

SQL基础教程

(第3版)

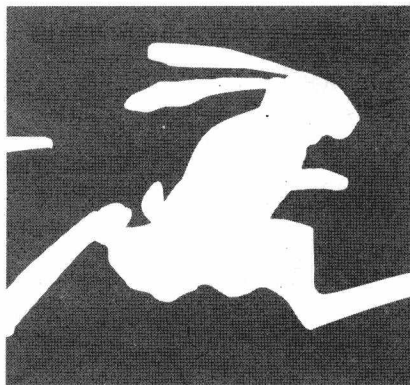
[美] Chris Fehily 著
冯宇晖 贾文峰 译

CREATE • nulls • DROP • INSERT • foreign keys • SELECT • columns • relational model • rows • views • indexes • relationship • normalization • joins • UPDATE • subqueries • primary keys • tables • transaction • CREATE • DROP • INSERT • foreign keys • SELECT • columns • relational model • views • indexes • relationships • normalization • joins • DELETE • UPDATE • queries • primary keys • tables • transactions • CREATE • nulls • DROP • foreign keys • SELECT • columns • relational model • rows • views • relationship • normalization • joins • UPDATE • subqueries • primary keys • tables • transaction • CREATE • DROP • INSERT • foreign keys • SELECT • columns • relational model • views • indexes • relationships • normalization • joins • DELETE • UPDATE • queries • primary keys • tables • transactions • CREATE • DROP • INSERT • foreign keys • SELECT • columns • relational model • rows • views • relationship • normalization • joins • UPDATE • subqueries • primary keys • tables • transaction • CREATE • DROP • INSERT • foreign keys • SELECT • columns • relational model • views • indexes • relationships • normalization • joins • DELETE • UPDATE • queries • primary keys • tables • transactions • CREATE • DROP • INSERT • foreign keys • SELECT • columns • relational model • rows • views • relationship • normalization • joins • UPDATE • subqueries • primary keys • tables • transaction • CREATE • DROP • INSERT • foreign keys • SELECT • columns • relational model • views • indexes • relationships • normalization • joins • DELETE • UPDATE • queries • primary keys • tables • transactions • CREATE • nulls • DROP

- Amazon五星图书，自学与参考两相宜
- 任务驱动，数百实例教你掌握SQL精髓
- 触类旁通，展现主流DBMS的SQL语句异同

TURING

图灵程序设计丛书 数据库系列



SQL基础教程

(第3版)

[美] Chris Fehily 著
冯宇晖 贾文峰 译

CREATE • nulls • DROP • INSERT • foreign keys • SELECT • columns • re
model • rows • views • indexes • relationship • normalization • joins •
• UPDATE • subqueries • primary keys • tables • transaction • CREATE
DROP • INSERT • foreign keys • SELECT • columns • relational model •
views • indexes • relationships • normalization • joins • DELETE • UPD
queries • primary keys • tables • transactions • CREATE • nulls • DROP
• foreign keys • SELECT • columns • relational • model • rows • views •
relationship • normalization • joins • UPDATE • subqueries • primary
keys • tables • transaction • CREATE • DROP • INSERT • foreign keys •
SELECT • columns • relational model • views • indexes • relationships
normalization • joins • DELETE • UPDATE • queries • primary keys • tab
transactions • CREATE • DROP • INSERT • foreign keys • SELECT • colur
relational model • views • indexes • relationships • normalization • j
oins • DELETE • UPDATE • queries • primary keys • tables • transaction
CREATE • nulls • DROP • INSERT • foreign keys • SELECT • columns • re
model • rows • views • indexes • relationship • normalization • joins •
• UPDATE • subqueries • primary keys • tables • transaction • CREATE
DROP • INSERT • foreign keys • SELECT • columns • relational model •

人民邮电出版社
北京

人民邮电出版社
样书
专用章

图书在版编目 (CIP) 数据

SQL 基础教程: 第3版 / (美) 费希利 (Fehily, C.) 著; 冯宇晖, 贾文峰译. —北京: 人民邮电出版社, 2009.12 (图灵程序设计丛书)

书名原文: SQL: Visual QuickStart Guide, Third Edition
ISBN 978-7-115-21350-1

I. S… II. ①费…②冯…③贾… III. 关系数据库-数据库管理系统-教材 IV. TP311.138

中国版本图书馆CIP数据核字 (2009) 第158150号

内 容 提 要

本书是一本 SQL 的入门书, 介绍如何使用最常用的 SQL 语言维护和查询数据库信息。书中介绍了各种 DBMS, 关系模型理论, SQL 语法, 从表中检索数据, 操作符和函数, 汇总和分组数据, 联结, 子查询, 集合操作, 创建、更改和删除表, 索引, 视图, 事务和 SQL 技巧等。本书比较了各种 DBMS 中的 SQL 实现, 并给出大量实例代码及经验技巧。

本书适合 SQL 初学者, 同时也可作为数据库应用开发人员和最终用户的参考书。

图灵程序设计丛书

SQL基础教程 (第3版)

- ◆ 著 [美] Chris Fehily
译 冯宇晖 贾文峰
责任编辑 傅志红
执行编辑 武 嘉
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
- ◆ 开本: 800×1000 1/16
印张: 22.5
字数: 629千字 2009年12月第1版
印数: 1-3 000册 2009年12月北京第1次印刷
著作权合同登记号 图字: 01-2008-4297号

ISBN 978-7-115-21350-1

定价: 49.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版权声明

Authorized translation from the English language edition entitled *SQL: Visual QuickStart Guide, Third Edition* by Chris Fehily, published by Pearson Education, Inc., publishing as Peachpit Press, Copyright © 2008 by Chris Fehily.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by an information storage retrieval system, without permission of Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by POST & TELECOM PRESS Copyright © 2009.

本书中文简体字版由美国Pearson Education 授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制本书内容。

版权所有，侵权必究。

译者序

SQL是关系数据库编程的国际标准语言。本书系统、全面地介绍了标准SQL，并讲解了Microsoft Access、Microsoft SQL Server、Oracle、IBM DB2、MySQL、PostgreSQL等SQL实现及其与标准SQL之间的差异。

本书结构严谨、内容丰富，给出大量实例代码及经验技巧。本书适合SQL初学者，同时也可作为数据库应用开发人员和最终用户的参考书。作为初学者，可通过学习本书快速、全面地掌握SQL；而作为数据库应用开发人员和最终用户，可以通过本书提高开发和应用关系数据库的能力。

本书由冯宇晖、贾文峰翻译，其中文前、第1~6章、第7章前6节由贾文峰翻译，第7章7~9节及第8~15章由冯宇晖翻译，全书由冯宇晖负责统稿。限于译者的水平，译文中难免有错误与不足之处，敬请读者和同行批评指正。译者的邮箱：fengyh1999@163.com和jwfw@lit.edu.cn。

前 言

SQL是一种标准的编程语言，用于创建、更新和检索存储在数据库中的信息。使用SQL，你可以将通常的问题（“我们的客户居住在何地？”）转化为数据库系统能够理解的语句（`SELECT DISTINCT city, state FROM customers;`）。你可能已经知道如何通过图形化的查询或报表工具来检索此类信息，但对于复杂的问题，或许你也意识到此类工具存在诸多限制和障碍——这时就需要SQL了。

可以使用SQL来增加、更新、删除数据和数据库对象。所有现代关系型DBMS（数据库管理系统）都支持SQL，但不同产品的支持情况会有所不同。

本书涉及大多数流行DBMS的最新版本，并专门新增一章来介绍SQL技巧，包括最新的编程技巧提示、细微变化和高级主题，以及其他零星知识。

关于 SQL

SQL可以如下表述：

- 是一种编程语言；
- 容易学习；
- 是说明性语言；
- 是交互式或嵌入式语言；
- 是标准化语言；
- 用于改变数据和数据库对象；
- 不是首字母缩写词。

它是一种编程语言。SQL是一种可以用来编写程序，实现创建、修改、查询数据库的形式语言。数据库系统执行SQL程序，完成用户指定的任务，并显示结果或错误信息。编程语言有别于自然语言，它是为了特殊用途而设计的，其词汇量小、语句书写呆板且必须明确无误。如果你没有得到希望的结果，只能怪你自己的程序包含某些错误，而不是计算机错误地执行了指令（程序测试是编程的一项重要工作）。

和其他形式语言一样，SQL由一系列语法和语义规则定义。语法规则决定可以使用哪些单词和符号，以及如何将它们结合在一起。语义规则决定着语法正确的语句的真实含义。注意，SQL语句可能语法正确但语义错误。第3章会介绍SQL的语法和语义。

数据库与DBMS

数据库并不等同于用户使用的数据库软件，“Oracle是数据库”是不正确的说法。数据库软件被称作DBMS。数据库只是DBMS的一个组成部分，是数据本身；也就是说，它是一个存储结构化信息的容器（由至少一个文件组成）。除了控制数据库中数据的组织、完整性和检索，DBMS还要负

责其他任务，如物理存储、安全、备份、错误处理等。

本书中可以认为DBMS是RDBMS的简写，在这里R代表关系型。关系型DBMS依照关系模型（见第2章）而不是层次模型或网状模型来组织数据。本书只讲解关系型系统，所以当书中出现DBMS时，就是指RDBMS，第一个字母R（关系型）被省略了。

它容易学习。同其他编程语言相比，SQL学起来更容易。如果以前没有写过任何程序，你会觉得从自然语言转向形式语言是很难的。不过，SQL语句读起来很像句子，容易学习。初学编程的人可能会理解SQL语句“SELECT au_fname, au_lname FROM authors ORDER BY au_lname;”的意思是“以姓氏为序列出作者的姓名”，但他们会发现，具有相同功能的C或Perl程序很难理解。

它是说明性语言。如果你从未编写过程序，那么跳过本段也不会影响你学习后面的内容。如果你用C或者PHP语言编写过程序，那意味着你有使用过程语言的经验，在编写这样的程序时需要指明得出结果所需的每个步骤。使用SQL这种说明性语言，只需描述想要的内容，而无需去管该如何做，数据库系统的优化器将决定“如何做”。因此，标准的SQL没有传统的流程控制结构，如if-then-else、while、for和goto语句。

为了说明这一点差异，本书给出了功能等同的Microsoft Access Visual Basic程序（VB属于过程语言）和SQL程序。代码0-1显示了从包含作者信息的表中检索作者姓名的VB程序。你无需了解整个程序，但要注意它如何使用Do Until循环来显式地定义如何提取数据。代码0-2显示了如何用一条SQL语句（与约20行的VB程序形成鲜明对比）完成相同的功能。使用SQL，只需指明要完成什么工作，数据库管理系统在内部决定并执行得出结果所需的具体操作。

代码0-1 这段Microsoft Access Visual Basic程序从一个包含作者信息的数据库表中获取姓名信息，并将结果存放在一个数组中

```
Sub GetAuthorNames()  
    Dim db As Database  
    Dim rs As Recordset  
    Dim i As Integer  
    Dim au_names() As String  
    Set db = CurrentDb()  
    Set rs = db.OpenRecordset("authors")  
    rs.MoveLast  
    ReDim au_names(rs.RecordCount - 1, 1)  
    With rs  
        .MoveFirst  
        i = 0  
        Do Until .EOF  
            au_names(i, 0) = ![au_fname]  
            au_names(i, 1) = ![au_lname]  
            i = i + 1  
            .MoveNext  
        Loop  
    End With  
    rs.Close  
    db.Close  
End Sub
```

此外，代码0-2是一个简单的SQL查询。在对简单的SQL查询添加排序、过滤、联结等常见的操作后，使用这样一条SELECT语句就可以完成的任务，使用过程代码则可能需要100多行。

代码0-2 这条SQL语句完成的查询与代码0-1中Visual Basic程序完成的查询相同。Access内部的优化器可决定获取数据的最佳方式

```
SELECT au_fname, au_lname
FROM authors;
```

它是交互式或嵌入式语言。在交互式SQL环境中，用户输入的SQL命令直接发送到数据库管理系统，得到结果后立即显示。DBMS的服务器同时拥有图形和命令行工具，用于接受用户输入的SQL语句或包含SQL程序（脚本）的文本文件。

在开发数据库应用程序时，可以将SQL语句“嵌入”到编写程序所用的宿主语言（host language）中。宿主语言通常是一种通用语言（如C++、Java或COBOL）或脚本语言（如Perl、PHP或Python）。例如，一个PHP CGI脚本可以用SQL语句来查询MySQL数据库；MySQL将查询的结果返回给PHP变量，以便进一步分析或显示在网页上。根据前面的例子，可以将SQL语句嵌入到Visual Basic程序中（代码0-3）。

代码0-3 这里，Visual Basic作为嵌入式SQL的宿主语言

```
Sub GetAuthorNames2()
    Dim db As Database
    Dim rs As Recordset
    Set db = CurrentDb()
    Set rs = db.OpenRecordset("SELECT au_fname, au_lname FROM authors;")
    ' --Do something with rs here.
    rs.Close
    db.Close
End Sub
```

本书只包括交互式SQL。一般情况下，任何可以交互使用的SQL语句，也都可以用在宿主语言中。但在DBMS、宿主语言和操作环境中，语法上略有差异。

它是标准化语言。SQL不属于任何公司。它是一个由国际标准化组织（ISO）和国际工程协会（IEC）共同领导的国际标准工作组定义的开放标准。美国国家标准协会（ANSI）参加了这个工作组，并已批准该标准（见图0-1）。因为“ISO/IEC SQL”不常用，所以在本书中使用更常见的“ANSI SQL”。本书依据的是2003年的SQL标准，除非另外指明，否则书中的ANSI SQL、SQL:2003和SQL所指是一样的。要了解更多相关内容，参见3.2节。

所有DBMS供应商都增加了专有功能来增强这种语言。这些扩展通常是额外的命令、关键字、函数、操作符、数据类型，还有流程控制结构（如if、while和goto语句）。微软、甲骨文和IBM对标准SQL增加了相当多的功能，于是分别形成了Transact-SQL、PL/SQL和SQL PL语言，可被视为这些供应商自己的语言，而不只是SQL的超集。某个供应商的扩展通常与其他供应商的产品不

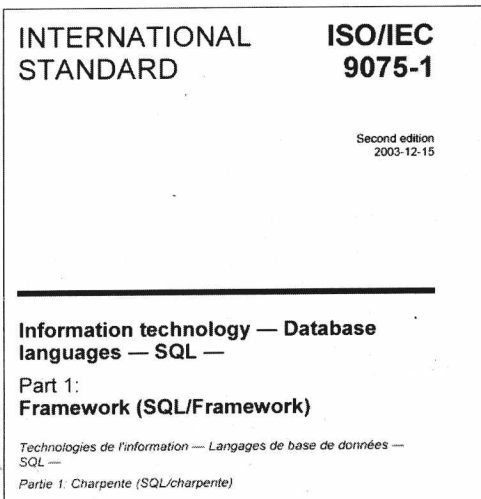


图0-1 这是ISO/IEC 9075:2003标准的封面，对SQL:2003给出了官方定义。你可以从www.ansi.org或www.iso.org购买电子版。该标准面向的读者不是SQL程序员，而是那些设计DBMS、编译器和优化器的人

兼容。本书没有介绍专有的SQL扩展，但当某供应商专有的SQL不符合书中给出的标准SQL例子时，会特别指出。

它用于改变数据和数据库对象。SQL语句分为以下3类。

- 数据操作语句（DML）用于在数据库中检索、计算、插入、编辑和删除数据。第4章至第10章介绍了数据操作语句SELECT、INSERT、UPDATE和DELETE。第14章介绍了语句START（或BEGIN）、COMMIT和ROLLBACK。
- 数据定义语句（DDL）用于创建、修改和销毁数据表、索引、视图等数据库对象。第11章至第13章介绍了数据定义语句CREATE、ALTER和DROP。
- 数据控制语句（DCL）用于授权某些用户查看、更改、删除数据或数据库对象。GRANT语句分配权限和角色（角色是一组权限的集合）。REVOKE语句用于取消权限。本书没有介绍GRANT和REVOKE，因为它们是数据库管理员的职责。书中介绍的所有DBMS（除了Access）都支持GRANT和REVOKE，但与SQL标准有一些差异。

它不是首字母缩写词。“SQL表示结构化查询语言”是一种普遍的误解。它只代表S-Q-L，不代表任何其他意思。为什么？因为ANSI就是这样解释的。官方名字是数据库语言SQL，如图0-1所示。而且，如果说SQL表示“结构化查询语言”，那会令刚开始学习SQL编程的新手感到困惑。业内人士认为“结构化查询语言”可能是SQL最糟糕的解释，因为SQL：

- 不是结构化的（无法将其分解成程序块或过程）；
- 不只用于查询（它不只包含SELECT语句）；
- 不是一种语言（它不满足图灵完备性，你在学习计算理论时会遇到这个概念）。

关于本书

本书将介绍如何使用SQL编程语言维护和查询数据库信息。第1章至第3章将介绍各种DBMS、关系模型和SQL语法，之后会继续沿袭读者已经熟悉的*Visual QuickStart*系列图书基于任务、图解式的编写风格。

尽管阅读本书不要求读者有编程经验，但希望你熟悉操作系统的文件系统，知道如何在命令行模式或shell（在较早的Windows版本里称作DOS提示符，在Mac OS X里称作Terminal）中输入命令。

本书只介绍最常用的语句，不是SQL的详尽指南。要了解更多的SQL语句知识，请参考所用DBMS文档或包含更详细SQL标准的参考书。

✓提示

- Peter Gulutzan和Trudy Pelzer所著的*SQL-99 Complete, Really* (CMP Books) 完整解释了SQL-99标准。要比SQL标准看着轻松，但没有提及具体的DBMS。
- Kevin Kline、Daniel Kline和Brand Hunt所著的*SQL in a Nutshell* (O'Reilly)包含了本书介绍的DBMS（除Access以外），是比较全面的SQL:2003参考书。它适合于已经有一定基础的SQL程序员。
- Troels Arvin的“Comparison of Different SQL Implementations”解释了DBMS实现SQL各种功能的不同之处，包括源文件和其他SQL图书、文章、资源的链接。它包括了SQL:2003和与本书相同的DBMS（除Access以外），见<http://troels.arvin.dk/db/rdbms>。

配套站点

在www.fehily.com,你可以找到修正、更新代码和可供下载的示例数据库(见2.6节)。单击Contact链接就可以给我发对本书的问题、建议、修改及批评意见。

SQL Server 与桌面DBMS

SQL服务器DBMS在客户/服务器网络中作为服务器端,它存储数据并响应客户端的SQL请求。客户端是一个应用程序或一台向服务器发送SQL请求并接受服务器响应的计算机。服务器真正执行对数据库的操作,客户端只是接收答案。如果网络使用了客户/服务器架构,客户端就是眼前桌子上的计算机,服务器则是放在另一房间、另一座大楼甚至另一个国家的强大的专业化设备。客户/服务器的请求和响应如何传递的规则在DBMS协议和类似ODBC、JDBC和ADO.NET的接口中定义。

桌面DBMS是独立的程序,它存储数据库并执行所有SQL指令,或作为SQL服务器的客户端。桌面DBMS不能接受其他客户端的请求(或者说,不能作为SQL服务器)。

SQL服务器包括Microsoft SQL Server、Oracle、DB2、MySQL和PostgreSQL。桌面系统包括Microsoft Access和FileMaker Pro。注意,SQL server(没有大写)指的是所有供应商的SQL服务器产品,SQL Server(大写)是Microsoft的SQL服务器产品。

读者对象

本书的读者对象是数据库应用开发人员和最终用户(不是数据库设计人员或管理人员),主要适合以下人员。

- 熟悉计算机但缺乏编程经验。
- 正在自学或跟随指导老师学习SQL。
- 对数据库没有兴趣,但因为工作需要必须处理大量结构化信息。这类人包括统计人员、流行病工作者、网站编程人员、气象工作者、工程师、会计、投资者、科研人员、分析师、销售代表、金融规划和交易员、办公室文员及经理人。
- 对易用但不够强大的图形查询工具不太满意的人。
- 正从桌面数据库转到DBMS服务器(参见“SQL Server与桌面DBMS”)。
- 已经了解一些SQL,不满足于简单的SELECT语句。
- 需要创建、修改或删除表、索引和视图等数据库对象。
- 需要将SQL代码嵌入到C、Java、Visual Basic、PHP、Perl或其他宿主语言。
- 作为网站程序员,需要在网页上展示查询结果。
- 需要桌面SQL的参考书。
- 因为数据列表已经变得太大、太复杂,而无法使用电子表格管理,正从Microsoft Excel转向Microsoft Access。

本书不适合希望学习以下内容的读者。

- 如何设计数据库(尽管在第2章简述了设计概念)。
- DBMS供应商在基本SQL语句上增加的扩展特性。

- 高级编程或管理。本书没有包含安装、权限、触发器、递归^①、存储过程、复制、备份、恢复、游标、整理、字符集、翻译、XML及面向对象扩展。

排版约定

本书使用以下排版约定。楷体表示引入的新术语。英文斜体表示变量。代码体表示SQL代码和代码及一般文本里的语法，也表示可执行的代码、文件名、目录（文件夹）名、URL和命令行提示符文本。粗代码体表示SQL代码片段和结果对应正文中的解释部分。斜代码体表示SQL代码中必须使用用来取代的变量，例如必须用真实的列名取代`column`。

语法定义

SQL是对行中断和每行长度都有限制的自由形式语言。为了使代码易于阅读和维护，本书在SQL语法图解和代码中使用一致的风格。

- 每个SQL语句开始于新行。
- 每次缩进为2个字符。
- 每个语句的子句开始于新的缩进的行。

```
SELECT au_fname, au_lname
      FROM authors
      ORDER BY au_lname;
```


- SQL是不区分大小写的，这意味着myname、MyName和MYNAME是同样的标识符。本书对于SQL关键字（如SELECT、NULL和CHARACTER）使用大写（见3.1节）；对于用户定义的值（如表、列和别名）使用小写。（对于一些DBMS，用户定义的标识符在引用时或其他一些场合是区分大小写的，因此最安全的方法，是在SQL程序中遵照标识符的大小写规定。）
- 表0-1显示了本书在语法图中使用的特殊符号。
- 在SQL代码中所有引号是直的引号（如'和"），而不是弯曲的引号（如’和“），弯曲的引号会使代码无法运行。
- 当一行太窄而无法容纳一整行代码或输出时，将其分为两段或两段以上；箭头→表示行的继续。

表0-1 语法定义

符 号	说 明
	竖线或管道符分隔可替换的项目，采用其中任何一个即可（不要输入竖线本身）。A B C读作“A或B或C”。不要将管道符与双管道符 混淆，后者是SQL的字符串连接符
[]	方括号包围一个或多个可选项（不要输入方括号本身）。[A B C]意味着“输入A或B或C，或不输入”，[D]意味着“输入D，或不输入”
{ }	大括号包围一个或多个必选项。（不要输入大括号本身。）{A B C}意味着“输入A或B或C”
...	省略号意味着前面的项可以重复任意次

^① 要理解递归，就要先了解递归——这就是递归。

在特定的DBMS下使用SQL

 这个图标表示某个产品未遵循SQL:2003标准。如果见到这个图标，就要注意特定供应商的SQL语法并不遵循标准，必须修改列出的SQL代码才能在DBMS上运行。例如，标准SQL连接两个字符串的操作符是||（双管道符），但Microsoft产品使用+（加号），MySQL使用CONCAT()函数，需要将例子代码中所有的 $a||b$ 变为 $a + b$ （如果使用Microsoft Access或Microsoft SQL Server）或CONCAT(a,b)（如果使用MySQL）。在大多数场合，SQL例子运行正常或只需进行小的修改。因为有的DBMS不支持某种功能，SQL代码偶尔也会无法工作。

本书涉及以下DBMS（详细内容见第1章）。

- Microsoft Access
- Microsoft SQL Server
- Oracle
- IBM DB2
- MySQL
- PostgreSQL

如果你使用其他DBMS（如Teradata、Sybase或Informix），遇到某个SQL例子无法运行，请查阅文档了解所用DBMS的SQL实现与SQL标准有何不同。

配置环境

要在计算机中改写后运行本书中的示例代码，需要用到：

- 文本编辑器
- 示例数据库
- 数据库管理系统

文本编辑器。在命令行提示符中输入简短或临时性的交互式SQL语句虽然很方便，但这样做就不能把有价值的SQL语句存储在文本文件里。文本编辑器是打开、创建、编辑文本文件的程序，它只包含可打印的字符、数字、符号，不支持字体、格式、不可见的符号、颜色、图像或其他文字处理软件中才有的复杂功能。操作系统都会包含免费的文本编辑器，如Windows有记事本，Unix有vi和emacs，Mac OS X有TextEdit。按照惯例，SQL文件的扩展名是`.sql`，但你也可以使用`.txt`（或其他任何扩展名）。

✓提示

- Windows用户可以不使用记事本而选择其他文本编辑工具，如TextPad（需要30美元，www.textpad.com）、EditPlus（需要30美元，www.editplus.com）或Vim（免费，www.vim.org）。
- 也可以在文字处理软件（如Microsoft Word）中输入SQL程序，然后将它们存储为纯文本文件，但这样做会带来维护方面的问题（且专业人士认为这种方式不太好）。

示例数据库。本书中大多数例子使用了同一个数据库，在2.6节中有这个数据库的说明。要创建这个示例数据库，请按照2.7节的指导进行操作。如果你正在使用DBMS服务器，就需要从数据库管理员那里得到许可才能运行创建、修改数据和数据库对象的SQL程序。

数据库管理系统。如何处理SQL？你不必处理——DBMS理解SQL，只要在其中输入SQL程序就行了。DBMS将运行程序并显示结果（见第1章）。

目 录

第 1 章 DBMS 介绍	1	2.6.4 表 titles_authors	35
1.1 运行 SQL 程序	1	2.6.5 表 royalties	36
1.2 Microsoft Access	3	2.7 创建示例数据库	37
1.3 Microsoft SQL Server	6	第 3 章 SQL 基础	40
1.3.1 SQL Server 2000	7	3.1 SQL 语法	40
1.3.2 SQL Server 2005/2008	9	3.2 SQL 标准和一致性	42
1.4 Oracle	10	3.3 标识符	43
1.5 IBM DB2	12	3.4 数据类型	44
1.6 MySQL	16	3.5 字符串类型	45
1.7 PostgreSQL	17	3.6 二进制大型对象类型	47
第 2 章 关系模型	20	3.7 精确数字类型	48
2.1 表、列和行	21	3.8 近似数字类型	49
2.1.1 表	21	3.9 布尔类型	50
2.1.2 列	21	3.10 日期和时间类型	51
2.1.3 行	22	3.11 时间间隔类型	53
2.2 主键	24	3.12 唯一标识符	54
2.3 外键	25	3.13 其他数据类型	55
2.4 联系	26	3.14 空值	55
2.4.1 一对一	27	第 4 章 从表中检索数据	58
2.4.2 一对多	27	4.1 使用 SELECT 和 FROM 检索列	58
2.4.3 多对多	28	4.2 使用 AS 创建列的别名	61
2.5 规范化	29	4.3 使用 DISTINCT 消除重复的行	62
2.5.1 第一范式	29	4.4 使用 ORDER BY 排序行	63
2.5.2 第二范式	30	4.5 使用 WHERE 筛选行	68
2.5.3 第三范式	31	4.6 使用 AND、OR 和 NOT 组合及求反条件	71
2.5.4 其他范式	32	4.6.1 AND 操作符	71
2.6 示例数据库	33	4.6.2 OR 操作符	72
2.6.1 表 authors	33	4.6.3 NOT 操作符	73
2.6.2 表 publishers	34	4.6.4 AND、OR 和 NOT 一起使用	74
2.6.3 表 titles	35	4.7 使用 LIKE 匹配模式	77

4.8 使用 BETWEEN 进行范围筛选	81	7.5 使用 CROSS JOIN 创建交叉联结	141
4.9 使用 IN 进行列表筛选	83	7.6 使用 NATURAL JOIN 创建自然联结	143
4.10 使用 IS NULL 测试空值	85	7.7 使用 INNER JOIN 创建内联结	146
第 5 章 操作符和函数	88	7.8 使用 OUTER JOIN 创建外联结	165
5.1 创建派生列	88	7.9 创建自联结	173
5.2 执行算术运算	89	第 8 章 子查询	177
5.3 确定计算的顺序	92	8.1 理解子查询	177
5.4 使用 连接串	92	8.2 子查询语法	179
5.5 使用 SUBSTRING() 提取子串	95	8.3 子查询和联结	179
5.6 使用 UPPER() 和 LOWER() 更改串的大 小写	97	8.4 简单子查询和相关子查询	182
5.7 使用 TRIM() 修整字符	99	8.4.1 简单子查询	183
5.8 使用 CHARACTER_LENGTH() 得到串长度	101	8.4.2 相关子查询	183
5.9 使用 POSITION() 查找子串	103	8.5 在子查询中限定列名	186
5.10 执行日期及时间间隔运算	105	8.6 子查询中的空值	187
5.11 获得当前日期和时间	106	8.7 使用子查询作为列表表达式	188
5.12 获得用户信息	108	8.8 使用比较操作符比较子查询的值	191
5.13 使用 CAST() 转换数据类型	109	8.9 使用 IN 测试集合成员资格	194
5.14 使用 CASE 计算条件值	112	8.10 使用 ALL 比较所有子查询的值	200
5.15 使用 COALESCE() 检查空值	115	8.11 使用 ANY 比较某些子查询的值	202
5.16 使用 NULLIF() 比较表达式	116	8.12 使用 EXISTS 检测存在性	205
第 6 章 汇总和分组数据	118	8.13 比较等价查询	209
6.1 使用聚合函数	118	第 9 章 集合操作	212
6.2 创建聚合表达式	119	9.1 使用 UNION 合并行	212
6.3 使用 MIN() 查找最小值	120	9.2 使用 INTERSECT 查找相同行	217
6.4 使用 MAX() 查找最大值	120	9.3 使用 EXCEPT 查找不同行	218
6.5 使用 SUM() 计算总和	121	第 10 章 插入、更新和删除行	220
6.6 使用 AVG() 计算平均值	122	10.1 显示表结构	220
6.7 使用 COUNT() 统计行数	124	10.2 使用 INSERT 插入行	223
6.8 使用 DISTINCT 聚合不重复的值	125	10.3 使用 UPDATE 更新行	228
6.9 使用 GROUP BY 分组行	127	10.4 使用 DELETE 删除行	232
6.10 使用 HAVING 筛选分组	132	第 11 章 创建、更改和删除表	235
第 7 章 联结	135	11.1 创建表	235
7.1 限定列名	135	11.2 理解约束	236
7.2 使用 AS 创建表的别名	136	11.3 使用 CREATE TABLE 创建新表	236
7.3 使用联结	137	11.4 使用 NOT NULL 禁止空值	238
7.4 使用 JOIN 或 WHERE 创建联结	139	11.5 使用 DEFAULT 确定默认值	240
		11.6 使用 PRIMARY KEY 指定主键	242

11.7 使用 FOREIGN KEY 指定外键	244	15.4.2 Microsoft SQL Server	293
11.8 使用 UNIQUE 确保值唯一	248	15.4.3 Oracle	294
11.9 使用 CHECK 创建检查约束	250	15.4.4 IBM DB2	295
11.10 使用 CREATE TEMPORARY TABLE 创建 临时表	252	15.4.5 MySQL	296
11.11 使用 CREATE TABLE AS 利用已存在表 创建新表	254	15.4.6 PostgreSQL	297
11.12 使用 ALTER TABLE 修改表	258	15.5 分配排名	298
11.13 使用 DROP TABLE 删除表	259	15.6 计算修整均值	299
第 12 章 索引	261	15.7 随机选取行	300
12.1 使用 CREATE INDEX 创建索引	261	15.8 处理重复值	302
12.2 使用 DROP INDEX 删除索引	264	15.9 创建电话列表	304
第 13 章 视图	265	15.10 检索元数据	305
13.1 使用 CREATE VIEW 创建视图	265	15.10.1 Microsoft Access	305
13.2 通过视图检索数据	269	15.10.2 Microsoft SQL Server	305
13.3 通过视图修改数据	271	15.10.3 Oracle	306
13.3.1 通过视图插入行	271	15.10.4 IBM DB2	307
13.3.2 通过视图更新行	272	15.10.5 MySQL	307
13.3.3 通过视图删除行	273	15.10.6 PostgreSQL	308
13.4 使用 DROP VIEW 删除视图	274	15.11 处理日期	308
第 14 章 事务	275	15.11.1 Microsoft Access	309
第 15 章 SQL 技巧	280	15.11.2 Microsoft SQL Server	310
15.1 动态统计	280	15.11.3 Oracle	310
15.2 产生序列	283	15.11.4 IBM DB2	311
15.3 发现等差数列、递增数列和等值数列	287	15.11.5 MySQL	312
15.4 限定返回行的数量	291	15.11.6 PostgreSQL	313
15.4.1 Microsoft Access	292	15.12 计算中值	315
15.4.2 Microsoft SQL Server	293	15.13 查询极值	316
15.4.3 Oracle	294	15.14 改变动态统计的中流	317
15.4.4 IBM DB2	295	15.15 旋转结果	318
15.4.5 MySQL	296	15.16 处理层次结构	320
15.4.6 PostgreSQL	297	索引	326
15.5 分配排名	298		
15.6 计算修整均值	299		
15.7 随机选取行	300		
15.8 处理重复值	302		
15.9 创建电话列表	304		
15.10 检索元数据	305		
15.10.1 Microsoft Access	305		
15.10.2 Microsoft SQL Server	305		
15.10.3 Oracle	306		
15.10.4 IBM DB2	307		
15.10.5 MySQL	307		
15.10.6 PostgreSQL	308		
15.11 处理日期	308		
15.11.1 Microsoft Access	309		
15.11.2 Microsoft SQL Server	310		
15.11.3 Oracle	310		
15.11.4 IBM DB2	311		
15.11.5 MySQL	312		
15.11.6 PostgreSQL	313		
15.12 计算中值	315		
15.13 查询极值	316		
15.14 改变动态统计的中流	317		
15.15 旋转结果	318		
15.16 处理层次结构	320		



运行SQL程序需要DBMS (Database Management System, 数据库管理系统)。可以让DBMS运行在桌面(本地)计算机上, 或者通过网络使用共享的DBMS。在后一种情况下, 可以用桌面计算机连接运行在另一台机器上的DBMS服务器。运行DBMS的计算机称为主机(host)。

因为本书主要介绍SQL而非DBMS, 所以将不再重复讲解安装和配置数据库软件。初看, 这像是一个一笔带过的借口, 但其实配置DBMS因供应商、产品、版本、版本类型和操作系统而异, 确实非常复杂。而且所有的DBMS都提供了大量的安装、管理、参考和指南文档(仅Oracle的安装手册就超过300页)。

1.1 运行 SQL 程序

本章将介绍如何在下面这些DBMS中运行SQL。

- ❑ Microsoft Access 2007
- ❑ Microsoft SQL Server 2008
- ❑ Oracle 11g
- ❑ IBM DB2 9.5
- ❑ MySQL 5.1
- ❑ PostgreSQL 8.3

这些系统是最流行的商业或开源DBMS。本书中的SQL例子已通过上述版本的测试。这些例子可以运行在更新的版本中, 但是不一定能运行于较早的版本中。后续的版本一般会更符合SQL标准。

Microsoft Access的图形化界面每次只能运行一条SQL语句, 而其他系统(所有的DBMS服务器)能够以交互模式(interactive mode)或脚本模式(script mode)运行SQL程序。在交互模式下, 可以在命令行提示符下输入各条SQL语句然后单独查看每条语句的结果, 因此输入和输出是交替的。在脚本模式(也称为批处理模式)下, 将完整的SQL程序保存在一个文本文件(称为脚本或批处理文件)中, 由命令行工具获取文件、执行程序并返回结果, 而不需要人为干预。本章所有例子使用了同一个示例数据库和代码1-1中所示的SQL程序。同时, 也介绍了命令行工具的少量语法, 完整的语法请参考DBMS文档。

代码1-1 这个名为listing0101.sql的文件包含了一条简单的SQL SELECT语句(在后续的DBMS例子中, 将用它来查询示例数据库)

```
SELECT au_fname, au_lname
```



```
FROM authors
ORDER BY au_lname;
```

命 令 行

大多数数据库专业人员喜欢通过DBMS命令行环境提交命令和SQL脚本，而不是采用鼠标操作菜单和图形化界面的窗口（数据库管理员不会通过单击的方式添加1 000个用户）。如果对DBMS不熟悉，可能会觉得命令行神秘且令人生畏，但一经体验就会发现它强大、简易而高速。图形化工具有一些优点，如：

- 支持剪贴板的剪切、复制和粘贴功能。
- 没有限制的水平和垂直滚动条。
- 通过拖动鼠标可以改变列的宽度。
- 便于查看历史命令和结果。

命令行爱好者也许想体验一下HenPlus (<http://henplus.sourceforge.net>)：一个跨DBMS、免费、全功能的SQL命令解释器。通过网页搜索`sql front end`或`sql client`，可以找到其他解释器。使用Windows的Unix爱好者能够通过Cygwin (www.cygwin.com) 或UWIN (www.research.att.com/sw/tools/uwin) 来运行流行的Unix shell。

2

路 径 名

路径名指定了在文件系统层次结构中一个目录或文件的唯一位置。绝对路径名是从目录树最顶层的节点（称为根）开始的完整的路径。相对路径名是相对于当前（或工作）目录的路径。在Windows中，绝对路径以一个反斜杠（\）或一个驱动器字母后跟一个冒号（:）和一个反斜杠（\）开始。在Unix或Mac OS X终端中，绝对路径以一个斜杠（/）开始。例如，`C:\Program Files\Microsoft SQL Server`（在Windows中）和`/usr/local/bin/mysql`（在Unix中）是绝对路径。`scripts\listing0101.sql`（在Windows中）和`doc/readme.txt`（在Unix中）是相对路径。网络中文件和文件夹的绝对路径也可以用双反斜杠和服务器名开始（例如，`\\某个服务器`）。在本书中，如果路径名包含空格，则会在整个路径名两端加上双引号。

路径名通常简称为路径。尽管依据上下文很容易明白指的是路径名还是路径，但是本书还是使用路径名，以避免与PATH环境变量混淆。

✓提示

- 当在脚本模式中指定SQL文件名时，可以使用绝对或相对路径名（参见“路径名”提要栏）。
- 要想从特定的目录（文件夹）运行命令行工具，路径中必须包括确实含有此工具的目录。路径是操作系统查找程序的目录列表。对于某些DBMS，由安装程序处理路径的详细信息；对于另一些DBMS，必须自行把工具目录添加到路径中。要查看路径，在命令行提示符下输入`path`（在Windows中）或`echo $PATH`（在Unix或Mac OS X终端中）。要更改路径，需要将工具所在的目录添加到`path`环境变量中。要了解更多内容，请查找环境变量的帮助（在Windows中），或者在登录初始化文件（通常被命名为`.bash_login`、`.bashrc`、`.cshrc`、`.login`、`.profile`或`.shrc`）中修改路径命令（在Unix或Mac OS X中）。

3