

## <<SQL编程风格>>

### 图书基本信息

书名：<<SQL编程风格>>

13位ISBN编号：9787115185822

10位ISBN编号：7115185824

出版时间：2008-10

出版时间：人民邮电出版社

作者：塞科

页数：194

译者：米全喜

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;SQL编程风格&gt;&gt;

## 前言

本书不是一本SQL入门书。

真的，如果你需要的是学习如何使用SQL进行编程，有其他更好的书。

本书应该是你买的第二本书，而不是第一本。

本书假设你已经能够编写一定水平的SQL，并且希望进一步提高。

如果你想要学习SQL编程技巧，可以买一本我编写的Joe Celkos SQL FOR Smarties(2005年第3版)。

在本书中，我想教给读者的，是如何以逻辑和说明性的方式编程，而不是以过程化或面向对象的方式——也就是要“用查询的思维看数据库”。

绝大多数SQL程序员都是在有了几年的过程化语言或面向对象语言编程经验之后才开始接触SQL的。他们拿到某个SQL产品，然后只能自学，使用的书都是“SQL for Brain-Dead Morons(SQL傻瓜书)”、“Le锄SQL in ten: Easy Lessons or Five Hard Ones(用10节容易的或5节复杂的课程学会SQL)”或其他更烂的书。

这太荒唐了!成为熟练的木匠或厨师都至少需要5年。

为什么你会相信人们在一个周末内就能变成SQL高手?

他们会变成糟糕的SQL程序员，只会使用本地SQL产品中的方言，并带有他们从前使用的编程语言的浓重口音。

可怕的是这些人常常不知道自己是拙劣的程序员。

一个极端情况是，整个公司的人都做得很差，他们从来没有见过高手。

## <<SQL编程风格>>

### 内容概要

《图灵程序设计丛书·SQL编程风格》针对数据库的设计与编程提出了一系列规则和建议，内容涵盖命名规范、代码版式、键的设计、数据编码方案、编码风格、视图和存储过程的使用以及SQL中的思考方式和一些试探法等多方面。

这些规则都给出了原理说明和例外情况，并列举了大量示例。

通过阅读《图灵程序设计丛书·SQL编程风格》，读者可以加深对SQL思维方式的理解决，改善SQL编程风格，并编写出可读性强、可移植且易于维护的SQL代码。

此外，书中的规则对于公司内部制定编程规范也具有很好的借鉴作用。

《图灵程序设计丛书·SQL编程风格》适合数据库管理人员和开发人员阅读，也可作为高等院校计算机专业师生的参考教材。

## 作者简介

Joe Celko世界著名的数据库专家，曾经担任ANSI SQL标准委员会成员达10年之久，他也是世界上读者数量最多的SQL图书作者之一。

他曾撰写过一系列专栏，并通过他的新闻组支持了数据库编程技术以及ANSI / ISO标准的发展。

除本书外，他还撰写了多部SQL经典著作，包括《SQL解惑（第2版）》（人民邮电出版社，2008）和《SQL权威指南》（即将由人民邮电出版社出版）。

## 书籍目录

第1章 名称与数据元素1.1 名称1.1.1 注意名称长度1.1.2 在名称中避免使用所有特殊字符1.1.3 避免使用引号分隔标识符1.1.4 实施大写规则以避免大小写区分问题1.2 遵循ISO-11179标准命名规范1.2.1 SQL的ISO-111791.2.2 抽象级别1.2.3 避免使用描述性前缀1.2.4 制定标准化的后缀1.2.5 表和视图名称应当是遵循业界标准的、集合、类或复数名称1.2.6 相关名基本上也要遵循与其他名称相同的命名规则1.2.7 关系表名应当是常用描述术语1.2.8 元数据模式访问对象的名称可以包含结构信息1.3 命名数据元素时遇到的问题1.3.1 避免模糊名称1.3.2 避免名称在不同的地方改变1.3.3 不要使用专有暴露的物理定位符第2章 字体、标点和间距2.1 版式与代码2.1.1 名称中只使用大小写字母、数字和下划线2.1.2 列名、参数和变量等标量小写2.1.3 模式对象名首字母大写2.1.4 保留字大写2.1.5 避免使用驼峰命名法2.2 单词间距2.3 遵循规范标点规则2.4 使用完全保留字2.5 如果在使用的SQL产品中有标准保留字,就不要使用专有保留字2.6 如果有标准语句,就不要使用专有语句2.7 疏排版面的隔空白道和垂直间距2.8 缩进2.9 使用行间距将语句分组第3章 数据定义语言3.1 将默认值放到合适的地方3.2 默认值的类型应当与列的类型相同3.3 不要使用专有数据类型3.4 将PRIMARYKEY声明放在CREATETABLE语句的开头3.5 将列按照逻辑顺序排列并按照逻辑组聚合3.6 将参考约束和操作在数据类型下面缩进3.7 在产品代码中为约束命名3.8 将CHECK ( ) 约束放在所检查的内容附近3.8.1 对数值考虑使用范围约束3.8.2 对于字符值考虑使用LIKE和SIMILARTO约束3.8.3 时间值是有长短的3.8.4 避免使用REAL和FLOAT数据类型3.9 将多列约束尽可能靠近这些列3.10 将表级别的CHECK ( ) 约束放到表声明的最后3.11 对多表约束使用CREATEASSERTION3.12 使CHECK ( ) 约束的目的唯一3.13 每个表都必须有键才能称为表3.13.1 自动编号不是关系型键3.13.2 文件不是表3.13.3 键的属性3.14 不要分割属性3.14.1 分割为多个表3.14.2 分割为多个列3.14.3 分割为多个行3.15 不要对RDBMS使用面向对象的设计3.15.1 表不是对象实例3.15.2 对RDBMS不要使用EAV设计第4章 尺度与测量4.1 测度论4.1.1 范围与颗粒度4.1.2 范围4.1.3 颗粒度、准确度和精度4.2 尺度类型4.2.1 名义尺度4.2.2 种类尺度4.2.3 绝对尺度4.2.4 顺序尺度4.2.5 级别尺度4.2.6 间距尺度4.2.7 比例尺度4.3 使用尺度4.4 尺度转换4.5 导出单位4.6 标点与标准单位4.7 在数据库中使用尺度的一般准则第5章 数据编码方案5.1 不好的编码方案5.2 编码方案类型5.2.1 枚举编码5.2.2 测量编码5.2.3 缩写编码5.2.4 算法编码5.2.5 层次编码5.2.6 向量编码5.2.7 拼接编码5.3 设计编码方案的一般准则5.3.1 现有的编码标准5.3.2 允许扩展5.3.3 使用显式的丢失值避免NULL5.3.4 为终端用户转换编码5.3.5 在数据库中保存编码5.4 多字符集第6章 编码选择6.1 选择标准构造,不要选择专有构造6.1.1 使用标准OUTERJOIN语法6.1.2 中缀INNERJOIN和CORSSJOIN语法是可选的,但是很好用6.1.3 使用ISO时间语法6.1.4 使用标准和可移植的函数6.2 选择紧凑格式,不要选择松散格式6.2.1 避免使用多余的括号6.2.2 使用CASE系列表达式6.2.3 避免使用冗余表达式6.2.4 寻找紧凑格式6.3 使用注释6.3.1 存储过程6.3.2 控制语句注释6.3.3 对子句的注释6.4 避免优化器提示6.5 触发器的优先级不应当高于DRI操作6.6 使用SQL存储过程6.7 避免在数据库中使用用户定义函数和扩展6.7.1 多语言问题6.7.2 可移植性问题6.7.3 优化问题6.8 避免使用过度的辅助索引6.9 避免使用关联子查询6.10 避免使用UNION6.11 测试SQL6.11.1 测试NULL所有可能的组合6.11.2 检查并测试所有的CHECK ( ) 约束6.11.3 注意字符列6.11.4 测试大小第7章 如何使用视图7.1 视图的命名规范与表一样7.2 视图提供行和列级别的安全性7.3 视图确保了有效访问路径7.4 视图对用户隐藏了复杂性7.5 视图确保了正确的数据派生7.6 视图将表和 / 或列重新命名7.7 视图实施复杂的完整性约束7.8 可更新的视图7.8.1 WITHCHECKOPTION子句7.8.2 INSTEADOF触发器7.9 每个视图都要有创建的原因7.10 避免视图的数量快速增长7.11 将视图与基表同步7.12 不恰当地使用视图7.12.1 用于域支持的视图7.12.2 单个解决方案的视图7.12.3 不要为每个基表都创建视图7.13 学习使用物化的视图第8章 如何编写存储过程8.1 大多数SQL4GL都不是用于应用程序的8.2 基本软件工程8.2.1 内聚8.2.2 耦合8.3 使用传统的结构化编程8.4 避免可移植性问题8.4.1 避免创建临时表8.4.2 避免使用游标8.4.3 面向集合的构造优于过程化代码8.5 标量与结构化参数的对比8.6 避免使用动态SQL8.6.1 性能8.6.2 SQL注入第9章 试探法9.1 将规格说明表达为清晰的语句9.2 在名词的后面加上“.....的集合”9.3 从问题语句中删除行为动词9.4 仍然可以使用存根9.5 不要担心数据的显示9.6 第一次尝试需要特别处理9.6.1 不要舍不得扔掉你对DDL的第一次尝试9.6.2 保存你对DML的第一次尝试9.7 不要以方框和箭头的方式思考9.8 画圆圈和集合图9.9 学习方言9.10 假设WHERE子句是“巨型变形虫”9.11 使用新闻

## <<SQL编程风格>>

组和因特网第10章 以SQL的方式思考10.1 不好的SQL编程方式与过程化语言10.2 把列当作字段思考10.3 以过程化而不是说明性的方式思考10.4 模式应该看起来像输入格式附录A 资源附录B 参考文献索引

## 章节摘录

第1章 名称与数据元素 1.1 名称 以前，每个程序员都有一套自己的命名规范。但是，他们常常都太有创造性了。我特别喜欢举的一个例子是，有一个人使用某类主题词作为他的COBOL段名：一段程序可能使用国家名，另外一段可能使用花卉，等等。即使就程序员而言，这显然也是很奇怪的行为，但是很多程序员的个人命名系统只有他们自己明白，别人都无法理解。

例如，我使用的第一个FORTRAN版本只允许6个字母的名称，所以我变得善于使用和发明6个字母的名称。开始编程时使用弱类型或无类型语言的程序员们都喜欢使用匈牙利表示法（参见Leszynski和Reddick）。

。老的习惯很难放弃。

当软件工程变成规范后，每个公司都制定了自己的命名规范，并使用某种数据字典实施这些规范。

。使用最广泛的一套规则可能要算是由美国国防部建立的MIL STD8320.1，但它在联邦政府之外却从未流行起来。

这与先前缺乏有效组织的体系相比，已经有了很大进步，但是每个公司都有很大的不同：有些对于名称构造有正式的规则，而另外一些则只是将赋予数据元素的第一个名称登记一下。

现在，我们有了ISO-11179标准，它正变得越来越普遍，是某些政府工作所需要的，并且正在被放入到数据储存库产品中。

一些工具和大量标准化编码方案也被放入到了这个标准中。

考虑到这一点，并考虑到XML是一种标准交换格式，ISO-11179在今后将成为元数据参考的方法。

## <<SQL编程风格>>

### 媒体关注与评论

“ Joe Celko是当今数据库界最著名的代表之一，他已经写了不少SQL编程的畅销书。但是。本书非常与众不同，它将教你如何转变思维方式，以逻辑和说明性的方式编程。成为一流的SQL开发人员。”  
——sQL-Server-Performance.com



## &lt;&lt;SQL编程风格&gt;&gt;

## 编辑推荐

《图灵程序设计丛书·SQL编程风格》中，世界级SQL专家Joe Celko针对数据库的设计与编程提出了一系列规则和建议，内容涵盖命名规范、代码版式、键的设计、数据编码方案、编码风格、SQL中的思考方式等多个方面。

可以作为软件公司内部编程规范的基础。

书中讲述了如何编写标准、高效、易于维护的SQL代码。

更重要的是。

还教授读者如何像优秀的SQL程序员那样思考，用查询的思维方式来理解数据库，从而大大改善SQL编程风格并提高SQL编程水平。

世界级大师的SQL编程规范，讲述如何编写标准、高效、易于维护的SQL代码，教你像优秀的SQL程序员那样思考。

数据库作为现代软件应用的核心之一，正在发挥越来越重要的作用。

很自然地，SQL在广大程序员的日常工作中也成了不可或缺的技术。

学会SQL并不难，但是要成为优秀的SQL程序员就绝非易事了。

大部分程序员都是在学习并从事了过程化或面向对象编程之后才转到SQL。

因此往往带有浓重的口音，而且常常缺乏自知之明。

<<SQL编程风格>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>