

- 基于 **SQL Server 2014**，由浅入深地详细介绍**最新 SQL** 技术，让读者始终站在**技术最前沿**
- 深入分析 SQL 的**运行机制**和**效率提升方法**，使读者知其然，更知其所以然
- 丰富的图表和**可操作性极强**的实例，便于读者理解，并能够**举一反三**

锋利的

SQL

(第2版)



张洪举 王晓文 著



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

锋利的



张洪举 王晓文 著



(第2版)

人民邮电出版社
北京

图书在版编目(CIP)数据

锋利的SQL / 张洪举, 王晓文著. — 2版. — 北京 : 人民邮电出版社, 2015. 5
ISBN 978-7-115-38871-1

I. ①锋… II. ①张… ②王… III. ①关系数据库系统 IV. ①TP311.138

中国版本图书馆CIP数据核字(2015)第074985号

内 容 提 要

本书从基础、开发、性能调整和实战4方面介绍了SQL技术及其应用, 包括数据库与架构、表管理、索引管理、基本查询、子查询、联接和APPLY运算符、操作结果集、窗口计算和表旋转、数据修改、视图、游标、存储过程、触发器、用户自定义函数、事务处理、并发访问控制、查询的优化与执行等内容。

本书既覆盖了改善效率和性能的普通SQL技术, 也深入探讨了SQL新技术, 更包含一些实用的查询解决方案, 希望本书能够成为引领读者进入SQL查询殿堂的捷径。此外, 本书在介绍各种查询语法时, 更注重对查询逻辑思维方式的引导和介绍, 帮助读者举一反三, 提升动手解决实际问题的能力。

本书可以作为SQL入门书, 也可以作为SQL程序员、DBA的参考书。

-
- ◆ 著 张洪举 王晓文
责任编辑 杨海玲
责任印制 张佳莹 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 29.5
字数: 697千字
印数: 4 001—6 500册
- 2015年5月第2版
2015年5月北京第1次印刷
-

定价: 69.00元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

前言

本书第1版是在4年前出版的，至今仍在被众多的SQL爱好者追捧，甚至有人不辞辛劳地逐页扫描，上传至网络进行分享。本书第1版是基于SQL Server 2008编写的，时光荏苒，目前SQL Server 2014已经发布，窗口函数等新功能不断丰富和发展，先前需要编写很复杂的代码才能完成的工作，现在寥寥几行就能把问题解决。无论是从代码可读性还是执行效率方面看，SQL Server 2014都值得我们去深入学习。因此，我们在第1版的基础上，对书的内容重新进行了补充和完善。此外，这次我们还补充了一些像“三值逻辑”这样的基础知识，因为在实际工作中，我经常发现，由于存在NULL值比较问题，导致查询结果不正确，费尽千辛万苦地分析代码，最终却发现犯了低级错误。

当初之所以要写这本书，主要出于两方面的原因：一是伴随着各种数据库技术日新月异的发展，无论是哪种数据库产品，想用有限的篇幅去描述它的全貌，几乎都是不可能完成的任务，所以我就考虑能否抽取出各种数据库产品中一些大家共同关心的内容，进行深入细致的挖掘，而SQL无疑是这方面的首选；二是在与一些开发公司的合作中，发现公司间的SQL应用情况的差异也很大，一些公司出于产品的可移植性考虑，拒绝使用一些新的SQL技术，甚至尽量避免在服务器上部署存储过程。所以我希望在深入地讨论一些常用SQL技术的同时，也尽可能多介绍一些SQL的新技术，从而消除大家对新技术的恐惧感，对新技术的推广能够起到一定的作用。

本书特点

本书既覆盖了改善效率和性能普通SQL技术，也深入探讨了SQL新技术，更包含一些实用的查询解决方案。希望本书能够成为引领读者进入SQL查询殿堂的捷径。

本书的内容是基于SQL Server数据库产品进行讨论的。不过，由于大家都在遵循ANSI-SQL标准，所以无论是哪种数据库产品的SQL，彼此之间的差别并不大。数据库开发人员在跨越不同的数据库产品时，一般不会遇到什么障碍。当然，本书不特定于某个SQL Server版本，在介绍一些新功能时，会介绍它是在哪个版本添加进来的。并且，本书在改写过程中，在添加新的查询解决方案的同时，保留了先前的解决方案，以便读者能够进行对比分析。

本书在介绍各种查询语法时，更注重对查询逻辑思维方式的引导和介绍，这样可以帮助读者在阅读之后举一反三，提升自己动手解决实际问题的能力。

本书适用读者

本书是按照由浅入深、循序渐进的方式对SQL进行介绍的，既包含了入门知识，也包含了深层次技术的讨论。即使是最基本的查询语句，我们也会尽力为读者提供解决深层次问题的能力。也就是说，同样一个问题，开发人员可以写几十甚至上百行的SQL语句来解决问题，也有可能仅通过一条SELECT语句就可以解决问题。对于SQL而言，虽然代码最简化并不一定是性能最优化，但至少是对思维能力的一种提升。

从这个角度而言,本书可以作为 SQL 入门书,也可以作为 SQL 程序员、DBA 的参考书。

本书内容与结构

本书共 19 章,可大致分为基础篇、开发篇、性能调整篇和实战篇 4 部分。

基础篇包括第 1 章至第 10 章。其中,第 1 章是对查询工具、书写规范等基本内容的介绍,第 2 章至第 5 章是对数据库、表和索引的介绍,第 5 章至第 10 章则介绍了使用 SELECT 进行数据查询和使用 INSERT、UPDATE、DELETE 进行数据修改的各个方面。本次修订新增了 SQL 理论基础,新增函数介绍、架构概念、标识列和序列对象等一致性约束介绍,内存表、选择列表的同时操作概念,查询的逻辑处理步骤的修订,使用 OFFSET-FETCH 限制查询结果和限制查询修改,窗口分布函数和窗口偏移函数、MERGE 与 OUTPUT 等,尤其是第 9 章的窗口计算部分,更是全部重新书写。

开发篇包括第 11 章至第 15 章。如果将 SELECT、INSERT、UPDATE 和 DELETE 作为基本查询语句,则 IF...ELSE、WHILE 和 TRY...CATCH 构造等则可以看作是 SQL 编程语句。在存储过程、触发器等对象中可以通过这些语句实现一些复杂的逻辑处理。如果你曾经是一位使用 VB 或 VC 的程序员,在学习 C/S 或 B/S 编程时,你应当掌握这种服务器端的编程工具,从而将业务逻辑计算合理地分布到服务器和客户端。本次修订对 CLR 存储过程、触发器、函数等以 Visual Studio 2013 为基础重新进行了编写。

性能调整篇包括第 16 章至第 18 章。第 16 章和第 17 章介绍的是事务处理机制和并发访问控制。其实,无论是微软还是 Oracle、IBM,其数据库产品的核心功能都是一样的,即在保证数据完整性的前提下提供最大的并发支持。数据库系统是通过“锁机制”来实现的,数据库引擎都提供有多种粒度的锁定模式,从而允许用户可以根据需要将资源锁定在适当的级别,尽量减少锁定开销。第 18 章则讨论了查询优化器的工作原理,重用查询计划,可以减少额外的编译开销,提高服务器性能。

实战篇仅包含第 19 章的内容,提供了同一时间范围内并发数统计、时间段天数统计、数字范围统计、地域范围内最大数统计等较为常见问题的解决方案。本次修订在原解决方案的基础上新增了窗口函数的解决方案。

系统要求

本书实例中使用的操作系统是 Windows 7,使用的数据库是 SQL Server 2014 开发者版本,开发工具是 Visual Studio 2013。为使 Visual Studio 2013 能够支持 SQL Server 2014 的 SQL 开发,需要将其升级到最新版本。同时,由于 SQL Server 示例数据库在 2014 版本中表结构变动较大,因此对书中的代码重新进行了调整和测试,该示例数据库可从 <http://msftdbprodsamples.codeplex.com/releases/view/125550> 下载。

代码示例

本书中使用的所有代码，请从作者的博客（<http://blog.csdn.net/zhanghongju>）上获取。

致谢

在本书的完成过程中，得到了诸多 SQL Server 技术专家和爱好者的支持与帮助。他们无私和热情的参与，使本书的内容更加实用和更具指导性，在此一并表示感谢。他们是王向东、秦广、魏兰花、凌亚东、王亚羽、陈雨薇、王光辉、高存亭、桑晓红、王新河、张宪国、李联国、韩燕军。

由于时间仓促，加之作者水平有限，书中不足之处在所难免，敬请读者批评指正。

张洪举

2015 年 1 月

SQL Server 2014 示例数据库安装步骤

SQL Server 2014 示例数据库安装步骤具体如下。

(1) 登录 <http://msftdbprodsamples.codeplex.com/releases/view/125550> 网站, 单击 Adventure Works 2014 Full Database Backup.zip 下载数据库文件。

(2) 下载完成后解压缩, 释放后的文件名为 AdventureWorks2014.bak。

(3) 打开 Microsoft SQL Server Management Studio, 在“对象资源管理器”中右键单击“数据库”节点, 在弹出的快捷菜单中选择“还原数据库”菜单项, 如图 1 所示。

(4) 在打开的“还原数据库”窗口中, 单击“设备”后面的“...”按钮, 在打开的对话框中单击“添加”按钮, 选择释放后的数据库备份文件 AdventureWorks2014.bak, 如图 2 所示。依次单击“确定”按钮即可还原数据库。

(5) 在“对象资源管理器”中右键单击“数据库”节点, 在弹出的快捷菜单中选择“刷新”菜单项, 即可看到数据库 AdventureWorks2014。

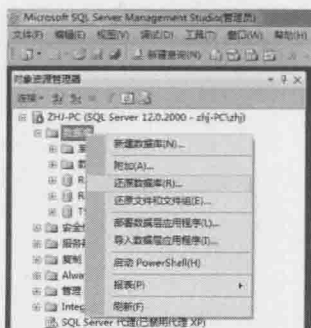


图 1 选择“还原数据库”菜单项



图 2 选择备份介质

目 录

基 础 篇

第 1 章 SQL 简介.....2	1.6.6 字符串串联运算符..... 31
1.1 SQL 的历史起源.....2	1.6.7 一元运算符..... 31
1.1.1 CODASYL.....3	1.7 常用函数..... 31
1.1.2 IMS.....3	1.7.1 聚合函数..... 31
1.1.3 RDBMS 和 SQL.....3	1.7.2 配置函数..... 33
1.1.4 ANSI 和 SQL 方言.....5	1.7.3 游标函数..... 34
1.2 SQL 的理论基础.....5	1.7.4 日期和时间函数..... 35
1.2.1 集合理论.....5	1.7.5 数学函数..... 37
1.2.2 谓词逻辑.....6	1.7.6 数据类型转换函数..... 39
1.2.3 关系模型.....7	1.7.7 字符串函数..... 43
1.3 Transact-SQL 语言的类型..... 10	1.7.8 文本和图像函数..... 45
1.3.1 DDL 语句..... 11	1.7.9 逻辑函数..... 46
1.3.2 DML 语句..... 12	1.8 查询工具..... 46
1.3.3 编程和流控制语句..... 12	1.8.1 Management Studio..... 46
1.3.4 SQL 语句的批处理..... 15	1.8.2 sqlcmd..... 47
1.4 Transact-SQL 语法..... 16	1.9 SQL 书写规范..... 48
1.4.1 标识符..... 16	1.9.1 大小写规范..... 49
1.4.2 数据类型..... 17	1.9.2 使用空格..... 50
1.4.3 函数..... 20	1.9.3 使用缩进..... 51
1.4.4 表达式..... 20	1.9.4 使用垂直空白道..... 51
1.4.5 运算符..... 21	1.9.5 使用分组..... 52
1.4.6 注释..... 21	第 2 章 数据库与架构..... 53
1.4.7 保留关键字..... 22	2.1 SQL Server 数据库基础..... 53
1.5 常量和变量..... 22	2.1.1 数据库的服务方式..... 53
1.5.1 常量..... 22	2.1.2 数据库实例..... 54
1.5.2 变量..... 24	2.1.3 架构与对象..... 55
1.6 运算符..... 26	2.1.4 数据库文件和文件组..... 56
1.6.1 算术运算符..... 26	2.2 创建数据库..... 58
1.6.2 赋值运算符..... 28	2.2.1 CREATE DATABASE 语句
1.6.3 位运算符..... 28	的语法格式..... 58
1.6.4 比较运算符..... 29	2.2.2 创建数据库示例..... 60
1.6.5 逻辑运算符..... 30	2.2.3 判断数据库是否已经存在..... 63

2.3 修改数据库	63	4.1 索引的基础知识.....	95
2.3.1 扩展数据库和文件	63	4.1.1 索引的类型.....	95
2.3.2 向数据库中添加、删除和修改 文件组	64	4.1.2 索引的特征.....	99
2.3.3 收缩数据库和文件	65	4.1.3 常规索引设计规则	99
2.3.4 设置数据库选项	68	4.2 创建索引.....	101
2.3.5 重命名数据库	71	4.2.1 最大索引限制	101
2.4 删除数据库	71	4.2.2 限制索引参与的数据类型	101
2.5 架构管理	72	4.2.3 创建聚集索引.....	102
2.5.1 创建架构	72	4.2.4 创建非聚集索引.....	103
2.5.2 修改架构	74	4.2.5 创建具有包含性列的索引	104
2.5.3 移动对象到一个新架构中.....	75	4.2.6 为计算列创建索引.....	105
2.5.4 删除架构	76	4.3 修改索引.....	107
第3章 表管理	77	4.3.1 禁用索引.....	107
3.1 表的物理存储方式	77	4.3.2 重新组织和重新生成索引	108
3.1.1 数据页	77	4.3.3 设置索引选项.....	110
3.1.2 区	78	4.3.4 重命名索引	111
3.2 创建表	79	4.4 删除索引.....	111
3.2.1 创建基本表	79	第5章 基本查询	112
3.2.2 使用 NULL 约束.....	79	5.1 基本的 SELECT 语句.....	112
3.2.3 使用默认约束和标识列	80	5.1.1 SELECT 语句的结构	112
3.2.4 其他数据完整性设置	86	5.1.2 数据库对象的引用规则	114
3.3 修改表	88	5.2 使用选择列表和表别名.....	115
3.3.1 为表添加新列	89	5.2.1 选择所有列	115
3.3.2 修改表中的列	89	5.2.2 选择特定列	115
3.3.3 删除表中的列	91	5.2.3 使用常量、函数和表达式	116
3.4 重命名和删除表	91	5.2.4 用表别名	121
3.5 临时表	92	5.3 使用 WHERE 子句筛选行.....	121
3.5.1 本地表和全局表	92	5.3.1 用比较搜索条件.....	121
3.5.2 表变量	93	5.3.2 使用范围搜索条件.....	123
3.6 内存优化表	93	5.3.3 使用列表搜索条件.....	123
第4章 索引管理.....	95	5.3.4 使用模式匹配搜索条件.....	125
		5.3.5 使用 NULL 比较搜索条件	127

5.4 使用 GROUP BY 子句进行分组计算 .. 129	6.2.3 含有 NOT IN 和 NOT EXISTS 的 子查询..... 162
5.5 使用 HAVING 子句从中间结果 筛选行..... 131	6.3 含有比较运算符的子查询..... 164
5.6 使用 ORDER BY 子句进行排序 132	6.4 使用 ANY、SOME 或 ALL 关键字 165
5.6.1 指定排序列 132	6.5 使用多层嵌套子查询..... 168
5.6.2 指定排序顺序 133	6.6 子查询应遵循的规则..... 168
5.6.3 指定排序规则 134	第 7 章 联接和 APPLY 运算符..... 172
5.7 使用 TOP 和 OFFSET-FETCH 限制 结果集..... 137	7.1 联接的基本知识..... 172
5.7.1 使用 TOP 选项..... 137	7.1.1 联接的语法格式..... 172
5.7.2 使用 OFFSET-FETCH 筛选 139	7.1.2 联接所使用的逻辑处理阶段..... 173
5.7.3 使用 SET ROWCOUNT 语句 140	7.1.3 列名限定和选择列表的使用 174
5.8 使用 DISTINCT 消除重复行 140	7.1.4 联接条件设定..... 175
5.9 同时操作..... 142	7.2 交叉联接..... 175
5.10 查询的逻辑处理..... 143	7.2.1 交叉联接的语法格式..... 176
5.10.1 逻辑处理过程简介 143	7.2.2 使用交叉联接查询全部数据..... 176
5.10.2 步骤 1: FROM..... 147	7.2.3 使用交叉联接优化查询性能..... 179
5.10.3 步骤 2: 使用 WHERE 筛选器 .. 151	7.2.4 为交叉联接添加 WHERE 子句... 180
5.10.4 步骤 3: 数据分组 153	7.3 内部联接..... 181
5.10.5 步骤 4: 使用 HAVING 筛选器... 154	7.3.1 内部联接的语法格式..... 181
5.10.6 步骤 5: 处理 SELECT 列表 154	7.3.2 等值内部联接..... 182
5.10.7 步骤 6: 使用 ORDER BY 子句 155	7.3.3 不等值联接..... 184
5.10.8 步骤 7: 使用 TOP 或 OFFSET/ FETCH..... 156	7.4 外部联接..... 186
第 6 章 子查询 157	7.4.1 外部联接的语法格式..... 186
6.1 在选择列表中使用子查询 157	7.4.2 左外部联接..... 187
6.1.1 子查询示例 157	7.4.3 右外部联接..... 190
6.1.2 子查询与联接的关系 159	7.4.4 完全外部联接..... 191
6.2 含有 IN 和 EXISTS 的子查询..... 159	7.5 自联接..... 192
6.2.1 含有 IN 的子查询 159	7.5.1 使用不同列实现自联接..... 192
6.2.2 含有 EXISTS 的子查询 161	7.5.2 使用同一列实现自联接..... 193
	7.6 多表联接..... 194
	7.6.1 顺序联接..... 194
	7.6.2 嵌套联接..... 196

7.6.3 指定联接的物理顺序	198	8.7.2 使用 ROLLUP 汇总数据	238
7.6.4 多表联接示例	199	8.7.3 区分空值和汇总值	238
7.7 联接算法	202	8.7.4 返回指定维度的汇总	239
7.7.1 嵌套循环联接	202	第 9 章 窗口计算和表旋转	242
7.7.2 合并联接	203	9.1 窗口和开窗函数简介	242
7.7.3 哈希联接	204	9.2 基于窗口的聚合计算	245
7.7.4 使用联接提示强制联接策略	206	9.2.1 窗口的分区	245
7.8 使用 APPLY 运算符	208	9.2.2 窗口的排序与框架	248
第 8 章 操作结果集	211	9.2.3 开窗聚合函数的嵌套	253
8.1 合并结果集	211	9.2.4 分区聚合计算与联接的比较	255
8.1.1 UNION 与 UNION ALL	212	9.3 基于窗口的排名计算	256
8.1.2 使用 ORDER BY 子句	213	9.3.1 ROW_NUMBER 函数	256
8.1.3 结果集的合并顺序	214	9.3.2 RANK 和 DENSE_RANK 函数	258
8.2 查询结果集的差异行	214	9.3.3 NTILE 函数	259
8.2.1 使用 EXCEPT 运算符	214	9.4 基于窗口的分布计算	261
8.2.2 查询全部差异行	216	9.4.1 PERCENT_RANK 函数	261
8.3 查询结果集的相同行	218	9.4.2 CUME_DIST 函数	262
8.3.1 使用 INTERSECT 运算符	218	9.4.3 PERCENTILE_CONT 和 PERCENTILE_DISC 函数	263
8.3.2 查询全部相同行	218	9.5 基于窗口偏移计算	264
8.4 UNION、EXCEPT 和 INTERSECT 的执行顺序	219	9.5.1 LAG 和 LEAD 函数	264
8.5 在其他语句中使用 UNION、EXCEPT 和 INTERSECT	221	9.5.2 FIRST_VALUE 和 LAST_VALUE 函数	265
8.5.1 在结果集操作中使用 INTO 子句 ..	221	9.6 表旋转	267
8.5.2 突破结果集操作的限制	222	9.6.1 PIVOT 运算符	267
8.6 使用公用表表达式	223	9.6.2 UNPIVOT 运算符	271
8.6.1 CTE 的语法结构	223	第 10 章 数据修改	273
8.6.2 多个 CTE 定义和 CTE 的多次引用 ..	226	10.1 插入数据	273
8.6.3 CTE 的间接嵌套	228	10.1.1 使用 INSERT 和 VALUES 插入行	273
8.6.4 使用递归 CTE	228	10.1.2 使用 INSERT 和 SELECT 子 查询插入行	275
8.7 汇总数据	236		
8.7.1 使用 CUBE 汇总数据	236		

10.1.3 使用 INSERT 和 EXEC 插入行 ..	275	10.4 合并数据.....	283
10.1.4 使用 SELECT INTO 插入行	276	10.5 使用 TOP 和 OFFSET-FETCH 限制	
10.1.5 使用 BULK INSERT 从文件		修改	286
插入行	277	10.5.1 限制插入数据.....	286
10.2 更新数据.....	278	10.5.2 限制更新数据.....	287
10.2.1 使用 SET 和 WHERE 子句		10.5.3 限制删除数据.....	288
更新数据	278	10.6 使用 OUTPUT 输出受影响行的	
10.2.2 使用 FROM 子句更新数据	279	信息	288
10.2.3 使用 CTE 和视图更新数据	280	10.6.1 INSERT 与 OUTPUT	289
10.3 删除数据.....	281	10.6.2 DELETE 与 OUTPUT	290
10.3.1 使用 DELETE 删除行	281	10.6.3 UPDATE 与 OUTPUT	291
10.3.2 使用 TRUNCATE TABLE 删除		10.6.4 MERGE 与 OUTPUT	292
所有行	282		

开 发 篇

第 11 章 视图	295	13.1.2 扩展存储过程.....	312
11.1 创建视图	295	13.1.3 系统存储过程.....	312
11.1.1 创建简单视图	295	13.2 SQL 存储过程.....	312
11.1.2 创建索引视图	296	13.2.1 创建存储过程.....	312
11.1.3 创建分区视图	297	13.2.2 修改存储过程.....	317
11.2 修改视图	297	13.2.3 存储过程的重新编译.....	318
11.3 更新视图中的数据	299	13.2.4 存储过程的错误处理	319
11.4 删除和重命名视图	302	13.3 CLR 存储过程.....	327
第 12 章 游标	303	13.3.1 创建一个具有输出参数的 CLR	
12.1 创建游标的步骤	303	存储过程.....	327
12.2 快速只进游标和可滚动游标	306	13.3.2 创建返回行集和信息的 CLR	
12.3 静态游标、动态游标和由键集驱动的		存储过程.....	333
游标.....	308	13.3.3 删除 CLR 存储过程和程序集...	335
12.4 使用可更新游标进行数据更新	309	13.3.4 CLR 与 SQL 存储过程的择取建议...	336
第 13 章 存储过程.....	311	13.4 嵌套存储过程.....	337
13.1 存储过程的类型	311	第 14 章 触发器.....	339
13.1.1 用户定义的存储过程	311	14.1 DML 触发器.....	339

14.1.1	AFTER 触发器	339
14.1.2	进行事务提交和回滚操作	341
14.1.3	检测对指定列的 UPDATE 或 INSERT 操作	343
14.1.4	指定 First 和 Last 触发器	346
14.1.5	嵌套和递归触发器	347
14.1.6	INSTEAD OF 触发器	350
14.2	使用 DDL 触发器	354
14.2.1	激发 DDL 触发器的 DDL 事件 和事件组	354
14.2.2	创建 DDL 触发器	357
14.3	CLR 触发器	360
14.3.1	SqlTriggerContext 类	360
14.3.2	创建 CLR DML 触发器的步骤	363
14.3.3	创建 CLR DDL 触发器的步骤	369
14.4	修改、删除和禁用触发器	371

14.4.1	DML 触发器	371
14.4.2	DDL 触发器	372
14.4.3	CLR 触发器	372
第 15 章 用户自定义函数		374
15.1	标量 UDF	375
15.2	表值 UDF	376
15.2.1	使用内联式表值 UDF 实现 参数化视图功能	376
15.2.2	使用多语句式表值 UDF 进行 复杂计算	377
15.3	CLR UDF	379
15.3.1	标量 UDF	379
15.3.2	表值 UDF	382
15.3.3	聚合 UDF	384
15.4	修改和删除 UDF	388

性能调整篇

第 16 章 事务处理	391
16.1 自动事务处理	391
16.2 显式事务处理	392
16.3 隐式事务处理	393
16.4 使用嵌套事务	394
16.5 使用事务保存点	397
第 17 章 并发访问控制	399
17.1 并发影响和并发控制类型	399
17.1.1 并发影响	399
17.1.2 并发控制	400
17.2 锁管理器的数据锁定	400
17.2.1 锁的粒度和层次结构	401
17.2.2 锁的模式	402
17.2.3 锁的兼容性	404

17.2.4 锁升级	404
17.3 自定义锁定	406
17.3.1 自定义锁的超时时间	406
17.3.2 使用表级锁提示	407
17.4 使用事务隔离级别	408
17.5 使用行版本的事务隔离级别	410
17.5.1 快照隔离和行版本控制的工作 原理	410
17.5.2 使用基于行版本控制的隔离 级别	411
17.6 处理死锁	414
17.6.1 防止死锁的方法	415
17.6.2 使用 TRY...CATCH 处理死锁	415

第 18 章 查询的优化与执行	418
18.1 查询的优化	418
18.1.1 查询计划定义的内容	418
18.1.2 生成查询计划	420
18.2 执行计划的缓存与执行	422
18.2.1 执行计划的副本和执行上下文...	422
18.2.2 执行计划的开销管理	423

18.3 执行计划的重用	423
18.3.1 通过简单参数化提高计划重用率...	424
18.3.2 通过强制参数化提高计划重用率	424
18.3.3 使用显式参数化提高计划重用率	426
18.4 执行计划的重新编译	426

实 战 篇

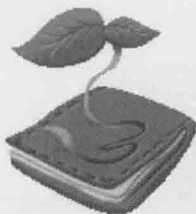
第 19 章 SQL 查询演练	430
19.1 同一时间范围内并发数统计	430
19.1.1 使用子查询	431
19.1.2 使用 CTE	432
19.1.3 使用窗口函数	433
19.2 时间段天数统计	435
19.3 数字范围统计	436
19.3.1 查找剩余空位区间和剩余空位编号	437
19.3.2 查找已用货位区间	439
19.4 地域范围内最大数统计	440
19.5 从分组中取前几行数据	441

19.5.1 使用联接获取前几行	442
19.5.2 使用窗口排名函数获取前几行...	443
19.6 取出多列中的非空值	444
19.6.1 姓名问题处理	445
19.6.2 工资问题处理	446
19.7 将数据由行转换为列	447
19.8 层次结构操作	448
19.8.1 检索祖先节点	450
19.8.2 检索子树节点	451
19.8.3 插入新节点	451
19.8.4 变更节点位置	453
19.8.5 hierarchyid 的索引策略	455

1

基础篇

- 第 1 章 SQL 简介
- 第 2 章 数据库与架构
- 第 3 章 表管理
- 第 4 章 索引管理
- 第 5 章 基本查询
- 第 6 章 子查询
- 第 7 章 联接和 APPLY 运算符
- 第 8 章 操作结果集
- 第 9 章 窗口计算和表旋转
- 第 10 章 数据修改



第1章 SQL 简介

SQL 的全称是结构化查询语言 (Structured Query Language)，是一种在关系数据库管理系统 (Relational Database Management Systems, RDBMS) 中查询和管理数据的标准语言。这是一种非常易读和易用的语言，只要稍微有一点英语基础，一些简单的数据查询、操作语句几乎都可以理解。但是，要想精通 SQL，并不是一件很容易的事情，一是因为在数据处理方面有许许多多的个案，要处理好这些个案，除了具有缜密的逻辑思维，还需要多练习和实践；二是与其他编程语言相比，SQL 是一种描述性语言，你只要说出想要什么结果，中间过程由数据库引擎去处理，其他编程语言则需要通过循环语句、判断语句等自己进行处理，如果出现什么问题，可以很快找到导致错误的语句，而对于 SQL 你就需要对寥寥几行的查询语句深入思考——数据库引擎为什么没有理解我的要求？所以说，数据库引擎的这种“封箱”操作给查找问题带来了麻烦，通常的解决方法就是将一个复杂语句拆解为几个简单语句，逐步验证中间结果的正确性，然后再重新组合为一条语句。从实际工作经验看，我不建议你写很复杂的语句来证明自己的高深，简单明了的语句对于后期的系统维护很有帮助，除非你的语句确实影响了处理效率。从本章开始，打好坚实的基础，并在此基础上不断提升自己的理论知识体系，当感受某个成功喜悦的时候，或许就会发现自己已经站在了 SQL 的巅峰。

本章我们将讲述一些最基本的 SQL 内容，如 SQL 的历史起源、ANSI 是什么和一些基本的数据库理论，以及 SQL 的语法元素和执行 SQL 的工具等。基本知识学习起来可能比较枯燥。但是，如果你是初学 SQL，这些基本知识对于学习好本书的后续内容却起着至关重要的作用。我也曾见过许多资深的数据库管理员会犯一些低级错误，一上午在不停地分析语句，最后却发现仅仅是“三值”逻辑问题。

1.1 SQL 的历史起源

在 20 世纪 60 年代，网状数据库系统 (如 CODASYL) 和分层数据库系统 (如 IMS TM) 是用于自动化银行业务、记账和订单处理系统的一流技术，这些系统是由于商业大型计算机的引入才启用的。而 SQL 是在 20 世纪 70 年代创建的一种基于关系数据库管理系统模型的数据查询、操作语言。

1.1.1 CODASYL

CODASYL 是美国数据系统语言协会（Conference on Data System Language）的英文缩写，该协会成立于 1957 年，主要目的是开发一种用于创建商业应用的通用语言。1959 年 5 月 28 日，该协会召开了首次会议，就语言开发进行讨论。这个语言实际上就是 Cobol 语言。

1963 年 6 月 10 日，美国加利福尼亚州的系统开发公司（System Development Corporation）举办了一个题为“基于计算机的数据库开发和管理”（Development and Management of a Computer-centered Data Base）的研讨会，首次提出并定义了数据库（Database）术语，即一组文件（表）的集合，其中文件是数据项（行）的有序集合，而每个数据项由数据以及一个或多个键组成。

1965 年，CODASYL 成立了“列表处理任务组”（List Processing Task Force），后更名为“数据库任务组”（Data Base Task Group）。1971 年 4 月，任务组发布了一份重要的报告，报告概述了网状数据模型，被称为 CODASYL 或 DBTG（即 Data Base Task Group 的缩写）数据模型。这个模型定义了数据库的几个关键概念，包括定义模式的语法、定义子模式的语法和数据操作语言。

1.1.2 IMS

IMS 是信息管理系统（Information Management System）的英文缩写，是 IBM 公司的产品，这是一款分层数据库管理和事务处理系统。

IMS 最初的开发目的是支持美国的阿波罗太空计划。1966 年，IBM 公司的 12 名成员、美国洛克维尔（American Rockwell）公司的 10 名成员和卡特彼勒公司（Caterpillar Tractor）的 3 名成员被集合起来，开始开发信息控制系统（Information Control System, ICS）和数据语言/接口（Data Language/Interface, DL/I），用于辅助跟踪建造太空船所需要的材料。其中，ICS 部分是用于存储和获取数据的数据库部分，而 DL/I 部分则是用来与之交互的查询语言。

在开发过程中，IBM 小组转移到加利福尼亚州南部的洛杉矶，并增加至 21 名开发人员。1967 年，IBM 团队完成了 ICS 的第一个版本。1968 年 4 月开始进行安装测试。1968 年 8 月 14 日，第一个安装成功，信息显示在美国国家航空航天局洛克维尔部门的 IBM 2740 打印机终端上。

1969 年，ICS 被更名为信息管理系统，即 IMS。自第一个版本问世以来，IBM 一直在不断完善和丰富 IMS 的功能。2007 年，IBM 推出了 IMS 10 版本。IMS 10 具备了增强的 XML 和网络服务功能，并且也是第一个将标准 XML 查询语言应用于层次结构化数据的数据库系统。

1.1.3 RDBMS 和 SQL

无论是 CODASYL，还是 IMS，虽然这些系统为早期系统提供了良好的基础，但它们的基本体