A SUPPLEMENT TO

NETWORK WORLD

The Newsweekly of Enterprise Network Strategies

Preparing for Distributed Databases and Applications

DB/EXPO 93

Includes DB/EXPO '93 Planning Guide starting on page 22.

It All Happens Right At Our Readers' Fingertips

Network World readers have complete control over what goes on, and off, their company's expanding multisite networks. Including the applications that make them run.

They set the standards. They choose the vendors. They make the purchase recommendations. They even sign the checks.

And if a product or service doesn't meet with their exacting criteria for function and performance, it doesn't get bought. Period.

That's what we call having a powerful hand.

Each day Network IS professionals grapple with issues of interoperability, client/server computing and LAN/WAN integration. Each day, they must get a handle on the latest details involving information security and strategies to ensure network success.

And each week, they turn to *Network World.*

How can we be so sure? Take a look at the recent results of leading industry studies. Simmons. Adams. ComNet. Just to name a few.

Each provides independent confirmation of our enviable position as the leading networking newsweekly. Each provides independent confirmation that week after week, Network IS professionals turn to *Network World* for the news and information they need to know. They have to. Their jobs depend on it.

Network World readers hold the future of their company's networks, and your products and services, in the palm of their hands. Can you afford to be left out of the picture?

To place *Network World* on your media schedule, call Thomas J. Wilson at (800) 622-1108 today.

For an executive summary of "The Roles and Editorial Needs of Network IS Professionals," call Julie Haskell at (800) 622-1108.

Network World, Inc., 161 Worcester Road, Framingham, MA 01701-9172
All brand or product names are trademarks or registered trademarks of their respective holders.

IDG

NETWORK WORLD

SUPPLEMENT STAFF

Editor
Jim Brown
Managing Editor
Karen Moltenbrey
Art Directors
Susan Champeny
Susan Slater
Production Director
Ann Finn
Imaging Manager
Anne Nickinello

NETWORK WORLD STAFF

Editor
John Gallant
Executive Editor
John Dix
Assistant Managing Editor — News
Paul Desmond
Senior Editors
Bob Brown
Michael Cooney
Jim Duffy
Wayne Eckerson
Caryn Gillooly
Maureen Molloy
Bob Wallace
Senior Writers
Joanne Cummings
Skip MacAskill
West Coast Senior Editor
Fredric Paul
Washington, D.C. Senior Correspondent
Ellen Messmer

Features Editor
Charles Bruno
Associate Features Editors
Susan Collins
Alison Conliffe
Kyle Nitzsche

Assistant Managing Editor — Production
Michelle Beaulieu
Copy Editors
Amy Koulouris
Laura Mascharka
Monica Sambataro

Assistant to the Editor
Cheryl Crivello
Editorial Assistant
Glenna Fasold

President/Publisher
Colin B. Ungaro
V.P. Finance and Operations
Mary Fanning
Vice President, Advertising Director
Thomas J. Wilson
Manager of Advertising Operations
Karen Wallace
Senior Account Coordinator
Linda Miller
Advertising Coordinator
Cathy McPherson

Network World, Inc.
An IDG Company
Network World Headquarters
161 Worcester Road
Framingham, Mass. 01701
(508) 875-6400
Fax: (508) 820-3467

Cover and Inside Illustrations
Michael McLaughlin

CONTENTS

COMMENT

Data. It's the lifeblood of any organization.

Try conducting business without the proper data close at hand. It's next to impossible.

What's even more of a problem is managing this most valuable corporate asset, and that's the primary reason why computers and networks exist: so users can store, access and process the data needed to conduct business.

But data management is becoming increasingly difficult, especially in enterprise networks that spread data stored in many different types of computers across departments and geographical boundaries.

Data management responsibility, like data, is now being spread across organizations. Network managers and database administrators are finding it necessary to learn more about what each other does in order to tackle the tough job of putting a distributed database management system and networked applications in place.

Because the world of distributed DBMSs and networked applications is so confusing and complex, you need a place where you can learn about the various technical alternatives and get a grip on the type of issues you'll face when implementing one.

And that's just what we've done – given you a place where you can get a comprehensive look at what it takes to prepare for distributed databases and applications.

In this special *Network World* supplement, we've set out to give you a firm understanding of what it takes to build a true distributed DBMS. We also spell out some alternatives to a true distributed DBMS and help you identify the type of tough decisions you'll face when putting together strategic distributed DBMS implementation and application development plans.

Because seeing is believing, we'll also tell you how Burlington Coat Factory has plowed ahead into this largely uncharted field and is already reaping the benefits.

If you're on your way to DB/Expo 93 or at the show now, this supplement will help you get some of the core distributed DBMS concepts down pat so you'll be able to attend conference sessions or visit vendor booths for more information that is specific to your needs. Of course, you can use the handy DB/Expo planning guide to map out which conference sessions to attend or vendor booths to visit.

And as you'll see, four of our five authors are speaking at the show, which gives you a chance to chat directly with them if you have any questions about their articles.

We hope you find this supplement as useful and helpful as we expect you will. Let us know what you think.

Jim Brown, Editor

Your free ticket to DB/EXPO '93!

THE WORLD'S LARGEST
DATABASE
CLIENT-SERVER
& INFORMATION
TECHNOLOGY
EXPOSITION AND CONFERENCE

DB O 93

Only DB/EXPO brings together – in one place – all the latest developments in relational databases, object technology, client/server computing, new CASE tools and much more. You no longer have to go to several different events to get the information you need. It's all here – at DB/EXPO '93!

See Over 500 Leading Edge Exhibits – Free!

Partial List:
Acius
Andyne Computing
AMDAHL
Andersen Consulting
Apple Computer
Bachman
Borland
Cincom
Cognos
Computer Associates
Cortex
Cross Access
DataEase
DEC
Encore Computer
Gupta Technologies
Harlequin
Hewlett Packard
IBM
Information Builders
Ingres
Intersystems
JYACC
Knowledgeware
Microsoft
Must Software
NCR
NeXT
Novell
Objectivity
Ontos
Oracle
Parc Place
Pick Systems
Platinum Software
Progress Software
Raima
SAP America
Sequent Computer
Servio
Smartstar
Software AG
Software Publishing
Tandem
Unidata
Uniface
Versant
Wang Lab
Wyse Technology
and many others!

For your FREE PASS call today:
1-800-2DB-EXPO

Five Exciting Keynote Addresses – Free

You'll hear the latest thinking from visionaries who are shaping the future of information technology!

Colin White
DB/EXPO Conference Director
"Information Technology: Yesterday, Today, Tomorrow"
Tuesday May 4, 9:00

Bill Gates
Microsoft Corporation
"Directions in Workgroup Computing"
Tuesday May 4, 9:45

Philippe Kahn
Borland International
"Application Development: Where are We Headed?"
Wednesday May 5, 9:45

Steven Jobs
NeXT Computer
"Object-Oriented Application Development"
Wednesday May 5, 12:30

Larry Ellison
Oracle Corporation
"Industry Directions"
Thursday May 6, 1:00

Umang Gupta
Gupta Corporation
"Client/Server Comes of Age"

Dozens of New Product Introductions – Free

At DB/EXPO you'll know about the newest and hottest products the minute they're announced. So, you can follow up immediately on products most critical to you.

One-on-One Consulting – Free

Co-Sponsored by **Marathon Systems**

At DB/EXPO's unique Product Education Center, experts help you define your requirements and choose the most relevant exhibits. You even get tips on what to look for and guidance on how to best implement your new ideas.

Special Free Bonus! Experience Virtual Reality

Co-Sponsored by **Digital Media Labs**

You'll go from the real world of the exhibit hall into a world that isn't there. Or is it? This mind-expanding demonstration has been created just for DB/EXPO.

User Group Events – Free

Many of the most prestigious user groups are holding meetings at DB/EXPO '93!

Exhibit Location, Dates and Hours

Moscone Center, San Francisco, CA

Tuesday, May 4, 11:00 A.M. to 5:00 P.M.
Wednesday, May 5, 11:00 A.M. to 6:00 P.M.
Thursday, May 6, 11:00 A.M. to 4:00 P.M.

NDN

24 Hour Registration REGISTER TODAY *Via FAX (415) 966-8934*

To pre-register and receive your badge in the mail, complete this form and return on or before April 5, 1993. After April 5, please bring this completed form to the show (without this pass, on-site registration is $25.00). Please copy this form for additional registrants. Mail or fax c/o: NDN Enterprises 444 Castro Street, Suite 1101 Mountain View, CA 94041

21

Name:
Title: Mail Stop:
Company:
Address:
City: State: Zip:
Phone:

A. Your Title (circle one):
Information Systems/MIS:
17 Management (CIO, VP, DIR, MGR, SUPV)
18 Staff (SYS ANALYST, PROG, ARCHITECT, EDP, DBA, DEVELOPER)
Company Management
19 Corporate (CHRM, CEO, PRES, VP, DIR, GM, OWNER, PARTNER)
20 Financial (CFO, VP, TREAS)
21 Administrative (VP, DIR, MGR)
22 Marketing (VP, DIR, MGR)
Engineering/Manufacturing
23 Management (VP, DIR, PLANT MGR, PROD MGR)
24 Staff (ENGR, RESEARCH, GROUP LEADER, TECH, PRGR, SW DES)
25 Consultant
26 Educator

B. Total Number of Employees at Your Company:
AA 1 - 99 BB 100 - 499 CC 500 - 999
DD 1000 - 2999 EE 3000 - 9999 FF 10,000 +

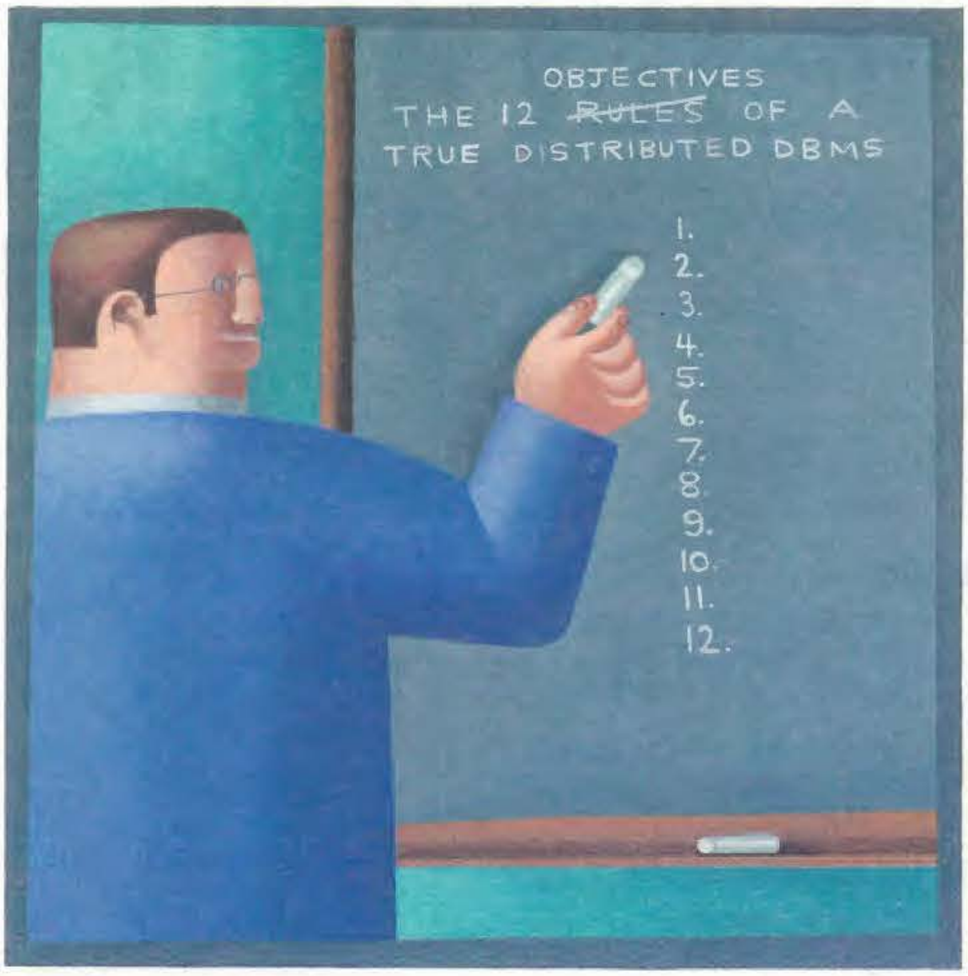
C. Your Interest:
A Mainframe DBMSs B LANs C CASE
D Expert Systems E End User Tools G UNIX DBMSs
H Micro DBMSs I Object Oriented Systems
M Database Machine O MAC P Client/Server
Q Windows R GUIs

D. Your Industry Group:
1 Developer of Computer Software
2 Manufacturer of Computer Hardware
3 Computer Reseller/Distributor 4 Transportation Services
5 Educational Institutions 6 Utilities
7 Trade/Wholesale/Retail
8 Construction/Mining/Agriculture
9 Other________ 10 Legal Services
11 Insurance 12 Banking
13 Government 14 Accounting
15 Communications 16 Medical

Please Register My Colleague, Too!

Name: Title: Company:
Address:
City: State: Zip:

A distributed DBMS lesson



BY DAVID MCGOVERAN

Outlining the key capabilities of a distributed DBMS.

A lot of buzzwords that attempt to define a distributed database management system have been flying around the DBMS industry during the last decade, leaving users confused about how to judge whether a DBMS is truly distributed.

But there are some steps users can take in navigating through the cloud of confusion. At the outset, users should strive to grasp some key distributed DBMS concepts. For instance, a distributed DBMS should have roots in relational DBMS technology because the relational database model lends itself to the fundamental principle of a distributed DBMS, enabling users to access and update data stored at multiple locations.

A distributed DBMS must also enable users to split a centralized database into many smaller databases that can be stored at multiple locations but appear as a single logical database. It should also provide tools for integrating existing databases into a single logical database.

Users can also brush up on the Twelve Objectives of Distributed Databases, which were authored by Hearldsburg, Calif.-based DBMS consultant Chris Date and spell out the core capabilities a distributed DBMS should have. While many people have tried to tightly define a distributed DBMS, Date has taken the most systematic approach.

There is no mandate that distributed DBMSs meet these 12 objectives, but many vendors strive to provide the listed capabilities. When he first wrote them in the late 1980s, Date referred to the objectives as rules, a decision he now regrets because the term "rules" implies there is such a mandate.

With the exception of noting that a distributed DBMS should be able to run over a variety of networking protocols, none of the objectives suggest a particular network topology or strategy. Network managers asked to support a well-implemented distributed DBMS need only be concerned with network installation, capacity and reliability issues. The distributed DBMS will automatically optimize use of network resources.

An objective view

According to Date, the first thing a distributed DBMS should provide is local autonomy, which means each database in the network must be managed separately. For instance, a local administrator must be able to control who can access data stored at that location as well as maintain the integrity of that data.

The local administrator should also be able to temporarily cede management control of the database to a remote location. The ability to transfer control comes in handy if tasks can be managed more efficiently by someone at a remote location.

Local autonomy prohibits failure of a database at any location from knocking out the entire distributed DBMS. Likewise, it enables the local database to keep running if network links to other databases are severed.

Date's second objective suggests a distributed DBMS should avoid dependence on any particular site. Depending too much on one location to perform critical operations goes against the concept of a distributed DBMS, which is to treat all locations as equals. Adherence to this objective makes the overall distributed DBMS less vulnerable to a single point of failure or bottlenecks.

In order to avoid reliance on a particular location, the distributed DBMS should be able to dispense such functions as data dictionary management, query processing, data concurrency control and recovery control among remote locations.

Date's third objective calls for distributed DBMSs to continuously operate even if one or more locations have been brought down. In particular, when software or hardware at one location is being upgraded or when locations and databases are being added or removed from the network, all other locations should be able to continue operating.

> Local autonomy prohibits failure of a database at any location from knocking out the entire distributed DBMS.

Meeting this objective requires a well-thought-out network design. If network links must be brought down to add, change or remove a database at one location, the links between other locations must remain operational. Likewise, the distributed DBMS should automatically reroute traffic around the site that has been brought down.

While these initial objectives lay the groundwork for a distributed DBMS, Date's fourth objective — location independence — starts to get to the heart of the matter. With location independence, users can move hardware, software and database resources around the net without having to rewrite applications to reflect the change. Instead, the changes are identified to the distributed DBMS by entering them in a data dictionary that all applications consult when trying to locate data.

Enabling physical elements to be moved around means users should view a distributed DBMS in logical terms. To set up this logical view, users create a set of symbolic names — known as logical constructs — for database objects and operations, as well as relationships between those objects and operations.

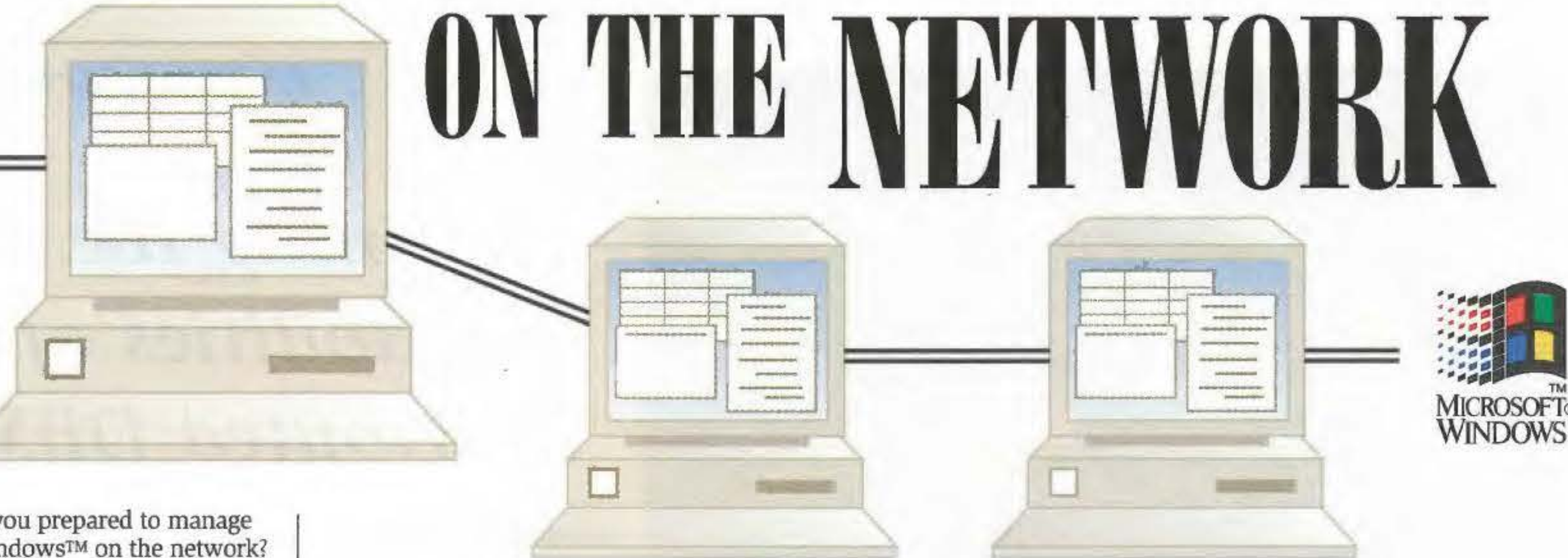
The key logical construct is a relational table. The implementation of this relational table is a physical construct that should be hidden from all users and appear to users as a set of rows and columns.

Other physical constructs include the disks used to store data, the methods used to access the data and the algorithms used to join tables stored on different nodes in the network. The relationship between physical and logical constructs is maintained by a database administrator (DBA) in a data dictionary or may be hard-coded in the distributed DBMS by the vendor.

When users want to reach database objects or perform an operation, they insert the logical construct name in the application, and the distributed DBMS automatically locates the database object or performs the operation. For example, when the logical construct name needed to create a new table is inserted into an application, the distributed DBMS automatically selects a physical storage format and location, and builds any necessary indices to optimize access to the table if it has such infor-

(continued on page S5)

MANAGING WINDOWS™ ON THE NETWORK



MICROSOFT® WINDOWS™

Are you prepared to manage Windows™ on the network? How will the launch of Windows NT™ impact your network tomorrow?

Today, over half the world's microcomputers and workstations are networked. And a rapidly increasing number of these computers are running Microsoft Windows as the user interface of choice.

As end-users embrace Windows and Windows based applications, a completely new set of challenges must be addressed by network professionals.

Managing Windows on the Network is a dynamic, information-packed new course offered by *Network World Technical Seminars.* As the critical issues faced by network professionals in managing the complexities of Windows are explored, you'll quickly discover sensible solutions for ensuring effective and efficient network operations in your own Windows-based environments.

YOU'RE IN GOOD COMPANY!

Following is a partial list of companies that have trained with *Network World...*

Dreyfus • Ameritech • AT&T Bell Labs • Motorola • Arthur Anderson • John Deere • Chipcom • New York Life • Bell Atlantic • Martin Marietta • Infonet • Miller Brewing • RJR Nabisco • Pratt & Whitney • Cabletron Systems • Nynex • Keystone • Dun & Bradstreet • NEC America • Federal Reserve Bank • Bank of America • ADP • Univ. of Wisconsin • EDS • Minolta Corp. • Delta Airlines • Dept. of Commerce • Toshiba • IBM • American Re • Rockwell • UCLA Medical Center • Ernst & Young • Northrop • AMGEN • American Express • Unisys • Dow Corning Corp. • Lockheed • Texas Instruments

NETWORK WORLD TECHNICAL SEMINARS

Co-sponsored by INFO WORLD

REGISTER TODAY FOR THE SEMINAR NEAREST YOU!

CALL 800-643-4668

OR

DIAL OUR FAX-BACK INFORMATION LINE FOR A COMPLETE SEMINAR OUTLINE AND REGISTRATION FORM

800-756-9430

When prompted request document #50.

ATTENDING THIS SEMINAR WILL HELP YOU...

- Optimize network performance
- Better configure your hardware environment
- Better manage user software access
- Quickly troubleshoot Windows on the network
- Manage Users and Groups in the Windows network environment
- Control security issues with Windows on the network
- Choose the best Windows installation method
- Understand installation and support options of Windows-based applications
- Explore third-party options
- Identify LAN specific problem areas
- Examine Windows NT™
  - its features and benefits
  - its interaction in your network environment
  - impact on your existing Windows investment

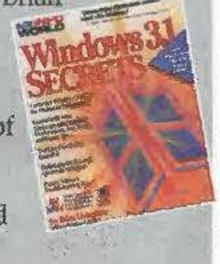
NETWORK WORLD TECHNICAL SEMINARS MAKE IT EASY FOR YOU TO ATTEND AND LEARN.

- One day intensive course minimizes time out of the office
- Multiple seminar dates and locations minimize travel costs

$350.00 registration fee includes:

Satisfaction 100% Guaranteed

- Comprehensive Seminar Workbook
- Copy of *Windows 3.1 Secrets®* by Brian Livingston
- Valuable diskette collection of software patches, utilities and tools
- Luncheon and break refreshments



(multiple attendee discounts available)

Note: If you can't attend, a full attendee materials kit is available for just $75.00!

- Highest quality expert speakers — *Managing Windows on the Network* is presented by Dr. Victor Rezmovic of Micro Automation Consultants
- Luncheon and breaks provide you with excellent opportunities to network with your peers
- Interactive format and comprehensive seminar materials give you information and ideas you can implement right away

DATES AND LOCATIONS

5/3/93	Boston, MA	Cambridge Marriott
5/4/93	New York, NY	Embassy Suites-Broadway
5/7/93	Chicago, IL	Embassy Suites-O'Hare
5/10/93	Los Angeles, CA	LAX Marriott
5/11/93	San Francisco, CA	S.F. Hilton
5/12/93	Seattle, WA	Sheraton
5/21/93	Washington, DC	Marriott Crystal City
6/2/93	Dallas, TX	InfoMart
6/3/93	Atlanta, GA	Hyatt Regency
6/4/93	Orlando, FL	Omni Orlando

Circle Reader Service #9

(continued from page S3)
mation as available disk space, expected disk load profiles and how the data in the new table will be used.

A programmer should never have to refer to network resources explicitly in a distributed DBMS application just because the application requests data from a remote database. Rather, the application should simply send messages to the distributed DBMS, which knows where the data is located and how to get to it. Thus, changes to the network will not require changes to application programs, only to entries in the distributed DBMS's data dictionary.

Location independence should be applied to data manipulation, data control, data definition and transaction management operations.

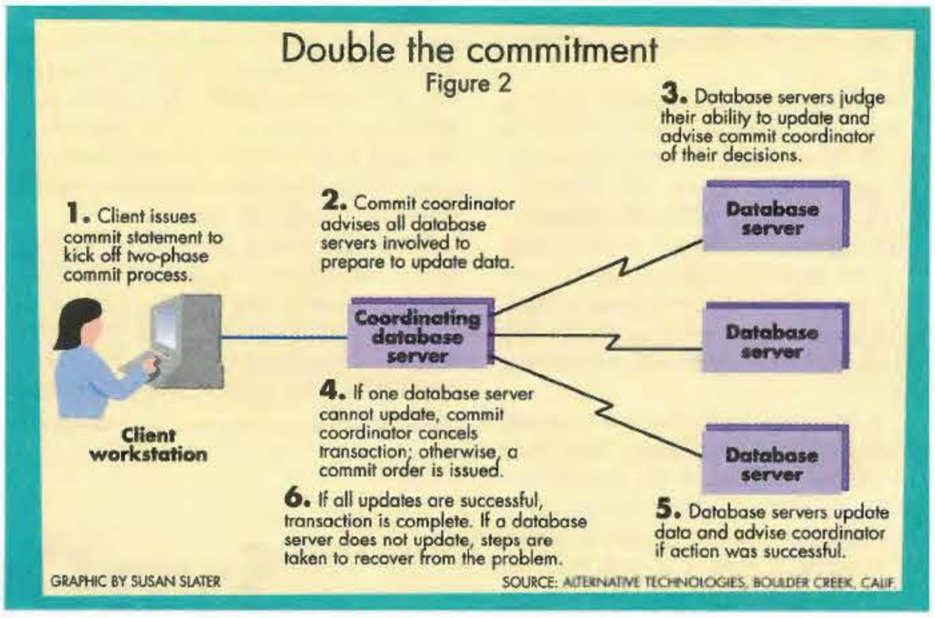
Current relational DBMS products may have a tough time meeting this objective because they frequently mix logical and physical constructs. For instance, most relational DBMS vendors mix references to physical resources with the data definition language component of SQL, thus permitting users to specify the physical location of a table in a "Create table" statement.

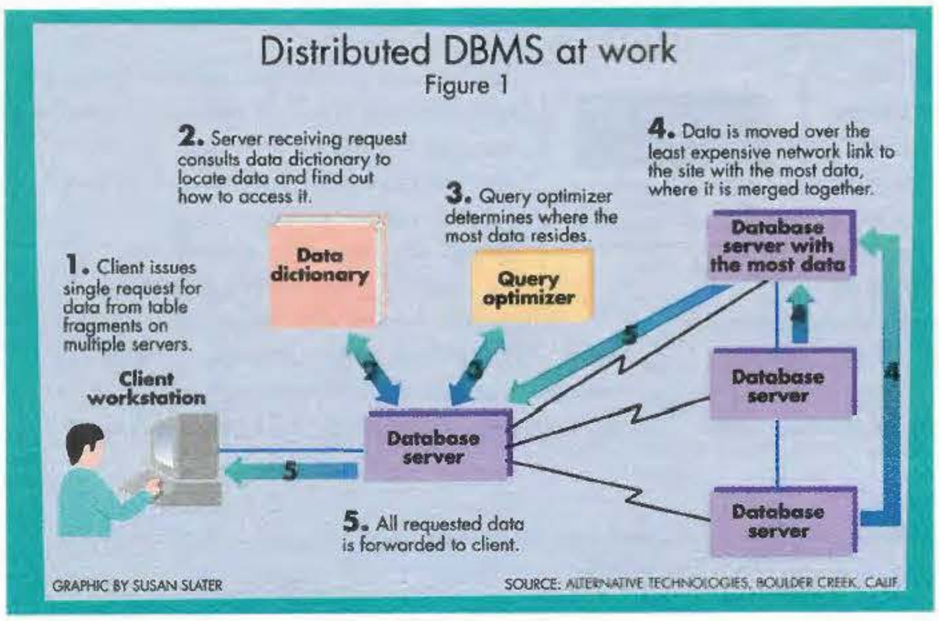
Strict separation of logical and physical constructs through a good data dictionary will permit a relational DBMS to support many powerful features, including transparent management of distributed data.

One way to achieve such separation is to provide a resource definition and management language to supply physical information to a distributed DBMS as well as enable it to make dynamic use of all resources based on declarative rules supplied by the DBA.

Taking the fifth

The fifth objective is fragmentation independence. To meet this objective, a distributed DBMS should be able to slice relational tables into horizontal or vertical pieces — called fragments — and not require end users or the DBA to know how the table was fragmented.





How a table is fragmented and where those fragments are located can be largely dynamic and based on access patterns. For example, an accounting database can be fragmented by region with each fragment stored in the regional office that uses it most often. However, each regional office could still access data from all of the other fragments on an ad hoc basis.

Meeting this objective also allows a distributed DBMS to automatically balance processing loads among network nodes and can significantly reduce the amount of data that must move from site to site.

Physical copies of a table or a table fragment that are stored at multiple locations and kept synchronized by a distributed DBMS are called replicates, and are created and maintained through a process called replication. Date addressed replication in his sixth objective, replication independence. There are a number of methods that a distributed DBMS can use to keep replicated data synchronized.

Theoretically, when data in one copy of a replicated database is updated, the change should immediately be propagated to all other copies. This is called synchronous update propagation.

In reality, this process can exact a toll on the network. So a variety of mechanisms have been proposed to relax this automatic update requirement. One proposal is to use snapshots of data in which the information is copied at a particular time, such as once a day, and users work with that copy. The copies are brought into synchronization with the original at specified intervals, a process known as asynchronous update propagation and one that is often more efficient in terms of network utilization.

All of these forms of replication should be transparent to users and applications. Users should also be able to create, destroy, alter and move replicated data at

(continued on page S6)

State of the distributed DBMS art

Today's commercially available distributed database management systems do a poor job of supporting all of Chris Date's Twelve Objectives of Distributed Databases.

In fact, vendors often put a twist on Date's objectives in an attempt to provide distributed DBMS products. For instance, site autonomy, a variation of Date's local autonomy objective, is now emerging. To some extent, local or site autonomy is easy to achieve because it is a native feature of a nondistributed DBMS — a vendor only has to add some distributed processing capabilities.

To comply with Date's next objective, most distributed DBMS implementations, such as Cincom Systems, Inc.'s Supra Server and Informix Software, Inc.'s Informix/Star, have eschewed reliance on a central site. One notable exception is Ingres' Ingres/Star, which uses a central-star node for distributed query optimization and dictionary access.

Support for continuous operation is considerably better. In fact, distributed DBMS vendors that support on-line transaction processing have provided continuous operation for some time.

Vendors are perhaps most noted for failing to meet the need for data and location independence. The reason is because there does not seem to be a product that enforces a global naming scheme that maps the logical name of a database or other resource to the actual physical name, regardless of its location.

While products such as Digital Equipment Corp.'s Rdb/VMS and Oracle Corp.'s Oracle 7 Distributed Database Extension provide good support for location independence, they offer access to multiple databases via peeks through the curtain instead of through a true method of linking databases.

Other than the forthcoming Oracle 7 Distributed Database Extension and Sybase, Inc. System 10 Replication Server, few products have any form of support for data fragmentation and replication. As a result, continuous operation, when fragments and replicas are created and destroyed, is not a relevant feature now.

Cincom's Supra Server, Informix's Informix/Star and Ingres' Ingres/Star provide distributed dictionary management and distributed query processing. But distributed views and integrity constraints are severely limited in most products.

Support for distributed transaction management also seems to be a problem. Sybase's SQL Server supports programmatic two-phase commit in which a programmer must write code to implement the details of the two-phase commit protocol into applications. However, it does not support transparent two-phase commit in which two-phase commit is handled automatically by the DBMS using functions written by the DBMS vendor.

Even when automatic two-phase commit is supported, global consistency seems to be a problem. For example, Oracle 7 supports automatic two-phase commit but does not support declarative referential integrity constraints that reference multiple sites.

In an effort to support open systems, most commercial relational DBMSs run under a variety of operating systems on a number of hardware platforms and support various networks. The main exceptions to this are proprietary relational DBMSs such as DEC's Rdb/VMS, which is expected to run under Microsoft Corp.'s Windows NT; IBM's DB2, although IBM recently announced the OS/2-based DB2/2; the Open Software Foundation, Inc.'s OSF/1; and Tandem Computer, Inc.'s NonStop SQL.

On one hand, de jure standards such as ANSI's SQL and the SQL Access Group's SQL/Call Level Interface, as well as proprietary de facto standards such as Borland International, Inc.'s Integrated Database Application Program Interface, IBM's Distributed Relational Database Architecture, Microsoft's Open Database Connectivity and Oracle's Glue have helped improve DBMS independence.

On the other hand, vendors still modify their products in an attempt to differentiate them from the competition. As a result, heterogeneous distributed DBMSs tend to provide least-common-denominator-functionality.

Besides using standard interfaces, DBMSs can be interconnected via a gateway, such as Apple Computer, Inc.'s Data Access Language, Information Builders, Inc.'s EDA/SQL, Oracle's SQL*Connect, Sybase's Open Server and OmniServer, and TechGnosis, Inc.'s SequeLink. But gateways are not generally designed to support distributed transactions.

— David McGoveran

(continued from page S5)
any time without affecting system operation.

Like fragmentation independence, replication independence permits the distributed DBMS to load balance automatically and significantly reduce the amount of data that must move across the network.

For example, a table of product prices that is infrequently updated may be replicated at each site in a retail store chain. This eliminates the need to access data held in a remote database when looking up a price.

A distributed DBMS should also support distributed query processing, Date's seventh objective. To meet this objective, a distributed DBMS should enable users to issue a single SQL query to read or update data at multiple sites or to both read and update data at multiple sites. Even though SQL does not permit updates to more than one table in a single statement, that table might be distributed over multiple sites.

Distributed query processing is largely made possible via a query optimizer, which is extremely important to network managers. It is largely the ability of the distributed DBMS to optimize queries that determine how much additional load the network must support.

For example, a user can request the names of company vice presidents making $200,000 or more a year. If the query optimizer is poor, records containing information for all employees would be returned to the user's workstation, which would then sift through the records to find those that meet the search criteria. A better query optimizer would return only those records that met the search criteria (see Figure 1, page S5).

Similarly, if the distributed DBMS query optimizer is not sensitive to network loads and capacities, the data returned by a request may be routed over the most direct network path even if that path is heavily loaded. A better query optimizer would seek a less loaded indirect route to improve efficiency.

Pieces of eight

The eighth distributed DBMS objective, distributed transaction management, has been the subject of a great deal of discussion recently because the requirements for maintaining database consistency are more important in a distributed DBMS.

To support distributed transaction management, a distributed DBMS should support several types of transactions, including remote and distributed requests as well as remote and distributed transactions. It should also ensure data consistency and concurrency.

Remote requests allow a single SQL request to be processed at a single remote location. Distributed requests allow a transaction consisting of multiple SQL requests to be processed at multiple remote or local locations. Each request in a distributed request can be processed at multiple locations. Distributed requests allow tables from multiple locations to be accessed using a relational join or union operation in which data from different tables are merged. A true distributed DBMS must support distributed requests.

Remote transactions allow a single transaction consisting of multiple SQL requests to be processed at a single remote location. Distributed transactions allow a transaction consisting of multiple SQL requests to be processed at multiple local or remote locations. Each SQL request in a distributed transaction can be processed only at a single location, but different requests within the same transaction can be processed at different locations.

When distributed transactions include update statements, the distributed DBMS must make sure data remains consistent, so it borrows from the relational DBMS world to accomplish this task. In a relational DBMS, the requirement of database consistency implies that either all nonprocedural statements are executed in their entirety or none of them are. Likewise, all statements within a transaction must be executed successfully or none of them are. This property is called atomicity, and it must continue to be enforced in a distributed DBMS.

In a nondistributed DBMS, atomicity is typically ensured by some form of journaling, which permits the database to be restored to its original state in the event of an error. In a distributed DBMS, each site must have its own journaling mechanism if it is to be autonomous and robust. In order to maintain global database consistency, each database executes its portion of a distributed transaction in cooperation with all others. Therefore, if one fails, they must all fail.

The principal method of ensuring atomicity in a distributed DBMS is a two-phase commit protocol, which coordinates changes to multiple databases. With two-phase commit, a coordinating database ensures that every database involved in a transaction makes necessary updates or none of them do. If a problem such as a database failure occurs during updating, the coordinator instructs each database to return to the state it was in before the attempted update (see Figure 2, page S5).

While a two-phase commit protocol can detect most failures in committing distributed transactions, problems arise if the commit coordinator fails during the update procedure. So some method of coordinating recovery of all participating databases after a coordinator failure must be implemented. It is this aspect of the two-phase commit mechanism that differs most radically among commercial implementations and which greatly determines relative efficiency and ease of distributed database administration.

Another important database consistency concept is serializability. When the transactions of concurrent users are executed serially — meaning in sequential order — there is no possibility of interference between them. But the DBMS can, in effect, support only one user at a time, which slows down processing.

If the individual statements in these same transactions are allowed to interleave, but still do not interfere with each other, they are said to be serializable. This enables the DBMS to handle multiple users at once, thus speeding up processing.

There are many techniques by which a relational DBMS can guarantee that concurrent transactions are serializable, even though individual statements of the transactions are interleaved for efficiency and better concurrency.

At the least, a distributed DBMS must manage access to shared data consistently, regardless of data location. Most systems use some form of locking mechanism for concurrency control. Regardless of the mechanism used, a distributed DBMS vendor should provide users with formal proof that the mechanism used is capable of automatically and transparently enforcing serializability of transactions while permitting shared access to data.

Next on Date's list are three objectives that can be lumped together: hardware, operating system and net independence. Objectives 9, 10, and 11 are intended to ensure that the distributed DBMS hardware and software platforms can interoperate using several networking protocols.

Last is DBMS independence, which states the need for heterogeneous distributed DBMS support. This requires a common interface to each DBMS in a distributed DBMS environment. Such an interface would have to support common languages, including a data manipulation language, a data control language, a data definition language and a transaction management language, as well as common data types, error and information status codes, communications formats and protocols — or at least a means to translate between different implementations of these.

Wrap it up

In general, a distributed DBMS that supports all of these objectives will use network resources efficiently and be easy to administer. And many relational DBMS vendors have added distributed capabilities to their products (see "State of the distributed DBMS art," page S5). Yet, Alternative Technologies' research shows that distributed DBMSs make up less than 5% of the DBMS installed base.

There are several reasons why users have not flocked to distributed DBMSs. Currently, many users require distributed processing and read-only data access, as opposed to a full-fledged distributed DBMS. Also, the cost of the processing power and network links needed to support current implementations of distributed query processing are too great for most on-line applications. The result is that applications perform remote requests and remote transactions, avoiding distributed transactions and distributed requests.

Furthermore, commercial products have provided almost no support for the design, development and maintenance of a distributed DBMS and its applications. Vendors clearly do not yet understand the issues well enough to train their customers in the use of the technology.

Most importantly, vendors have failed to recognize the importance of integrating databases into a single logical view.

All is not lost, however. The functionality of commercial distributed DBMSs will improve significantly during the 1990s. Although it is tough to find a true distributed DBMS today, some of the functionality provided is impressive. For those few users that understand the limitations of today's products and currently have a need for a true distributed DBMS, current products do provide a way forward. But users should take it slow, plan well and insist on fidelity to the relational model. ♦

McGoveran is president of Alternative Technologies, a relational DBMS consultancy in Boulder Creek, Calif., and can be reached at (408) 425-1859.

Chris Date provided valuable advice for this article.