33—38 数据库 数据库技术

计算机科学Vol.20№. 1 1993

九十年代初的数据库技术

施伯乐 周傲英 （复旦大学计算机科学系 200437上海）

摘要

In this paper, a survey of the new database techniques in 1991 is given. It describes some developments on the following researchs, rule processing (optimization implemetation, function extension), object-oriented databases and their implementation techniques, active databases, spatial databases, and heterogeneous databases, etc.

进入九十年代以来，人们越来越意识到，站在下世纪初各种信息管理的应用的立场来看，我们对支持这些应用的数据库原理和技术的理解还很肤浅。制造科学、科学形象化、机器人学、光学存储、以及高速通讯等领域的迅速发展已经对现有的数据库理论和实践构成了威胁。可以说，我们现在正处于新旧技术交替的关键点。新技术的研究正在全面铺开。尽管人们对未来一些重要数据库问题的解决办法尚不明瞭，各种技术也很不成熟，但从九一年国际数据库界的研究动态可以看出，研究目标日趋明朗，研究内容也日趋集中。下面将根据九一年内召开的与数据库最为相关的三个国际会议的情况概述九一年数据库的发展。这三个会议是：国际超大型数据库会议（简称VLDB'91，1991年9月在西班牙布鲁塞尔召开），ACM数据管理国际会议（简称SIGMOD'91）和ACM数据库系统原理研讨会（简称PODS'91）。以上两个会议于1991年5月同时在美国卡罗那多州丹佛市召开。我们将从规则处理、OODB及实现技术、主动数据库、空间数据库等方面予以概述。

一、规则处理

在新一代数据管理技术的研究中，规则首先被引入传统数据库。从1978年H.Gallaire和J.Minker编辑出版了会议文集"逻辑和数据库"开始，扩充现有DBMS以处理规则这一研究受到了广泛的关注，取得了极大的进展。到90年代初期，许多大学和研究机构推出的所谓知识库系统和演绎数据库系统都是以说明性的逻辑数据语言（其语句是规则）作为核心语言的。OODB从另一角度探索研究下一代的数据管理技术。关于OODB的研究，美国高级DBMS功能委员会在1990年的"第三代数据库系统宣言"中指出："OODB研究人员一般都忽视了规则的重要性。"此外，在主动数据库（active DB）中，规则已被用于为DBMS提供主动行为（稍后将讨论）。由此可见，规则已成为未来数据库系统的一个主要特征。下面将从规则的优化实现和功能扩展两方面进行讨论。

1. 优化实现

在DBMS中扩充处理规则的能力，关键在于系统的效率，而效率又取决于递归查询的优化。递归查询优化是一经典问题，也一直是研究的热点。近年来，围绕无函数Horn子句程序（称为DATALOG）的查询优化，人们做了大量的工作。在数据密集型的应用中，自底向上计值方法更具吸引力。其优化

可分为三类：程序优化、规则改写、和运行时优化。程序优化包括递归有界性检测和规则线性化等。这类优化技术是基于逻辑程序等价性的。研究动因是试图用非递归规则取代递归规则，用线性递归取代非线性递归。人们对有界性有较深入的研究，但距实用还相差较远。一是因为对一般应用中有界递归存在的概率还没有明确的认识，一是因为现有的有界性判定算法还不完善，能处理的程序类还很有限。近来的工作集中在为更广泛的一类程序划出有界性的（多项式）可判定/不可判定的边界，并给出高效判定算法。PODS′91中的"DATALOG有界性工具"一文给出了用于证明递归有界性的（不）可判定性的一些工具，包括图灵机的死机问题、半线性集合理论等。文章利用这些工具推广了已有结论。

递归是语言（如DATALOG）表达能力增加的源泉，也是复杂度升高的原因。因此，必须有效地处理递归。有界性可用来消除递归，而有界性在一般情形下又是不可判定的。对于有界性不得而知的程序而言，另一颇具吸引力的方法就是对程序进行变换，使结果产生的等价程序的计值变得更低廉更高效。PODS′91中题为"结构查询优化——演绎数据库语义优化的统一框架"的论文提出了"因式分解(factoring)"技术以确定子目标的"证明树可消除性"。这一性质是"递归冗余性"的推广。满足证明树可消除性的子目标不一定是递归冗余的，但它在查询谓词的某一确定子树中的重复出现却是冗余的。利用这一性质可以降低递归的"强度"。

规则改写技术中最具代表性的是魔集法和Alexander法。魔集法是通用的递归查询优化技术，它根据查询对程序进行改写。结果产生的程序的自底向上计值对原始程序更有效。右线性变换和上下文右线性变换也属于此类。这些优化技术都是基于完全的边侧信息传递策略的。PODS′91中的"过约束和右线性查询"提出了关于一类线性查询的规则改写优化技术。这些技术是基于部分边侧信息传递策略的，即并非每个可用的约束信息都传递到下一个子目标。它对右线性变换进行了改进，给出了新的规则改写方法，克服了原始方法在约束参量过多的情形下的无效性。另一篇题为"关于递归DATALOG查询的期望大小"一文中，作者通过分析指出，改写技术中递归谓词元数(arity)的降低比单纯的相关事实的减少更有效。这应验和推广了以下事实：右/左线性变换技术，如果是可用的，总比魔集法来得有效。前一类方法改写规则时会降低递归谓词的元数，而魔集法则只单纯地削减相关事实。该文得出的结论对今后改写技术的研究有一定的指导意义。

以上两类优化技术都是在规则编译阶段完成的。就系统整体性能而言，运行时的优化也是必不可少的。PODS′91中的"查询计值期间冗余组的检测"一文说明了利用一简单的运行时测试可以检测程序自底向上计值期间的冗余元组，从而减少元组的重复产生。这个测试既要利用程序的特性，又要利用执行期间保存下来的信息。

递归查询计值的自底向上方法与自顶向下方法相比有不少优点。可是，自底向上方法把计值过程中产生的事实保存下来直到结束，计值期间产生的事实量可能非常大。因而，在计值期间，丢弃那些不再有用的事实以减少空间需求量就显得很有意义，丢弃事实还可降低I/O代价和索引维护代价。SIGMOD′91中"逻辑程序的自底向上计值的空间优化"就是讨论计值期间的空间优化问题的。

2.功能扩展

为了满足新一代DBMS应用（如CAD、CAM）的要求，人们已提出了一些新的数据模型和查询语言来存储和管理复杂对象。其中，查询语言可粗略分为以下三类：基于演算的、基于代数的，和基于规则的。COL和HiLog语言都是基于规则的复杂对象语

言。COL语言是DATALOG语言的扩展，它允许操纵利用元组和集合构造子生成的结构化值，即规则中谓词的参量可以是元组、集合、函数等。对分层的COL程序，其语义可由一最小的合理模型给出。这一模型可用不动点的一个有限序列计算出。ACM TODS九一年第一期上"带函数和集合的基于规则的语言"详细地介绍了COL语言，并讨论了其实现问题。HiLog是一个二阶逻辑语言，其显著特征是具有一阶语义。HiLog程序允许谓词名作为其它谓词的参量出现，事实上，任意的项都可以作为谓词名出现。HiLog已被斯坦福大学NAIL！研究组选为DATALOG的换代语言，以此作为新一代演绎数据库系统的核心语言。在PODS'91的"关于HiLog的否定"一文中，作者T.Ross对规则体内含有否定子目标的HiLog程序进行了研究，把模块分层（modular stratification）概念、稳定模型(stable mode)和良构（well-founded）语义推广到了HiLog程序，并在模块分层的HiLog程序中推广应用魔集法。

近来，研究用面向对象的数据模型支持说明性语言或设计说明性的面向对象数据库语言是一很活跃的方向。当前的Horn子句语言（如DATALOG，LDL）中没有方法（method）的概念，也不具备多形性（polymorphism）。SIGMOD'91中的"LLO：一个带方法和方法继承的面向对象演绎语言"一文从面向对象的角度考察了Horn子句程序，认为可以对Horn子句语言进行扩展，通过支持合适的数据模型可以把面向对象的特征集成进去。在该文中，作者建立了可为演绎的OO语言提供合适支持的数据模型，描述了集成OO特征（如对象标识、方法和方法继承等）的类型化语言。在LLO中，方法是通过规则来定义的，方法继承是通过类型化和合一机制来完成的。

无论是从理论还是从实用的角度来看，对非确定（nondeterministic）数据库语言的研究都是必要的。这是因为，有一些非确定的查询，如果用确定的语言来实现则是不直观和低效的。典型的例子就是如"从每个系任意挑选出N个学生"这样的抽样查询。DATALOG没有提供定义抽样查询的工具；LDL中引入了选择（choice）构造以处理非确定查询。但choice处理多样本情形（如上例，$N>1$）比较困难且不直观。此外，对非确定的查询，其计值有一定的自由度。这一特性可用来进行查询优化。SIGMOD'91中的"一个非确定演绎数据库语言"描述了IDLOG语言，说明了IDLOG如何定义抽样查询以及如何用它来优化DATALOG程序。文章还讨论了IDLOG的表达能力。

正如在1990年美国国家科学基金会召开的"DBMS未来研究方向工作会议报告"中指出的那样，未来的数据库应用中，如卫星图片识别、医疗诊断等，需要在不精确的情况下进行推理。人工智能界多年来一直在致力于解决这一问题。对之进行进一步的研究是必要的，因为我们不仅要求能够处理不精确数据，而且还要求是在数据量较大的情形下有效地处理。SIGMOD'91的"演绎数据库不精确推理的新方向"一文就是研究这一问题的。作者提出了利用带有条件概率的不精确规则的解决办法。给出了可直接在具有DATALOG接口的演绎DB上实现的非单调不精确推理演算。

二、OODB及其实现技术

OODB提供了丰富的数据建模和操纵能力，被认为是满足现今及未来各类数据库应用要求的理想选择。OODB是当今数据库研究和开发的热点之一。几年来，已有各种原型系统推出。91年的PODS和SIGMOD会议也较深入地讨论了OODB的一些理论问题和实现技术。

OODB的数据模型主要可分为两大类：复杂对象模型和纯粹面向对象模型。复杂对象模型则起源于关系数据模型的扩展；纯粹面向对象模型则起源于函数数据模型。PODS'91中"纯粹OODB中结构化值的表达能

力"提供了一个通用框架以研究含有基于对象创建的查询语言的纯粹OODB模型中抽象(abstraction)的思想。抽象是表示这类语言中诸如聚集(aggregation)和集合等结构化值的关键操作。利用这一框架，可以研究抽象的表达能力，可以更好地理解该领域近期工作的具体特征。几年来，人们已经基于OODB的复杂对象模型提出了一些复杂对象查询语言。可是，对这些查询语言的复杂性和表达力并未深刻理解。PODS'91中的"复杂对象的易处理的查询语言"研究了复杂对象数据库的几种查询语言的表达力和复杂度，提供了这些数据库上查询复杂度的评价工具，给出了关于易处理查询语言的构造的建议。

在具体实现技术方面，SIGMOD'91中有两篇论文着重讨论了OODB中对象聚类(clustering)问题。对象聚类是影响OODB性能的重要因素。可是，在当前关于对象聚类的大多数工作中，优化什么以及聚类方法的优化程度都不清楚。"对象库聚类的随机方法"一文精确地处理了聚类中的根本问题，即：给定OODB以及它上面的存取模式的期望概率统计描述，什么将构成一个最优对象聚类？如何找出或近似这一最优聚类？文章给出了聚类问题的一个系统模型，讨论了系统中存取模式的两个精确模型。对于"独立恒等分布"模型，可以找出严格的最优聚类策略；对更强的"简单马尔可夫链"模型，聚类问题是NP-完全的。文章为此给出了启发式的聚类算法。

各种静态聚类策略可以改善OODB的性能，但对象更新和多联系的存在会损坏聚类产生的结构的有效性。更新会破坏初始聚类产生的结构；在多联系存在的环境下，基于某一联系对对象进行聚类会牲牺另外的联系。"OODB中复杂对象的有效聚类"一文研究了更新的影响，提出了一种对磁盘上相关对象进行重组织的重聚类模式，给出了评估重聚类的利益和开销的代价模型。对于对象间存在多联系的情形，文章给出了一种分层次的聚类模式把相关的对象组织到一个聚类序列中去。

PODS'91中另一篇文章"以多级存储方式管理持久对象"描述了一个包含多级存储器的存储持久对象的系统结构。长期以来，DBMS总是假设所有可存取数据都放在磁盘上，而未来的对象管理器将要求分主存（包括cache）、磁盘和第三级存储器甚至更高级存储器来管理超大型对象库。文章提出了满足这些要求的系统结构，给出了所需的查询优化器的概貌。

关于OODB和主动数据库的结合，VLDB'91中有一个专题，我们将在下节讨论。

三、主动数据库

主动数据库(active databases)最近已成为一个重要的、活跃的研究领域。传统数据库只执行那些在用户请求中应用代码中说明了的操作。而在主动数据库中，操作可以象一触发子(trigger)一样去激活其它操作的执行，这有很多用途，比如说，自动进行权限审核、存取登录、完整性约束维护和报警等。VLDB'91中有两个专题是关于主动数据库的。本节中引用的论文，如果不加说明则都是VLDB'91中的。

"主动OODB模型"给出了主动OODB的一个逻辑模型。其主要思想是利用OODB中诸如封装、继承等一类标准概念。在这一模型中，触发子也被当作一类特殊的方法被封装到适当的对象中。执行模型则使用了嵌套事务。

对主动数据库的研究似乎大多是从头开始研制新的语言构造（例如，"主动数据库程序设计的语言构造"一文提出的Heraclitn〔Rel〕），定义新的执行范式，和设计新的实现技术。"Alert：一个把被动DBMS转换成主动DBMS的系统结构"一文描述了Alert的设计及在Starburst系统上的实现。Alert是设计用来把被动SQL DBMS转换成主动DBMS的扩展系统。其显著特点是尽可能重用

被动DBMS技术，对被动DBMS的语言和实现作最小改动。Alert提供了一个层次结构，它允许在顶层支持各种产生式规则语言的语义。规则可以定义在用户定义的或内部的数据库操作之上。

规则已被广泛用于为DBMS提供主动行为。把产生式规则(即情形-动作规则)集成进数据库系统的动因是：产生式规则为表达触发子、报警、断言、完整性约束、存取约束、导出数据、快照等提供了统一的机制。规则还可支持数据库上的推理。以往的把规则加入OODB中去的尝试都导致规则与其它对象间的分歧。"OODB中的规则管理：一个统一的方法"一文提出了一个管理规则和对象的统一的方法，规则被看作"一等"对象，也用属性和方法来描述。按照这种方法，规则的管理操作可视为方法并用方法实现。按统一的方法，系统对规则及其它对象不加区分。这样，规则可以和另外的对象发关生系，可以按层次来安排。把规则当作对象的另外的优点是：为对象引入的任何设施（例如，索引机制、事务机制、锁机制、显示设施）都自动适用于规则。

四、空间数据库

在许多科学数据库中，数据元素都是二维或三维的点、线、多边形等。这些领域的数据库应用极大地依赖于空间数据，空间数据库 (Spatial databases) 也就应运而生。要在数据库环境中支持空间对象，就必须对DBMS的各个层次进行扩展。在用户接口层次，要扩展查询语言，使之可以定义和操作空间对象，同时又保持语言的说明性特点。在物理和存取方法层次，必须在DBMS中加入存贮、组织和存取空间对象的新方法以保证对空间对象的有效处理。在中间层次，必须有新的方法来把扩展的用户查询映射到扩展的物理和存取方法层次上去。

近年来研制的许多可扩充的或面向对象的数据库系统，如Starburst, EXODUS、POSTGRES、GENSIS、DASDBS等在用户接口层次都提供了定义和操作复杂对象（包括空间对象）的说明性语言，而中间层次的扩展主要涉及查询处理器和查询优化器的设计和实现。过去，很少有人把注意力放在空间查询的处理和优化上。VLDB'91中的题为"空间查询处理的优化策略"的论文讨论了标准的查询处理和优化技术在空间数据库环境中的应用，给出了针对空间数据的新的处理和优化策略。这些策略不仅限于空间数据，也可以推广以处理多介质数据库。关于物理和存取方法层的扩展，SIGMOD'91中有一篇文章"段索引：多维区间数据的动态索引技术"引入段索引的概念，给出了相应的索引技术。段索引是数据库索引结构的扩展，它是针对多维区间数据的索引技术，这类技术对于改善由多维的长度分布不均匀的区间组成的空间数据库的性能是很有用的。

测量数据是从自然界连续过程中采样得到的数据组成的。例如室外温度、地下水位、植物生长等数据，通常，数据量很大。为了便于管理，可以把数据存放在关系DBMS中。但大多数情况下，关系操作不直接适用于这些数据。由于可扩充的DBMS的研制尚处在原型阶段，还不能投入实用。因而，SIGMOD'91中"包含嵌入式内插过程的数据库查询的优化和计值"一文在商品化的关系DBMS (INGRES) 上进行扩展，使得从事测量的专业人员可以在任意点上查询一个测量过程得到的值。这是通过把用高级语言实现的内插过程集成进DBMS的查询部分来实现的。对关系DBMS查询语言 (SQL) 的扩展满足了用户查询原始测量的遗漏值、计算测量结果的等距序列、以及用内插方法比较不同序列的要求。文章描述了这些扩展引起的查询计值和优化问题。

五、其它

在VLDB'91的"九十年代及以后的数据库技术"的专题讨论会上，与会者认为，下一代数据库应用有以下几方面的要求：特大

量（兆兆字节）数据的管理、复杂对象的管理、多介质的支持、规则处理、异构分布式数据库的集成以及对时间和不精确性的支持。

随着通讯和网络技术的飞速发展，随着社会信息化程度的提高，越来越多的机构要求把一些异构的、分布式的、自治的现有数据库集成起来。于是就要求有一级数据库管理软件来从分布于计算机网络各位置上的数据库系统中存取数据。这样的软件系统就称为异构分布式数据库管理系统(HDDBMS)。SIGMOD'91唯一的一个专题讨论会就是关于HDDBMS的。与会者认为，尽管HDDBMS已从同构的分布式DBMS研究中获益匪浅，但HDDBMS成为现实可用的系统之前，还有许多问题有待于解决。例如，如何满意地解决性能、可扩充性、安全性等问题？哪一类代价模型对HDDBMS查询优化较为合适？哪一种DBMS标准（如锁机制、恢复机制）对未来的HDDBMS更为关键。VLDB'91中有一辅导报告题为"管理分布式、异构和自治数据库的联邦数据库系统"。HYDRO是正在研制的在包含有IBM AS/400 DBMS和其它系统的网络环境下的HDDBMS。SIGMOD'91中题为"HYDRO：一个HDDBMS"介绍了系统及其实现技术。

如何使异构、分布式的数据库表现出来就好象它们构成了单一数据库的一部分，这称为互操作性(Interoperability)。SIGMOD'91中的"实现有模式差异的数据库互操作性的语言特征"一文介绍了HP实验室在实现不同的关系数据库的互操作性方面的工作。现有的关系语言能力不足以提供数据库的互操作性。即使数据库都是关系型的，统一的多数据库视图也不能调解其模式差异。模式差异指的是现实世界同样类型的信息在不同的数据库中可用不同的关系模式表示。为实现这类互操作性，高阶表达式、高阶视图定义及视图可更新性都是必不可少的。这篇文章给出了一种称为IDL（互操作数据库语言）的高阶语言。IDL扩展了Horn子句语句，它在表达力上包含了LDL、DATALOG、SQL等。文章定义了IDL的语法和语义，并用一系列例子说明了它能实现互操作性的要求。

为了研制开发满足新的应用要求的DBMS，对新情况下数据结构的研究是必不可少的。SIGMOD'91中有一专题是关于B树的。其中一篇讨论了B树并发控制算法的性能；一篇研究了B^+树的多版本维护问题；另一篇讨论了如何利用多磁盘来改善B树结构文件的性能。对存储和管理复杂对象的数据结构文件的研究将随着新一代数据库系统的研究的深入而呈现出广阔的前景。此外，从九一年的这三个会议来看，事务处理和并发控制等传统内容的研究仍占有一定的比例。

参考文献

[1] Proc.of the 1991 ACM SIGMOD International Conference on Management of Data.

[2] Proc.of the 10th ACM Symposium on Principle of Database Systems.

[3] Proc. of the 17th International Conference on Very Large DataBases.