辑器窗口内的 T-SQL 语句,可用“文件”→“另存为”命令命名并存入指定目录。

在 stsco 数据库中创建 score 表的语句如下。

```
USE stsco
USE stsco
CREATE TABLE score
    (
        stid char (6) NOT NULL,
        cid char(3) NOT NULL,
        grade int NULL,
        PRIMARY KEY(stid,cid)
    )
GO
```

注意:如果主键由多个列组成,可采用 PRIMARY KEY(列 1,列 2,…)语句,其中一个列将允许重复值,但是主键中所有列的值的各种组合必须是唯一的。

score 表的表结构如图 3.6 所示。

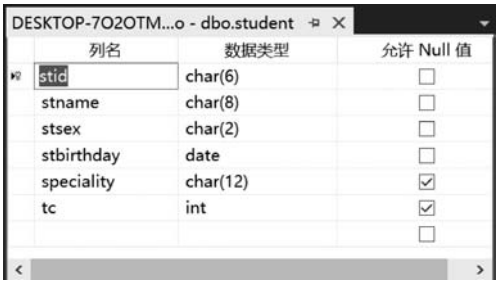

图 3.5　student 表的表结构

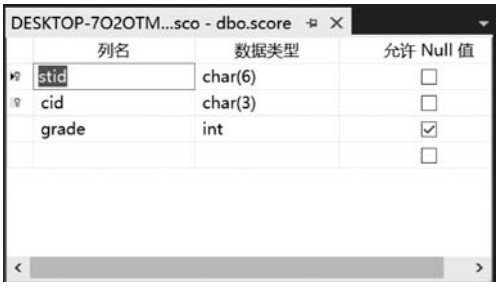

图 3.6　score 表的表结构

【例 3.12】 在 test 数据库中创建 clients 表。

```
USE test
CREATE TABLE clients
   (
      cno int,
      cname char(8),
      csex char(2),
      address char(40)
   )
```

2. 由其他表创建新表

使用 SELECT INTO 语句创建一个新表,并用 SELECT 的结果集填充该表。

语法格式:

```
SELECT 列名表 INTO 表 1 FROM 表 2
```

该语句的功能是由“表 2”的“列名表”来创建新表“表 1”。

【例 3.13】 在 stsco 数据库中,由 student 表创建 student1 表。

```
USE stsco
SELECT stid,stname,stbirthday INTO student1
FROM student
```

3.5.2 修改表

使用 ALTER TABLE 语句修改表的结构。

语法格式:

```
ALTER TABLE table_name
{
 ALTER COLUMN column_name
 {
        new_data_type [ (precision,[,scale])] [NULL | NOT NULL]
        | {ADD | DROP } { ROWGUIDCOL | PERSISTED | NOT FOR REPLICATION | SPARSE }
 }/
 | ADD {[< colume_definition >]}[,…n]
 | DROP {[CONSTRAINT] constraint_name | COLUMN column}[,…n]
}
```

说明:

(1) table_name 为表名。

(2) ALTER COLUMN 子句:修改表中指定列的属性。

(3) ADD 子句:增加表中的列。

(4) DROP 子句:删除表中的列或约束。

【例 3.14】 在 student1 表中新增加一列 remarks。

```
USE stsco
```

```
ALTER TABLE student1 ADD remarks char(10)
```

3.5.3 删除表

使用 DROP TABLE 语句删除表。

语法格式:

```
DROP TABLE table_name
```

其中,table_name 是要删除的表的名称。

【例 3.15】 删除 stsco 数据库中的 student1 表。

```
USE stsco
DROP TABLE student1
```

3.6 使用图形用户界面创建 SQL Server 表

使用图形用户界面创建 SQL Server 表包括创建表、修改表、删除表等内容。

1. 创建表

【例 3.16】 在 stsco 数据库中创建 student 表(学生表)。

操作步骤如下。

(1) 启动 SQL Server Management Studio,在"对象资源管理器"中展开"数据库"节点,选中 stsco 数据库,展开该数据库,选中"表",右击,在弹出的快捷菜单中选择"新建"→"表"命令,如图 3.7 所示。

图 3.7 选择"新建"→"表"命令

(2) 屏幕出现表设计器窗口,根据已经设计好的 student 的表结构分别输入或选择各列的数据类型、长度、允许 Null 值,根据需要,可以在每列的"列属性"表格填入相应内容,输入完成后的结果如图 3.8 所示。

(3) 在 stid 行上右击，在弹出的快捷菜单中选择“设置主键”命令，如图 3.9 所示，此时，stid 左边会出现一个钥匙图标。

图 3.8　输入或选择各列的数据类型、长度、允许 Null 值

图 3.9　选择“设置主键”命令

注意：如果主键由两个或两个以上的列组成，需要按住 Ctrl 键选择多个列，再在右键快捷菜单中选择“设置主键”命令。

(4) 单击工具栏中的“保存”按钮，出现“选择名称”对话框，输入表名称 student，如图 3.10 所示，单击“确定”按钮即可创建 student 表，如图 3.11 所示。

图 3.10　设置表的名称

图 3.11　创建 student 表

2. 修改表

在 SQL Server 中,当用户使用 SQL Server Management Studio 修改表的结构(如增加列、删除列、修改已有列的属性等)时,必须要删除原表,再创建新表才能完成表的更改。如果强行更改,会弹出不允许保存更改对话框。

为了在进行表的修改时不出现此对话框,需要进行如下操作。

在 SQL Server Management Studio 面板中选择“工具”→“选项”命令,在出现的“选项”窗口中展开“设计器”,选择“表设计器和数据库设计器”选项,取消勾选窗口右面的“阻止保存要求重新创建表的更改”复选框,如图 3.12 所示,单击“确定”按钮,就可进行表的修改了。

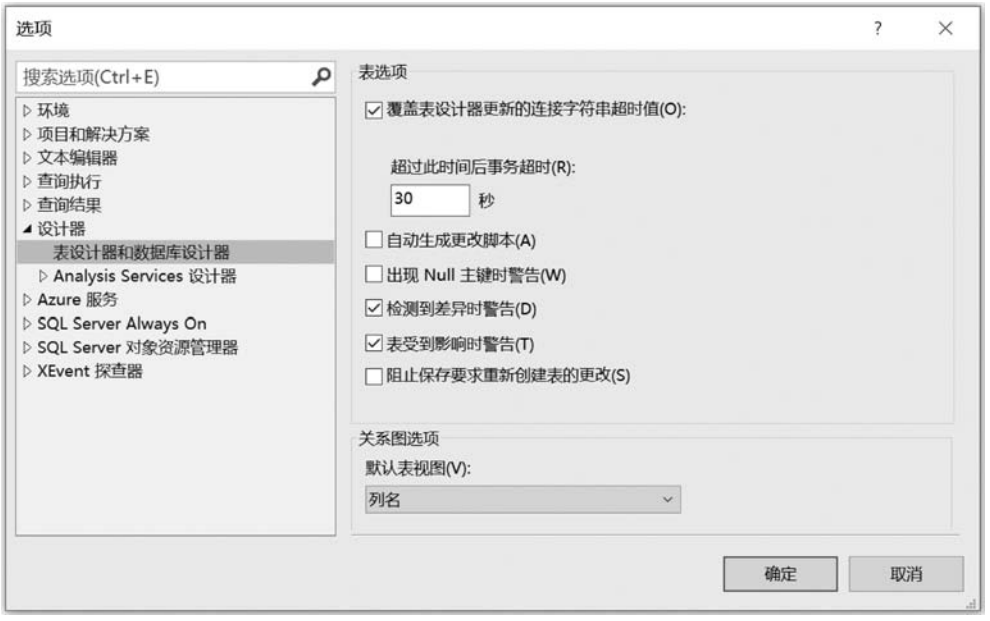

图 3.12　解除阻止保存的选项

【例 3.17】 在 student 表中增加一列 stclass(班级),在 tc 列之前,然后删除该列。

(1) 启动 SQL Server Management Studio,在“对象资源管理器”中展开“数据库”节点,选中 stsco 数据库,展开该数据库,选中表,将其展开,选中表 dbo.student,右击,在弹出的快捷菜单中选择“设计”命令,打开“表设计器”窗口,为了在 tc 列之前加入新列,右击该列,在弹出的快捷菜单中选择“插入列”命令,如图 3.13 所示。

(2) 在“表设计器”窗口中的 tc 列前出现空白行,输入列名 stclass,选择数据类型 char(6),允许空,如图 3.14 所示,完成插入新列操作。

(3) 在“表设计器”窗口中选择需删除的 stclass 列,右击,在弹出的快捷菜单中选择“删除列”命令,该列即被删除,如图 3.15 所示。

【例 3.18】 将 def 表(已创建)表名修改为 xyz 表。

(1) 启动 SQL Server Management Studio,在“对象资源管理器”中展开“数据库”节点,选中 stsco 数据库,展开该数据库,选中表,将其展开,选中表 dbo.def,右击,在弹出的快捷菜单中选择“重命名”命令。

图 3.13　选择“插入列”命令

图 3.14　插入新列

图 3.15　选择“删除列”命令

(2) 此时,表 dbo.def 的名称已变为可编辑,将名称修改为 dbo.xyz,完成表名修改。

3. 删除表

删除表时,表的结构定义、表中的所有数据以及表的索引、触发器、约束等都被删除,删除表操作时一定要谨慎小心。

【例 3.19】 删除 xyz 表(已创建)。

(1) 启动 SQL Server Management Studio,在“对象资源管理器”中展开“数据库”节点,选中 stsco 数据库,展开该数据库,选中表,将其展开,选中表 dbo.xyz,右击,在弹出的快捷

菜单中选择“删除”命令。

(2) 系统弹出“删除对象”对话框,单击“确定”按钮,即可删除 xyz 表。

3.7 小　　结

本章主要介绍了以下内容。

(1) 数据库是 SQL Server 存储和管理数据的基本对象。

从用户的观点看,组成数据库的逻辑成分称为数据库对象,SQL Server 的数据库对象包括表(table)、视图(view)、索引(index)、存储过程(stored procedure)、触发器(trigger)等。SQL Server 的数据库有两类:一类是系统数据库,另一类是用户数据库。SQL Server 在安装时创建 4 个系统数据库:master、model、msdb 和 tempdb。用户数据库是由用户创建的数据库。

从系统的观点看,数据库是存储逻辑数据库的各种对象的实体,它们存放在计算机的存储介质中,从这个角度我们称数据库为物理数据库。SQL Server 的物理数据库架构包括页和区、数据库文件、数据库文件组等。

(2) 使用图形用户界面创建 SQL Server 数据库:包括创建数据库、修改数据库、删除数据库。

(3) 使用 T-SQL 语句创建 SQL Server 数据库:包括使用 CREATE DATABASE 语句创建数据库、使用 ALTER DATABASE 语句修改数据库、使用 DROP DATABASE 语句删除数据库。

(4) 表是 SQL Server 中最基本的数据库对象,它是用于存储数据的一种逻辑结构,由行和列组成。表结构包含一组固定的列,列由数据类型、长度、允许 Null 值等组成。创建表之前,首先要确定表名和表的属性,表所包含的列名、列的数据类型、长度、是否为空、是否主键等,然后进行表结构设计。

(5) SQL Server 支持两类数据类型:系统数据类型和用户自定义数据类型。系统数据类型包括整数型、精确数值型、浮点型、货币型、位型、字符型、Unicode 字符型、文本型、二进制型、日期时间类型、时间戳型、图像型、其他数据类型等。用户自定义数据类型的创建和删除。

(6) 以命令方式创建 SQL Server 表的语句有:创建表用 CREATE TABLE 语句、修改表用 ALTER TABLE 语句、删除表用 DROP TABLE 语句。

(7) 以图形界面方式创建 SQL Server 表,包括创建表、修改表、删除表等内容。

习　题　3

一、选择题

3.1　在 SQL Server 中创建用户数据库,其主要数据文件的大小必须大于_____。

A. master 数据库的大小　　B. model 数据库的大小

C. msdb 数据库的大小　　D. 3MB

3.2　在 SQL Server 中,如果数据库 tempdb 的空间不足,可能会造成一些操作无法进

行，此时需要扩大 tempdb 的空间。下列关于扩大 tempdb 空间的方法，错误的是_____。

A. 手工扩大 tempdb 中某数据文件的大小

B. 设置 tempdb 中的数据文件为自动增长方式，每当空间不够时让其自动增长

C. 手工为 tempdb 增加一个数据文件

D. 删除 tempdb 中的日志内容，以获得更多的数据空间

3.3 在 SQL Server 中创建用户数据库，实际就是定义数据库所包含的文件以及文件的属性。下列不属于数据文件属性的是_____。

A. 初始大小　　B. 物理文件名

C. 文件结构　　D. 最大大小

3.4 SQL Server 数据库是由文件组成的。下列关于数据库所包含的文件的说法中，正确的是_____。

A. 一个数据库可包含多个主要数据文件和多个日志文件

B. 一个数据库只能包含一个主要数据文件和一个日志文件

C. 一个数据库可包含多个次要数据文件，但只能包含一个日志文件

D. 一个数据库可包含多个次要数据文件和多个日志文件

3.5 在 SQL Server 系统数据库中，存放用户数据库公共信息的是_____。

A. master　　B. model

C. msdb　　D. tempdb

3.6 出生日期字段不宜选择_____。

A. datetime　　B. bit

C. char　　D. date

3.7 性别字段不宜选择_____。

A. char　　B. tinyint

C. int　　D. float

3.8 _____字段可以采用默认值。

A. 出生日期　　B. 姓名

C. 专业　　D. 学号

3.9 设在 SQL Server 中，某关系表需要存储职工的工资信息，工资的范围为 2000～6000，采用整型类型存储。下列数据类型中最合适的是_____。

A. int　　B. smallint

C. tinyint　　D. bigint

二、填空题

3.10 从用户的观点看，组成数据库的_____称为数据库对象。

3.11 SQL Server 的数据库对象包括表、_____、索引、存储过程、触发器等。

3.12 SQL Server 的物理数据库架构包括页和区、_____、数据库文件组等。

3.13 SQL Server 数据库的每个页的大小是 8KB，每个区的大小是_____。

3.14 SQL Server 使用的数据库文件有主数据文件、辅助数据文件、_____三类。

3.15 表结构包含一组固定的列，列由列名、_____、长度、允许 Null 值等组成。

3.16 空值通常表示未知、_____或将在以后添加的数据。

3.17 创建表之前,首先要确定表名和表的属性,表所包含的_____、列的数据类型、长度、是否为空、是否主键等,然后进行表结构设计。

3.18 整数型包括 bigint、int、smallint 和_____四类。

3.19 字符型包括固定长度字符数据类型和_____两类。

3.20 Unicode 字符型用于支持国际上_____的字符数据的存储和处理。

三、问答题

3.21 SQL Server 有哪些数据库对象?

3.22 SQL Server 数据库中包含哪几种文件?

3.23 简述使用图形用户界面创建 SQL Server 数据库的步骤。

3.24 使用 T-SQL 语句创建数据库包含哪些语句?

3.25 什么是表?什么是表结构?

3.26 简述 SQL Server 常用数据类型。

3.27 分别写出 student、course、score 的表结构。

3.28 可以使用哪些方式创建数据表?

3.29 简述以命令方式创建 SQL Server 表的语句。

四、应用题

3.30 使用图形用户界面创建 mydb 数据库,主数据文件为 mydb.mdf,初始大小为 15MB,增量为 15%,最大文件 150MB,日志文件为 mydb_log.ldf,初始大小为 1MB,增量 8%,增长无限制。

3.31 使用 T-SQL 语句创建 mydb1 数据库,主数据文件的初始大小、增量、增长和日志文件初始大小、增量、增长与上题相同。

3.32 在 stsco 数据库中,以命令方式分别创建 student 表、course 表、score 表、teacher 表和 lecture 表,表结构参见附录 B。

3.33 在 stsco 数据库中,以图形界面方式分别创建 student1 表、course1 表、score1 表、teacher1 表和 lecture1 表,表结构参见附录 B。

实验 3 创建数据库和创建表

实验 3.1 创建数据库

1. 实验目的及要求

(1) 理解 SQL Server 数据库的基本概念。

(2) 掌握使用 T-SQL 语句创建数据库、修改数据库、删除数据库的命令和方法,具备编写和调试创建数据库、修改数据库、删除数据库的代码的能力。

2. 验证性实验

使用 T-SQL 语句创建商店实验数据库 storepm,数据库 storepm 在实验中多次用到,其主数据文件为 storepm.mdf,初始大小为 15MB,增量为 1MB,增长无限制,日志文件为 storepm_log.ldf,初始大小为 1MB,增量 10%,增长无限制。

(1) 创建数据库 storepm。

```
CREATE DATABASE storepm
   ON
   (
      NAME = 'storepm',
      FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\storepm.
mdf',
      SIZE = 15MB,
      MAXSIZE = UNLIMITED,
      FILEGROWTH = 1MB
   )
   LOG ON
   (
      NAME = 'storepm_log',
      FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\storepm_
log.ldf',
      SIZE = 1MB,
      MAXSIZE = UNLIMITED,
      FILEGROWTH = 10 %
   )
```

(2) 修改数据库 storepm，首先增加数据文件 storepmbk.ndf，再删除数据文件 storepmbk.ndf。

```
ALTER DATABASE storepm
   ADD FILE
   (
      NAME = 'storepmadd',
      FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\
storepmadd.ndf',
      SIZE = 10MB,
      MAXSIZE = 150MB,
      FILEGROWTH = 2MB
   )

ALTER DATABASE storepm
   REMOVE FILE storepmadd
```

(3) 删除数据库 storepm。

```
DROP DATABASE storepm
```

3. 设计性实验

使用 T-SQL 语句创建图书借阅实验数据库 librarypm，主数据文件为 librarypm.mdf，初始大小为 10MB，增量为 10%，最大文件 200MB，日志文件为 librarypm_log.ldf，初始大小为 2MB，增量 1MB，最大文件 50MB。

(1) 创建数据库 librarypm。

(2) 修改数据库 librarypm，首先增加数据文件 librarypmbk.ndf 和日志文件 librarypmbk_log.ldf，再删除数据文件 librarypmbk.ndf 和日志文件 librarypmbk_log.ldf。

(3) 删除数据库 librarypm。

4. 观察与思考

(1) 在数据库 storepm 已存在的情况下,使用 CREATE DATABASE 语句创建数据库 storepm,查看错误信息。怎样避免数据库已存在又创建的错误?

(2) 能够删除系统数据库吗?

实验 3.2 创建表

1. 实验目的及要求

(1) 理解数据定义语言的概念和 CREATE TABLE 语句、ALTER TABLE 语句、DROP TABLE 语句的语法格式。

(2) 理解表的基本概念。

(3) 掌握使用数据定义语言创建表的操作,具备编写和调试创建表、修改表、删除表的代码的能力。

2. 验证性实验

商店实验数据库 storepm 是实验中多次用到的数据库,包含部门表 DeptInfo、员工表 EmplInfo、订单表 OrderInfo、订单明细表 DetailInfo 和商品表 GoodsInfo,它们的表结构分别如表 3.4～表 3.8 所示。

表 3.4 DeptInfo 表的表结构

列 名	数据类型	允许 NULL 值	是否主键	说 明
DeptNo	varchar(4)		主键	部门号
DeptName	varchar(20)			部门名称

表 3.5 EmplInfo 表的表结构

列 名	数据类型	允许 NULL 值	是否主键	说 明
EmplNo	varchar(4)		主键	员工号
EmplName	varchar(8)			姓名
Sex	varchar(2)			性别
Birthday	date			出生日期
Native	varchar(20)	√		籍贯
Wages	decimal(8,2)			工资
DeptNo	varchar(4)	√		部门号

表 3.6 OrderInfo 表的表结构

列 名	数据类型	允许 NULL 值	是否主键	说 明
OrderNo	varchar(6)		主键	订单号
EmplNo	varchar(4)	√		员工号
CurstomerNo	varchar(4)	√		客户号
SaleDate	date			销售日期
Cost	decimal(10,2)			总金额

表 3.7　DetailInfo 表的表结构

列　　名	数 据 类 型	允许 NULL 值	是否主键	说　　明
OrderNo	varchar(6)		主键	订单号
GoodsNo	varchar(4)		主键	商品号
SaleUnitPrice	decimal(8,2)			销售单价
Quantity	int			销售数量
Total	decimal(10,2)			总价
Discount	float			折扣率
DiscountTotal	decimal(10,2)			折扣总价

表 3.8　GoodsInfo 表的表结构

列　　名	数 据 类 型	允许 NULL 值	是否主键	说　　明
GoodsNo	varchar(4)		主键	商品号
GoodsName	varchar(30)			商品名称
ClassificationName	varchar(30)			商品类型
UnitPrice	decimal(8,2)	√		单价
StockQuantity	int			库存量

在数据库 storepm 中，验证和调试创建表、修改表、删除表的代码。

(1) 创建 EmplInfo 表。

```
CREATE TABLE EmplInfo
   (
      EmplNo varchar(4) NOT NULL PRIMARY KEY,
      EmplName varchar(8) NOT NULL,
      Sex varchar(2) NOT NULL,
      Birthday date NOT NULL,
      Native varchar(20) NULL,
      Wages decimal(8,2) NOT NULL,
      DeptNO varchar(4) NULL
   )
```

(2) 由 EmplInfo 表创建 EmplInfo1 表。

```
SELECT EmplNO,EmplName,Sex,Birthday,Native,Wages,DeptNO INTO EmplInfo1
FROM EmplInfo
```

(3) 在 EmplInfo 表中增加一列 Eno，不为空。

```
ALTER TABLE EmplInfo
ADD Eno varchar(4) NOT NULL
```

(4) 将 EmplInfo1 表的列 Sex 的数据类型改为 char，可空。

```
ALTER TABLE EmplInfo1
ALTER COLUMN Sex char(2) NULL
```

(5) 在 EmplInfo 表中删除列 Eno。

```
ALTER TABLE EmplInfo
DROP COLUMN Eno
```

(6) 删除 EmplInfo1 表。

```
DROP TABLE EmplInfo1
```

3. 设计性试验

在数据库 storepm 中,设计、编写和调试创建表、修改表、删除表的代码。

(1) 创建 GoodsInfo 表。

(2) 由 GoodsInfo 表创建 GoodsInfo1 表。

(3) 在 GoodsInfo 表中增加一列 Gno,不为空。

(4) 将 GoodsInfo1 表的列 UnitPrice 的数据类型改为 money。

(5) 在 GoodsInfo 表中删除列 Gno。

(6) 删除 GoodsInfo1 表。

4. 观察与思考

(1) 在创建表的语句中,NOT NULL 的作用是什么?

(2) 一个表可以设置几个主键?

(3) 主键列能否修改为 NULL?