计算机科学1993 Vol.20№.3

数据库的互操作性

唐雪飞 （成都电子科技大学微机所 成都610054）

摘 要

当前，用户们正面临着一个多厂商异种数据库环境，对数据库间的互操作造成了极大障碍。笔者认为，就关系型数据库而言，信关乃不失为实现互操作性的可取方法；与IBM的DRDA相比，SAG规范更适合于实现基于多厂商异种平台之间透明的数据查询。要彻底解决异种数据库互操作性问题，还有赖于一种新型的数据库技术的出现——很可能是面向对象的数据库技术。

一、问 题

随着社会的发展，人们对数据信息的需求越来越广泛，不仅局限于一个部门内各数据库间的相互访问，还涉及各部门间乃至各单位间的数据共享。当今用户们所面临着的是：一个多厂商异种数据库、异种操作系统和异种网络协议的环境，后者对数据库间的互操作造成极大的障碍。造成这种局面的原因，主要是以下两个方面：

1． 标准问题：由于关系数据库国际标准SQL存在一些未定义之处，导致了各数据库厂商的SQL不兼容，用户从各个厂商获得的所谓符合ANSI/ISO标准的数据库产品都各操自己的SQL方言，彼此无法沟通。

2． 采购问题：由于计算机及其软件的价格下降，各部门所购买的数据库产品可能来自不同的厂商，从而使企业内部的数据共享面临着多厂商异种数据库的互操作问题。

二、数据库信关

多厂商异种关系数据库和系统的存在，为用户提供了可选机会，但异质引起的互操作问题却又影响着用户的自由选择。数据库信关可以帮助解决后一个困难。

“信关”（Gateway）这个术语来源于通信行业，它指的是连接两个或更多网络的黑盒。由于通过信关连接起来的网络可以使用不同的协议，信关的作用实际上是进行适当的转换。类似地，一个数据库信关允许一个“本地”DBMS用户访问另一个（可以为“远程”的）同种或不同种平台上DBMS，用户不必知道远程数据库所使用的存取机制。所以一个数据库信关的作用实际上相当于一个界面转换器。

客户、信关和服务器可以驻存在不同的平台上，而信关的本身实现又很适宜于借用桥接网络协议，这就能解决实现异种平台（包括操作系统和网络）上异种数据库的互操作问题，其示意图如图1所示。从图中可看出，一个数据库信关的逻辑成份包含以下两个内容：

1． 客户API（应用编程界面）库。利用它客户向服务器提交远程数据请求，并处理服务器的响应。

2． 服务器API库。是客户API库的镜象。客户API库的子例程发出请求，而服务器API库的子例程则产生对应于这些请求的事件。同时还利用它返回结果。

除了处理远程数据请求，数据库信关还应映射数据字典结构的差别：一种方法是在服务器的数据字典之上加一层兼容的数据字

唐雪飞 博士生

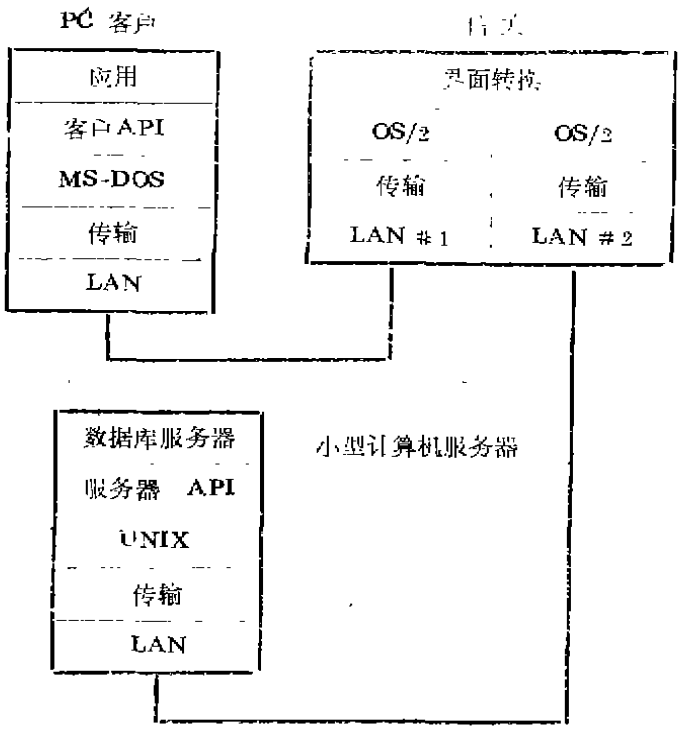

图 1

典集，它为客户应用程序提供一个公共的系统目录视图；另一种方法则是扫描远程DBMS的数据字典，存贮这些信息，以便利用它们构成远程DBMS的SQL。另外，由于驻存在不同平台上的数据库的数据类型和格式有所不同，数据库信关也要提供必要的数据类型及格式的转换。

数据库信关的发展大致经历了三个阶段：早期的数据库信关一般是用来与一个或多个异种数据库进行通信的一种定制的工具或实用程序。这种方式首先是由第三方工具商采用的，允许用户的应用在选定的一些数据库和客户平台上运行。第二阶段的数据库信关是当主要的RDBMS厂商为它们的工具提供到其它数据库（主要为IBM的DB2和DEC的Rdb）的界面时出现的，目的是扩展市场，满足用户的互操作性要求。但厂商们极少提供到一竞争对手的信关。例如不存在一个Oracle的应用程序访问一个Ingres数据库的信关，以上两个阶段的两种方式对用户选择合适的客户/服务器技术帮助不大。第三阶段是以Sybase的 Open Client 和 Open Server 等一类产品的问世为标志。它们是第一个用客户/服务器体系结构提供更大开放度的互操作技术。通过发放独立于它们的数据库或工具集的Client API和Server API许可证，Sybase为用户提供了连接到任何数据资源或数据服务的媒介。

数据库信关除了解决互操作问题外，还有许多好处。例如，数据库信关并不要求用户废弃现有的应用程序，而可把它们与新的数据库技术（如OODBMS等）集成起来，从而保护了用户过去的投资；另外，通过信关，也可以把第三方为别的厂商（如Oracle，Ingres或DB2等）开发的工具连接到自己的数据库产品上，关于数据库信关的优点还可列出许多，但它有一个最大的弱点，即n个异种数据库中要实现任意两个数据库间的互操作，就必须提供n(n−1)/2个信关，令人难以接受。不仅如此，甚至有些异种数据库间的数据格式、句法或语意的转换是行不通的，而且利用数据库信关动用远地异种数据库不易达到完整透明。所以有人说数据库信关只是连接异种数据库和实现互操作的权宜之计。因为，一旦一个统一的应用界面标准被完善并被广泛采纳，则信关的作用就将告退。关于这个问题在下面标准的讨论中会涉及到，但我们的看法是，尽管新的"完善"的界面标准肯定会被采纳，但各厂商为了商业上的竞争，必须保持各自的独特之处，而对标准进行适当的扩充，从而必定产生新的互操作性问题。此外，技术的继承性（新、旧技术的集成）也需要数据库信关，所以，作为解决数据库互操作问题的一种实际的方

法，数据库信关还将继续存在和发展下去。尤其在客户/服务器体系结构的环境里，有着极具吸引力的发展前景。

三、发展标准

由于缺乏一个严密而完善的SQL标准，使各厂商的RDBMS产品虽然名义上都符合ANSI/ISO的规范，却又各操自己的SQL方言，甚至连IBM的四个基于SQL的RDBMS—DB2、SQL/DS、SQL/400和OS/2 Database Manager，都彼此互不兼容，这为数据库的互操作造成了很大的障碍。为了使多厂商的RDBMS能够彼此对话，用户必须安装错综复杂的信关和协议转换器，并且还要祈求他们的应用不要太依赖于一个SQL方言，这些都给用户带来了苛刻的要求。

SQL Access Group (SAG)和IBM试图从标准一致化的角度来解决这个问题，即建立能够实现基于SQL的异种关系数据库互操作的规范，但它们采用的途径却大不相同。

SAG成立于1989年。它的目标是建立基于SQL的异种关系型数据库互操作的技术规范。SAG由40多家公司组成，其中包括Digital Equipment, Hewlett-Packard,Sun, Sybase, Informix Software,Borland和Microsoft。值得一提的是所列的公司中没有IBM (IBM拥有自己的一套解决办法)。该组织的工作已被X/Open采纳作为XPG(X/Open Portability Guide) 的一部分，可以从X/Open公司获得SAG的文本。

到目前为止，SAG已经产生了两个规范：

1. 应用编程界面 (API) 规范：定义一个嵌埋式数据库语言规范，该规范基于ANSI和OSI的SQL定义 (ANSI X3.135—1989和X3.168—1989)。

2. 格式与协议 (FAP) 规范：用于数据库应用与一RDBMS间的通信，该规范利用了ISO远程数据库访问 (RDA) 委员会的工作。

SAG的规范对其作为基础的标准进行了丰富和完美，对原标准中认为应由实现者定义的部分进行了明确的定义。例如，它们比较严格地限制了实现者的多种选择，使得按此写出的可移植的应用程序，在上述限制内工作的系统上可以相互操作。

SAG工作的第一次亮相是在1991年7月，该组织演示了它的新的规范，当时10个不同的数据库应用程序（包括HP和NewWave Access和Microsoft的Excel等）与9个不同的RDBMS(包括Oracle和DEC的Rdb/VMS等)一起工作，由一个应用产生的数据可被别的应用查询和索取。

虽然SAG并不是一个标准化组织，但它的规范已被X/Open采纳。SAG和它的许多成员也正在与ANSI和ISO合作，对正式的SQL标准进行改进。SAG的工作还在继续中，除了测试和验证SAG规范的实现外，还在进一步扩展它的规范。1991年12月，TCP/IP被加入其规范中，在此之前只包含了符合ISO/OSI模型的网络。

SAG的工作大体分为三个阶段：第一阶段（到91年7月为止）业已完成。该阶段的成果是规范，后者可以从X/Open获得；公开演示了包括19个客户和服务器原型的多厂商互操作性。第二阶段正在进行中，包括性能测试，TCP/IP的采纳，一个调用级API以及存贮于服务器中的已预编译的SQL语言。第三阶段则将着重于多服务器事务、存储式过程 (stored procedure)、大型对象和增强的安全性。

尽管在91年7月就进行了多厂商关系数据库互操作的演示，并且有40多家厂商声称要在未来的版本中支持这个规范，但最早也还要到1993年初才可望获得符合SAG规范的产品。Microsoft公司计划在明年初在其新产品中首次亮相的开放数据库连接 (ODBC) 技术是基于SAG调用级界面 (CLI)的。DEC于3月份宣布的 Accessworks是在该公司的MicroVAX,VAX4000和VAX6000小型机上运行的数据库访问软件包，Accessworks将

最终包含SAG的FAP和API，使DOS、Macintosh、UNIX、OS/2和VMS的客户能访问IBM的DB2、Oracle公司的Oracle以及DEC自己的Rdb和RMS数据库。另外网络产品制造商Retix也将在明年早期把SAG的规范嵌入其网络路由器和开关中。

然而，众所周知的数据库公司，象Oracle、*Informix*、*Sybase*、*Ingres*和*Borland*等，却还没有宣布任何基于这个规范的产品。

针对同样的数据库互操作问题，IBM的策略是开发一个分布式关系型数据库体系结构DRDA。DRDA最初目标是为在IBM的四个RDBMS之间进行互操作提供一个媒介。后来，IBM也向别的公司公开了DRDA规范并组织了研讨会，故DRDA也可用于多厂商的互操作性。从这个意义上讲DRDA是与SAG的规范对等的；后者是一个用于多厂商互操作性的规范，而前者则为IBM的那些RDBMS定义了应用编程界面、数据格式与网络协议。

在DRDA环境中，IBM的四个数据库各操自己的SQL方言；外加另三个IBM的SQL方言用于解释收到的询问或回复所使用的方言。当被询问时，每个RDBMS对发出询问的数据库的回复用的都是自己版本的SQL，而由发出询问的RDBMS来解释该方言。

IBM的DB2 V2.3支持DRDA，亦即用户们已经可以购到支持DRDA的产品，并且已有20多家厂商声称它们的产品对DRDA的支持，包括HP，Oracle等，采用的方式是作为DRDA的客户。

对于刚刚开始实现分布和链接多个DB2的大型数据库来说，DRDA是很可取的，但它还存在一些缺陷，其中最明显的是只对IBM的透明性。而且DRDA是单向的，人们还不能借助它来访问非IBM的数据。

SAG规范和IBM的DRDA都是为了解决基于SQL的异种RDBMS互操作问题所进行的努力，从以上的讨论中可以看出它们各自走的是两条不同的道路。下节将对SAG规范和DRDA这两者的差别作更深入详细的比较，且分为非技术性和技术性两个方面来考察。

四、两种标准的比较

4.1 非技术性差别

SAG规范与DRDA之间的非技术性差别，可以从以下四个方面来进行比较。

■最终的标准类型。存在两种类型的标准：实事上的和正式的。SAG的目标是促进正式标准的形成。各公司在SAG技术委员会中的代表往往也是ANSI相应的委员会中的代表。SAG对标准的改进建议，这是由这些代表提交给ANSI $X3H2$ (SQL) 和$X3H2.1$ (RDA)委员会进行表决的，随着这些建议被加入ANSI标准中，它们便成为了正式标准。DRDA是IBM的一个体系结构，如果有足够多的公司决定支持它的DRDA，则一定时间之后它就可能成为事实上的标准。

■规范的所有权。所有权决定了谁来控制一个规范的内容。SAG规范被国家或国际标准化组织和厂商们组成的联盟来控制。任何希望影响其方向的公司，都可以加入SAG及标准化组织，进行工作以影响标准的制订。DRDA则归IBM拥有，它的规范无疑地是受IBM所控制。

■技术的开放性。SAG规范和DRDA都在发展着。SAG规范的发展方向是通过一个委员会的工作程序决定的。在这个委员会中所有成员公司都绝对平等，各成员公司在它参加的每个委员会中都有一次表决权。DRDA的改进程序则由IBM的一个内部体系结构委员会管理，后者由来自IBM的四个RDBMS的代表组成。虽然IBM也为有兴趣的公司提供了发表建议的机会，但必定不会象SAG那样平等。

■实现的难易。把一个体系结构或者一个标准从书面阶段变成一个可操作的实现是一个长期的艰苦过程——尤其是多厂商互操作性方面。软件工具可以大大地加快一个实现工作，SAG的消息格式是由ASN.1模型定

义的，许多消息格式编译器可以自动地生成完整的消息编码和解码子例程集合。对于DRDA的实现，这是IBM公司内部的事，易于得到统一，所以实现会顺利得多。但由于没有相应的工具可用，这将增加开发成本并产生编程可移植性及互操作性问题。此外，对含意的正确解释对实现者来说是至关重要的。由于每个成员公司在SAG技术委员会中都有代表，各公司在实现时对含意的理解都有自己的专家可供咨询；而DRDA手册中充满了来自各个IBM环境的专用词汇，由于所有的 DRDA 的设计者都来自IBM，极少有IBM之外的专家能够深刻地领会独特而精心设计的 IBM 的体系结构，而这正是实现DRDA所必需的。

从以上的比较中，可以看出，由于参与者的广泛性，必将使得SAG的规范会沿着更为开放的道路发展，而DRDA则要封闭得多。但兰色巨人是世界上最大的计算机公司，其DRDA一旦成为事实上的标准，则各厂商们将被迫接受事实而去实现 DRDA。实际上Oracle公司已经声称在未来的产品中将同时支持SAG规范和IBM的DRDA。

4.2 技术性差别

SAG规范与DRDA在技术上也有很大的差别。在此主要从语言、目录表、消息/数据值的编码及网络需求等诸方面来进行比较。

■语言。SAG规范和DRDA都使用SQL作为查询语言，但使用的方法不同；SAG采取了一个公共集的途径，被使用的是一个单一的语言定义。这种一致的SQL语法和语义，使希望得到可移植应用的开发者们不必在各种方言中寻找一个公共子集，同时使目标服务器的选择可延缓到运行时才进行。在符合SAG规范的环境中，访问多个不同服务器的客户应用可以始终使用一个单一的 SQL。DRDA 采取一种 "哪一种都行 (anything goes)"的模式，即任何SQL变种都可以从客户流到服务器，而客户应用程序所面向的都是各服务器专用的SQL语法、语义和数据类型。可移植应用的开发者必须事先找出为要访问的服务器所支持的一个公共子集。且对目标服务器的选择必须在开发初期就先确定下来。在DRDA环境中，访问三个不同的服务器的同一个客户应用，可能要在同一程序中包含三种不同的SQL变种。

■目录表。又称模式信息表，提供元数据。它描述由一特别的服务器所管理的数据。SAG规范定义了一个具有标准属性和值的目录表集合，这些表基于相应的SQL2定义。DRDA对目录管理表采取的是"哪一个都行"的策略。它不定义一个标准目录表集合，应用程序是通过向由服务器的数据管理系统提供的目录表发出查询请求，来获得元数据信息的。这要求客户应用必须熟悉每种可能要访问的数据系统所提供的目录信息的结构。

■消息/数据值的编码。SAG 规范与DRDA 对协议信息和数据值进行编码的方法，完全不同。基本差别为：SAG规范使用OSI标准，而DRDA使用IBM 体系结构，如下表所示：

	SAG规范	DRDA
编码策略	正规格式	由接受者确定
抽象语法	ISO ASN.1	IBM DDM的FD:OCA
传递语法	ISO RER for ASN.1	IBM DDM和FD:OCA
磋商吗?	Yes	No

■ 网络需求。SAG 规范与DRDA 都有各自的通信网络环境，如下表所示：

	SAG规范	DRDA
网络环境	ISO OSI	IBM SNA
高层协议	ISO ACSE和表示层	IBM LU6.2
安全性	SQL Access和ACSE	IBM LU6.2
双工	全	半

SAG规范基于OSI参考模型并采用OSI的寻址和命名结构。只要不太大地破坏其格式和协议，SAG规范可以适合于任何提供端到端、全双工及虚拟线路类型连接的通信网络，TCP/IP和DECnet的peer-to-peer通信就可以提供SAG规范所需要的服务。DRDA把SNA LU6.2用于客户/服务器通信，DRDA与LU6.2关系非常密切，还不能把LU6.2的命名、动词及协议明显地映射成另一网络环境的类似机构。

■数据转换。在DRDA中，由消息的接收者来执行发送者的数据格式与其自身平台上的格式之间的转换。这种方式使数据转换和信息的最终损耗最少。但不幸的是，这一机制要求双方对于n种编码格式，至少要实现n-1种数据转换子例程集合。在SAG规范中，被传送的值是以一种定义良好的、独立于平台的标准的形式来表示的，发送方把自己的数据转换成标准格式，而接收方把标准格式转换成自己平台上使用的格式。虽然要经过两次转换，但每个平台上只需一个数据转换子例程集合。所以DRDA的机制在相似平台的环境中很有效，而在异种平台的网络环境中，则SAG的规范更现实些。

SAG规范与DRDA在技术上的差别还不止这些，以上列出的是几个主要的环节。但这已足以看出SAG规范更适合于实现基于多厂商异种平台之间透明的数据查询。

五、未来

由于标准的不完善，造成各厂商按标准实现的各种版本互不兼容，彼此无法沟通而产生了数据库互操作的问题。作为一种“权宜”之计，人们开发了用于界面转换的数据库信关。而为了从根本上解决问题，SAG和IBM又都在完善标准。在发展数据库互操作性规范方面进行了道路不同的努力。随着正式的或事实上的“完善”标准的出现，符合这些新标准的产品是否真的可以实现互操作了呢？回答是“不完全肯定”，随着标准的“完善”，只能说实现互操作的基础更好了，起点更高了。由于商业竞争的需要，各个厂商为保持其地位，必将在自己的产品中加入独有的特征，从而产生新的不兼容性，也为互操作造成新的障碍，于是又必须建立新的数据库信关来满足对异种数据库的数据共享的要求。这可能是一种永不完结的循环。那么是否数据库的互操作性就无法真正实现呢？人们已发现，关系型数据库系统由于其内在的不可克服的缺陷，而导致了互操作的不可实现，出路很可能是新一代的数据库技术——面向对象的数据库技术。对此研究者们正开始这方面新的探索。

感谢 完成本研究的过程中，特别得到了导师刘锦德教授的悉心指教，特致以衷心的感谢。（参考文献共34篇略）

（接第34页）

参考文献

[1] Panayiotis K. G,et al.,ACTA: A Framework for Specifying and Reasoning about Structure and Behavior, SIGMOD'90

[2] Alt R. K. et al.,Transaction Management in Engineering Database, IEEE Data Base Week 1983

[3] Calton Pu. et al.,Split-Transactions for Open-Ended Activities, Proceedings of 14th VLDB Conference 1989

[4] Hector Garcia, et al., SAGAS, ACM SIGMOD'87

[5] Jorge F. G, et al., Transaction Management in an Object-Oriented Database System, SIGMOD'88

[6] Pu. C., Replication and Nested Transactions in the Eden Distributed System, VLDB'86

[7] 施伯乐等，《数据库理论新领域》