

任务

1



分销系统数据库的设计与生成

面对纷繁复杂的海量信息，如何对其进行有效的管理和利用？数据库技术应运而生，并发展成为一门综合性数据管理技术。由 Microsoft 发布的 SQL Server 产品是一款典型的关系型数据库管理系统，应用越来越广泛。

一、任务目标

1. 掌握数据库、数据库管理系统和数据库系统的基本概念。
2. 了解数据模型的概念、掌握管理数据结构，掌握关系数据库的定义。
3. 了解并基本掌握 E-R 实体联系图。
4. 掌握 SQL Server 数据库的基本组成和有关知识。
5. 初步了解 Transact-SQL 语句的基础知识。
6. 掌握数据库的创建。

二、教学任务

1. 介绍数据库的基本概念。
2. 介绍数据库系统模型。
3. 介绍关系模型的完整性约束。
4. 介绍关系型数据库范式理论。
5. 设计分销系统数据库。
6. 介绍 SQL Server 数据库的常用对象、数据库结构和系统数据库等基本知识。
7. SQL Server Management Studio 的初步使用。
8. 用图形化工具创建数据库。
9. 使用 Transact-SQL 创建数据库。

1.1 分销系统数据库的规划设计

分销系统是对营销业务中的客户资料、销售订单、供应商资料、采购订单、付款单、收款单以及仓库物资等进行管理，设计良好的数据库能给系统的高效运作提供保障。

数据库系统设计包括数据模型设计以及围绕数据模型的应用程序开发两大部分，这里主要介绍数据模型设计，也就是设计一组二维表，定义这些表的列名、列的数据类型以及表的数据完整性约束规则。

数据库设计过程一般包括：需求分析、概念模型设计、逻辑设计、物理设计和实施维护。

1.1.1 分销系统数据库的需求分析

用户的需求具体体现了各种信息的提供、保存、更新和查询，这就要求数据库结构能充分满足各种信息的输入和输出。收集基本数据、数据结构以及数据处理的流程，组成一份详尽的数字字典，为后面的具体设计打下基础。

在分析调查了有关分销系统信息需求的基础上，将得到如图 1-1 所示的分销系统部分数据流程图。

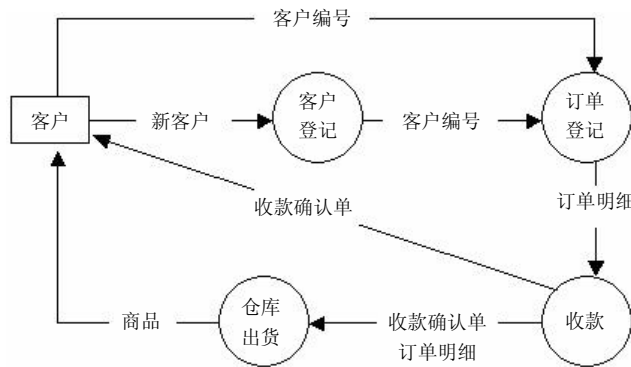


图 1-1 分销系统部分数据流程图

通过对分销系统的业务流程和数据流的分析，设计如下的数据项和数据结构。

客户资料：包括客户编号、客户名称、联系人、电话、传真、地址、送货地址和备注等数据项目。

销售订单：包括销售订单号、日期、客户编号、客户名称、联系人、联系电话、送货地址、总金额等数据项目。

销售订单明细：包括销售订单号、序号、商品编码、商品名称、规格型号、单位、数量、单价、金额和备注等数据项目。

收款单：收款单号、收款日期、收款人、客户编号、客户名称、应收总额、收款金额和备注等数据项目。

出库单：出库单号、日期、客户编号、客户名称、联系人、联系电话、送货地址、总金额等数据项目。

出库明细：出库单号、序号、销售订单号、商品编码、规格型号、单位、数量、单价、金额等数据项目。

1.1.2 分销系统数据库的概念模型设计

根据上面的设计规划出的实体有客户实体、销售订单实体、销售订单明细实体、收款单实体、出库单实体、出库明细实体等。部分实体具体的 E-R 图描述如图 1-2 至图 1-4 所示。

对于其他实体 E-R 图类似的可以很容易画出，实体与实体之间的关系也不难得出。

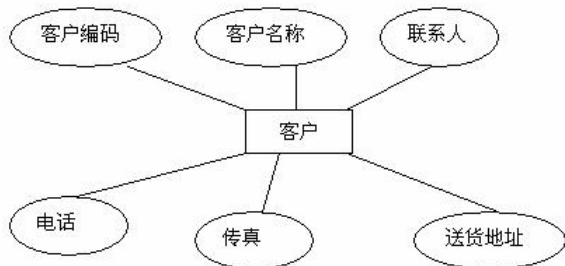


图 1-2 客户实体 E-R 图

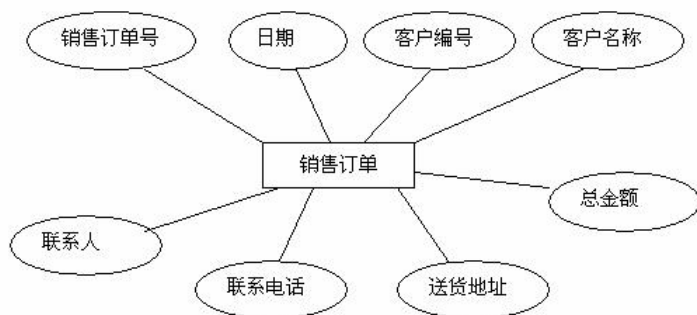


图 1-3 销售订单实体 E-R 图

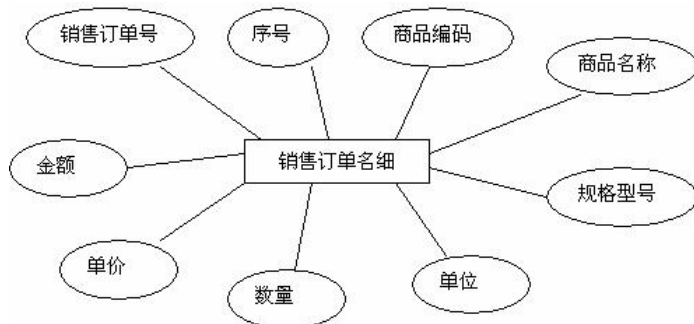


图 1-4 销售订单明细实体 E-R 图

1.1.3 分销系统数据库的逻辑设计

现在需要把数据库概念结构转化为 SQL Server 数据库系统所支持的实际数据模型也就是数据库的逻辑结构。

分析逻辑概念中的实体与实体之间的关系，再形成数据库中表与表之间的关系。

分销系统的部分表设计结果见表 1-1 至表 1-4，每个表格表示数据库中的一个表。

表 1-1 客户资料

字段名称	数据类型	是否允许为空	是否为主键	备注
客户编码	varchar(20)	否	是	
客户名称	varchar(100)	否	否	
联系人	varchar(20)	否	否	
电话	varchar(50)	否	否	
传真	varchar(50)	是	否	
地址	varchar(200)	是	否	
送货地址	varchar(200)	是	否	
备注	varchar(500)	是	否	

表 1-2 销售订单

字段名称	数据类型	是否允许为空	是否为主键	备注
销售订单号	varchar(20)	否	是	
日期	datetime	否	否	
客户编码	varchar(20)	否	否	外键：客户资料（客户编码）
客户名称	varchar(100)	否	否	
联系人	varchar(20)	否	否	
联系电话	varchar(50)	否	否	
送货地址	varchar(200)	否	否	
总金额	numeric(12,4)	否	否	
备注	varchar(500)	是	否	

表 1-3 销售订单明细表

字段名称	数据类型	是否允许为空	是否为主键	备注
销售订单号	varchar(20)	否	组合主键	外键：销售订单（销售订单号）
序号	int	否		
商品编码	varchar(20)	否	否	外键：商品资料（商品编码）
商品名称	varchar(50)	否	否	
规格型号	varchar(100)	否	否	
单位	numeric(12,4)	否	否	
数量	numeric(12,4)	否	否	
单价	numeric(12,4)	否	否	
金额	numeric(12,4)	否	否	
备注	varchar(500)	是	否	

表 1-4 收款单

字段名称	数据类型	是否允许为空	是否为主键	备注
收款单号	varchar(20)	否	是	
收款日期	datetime	否	否	
收款人	varchar(100)	否	否	
客户编码	varchar(20)	是	否	外键: 客户资料(客户编码)
客户名称	varchar(100)	是	否	
应收总额	numeric(12,2)	是	否	
收款金额	numeric(12,2)	否	否	
备注	varchar(500)	是	否	

1.2 分销系统数据库的创建

数据库主要是存储了数据的表的集合以及其他的数据库对象。

SQL Server 可以同时支持许多数据库,每一个数据库既可以存储与另一个数据库相关的数据,也可以存储不相关的数据。

1.2.1 SQL Server 数据库基本知识

1. 数据库对象

SQL Server 数据库中的数据在逻辑上被组织成一系列对象,当一个用户连接到数据库后,他所看到的是这些逻辑对象,而不是物理的数据库文件。

SQL Server 中有以下数据库对象:关系图、表、视图、存储过程、规则、默认、用户自定义的数据类型、用户自定义函数等。

2. 标识符命名规则

SQL Server 中,常规标识符命名规则如下:

(1) 标识符长度是 1~128 个字符。

(2) 第一个字符必须是大小写字母,以及来自其他语言的字母字符,也可以是汉字、下划线(_)、at 符号(@)或数字符号(#)。标识符以单个字符@开始,表示该标识为局部变量或参数,以两个字符@@开始,表示该标识为全局变量;标识符以单个字符#开始,表示该标识为临时数据库对象,以两个字符##开始,表示该标识为全局临时数据库对象。

(3) 后续字符可以是字母、基本拉丁字母、十进制数字、at 符号(@)、美元符号(\$)、数字符号(#)或下划线(_)。

(4) 标识符不能是 Transact-SQL 的保留字。

(5) 不允许嵌入空格或其他特殊字符。

3. 数据库结构

(1) 数据库文件。

SQL Server 用文件来存放数据库,数据库中的所有数据和对象,例如表、触发器和视图,都存储在数据库文件中,数据库文件有 3 类。

- 主数据文件 (Primary File): 用来存放数据, 该文件是数据库的起点, 每个数据库都必须有且只有一个主数据文件。
- 数据文件 (Secondary File): 也用来存放数据, 这些次数据文件是可选的, 它们可以存储那些不在主数据文件中的全部数据和对象。一个数据库可以没有也可以有多个次数据文件。
- 事务日志文件 (Transaction Log File): 存放事务日志, 这些事务日志文件保存了用于恢复数据库的全部事务日志信息。每个数据库都必须有一个或多个日志文件。

一般情况下, 一个简单的数据库可以只有一个主数据文件和一个事务日志文件。如果数据库很大, 则可以设置多个次数据文件和事务日志文件, 并将它们放在不同的磁盘上。

默认情况下, 数据库文件存放在\MSSQL\DATA\目录下, 数据文件名为“数据库名_Data.MDF”, 日志文件名为“数据库名_Log.LDF”。数据库的创建者可以在创建时指定其他的路径和文件名, 也可以添加次数据文件和更多的事务日志文件。

(2) 数据库文件组。

文件组就是文件的集合。为了管理和分配数据的方便, 文件组允许多个数据库文件组成一个组, 对它们整体进行管理。比如, 可以将 3 个数据文件 (data1.mdf、data2.mdf 和 data3.mdf) 分别创建在 3 个盘上, 将这 3 个文件组成文件组 fgroup1, 在创建表的时候就可以指定一个表创建在文件组 fgroup1 上。这样该表的数据就可以分布在 3 个盘上, 在对该表执行查询时, 可以并行操作, 大大提高了查询效果。

(3) 文件和文件组规则。

SQL Server 的数据库文件和文件组必须遵循以下规则:

- 一个文件或文件组只能被一个数据库使用, 不能用于多个数据库。
- 一个文件只能属于一个文件组。
- 一个数据库的数据信息和日志信息不能放在同一个文件或者文件组中, 数据文件和日志文件总是分开的。
- 日志文件永远也不能是任何文件组的一部分, 日志文件总是分开的。

(4) 数据库空间管理。

SQL Server 可管理的最小空间是以页为单位的, 每一页的大小是 8KB, 即 8192 字节。在表中, 每一行数据不能跨页存储。这样, 表中每一行的字节数不能超过 8192 字节。每 8 个连续页称为一个簇, 即簇的大小是 64KB。每个表或者索引最少要占一个簇的空间, 也就是说每一个表或者索引最小也是 64KB。

4. 事务日志

每个数据库都有一个相关的事务日志, 事务日志记录了 SQL Server 所有的事务和由这些事务引起的数据库的变化, 即事务日志记录了对数据库的所有修改操作。在数据库中数据的任何改变写到磁盘之前, 这个改变首先在事务日志中做了记录。

事务日志记录了每一个事务的开始、对数据的改变和取消修改的足够信息, 随着对数据库的操作, 日志是连续增加的。SQL Server 使用数据库的事务日志来恢复事务。所以事务日志可以具有以下 3 个作用。

(1) 恢复单个事务。

当执行 ROLLBACK 语句, 或 SQL Server 发现错误时, 回滚未完成的事务所做的修改。

(2) 在 SQL Server 启动时恢复所有未完成的事务。

当 SQL Server 出错停机时, 事务的改变可能一部分被写入数据库文件, 而另一部分还没有被写入, 这样就造成了数据库中数据的不一致。事务日志可以使 SQL Server 在重新启动时回滚所有未完成的事务, 以保证数据库的一致状态。

(3) 恢复数据库时, 将数据库向前滚动到出错前一秒的状态

数据库从全库备份或差异备份恢复后, 利用事务日志可以回滚所有未完成的事务, 使数据库恢复到出错前一秒的状态。

5. 系统数据库

打开 SQL Server Management Studio, 在创建任何数据库之前, 展开对象资源管理器的“数据库”节点, 可以看到已经有了“系统数据库”节点, 该节点下有 4 个数据库。它们是 SQL Server 的系统数据库。与用户数据库不同, 系统数据库是在安装 SQL Server 时由安装程序自动创建的。

SQL Server 有 4 个系统数据库, 它们分别是: master 数据库、tempdb 数据库、model 数据库和 msdb 数据库。

(1) master 数据库。

它记录了 SQL Server 系统级的信息, 包括系统中所有的登录账户、系统配置信息、所有数据库信息、所有用户数据库的主文件地址等, 这些信息都记录在 master 数据库的表中。为了与用户创建的表相区别称为系统表, 表名都以 sys 开头。

master 数据库中还有很多系统存储过程和扩展存储过程。

(2) tempdb 数据库。

用于存放所有连接到系统的用户的临时表和临时存储过程, 以及 SQL Server 产生的其他临时性的对象。tempdb 是 SQL Server 中负担最重的数据库, 因为几乎所有的查询都需要使用它。

在 SQL Server 关闭时 tempdb 数据库中的所有对象都被删除, 每次启动 SQL Server 时都会重新创建 tempdb 数据库。

tempdb 数据库可以按照需要自动增长, 每次系统启动时, tempdb 数据库都被重置为默认的大小 (8MB)。在以后的工作中, 当 tempdb 数据库空间不够时, 系统都将自动扩展 tempdb 数据库的大小。

(3) model 数据库。

model 数据库是系统所有数据库的模板, 这个数据库相当于一个模子, 所有在系统中创建的新数据库的内容, 在刚创建时都和 model 数据库完全一样。

刚刚完成 SQL Server 安装时, model 数据库就已经有了 18 个表以及一些视图和存储过程, 因此用户创建的每个数据库中都将有这些对象。这 18 个表是另一类的系统表, 它们的表名也是以 sys 开头的, 其内容是有关数据库的结构等重要信息。

(4) msdb 数据库。

SQL Server 代理 (SQL Server Agent) 使用 msdb 数据库来计划警报和作业, 并记录操作员备份和还原历史记录。

6. 估算数据库的空间需求

数据库管理员的主要任务之一就是创建数据库, 并且需要为每个文件指定容量。必须尽

可能准确地估算数据库容量，以免浪费磁盘空间资源或者因估计不足造成数据库的空间不够。

许多因素会影响数据库最终的大小，在估算数据库容量时要考虑如下因素：

- 每行记录的大小。
- 记录数量。
- 表的数量。
- 索引的数量及索引大小。
- 数据库的对象的数量和大小。
- 事务日志的大小。
- 数据库的计划增加量。

7. 确定数据库的数目

当一个企业或部门确定要建立数据库系统之后，接着就要确定这个数据库系统与企业中其他部分的关系。因此，需要分析企业的基本业务功能，确定数据库支持的业务范围，是建立一个综合的数据库，还是建立若干个专门的数据库。

从理论上讲，可以建立一个支持企业全部活动的包罗万象的大型综合数据库，也可以建立若干个支持范围不同的公用或专用数据库。

一般来讲，前者难度较大，效率也不高；后者比较分散，但相对灵巧，必要时可通过联接操作将有关数据联接起来，而数据的全局共享一般可利用建立在数据库上的应用系统来实现。在各种规模的企业中，同时在不同部门内维护众多中小型数据库已成为一种常见现象。通常情况下，这些数据库或服务器被置于 IT 部门负责日常维护管理的系统范畴之外，因此，它们无法达到企业在设计、实现及维护方面所制订的 IT 标准。

1.2.2 使用 SQL Server Management Studio 创建数据库

使用数据库存储数据，首先要创建数据库。前面已介绍过，一个数据库必须至少包含一个主数据文件和一个事务日志文件。所以创建数据库就是创建主数据文件和事务日志文件。在 SQL Server 中，可以使用 SQL Server Management Studio 创建数据库，也可以使用 Transact-SQL 创建数据库。

任务 1-1：使用 SQL Server Management Studio 创建分销系统数据库。

具体步骤如下：

(1) 启动 SQL Server Management Studio，登录服务器类型为“数据库引擎”，并使用 Windows 或 SQL Server 身份验证建立连接。

(2) 连接成功后，在对象资源管理器中，右击“数据库”节点，从弹出的快捷菜单中选择“新建数据库”命令，打开“新建数据库”窗口，如图 1-5 所示。

(3) 默认显示的是“常规”页面。首先，在“数据库名称”文本框中输入数据库名称“分销系统”。再输入数据库所有者，用户可以使用“默认值”，也可以应通过单击文本框右边的“浏览”按钮选择所有者。

(4) 在下面的“数据库文件”列表中列出了数据库包含的主数据文件和事务日志文件的逻辑名称、文件类型、文件组、初始大小、自动增长和路径等。用户可以自行设置初始大小、自动增长值和数据库存放的位置。

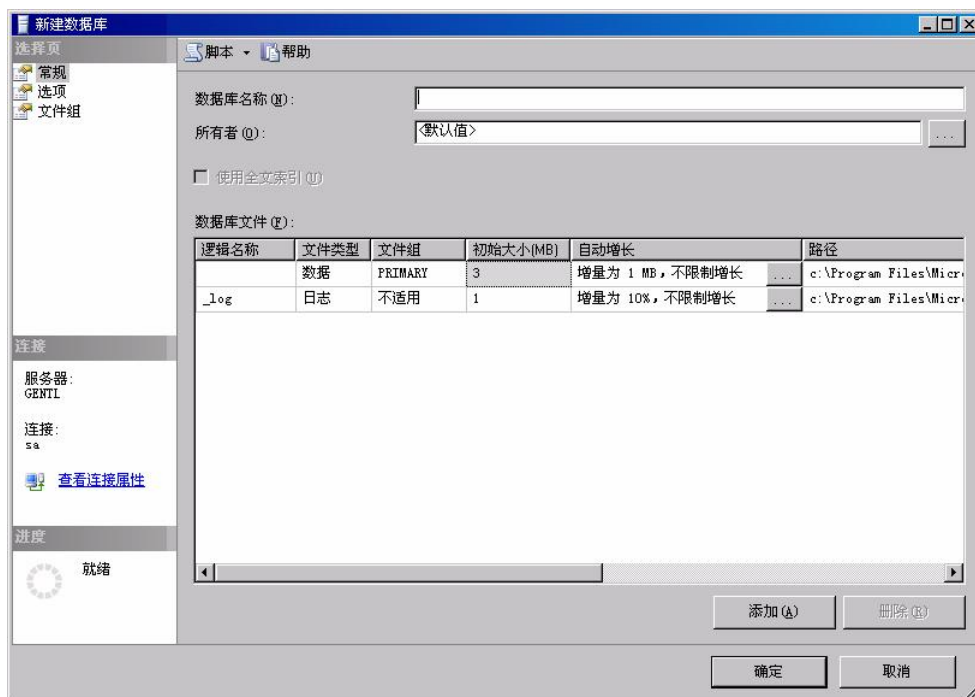


图 1-5 “新建数据库”对话框

(5) 在“选项”页面中可以定义所创建数据库的排序规则、恢复模式、兼容级别、恢复、游标等其他选项。

(6) 在“文件组”页面中可以查看数据库中的所有文件组，包括主文件组和次文件组。还可以通过单击“添加”或“删除”按钮来添加或删除次文件组。

(7) 全部设置完毕后，单击“确定”按钮即可完成分销系统数据库的创建。在对象资源管理器中右击“数据库”节点，从弹出的快捷菜单中选择“刷新”命令。展开“数据库”节点，可看到创建好的“分销系统”数据库正在列表中。

1.2.3 Transact-SQL 创建数据库

在 SQL Server 中，可以使用 Create Database 语句来创建数据库，语法格式如下：

```
CREATE DATABASE database_name
    [ON
      {[PRIMARY](NAME = logical_file_name,
        FILENAME = 'os_file_name'
        [,SIZE = size]
        [,MAXSIZE = max_size]
        [,FILEGROWTH = groth_increment])
      }[,...n]
    ]
    FILEGROUP filegroup_name < filespec > [,...n ]
    [LOG ON
      { (NAME = logical_file_name,
        FILENAME = 'os_file_name'
```

```

[SIZE = size]
[.MAXSIZE = max_size]
[.FILEGROWTH = growth_increment])
}[,...n]
]
[FOR RESTORE]

```

下面详细介绍一些各选项的含义。

(1) **PRIMARY**: 该选项是一个关键字, 用来指定主文件组中的主文件。

主文件组中不仅包含了数据库系统表中的全部内容, 而且还包含了没有在用户文件组中包含的全部对象。一个数据库只能有一个主文件。在默认情况下, 即在指定 **PRIMARY** 关键字时, 列在语句中的第一个文件就是主文件。

(2) **NAME**: 该选项用来指定数据库的逻辑名称。

这是在 SQL Server 系统中使用的名称, 是数据库在 SQL Server 中的标识符。

(3) **FILENAME**: 该选项用来指定数据库所在文件的操作系统文件名称和路径。

在 `os_file_name` 中的路径必须是 SQL Server 所在服务器上的一个文件夹。该操作系统文件名与 **NAME** 的逻辑名称是一一对应的。

(4) **SIZE**: 该选项用来指定数据库操作系统文件的大小。

在指定文件大小的时候既可以用 **MB** 单位, 也可以使用 **KB** 单位。如果没有指定单位, 系统默认的单位是 **MB**。文件最小是 **1MB**, 也就是说, 数据库所在的文件不能小于 **1MB**。在默认情况下, 数据库数据文件的大小是 **1MB**, 数据库日志文件的大小也是 **1MB**。

(5) **MAXSIZE**: 该选项用来指定操作系统文件可以增长的最大尺寸。

在指定文件增长尺寸时, 既可以使用 **MB** 单位, 也可以使用 **KB** 单位。如果没有指定单位, 系统默认的单位是 **MB**。如果没有指定文件可以增长的最大尺寸, 系统的增长是没有限制的, 可以占满整个磁盘空间。

(6) **FILEGROWTH**: 该选项用来指定文件的增量。当然该选项不能与 **MAXSIZE** 选项有冲突。该选项指定的数据值为 **0** 时, 表示文件不能增长。该选项可以用 **MB**、**KB** 和百分比指定。

任务 1-2: 创建一个数据库, 名称为 `student`。

```
CREATE DATABASE student
```

任务 1-3: 创建一个数据库 `CUSTOMER`, 该数据库的主数据文件的逻辑名称是 `CUSTOMER_DATA`, 操作系统文件是 `CUSTOMER_DATA.MDF`, 大小是 **15MB**, 最大是 **30MB**, 以 **20%** 的速度增加; 该数据库的日志文件的逻辑名称是 `CUSTOMER_LOG`, 操作系统文件是 `CUSTOMER_LOG.LDF`, 大小是 **3MB**, 最大是 **10MB**, 以 **1MB** 的速度增加。

注: 以下代码的前提是硬盘上存在目录 `E:\yxl\`, 或者将 `E:\yxl\` 改成一个已经存在的目录, 否则将不能成功执行。

```

CREATE DATABASE CUSTOMER
ON
    PRIMARY (NAME = customer_data,
    FILENAME='e:\yxl\customer_data.mdf',
    SIZE = 15MB,
    MAXSIZE = 30MB,

```

```
FILEGROWTH=20%)  
LOG ON  
(NAME = customer_log,  
FILENAME = 'e:\yx1\customer_log.ldf',  
SIZE = 3MB,  
MAXSIZE = 10MB,  
FILEGROWTH = 1MB)
```

1.2.4 Transact-SQL 删除数据库

删除数据库的语法如下：

```
DROP DATABASE database_name
```

任务 1-4：删除任务 1-2 建立的 student 数据库。

```
DROP DATABASE student
```

任务 1-5：删除任务 1-3 建立的 CUSTOMER 数据库。

```
DROP DATABASE CUSTOMER
```