

数据库结构式查询语言 SQL

数据库结构式查询语言 S Q L

吕春莲 等编译
孔祥清
胡 杉

国家统计局计算中心

数据库结构式查询语言 SQL

清华大学出版社
北京

ISBN 7-302-04111-2

数据库结构式查询语言

S Q L

吕春莲
孔祥清 等编译
胡 杉

国家统计局计算中心

前 言

自关系型数据库技术诞生以来，它已得到广泛的成功的应用，由此也产生了对一种通用性关系型数据库操作语言的要求。人们希望这种语言对于不十分熟悉数据库技术的用户比较友好，对于程序工作人员与应用开发人员都能有足够的方便性。经过不长时间的酝酿与开发工作，满足这种需求的结构式查询语言SQL产生了。最终用户可以比较容易地学习、掌握这种语言，从而可以很快掌握数据库的操作。SQL也容易嵌入过程式高级语言如C、COBOL、FORTRAN和PL/1等等，利于使用高级语言进行应用开发工作。SQL语言也为最终用户和程序开发人员之间提供了一种讨论问题的通用途径，从而能够将纸质上的文件顺利地转换到计算机上的数据库系统中去。

本书介绍了SQL的全貌，并考虑了SQL语言的发展趋势，考虑了当前市场上多数关系型数据库系统的应用形势，没有完全局限在SQL的标准上。

本书的主要目的是帮助读者学习、掌握SQL的应用，所以，在对SQL的整个结构作了简介后，从数据库的查询、表的建立与操作、报表生成及数据库管理几个方面对SQL进行了详细介绍，并通过实例一步一步地展现了SQL的工作方式。

本书的编译编写主要参考了美国国家标准协会的SQL标准：American National Standard for information systems——database language——SQL, ANSI X3.135—1986；以及卡罗林（Carolyn J. Hursch）博士和杰克（Jack L. Hursch）博士于1988年出版的《SQL THE STRUCTURED QUERY LANGUAGE》一书。

参加本书编译工作的有吕春莲、孔祥清、胡杉等同志。

本书由江道琪同志审校。

内容简介

本书介绍了目前广泛使用的关系型数据库子语言——结构式查询语言 SQL。在对 SQL 的产生与发展作了简单介绍后，本书从数据库的查询、表的建立与操作、报表输出及数据库管理几个方面，通过一个人事数据库实例介绍了 SQL 语言各方面的内容与使用方法，其中包括一些目前应用较多的 SQL 扩充内容。最后，本书对目前市场上使用 SQL 的若干数据库系统作了简介。

本书适合于开始学习、掌握 SQL 语言以操作其关系型数据库的各行各业各类人员阅读，对于掌握一些 SQL 基本命令，而想进一步提高掌握水平的计算机人员也会大有帮助，而且对于用过其他数据库操作语言，而想改用 SQL 语言的数据库使用人员，同样也是一本合适的读物。

《数据库结构式查询语言SQL》

目 录

第一章 SQL的产生与构成概论

1. SQL的产生与发展	(1)
2. SQL构成概论	
2.1 概论	(3)
2.2 SQL的命令	(3)
2.3 保留字	(4)
2.4 数据类型	(5)
2.5 SQL的数字运算符、逻辑运算符与谓词	(7)
2.6 数据定义语言 (DDL)	(7)
2.6.1 生成表语句	(8)
2.6.2 生成索引语句	(8)
2.6.3 修改表语句	(8)
2.6.4 删除表语句	(8)
2.6.5 删除索引语句	(8)
2.7 数据管理语言 (DML)	(8)
2.7.1 插入 (INSERT) 语句	(9)
2.7.2 更改 (UPDATE) 语句	(9)
2.7.3 删除 (DELETE) 语句	(9)
2.7.4 查询 (SELECT) 语句	(9)
2.7.5 视图生成语句	(10)
2.8 数据控制语言 (DCL)	(10)
2.9 表操作表达式 (子句)	(10)
2.10 聚合 (组合) 功能 (Aggregate Functions)	(11)
2.11 子查询	(11)

第二章 数据库的查询

3. SQL的运行方式说明	
3.1 样本数据库的介绍及有关概念	(13)
3.2 表与表之间的关系	(16)

3.3	运行SQL的硬设备与软件系统环境	(16)
3.4	SQL命令的输入方式	(17)
3.5	SQL命令的修改与编辑	(17)
3.5.1	列出当前的SQL命令	(18)
3.5.2	编辑当前命令	(19)
3.5.3	重新运行当前命令	(19)
3.5.4	追加一个新行	(20)
3.5.5	向一行追加文本	(20)
3.5.6	删除一行	(21)
3.6	SQL命令的存贮和重新调用	(21)
3.6.1	存贮一个命令	(21)
3.6.2	命令的重新调用	(22)
3.6.3	START命令	(22)
3.7	关于LOGIN, SQL文件	(23)
3.8	退出SQL运行环境	(23)
4.	从表中选择列	
4.1	概述	(24)
4.2	选择命令	(25)
4.3	从表中选择指定的列	(25)
4.4	选择表中的全部列	(25)
4.5	控制列的排列	(27)
4.6	防止重复行的选择	(27)
4.7	建立和使用列别名	(28)
5.	从表中选择行	
5.1	概述	(29)
5.2	WHERE子句: 从表中选择指定行	(29)
5.3	WHERE子句的指定条件	(30)
5.3.1	指定条件	(31)
5.3.2	表达式、复合表达式	(32)
5.3.2.1	带AND的复合逻辑表达式	(32)
5.3.2.2	用OR连接的逻辑表达式	(33)
5.3.2.3	运算符的优先级	(34)
5.3.2.4	求反表达式	(35)
5.4	选择某一范围中的行	(36)
5.4.1	与列值中的任一值相匹配	(37)
5.4.2	匹配字型	(39)
5.5	控制显示行的顺序, ORDER BY子句	(40)
5.5.1	升序或降序	(41)

5.5.2	根据多个列的值排序	(42)
5.5.3	空值排序	(42)
5.6	SELECT命令中子句的顺序	(43)
6.	连接表	
6.1	概述	(43)
6.2	连接表	(43)
6.3	在WHERE子句中连接行的规定	(44)
6.4	如何引用表	(45)
6.5	相等连接和不等连接	(46)
6.5.1	外连接	(48)
6.5.2	表自身的连接	(50)
7.	数字型数据的显示与操作	
7.1	概述	(51)
7.2	数字数据的显示	(51)
7.3	算术表达式	(52)
7.3.1	SELECT命令中的表达式	(52)
7.3.2	WHERE子句中的表达式	(53)
7.3.3	ORDER BY子句中的表达式	(53)
7.3.4	具有几种运算符的表达式	(54)
7.4	算术函数	(55)
7.5	聚合(组合)功能	(57)
7.5.1	汇总(Summarizing)组合功能	(57)
7.5.2	计数(count)函数	(59)
7.5.3	组合功能与单个功能的不相容情况	(60)
7.5.4	子查询中的组合功能	(60)
7.5.5	多个行组分别汇总	(61)
7.5.6	怎样选择要进行汇总的成组记录	(62)
7.5.7	给计算的列作标记	(64)
8.	字符型数据值的显示与操作	
8.1	概述	(65)
8.2	显示字符型数据值	(65)
8.3	字符表达式	(66)
8.4	字符常数中的撇号	(67)
8.5	字符函数	(68)
8.5.1	字符函数的例子	(68)
8.5.2	能返回不同类型值的函数	(71)
8.6	字符常数	(71)
9.	日期型数据值的显示与操作	

9.1	概述	(72)
9.2	日期显示	(72)
9.2.1	标准的日期格式和替换格式	(72)
9.2.2	格式模型的例	(75)
9.2.3	一天的时间表示	(77)
9.3	日期运算	(78)
9.3.1	日期函数ADD_MONTHS	(78)
9.3.2	伪列SYSDATE	(80)
9.3.3	一天时间对日期运算的作用	(82)
9.4	在SQL命令中使用日期值	(83)
10.	空值的显示与处理	
10.1	概述	(84)
10.2	什么是空值	(84)
10.3	检索空值	(85)
10.4	按顺序排列有空值的行	(88)
10.5	在表达式和函数里的空值	(89)
10.6	空值函数: NVL	(91)
10.7	带字符值和日期值的NVL	(92)
11.	子查询语句的运用	
11.1	概述	(93)
11.2	怎样运用子查询语句	(93)
11.3	返回一组值的子查询语句: ANY和ALL谓词	(95)
11.4	又一种返回一组值的子查询语句: IN和NOT IN谓词	(96)
11.5	返回一个以上的列的子查询语句	(98)
11.6	多重子查询语句	(98)
11.7	引用多个表的子查询	(102)
11.8	与主查询相关的子查询语句	(103)
11.9	测试真假的子查询语句: EXISTS谓词	(105)
12.	表中的树形结构信息	
12.1	概述	(106)
12.2	树形结构信息的概念	(106)
12.2.1	在表中怎样表示树	(107)
12.2.2	层次码	(108)
12.3	确定树根	(111)
12.4	选择路径: 优先运算符	(113)
12.5	行的选择	(114)
12.6	树的SQLPLUS定义	(117)
12.7	其它用途	(117)

第三章 表的建立与操作

13. 建表	
13.1 概述	(118)
13.2 建表命令	(118)
13.3 表的命名	(119)
13.4 列的描述	(120)
13.4.1 列名	(120)
13.4.2 数据类型	(120)
13.4.3 禁止用空值	(122)
14. 在表中插入, 更新和删除行	
14.1 概述	(122)
14.2 向表中插入行	(123)
14.3 插入空值	(124)
14.4 插入日期值	(125)
14.4.1 日期格式	(125)
14.4.2 时刻	(126)
14.5 两表之间行的拷贝; 查询插入	(126)
14.6 控制何时修改有效	(128)
14.7 更新行中的字段	(129)
14.8 更新若干行	(130)
14.9 利用查询来选择要更新的行	(131)
14.10 利用查询提供更新的值	(131)
14.11 从表中删去一行	(132)
14.12 从表中删除若干行	(132)
14.13 自动提交任务; 设置自动提交	(133)
14.14 从事务处理失败自动恢复	(134)
15. 修改表	
15.1 概述	(134)
15.2 改变列的宽度	(135)
15.3 增加列	(135)
16. 建立和使用视图	
16.1 概述	(140)
16.2 视图是什么, 有怎样的用途	(140)
16.3 建立视图	(141)
16.4 查询视图	(141)
16.5 更新视图	(143)
16.6 多个表的视图	(143)

16.7 在视图中使用表达式和函数	(145)
16.7.1 视图中的组合功能	(146)
16.7.2 修改用于定义视图的表	(147)

第四章 报表的生成

17. 将查询结果格式化为报表	
17.1 概述	(148)
17.2 什么是报表	(148)
17.3 列定义命令	(151)
17.3.1 列标题	(151)
17.3.2 格式化列	(152)
17.4 页标题	(153)
17.4.1 TTITLE命令	(153)
17.4.2 确定页标题的位置	(155)
17.4.3 COL子句	(155)
17.4.4 BTITLE命令	(156)
17.4.5 显示、清除和压缩页标题	(157)
17.5 组织行组: BREAK命令	(158)
17.5.1 BREAK 命令的类型	(158)
17.5.2 中断时的工作	(159)
17.5.3 在断点进行汇总等计算	(160)
17.6 页面大小命令	(162)
17.7 存储和修改报表	(164)
18. 生成报表的高级技术	
18.1 概述	(166)
18.2 跨组汇总计算	(167)
18.3 把值写成总数的百分比	(175)

第五章 数据库的管

19. 数据的共享与安全性	
19.1 概述	(186)
19.2 访问其他用户的表	(186)
19.2.1 用户名作表名前缀	(186)
19.2.2 为表创建同义名	(186)
19.3 表的共享	(187)
19.3.1 封锁类型	(188)
19.3.2 封锁持续时间	(189)
19.3.3 死锁的解决方法	(189)

19.3.4	LOCK TABLE语句	(189)
19.3.5	SELECT...FOR UPDATE OF 语句	(190)
19.3.6	建立封锁的其它命令	(190)
19.4	授权	(191)
19.4.1	授权命令	(191)
19.4.2	授全部特权	(191)
19.4.3	为所有用户授权	(192)
19.4.4	权力传递	(192)
19.4.5	权力侵犯与保护	(192)
19.5	撤销特权	(192)
19.6	为视图授特权的作用	(193)
19.6.1	限制访问表中部分数据	(193)
19.6.2	其意义依赖于用户的视图	(194)
19.6.3	改变口令	(195)
19.7	更新用户名	(196)
20.	索引	
20.1	概述	(196)
20.2	什么叫索引	(197)
20.3	建立索引	(197)
20.4	索引的用途	(198)
20.4.1	提高查询速度	(198)
20.4.2	保证唯一性	(198)
20.4.3	加速连接查询	(199)
20.5	删除索引	(200)
20.6	建立索引的准则	(201)
21.	表和视图的拷贝与删除	
21.1	概述	(201)
21.2	拷贝表和视图	(201)
21.3	删除表和视图	(202)
21.3.1	删除表	(202)
21.3.2	删除视图	(202)
21.4	改变表名或视图名	(203)
22.	数据字典	
22.1	概述	(203)
22.2	描述数据字典的表的例	(204)
22.3	描述其他表的表	(205)
22.4	描述列定义的表	(206)
22.5	记录关于表和列的注释	(207)

22.6	一个表的空间需求.....	(209)
22.7	索引表.....	(210)
22.8	一个描述视图定义的表.....	(211)
22.9	显示你的数据库的特权.....	(211)

第六章 实用SQL简介

23.	若干实用的SQL	
23.1	概述.....	(213)
23.2	IBM的System R	(213)
23.3	IBM的IMS.....	(213)
23.4	SQL/Data System (SQL/DS) 数据库.....	(213)
23.5	Database 2 (DB 2)	(214)
23.6	ORACLE.....	(214)
23.7	Informix.....	(214)
23.8	SQLBASE.....	(215)
23.9	XDB.....	(215)
23.10	Ingress	(215)
23.11	SQL作为知识库的查询语言	(215)

第一章 SQL的产生与构成概论

1. SQL的产生与发展

早在1970年,当E. F. Codd在《大型共享数据库的关系型数据模型》一文中介绍关系型数据库时,他就提议到:采用关系型数据模型,使我们开发一种基于应用谓词运算的通用数据子语言成为可能。尽管他指出了对这样一种语言的需求及其优点,但在那时他还没有试图去设计之。第二年,他在《数据库子语言的关系完备性》一文中讨论了“关系完备性”概念,这是一个由Codd提出的术语,现已广泛使用。

和计算机领域中一般先进技术的发展速度相比,对关系型概念的接受比较慢。直到1974年,查伯林(Chamberlin)和波易斯(Boyce)在《SEQUEL:结构化英语查询语言》一文中提出了结构化查询语言的形式。后来,他们和另外二人发表了《作为关系表达式的指定查询:数据子语言SQUARE》,与SEQUEL很相似,可是用数学表达式代换了SEQUEL的英语术语表达。两者都满足Codd提出的关系完备性。这里关系完备性的含义是“至少具有元组关系运算的丰富功能”。1976年,查伯林等人将SQUARE改名为SEQUEL2,并用作IBM的研究数据库System R的查询语言。

到1980年,查伯林写了《使用SQL数据库子语言的用户经验总结》一文,已采用了现在人们使用的名称形式:SQL,解释为“结构式查询语言”。由各方面用户组成的一个小组对SQL进行了测试,由此对SQL进行了一些补充,增加了外联结功能,这一点Codd先生在1979年已提出。后来还有若干进一步的改进,直到形成今天的SQL。

近十几年来,已产生了很多关系型数据库,比起在它之前的层次型和网状型数据库已越来越普遍,且有加速的趋势。为使日益增加的关系型数据库教材有一定规范性,Codd于1985年写下了十二条原则,指出至少必须满足其中的六条原则,一个数据库才能被称为是充分关系型的。在这十二条之前还有一条最基本的原则,称为“零原则”如下:

关系型数据库的管理:任何被广而称之为是或自称为是关系型数据库的管理系统,必须能够通过它的关系型功能对数据库进行完全的管理。

十二条规范化原则如下:

一、信息的表达:关系型数据库中所有信息明确地表达在逻辑层,并且确切地以一种方式即表中的值来表达。

二、逻辑存取性保证:关系型数据库中保证所有每一个数据都可以借助于表名、主键值和列名的联结进行逻辑存取。

三、空信息的系统表示:完全的关系型数据库管理系统支持使用空值。空值既不同于空格字符串,也不同于零值或任何其他数字,它表示尚未得到的或还不能使用的信息。空值的表达采用了独立于数据类型的一种系统方式。

四、动态的联机登录方式：数据库的描述法采用了本地级上的与常规数据相同的方式，所以，经授权的多个用户可以使用相同的关系型语言象使用平常的数据那样进行其查询工作。

五、全面的数据子语言：一个关系式的系统可以支持几种子语言和若干终端使用模式（例如“填写空格”模式）。其中至少要有一种子语言，其语句可以按照定义明确的句法、以字符串的形式表达出来，并且全面支持以下各要求：

数据定义

视图定义

数据操纵（交互式操作和通过程序操作）

完整性约束

权限设置

事务界限确定（事务开始、事务认可和事务重新运行）

六、可更新的视图：所有理论上可以更新的视图都可以由系统进行更新。

七、高级插入、更改与删除能力：以单一运算域处理一个基本关系或一个导出关系的功能不仅要用到数据的检索，也要用到数据的插入、更改和删除。

八、物理数据的独立性：无论存贮的表达上或是存取的方式上有什么变化，应用程序和终端活动都应维持逻辑上的不变性。

九、逻辑数据的独立性：对于数据库中的基本表，当理论上允许的一类不受影响的信息保存方式有任何变动时，应用程序和终端活动都维持其逻辑上的不变性。

十、数据完整性约束的独立性：对于特定的数据库，其完整性约束必须可以在关系型数据子语言中加以定义，也必须在登录中可以存贮。但在应用程序中既不需要定义，也不需要存贮。

十一、独立的分布性：一个系统，不论它是否支持数据库的分布性，它都必须有一个能够支持分布式数据库的数据子语言，而不必去加强或补充应用程序和终端活动。

十二、不可推翻性原则：如果一个关系型系统具有一个低层次（每次一个记录）的子语言，则它不可以推翻或越过一个高层次（每次多个记录）关系型子语言中所表达的完整性原则和约束。

SQL语言的关系完备性已得到证明。

事实上，SQL语言的开发者事先就明白它应当是一种什么语言、应当要求它做些什么。因此，SQL语言有着坚实的理论基础。在计算机语言发展史上这大概还是第一次。今天我们所应用的绝大多数计算机语言都是在某种基本思想基础上产生，然后从满足许多问题的需要出发不断给出相应的补充而最后成型。由于在开发SQL语言之前已经有了对它的详细要求，所以SQL语言非常精致、简洁，构造的命令数量相当少，却能满足非常复杂的数据库的绝大多数要求。它的简易性令不论是偶尔需要使用SQL的用户还是复杂任务的开发人员都感到使用它非常方便。SQL可以用于各种特定的查询，也可以嵌入到一个高级语言的主语言程序中使用。

现在，在关系型数据库的查询中已经使用着若干种结构化语言，而SQL则是作为商品化产品而被最广泛地使用的一种。美国国家标准协会（ANSI）已公布了称为数据库

语言SQL的句法和语义标准。需要增补的标准也正在考虑之中，其中含有关于一致性的内容，增强事务管理、规定某些用户定义准则、增强字符处理功能的内容，还有支持民族字符集的内容。

近年来在SQL的试用中，由于用户的补充、开发工作，SQL已有了不少变化。本书中将包含一些虽未进入ANSI标准但已出现在一些教材上、IBM的SQL版本中以及ORACLE的SQL版本中的内容。

2. SQL构成概论

2.1 概述

由于SQL除了能够查询数据库之外，还有很多其他功能，所以，尽管SQL被称为结构查询语言，实际上它含有更多的内容。例如定义数据结构，修改数据库中的数据，以及规定安全性限制。相应地，以上每一种功能有其一套语言表达式，依次称为数据定义语言DDL，数据管理语言DML，和数据控制语言DCL。本章对以上三种功能将给以概述，详细说明在后面的有关章节里，

SQL还含有一套词汇与符号，用以表达各种命令的含义与结果。如：值表达式，逻辑运算符，各种谓词，表的表达式，聚合功能，以及子查询。它们的确切句法将在以后章节里展示，并给出如何应用的细节及应用实例。

使用SQL数据库语言的实用关系数据库里含有一个目录表。尽管这个目录表是为了有效地使用SQL的必须的、最基本的表，但它实际上是一个表示系统面貌的表，并不是SQL的一部分。目录表中存有数据库系统包含的基本表、视图、存取权限、用户标识等信息，可以用SQL的选择（SELECT）语句查看。它可以被看作是一个系统的数据库，含有各种信息表，表中列有用户建立的或可存取的数据库情况。这个目录表的生成依赖于运行过程，目的是向用户提供数据库的内容情况。用户不可以使用更改（UPDATE）语句、插入（INSERT INTO）语句或删除（DELETE）语句对目录表进行操作。实际上，对数据库的各种操作是以目录表为基础的，所以用户对它做的任何改动都会破坏数据库的完整性。

2.2 SQL的命令

SQL语言由一套命令和命令的使用规则构成。表2.1给出了基本的SQL命令及其简要说明。

表2.1 SQL命令表

命 令	说 明
ALTER TABLE	对一个已存在的表增加一列或修改一列 (ANSI标准SQL中不含有本命令)
COMMENT	在数据字典中对表或列加一个注释
COMMIT	提交 (使之成为永久的) 已完成的对数据库的修改
CREATE INDEX	为表建立索引
CREATE TABLE	建立表并定义它的列及其特性
CREATE VIEW	定义一个表或多个表和 (或) 其他视图之上的一个视图
DELETE	从一个表中删除一些行
DROP	从数据库中删除一个表或索引等等
GRANT	产生用户标识, 指定口令密码、权限资格, 数据库对多个用户的使用授权, 以及各用户对于表或视图的使用权限
INSERT	给表插入新记录行
REVOKE	对某些用户取消数据库使用权限或对表的存取权
ROLLBACK	在产生系统错误时, 取消所有未经提交的修改工作, 以维护数据库的完整性。也可以用于用户在提交之前取消未完成的不正确的工作。
SELECT	从一个或多个表中选出一些行和列
UPDATE	更改一个表中某些字段的值

2.3 保留字

构成SQL命令、以及使用这些命令时指定的英语单词称为保留字。这些单词不可以用作用户标识。另外, 在没有通过系统运行对这些词加上特殊符号标志 (有了特殊符号, 可以与保留字区别开来) 时, 也不可以用作表名、列名、视图名等。表2.2列出了这些保留字。