

<<SQL范例完全自学手册>>

图书基本信息

书名：<<SQL范例完全自学手册>>

13位ISBN编号：9787115212948

10位ISBN编号：7115212945

出版时间：2009-10

出版时间：明日科技、庞娅娟、房大伟、张跃廷 人民邮电出版社 (2009-10出版)

作者：庞娅娟，房大伟，张跃廷 著

页数：430

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>

<<SQL范例完全自学手册>>

前言

SQL (Structured Query Language, 结构化查询语言) 语言是一种组织、管理和检索计算机数据库存储数据的工具。

它是一种计算机语言, 可以用它与数据库交互。

事实上, SQL使用的是一种特殊类型的数据库, 即关系数据库。

SQL本身不是一个数据库管理系统, 也不是一个独立的产品, 而是数据库管理系统不可缺少的组成部分, 它是与DBMS通信的一种语言和工具。

由于其功能丰富, 语言简洁, 使用方法灵活, 因此备受用户和计算机业界的青睐, 被众多计算机公司和软件公司采用。

经过多年的发展, SQL语言已成为关系数据库的标准语言。

目前, 虽然介绍SQL的图书多不胜数, 但真正适合初学者自学的图书并不多, 本书就在这种情况下应运而生, 本书精选了大量SQL项目开发中常见的热点和关键问题, 并以实例的形式对这些问题进行了逐个击破, 使得读者能够在实践中学习SQL, 并迅速成为SQL项目开发高手。

为何选用本书对于初学者, 本书提供了四位一体的全新学习方式, 可以帮助读者在编程中不断体会编程乐趣, 快速提高编程能力。

下面是四位一体式学习方式图。

图书 (光盘源程序) + 视频录像讲解 (本书提供8小时实例录像) + 实践练习任务 (光盘提供实践练习源程序) + 网上支持 (实例视频讲解和论坛支持服务)。

<<SQL范例完全自学手册>>

内容概要

《SQL范例完全自学手册》是一本集查询、使用、学习和练习为一体的自学手册，书中介绍了应用SQL进行开发的各种技术和技巧。

全书分为17章，内容包括SQL基本查询，WHERE子句过滤，ORDERBY子句排序，数值查询，字符串查询，日期查询，聚合数据检索，分组统计数据，子查询，交叉表、递归查询与分布式查询，插入、更新和删除数据，XML数据检索，优化操作，管理和维护数据库、SQL编程与应用等。

《SQL范例完全自学手册》精选了154个典型实例，所选实例覆盖了SQL数据库开发中的热点问题和关键问题。

全书按实际应用进行分类，可以使读者在短时间内掌握更多实用技术，快速提高编程水平。

《SQL范例完全自学手册》附有配套DVD光盘，光盘中提供了实例的全程视频讲解及所有实例源代码，这些源代码都经过精心调试，在WindowsXP、Windows2000和Windows2003下测试通过。

《SQL范例完全自学手册》内容详尽，实例丰富，非常适合作为编程人员及项目开发人员的学习用书。

<<SQL范例完全自学手册>>

书籍目录

第1章 SQL基本查询 实例001 在SELECT语句中使用计算列及列别名 2实例002 定义批处理(BEGIN...END语句) 4实例003 查询指定行数据 7实例004 去除结果集中的重复行/列数据(DISTINCT关键字) 9实例005 计算数据在结果集中的行号(ROW_NUMBER()函数) 11实例006 随机查询n行数据(NEWID()函数) 13实例007 查询指定长度的文本类型数据(READTEXT命令) 15实例008 格式化结果集数据(CONVERT()函数、CAST()函数) 18实例009 使用iif()函数根据指定条件显示查询结果 21实例010 利用查询结果集生成表/临时表 22第2章 SELECT语句:WHERE子句过滤 实例011 运用AND与OR运算符有效的排列查询条件 28实例012 使用比较运算符查询指定范围内的数据 30实例013 通过IN运算符给出查询范围 33实例014 使用BETWEEN/NOT BETWEEN...AND查询范围数据 35实例015 利用LIKE并引用通配符、转义字符进行模糊查询 38实例016 查询为空/不为空的数据(IS NULL/IS NOT NULL) 41实例017 在查询中处理字段值为NULL的数据 44实例018 使用谓词查询数据(ALL/ANY) 46实例019 通过EXISTS运算符查询符合条件的数据 48实例020 使用NOT运算符查询否定条件的数据(与运算符等效) 51第3章 SELECT语句:ORDER BY子句排序 实例021 多字段的升序/降序排列查询结果集 56实例022 按照姓氏笔画/音序排序 57实例023 按升序排列“名列前三”的数据 60实例024 对重复值的排序 62实例025 在ORDER BY子句中使用表达式 63实例026 在ORDER BY子句中使用日期函数 66实例027 排序包含字母及数字的字段 69实例028 随机排序指定行数的查询结果集 71实例029 根据一定条件进行动态排序 72实例030 在视图中使用ORDER BY子句 75第4章 数值查询 实例031 在查询时对数值进行取整 78实例032 对查询到的数值四舍五入 80实例033 利用三角函数计算数值 83实例034 根据查询数值求弧度/角度(DEGREES()函数、RADIANS()函数) 91实例035 求数值的指数/对数 93实例036 实现数值的进制转换 98实例037 根据生成的随机数查询记录(RAND()函数) 100实例038 根据查询数值的符号显示具体文本(SIGN()函数) 101实例039 求数值的绝对值(ABS()函数) 103实例040 判断是否为数值(ISNUMERIC()函数) 105第5章 字符串查询 实例041 在字符串中去除空格 108实例042 在字符串中插入空格 110实例043 实现字符串的大小写转换 113实例044 转换为字符数据 115实例045 返回字符在字符串中出现的次数 119实例046 返回字符串中的子串 122实例047 删除字符串中的子串 123实例048 查找并替换字符串 125实例049 比较两个字符串 127第6章 日期查询 实例050 查询指定时间段的数据 130实例051 转换日期格式 132实例052 计算两个日期的距离(DATEDIFF()函数) 135实例053 按指定日期的天、月、年分别查询数据 137实例054 返回当前日期时间及对应星期数 140实例055 查询指定时间间隔的数据(DATEADD()函数) 142实例056 综合运用DATEDIFF()与DATEADD()函数 144实例057 判断是否为日期(ISDATE()函数) 147第7章 聚合数据检索 实例058 对多列的求和运算 150实例059 求大于平均值的数据 152实例060 求最大/最小值 155实例061 求无重复/不为空的记录数 158实例062 随机查询求和 161实例063 统计某个值出现的次数 163第8章 分组统计数据 实例064 分组汇总数据并排序 168实例065 计算各组数据的平均值(将NULL转换为0) 169实例066 使用ALL查询全部分组 172实例067 使用ROLLUP生成部分分组 174实例068 使用CUBE生成带有小计和总计的交叉表 176实例069 使用WHERE子句与HAVING子句指定查询及分组条件 178实例070 使用OVER子句和聚合函数实现分区聚合数据 181实例071 使用COMPUTE返回数据明细及汇总结果(多个结果集) 184实例072 使用COMPUTE BY返回分组信息及对应汇总数据(多个结果集) 186第9章 子查询 实例073 在SELECT子句中使用子查询 190实例074 在WHERE子句中通过比较运算符引入子查询 192实例075 在FROM子句中使用子查询 195实例076 使用IN运算符引入子查询 197实例077 使用NOT IN运算符引入子查询 200实例078 利用ALL谓词引入子查询 202实例079 利用ANY/SOME谓词引入子查询 205实例080 使用EXISTS运算符引入子查询 207实例081 使用NOT EXISTS运算符引入子查询 210实例082 在HAVING子句使用子查询过滤一组数据 211实例083 在UPDATE语句中使用子查询更新数据 213实例084 在INSERT INTO语句中使用子查询添加数据 215实例085 在DELETE语句中使用子查询删除数据 217第10章 连接查询 实例086 实现两表的笛卡尔乘积查询(CROSS JOIN) 220实例087 在WHERE子句中设置连接条件 221实例088 使用INNER JOIN实现自身连接 223实例089

<<SQL范例完全自学手册>>

使用INNER JOIN实现等值连接 224实例090 使用INNER JOIN实现不等连接 225实例091 多表内连接的查询 227实例092 使用LEFT OUTER JOIN以左表为依据查询匹配数据 228实例093 使用RIGHT OUTER JOIN以右表为依据查询匹配数据 230实例094 使用FULL JOIN查询两表匹配的所有数据 232实例095 多表外连接的查询 233第11章 组合查询 实例096 使用UNION组合多个结果集 238实例097 使用UNION ALL组合保留重复数据 239实例098 排序组合后的查询数据 241实例099 求两个结果集的交集(INTERSECT运算符) 243实例100 求两个结果集的差集(EXCEPT运算符) 245第12章 交叉表、递归与分布式查询 实例101 使用CASE语句指定显示的文本 248实例102 使用CASE语句更新数据 249实例103 通过CASE语句生成静态交叉表 251实例104 实现结果集中记录的行列转换 253实例105 使用PIVOT运算符实现交叉表 256实例106 使用UNPIVOT运算符反转交叉表 259实例107 CTE递归查询 262实例108 调用系统存储过程进行分布式查询 265实例109 使用函数实现分布式查询 267实例110 运用分布式事务 269第13章 插入、更新和删除数据 实例111 使用INSERT INTO语句插入数据 272实例112 使用UNION ALL运算符批量插入数据 275实例113 使用INSERT SELECT语句批量插入数据 277实例114 实现数据库与Excel数据的导入导出 279实例115 实现SQL Server与Access数据的导入导出 283实例116 使用UPDATE语句更新数据 286实例117 使用DELETE语句删除数据 289实例118 删除表中相同的行数据(游标) 292实例119 删除完全重复的行(临时表) 299第14章 XML数据检索 实例120 以XML形式显示检索数据(FOR XML RAW) 304实例121 检索数据以生成XML文件(FOR XML AUTO) 310实例122 将XML数据插入到数据表中 314实例123 利用XML索引检索数据 317实例124 查找XML节点(QUERY方法) 321实例125 提取XML节点元素值(VALUE方法) 324实例126 插入XML元素节点 327实例127 删除XML元素节点 331实例128 更新XML元素节点的值 334实例129 将XML节点转变为对应字段(NODES方法) 337第15章 优化操作 实例130 创建索引加快数据引用(非聚集、惟一、聚集索引) 342实例131 使用约束维护数据完整性(主键、惟一、检查约束) 349实例132 使用视图格式化检索数据 354实例133 执行带参数的存储过程以动态检索数据 359实例134 定义触发器以自动执行SQL语句 364实例135 异常错误处理 369实例136 应用事务维护数据的一致性 371实例137 在事务中的表级锁定 374实例138 实现事务并发控制 377第16章 管理和维护数据库 实例139 创建/修改数据库 384实例140 创建/修改数据表 388实例141 备份数据库 394实例142 恢复数据库 397实例143 附加/分离数据库 400实例144 查看数据库对象 402实例145 加密数据库中的数据 405实例146 防止SQL注入攻击 406第17章 SQL编程与应用 实例147 在C语言中使用SQL通信区 410实例148 嵌入式SQL中的主变量 412实例149 嵌入式SQL中的游标操作 415实例150 在高级语言中嵌入SQL语句 417实例151 在数据库管理系统中嵌入SQL语句 420实例152 预编译嵌入式SQL语句 422实例153 执行嵌入式SQL语句 424实例154 动态执行SQL语句 424附录

章节摘录

插图：技术要点通过创建非簇索引、惟一索引、多字段簇索引可以加快访问数据的速度。

1.索引的概念索引是一个单独的、物理的数据库结构，在SQL Server中，索引是为了加速对表中数据行的检索而创建的一种分散存储结构。

它是针对一个表而建立的，每个索引页面中的行都含有逻辑指针，指向数据表中的物理位置，以便加速检索物理数据。

因此，对表中的列是否创建索引，将对查询速度有很大的影响。

一个表的存储是由两部分组成的，一部分用来存放表的数据页，另一部分存放索引页。

通常索引页面对于数据页面来说小得多。

在进行数据检索时，系统首先搜索索引页面，从中找到所需数据的指针，然后直接通过该指针从数据页面中读取数据，从而提高查询速度。

数据库使用索引的方式与使用书的目录很相似，索引允许用户不必像翻阅完整本书那样就能迅速地找到所需要的信息。

在数据库中，使用索引可以迅速地找到表中的数据，而不必扫描整个表。

在了解了索引的基本概念的同时，讲解索引的优缺点及使用索引的条件：（1）使用索引的优点。

创建惟一性索引，保证数据库表中每一行数据的惟一性。

大大加快数据的检索速度，这也是创建索引的最主要原因。

加速表与表之间的连接，特别是在实现数据的参考完整性方面特别有意义。

在使用分组（GROUP BY）和排序（ORDER BY）子句进行数据检索时，同样可以减少查询中分组和排序的时间。

通过使用索引，可以在查询的过程中使用优化隐藏器，提高系统的性能。

（2）使用索引的缺点。

创建索引和维护索引要耗费时间，这种时间随着数据量的增加而增加。

索引需要占物理空间。

除了数据表占数据空间之外，每一个索引还要占一定的物理空间，如果要建立簇索引，那么需要的空间就会更大。

当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，降低了数据的维护速度。

（3）使用索引的条件。

考虑建立索引的列如下。

表的主键。

一般而言，存取表的最常用的方法是通过主键来进行的。

因此，应该在主键上建立索引。

连接中频繁使用的列（外键）。

这是因为用于连接的列若按顺序存放，系统可以很快地执行连接。

在某一范围内频繁搜索的列和按排序频繁检索的列。

建立索引是需要一定的开销的（包括时间和空间上）。

当进行INSERT和UPDATE操作时，维护索引也要花费时间和空间，因此，没有必要对表中的所有列都建立索引。

如何有效地使用索引要看建立索引和维护索引的代价与因建立索引所节省时间相比较而定。

一般来说，如下情况的列建议不考虑建立索引。

很少或从来不在查询中引用的列，因为系统很少或从来 not 根据这个列的值去查找行。

只有两个或若干个值的列（例如性别：男/女），显现不出使用索引的好处。

<<SQL范例完全自学手册>>

编辑推荐

《SQL范例完全自学手册》：软件工程师入门，双格式视频编程随身学。

<<SQL范例完全自学手册>>

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>