Broadview®
www.broadview.com.cn

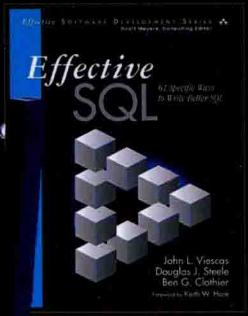
Pearson

· 原味精品书系 ·

Effective SQL（英文版）

编写高质量SQL语句的61条有效方法

Effective SQL
61 Specific Ways to Write Better SQL


Effective Software Development Series
Scott Meyers, Consulting Editor
Effective SQL
61 Specific Ways to Write Better SQL
John L. Viescas
Douglas J. Steele
Ben G. Clothier
Foreword by Keith W. Hare

[法] John L. Viescas
[加] Douglas J. Steele 著
[美] Ben G. Clothier

中国工信出版集团 電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Effective SQL（英文版）

编写高质量SQL语句的61条有效方法

Effective SQL

61 Specific Ways to Write Better SQL

[法] John L. Viescas
[加] Douglas J. Steele 著
[美] Ben G. Clothier

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内容简介

本书结合了对案例问题的分析与解决方案的实现，向读者展示了如何通过 SQL 编写解决复杂问题，以及如何通过数据库设计简化数据管理。这是一本将 SQL 高级功能实践与 SQL 实际应用示例完美结合的、面向所有版本 SQL 方言（其中包括 IBM DB2、Access、SQL Server、MySQL、Oracle、PostgreSQL）的编程指南。此外，SQL 语法只是本书的一小部分内容，从数据库设计与优化，到分层数据与元数据管理，本书都有所涉及。如果读者有一定的 SQL 基础，那么本书将助你成为顶级 SQL 专家。

版权贸易合同登记号　图字：01-2017-3321

图书在版编目（CIP）数据

Effective SQL：编写高质量 SQL 语句的 61 条有效方法 = Effective SQL: 61 Specific Ways to Write Better SQL：英文 /（法）约翰 · L · 威克斯（John L.Viescas），（加）道格拉斯 · J · 斯蒂尔（Douglas J.Steele），（美）本 · G · 克洛希尔（Ben G.Clothier）著. — 北京：电子工业出版社，2017.8
（原味精品书系）
ISBN 978-7-121-32284-6
I. ①E… II. ①约…②道…③本… III. ①SQL 语言－英文 IV. ①TP311.132.3

中国版本图书馆 CIP 数据核字（2017）第 176715 号

责任编辑：张春雨
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　　　　北京市海淀区万寿路 173 信箱　　　邮编：100036
开　　本：787×980　1/16　　印张：21.75　　字数：430.8 千字
版　　次：2017 年 8 月第 1 版
印　　次：2017 年 8 月第 1 次印刷
定　　价：89.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。
质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。
本书咨询联系方式：（010）51260888-819　faq@phei.com.cn。

序言

SQL 在成为国际标准数据库语言的 30 余年里，已经在众多数据库产品中得以实现。现在 SQL 无处不在，高性能事务处理系统、智能手机应用程序以及后端 Web 接口程序中都在使用 SQL 语言。甚至有一类 NoSQL 数据库，其共同特点是（或曾经是）它们不使用 SQL。随着 NoSQL 数据库增加了 SQL 语言接口，现在 NoSQL 中“No”的意思是“不仅仅是”SQL（Not Only SQL）。

由于 SQL 的普遍性，你很可能在很多产品和系统环境中都使用过 SQL。有一个针对 SQL 语言的诟病（也许是对的），虽然 SQL 语言在不同的数据库产品中是类似的，但还是有细微的差别。这些差别源于对 SQL 标准的不同理解、不同开发模式或者不同的底层架构。通过实例来对比不同产品 SQL 方言中的细微差别，对于理解这些 SQL 差异是非常有用的。本书为 SQL 查询提供了一个罗塞塔石碑（解释古埃及象形文字的可靠线索），向我们展示了如何使用不同的 SQL 方言编写 SQL 查询并解释了 SQL 方言中的差异。

我经常强调最好的学习方法是从错误中学习。也就是说，那些懂得从自己和别人的错误中学习的人，才能够获得更多。本书包含了一些不完整、不正确的 SQL 查询实例，并向读者解释了这些 SQL 不完整、不正确的原因，以此通过别人的错误进行学习。

SQL 是一种功能强大且复杂的数据库语言。作为美国和国际 SQL 标准委员会的数据库顾问和参与者，我见过很多查询没有充分利用 SQL 的能力。充分理解 SQL 的语言能力以及复杂性的程序开发人员，不仅可以充分利用 SQL 的功能构建性能良好的应用程序，还能高效地开发应用程序。本书提供了 61 个具体实例来帮助大家学习。

Keith W. Hare

JCC 咨询有限公司高级顾问

美国 INCITS dm32.2—— SQL 标准委员会副主席

ISO/IEC JTC1 SC32 WG3 国际 SQL 标准委员会召集人

致谢

一位著名的政治家曾经说过："抚养一个孩子需要一个村庄"。如果你曾经写过一本书，不管是技术方面的还是其他方面的，你会明白把你的"孩子"变成一本成功的书需要一个伟大的团队。

首先，非常感谢我们的策划编辑和项目经理 Trina MacDonald。Trina 不仅缠着 John 使其坚持将 *SQL Queries for Mere Mortals* 一书打造成为 *Effectiue Software Development* 系列丛书，还带领着团队度过了重重难关。John 组建了一支真正国际化的团队来完成这本书，感谢大家工作上的不辞辛劳，尤为感谢 Tom Wickerath 在项目前后期给予的协助和技术审校。

Trina 向我们推荐了本书的内容指导 Songlin Qiu，在他的帮助下我们了解了编写 *Effectiue Software Development* 系列丛书的各个细节，非常感谢 Songlin 给予我们的指导。

Trina 精心挑选了庞大的技术编辑团队，他们辛苦地审核和调试了我们的数百个例子，并提供了很多有价值的反馈。感谢负责 MySQL 的 Morgan Tocker 和 Dave Stokes；感谢负责 PostgreSQL 的 Richard Broersma Jr.；感谢负责 IBM DB2 和 Oracle 的 Craig Mullins；感谢负责 Oracle 的 Vivek Sharma。

在写书的过程中，系列丛书编辑、畅销书 *Effective C++, Third Edition* 的作者 Scott Meyers，也参与到我们的团队中，在如何把本书变成 *Effectiue Software Development* 系列丛书方面给了我们很多宝贵的建议。希望本书可以成为 *Effectiue Software Development* 系列丛书的典范。

Julie Nahil、Anna Popick 的制作团队和 Barbara Wood 帮助我们使书符合出版的要求。我们不能没有你们！

最后，非常感谢我们的家庭忍受我们在漫长的夜晚写稿和调试例子。他们持久的耐心值得大书特书！

John Viescas
法国，巴黎

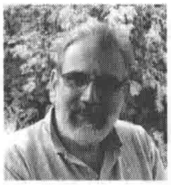
Douglas Steele
加拿大，安大略省，圣凯瑟琳

Ben Clothier
美国，得克萨斯州，康弗斯

关于作者

John L. Viescas 是一位有着超过 45 年从业经验的独立数据库顾问。他的职业生涯从系统分析师开始，为 IBM 大型机系统设计大型数据库应用。他在达拉斯的应用数据研究中心工作了 6 年，在那里他带领 30 多名员工负责 IBM 大型计算机数据库产品的研究、开发以及客户支持工作。在应用数据研究工作期间，John 完成了达拉斯得克萨斯大学的商业金融学位，并以优异的成绩毕业。

John 1988 年加入 Tandem 计算机公司，在那里他负责在 Tandem 公司美国西部销售区开发和实施营销方案的数据库。他开发并交付了用于技术研讨会的关系型数据库管理系统——NonStop SQL。John 1989 年写了他的第一本书，*A Quick Reference Guide to SQL*（Microsoft Press，1989 年），该书是一本对比了 ANSI-86 SQL 标准、IBM 的 DB2、微软的 SQL Server、甲骨文公司的 Oracle，以及 Tandem 公司的 NonStop SQL 之间的语法相似性的研究类书籍。他于 1992 年从 Tandem 公司公休时写了 *Running Microsoft® Access*（第 1 版，Microsoft Press，1992 年）。他已经写了 4 个版本的 *Running Microsoft® Access*，以及 *Running* 系列的后续作品——3 个版本的 *Microsoft Office Access Inside Out*（Microsoft Press，2003 年、2007 年、2010 年）和 *Building Microsoft® Access Applications*（Microsoft Press，2005 年）。他也是畅销书籍 *SQL Queries for Mere Mortals®, Third Edition*（Addison Wesley，2014 年）的作者。John 目前保持着连续多年被微软授予微软数据库管理系统最有价值专家（MVP，Most Valuable Professional）的纪录（1993 年至 2015 年）。John 与他的妻子在法国巴黎定居了 30 多年。



Douglas J. Steele 从事包括大型机和个人机在内的计算机相关工作超过 45 年（是的，他一开始是用穿孔卡的！）。在 2012 年退休前，他在一家大型国际石油公司工作了 31 年。尽管他职业生涯的高光时刻是通过发展 SCCM 任务序列将 Windows 7 推广到全球超过 10 万台电脑上，但是数据库和数据建模是他的主要工作方向。

Douglas 被微软认证为最有价值专家（MVP）超过 17 年，他撰写了大量关于 Access 数据库的文章，Douglas 是 *Microsoft ® Access ® Solutions: Tips, Tricks, and Secrets from Microsoft Access MVPs*（Wiley，2010 年）的合著者，也是很多书的技术编辑。

Douglas 是滑铁卢大学系统设计工程硕士，专注于设计非传统电脑用户的用户界面研究。（当然，在 20 世纪 70 年代末，很少有人是传统的电脑用户！）他的专业研究源于他的音乐背景（他拥有多伦多英国皇家音乐学院钢琴演奏准学士学位）。他还痴迷于啤酒并毕业于尼亚加拉学院酿酒及啤酒厂操作管理专业（滨湖尼亚加拉，安大略省，加拿大）。

Douglas 和他的妻子在加拿大安大略省圣凯瑟琳定居超过 34 年。读者可以通过邮箱 mvphelp@gmail.com 联系 Douglas。

Ben G. Clothier 是芝加哥首屈一指的 Access 和 SQL Server 开发企业 IT Impact 公司的解决方案架构师。他曾在著名的 J Street Technology 和 Advisicon 公司做过自由顾问，主要从事从小型个人解决方案到公司全业务应用程序 Access 项目的相关工作。值得一提的项目包括：一家水泥公司的工作跟踪和库存管理系统、给保险商使用的医疗保险计划生成器以及国际航运公司的订单管理系统。Ben 在 UtterAccess 是系统管理员，并且和 Teresa Hennig、George Hepworth、Doug Yudovich 合著了 *Professional Access® 2013 Programming*（Wiley，2013 年）；并与 Tim Runcie 和 George Hepworth 一起合著了 *Microsoft® Access in a SharePoint World*（Advisicon，2011 年）；Ben 还是 *Microsoft® Access® 2010 Programmer's Reference*（Wiley，2010 年）一书的特约作者。他拥有微软 SQL Server 2012 解决方案认证和 MySQL 5 认证开发者等证书。从 2009 年开始 Ben 一直是微软的最有价值专家（MVP）。

Ben 和他的妻子 Suzanne、儿子 Harry 住在美国得克萨斯州的圣安东尼奥。

关于技术编辑

Richard Anthony Broersma Jr. 是 Mangan 公司（美国加利福尼亚州长滩市）的系统工程师，他有 11 年的 PostgreSQL 数据库应用程序开发经验。

Craig S. Mullins 是一位数据管理战略家、研究员和顾问。他是 Mullins Consulting 公司的总裁兼首席顾问，Craig 被 IBM 任命为 IBM 金牌顾问和 IBM 首席分析师。Craig 在各类数据库系统开发方面拥有超过 30 年的经验，从 DB2 的第一个版本开始他就在使用。你可能会通过 Craig 的畅销书知道他：*DB2 Developer's Guide, Sixth Edition*（IBM，2012 年）和 *Database Administration: The Complete Guide to DBA Practices and Procedures, Second Edition*（Addison-Wesley，2012 年）。

Vivek Sharma 目前是甲骨文公司的 Oracle 核心技术与混合云解决方案部门的特邀技术专家。他在 Oracle 技术方面有着超过 15 年的工作经验，Vivek 在 Oracle 公司开始了自己的职业生涯，在成为全职 Oracle 数据库性能架构师之前，他的主要工作是 Oracle 报表系统的开发。作为 Oracle 数据库专家，Vivek 大部分时间在帮助客户获得最佳的 Oracle 系统并确保用户在数据库方面的投资是值得的，他还是著名的甲骨文精英工程交换和服务器技术合作项目的成员。他在 2012 年和 2015 年间成为甲骨文印度用户团体“年度发言人”。他会在他的博客 viveklsharma.wordpress.com 以及 OTN（www.oracle.com/technetwork/index.html）上发布相关的 Oracle 数据库技术文章。

Dave Stokes 是甲骨文公司的 MySQL 社区经理。他以前曾是 MySQL AB 和 Sun 的 MySQL 认证经理。他所工作的企业按字母顺序从美国心脏协会（American Heart Association）排到施乐公司（Xerox），工作从反潜工作者（anti-submarine warfare）排到互联网开发者（Web developer）。

Morgan Tocker 是甲骨文公司 MySQL 服务器的产品经理。他曾从事过各种工作，包括支持、培训与社区工作。Morgan 定居于加拿大多伦多。

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **提交勘误**：您对书中内容的修改意见可在 提交勘误 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动**：在页面下方 读者评论 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：*http://www.broadview.com.cn/32284*

目录

Introduction

Structured Query Language, or SQL, is the standard language for communicating with most database systems. We assume that because you are looking at this book, you have a need to get information from a database system that uses SQL.

This book is targeted at the application developers and junior database administrators (DBAs) who regularly work with SQL as part of their jobs. We assume that you are already familiar with the basic SQL syntax and focus on providing useful tips to get the most out of the SQL language. We have found that the mindset required is quite different from what works for computer programming as we move away from a procedural-based approach to solving problems toward a set-based approach.

A relational database management system (RDBMS) is a software application program you use to create, maintain, modify, and manipulate a relational database. Many RDBMS programs also provide the tools you need to create end-user applications that interact with the data stored in the database. RDBMS programs have continually evolved since their first appearance, and they are becoming more full-featured and powerful as advances occur in hardware technology and operating environments.

A Brief History of SQL

Dr. Edgar F. Codd (1923–2003), an IBM research scientist, first conceived the relational database model in 1969. He was looking into new ways to handle large amounts of data in the late 1960s and began thinking of how to apply mathematical principles to solve the myriad problems he had been encountering.

After Dr. Codd presented the relational database model to the world in 1970, organizations such as universities and research laboratories began efforts to develop a language that could be used as the foundation of a database system that supported the relational model. Initial work led to the development of several different languages in the early to mid-1970s. One such effort occurred at IBM's Santa Teresa Research Laboratory in San Jose, California.

IBM began a major research project in the early 1970s called System/R, intending to prove the viability of the relational model and to gain some experience in designing and implementing a relational database. Their initial endeavors between 1974 and 1975 proved successful, and they managed to produce a minimal prototype of a relational database.

At the same time they were working on developing a relational database, researchers were also working to define a database language. In 1974, Dr. Donald Chamberlin and his colleagues developed Structured English Query Language (SEQUEL), which allowed users to query a relational database using clearly defined English-style sentences. The initial success of their prototype database, SEQUEL-XRM, encouraged Dr. Chamberlin and his staff to continue their research. They revised SEQUEL into SEQUEL/2 between 1976 and 1977, but they had to change the name SEQUEL to SQL (Structured Query Language or SQL Query Language) for legal reasons—someone else had already used the acronym SEQUEL. To this day, many people still pronounce SQL as "sequel," although the widely accepted "official" pronunciation is "ess-cue-el."

Although IBM's System/R and SQL proved that relational databases were feasible, hardware technology at the time was not sufficiently powerful to make the product appealing to businesses.

In 1977 a group of engineers in Menlo Park, California, formed Relational Software, Inc., for the purpose of building a new relational database product based on SQL that they called Oracle. Relational Software shipped its product in 1979, providing the first commercially available RDBMS. One of Oracle's advantages was that it ran on Digital's VAX minicomputers instead of the more expensive IBM mainframes. Relational Software has since been renamed Oracle Corporation and is one of the leading vendors of RDBMS software.

At roughly the same time, Michael Stonebraker, Eugene Wong, and several other professors at the University of California's Berkeley computer laboratories were also researching relational database technology. They developed a prototype relational database that they named Ingres. Ingres included a database language called Query Language (QUEL), which was much more structured than SQL but made less use of English-like statements. However, it became clear that SQL was emerging as the standard database language, so Ingres was eventually converted to an SQL-based RDBMS. Several professors left Berkeley in 1980 to form Relational Technology, Inc., and in 1981 they announced the first commercial version of Ingres. Relational Technology has gone through several transformations. Formerly owned by Computer Associates International, Inc., and now part of Actian, Ingres is still one of the leading database products in the industry today.

Meanwhile, IBM announced its own RDBMS called SQL/Data System (SQL/DS) in 1981 and began shipping it in 1982. In 1983, the company introduced a new RDBMS product called Database 2 (DB2), which could be used on IBM mainframes using IBM's mainstream MVS operating system. First shipped in 1985, DB2 has become IBM's premier RDBMS, and its technology has been incorporated into the entire IBM product line.

With the flurry of activity surrounding the development of database languages, the idea of standardization was tossed about within the database community. However, no consensus or agreement as to who should set the standard or which dialect it should be based upon was ever reached, so each vendor continued to develop and improve its own database product in the hope that it—and, by extension, its dialect of SQL—would become the industry standard.

Customer feedback and demand drove many vendors to include certain elements in their SQL dialects, and in time an unofficial standard emerged. It was a small specification by today's standards, as it encompassed only those elements that were similar across the various SQL dialects. However, this specification (such as it was) did provide database customers with a core set of criteria by which to judge the various database programs on the market, and it also gave users knowledge that they could leverage from one database program to another.

In 1982, the American National Standards Institute (ANSI) responded to the growing need for an official relational database language standard by commissioning its X3 organization's database technical committee, X3H2, to develop a proposal for such a standard. After much effort (which included many improvements to SQL), the committee realized that its new standard had become incompatible with existing major SQL dialects, and the changes made to SQL did not improve it significantly enough to warrant the incompatibilities. As a result, they reverted to what was really just a minimal set of "least common denominator" requirements to which database vendors could conform.

ANSI ratified this standard, "ANSI X3.135-1986 Database Language SQL," which became commonly known as SQL/86, in 1986. In essence, it conferred official status on the elements that were similar among the various SQL dialects and that many database vendors had already implemented. Although the committee was aware of its shortcomings, at least the new standard provided a specific foundation from which the language and its implementations could be developed further.

The International Organization for Standardization (ISO) approved its own document (which corresponded exactly with ANSI SQL/86) as an international standard in 1987 and published it as "ISO 9075:1987 Database Language SQL." (Both standards are still often referred to as just SQL/86.) The international database vendor community could now work from the same standards as vendors in the United States. Despite the fact that SQL gained the status of an official standard, the language was far from being complete.

SQL/86 was soon criticized in public reviews, by the government, and by industry pundits such as C. J. Date for problems such as redundancy within the SQL syntax (there were several ways to define the same query), lack of support for certain relational operators, and lack of referential integrity.

Both ISO and ANSI adopted refined versions of their standards in an attempt to address the criticisms, especially with respect to referential integrity. ISO published "ISO 9075: 1989 Database Language SQL with Integrity Enhancement" in mid-1989, and ANSI adopted its "X3.135-1989 Database Language SQL with Integrity Enhancement," also often referred to as SQL/89, late that same year.

It was generally recognized that SQL/86 and SQL/89 lacked some of the most fundamental features needed for a successful database system. For example, neither standard specified how to make changes to the database structure once it was defined. It was not possible to modify or delete any structural component, or to make changes to the security of the database, despite the fact that all vendors provided ways to do this in their commercial products. (For example, you could CREATE a database object, but no ALTER or DROP syntax was defined.)

Not wanting to provide yet another "least common denominator" standard, both ANSI and ISO continued working on major revisions to SQL that would make it a complete and robust language. The new version (SQL/92) would include features that most major database vendors had already widely implemented, but it also included features that had not yet gained wide acceptance, as well as new features that were substantially beyond those currently implemented.

ANSI and ISO published their new SQL Standards—"X3.135-1992 Database Language SQL" and "ISO/IEC 9075:1992 Database Language