

三位世界顶级数据库顾问和讲师、Microsoft MVP合力撰写，国际SQL标准委员会负责人作序推荐，多位资深数据库专家一致好评

通过通俗易懂的语言、大量的示例展示利用SQL解决现实中的问题来传授编写高质量查询语句的方法，示例囊括市面上的多种主流数据库产品

Effective SQL

61 Specific Ways to Write Better SQL

Effective SQL

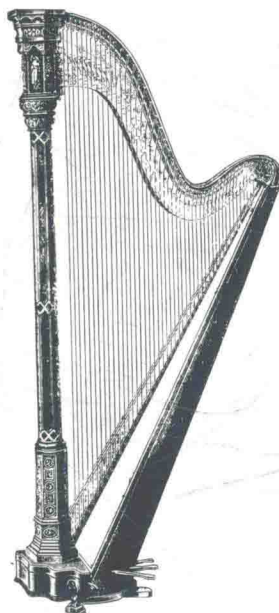
编写高质量SQL语句的61个有效方法

[法] 约翰·L. 维卡斯 (John L. Viescas)

[加] 道格拉斯·J. 斯蒂尔 (Douglas J. Steele) 著

[美] 本·G. 克洛西尔 (Ben G. Clothier)

文浩 译



机械工业出版社
China Machine Press

Effective SQL
61 Specific Ways to Write Better SQL

Effective SQL

编写高质量SQL语句的61个有效方法

[法] 约翰·L. 维卡斯 (John L. Viescas)

[加] 道格拉斯·J. 斯蒂尔 (Douglas J. Steele) 著

[美] 本·G. 克洛西尔 (Ben G. Clothier)

文浩 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Effective SQL : 编写高质量 SQL 语句的 61 个有效方法 / (法) 约翰·L. 维卡斯 (John L. Viescas) 等著; 文浩译. —北京: 机械工业出版社, 2018.6

(Effective 系列丛书)

书名原文: Effective SQL: 61 Specific Ways to Write Better SQL

ISBN 978-7-111-60066-4

I. E… II. ① 约… ② 文… III. 关系数据库系统—程序设计 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2018) 第 117712 号

本书版权登记号: 图字 01-2017-0906

Authorized translation from the English language edition, entitled Effective SQL: 61 Specific Ways to Write Better SQL, ISBN: 9780134578897 by John L. Viescas, Douglas J. Steele, Ben G. Clothier, published by Pearson Education, Inc., Copyright © 2017 Pearson Education, Inc.

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

Chinese simplified language edition published by China Machine Press, Copyright © 2018.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括香港、澳门特别行政区及台湾地区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

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

Effective SQL

编写高质量 SQL 语句的 61 个有效方法

出版发行: 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: 缪 杰

责任校对: 殷 虹

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2018 年 7 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 16.25

书 号: ISBN 978-7-111-60066-4

定 价: 69.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

Praise 本书赞誉

“鉴于作者在业内的声誉，这本书想必会让人印象深刻，可实际仍超出预期，我彻底被它震撼了。大部分 SQL 书籍只告诉你“怎么做”，而本书还会告诉你“为什么要这样做”。大部分 SQL 书籍将数据库设计与实现分开，而本书却将数据库设计原则与 SQL 实现的各个方面相结合。大部分 SQL 书籍都陈放在我的书架上，而本书永远放在我的办公桌上。”

——Roger Carlson, Microsoft Access MVP (2006~2015)

“学习 SQL 的基础知识很容易，但是要构建准确高效的 SQL 语句却非常困难，尤其是针对需求复杂的核心系统。但现在，有了这本书，编写高效的 SQL 语句会比以前快很多，无论你使用的是哪种 DBMS。”

——Craig S. Mullins, Mullins Consulting 公司, DB2 金牌顾问与 IBM 顶级分析师

“这本书可谓精妙绝伦。全书用初学者也能读懂的浅易语言编写，同时，那些技巧与窍门使行业专家也能从中受益。本书适合于任何专注数据库设计、管理或编程的人，他们的书架上就应该放着这样一本书。”

——Graham Mandeno, 数据库顾问与 Microsoft MVP (1996~2015)

“对于任何关系型数据库或 SQL 数据库的程序员与设计师来说，本书都是一本卓越的读物——语言通俗易懂，示例知行合一。示例囊括了市面上主流的数据库产品，包括 Oracle、DB2、SQL Server、MySQL 和 PostgreSQL。书中向读者展示如何运用各种高级技巧解决诸如层次数据和计数表之类的复杂问题，同时讲解在 SQL 语句中使用 GROUP BY、EXISTS、IN、关联与非关联子查询、窗口函数和关联查询时的内部工作原理以及对性能产生的影响。书中的技巧举世无双，示例妙趣横生，绝对不同凡响。”

——Tim Quinlan, 数据库架构师与 Oracle 认证 DBA

“这本书适合有不同 SQL 方言工作经验的人。每个方法都是独立的，你可以按需阅读。自 1992 年以来，我一直从事与 SQL 相关的各种工作，想来也算经验丰富，但尽管如此，

我仍然从本书中学到了一些新的技巧。”

——Tom Moreau 博士，SQL Server MVP (2001~2012)

“这本书讲述了 SQL 的使用技巧，内容紧凑、直观有力、语言通俗易懂——通过用 SQL 解决现实中的问题来传授如何编写查询语句，并且解释了“数据如何存储”与“数据如何被查询”之间的关系，以获得准确高效的查询结果。”

——Kenneth D. Snell 博士，数据库咨询师与前 Microsoft Access MVP

“到目前为止，还没有一本书能够说清楚如何从初级数据库管理员成长为高级数据库管理员，这个问题一直困扰着大家。本书结合现实中真实存在的问题，引导你如何从基础的结构化查询语言 (Structured Query Language, SQL) 进阶到运用更为高级的技巧来解决问题，它就像学习路线图、指南、教练，或者说像破解古埃及文明的“罗塞塔石碑”一样神奇。与其瞎忙或到处寻找答案，不如帮自己一个忙：直接买这本书吧。这样你不仅能学到甚至连数据库顾问都要花数年才能想出来的各种解决方案，还能详细了解不同数据库做法迥异的原因。为了节省时间、精力，避免徒劳无益，买这本书就对了。”

——Dave Stokes，MySQL 社区管理员，Oracle 公司职员

“本书是数据库开发人员必不可少的一本书。书中展示了如何使用强大的 SQL 一步一步地解决现实中的问题。作者使用深入浅出的语言详细比较了各种解决方案的优缺点。众所周知，在 SQL 中完成一件事有很多种方法，作者解释了为什么某些特定的查询语句会比其他查询语句更有效率。我最喜欢的是每一条结尾的总结，其中再次强调了该条的重点并提醒读者哪些是需要注意的陷阱。向所有从事数据库开发的同行们强烈推荐本书。”

——Leo (theDBguyTM)，UtterAccess 论坛版主与 Microsoft Access MVP

“这本书讲述了如何编写高效 SQL 以及解决特定问题的不同解决方案，不仅适合普通的开发人员阅读，而且适合高级 DBA 阅读。在我看来，阅读本书是非常有必要的。还有一个原因是，这本书涵盖了大多数常用的关系数据库管理系统，如果有 RDBMS 迁移的需求，本书刚好能派上用场。作者做了件很了不起的事情，我表示由衷的感谢。”

——Vivek Sharma，Oracle 亚太地区混合云解决方案，核心云技术专家

The Translator's Words 译者序

大多数软件开发人员都会接触数据库开发，SQL 的语法虽然很容易学习，但是要写出高效的 SQL 语句却很困难。首先你必须了解项目需求，然后还要熟悉你所使用的数据库产品，最后才能有针对性地写出有用的 SQL 查询。但实际上能做到以上两点的开发人员很少，因为开发人员往往更熟悉项目需求，而疏于数据库与 SQL 技能；DBA 熟悉数据库的各种调优技能，但不太了解项目各方面的需求，编写 SQL 也会感觉心余力绌。作为一本经验总结性的著作，本书将帮助开发人员摆脱这个窘境，是开发人员甚至 DBA 不可或缺的手边书。

在翻译这本书的过程中，我深深地体会到作者的良苦用心，作者擅长运用通俗易懂的语言阐述自己的观点，并配合贴近真实项目的案例佐证修改之后 SQL 的高效与精确。翻译中，我往往会受作者一言启发，犹如醍醐灌顶，恍然大悟！建议大家在阅读每一条方法时着重理解作者要解决的问题，当项目中遇到类似的问题时，再回过头来重新习读，这样可能效果更好。

很荣幸能够有机会承担本书的翻译工作。在翻译过程中，经常会为一句话、一个术语进行反复讨论，到处查找资料，力图使本书的翻译能正确、贴切地反映原文的意思，同时注意使句子、段落符合中国人的语言习惯。真挚地希望你能从本书中获益，这是作者的初衷，也是我的愿望！

由于我的能力有限，表达方式也有所欠缺，因此译文中不可避免会有疏漏和不足之处，希望广大读者多多指正，我将不胜感激。

文浩

序 *Foreword*

从成为国际标准数据库查询语言以来，SQL 在 30 年中被众多数据库产品采用。如今，SQL 已经无处不在。高性能事务处理系统、智能手机应用程序或 Web 应用程序的后端系统，甚至有一类因不支持（或者说之前不支持）SQL 而被称为 NoSQL 的数据库也在使用 SQL。NoSQL 数据库已经添加了 SQL 接口，“No”的含义现在被解释为“不仅”是 SQL。

由于 SQL 的普及，你很可能在不同产品或者开发环境中接触过 SQL。SQL 有一点为人所诟病（或许是对的）：虽然在不同数据库产品之间极其相似，但还是存在一些细微的差别。这些差别是由解读标准的不同、开发风格的不同以及底层架构的不同导致的。为了理解这些差别，使用示例对照的方式比较不同 SQL 方言的细微差别是非常有帮助的。本书就像是一块解读 SQL 查询的“罗塞塔石碑”，书中展示了如何使用不同数据库方言编写查询语句，并解释了不同方言之间的差别。

我常说，犯错是最好的学习方法，正所谓“吃一堑，长一智”。这么说的理由是，懂得最多的人往往犯过的错也最多，并且知道从别人的错误中吸取教训。本书包含了一些不完整或不正确的 SQL 查询示例，作者也解释了它们为什么是不完整或不正确的。你可以抓住这些机会从别人的错误中学习。

SQL 是一种既强大又复杂的数据库语言。作为一名数据库顾问兼美国与国际 SQL 标准委员会的成员，我看到很多查询都没有充分利用好 SQL 的特性。能够完全理解 SQL 的强大与复杂性并充分利用 SQL 特性的应用程序开发人员，不仅可以构建健壮的应用程序，还可以加快开发的速度。本书中的 61 个特殊示例可以帮助你习得这些知识。

——Keith W. Hare

JCC 咨询公司高级顾问

美国 SQL 标准委员会 INCITS DM32.2 副主席

国际 SQL 标准委员会 ISO/IEC JTC1 SC32 WG3 负责人

结构化查询语言 (Structured Query Language) 简称 SQL, 是与大多数数据库系统通信的标准语言。如果你正在阅读本书并希望从数据库系统中获取信息, 那么就需要使用 SQL。

本书面向从事 SQL 工作的开发人员和初级数据库管理员 (DBA), 适合对 SQL 的基本语法比较熟悉并且希望再获得一些有用的技巧以便更高效地使用 SQL 语言的人。而且我们发现, 当从计算机编程惯用的基于过程的方式转变为基于集合的方式来解决问题时, 所需的思维方式是截然不同的。

关系数据库管理系统 (RDBMS) 是一种软件应用程序, 用于创建、维护、修改和操作关系数据库。许多关系数据库系统也提供了用于操作数据库中数据的客户端工具。关系数据库系统自出现以来一直在不断发展, 并随着硬件技术和操作系统环境的进步变得更加完善和强大。

SQL 简史

IBM 研究员 Edgar F. Codd 博士 (1923—2003) 在 1969 年首次提出了关系数据库模型。他在 20 世纪 60 年代后期研究了处理大量数据的新方法, 并开始思考如何应用数学原理解决遇到的各种问题。

自 1970 年 Codd 博士向世界提出了关系数据库模型之后, 许多组织 (如大学和研究实验室) 开始致力于开发一种语言, 用作支持关系数据库的基础。20 世纪 70 年代中期几种不同的语言出现, 其中一个正是来自位于加利福尼亚州圣何塞的 IBM 圣特雷莎研究实验室的努力。

20 世纪 70 年代初, IBM 启动了一个名为 System/R 的重大科研项目, 旨在证明关系模型的可行性, 并希望在设计和实现关系数据库方面获得一些经验。1974~1975 年, 他们的初次实验获得成功, 创建了一个关系数据库的迷你原型。

在开发关系数据库的同时, 研究人员也在努力定义数据库语言。1974 年, Donald Chamberlin 博士和他的同事发明了结构化英语查询语言 (Structured English Query Language,

SEQUEL), 这门语言允许用户使用清晰易懂的英语句子操作关系数据库。原型数据库 SEQUEL-XRM 的初步成功, 激励着 Chamberlin 和他的同事, 他们决定继续研究。1976~1977 年, 他们把语言名称从 SEQUEL 修改为 SEQUEL/2, 但是不巧, SEQUEL 缩写已经被别人使用了, 出于法律原因, 他们不得不将 SEQUEL 更名为 SQL (结构化查询语言或 SQL 查询语言)。时至今日, 虽然大家已经广泛接受了官方发音 “ess-cue-el”, 但是许多人还是将 SQL 读作 “sequel”。

虽然 IBM 的 System/R 项目与 SQL 语言证明关系数据库是可行的, 但是由于当时的硬件技术水平太低, 这款产品并没有商用。

1977 年, 加利福尼亚州门罗公园的一群工程师创办了 Relational Software 公司, 他们开发了一套基于 SQL 的关系数据库产品并命名为 Oracle。1979 年, Relational Software 公司发布了这款产品, 使之成为第一个商业化的关系数据库产品。Oracle 的一大优势是能运行在 Digital 的 VAX 小型机上, 而不是昂贵的 IBM 大型机。Relational Software 公司此后更名为 Oracle 公司, 成为 RDBMS 软件领域领先的厂商之一。

大约在同一时间, 来自加利福尼亚大学伯克利分校计算机实验室的 Michael Stonebraker、Eugene Wong 和其他几位教授也在研究关系数据库技术。他们也开发了一个关系数据库的原型, 命名为 Ingres。Ingres 使用一种称为查询语言 (QUEL) 的数据库语言, 它比 SQL 结构更为清晰, 而且使用了更少的类似英语的词句。但是, 很明显当时 SQL 正在慢慢变成数据库标准语言, 所以 Ingres 最终改为基于 SQL 的关系数据库。1980 年, 这几位教授离开伯克利, 成立了 Relational Technology 公司。1981 年, 他们发布了 Ingres 的第一个商业版本。Relational Technology 公司之后经历了几次变革。Ingres 之前为 Computer Associates International 公司所有, 现在属于 Actian 公司, 但不管怎么样, Ingres 仍然是行业数一数二的数据库产品。

1981 年, IBM 也宣布开发自己的关系数据库, 名为 SQL/DataSys (SQL / DS), 并于 1982 年发布。1983 年, IBM 公司推出了一种名为 Database 2 (DB2) 的新型关系数据库产品, 它可以在安装 IBM 的主流 MVS 操作系统的 IBM 大型机上运行。1985 年第一次发布之后, DB2 就成为 IBM 首屈一指的关系数据库, 其技术已经融入 IBM 整个产品线中。

随着数据库语言的不断发展, 语言标准化的想法在数据库社区中呼声渐长。但是, 由谁来制定标准, 或者应该参照哪个方言来制定一直没有达成共识, 所以每个开发商还在不断开发与改进他们自己的数据库产品, 并希望有朝一日自己的数据库方言能够成为行业标准。

许多开发商都会根据用户的反馈和需求在自己的数据库方言中增加新的元素, 从而形成了早期非正规的标准。相对现在的标准来说它只能算是很小的一部分, 因为它只包含各种 SQL 方言中相似的部分。然而, 正是这一小部分规范 (尽管不是很完善) 为数据库用户提供了一套权威的评判标准, 通过这些标准可以判断市场上的各种数据库, 并且用户可以在不同数据库间切换。

1982 年,为了满足关系数据库语言标准化日益增长的需求,美国国家标准协会(ANSI)委托旗下 X3 组织数据库技术委员会 X3H2 制定一个标准。一番周折之后(包括对 SQL 的大量改进),委员会才意识到新的标准与市面上主要的 SQL 方言不兼容,对 SQL 的修改也并没有带来显著提高,不兼容是肯定的。最后,他们只保留了数据库厂商都能够遵循的最小集合。

1986 年,美国国家标准协会(ANSI)正式采纳了“ANSI X3.135-1986 数据库语言 SQL”这个标准,也就是我们熟知的 SQL/86 标准。本质上,它就是将各种 SQL 方言之间的相似部分提取出来并标准化,其实许多数据库厂商早就实现了。尽管这个标准还不太完善,但是至少为数据库语言的未来奠定了坚实的基础。

1987 年,国际标准化组织(ISO)正式批准了自己的数据库标准(几乎和 ANSI SQL/86 一样)作为国际标准,标准文件为“ISO 9075 : 1987 数据库语言 SQL”(这两个标准通常都简称为 SQL/86)。国际数据库社区厂商与美国数据库厂商使用了相同的数据库标准,他们可以联合工作。尽管 SQL 已经标准化了,但是语言离完成还很远。

很快 SQL/86 受到舆论、政府和行业专家的多方批评,例如著名的数据库大师 C. J. Date 认为 SQL 语法很冗余(相同的查询有不同的写法),缺乏对某些关系运算符以及引用完整性的支持。

为了解决这个问题,ISO 和 ANSI 修改了标准,只采用了原来标准中的核心部分,而且特别增加了对数据引用完整性的支持。1989 年中后期,ISO 发布了名为“ISO 9075 : 1989 数据库语言 SQL 与完整性增强”的标准文件。同年末期,ANSI 也发布了名为“X3.135-1989 数据库语言 SQL 与完整性增强”的标准文件。这两个标准通常简称为 SQL/89 标准。

人们普遍认为,SQL/86 和 SQL/89 标准缺乏数据库系统应有的最基本功能。例如,标准中没有定义如何修改或删除数据库结构,也没有定义修改数据库的安全权限的方法,即便所有数据库厂商都已经在自己的产品中实现了这些功能(例如,使用 CREATE 语法创建了数据库对象,但是标准中没有定义 ALTER 或 DROP 语法来修改这个数据库对象)。

ANSI 和 ISO 不想再制定一个所有数据库厂商都能支持的“最小集合”,他们把工作重点放在了 SQL 的修订上,这也许能让 SQL 变得更加完善。新标准(SQL/92)包括了大多数数据库厂商已经广泛支持的功能,但是也包括一些只有少数数据库厂商支持的功能,甚至还有一些从未实现过的功能。

1992 年 10 月,ANSI 与 ISO 分别发布了新的 SQL 标准文件“X3.135-1992 数据库语言 SQL”和“ISO/IEC 9075 : 1992 数据库语言 SQL”。SQL/92 标准的内容远多于 SQL/89,而且涉及的范围更广。例如,新标准中定义了创建数据库结构之后修改它的方法,增加了对字符串、日期和时间的操作运算符,并定义了额外的安全特性。相比之前的标准,SQL/92 算是个重要的里程碑。

当数据库厂商致力于实现 SQL/92 标准中的功能时,他们也在开发和实现自己的功能,从而在 SQL 标准的基础上扩展新的功能。虽然这些扩展(例如支持更多的数据类型,在 SQL/92 中只有 6 种数据类型)能为产品提供更多的功能,而且还可以区分其他的数据库产

品，但同时也存在很多隐患。最主要的问题是，扩展导致每个数据库厂商的 SQL 方言与原始标准相差甚远，这就破坏了应用程序在不同数据库之间运行的可移植性。

1997 年，ANSI 的 X3 组织更名为国家信息技术标准委员会（NCITS），原来负责 SQL 标准的技术委员会现在称为 ANSI NCITS-H2。由于 SQL 标准复杂性的激增，ANSI 和 ISO 标准委员会将标准分成 12 个有独立编号的部分和一个附录，这样做的原因是，他们已经开始 SQL3 的工作（这样命名是因为它是标准的第三个主要修订版本），分开之后每个部分的工作可以同时进行。自 1997 年至今，共制定了两个新的部分。

本书中所有的内容都是基于最新版的 ISO SQL 标准：数据库语言——SQL/ 基础（文档 ISO/IEC 9075-2:2011）——目前已经在大多数商业数据库系统中实现。ANSI 也采用了 ISO 标准，这个标准因此成了真正的国际化标准。我们还参考了 IBM DB2、Microsoft Access、Microsoft SQL Server、MySQL、Oracle 和 PostgreSQL 的最新文档，在需要解释不同数据库产品的不同语法时会用到。虽然你在本书中学习的大多数 SQL 不是针对某个特定的数据库软件产品，但是我们会在必要的时候向你展示与特定数据库产品相关的示例。

我们关注的数据库

虽然上一节提到了 SQL 标准，但是这并不意味着所有数据库系统都是一样的。DB-Engines 网站收集整理了大量的数据库系统，并按照流行度每月公布一次排名，网址为 <http://db-engines.com/en/ranking/relational+dbms>。

有 6 个数据库系统持续保持排行榜最热门 DBMS 好几个月，按字母顺序依次为（括号里的版本是书中测试使用的数据库版本）：

- 1) IBM DB2 (Linux、UNIX 和 Windows 版的 DB2, v10.5.700.368)
- 2) Microsoft Access (Microsoft Access 2007, 向上兼容 2010、2013、2016 或更高版本)
- 3) Microsoft SQL Server (Microsoft SQL Server 2012 11.0.5343.0)
- 4) MySQL (MySQL 社区版 5.7.11)
- 5) Oracle 数据库 (Oracle Database 11g Express 版本 11.2.0.2.0)
- 6) PostgreSQL (PostgreSQL 9.5.2)

这并不意味着书中的代码不能在这 6 种数据库之外的数据库中运行，我们只是没有在其他数据库或其他版本上测试而已。阅读本书时，你可能会注意到，在针对不同数据库时，我们都添加了说明（标有注意的部分）。注意的内容只涵盖这 6 种数据库。如果你使用的是其他数据库，运行示例遇到任何问题时，请查阅相关数据库文档。

示例数据库

为了阐明确书中的概念，我们创建了一些用于举例的数据库，如下：

1) 啤酒风格：这是一个很有趣的游戏，根据 Michael Larson 的书《Beer: What to Drink Next》(Sterling Epicure, 2014) 中提到的，对 89 种不同风格的啤酒进行分类。

2) 演出代理：这个数据库用来管理演员、代理、客户和预订。你将使用类似的设计来处理待办事项和酒店预订。

3) 菜谱：这个数据库可以用来保存和管理所有喜爱的食谱。

4) 销售订单：这个是典型的商店订单数据库，销售自行车、滑板以及相关配件。

5) 学生课程：这个数据库用来管理学生的信息、他们参加的课程，以及课程成绩。

我们还为某些条目提供了特殊的示例数据库，在这些条目里面就可以找到与这些示例数据库相关的创建代码。你可以在 GitHub 上找到与本书相关的数据库表结构和示例数据。

在 GitHub 上查找示例

很多技术书籍都会附带一个包含全书代码的光盘。我们觉得这太局限了，所以决定把书中的所有示例都放在 GitHub 上，网址为 <https://github.com/TexanInParis/Effective-SQL>。

GitHub 上，最上层的 6 个文件夹是我们关注的几个数据库系统。6 个文件夹中分别包含与书中章节相对应的 10 个文件夹和 1 个示例数据库文件夹。

在这 10 个文件夹中分别包含一些 SQL 代码文件，文件名与对应章节内的代码清单的编号一致。注意，这些代码不一定适用于所有数据库。如果存在不兼容某些数据库的情况，在每个章节文件夹下面我们都创建了一个名为 README 的文件，用来记录不同数据库之间的差异。对于 Microsoft Access，README 文件记录的是哪个示例数据库包含对应章节的代码清单。

GitHub 根目录还有一个 Listings.xlsx 文件，这个文件记录了每个示例数据库对应哪些代码清单，同时还记录了这些代码清单能够适用于 6 个数据库系统中的哪几个。

每个示例数据库文件夹中都包含许多 SQL 文件，除了 Microsoft Access 示例数据库，Microsoft Access 文件夹下面包含的都是 Microsoft Access 2007 格式的 .accdb 文件。我们使用 Microsoft Access 2007 格式的原因是它能兼容版本 12 (2007) 以上的所有版本。示例数据库文件夹下面的某些文件是用来创建数据库表结构的，另外一些是用来插入示例数据库的数据的。(注意，书中的某些条目依赖于某些特定的数据。这些条目需要的数据库结构和数据有时就包含在章节的代码清单中。)



注意 在准备本书的代码清单时，有些时候会遇到每行 63 个字符的限制，这个限制是纸质书规定的。某些代码清单可能存在编辑错误。当你不确定的时候，请参考我们在 GitHub 上的代码清单，上面的代码清单都测试过，我们确信都是正确的。

章节概要

正如本书的书名所示，书中共包含了 61 个方法。每个方法条目都是相对独立的。如果你使用某个条目里面的内容，不需要阅读其他条目。当然，也有特例，有些条目里的内容依赖于其他条目。在这种情况下，我们尽可能地提供它们之间的依赖关系，但绝大多数情况下，我们都提供了它们之间的引用关系，所以你可以自行查阅。

虽然每个条目正如我们说的那样相互独立，但是仍然可以分成不同的主题。我们把它分成以下 10 个主题：

1) 数据模型设计：如果你面对的是糟糕的数据模型设计，想要编写高效的 SQL 语句是不太可能的，该章包含一些基本的数据模型设计原则。如果你的数据库设计违反了该章中讨论的任何原则，你应当找出问题所在并修复。

2) 可编程性与索引设计：仅靠良好的逻辑数据模型设计是写不出高效的 SQL 语句的。你还必须保证设计被正确实施，否则你可能会发现使用 SQL 从数据库中高效查询数据的能力会大打折扣。该章将帮助你了解索引的重要性，以及如何正确地使用它。

3) 当你不能改变设计时：有时，就算你使出浑身解数，也没办法处理在你控制之外的外部数据。该章将帮助你解决这个难题。

4) 过滤与查找数据：从数据库中查询或过滤数据的能力是 SQL 的主要功能之一。该章探讨了几种不同的方法，可以用来查询所需的数据。

5) 聚合：SQL 标准一直都有聚合数据的功能。但是，通常你得提供诸如“按客户统计总数”“按天统计订单数”或者“按月统计每个类型的平均销售额”等的统计方式。它总是跟在“按”“以”或者“每”后面。该章将介绍高效聚合数据的技术，其中还会介绍如何使用窗口函数解决更为复杂的聚合问题。

6) 子查询：使用子查询有很多种方法。本章将介绍在 SQL 语句中使用子查询获得更多灵活性的各种方法。

7) 获取与分析元数据：有时候会觉得信息不够。你需要有关数据的信息，甚至可能需要有关数据如何被查询到的信息。这种情况下，使用 SQL 获取数据库元数据可能会有帮助。该章可能会针对某个数据库产品，但我们希望书中提供了足够的信息，以便你可以将这些原则应用到数据库系统中。

8) 笛卡儿积：笛卡儿积是将一个表中的所有行与第二个表中的所有行组合在一起的结果。虽然可能不像其他表连接查询那样常用，但该章中展示的案例如果不使用笛卡儿积是没办法解决的。

9) 计数表：另外一个有用的工具就是计数表，通常是具有单列连续递增数字的表、单列连续递增日期，或者更复杂一点的用作记录行列转换函数所需列名的表。笛卡儿积依赖于表中实际存储的数据，而计数表则不受此限制。该章介绍了几种只有使用计数表才能解决的问题。

10) 层次数据建模：在关系数据库中，层次数据建模并不罕见。糟糕的是，它恰好是 SQL 的一个软肋。该章将帮助你如何在数据规范化和查询与维护元数据的便利性之间进行权衡。

每种数据库系统都有一些用来计算或者操作日期与时间的函数。同样，每种数据库系统也有自己的日期与时间的数据库类型和计算规则。由于这些差异的存在，我们特意提供了一个附录，在使用日期和时间的时候提供给你一些帮助。虽然我们确信已经总结了所有的数据类型和运算符，但还是强烈建议你参考相关的数据库文档，了解你使用的函数的特定语法。

致 谢 *Acknowledgements*

有位著名的政客曾经说过：“养育一个孩子，需要全村的力量。”如果你想写一本书，不管是技术还是非技术方面的，都需要一个强大的团队，才能把你的“孩子”变成一本好书。

首先，要感谢 Trina MacDonald，我们的策划编辑兼项目经理。自 John 的畅销书《SQL Queries for Mere Mortals》出版以来，他就一直缠着 John，不仅促成了 John 与 Effective 软件开发系列的合作，还带领这个项目成功度过各个阶段。John 组建了一个真正国际化的团队，这是完成本书的关键，他个人非常感谢团队成员的辛勤付出。此外，特别感谢 Tom Wickerath 在项目早期与后期技术审校期间提供的帮助。

之后，Trina 把我们介绍给了 Songlin Qiu，我们的开发编辑，她帮助我们了解了写一本 Effective 系列书籍的来龙去脉。非常感谢 Songlin 的指导。

接下来，Trina 邀请了一大批技术编辑，他们辛勤地审校并调试了数百个例子，给了我们很多有价值的反馈。感谢负责 MySQL 的 Morgan Tocker 与 Dave Stokes，负责 PostgreSQL 的 Richard Broersma Jr.，负责 IBM DB2 的 Craig Mullins 以及负责 Oracle 的 Vivek Sharma。感谢大家！

在这期间，此图书系列的编辑也是畅销书《Effective C++》的作者 Scott Meyers，也给我们提供了很多宝贵的意见。希望本书能让 Effective 系列之父引以为豪。

然后，感谢来自制作团队的 Julie Nahil、Anna Popick 和 Barbara Wood，他们帮助我们 将书装订成册以便出版。没有他们，我们不可能完成这项工作。

最后，感谢我们的家人，多少个漫长的夜晚他们独自承受，而我们却忙于编辑初稿和整理示例。真心感谢他们的理解与耐心！

——John Viescas

法国巴黎

——Doug Steele

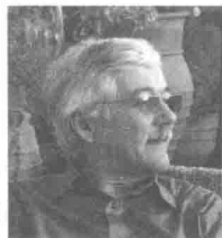
加拿大安大略省凯瑟琳街

——Ben Clothier

美国德克萨斯州康弗斯

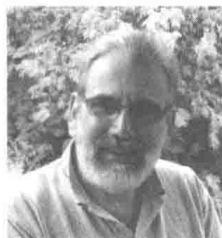
About the Authors 关于作者

John L. Viescas 是一位拥有超过 45 年工作经验的独立数据库顾问。他最初是一名系统分析师，曾为 IBM 大型机系统设计过大规模的数据库应用程序。他在德克萨斯州达拉斯做过 6 年的数据研究工作，在那里，他带领着一个超过 30 人的团队，负责 IBM 大型机数据库产品的研究、开发与客户支持工作。在达拉斯做数据研究期间，John 完成了他在德克萨斯大学达拉斯分校的商业与金融学专业的学业，并以优异的成绩获得该专业的学士学位。



1988 年，John 加入 Tandem 计算机公司，负责美国西部销售片区的数据库营销程序的开发与实施。他是 Tandem 公司关系数据库管理系统 NonStop SQL 技术研讨会的发起者。John 的第一本书《A Quick Reference Guide to SQL》(Microsoft Press, 1989)，作为一个研究项目记录了 SQL 语法在 ANSI-86 SQL 标准、IBM 的 DB2、Microsoft 的 SQL Server、Oracle 公司的 Oracle 以及 Tandem 的 NonStop SQL 之间的相似性。他从 Tandem 休假期间，写了《Running Microsoft® Access》(Microsoft Press, 1992) 的第 1 版。之后又陆陆续续写了 4 本 Running 系列的书籍，3 本《Microsoft® Office Access Inside Out》(Microsoft Press, 2003、2007 和 2010) 即 Running 系列的后续篇，以及《Building Microsoft® Access Applications》(Microsoft Press, 2005)。他也是畅销书《SQL Queries for Mere Mortals》(第 3 版，Addison-Wesley, 2014) 的作者。John 目前是连续被评为 Microsoft Access MVP (最有价值专家) 时间最长的纪录保持者，1993~2015 年每年都获此殊荣。John 与他的妻子一直住在法国巴黎，已经 30 多年了。

Douglas J. Steele 与计算机打交道已经超过 45 年了，用过大型机和小型计算机 (最初甚至还用过穿孔卡!)。2012 年退休之前，他曾在一家大型国际石油公司就职超过 31 年。虽然在职业生涯的后期他主要从事系统中心配置管理器 (System Center Configuration Manager, SCCM) 任务序列的开发，将 Windows 7 安装在全球超过 100 000 台计算机上，但是



他大部分时间关注的焦点还是数据库与数据建模。

Doug 连续 17 年被评为 Microsoft MVP，他发表了许多关于 Access 的文章，是《Microsoft® Access® Solutions: Tips, Tricks, and Secrets from Microsoft® Access® MVPs》(Wiley, 2010) 的合著者，担任过许多书的技术编辑。

Doug 拥有滑铁卢大学（加拿大安大略省）系统设计工程专业硕士学位，他的研究主要集中在为特殊计算机用户设计用户界面（但话又说回来，20 世纪 70 年代末，很少有计算机用户是普通的！）。这项研究源于他的音乐背景（他是多伦多皇家音乐学院钢琴表演的会员）。他还痴迷于酿酒，曾毕业于尼亚加拉学院（安大略省尼亚加拉湖）的酿酒师与酿酒厂运营管理专业。

Doug 和他的爱妻一直住在加拿大安大略省凯瑟琳街，已经超过 34 年了。你可以通过以下邮箱联系他：AccessMVPHelp@gmail.com。

Ben G. Clothier 是 IT Impact 公司的解决方案架构师，IT Impact 公司是一家在 Access 和 SQL Server 领域领先的开发商，总部位于伊利诺伊州芝加哥市。他曾担任很多公司的独立顾问，包括 J Street Technology 和 Advisicon 等知名公司，负责把一个小型的 Access 项目从个人解决方案扩大为整个公司业务线的整体解决方案。代表项目包括为水泥公司做的工作记录跟踪与库存管理功能，为保险公司做的医疗保险计划生成器，还有为某国际航运公司做的订单管理功能。Ben 是 UtterAccess 论坛的管理员，并且与 Teresa Hennig、George Hepworth 和 Doug Yudovich 合著了《Professional Access® 2013 Programming》(Wiley, 2013)，与 Tim Runcie 和 George Hepworth 合著了《Microsoft® Access® in a SharePoint World》(Advisicon, 2011)，也是《Microsoft® Access® 2010 Programmer's Reference》(Wiley, 2010) 的特邀作者。他拥有 Microsoft SQL Server 2012 解决方案助理认证，也是 MySQL 5.0 认证开发人员。自 2009 年以来，他一直被微软评为 MVP。



Ben 和妻子 Suzanne、儿子 Harry 住在德克萨斯州的圣安东尼奥。