



开发人员专业技术丛书

SQL Fundamentals

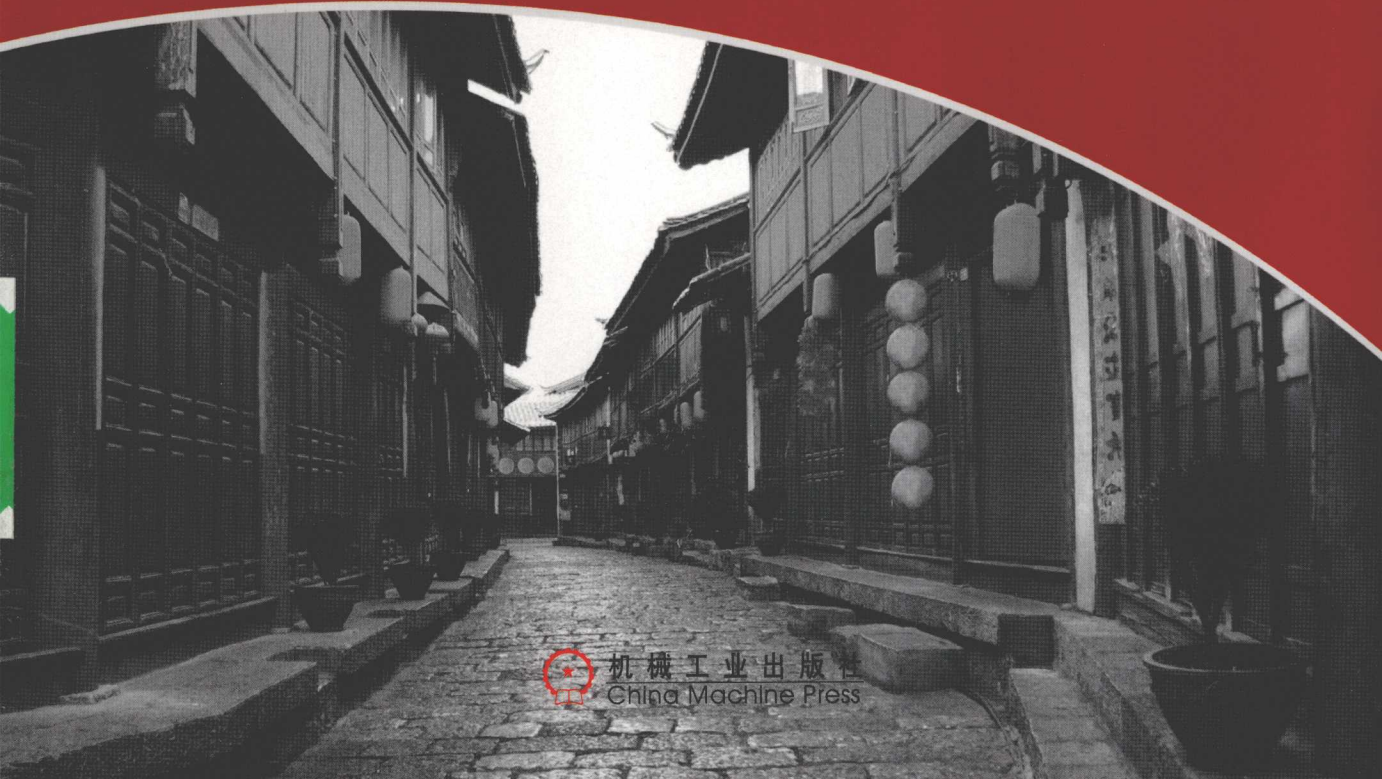
Third Edition

SQL编程基础

(原书第3版)

(美) John J. Patrick 著

刘红伟 董民辉 等译



机械工业出版社
China Machine Press

SQL Fundamentals

Third Edition

SQL编程基础

(原书第3版)

(美) John J. Patrick 著

刘红伟 董民辉 等译



机械工业出版社
China Machine Press

SQL 表示 Structured Query Language, 即结构化查询语言, 是一种数据库语言, 也是最重要的计算机语言之一。

本书根据 John J. Patrick 在美国加州大学伯克利分校讲授的 SQL 课程编写, 详细介绍与 SQL 相关的概念和技术。书中的每一个概念都用了一个 SQL 范例进行说明。在大多数情况下都设置了一个任务, 然后给出了完成这项任务的 SQL 代码。本书还展示了数据的原表和结果表。全书共有 200 多个例子。其中的每一部分都是独立的。读者可以依次阅读本书或者选择性地阅读需要的部分。

本书适合想要学习和掌握 SQL 的读者阅读, 尤其适合高等院校计算机专业学生学习使用, 也可供 IT 专业人员参考。

Simplified Chinese edition copyright © 2009 by Pearson Education Asia Limited and China Machine Press.

Original English language title: SQL Fundamentals, Third Edition (ISBN 0-13-712602-6) by John J. Patrick, Copyright © 2009.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education, Inc.

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2009-2210

图书在版编目 (CIP) 数据

SQL 编程基础(原书第 3 版)/(美)帕特里克(Patrick, J. J.)著; 刘红伟等译. —北京: 机械工业出版社, 2009. 5

(华章程序员书库)

书名原文: SQL Fundamentals, Third Edition

ISBN 978-7-111-26541-2

I. S… II. ①帕… ②刘… III. 关系数据库 - 数据库管理系统 - 程序设计
IV. TP311. 138

中国版本图书馆 CIP 数据核字(2009)第 033467 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 王春华

北京瑞德印刷有限公司印刷

2009 年 8 月第 1 版第 1 次印刷

186mm × 240mm · 31.5 印张

标准书号: ISBN 978-7-111-26541-2

定价 69.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

本社购书热线: (010)68326294

译者序

SQL 表示 Structured Query Language, 即结构化查询语言, 它已经成为数据库应用的标准语言, 因此, 也是最重要的计算机语言之一。对于计算机专业人士来说, SQL 是必须掌握的语言之一, 而对于想要高效率地使用商业数据库的一般读者来讲, 掌握 SQL 可以在应用数据库的时候达到事半功倍的效果。

本书根据 John Patrick 在美国加州大学伯克利分校讲授的 SQL 课程编写, 详细介绍与 SQL 相关的概念和技术。读者不需要任何数据库语言的应用经验, 就可以通过本书详细的介绍和丰富的实例, 熟练地掌握和使用 SQL。

和其他的 SQL 相关图书相比, 本书具有以下特色:

- 不仅介绍基本概念, 而且对每个概念都用了一个 SQL 范例进行说明。在大多数情况下都设置了一个任务, 然后给出了完成这项任务的 SQL 代码。
- 展示了数据的原表和结果表。这样, 读者可以清晰地看到每一个 SQL 范例执行前后的变化, 了解 SQL 语句的具体功能。
- 提供 Oracle SQL 和 Access SQL 两个版本的代码。Oracle 和 Access 分别是应用最广泛的企业级数据库和个人业务数据库。它们很好地覆盖了读者和用户的需求。
- 每个小节的最后提出“检查理解”的练习, 供读者实践本节学习到的知识, 巩固对相关知识点的掌握。
- 配套网站 (<http://groups.google.com/group/sqlfun>) 有丰富的辅助学习资源, 包括创建数据库的代码(针对 Oracle 和 Access 的多个版本)、各章的范例代码、“检查理解”的解决方案、开放的学习讨论论坛。

我们很高兴能够翻译这样一本 SQL 入门好书, 希望读者能够愉快地阅读本书和学习 SQL。参加本书翻译工作的有刘红伟、景龙、景文正、孙海军、李振胜、李秋强、楚亚军、景文生、王志刚、安宁宁、党耀云、谯谊、武文超、庞晋才、刘永昌和孙连升。在阅读本书的过程中, 如果有任何评论和建议, 读者可以通过 liuhongwei198004@sina.com 和译者交流。

译者

2009 年 2 月

前 言

SQL 是最重要的计算机语言之一。它是一种数据库语言。当我们在一个大型的信息库中查找需要的信息时，执行查找的代码很可能正在使用 SQL。分享信息以便与其他人协作的很多应用程序也要使用 SQL。

SQL 已在世界上 100 多种软件产品中使用，并且新的软件产品还在随时增加。本书介绍如何最大程度地利用好数据库，如何使用 SQL 来解决实际问题。本书用到最广为应用的 SQL 产品，即 Oracle 和 Microsoft Access，这两种产品广为使用、容易获取，并且可以在个人计算机上运行。通过详细学习这两种产品，你将具备使用基于 SQL 的任何产品所需的所有基本技能。

本书的组织方式

本书用一种非正式对话的方式带领读者学习 SQL 的主题。本书并列地展示了用 Oracle 和 Access 完成同一任务，以便读者可以看到两者的相似性和不同之处。大多数主题都通过 SQL 代码的一个示例来说明。在这些示例中，我们有意让表很小，从而更容易地检查和理解示例。

每个 SQL 代码示例都首先设定一个任务。然后，给出执行这个任务的 SQL 代码。我尽可能地编写在 Oracle 和 Access 上都能工作的 SQL 代码，然而，有时候我不可能这么做，因此我会针对 Oracle 编写 SQL 代码的一个版本，而针对 Access 编写另一个不同的版本。

为了让本书易于阅读，每个 SQL 示例都给出了初始的数据表和最终的数据表。这使得我们能够检查自己是否理解 SQL 代码做了些什么。我让示例尽可能地小，以便检查。

每个示例的后面常常给出一个提示，说明有关 SQL 代码或数据表的任何较难理解的地方。

最后，我们给出一个问题供读者解决，以便检查自己对于这一主题的理解。你可以自己决定是否去做这些练习。通常，这些问题都较容易解决，只需要对示例中的 SQL 代码略做修改。如果决定要做一个练习，本书的 Web 站点可以帮助你判断自己的解答是否正确。

本书中的每段 SQL 代码示例都设计为互不依赖而彼此独立，不需要之前的章节所做的任何修改。这就使得读者可以跳过本书中的某些内容，按照任意顺序来阅读章节。有些人可能想要从头到尾地阅读本书，但是，不一定非这样不可。

一定要阅读附录，学习如何运行 Oracle 和 Access 的实用技巧。所有示例的数据库文件和代码都可以从本书的 Web 站点获取。在本书中的几个地方，我表达了自己对于计算机技术的观点，而这是很多其他的技术图书避免去做的事情。这些观点是我的个人观点，我完全对它们负责。我也保留改变自己的观点的权利。如果我这么做了，我会在本书的 Web 站点上放上我的新观点，以及促使我改变自己想法的原因。

本书 Web 站点

本书的 Web 站点是一个名为“sqlfun”的 Google group，其 Web 地址是 <http://groups.google.com/group/sqlfun>。

也可以通过 E-mail: sqlfun@ gmail. com 联络我。

这个 Web 站点包括如下内容:

- 创建本书中用到的表的所有 Oracle SQL 代码。
- 带有本书中用到的所有数据表的 Access 数据库。有针对数个版本的 Access 的数据库可供使用。
- 检查读者对本书中的问题的解答方式。
- 勘误表(如果书中有错的话)。
- 开放的讨论区, 可以在此发表评论、提出希望我解答的问题。

欢迎访问本书的 Web 站点。

致谢

很多人对本书贡献良多。我想要借此感谢他们在我编写本书期间给予的所有支持。这些想法和反馈帮助我提高了本书内容的质量, 改进了我表述这些内容的方式。特别地, 我要感谢以下同仁对于本书第 3 版的建议和帮助:

- Dejang Liu
- Alma Lynn Bane

对本书以前版本提供帮助的人包括:

- Anila Manning 对于本书第 2 版的编写提供了很多帮助。
- Paul Reavis, 他与我一起在加州大学伯克利继续教育学校教授这门课程。
- Todd Matson 审阅了 Access 相关的内容。
- Faysal Shaarani 和 Bill Allaway 审阅了 Oracle 相关的内容。
- Spencer Brucker 和加州大学伯克利继续教育学校都支持我教授 SQL 基础课程并准备书中的素材。
- Prentice Hall 的所有同仁, 尤其是编辑 Bernard Goodwin; Moore Media 公司的制作编辑 Vanessa Moore; 本书最初的编辑 Michael Meehan 和 Jeffery Pepper; 以及我没有直接共事的很多其他人。
- 尤其感谢我的母亲 Jean Praninskas; 还有我的儿子 Richard Watts, 他也审阅了本书。

还要感谢 Brian Akitore、Mehran Ansari、Asa Ashraf、Anne Bester、Sandra Bush、Connie Chong、Patricia Cleveland、Robert D'Antony、Gan Davnarrain、Bruce Douglass、James Drummond、Ron Duckworth、Dean Evans、Steve Fajardo、Earl Gardner、Wolday Gebremichael、Neelam Hasni、Reda Ismail、Marques Junior、John Karsnak、Allyson Kinney、Gladys Lattier、Brian Lester、Mahen Luximan、Alex McDougall、E. Muljadi、Satyendra Narayan、Bade Oyebamiji、Stefan Pantazi、Todd Perchert、Oxana Rakova、Jacob Relles、Ricardo Ribeiro、Cindy Roberts、John Rusk、Ty Seward、Gary Shapiro、David Smith、Kenneth Smtih、Joan Spasyk、Patricia Warfel 和 William White。

目 录

译者序 前 言

第 1 章 在表中存储信息	1
简介	1
1-1 什么是 SQL	1
1-2 什么是关系数据库? 为什么要使用它	2
1-3 为什么学习 SQL	3
1-4 本书内容简介	5
表的组成部分	5
1-5 数据存储于表中	5
1-6 一行表示一个对象及其相关信息	6
1-7 一列表示一类信息	7
1-8 单元格是表的最小部分	8
1-9 每个单元格应该只表示一样东西	9
1-10 使用主键列标识每一个行	9
1-11 大部分表都很“瘦长”	10
表的示例	11
1-12 Oracle 和 Access 中表的示例	11
1-13 l_employees 表的一些设计决策	13
1-14 Lunches 数据库	14
小结	18
第 2 章 从表中获取信息	19
select 语句	19
2-1 目标: 从表中获取几个列和行	19
2-2 select 语句概述	19
select 子句	21
2-3 select 子句概述	21
2-4 使用 select 子句获取若干列	22
2-5 使用 select 子句获取所有的列	24
2-6 使用 select 语句获取某个列中的不同值	26
2-7 使用 select 子句获得几个列中的不同值	29

where 子句	30
2-8 where 子句概览	30
2-9 在 where 子句中使用等于条件	31
2-10 在 where 子句中使用小于条件	34
2-11 在 where 子句中使用不等于条件	35
2-12 在 where 子句中使用 in 条件	37
2-13 在 where 子句中使用 between 条件	38
2-14 在 where 子句中使用 like 条件	40
2-15 在 where 子句中使用 is null 条件	42
order by 子句	44
2-16 order by 子句概述	44
2-17 通过一列升序排列行	45
2-18 通过若干列升序排列行	46
2-19 根据若干列以各种顺序排序各行	47
2-20 整个过程	48
小结	49
第 3 章 where 子句中的复合条件	51
where 子句中的复合条件	51
3-1 在 where 子句中使用复合条件	51
3-2 将 not 与 in、between、like 和 is null 条件一起使用	53
3-3 where 子句中复杂条件的标准形式	55
3-4 一个常见错误	56
常数值	58
3-5 在 select 子句中使用常数值	58
3-6 使用常数表	60
标点符号问题	62
3-7 现在需要知道的标点符号	62
3-8 标点符号参考	65
区分大小写	70
3-9 Oracle 中的区分大小写	70
3-10 关于 SQL 中区分大小写的争论	73
3-11 有一个选择	73

3-12 在 Oracle SQL Command Line 环境中关闭区分大小写	73	第5章 数据字典和其他 Oracle 主题 ...	109
3-13 Access 中区分大小写	74	提交、回滚和事务	109
三值逻辑	75	5-1 commit 和 rollback 命令	109
3-14 使用三值逻辑的 SQL	75	5-2 Autocommit 选项	109
错误消息	77	5-3 事务	110
3-15 错误消息常常是错的	77	通过视图修改数据	113
练习	78	5-4 通过视图更改数据	113
3-16 练习 1	78	5-5 通过视图更改数据的例子	114
3-17 练习 2	80	5-6 使用 With Check Option 的视图 ...	120
3-18 练习 3	81	Oracle 中的 SQL Commands 页面	122
小结	83	5-7 SQL Commands 页面概览	122
第4章 保存结果	84	5-8 Autocommit 选项	124
在新表或者视图中保存结果	84	5-9 Explain 选项	124
4-1 从 select 语句的结果中创建 一个新表	84	使用 Oracle 数据字典：第一部分	124
4-2 从 select 语句的结果创建 一个新的视图	87	5-10 数据字典概述	125
4-3 表和视图的相似之处	89	5-11 如何查找所有表的名字	126
4-4 表和视图的不同之处	89	5-12 如何查找所有视图的名字	127
4-5 删除表	90	5-13 如何查找定义视图的 select 语句 ...	128
4-6 删除视图	90	5-14 如何查找表或者视图中的列名	128
4-7 一个视图可以建立在另一个 视图之上	91	5-15 如何查找表的主键	130
4-8 预防性删除	93	小结	132
修改表中的数据	95	第6章 创建自己的表	133
4-9 添加新行到表中	95	创建表	133
4-10 添加若干新行到一个表中	97	6-1 create table 命令	133
4-11 更改已经存在于表的行中的数据 ...	99	6-2 Oracle 和 Access 中的数据类型 ...	134
4-12 从表中删除行	100	6-3 文本数据类型	137
使用 GUI 修改表中的数据	101	6-4 数字数据类型	140
4-13 使用 Oracle GUI 来修改表中的 数据	102	6-5 日期/时间数据类型	140
4-14 使用 Access GUI 来修改表中的 数据	104	6-6 其他数据类型	141
修改表中数据的限制	106	6-7 将数据放入新表中	141
4-15 使用 insert、update 和 delete 的 约束条件	106	6-8 在 Oracle 中创建 l_employees 表 ...	142
4-16 安全限制	107	更改表	144
小结	107	6-9 为表添加主键	144
		6-10 更改表的主键	145
		6-11 向表中添加一个新列	146
		6-12 扩展列的长度	147
		6-13 从表中删除一个列	148
		6-14 对表进行其他更改	148
		带有重复行的表	150

6-15 重复行的问题	150	8-4 not null 约束条件	181
6-16 如何删除重复行	152	8-5 primary key 约束条件	182
6-17 如何区别重复行	153	8-6 对字段数据类型和长度的限制 ...	183
小结	155	参照完整性	183
第7章 格式、序列和索引	156	8-7 参照完整性的概念	183
格式	156	8-8 参照完整性的示例	184
7-1 日期格式	156	8-9 对 RI 保护的数据表使用 insert 和 update	185
7-2 显示已经格式化的日期	158	8-10 通过 RI 允许对数据表使用 insert 和 update	186
7-3 输入已格式化的日期	159	8-11 RI 防止对查找表使用 update 和 delete	187
7-4 Oracle 中的其他格式	161	8-12 如何从查找表中删除编码	187
7-5 Access 中的格式	161	8-13 如何在查找表中更改编码	189
序列	163	8-14 RI 作为表之间的关系	190
7-6 在 Oracle 中创建序列	163	8-15 在 Access GUI 中建立 RI	190
7-7 在 Oracle 中使用序列	164	RI 的 delete 选项和 update 选项	192
7-8 在 Access 中使用序列	165	8-16 删除和更新查找表的三个选项	192
索引	167	8-17 删除规则: set null	193
7-9 创建索引	167	8-18 删除规则: cascade	194
7-10 优化器	168	8-19 更新规则: cascade	196
7-11 索引如何工作的示例	168	验证参照完整性	197
在数据字典中查找更多的信息: 第二部分 ...	169	8-20 主键的两种含义	197
7-12 如何查找有关列的数据类型的 信息	169	8-21 使用两个或者更多的列作为主键 ...	199
7-13 如何查找有关序列的信息	171	8-22 查找表和数据表可以是同一个表 ...	200
7-14 如何找到有关索引的信息	172	如何在 create table 语句中编写约束条件	201
7-15 如何找到所有数据库对象的 相关信息	174	8-23 约束条件常常在 create table 语句中编写	201
7-16 如何使用 Oracle 中的数据 字典表	175	小结	203
7-17 如何使用数据字典列的索引	176	第9章 行函数	204
练习	176	行函数简介	204
7-18 创建一个想要庆祝的 节日的表	176	9-1 直接从初始表中获得数据	204
小结	178	9-2 什么是行函数	204
第8章 数据完整性	179	9-3 select 子句中行函数的例子	207
表上的约束条件	179	9-4 select 语句的所有子句中使用 行函数的例子	208
8-1 约束条件使数据保持一致	179	9-5 定义行函数作为第一步	209
8-2 check 约束条件	179	数字函数	211
8-3 unique 约束条件	180	9-6 数字函数	211

9-7 如何测试行函数	212	11-2 列函数列表	254
9-8 测试数字行函数的另一种方法 ...	213	最大值和最小值	256
文本函数	215	11-3 找出最大值和最小值	256
9-9 文本函数	215	11-4 列函数和 where 子句一起使用 ...	258
9-10 组合名和姓	218	11-5 找出拥有最大值和最小值的行 ...	259
9-11 分隔名和姓	220	计数	261
9-12 设计电话号码的格式	221	11-6 对行和数据计数	261
日期函数	223	11-7 零计数: 第一部分	262
9-13 日期函数	223	11-8 对列中相异值计数	263
9-14 日期函数的一个例子	226	11-9 计算两个或多个列中相异值的 数目	265
9-15 从日期中删除时间	227	求和与平均	267
小结	229	11-10 求和函数与平均函数	267
第 10 章 使用行函数	230	11-11 加法存在的问题以及如何解决 ...	268
特殊行函数	230	其他主题	272
10-1 其他行函数	230	11-12 并不总将 null 更改为零	272
10-2 使用函数识别用户和日期	231	11-13 计算列中 null 的数目	273
10-3 使用函数把 null 改为其他值	232	11-14 对相异值计数	274
10-4 使用函数更改数据类型	234	小结	276
使用行函数的文档	236	第 12 章 控制汇总级别	277
10-5 使用 Oracle 文档	236	将表划分为行的组并汇总每个组	277
10-6 使用 Access 文档	237	12-1 列中数据组的汇总	277
10-7 使用 Access Expression Builder 查看行函数	237	12-2 group by 子句	278
创建数字和日期模式	239	12-3 由两个或更多列形成的组	280
10-8 创建一个简单的数字模式	239	12-4 有多个分组列时的 null 组	282
10-9 创建一个复杂的数字模式	241	12-5 在同一条 select 语句中, 已汇总的 数据不能与没有汇总的数据混在 一起	284
10-10 列出一个星期的所有星期数 ...	242	12-6 解决方案 1: 将更多的列添加到 group by 子句中	286
10-11 创建一个工作日日历	244	12-7 解决方案 2: 将查询划分为两条 单独的 select 语句	287
10-12 如何计算从出生到现在的天数 ...	247	12-8 如何创建有多个小计和一个总计 的报表	289
10-13 如何查找出生后满 10 000 天的 日期	248	12-9 零计数: 第二部分	289
10-14 给 Oracle 和 Access 中的报表 编行号	249	12-10 零计数: 第三部分	290
10-15 可选内容: 解代数方程的一种 简单方法	250	删除某些已汇总的数据	292
小结	253	12-11 having 子句	292
第 11 章 汇总数据	254	12-12 比较 having 子句与 where 子句 ...	293
列函数简介	254		
11-1 汇总一个列中的所有数据	254		

12-13	在一个表上使用 select 语句的全过程	294	14-1	外连接派生于内连接	326
12-14	having 子句没有增加 select 语句的功能	294	14-2	外连接的 3 种类型	327
12-15	使用 where 子句来删除原始数据	296	14-3	左外连接	328
12-16	如何对一个列函数应用另一个列函数并且绕开其他限制	297	14-4	右外连接	330
小结		299	14-5	全外连接	331
第 13 章 内连接		300	14-6	union 简介	333
连接简介		300	14-7	带有匹配列的两个表的 union 的例子	334
13-1	一个查询可以使用几个表中的数据	300	外连接的应用		336
13-2	最佳方法是一次连接两个表	301	14-8	零计数: 第四部分	336
两个表的内连接		301	14-9	组合外连接与数据选取	337
13-3	一对一关系	302	14-10	按排序顺序的全外连接	339
13-4	多对一关系	305	14-11	找出模式中的缺陷	341
13-5	一对多关系	306	14-12	用两列或多列进行比较	342
13-6	多对多关系	307	14-13	比较两个不同的全外连接	344
13-7	删除不匹配的行	308	14-14	问题: 左外连接和右外连接可能很难处理	346
13-8	删除匹配列中带有 null 的行	309	小结		348
13-9	编写 SQL 语句进行内连接的 5 种方式	310	第 15 章 union 和 union all		350
连接条件的变体		312	union 的基础知识		350
13-10	用两个或多个匹配列连接	312	15-1	union 与连接之间的区别	350
13-11	用 between 匹配一系列值的连接	313	15-2	union 和 union all 的区别	352
13-12	使用大于条件的连接	314	15-3	union 中的 select 语句	353
13-13	用行函数进行连接	316	15-4	union 中的 order by 子句	354
13-14	在 from 子句中编写连接条件	316	15-5	创建一个包含 union 的表或者视图	357
连接应用		318	15-6	union 中的自动数据类型转换	358
13-15	查找表	318	非常规的 union		361
13-16	组合连接和数据的选取	320	15-7	不同数据类型表的 union	361
13-17	联合使用连接	322	15-8	不同列数的两个表的 union	362
13-18	如何从数据字典中找出主键列	323	union 的应用		364
13-19	用内连接组合三个或更多的表	324	15-9	确定两个表是否相同	364
小结		325	15-10	在 union 中使用直接量确定数据的来源	365
第 14 章 外连接		326	15-11	给异常、警告和错误的标志附加消息	367
外连接简介		326	15-12	将数据从一个列中分到两个不同的列中	368
			15-13	将两个函数应用到数据的不同部分	369

15-14 三个表或更多表的 union	370	17-3 在 from 子句中一次连接几个表 ...	412
Oracle 中的交集和差集	371	丢失信息	413
15-15 交集	371	17-4 小心使用内连接	413
15-16 差集	372	17-5 小心使用左外连接和右外连接	413
小结	374	17-6 全外连接保留了所有的信息	413
第 16 章 交叉连接和自连接	375	17-7 几个表的全外连接	413
交叉连接	375	注意计算机的效率	415
16-1 交叉连接的定义	375	17-8 监视查询	415
16-2 交叉连接重要的原因	376	17-9 使用索引	415
16-3 内连接源自交叉连接	376	17-10 尽早在过程中选择想要的数据库	416
16-4 内连接的属性	379	17-11 使用表保存总结数据	416
16-5 连接条件中的错误可以显示为一个交叉连接	379	17-12 使用几种方法编写 SQL	416
16-6 使用交叉连接列出所有可能的组合	381	标准化连接表的方式	416
16-7 三维或多维时的其他布局	382	17-13 连接是数据库设计的一部分 ...	416
16-8 避免对大型表使用交叉连接	383	17-14 视图可以标准化表的连接方式 ...	417
自连接	384	17-15 临时报表	420
16-9 表为什么与自己连接	384	小结	420
16-10 自连接的例子	385	第 18 章 if-then-else、参数查询和子查询	421
16-11 处理事件序列	387	子查询	421
16-12 生成从 0 到 999 的数字	389	if-then-else 逻辑	421
Access 中的 CrossTab 查询	391	18-1 Oracle 中的 case 函数和 decode 函数	421
16-13 当有两维时的交叉表查询	391	18-2 Access 中的 Immediate If (iif) 函数	424
16-14 增加到四维的 CrossTab 查询 ...	395	18-3 给行附加消息	426
16-15 更多维的 CrossTab 查询	396	18-4 将一个列的数据划分为两个不同的列	427
16-16 显示谁参加每次午餐的 CrossTab	399	18-5 对数据的不同部分应用两个函数	429
16-17 显示每次午餐的食物	402	参数查询	430
Oracle 中的 CrossTab 查询	405	18-6 Oracle 中的参数查询	431
16-18 Oracle 中的 CrossTab 查询——第一部分	405	18-7 Oracle 中多次使用参数	433
16-19 Oracle 中的 CrossTab 查询——第二部分	406	18-8 在 Oracle 中定义参数的更多方法 ...	434
小结	408	18-9 Access 中的参数查询	436
第 17 章 在产品数据库中组合表	409	18-10 Access 中带有两个参数的查询 ...	437
连接三个或者更多表的方法	409	18-11 Access 中的参数限制	437
17-1 使用一系列步骤连接几个表	409	子查询	438
17-2 在 where 子句中一次连接几个表 ...	411	18-12 子查询简介	438
		18-13 产生一系列值的子查询	439

18-14 产生一个值的子查询	441	19-11 几个人同时使用相同的表	459
18-15 存在 null 值时避免使用 not in ...	442	Oracle 数据字典和多用户环境	460
子查询的应用	443	19-12 ALL 和 USER	460
18-16 用于 update 命令中的子查询 ...	444	19-13 如何在数据字典中找到所需 的表	460
18-17 找出两个表之间的不同之处 ...	445	19-14 如何找到列的含义	460
18-18 使用最新数据	446	小结	461
子查询的原有特性	446	第 20 章 SQL 设计	462
18-19 相关子查询	447	SQL 的最初设计目标	462
18-20 使用 exists 的子查询	448	20-1 做一件事并且做好它	462
18-21 使用子查询编写外连接	448	20-2 关注信息	462
18-22 嵌套子查询	449	20-3 保持 SQL 语言的简单性	463
18-23 子查询的使用限制	449	20-4 协调人们一起工作	463
18-24 许多子查询也可以写作连接 ...	449	新的界面	464
小结	450	20-5 窗体	464
第 19 章 多用户环境	451	20-6 报表	464
数据库配置	451	20-7 Web 工具	464
19-1 单用户环境	451	20-8 更小的数据库	466
19-2 多用户环境	452	20-9 OLTP	466
19-3 分布式环境	452	20-10 数据仓库	467
19-4 通过 Internet 连接	454	小结	467
多用户环境中的操作	454	附录 A 获取和安装 Oracle	468
19-5 如何使用别人拥有的表	454	附录 B Oracle 快速入门	477
19-6 代名	455	附录 C Access 快速入门	481
19-7 快照	456	附录 D Lunches 数据库的图	489
安全性和特权	457		
19-8 识别用户	457		
19-9 特权	457		
19-10 角色	458		

第 1 章 在表中存储信息

在关系数据库中，所有的数据都存储在表中，并且所有的结果都在表中表示。在本章中，我们详细地学习表。

简介

1-1 什么是 SQL

SQL 表示 Structured Query Language (结构化查询语言)。它读作“S-Q-L”，也可以读作“sequel”。

SQL 是一种设计用来从存储在关系数据库中的数据获取信息的语言。稍后我们将介绍什么是关系数据库。现在，我们可以把它看作是在计算机上组织大量数据的一种方法。SQL 允许我们从一个巨大的数据集中找到想要的信息。本书的目的是介绍如何从数据库获取想要的信息。

SQL 和其他的大多数语言不同。使用 SQL 的时候，我们描述想要的信息的类型。然后，计算机确定用来获取数据的最佳过程并且执行这个过程。这叫做**声明式计算机语言** (declarative computer language)，因为其关注点在结果上：我们指定结果应该看起来像什么样子。计算机允许使用任何处理方法，只要它能得到正确的结果。

大多数其他计算机语言是**过程式的** (procedural)，如 C、Cobol、Java、汇编语言、Fortran、Visual Basic 等。我们可以使用这些语言描述应用于数据的过程，而无须描述结果。结果是将过程应用到数据后所得到的任何东西。

我们用类比来比较这两种方法。假设我清晨去一家咖啡店。如果使用 SQL 支持的声明式方法，我可以说：“我想要一杯咖啡和一个多纳圈。”而使用过程式方法，我不能那样说，我必须说出如何得到结果，并且给出获得结果的一个特定过程。也就是说，必须说出如何制作咖啡和多纳圈。因此，对于咖啡，我必须说：“将一些烘烤好的咖啡豆磨成粉，添加开水，冲泡一会儿，再将它倒入一个茶杯，然后给我。”对于多纳圈，我必须读出菜谱中记录的制作方法。显然，声明式方法更贴近我们通常的说话方式，并且它更易于为大多数人所使用。

相对于大多数其他的计算机语言，SQL 易于使用，实际上这也正是 SQL 如此流行和重要的主要原因。常常有人声称可以在一两天内学会 SQL。我认为这种说法更像是一种愿望，实际上可不是这样。SQL 毕竟是一种计算机语言，计算机还不像电话那样易于使用，至少现在还不是这样。

尽管如此，SQL 还是易于使用的。经过一天的培训，大部分人可以学会如何获取许多有用的信息，这包括那些不是程序员的人。组织机构中的所有人，从秘书到副总裁，都可以使用 SQL 获取他们做出商业决策所需要的信息。这正是我们所希望的，并且在很大程度上，这已经成为事实。

信息本身并不强大。只有当一个组织中的人在需要使用信息的时候能够得到信息，信息才变得功能强大。SQL 是传递这些信息的工具。

关于 SQL 的注意事项

- SQL 是从关系数据库中获取信息的声明式语言。
- SQL 会告诉我们获取什么信息，而不是告诉我们如何获取信息。
- SQL 的基本知识易于掌握。
- SQL 使得人们能够控制信息。
- SQL 允许人们使用新的方式来处理信息。
- SQL 通过给需要的人提供信息来使信息变得强大。

1-2 什么是关系数据库？为什么要使用它

关系数据库是一种用来在计算机中组织数据的方法。虽然也有其他方法用来组织计算机中的数据，但是在本书中，除了谈到每种方法的一些优点和缺点之外，我们不打算详细讨论其他方法。目前，我们只关注关系数据库的一些优点。

SQL 是人们将数据组织到关系数据库中的主要原因之一。使用 SQL，整个组织中的人都可以毫不费劲地从数据中获得信息。这是非常重要的。

另一个原因是：许多人可以在同一时间使用关系数据库中的数据。有的时候，成百上千的人可以共享数据库中的数据。所有人都可以看见数据并修改数据（如果他们有修改权限）。从商业观点来看，这提供了一个协调所有职员的方法，并且可以让每一个人都使用相同的信息主体。

第三个原因是：关系数据库是为了使信息可以随时进行更改而设计的。我们可能需要重新组织信息或者为其添加新的信息片段。关系数据库的设计目标是要使这种修改变得容易。大多数其他的计算机系统很难更改。它们假设我们在开始构造系统之前就了解所有的需求。我的经验是：人们并不善于预测将来，即便有时候他们声称可以做到，但在这里，我要说自己对关系数据库有一些偏爱。

从计算机程序员的观点来看，关系数据库的灵活性和 SQL 的可用性使得我们可以比使用传统技术更迅速地来开发计算机应用程序。一些组织利用了这一优点，而另一些组织则没有。

开发关系数据库的思想最初产生于 20 世纪 70 年代初期，是为了处理大量数据和数以百万计的记录。最初，关系数据库被想象成后端处理器，它为那些使用过程式语言（如 C 语言或 Cobol 语言）编写的计算机应用程序提供信息。即使到现在，关系数据库还遗留着这些痕迹。

不过在今天，这种思想变得如此成功，以至于整个信息系统常常构建为关系数据库，并且无须很多过程式代码（除了支持输入表单）。最初开发来为过程式代码做配角的关系数据库现在已经成为了主角，许多过程式代码已经不再需要了。

在关系数据库中，所有数据都保存在表里，表是由列和行组成的二维结构。本章后面部分将详细介绍表。在使用了一段时间的表以后，你会发现表为处理数据提供了一个非常有用的结构。它们易于修改，可以在同一时间与所有用户共享数据，并且可以在表中的数据上运行 SQL。许多人开始从表的角度考虑他们的数据，表已经成为处理数据的主要手段。

现在，人们使用较小的个人数据库来保存地址簿、制作音乐的目录、组织图书馆或者记录自己的财务信息。商业应用程序也作为关系数据库而建立。许多人更愿意将他们的数据存入数据库

中，即使其中只有少量的记录。

关系数据库的起源

- 关系数据库最初是在 20 世纪 70 年代开发的，其目的是以一致并且相关的方式来组织大量信息。
- 关系数据库允许上千人同时使用相同的信息。
- 关系数据库总是保持信息的实时性和一致性。
- 关系数据库使组织中所有级别的人（从秘书到副总裁）都可以很容易地获取信息。它们使用 SQL、表单、标准化的报表和临时报表及时地将信息传递给人们。
- 关系数据库是作为信息服务器后端设计的。这意味着大部分人将不会直接操作数据库，而是操作另一个层的软件。这个软件从数据库中获得信息，然后根据每个人的需要来修改信息。
- 关系数据库使得人们在需要使用信息的时候能够获得当前信息。

现在，关系数据库有哪些变化

- 除了已经描述的大型数据库之外，现在还有许多处理较少信息量的小型数据库。这些数据库可以供个人使用或者几个人分享。
- 关系数据库非常成功，易于使用，使用它的应用程序要比最初设想的应用范围广泛了许多。
- 现在许多人直接使用数据库，而不是通过另一个软件来使用数据库。
- 许多人更愿意将自己的数据保存在数据库中。他们觉得关系数据库为处理各种类型的数据提供了一个实用且高效的框架。

1-3 为什么学习 SQL

SQL 应用在 100 多种软件产品中。一旦学会了 SQL，我们能够使用所有这些产品。当然，我们需要稍微熟悉每种产品的特性，很快你就会熟悉它们并且知道如何使用它们。我们可以重复地多次使用这一组技巧。

主要 SQL 产品	其他 SQL 产品(以及基于 SQL 的产品)
Oracle	4th Dimension
Microsoft SQL Server	SQL Base
Microsoft Access	CSQL
MySQL	FileMaker PRO
DB2(IBM Data Server)	Helix Database
Informix	ODBC
PostgreSQL	Ingres
Sybase	MonetDB
Microsoft Visual FoxPro	H2

(续)

主要 SQL 产品	其他 SQL 产品(以及基于 SQL 的产品)
NonStop SQL	MaxDB
Dataphor	VMDS
Teradata	TimesTen
	Openbase
	eXtremeDB
	Interbase
	OpenEdge ABL
	SmallSQL
	Lintier SQL DMBS
	Derby
	Adabas D
	Greenplum Database
	HSQldb
	Alpha_Five
	One \$ DB
	ScimoreDB
	Pervasive PSQL
	Gladius DB
	Daffodil database
	solidDB
	(及更多产品)

SQL 得到广泛使用的原因之一是：相对于其他许多计算机语言来说，它易于学习。另一个原因是：它打开了通向关系数据库的大门，而关系数据库则提供了许多优点。有一些人说 SQL 是关系数据库最好的特性，并且 SQL 是关系数据库获得成功的原因。另一些人则说是关系数据库使得 SQL 获得成功。大部分人认同 SQL 和关系数据库是一个成功组合。

SQL 是最成功的声明式计算机语言——我们可以告诉这种语言我们想要什么，而不是告诉它如何得到我们想要的东西。虽然也有其他一些声明式语言和报表生成工具，但它们大多在功能上有更大的局限性。SQL 更强大，可以应用于更多的场合。

SQL 可以帮助我们从数据库获取信息，而这些信息对于不懂 SQL 语言的用户来说是无法获取的。学会 SQL 可以帮助我们迅速学习和掌握基于 SQL 的众多产品。

最后(不要告诉你的老板)，学习 SQL 是愉快而充满了乐趣的过程。它可以拓展我们的思路并且给我们以思考问题的新工具。我们可能开始从一个新的视角看到一些事物。