

# 第 1 部分 SQL Server 2008 基础

## 第 1 章 数据库基础知识

为了更好地学习 SQL Server，首先需要介绍一下数据库的基本概念，如果学习过数据库原理，那么本章数据库原理部分仅仅作为一个参考。

### 1.1 数据库基本概念

#### 1.1.1 数据库与数据库管理系统



##### 1. 数据库

数据库 (DB) 是存放数据的仓库，而且这些数据存在一定的关联，并按一定的格式存放在计算机上。从广义上讲，数据不仅包含数字，还包括文本、图像、音频、视频等。

例如，把一个学校的学生、课程、学生成绩等数据有序地组织并存放在计算机内，就可以构成一个数据库。因此，数据库由一些持久的相互关联的数据的集合组成，并以一定的组织形式存放在计算机的存储介质中。

##### 2. 数据库管理系统

数据库管理系统 (DBMS) 是管理数据库的系统，它按一定的数据模型组织数据。DBMS 应提供如下功能：

- (1) 数据定义功能：可定义数据库中的数据对象。
- (2) 数据操纵功能：可对数据库表进行基本操作，如插入、删除、修改、查询等。
- (3) 数据的完整性检查功能：保证用户输入的数据满足相应的约束条件。
- (4) 数据库的安全保护功能：保证只有具有权限的用户才能访问数据库中的数据。
- (5) 数据库的并发控制功能：使多个应用程序可在同一时刻并发地访问数据库的数据。
- (6) 数据库系统的故障恢复功能：使数据库在运行出现故障时进行数据库恢复，以保证数据库可靠运行。
- (7) 在网络环境下访问数据库的功能。

(8) 方便、有效地存取数据库信息的接口和工具。编程人员通过程序开发工具与数据库的接口编写数据库应用程序。数据库系统管理员 (DBA, DataBase Administrator) 通过提供的工具对数据库进行管理。

数据、数据库、数据库管理系统与操作数据库的应用程序，加上支撑它们的硬件平台、软件平台以及与数据库有关的人员一起构成了一个完整的数据库系统。如图 1.1 所示描述了数据库系统的构成。

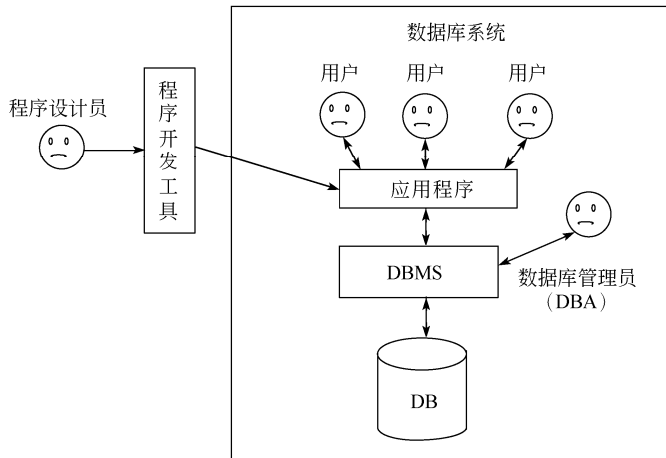


图 1.1 数据库系统的构成

## 1.1.2 数据模型和关系数据库

### 1. 主要数据模型

数据库管理系统根据数据模型对数据进行存储和管理，数据库管理系统采用的数据模型主要有层次模型、网状模型和关系模型。

(1) 层次模型：以树状层次结构组织数据。如图 1.2 所示为某学校按层次模型组织的数据示例。

(2) 网状模型：每一个数据用一个节点表示，每个节点与其他节点都有联系，这样，数据库中的所有数据节点就构成了一个复杂的网络。如图 1.3 所示为按网状模型组织的数据示例。

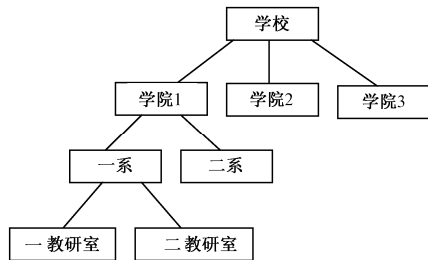


图 1.2 按层次模型组织的数据示例

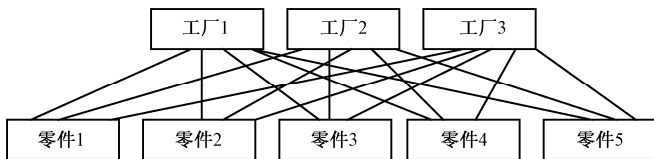


图 1.3 按网状模型组织的数据示例

(3) 关系模型：以二维表格（关系表）的形式组织数据库中的数据。例如，学生成绩管理系统所涉及的“学生”“课程”和“成绩”三个表中，“学生”表涉及的主要信息有：学号、姓名、性别、出生时间、专业、总学分、备注；“课程”表涉及的主要信息有：课程号、课程名、开课学期、学时和学分；“成绩”表涉及的主要信息有：学号、课程号和成绩。如表 1.1、表 1.2 和表 1.3 所示分别描述了学生成绩管理系统中“学生”“课程”和“成绩”三个表的部分数据。



表 1.1 “学生”表

学号	姓名	性别	出生时间	专业	总学分	备注
081101	王林	男	1990-02-10	计算机	50	
081103	王燕	女	1989-10-06	计算机	50	

续表

学号	姓名	性别	出生时间	专业	总学分	备注
081108	林一帆	男	1989-08-05	计算机	52	已提前修完一门课
081202	王林	男	1989-01-29	通信工程	40	有一门课不及格，待补考
081204	马琳琳	女	1989-02-10	通信工程	42	

表 1.2 “课程”表

课程号	课程名	开课学期	学时	学分
0101	计算机基础	1	80	5
0102	程序设计与语言	2	68	4
0206	离散数学	4	68	4

表 1.3 “成绩”表

学号	课程号	成绩	学号	课程号	成绩
081101	101	80	081108	101	85
081101	102	78	081108	102	64
081101	206	76	081108	206	87
081103	101	62	081202	101	65
081103	102	70	081204	101	91

表格中的一行称为一个记录，一列称为一个字段，每列的标题称为字段名。如果给每个关系表取一个名字，则有  $n$  个字段的关系表的结构可表示为：关系表名（字段名 1，…，字段名  $n$ ），通常把关系表的结构称为关系模式。

在关系表中，如果一个字段或几个字段组合的值可唯一标识其对应记录，则称该字段或字段组合为码。例如，表 1.1 中的“学号”可唯一标识每一个学生，表 1.2 中的“课程号”可唯一标识每一门课程，表 1.3 中的“学号”和“课程号”组合可唯一标识每一个学生一门课程的成绩。

有时一个表可能有多个码，例如表 1.1 中，姓名不允许重名，则“学号”“姓名”均是学生信息表的码。对于每一个关系表，通常可指定一个码为“主码”，在关系模式中一般用下横线标出主码。

设表 1.1 的名字为 XSB，关系模式可分别表示为：XSB（学号，姓名，性别，出生时间，专业，总学分，备注）。

设表 1.2 的名字为 KCB，关系模式可分别表示为：KCB（课程号，课程名，开课学期，学时，学分）。

设表 1.3 的名字为 CJB，关系模式可分别表示为：CJB（学号，课程号，成绩）。

## 2. 其他数据模型

随着信息管理内容的不断扩展和新技术的层出不穷，数据库技术面临着前所未有的挑战。面对新的数据形式，除了层次模型、网状模型、关系模型，人们提出了丰富多样的数据模型，例如面向对象模型、半结构化模型等，同时也提出了众多新的数据库技术（XML 数据管理、数据流管理、Web 数据集成、数据挖掘等）。

## 3. 流行关系数据库产品

关系数据库分为两类：一类是桌面数据库，另一类是客户/服务器数据库。

一般而言，桌面数据库用于小型的、单机的应用程序，它不需要网络和服务器，实现起来比较方便，但它只提供数据的存取功能，例如 Access、FoxPro、Excel 和 dBase 等。

客户/服务器数据库主要适用于大型的、多用户的数据库管理系统，包括两部分：一部分驻留

在客户机上，用于向用户显示信息及实现与用户的交互；另一部分驻留在服务器中，主要用来实现对数据库的操作和对数据的计算处理。例如，大型关系型数据库管理系统一般为 Oracle、SQL Server、DB2、Informix 和 Sybase 等，小型关系型数据库管理系统一般为 MySQL、PostgreSQL 和 SQLite。其中 SQLite 是一个强大的嵌入式关系型数据库管理系统；MySQL 是最流行的 RDBMS；PostgreSQL 是比较先进的 SQL 型开源 objective-RDBMS。

在开发数据库应用程序时，也可以将它们放在一台计算机上进行调试，调试完成再将数据库放到服务器上。



### 1.1.3 关系型数据库语言

SQL (Structured Query Language, 结构化查询语言) 是用于关系数据库查询的结构化语言。SQL 的功能包括数据查询、数据操纵、数据定义和数据控制 4 部分。

## 1.2 数据库设计



### 1.2.1 概念结构设计

通常，把每一类数据对象的个体称为“实体”，而每一类对象个体的集合称为“实体集”，因此，在学生成绩管理系统中主要涉及“学生”和“课程”两个实体集。

其他非主要的实体可以很多，如班级、班长、任课教师、辅导员等实体。把每个实体集涉及的信息项称为属性。就“学生”实体集而言，它的属性有学号、姓名、性别、出生时间、专业、总学分和备注；“课程”实体集属性有课程号、课程名、开课学期、学时和学分。

实体集中的实体彼此是可区别的。如果实体集中的属性或最小属性组合的值能唯一标识其对应实体，则将该属性或属性组合称为码。对于每一个实体集，可指定一个码为主码。

如果用矩形框表示实体集，用圆角矩形框表示属性，用线段连接实体集与属性，当一个属性或属性组合指定为主码时，在实体集与属性的连接线上标记一斜线，则可以用如图 1.4 所示形式描述学生成绩管理系统中的实体集及每个实体集涉及的属性。

实体集 A 和实体集 B 之间存在各种关系，把这些关系称为“联系”，通常使用菱形表示联系。将实体集及实体集联系的图表示称为实体 (Entity)-联系 (Relationship) 模型。

E-R 图就是 E-R 模型的描述方法，即实体-联系图。通常，关系数据库的设计者使用 E-R 图来对信息世界建模。从分析用户项目涉及的数据对象及数据对象之间的联系出发，到获取 E-R 图的这一过程称为概念结构设计。

两个实体集 A 和 B 之间的联系可能是以下三种情况之一。

#### 1. 一对一的联系 (1:1)

A 中的一个实体至多与 B 中的一个实体相联系，B 中的一个实体也至多与 A 中的一个实体相联系。例如，“班级”与“正班长”这两个实体集之间的联系是一对一的联系，因为一个班级只有一个正班长，反过来，一个正班长只属于一个班级。“班级”与“正班长”两个实体集的 E-R 模型如图 1.5 所示。

#### 2. 一对多的联系 (1:n)

A 中的一个实体可以与 B 中的多个实体相联系，而 B 中的一个实体至多与 A 中的一个实体相联系。例如，“班级”与“学生”这两个实体集之间的联系是一对多的联系，因为一个班级可有若干学生，

反过来，一个学生只能属于一个班级。“班级”与“学生”两个实体集的 E-R 模型如图 1.6 所示。

### 3. 多对多的联系 ( $m:n$ )

A 中的一个实体可以与 B 中的多个实体相联系，而 B 中的一个实体也可与 A 中的多个实体相联系。例如，“学生”与“课程”这两个实体集之间的联系是多对多的联系，因为一个学生可选多门课程，反过来，一门课程可被多个学生选修。“学生”与“课程”两个实体集的 E-R 模型如图 1.7 所示。

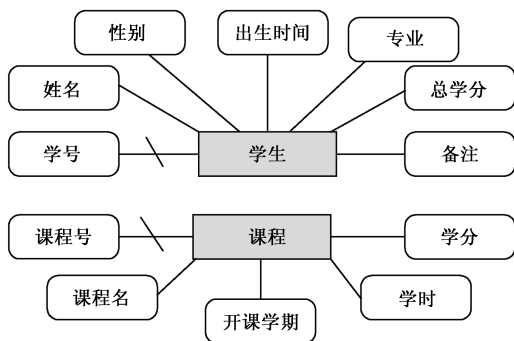


图 1.4 “学生”和“课程”实体集属性的描述

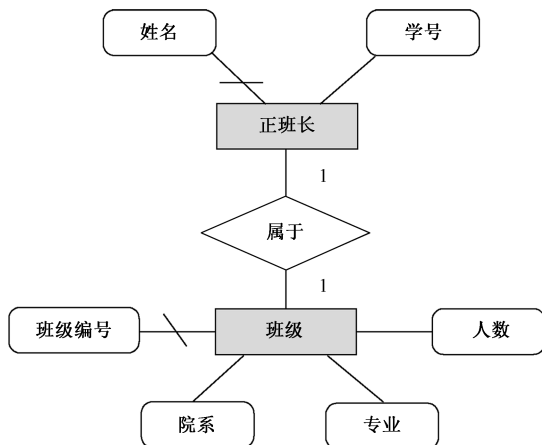


图 1.5 “班级”与“正班长”实体集的 E-R 模型

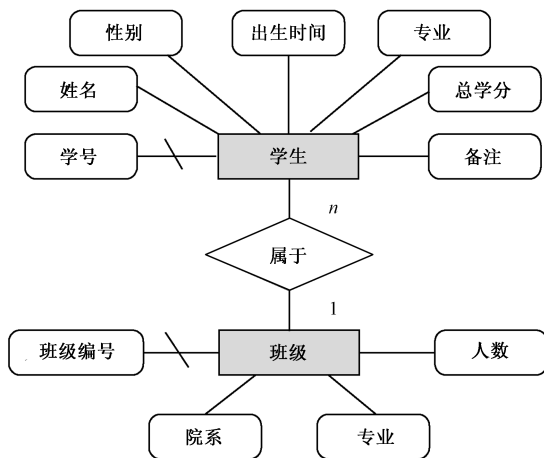


图 1.6 “班级”与“学生”两个实体集的 E-R 模型

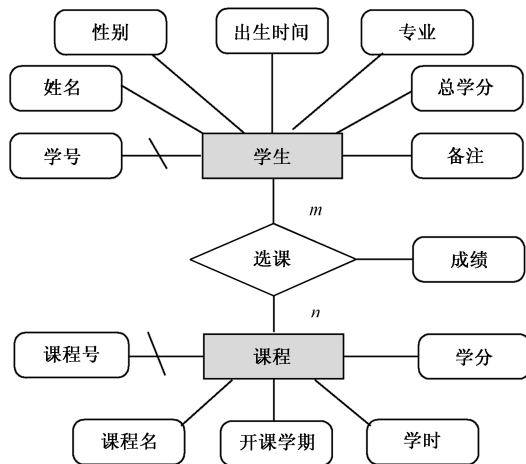


图 1.7 “学生”与“课程”实体集的 E-R 模型

## 1.2.2 逻辑结构设计

用 E-R 图描述学生成绩管理系统中实体集与实体集之间的联系，目的是以 E-R 图为工具，设计关系型的数据库，即确定应用系统所使用的数据库应包含哪些表？每个表的结构是怎样的？前面已介绍了实体集之间的联系，下面将介绍根据三种联系从 E-R 图获得关系模式的方法。

### 1. (1:1) 联系的 E-R 图到关系模式的转换

对于 (1:1) 的联系，既可单独对应一个关系模式，也可以不单独对应一个关系模式。

(1) 联系单独对应一个关系模式，则由联系属性、参与联系的各实体集的主码属性构成关系模式，其主码可选参与联系的实体集的任一方的主码。



例如, 考虑图 1.5 描述的“班级 (BJB)”与“正班长 (BZB)”实体集通过属于 (SYB) 联系 E-R 模型, 可设计如下关系模式 (下横线表示该字段为主码):

BJB (班级编号, 院系, 专业, 人数)

BZB (学号, 姓名)

SYB (学号, 班级编号)

(2) 联系不单独对应一个关系模式, 联系的属性及一方的主码加入另一方实体集对应的关系模式中。

例如, 考虑图 1.5 描述的“班级 (BJB)”与“正班长 (BZB)”实体集通过属于 (SYB) 联系 E-R 模型, 可设计如下关系模式:

BJB (班级编号, 院系, 专业, 人数)

BZB (学号, 姓名, 班级编号)

或者:

BJB (班级编号, 院系, 专业, 人数, 学号)

BZB (学号, 姓名)

### 2. (1:n) 联系的 E-R 图到关系模式的转换

对于 (1:n) 的联系, 既可单独对应一个关系模式, 也可以不单独对应一个关系模式。

(1) 联系单独对应一个关系模式, 则由联系的属性、参与联系的各实体集的主码属性构成关系模式,  $n$  端的主码作为该关系模式的主码。

例如, 考虑图 1.6 描述的“班级 (BJB)”与“学生 (XSB)”实体集 E-R 模型, 可设计如下关系模式:

BJB (班级编号, 院系, 专业, 人数)

XSB (学号, 姓名, 性别, 出生时间, 专业, 总学分, 备注)

SYB (学号, 班级编号)

(2) 联系不单独对应一个关系模式, 则将联系的属性及 1 端的主码加入  $n$  端实体集对应的关系模式中, 主码仍为  $n$  端的主码。

例如, 图 1.6 描述的“班级 (BJB)”与“学生 (XSB)”实体集 E-R 模型可设计如下关系模式:

BJB (班级编号, 院系, 专业, 人数)

XSB (学号, 姓名, 性别, 出生时间, 专业, 总学分, 备注, 班级编号)

### 3. (m:n) 联系的 E-R 图到关系模式的转换

对于 (m:n) 的联系, 单独对应一个关系模式, 该关系模式包括联系的属性、参与联系的各实体集的主码属性, 该关系模式的主码由各实体集的主码属性共同组成。

例如, 图 1.7 描述的“学生 (XSB)”与“课程 (KCB)”实体集之间的联系可设计如下关系模式:

XSB (学号, 姓名, 性别, 出生时间, 专业, 总学分, 备注)

KCB (课程号, 课程名称, 开课学期, 学时, 学分)

CJB (学号, 课程号, 成绩)

关系模式 CJB 的主码是由“学号”和“课程号”两个属性组合起来构成的, 一个关系模式只能有一个主码。

至此, 已介绍了根据 E-R 图设计关系模式的方法, 通常这一设计过程称为逻辑结构设计。

在设计好一个项目的关系模式后, 就可以在数据库管理系统环境下创建数据库、关系表及其他数据库对象, 输入相应数据, 并根据需要对数据库中的数据进行各种操作。

### 1.2.3 数据库物理设计

数据的物理模型即指数据的存储结构,如对数据库物理文件、索引文件的组织方式、文件的存取路径、内存的管理等。物理模型对用户是不可见的,它不仅与数据库管理系统有关,还与操作系统甚至硬件有关。

## 1.3 SQL Server 2008 环境

### 1.3.1 SQL Server 2008 概述

SQL Server 2008 是 Microsoft 公司推出的一种关系数据库管理系统。

### 1.3.2 SQL Server 2008 的安装

#### 1. 安装 SQL Server 2008 的硬件和软件要求

为了正确安装和运行 SQL Server 2008,计算机必须满足以下配置要求。

##### (1) 硬件:

处理器: 需要 Pentium (兼容) 处理器,处理器速度最好在 2 GHz 以上。

内存: 2 GB 以上。

硬盘: 1.7 GB 的安装空间以及必要的预留空间。

##### (2) 软件:

①运行在 Windows (32 位) 操作系统之上。SQL Server 2008 设计了不同的分支版本,每个版本对操作系统的要求不尽相同。

②安装组件要求。SQL Server 2008 安装时需要的组件如下:

- .NET Framework 3.5。
- SQL Server Native Client。
- SQL Server 安装程序支持文件。
- Microsoft Windows Installer 4.5 或更高版本。
- Microsoft 数据访问组件 (MDAC) 2.8 SP1 或更高版本。

这些组件将会在安装 SQL Server 2008 的过程中自动安装,不需要用户单独安装。

#### 2. SQL Server 2008 的安装

用户应该根据不同的需求选择合适的版本进行安装,安装步骤类似。SQL Server 2008 中文企业评估版安装步骤如下。

第 1 步 插入自动运行的安装光盘或双击已经下载的 SQL Server 2008 安装程序。SQL Server 2008 需要 .NET Framework 3.5 的支持,安装程序启动后会检测系统是否已经安装了 .NET Framework 3.5,如果没有安装,则弹出要求安装的对话框,单击“确定”按钮,等待一段时间后进入如图 1.8 所示的“.NET Framework 3.5 许可协议”窗口,同意许可条款并单击“安装”按钮开始安装,安装完成后单击“安装完成”窗口中的“退出”按钮即可。

第 2 步 安装完 .NET Framework 3.5 后可能会弹出需要安装补丁的对话框,这些是安装 SQL Server

2008 前必须安装的补丁。安装完该补丁后重启计算机。



注意:

有关 Visual Studio 2008 SP1 补丁和 SQL Server 2008 的一些说明如下。

由于 SQL Server 2008 会使用 Visual Studio 2008 SP1 补丁的某些功能 (可以理解为 .NET Framework 3.5 SP1 的功能), 所以会发生冲突, 即如果计算机上已经安装了 Visual Studio 2008 或 .NET Framework 3.5, 但是没有安装 Visual Studio 2008 SP1 或 .NET Framework 3.5 SP1, 那么在安装 SQL Server 2008 时会导致安装失败。有以下三个选项:

(1) 如果没有安装过 Visual Studio 2008 或 .NET Framework 3.5, 那么用户可以安装 SQL Server 2008, 不会受到影响, 因为 SQL Server 2008 安装程序带有所需的所有组件。

(2) 如果已经安装过 Visual Studio 2008 或 .NET Framework 3.5, 那么必须在安装 Visual Studio 2008 SP1 或 .NET Framework 3.5 SP1 之后再安装 SQL Server 2008, 否则使用选项 (3)。

(3) 如果不想安装 Visual Studio 2008 SP1 或 .NET Framework 3.5 SP1, 那么安装 SQL Server 2008 时需要使用手动安装, 例如, 不要选择 SQL Server 的 Analysis Services、Integration Services 或 Business Intelligence Development Studio 功能。

第 3 步 重启计算机后重新启动安装程序, 进入“SQL Server 安装中心”窗口, 单击左边菜单栏中的“安装”选项卡, 在窗口右边将列出可以进行的安装方式, 如图 1.9 所示。单击“全新 SQL Server 独立安装或向现有安装添加功能”选项, 将安装全新的 SQL Server 2008。



图 1.8 .NET Framework 3.5 许可协议



图 1.9 SQL Server 2008 安装中心

第 4 步 安装程序将检查 SQL Server 安装程序支持文件时可能发生的问题, 并将检查信息显示在“安装程序支持规则”窗口中, 如图 1.10 所示。如果有检查未通过的规则, 必须进行更正, 否则安装将无法继续。

第 5 步 安装程序支持规则全部通过后单击“确定”按钮进入“产品密钥”窗口, 如图 1.11 所示。在“指定可用版本”选项中选择“Enterprise Evaluation”, 在“输入产品密钥”选项中输入企业评估版的 25 位产品密钥, 完成后单击“下一步”按钮。

第 6 步 进入“许可条款”窗口, 阅读并接受许可条款, 单击“下一步”按钮。进入“安装程序支持文件”窗口, 如图 1.12 所示, 单击“安装”按钮安装 SQL Server 必备组件。安装完成后重新进入“安装程序支持规则”窗口, 如图 1.13 所示。如果通过, 则单击“下一步”按钮。





图 1.10 “安装程序支持规则”窗口



图 1.11 “产品密钥”窗口



图 1.12 “安装程序支持文件”窗口



图 1.13 安装程序支持规则检查

第 7 步 进入“功能选择”窗口, 在“功能”区域中选择要安装的功能组件, 用户可以根据自身需求来安装组件, 这里单击“全选”按钮安装全部组件。单击“下一步”按钮进入“实例配置”窗口。

第 8 步 在“实例配置”窗口中进行实例配置, 如果是第一次安装, 则既可以使用默认实例, 也可以自行指定实例名称。如果当前服务器上已经安装了一个默认的实例, 则再次安装时必须指定一个实例名称。自定义实例名的方法为: 选择“命名实例”单选按钮, 在后面的文本框中输入用户自定义的实例名称。如果选择“默认实例”单选按钮, 则实例名称默认为“MSSQLSERVER”。这里选择“命名实例”单选按钮, 名称为“SQL2008”, 如图 1.14 所示。

第 9 步 实例配置完成后单击“下一步”按钮进入“磁盘空间要求”窗口, 在该窗口中显示安装 SQL Server 2008 所需要的磁盘容量。单击“下一步”按钮进入“服务器配置”窗口, 在“服务账户”选项卡中为每个 SQL Server 服务单独配置用户名和密码及启动类型。“账户名”可以在下拉框中进行选择, 也可以单击“对所有 SQL Server 服务使用相同的账户”按钮, 为所有的服务分配一个相同的登录账户。配置完成后的界面如图 1.15 所示, 单击“下一步”按钮。

第 10 步 进入“数据库引擎配置”窗口, 在“账户设置”选项卡中选择“身份验证模式”。“身份验证模式”是一种安全模式, 用于验证客户端与服务器的连接, 它有两个选项: “Windows 身份验证模式”和“混合模式 (SQL Server 身份验证和 Windows 身份验证)”。在“Windows 身份验证模式”中, 用户通过 Windows 账户连接时, 使用 Windows 操作系统中的信息验证账户名和密码; 在“混合模式 (SQL Server 身份验证和 Windows 身份验证)”中, 允许用户使用 Windows 身份验证或 SQL Server 身份验证进行连接。建立连接后, 系统的安全机制对于这两种连接是一样的。



图 1.14 “实例配置”窗口



图 1.15 “服务器配置”窗口

本书选择“混合模式 (SQL Server 身份验证和 Windows 身份验证)”为身份验证模式，并为内置的系统管理员账户“sa”设置密码，为了便于介绍，这里密码设为“123456”，如图 1.16 所示。在实际操作过程中，密码要尽量复杂，以提高安全性。

另外，必须为 SQL Server 实例指定至少一个 SQL Server 管理员。单击“添加当前用户”按钮，添加当前 Windows 账户为 SQL Server 管理员。若要添加其他账户，可以单击“添加”按钮。在“数据库引擎配置”窗口的“数据目录”选项卡中还可以修改各种数据库的安装目录和备份目录，这里使用默认的目录。

第 11 步 单击“下一步”按钮进入“Analysis Services 配置”窗口，对 Analysis Services 进行设置，单击“添加当前用户”按钮指定当前 Windows 登录用户对 Analysis Services 具有管理权限。单击“下一步”按钮进入“Reporting Services 配置”窗口，选择“安装本机模式默认配置”选项，单击“下一步”按钮进入“错误和使用情况报告”窗口，用户可以根据需求在复选框中选择相应的选项。

第 12 步 单击“下一步”按钮进入“安装规则”窗口，窗口中将显示安装规则的通过情况，如图 1.17 所示。如果全部通过，则可以单击“下一步”按钮。



图 1.16 身份验证模式选择



图 1.17 “安装规则”窗口

第 13 步 进入“准备安装”窗口，单击“安装”按钮开始安装，等待一段时间后安装完成，窗口中将显示已经成功安装的功能组件，如图 1.18 所示。单击“下一步”按钮，在“完成”窗口中单击“关闭”按钮即可结束安装。



图 1.18 安装进度显示

### 1.3.3 SQL Server 2008 服务器组件

SQL Server 2008 是一个功能全面整合的数据平台，它包含了数据库引擎（Database Engine）、Analysis Services、Integration Services 和 Reporting Services 等组件。SQL Server 2008 的版本不同，提供的组件可能也不相同。

SQL Server 2008 服务器组件可由 SQL Server 配置管理器启动、停止或暂停，这些组件在 Windows 操作系统上作为服务运行。

(1) Database Engine（数据库引擎）。数据库引擎是 SQL Server 2008 用于存储、处理和保护数据的核心服务，例如，查询数据、创建数据库、创建表和视图等操作都是由数据库引擎完成的。数据库引擎提供了受控访问和快速事务处理功能，还提供了大量支持以保持可用性。Service Broker（服务代理）、Replication（复制技术）和 Full Text Search（全文搜索）都是数据库引擎的一部分。

SQL Server 2008 支持在同一台计算机上同时运行多个 SQL Server 数据库引擎实例。每个 SQL Server 数据库引擎实例各有一套不为其他实例共享的系统及用户数据库，应用程序连接同一台计算机上的 SQL Server 数据库引擎实例的方式与连接其他计算机上运行的 SQL Server 数据库引擎的方式基本相同。SQL Server 2008 实例有两种类型：

①默认实例。SQL Server 2008 默认实例仅由运行该实例的计算机的名称唯一标识，它没有单独的实例名，默认实例的服务名称为 MSSQLServer。如果应用程序在请求连接 SQL Server 时只指定了计算机名，则 SQL Server 客户端组件将尝试连接这台计算机上的数据库引擎默认实例。一台计算机上只能有一个默认实例，而默认实例可以是 SQL Server 的任何版本。

②命名实例。除默认实例外，所有数据库引擎实例都可以由安装该实例的过程中指定的实例名标识。应用程序必须提供准备连接的计算机的名称和命名实例的实例名。计算机名和实例名格式为“计算机名\实例名”，命名实例的服务名称即为指定的实例名。

(2) Analysis Services（分析服务，简称 SSAS）。分析服务为商业智能应用程序提供联机分析处理（OLAP）和数据挖掘功能。

(3) Integration Services（集成服务，简称 SSIS）。集成服务主要用于清理、聚合、合并、复制数据的转换以及管理 SSIS 包。除此之外，它还提供生产并调试 SSIS 包的图形向导工具、用于执行 FTP 操作、电子邮件消息传递等工作流功能的任务。

(4) Reporting Services（报表服务，简称 SSRS）。报表服务基于服务器的报表平台，可以用来创

建和管理包含关系数据源和多维数据源中的数据的表格、矩阵、图形和自由格式的报表。

### 1.3.4 SQL Server 2008 管理和开发工具

#### 1. SQL Server 2008 管理工具

Microsoft SQL Server 2008 完成安装后，可在“开始”菜单中查看安装了哪些工具。另外，还可以使用这些图形化工具和命令实用工具进一步配置 SQL Server。表 1.4 列举了用来管理 SQL Server 2008 实例的工具。

表 1.4 SQL Server 管理工具

管理工具	说明
SQL Server Management Studio	用于编辑和执行查询，并用于启动标准向导任务
SQL Server Profiler	提供用于监视 SQL Server 数据库引擎实例或 Analysis Services 实例的图形用户界面
数据库引擎优化顾问	可以协助创建索引、索引视图和分区最佳组合
SQL Server Business Intelligence Development Studio	用于包括 Analysis Services、Integration Services 和 Reporting Services 项目在内的商业解决方案的集成开发环境
Reporting Services 配置管理器	提供报表服务器配置的统一查看、设置和管理方式
SQL Server 配置管理器	管理服务器和客户端网络配置
SQL Server 安装中心	安装、升级或更改 SQL Server 2008 实例中的组件

这里对表 1.4 中的“SQL Server 配置管理器”补充说明如下：

SQL Server 配置管理器用于管理与 SQL Server 2008 相关的服务。尽管其中许多任务可以使用 Microsoft Windows 服务对话框来完成，但值得注意的是，SQL Server 配置管理器还可以对其管理的服务执行更多的操作，例如在服务账户更改后应用正确的权限。

单击“开始”→“所有程序”→“Microsoft SQL Server 2008”→“配置工具”→“SQL Server 配置管理器”，在弹出窗口的左边菜单栏中选择“SQL Server 服务”，即可在窗口右边出现的服务列表中对各个服务进行操作，如图 1.19 所示。



图 1.19 SQL Server 配置管理器

使用 SQL Server 配置管理器可以完成下列服务任务：

- (1) 启动、停止和暂停服务，双击服务列表中的某个服务即可进行操作。
- (2) 将服务配置为自动启动或手动启动、禁用服务或者更改其他服务设置。
- (3) 更改 SQL Server 服务所使用的账户的密码。
- (4) 查看服务的属性。

(5) 启用或禁用 SQL Server 网络协议。

(6) 配置 SQL Server 网络协议。

SQL Server 2008 中还有一些组件作为服务运行，如图 1.19 所示。

① SQL Server 代理。SQL Server 代理是一种 Windows 服务，主要用于执行作业、监视 SQL Server、激发警报以及允许自动执行某些管理任务。SQL Server 代理的配置信息主要存放在系统数据库 msdb 的表中。在 SQL Server 2008 中，必须将 SQL Server 代理配置成具有 sysadmin 固定服务器角色的用户才可以执行其自动化功能，而且该账户必须拥有诸如服务登录、批处理作业登录、以操作系统方式登录等 Windows 权限。

② SQL Server Browser (浏览器)。此服务将命名管道和 TCP 端口信息返回给客户端应用程序。在用户希望远程连接 SQL Server 2008 时，如果用户通过使用实例名称来运行 SQL Server 2008，并且在连接字符串中没有使用特定的 TCP/IP 端口号，则必须启用 SQL Server Browser 服务以允许远程连接。

③ SQL Full-Text Filter Daemon Launcher (全文搜索)。此服务用于快速构建结构化或半结构化数据的内容和属性的全文索引，以允许对数据进行快速的语言搜索。

其中，SQL Server 代理和 SQL Full-Text Filter Daemon Launcher 默认是禁用的。

## 2. SQL Server Management Studio 环境

SQL Server 2008 使用的图形界面管理工具是“SQL Server Management Studio”(简称 SSMS)。在“SQL Server Management Studio”中主要有两个工具：“图形化的管理工具(对象资源管理器)”和“Transact SQL 编辑器(查询分析器)”。此外还拥有“解决方案资源管理器”窗格、“模板资源管理器”窗格和“注册服务器”窗格等，如图 1.20 所示。

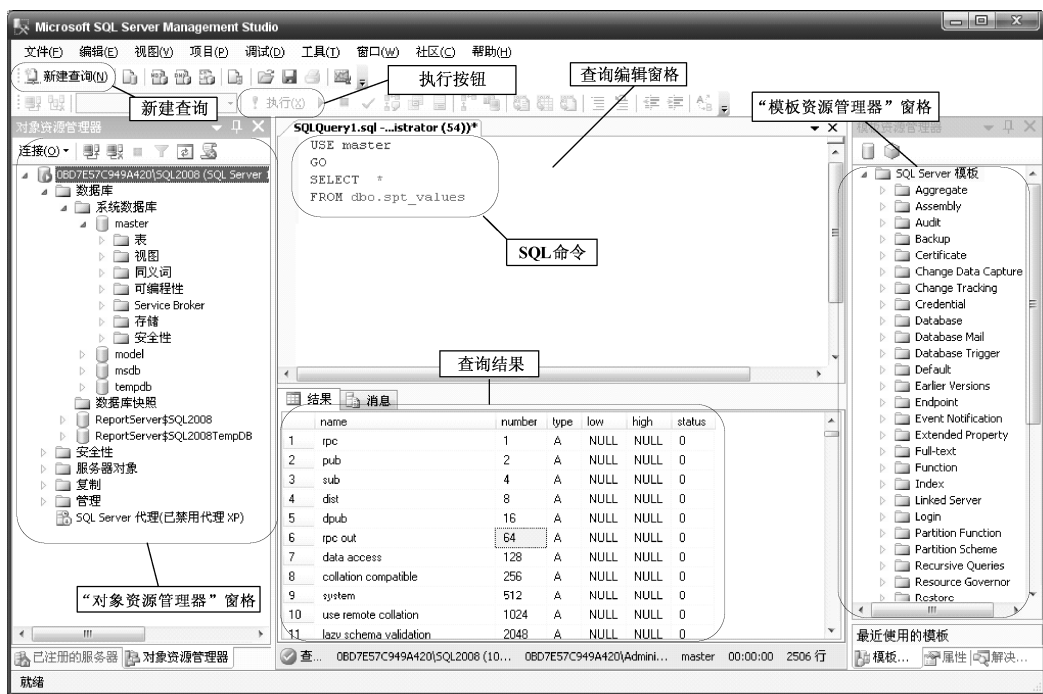


图 1.20 SQL Server Management Studio

(1) “对象资源管理器”与“查询分析器”。在 SQL Server Management Studio 中，用户可以直接通过 SQL Server 2008 的“对象资源管理器”窗口来操作数据库。

Transact SQL（简称 T-SQL）是一种 SQL 语言，与其他各种类型的 SQL 语言一样，使用 T-SQL 语言可以实现从查询到对象建立的所有任务。编写 T-SQL 脚本的方法很简单，只需要用户在“SQL Server Management Studio”面板中单击“新建查询”按钮，在“查询分析器”窗口中输入相应的 SQL 命令，单击“！执行”按钮，系统执行该命令后会将执行的结果自动返回到“SQL Server Management Studio”的结果窗口中显示。

打开“SQL Server Management Studio”的方法如下：

在桌面上单击“开始”→“所有程序”→“SQL Server 2008”→“SQL Server Management Studio”，在出现的“连接到服务器”对话框中，单击“连接”按钮，如图 1.21 所示，就可以以 Windows 身份验证模式启动“SQL Server Management Studio”，并以计算机系统管理员身份连接到 SQL Server 服务器。



图 1.21 “连接到服务器”对话框

观察“SQL Server Management Studio”中的“对象资源管理器”窗格可以发现，在“对象资源管理器”窗格中可以浏览所有的数据库及其对象。

①利用“对象资源管理器”查看数据库对象。以 Windows 身份验证模式登录到“SQL Server Management Studio”，在“对象资源管理器”窗格中展开“数据库”，选择“系统数据库”中的“master”数据库并展开，则将列出该数据库中所包含的所有对象，如表、视图、存储过程等。

②利用“查询分析器”查询“master”数据库中表 `dbo.spt_values` 的数据。在 SQL Server Management Studio 面板中单击“新建查询”按钮，在打开的“查询编辑器”窗口中输入以下命令：

```
USE master
GO
SELECT *
FROM dbo.spt_values
```

单击“！执行”按钮，该查询执行的结果如图 1.20 所示。

(2)“模板资源管理器”。在“SQL Server Management Studio”的“查询分析器”中使用 T-SQL 脚本可以实现从查询到对象建立的所有任务。而使用脚本编制数据库对象与使用图形化向导编制数据库对象相比，最大的优点是使用脚本化方式具有图形化向导方式所无法比拟的灵活性。但是，高度的灵活性，也就意味着使用它的时候有着比图形化向导方式更高的难度。为了降低难度，“SQL Server Management Studio”提供了“模板资源管理器”来降低编写脚本的难度。

在“SQL Server Management Studio”的菜单栏中单击“视图”，选择“模板资源管理器”菜单项，界面右侧将出现“模板资源管理器”窗格，如图 1.20 所示。在“模板资源管理器”窗格中除了可以找到超过 100 个对象以及 T-SQL 任务的模板之外，还包括备份和恢复数据库等管理任务。

例如，在图 1.20 中可以双击“Database”→“create\_database”图标，打开创建数据库的脚本模板。

(3)“已注册的服务器”。“SQL Server Management Studio”界面有一个单独的可以同时处理多台

服务器的“已注册的服务器”窗格。可以用 IP 地址来注册数据库服务器，也可以用比较容易分辨的名称为服务器命名，甚至还可以为服务器添加描述。名称和描述会在“已注册的服务器”窗格中显示。

如果想要知道现在正在使用的是哪台服务器，只需要单击“SQL Server Management Studio”菜单栏中的“视图”，选择“已注册的服务器”菜单项，即可打开“已注册的服务器”窗格。

通过“SQL Server Management Studio”注册服务器，可以保存实例连接信息、连接和分组实例，查看实例运行状态。

为了通过在“SQL Server Management Studio”的“已注册的服务器”组件中注册服务器，保存经常访问的服务器的连接信息，可以在“对象资源管理器”中进行连接时注册服务器。

在“对象资源管理器”中注册服务器的主要步骤如下：

启动“SQL Server Management Studio”，在菜单栏中选择“视图”，在弹出的子菜单中选择“已注册的服务器”菜单项，弹出“已注册的服务器”窗格，如图 1.22 所示。右击“数据库引擎”节点下的“Local Server Groups”，在弹出的快捷菜单中选择“新建服务器注册”菜单项，打开“新建服务器注册”对话框，单击“常规”选项卡，在“服务器名称”文本框中输入要注册的服务器名称，如图 1.23 所示。在“连接属性”选项卡中，可以指定要连接到的数据库名称和使用的网络协议等其他信息。



图 1.22 “已注册的服务器”窗口



图 1.23 “新建服务器注册”对话框

这里服务器名称应该填要注册的 SQL Server 服务器实例名。在“身份验证”选项上，“身份验证模式”可以选择接受默认设置“Windows 身份验证”，或选择“SQL Server 身份验证”并填写“用户名”和“密码”。设置完成后单击“测试”按钮，测试连接若成功，则单击“保存”按钮，完成新建服务器注册的设置。此时，在“已注册的服务器”窗格就可以看到刚才所注册的服务器的图标了。

(4)“解决方案资源管理器”。在“SQL Server Management Studio”中，“解决方案资源管理器”是用来管理项目方案资源的有效工具。如果读者使用过微软的 Visual Studio 集成开发环境，那么用户对项目和方案的概念就不会感到陌生。在“解决方案资源管理器”中，项目可以将一组文件结合在一起作为组进行访问。创建新项目的步骤如下：

第 1 步 单击菜单栏中的“文件”，在弹出的子菜单中选择“新建”菜单项，单击“项目”，选择所要创建的项目类型。类型主要有“SQL Server 脚本”“Analysis Services 脚本(分析服务脚本)”和“SQL Server Compact Edition 脚本”。然后为创建的项目或方案命名，并选择文件的存储路径，单击“确定”按钮，完成项目的创建过程。

第 2 步 接下来就可以为该项目创建一个或多个（如果所创建的项目接触的数据库不止一个）数据库连接或者添加已经存在的项目文件，如图 1.24 所示，只需要在“解决方案资源管理器”内的“SQL

Server 脚本 1”上右击鼠标，在弹出的快捷菜单中选择要添加的项目即可。

如果在“SQL Server Management Studio”中找不到“解决方案资源管理器”，可以单击面板上的“视图”→“解决方案资源管理器”，打开“解决方案资源管理器”窗口。

有关“SQL Server Management Studio”环境的使用在后面的章节中还会有相关介绍。



图 1.24 “解决方案资源管理器”窗口