计算机科学 1994Vol. 21 No. 4

SQL的现状及其研究与发展

胡志平 麦中凡
（北京航空航天大学计算机系 北京 100083）

摘 要 The research and development of SQL technology is still a focus in the field of database. This paper summarizes the work on SQL standard and its present situation, and discusses the continuous functional modifications of database's improviment in the mean time. Based on most people's ideas, we emphisize the later research on SQL3 about procedure language, type system and query technology.

关键词 SQL, Database, Standard, Relation, Object, Type, Model

一、SQL简介

如果说七十年代关系数据库还处于研制实验阶段，那么到了八十年代大量关系数据库商用产品的面市几乎占据了整个数据库市场，其中的原因在于它们使用了统一的语言界面，那就是，SQL成为了关系数据库的通用语言。

1974年，IBM公司San Jose研究实验室的D. D. Chamberlin在开发SEQUEL-XRM原型时提出了SEQUEL(Structured English Query Language)，1976年又修改成为SEQUEL/2，也就是目前的SQL(Structured Query Language)。SEQUEL/2主要成形于IBM的System R原型中[2]，并在原有基础上大加扩展，其后逐渐完善并最终成为一种完备的结构化数据库查询语言。

到目前为止，SQL语言已广泛用于关系数据库中，起初的SQL也是围绕着关系模型而制定的，SQL语言在关系数据库中主要有四大功能：查询，操纵(Manipulation)，定义，控制[1]，这四大功能基本上构成了关系数据库的功能主体，从七十年代到九十年代，短短的二十年间，关系数据库能广泛地流行起来，与SQL语言的作用是分不开的。

随着现代计算技术的进一步发展，特别是近年来OO技术的兴起，关系模型有了诸多的限制，SQL语言将如何适应新的对象模型成为SQL发展的目标，这也是本文中将要讨论的SQL3所要完成的工作。

二、SQL标准的产生与发展

众所周知，美国国家标准委员会ANSI和国际标准化组织ISO是世界上最具有权威的两大标准化组织，1986年10月，ANSI正式把SQL作为关系数据库的标准语言，并颁布了标准的SQL文本，X3. 135-1986，此后不久，ISO也作出了同样的决定，公布了IS9075，1986标准。这就是我们通常所说的SQL-86。SQL-86为软件制造商提供了一种极大的可能性，那就是无论在那种机器平台上，还是何种数据库系统，都可以采用SQL作为共同的数据存取或标准接口，从而使未来的数据库世界有可能连接成为一个统一的整体。这个前景是十分诱人而意义重大的，因而有

al symposium on database in parallel and distributed systems, July, 1990.

[6] Copeland, G. et al., Data Placement in Bubba. In Proceedings of ACM-SIGMOD international conference on management of data, May, 1988.

[7] Donovan A. Schneidr and David J. DeWitt, A Performance evaluation of four parallel join algorithms in a shared-nothing multiprocessor environment, In Proceedings of ACM-SIGMOD international conference on management of data, 1989.

[8] M. Seetha Lakshmi and Philip S. Yu, Limiting factors of join performance on parallel processors, In Proceedings of the 5th international conference on data engineering, 1989.

[9] Christopher B. Walton et al., A taxonomy and performance model of data skew effects in parallel joins, In Proceedings of the 17th international conference on VLDB, Sept., 1991.

入把 SQL 被采纳为标准及其后的发展称为是“一场革命”[3]。

然而就 SQL-86 本身而言，它基本上只是对当时市场上几个产品的概括，数字设备公司的 Andrew Eisenberg 去年在第 19 届 VLDB 大会上形象地称之为“一块秃骨”[1]，而事实上的 SQL-86 是为了使实际产品不需要作重大修改就可以符合此标准而设计的小型语言。尽管如此，SQL-86 仍然从根本上反映了关系数据库的构造思想，并开创了关系数据库快速发展的前景。

与此同时，另一标准化组织，美国国家标准和技术署 NIST 公布了相应的标准处理信息：FIPS-127，FIPS 是联邦政府提供给购买商开发产品的指导准则。FIPSSQL 要求测试扩展标准并注上 FIPS 标志，这在一定程度上完善并保证了标准的推广。

甚至在 SQL-86 正式公布之前，修订工作就已经开始，最初对现有标准以附录形式作为补充，直到 1989 年，ISO 正式公布了名为 IS9075：1989 的“数据库语言 SQL”的修订标准，ANSI 也发表了同样的校订版标准，名为 X3. 135-1989。这个新标准也叫 SQL-89。但 SQL-89 并没有作根本上的修改，只是增加了引用完整性的语法和语义，如增加了缺省值，检查约束和简单引用约束等。其后，NIST 很快出版了 FIPS-127-1。FIPS-127-1 基本上与 SQL-89 是一致的，但流露出对嵌入式 SQL 语言的担心，由于 SQL 语言与宿主语言的差异，无疑破坏了程序的一致性，形成明显的“语义断层”。为了进一步解决这个问题，ANSI 发表了第二个标准 X3. 169-1989，“嵌入式 SQL 语言”，而 ISO 则没有发表相关的标准，只是增加了一个关于嵌入式 SQL 定义的附录。

SQL-89 在数据库发展史上起过相当的作用，许多数据库产品，如著名的 Orcale 等，都是因为采用了 SQL 标准而流行至今。尽管如此，SQL 标准的修订工作也始终未停止，1992 年底，ANSI 的 X3. 135-1992，ISO 的 IS9075：1992 标准相继问世，成为当前的最新标准，这也就是 SQL-92，SQL-92 是 SQL-89 的超集，主要是对语言表达式作了较大的扩充。

然而，即使到了离今天不久的 SQL-92，也未能把当今十分流行的 OO 技术包含进来，成为 SQL 标准的美中不足。但事实上，一个更全面的 SQL 标准正在研制之中，这就是 SQL3。SQL3 增加了面向对象的支持，这一标准可望在 1995 或 1996 年正式出版。

三、SQL-86、SQL-89 和 SQL-92

SQL-86、SQL-89 和 SQL-92 都是基于关系模型的数据库语言，因此可以通称为关系 SQL。所谓关系模型，是一个建立在关系概念上的模型；关系用非形式化语言可以写成笛卡尔乘积 $D1 \times D2 \times \cdots \times Dn$。[3]关系 SQL 正是基于这样一张表的结构，定义了以下四个功能：

1. SQL 数据查询，即检索操作。检索的基本概念是映象(Mapping)，所以 SQL 语言亦称为基于映象的语言，其基本结构是由 SELECT-FROM-WHERE 组成的查询块，代表着关系数据库中的投影、选取、更新等操作的组合。执行查询的结果仍然是一个表(关系)。

2. SQL 数据操作(DML)，也叫作数据存取操作，主要包括三种语句：插入、修改和删除，即 INSERT、UPDATE 和 DELETE 操作。

3. SQL 数据定义(DDL)，即 SQL DDL 语句，它包括对基本表、视图、索引、聚集的定义和撤消以及为基本表设置同义名等。

4. SQL 数据控制(DCL)，即 SQL DCL 语句，它包括控制、授予、或撤消存取数据库的某些权限，控制权限传递、读写锁的定义与取消等。最常见的用户注册、登录口令、授予用户权限等，通常这些都是由 DBA 处理的。

以上这四大功能在 SQL-86 中基本实现并在 SQL-89 中加以扩充，在 SQL-86 核心中还定义了三种数据库访问方式：模块式、嵌入式和交互式，这三种方式至今仍保留在数据库产品中。而后来的 SQL-92 除了和 SQL-89 基本兼容外，更增加了许多新特性，SQL-92 对语言作了较大的扩充，纳入了那些广为实现的而 SQL-86 和 SQL-89 又未纳入的特性，诸如数据库演进、动态 SQL 和附加的集合操作等，除此之外，SQL-92 还被用作定义增加完整引用性工具，有更好的数据库约束，并改善了正交性。这些特性的实现，无疑为 SQL 成为高级结构化语言提供前题，如果真是这样，SQL 语言会更具魅力，因为它实际上更类似于 4GL，即只管“干什么”，而不管“怎么干”。

更详细地说，SQL-92 在 SQL-86 和 SQL-89 基础上的扩展可以归纳为以下八个方面：

1. 增加了许多附加类型。类型定义是数据库语言功能完备的标志之一。SQL-92 在原有类型基础上增加了可变长度的字符串类型、位串、复合字符集、时间及内部值等。

2. 在表达式内部允许使用标志符，设置 CAST 和 CASE 表达式，并新增加了导出表和连接表的概念。尤其是标志符的创建扩大了数据库管理能力，建立了持久性基础。

3. 进一步完善了正交性。正交性意味着查询的无异义性。在 SQL-92 中，允许在行及表中设置构造子，行语句上的操作不受标量元名称的影响以及在表达式操作中允许进行子查询。

4. SQL-92 中的目录包含了模式，而且每一个模式都有单独的宿主(以正交性的标志符表示)，模式中的对象可以用明确的或不明确的目录名和模式来命名。

5. 设置域(Domains)的概念，可以在定义列时来代替数据类型。所谓域其实就是包含了数据类型并带有缺省值、一组约束、校验因子等任选项的持久定义。

6. SQL-92 中提出了暂时表的概念，主要有创建性全程暂时表，创建性局部暂时表和声明性局部暂时表三种类型，暂时表的设置增加了程序的并发性并减少了存贮开销。

7. 在模式操作语言方面，SQL-89 只定义了创建对象的语法，而 SQL-92 可以进一步在权限范围内进行 Drop 以及 Revoke 等操作，并可登记在信息模式表中。

8. 动态 SQL 是 SQL-92 新增加的一个显著的特性，它大大地提高了 SQL 语言在程序设计方面的能力，概括起来，可以有以下六点：

1)值在尚未确定时，进程可以将其作为参数来接收。

2)在最简单的情况下，仅仅执行一个完整的非 SELECT 语句。

3)如果一个语句需要执行不止一次，那么可以使用 PREPARED 语句并单独执行。

4)在一个 PREPARED 语句中可以包含动态参数。如果不知道动态参数的个数，则可以使用一个描述符域，所谓描述符域是指保存在 SQL 工具中的结构，该结构中保存了数据项。

5)一个 PREPARED 语句可以用来打开一个指针。

6)如果动态语句的数目不能确定，可以使用扩展的动态语句名，指针和描述符域名。其中，扩展名可以是全程的，也可以是局部的，这主要取决于模型或对话的规模。

SQL-92 在 SQL-86 和 SQL-89 基础上还作了许多有效的扩展。虽然 SQL-92 公布没多久，但使用它的产品已很快见实效了，像 Microsoft Access，它采纳符合 SQL-92 的思想，目前已广泛流行在各种产品开发中。

四、新一代的 SQL3

就在 SQL-92 还未正式公布之前，一个代号为 SQL3 的新规范已经开始研制。当前的 SQL 语言虽然已经成了数据库的通用语言，但是随着计算机应用领域的不断扩展，关系模型表现出诸多局限，SQL 语言也因其没有充分利用语言技术的命名(naming)、类型(type)以及联编(binding)等机制，从而没有达到充分的正交性、可扩充性和功能性[5]。特别是进入八十年代以来，数据库技术中对象模型已深入人心，人们发现，用对象模型代替关系模型，将关系"存贮"到对象的属性和行为特性中去，不但能够更好地表达客观事物，完成数据的查询和操纵功能，保证数据的正交性，而且还可以实现数据的持久性定义，实现封装、继承、命名存贮等机制。从这一角度来说，适应技术的要求，扩展 SQL 语言在这一方面的功能也成为时代的需要。

然而，在 SQL3 还未正式出版之前，许多 ODBMS 产品已经广泛流行在程序应用之中。因为没有共同的标准可循，这些产品使用的语言接口各有所异，为解决这个问题，有人提出了后关系数据库语言 DBPL，[5]DBPL 的一大特色就是强调正交性，提倡用单一的语言框架来统一不同的抽象层次，它有一致的命名，类型及联编机制。DBPL 是为了补充关系 SQL 的不足而作的修改，与现在所说的 SQL3 相比，它们都存在着一些本质的相同之处：

其一，在现有的 SQL 标准中，类型虽然在不断丰富，但用户不能定义新的类型，而 SQL3 和 DBPL 则允许用户定义新类型，并增加了构造函数、析构函数以及一些操作功能。

其二，SQL3 和 DBPL 提供了类型完备性，即所有的类型构造子享有同等的地位，这样，很自然地导致了支持复杂对象的数据模型的非第一范式关系，从而打破了关系模型只限于关系中属性值是基本类型的约束。

其三，当前 SQL 中所有数据组织成关系(即表格)的形式，简单对象与复杂对象并无区别，为实现共享，共享变量放在公共段中，给数据库实现和维护带来不便。但在 SQL3 和 DBPL 中，永久变量即共享变量，用户可以清楚地了解共享变量。

尽管 SQL3 与 DBPL 有许多相似，但研制中的 SQL3 仍然是 SQL-92 的一个超集，在访问方式、功能实现上与 DBPL 有着很大的区别。到目前为止，SQL3 主要对 SQL 语言作了三方面的扩充，为对象存贮提

供了可能。

4.1 SQL3 在程序设计方面的扩展

我们已经知道，许多高级语言都已经实现了条件、开关、递归等复合功能。到目前为止，SQL 语言不论在交互式下，抑或是嵌入到高级语言中，都是单条顺序执行的。这无疑带了程序设计的不灵活性，SQL3 在这一方面所作的扩展是增加了大量的复合语句，从这一点来说，将来的 SQL3 确切地说仍是面向结构的语言，而不是面向对象的。不过，从后面的抽象数据类型 ADT 可以知道，SQL3 仍然是支持对象管理的，这一点与 C++极为相似。

SQL3 复合语句的一大特色就是 COMPOUND 语句，我们用 BNF 范式说明如下：

```
〈compound statement〉::=
        [〈beginning lable〉〈colon〉]
        BEGIN [[NOT] ATOMIC]
        [〈local declaration list〉]
        [〈sql statement list〉]
        END[〈ending list〉]
```

COMPOUND 语句为 SQL3 提供了很强大的功能，它可以用于变量声明、异常声明、异常处理、事务处理并可以使用动态 COMPOUND 语句。由 COMPOUND 语句声明的对象名字必须是唯一的，所有的对象类型可以为全程的，也可以为局部的，并且有同样的生命周期；用于异常处理的 COMPOUND 语句实际上是将一组异常情况、SQLSATATE 值与相应的行为联系在一起，形成触发/点火规则，同时，SQL3 还提供了 COMMIT 和 ROLLBACK 与事务处理有关的 COMPOUND 语句，并且形成事务链；一旦 COMPOUND 语句中的 COMMIT 或 ROLLBACK 发生，就将开始一个新的事务；动态 COMPOUND 语句则指出，在 SQL3 中的控制语句，包括 COMPOUND 语句，都是可以预置和执行的。另外，动态参数可以是输入参数，也可以是输出参数。

除了 COMPOUND 语句以外，SQL3 同时也增加了许多业已在其它高级语言中实现了的复合语句，如 CASE 语句、IF 语句、LOOP 语句、FOR 语句以及 ASSIGNMENT、SIGNAL、RESIGNAL 等语句，这些语句的出现，极大地丰富了 SQL 的语言能力。此外，SQL3 在模型和例程中增加了 CREATE 和 DROP 命令，使用了带 RETURN 语句的 FUNCTION。SQL3 这种融进高级语言的特色，又作为数据库的标准接口，可以说是在语言结构方面发挥得淋漓尽致了。

4.2 SQL3 在类型系统方面的扩展

SQL3 定义了抽象数据类型 ADT(Abstract Data Type)，可以说是"SQL 本身的一场革命"，人们常提到的 MOOSE(Major Object-Oriented SQL Extension)[1]，就是指这一方面的扩展，SQL3 也因此将 SQL 语言从关系数据库推向了对象数据库，我们可以从以下五个方面来说明 SQL3 的类型系统：

1)SQL3 通过使用 CREATE TYPE 来声明一个抽象数据类型 ADT，一个 ADT 声明由一系列属性和过程/函数组成，其中 ADT 属性可以是存贮属性，也可以是虚拟属性，ADT 函数则通过 ACTOR、CONSTRUCTOR 或 DESTRUCTOR 来决定。构造函数(CONSTRUCTOR)创建并初始一个 ADT 实例，而析构函数(DESTRUCTOR)正好做了相反的工作；ACTOR 是行为函数，它代表该 ADT 的行为属性。ADT 可以包含不止一个构造函数或析构函数，所有的 ADT 的内置函数均可以重载到其他 ADT 声明中去。

SQL3 中 ADT 的提出无疑为 OO 技术提供了最大的支持。一个 ADT 实例就是该 ADT 的对象，ADT 提供了对象封装的可能性和对象的可重用性，无论属性还是函数均可以标志以 PUBLIC、PRIVATE 和 PROTECTED，每一个 ADT 实例都有一个唯一的对象标识符(OID)，OID 是 ADT 实例之外的值而不是实例本身值的一部分，在整个全局对象中，它是唯一的。在创建一个 ADT 实例时，OID 也就按相应的生成规则唯一创建了。

2)SQL3 引进了子类型和超类型的概念。所谓子类型就是通过继承其它类型的属性的函数的类型，被继承的类型就是超类型。子类型可以继承不止一个超类型的属性和函数，这就是多继承。所继承的属性可以是存贮属性，也可以是虚拟属性，并且子类型可以增加自己的属性和函数。

3)SQL3 中可以用一个已存在的 ADT 或生成的类型通过 CREATE DISTINCTTYPE 语句来定义新的类型，即异类型。如果 B 是 A 的异类型，那么除了 A 中所有带 A 名字的量都改为 B 以外，其余标志量均相同，而事实上，A 与 B 的实例又是不可比的。异类型的使用简化了许多重复声明，也能更好地维护程序的一致性。

4)SQL3 提供了"样板"功能，ADT 类型定义时允许带有参数，即参数化类型，其中参数可以是值参数，也可以是类型参数。参数化类型是通过 CREATE TYPETEMPLATE 语句来声明的，而且，和 ADT 一样，它也可以定义为其它参数化类型的子类，参数化类型为提供不同的实参而生成新的类型提供了便利。

5)作为内置的参数化类型,SQL3 提供了最常见的聚集类型 SET,MULTISET 和 LIST。用户可以通过预先定义好的类型 T 来创建新类型 SET(T),MULTISET(T)和 LIST(T)。聚集类型的元素可以是任何一种类型,其中包括聚集类型。值得一提的是,SQL3 允许将定义在类型 T 上的函数 F 使用到聚集类型上,如果 R 是 F 的结果类型,那么应用到类型 SET(T)是的结果类型就是 SET(R),对其他聚集类型来说也是如此。

从以上这些功能看来,这里所论述的 SQL3 似乎是在传统的关系 SQL 的基础上加入对象模式,然而这里也只是对尚未出台的 SQL3 作了些预测性的探讨,研制中的 SQL3 的类型系统功能的完善仍然是一个热点,将来的 SQL3 也可能在元组(记录)类型,类型基本表,查询表达式,类型检查规则,ADT 重载符的使用方式等方面作出更好的修改。

4.3 SQL3 在查询语言方面的扩展

查询是数据库的基本操作,SQL3 也将在这一方面作出相应的扩展。突出的一点就是触发器规则。触发器规则是指一个命名的对象在"触发事务点火",或者说,在满足某个任选的条件时,该事件即予发生。SQL3 是通过在基本表上指定 INSERT,UPDATE,DELETE 等行为构成触发事件的。一个触发器定义可以是参考表中的值或新值,也可以是指定的列。另外,SQL3 中有递归执行,该递归可以是线性的,也可以是非线性的,递归联合中的任选子句可以指定一些附加的要求。

SQL3 还在查询语言中增加了一个新的谓词,如 SIRLIAR TO,FOR ALL,FORSOME 等的一些附加的内置数据类型,如数字有序集,布尔值等。SQL3 所作的这些扩展都取自于其它查询语言的长处,所以是具有代表性的。

五、结束语

SQL 语言历经 SQL-86、SQL-89 到 SQL-92 直到今后的 SQL3,表征着当今计算机技术的发展状况,而且 SQL3 本身尚处于研制之中,还在不断地完善,同时,本文也只是抽取了计算机界关于 SQL3 的一些共识,并进行了一些探讨,将来的 SQL3 也许还会包含更新的技术。另外,OO 产品的规范化也会促进 SQL3 技术的成熟,为了统一市场,去年,美国的 ODMG(Object Database Management Group)召集五家主要的面向对象产品公司,即 Object Design,Versant 等,共同讨论面向对象产品的统一规范,最后,ODMG 推出了对象数据库管理语言 ODML,作为 ODBMS 的规范语言[6],ODML 从某种角度上讲是发挥了对象 SQL 的作用,ODMG 也指出,经过一段时间的实验和经验积累以后,即使是 ANSI 和 ISO 不主动采纳其规范,ODMG 也会将其提交给 ANSI 和 ISO,也许,SQL3 标准也将会进一步地吸收 ODML 的内容。

IBM 公司的 Almaden 研究中心的 Partriera Selinger 曾撰文指出[7],即使是到了二十一世纪,数据库系统的软件各层仍然需要其界面接口(即 SQL 语言)来完成用户复杂的需求。SQL 语言仍然是关系数据库的核心,SQL 本身将日益丰富,甚至,编绎型 SQL 也会出现。我们从 SQL 语言短短的二十年发展史已经明确地看到了这一点,SQL 的研究和发展仍然大有作为,有待更多的科技工作者去探索。

参考文献

[1] VLDB' 93 TUTORIALS,19th International Conference on VLDB,August 24-27,1993

[2] C. J. Date,A guide to the SQL Standard,Addison-Wesley Pub.Co. 1987

[3] 李昭原,刘又城,《数据库原理》,北航出版社

[4] Jim Melton,与 DBMS 有关的标准制定工作(中译)

[5] 丁宝康,董小洁,后关系数据库语言 DBPL,《计算机科学》,Vol. 20,№. 6,1993

[6] Hiroshi Ishikawa,Kazumi Kubota,An Active Object-Oriented Database: A Multi-Paradigm Approach to Constraint Management,VLDB' 93 Proceedings,1993

[7] Particia G. Selinger,Predictions and Challengers for Database Systems in the Year 2000,VLDB' 93,Proceedings 1993

[8] R. G. G. Gattell,The Object Database Standard: ODMG-93,MorganKaufmann Pub.Co. 1994