Chapter 24

Distributed DBMSs – Concepts and Design

Chapter 24 - Objectives

- **Concepts.**
- **Advantages and disadvantages of distributed databases.**
- **Functions and architecture for a DDBMS.**
- **Distributed database design.**
- **Levels of transparency.**
- **Comparison criteria for DDBMSs.**

Concepts

Distributed Database

A logically interrelated collection of shared data (and a description of this data), physically distributed over a computer network.

Distributed DBMS

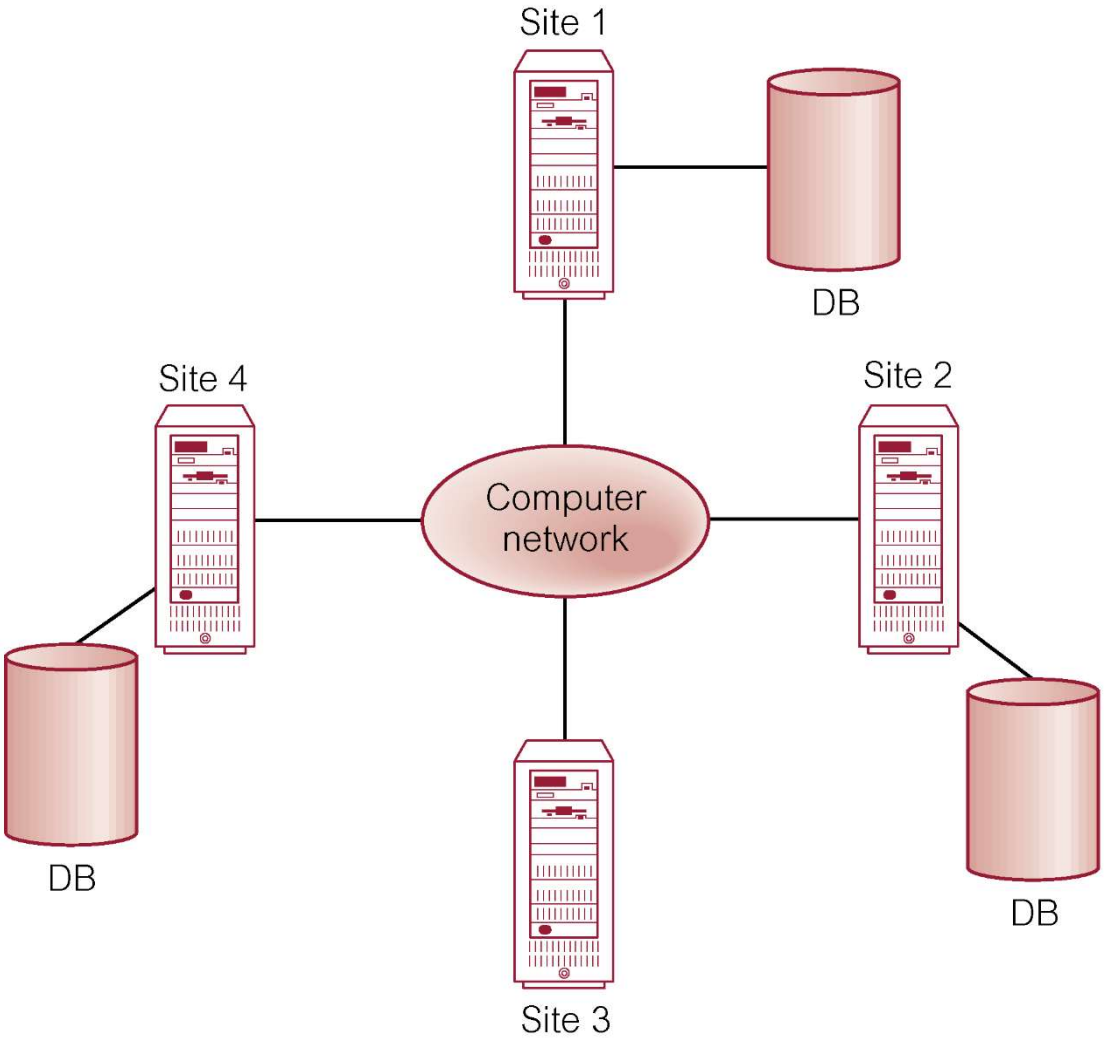
Software system that permits the management of the distributed database and makes the distribution transparent to users.

- A DDBMS consists of a single logical database that is split into a number of fragments.
- Each fragment is stored on one or more computers under the control of a separate DBMS *(Local Autonomy)*, with the computers connected by a communications network.
- Each site is capable of independently processing user requests that require access to local data and is also capable of processing data stored on other computers in the network.
- DDBMS, the system is expected to make the distribution transparent (invisible) to the user.

Concepts

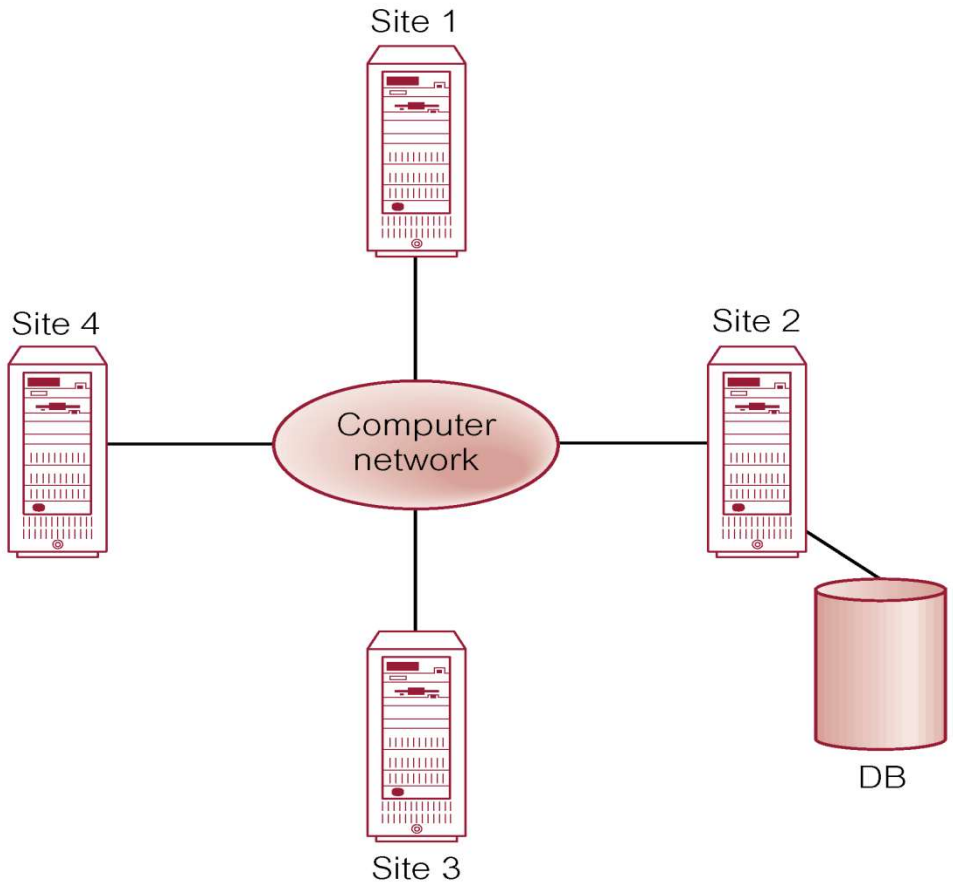
- Collection of logically-related shared data.
- Data split into fragments.
- Fragments may be replicated.
- Fragments/replicas allocated to sites.
- Sites linked by a communications network.
- Data at each site is under control of a DBMS.
- DBMSs handle local applications autonomously.
- Each DBMS participates in at least one global application.

Distributed DBMS



Distributed Processing

A centralized database that can be accessed over a computer network.



Parallel DBMS

A DBMS running across multiple processors and disks designed to execute operations in parallel, whenever possible, to improve performance.

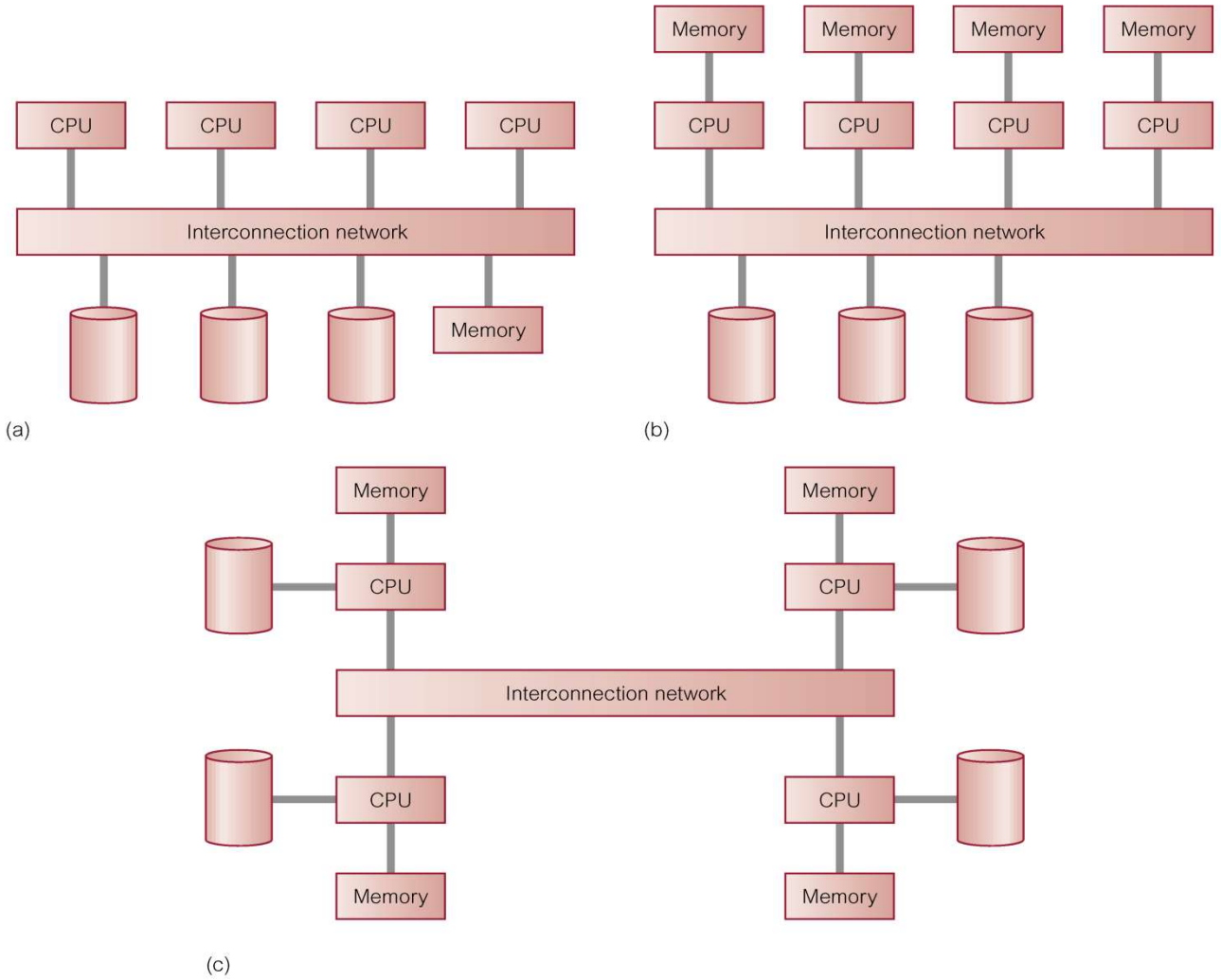
- Based on premise that single processor systems can no longer meet requirements for cost-effective scalability, reliability, and performance.

- Parallel DBMSs link multiple, smaller machines to achieve same throughput as single, larger machine, with greater scalability and reliability.

Main architectures for parallel DBMSs are:

(a) shared memory
(b) shared disk
(c) shared nothing

Parallel DBMS



(a) shared memory

(b) shared disk

(c) shared nothing

Advantages of DDBMSs

- Reflects organizational structure
- Improved shareability and local autonomy
- Improved availability
- Improved reliability
- Improved performance
- Economics
- Modular growth

Disadvantages of DDBMSs

- Complexity
- Cost
- Security
- Integrity control more difficult
- Lack of standards
- Lack of experience
- Database design more complex

Types of DDBMS

- Homogeneous DDBMS
- Heterogeneous DDBMS

Homogeneous DDBMS

- All sites use same DBMS product.
- Much easier to design and manage.
- Approach provides incremental growth and allows increased performance.

Heterogeneous DDBMS

- Sites may run different DBMS products, with possibly different underlying data models.
- Occurs when sites have implemented their own databases and integration is considered later.
- Translations required to allow for:
  - Different hardware: the translation is straightforward and involves only change of codes and word lengths
  - Different DBMS products: If the DBMS products are different, the translation is complicated involving the mapping of data structures.
    - Relations in the relational data model are mapped to records and sets in the network model.
    - SQL SELECT statements are mapped to the network ***FIND*** and ***GET*** statement
  - Different hardware and different DBMS products: Extremely complex
- Typical solution is to use *gateways*.
  - To convert the language and model of each different DBMS into the language and model of the relational system.

Functions of a DDBMS

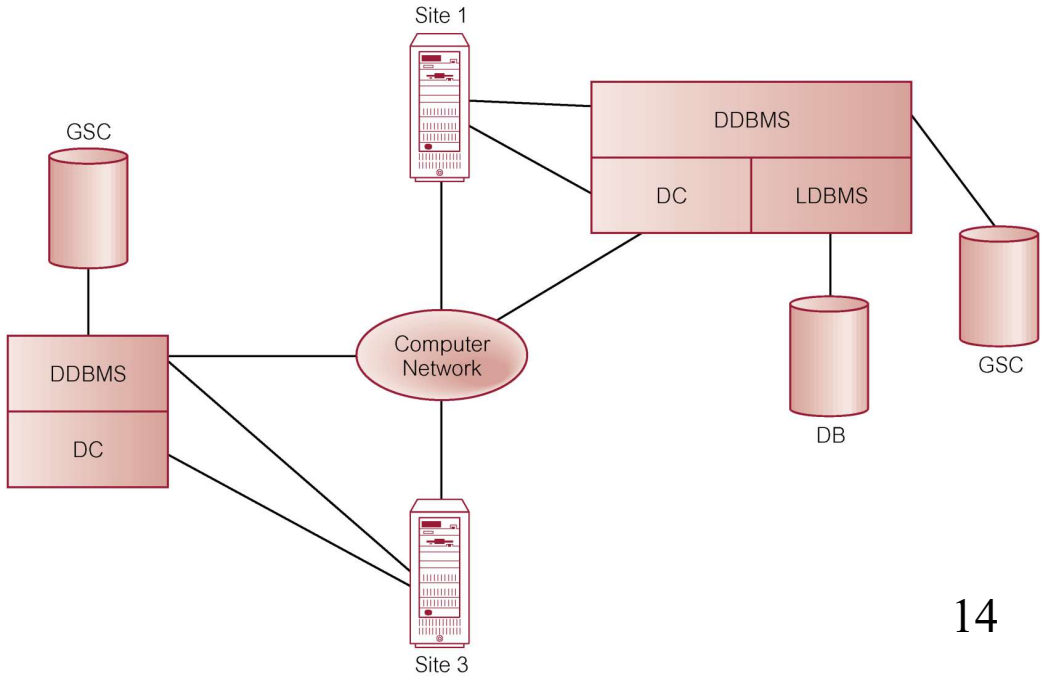
- Expect DDBMS to have at least the functionality of a DBMS and following additional functionalities:
  - ***Extended communication services*** to provide access to remote sites and allow the transfer of queries and data among the sites using a network
  - ***Extended system catalog*** to store data distribution details
  - ***Distributed query processing*** including query optimization and remote data access;
  - ***Extended security control*** to maintain appropriate authorization/access privileges to the distributed data
  - ***Extended concurrency control*** to maintain consistency of distributed and possibly replicated data;
  - ***Extended recovery services*** to take account of failures of individual sites and the failures of communication links.

Components of a DDBMS

- **Local DBMS (LDBMS) component:** for controlling the local data at each site that has a database
- **Data communications (DC) component:** contains information about the sites and the links.
- **Global system catalog (GSC):** information specific to the distributed nature of the system, such as the fragmentation, replication, and allocation schemas

Distributed DBMS (DDBMS) component: is the controlling unit of the entire system and manages:

- Distribution transparency
- Transaction transparency
- Performance transparency
- DBMS transparency.



Distributed Database Design

- Three key issues:
  - Fragmentation
  - Allocation
  - Replication

Fragmentation

Relation may be divided into a number of sub-relations, which are then distributed.

Allocation

Each fragment is stored at site with “optimal” distribution.

Replication

Copy of fragment may be maintained at several sites.

Fragmentation

- **Definition and allocation of fragments carried out strategically to achieve:**
  - **Locality of Reference:** data should be stored close to where it is used. If a fragment is used at several sites, replicate it.
  - **Improved Reliability and Availability:** Improved by replication: there is another copy of the fragment available at another site in the event of one site failing.
  - **Improved Performance.**
  - **Balanced Storage Capacities and Costs:** must be balanced against *locality of reference.*
  - **Minimal Communication Costs:** consider *locality of reference.*
- **Involves analyzing most important applications, based on quantitative/ qualitative information.**
- **Quantitative information may include:**
  - **frequency with which an application is run;**
  - **site from which an application is run;**
  - **performance criteria for transactions and applications.**
- **Qualitative information may include transactions that are executed by application, type of access (read or write), and predicates of read operations.**

Data Allocation

- Four alternative strategies regarding placement of data:
  - Centralized,
  - Partitioned (or Fragmented),
  - Complete Replication,
  - Selective Replication.

Centralized: Consists of single database and DBMS stored at one site with users distributed across the network.

Partitioned: Database partitioned into disjoint fragments, each fragment assigned to one site.

Complete Replication: Consists of maintaining complete copy of database at each site.

Selective Replication: Combination of partitioning, replication, and centralization.

Comparison of Strategies for Data Distribution

	LOCALITY OF REFERENCE	RELIABILITY AND AVAILABILITY	PERFORMANCE	STORAGE COSTS	COMMUNICATION COSTS
Centralized	Lowest	Lowest	Unsatisfactory	Lowest	Highest
Fragmented	High[a]	Low for item; high for system	Satisfactory[a]	Lowest	Low[a]
Complete replication	Highest	Highest	Best for read	Highest	High for update; low for read
Selective replication	High[a]	Low for item; high for system	Satisfactory[a]	Average	Low[a]

[a]Indicates subject to good design.

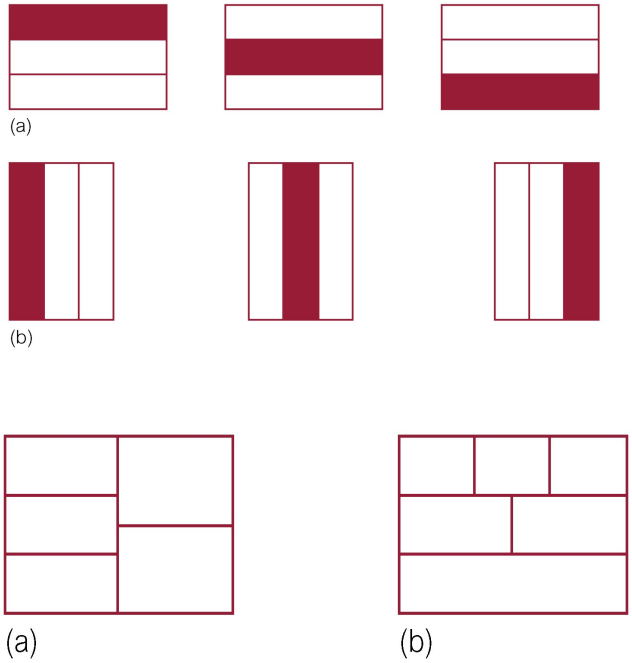
Why Fragment?

- **Usage**
  - Applications work with views rather than entire relations.
- **Efficiency**
  - Data is stored close to where it is most frequently used.
  - Data that is not needed by local applications is not stored.
- **Parallelism**
  - With fragments as unit of distribution, transaction can be divided into several subqueries that operate on fragments.
- **Security**
  - Data not required by local applications is not stored and so not available to unauthorized users

- **Disadvantages**
  - Performance,
  - Integrity.

Types of Fragmentation

- **Four types of fragmentation:**
  - **Horizontal**
    - Groups together tuples that have some common property; for example, the tuples are all used by the same application or at the same site.
  - **Vertical**
    - Groups together the attributes in a relation that are used jointly by the important transactions.
  - **Mixed**
    - Consists of a horizontal fragment that is subsequently vertically fragmented, or a vertical fragment that is then horizontally fragmented.
  - **Derived**
    - Semijoin parent and child relations. Most common records could be fragmented together
- **Other possibility is no fragmentation:**
  - If relation is small and not updated frequently, may be better not to fragment relation.



Types of Fragmentation

Fragment P_1

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003

Fragment P_2

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40	SG14	B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

Figure 24.10 Horizontal fragmentation of PropertyForRent by property type.

Fragment S_1

staffNo	position	sex	DOB	salary
SL21	Manager	M	1-Oct-45	30000
SG37	Assistant	F	10-Nov-60	12000
SG14	Supervisor	M	24-Mar-58	18000
SA9	Assistant	F	19-Feb-70	9000
SG5	Manager	F	3-Jun-40	24000
SL41	Assistant	F	13-Jun-65	9000

Fragment S_2

staffNo	fName	lName	branchNo
SL21	John	White	B005
SG37	Ann	Beech	B003
SG14	David	Ford	B003
SA9	Mary	Howe	B007
SG5	Susan	Brand	B003
SL41	Julie	Lee	B005

Figure 24.11 Vertical fragmentation of Staff.

Fragment S_1

staffNo	position	sex	DOB	salary
SL21	Manager	M	1-Oct-45	30000
SG37	Assistant	F	10-Nov-60	12000
SG14	Supervisor	M	24-Mar-58	18000
SA9	Assistant	F	19-Feb-70	9000
SG5	Manager	F	3-Jun-40	24000
SL41	Assistant	F	13-Jun-65	9000

Fragment S_{21}

staffNo	fName	lName	branchNo
SG37	Ann	Beech	B003
SG14	David	Ford	B003
SG5	Susan	Brand	B003

Fragment S_{22}

staffNo	fName	lName	branchNo
SL21	John	White	B005
SL41	Julie	Lee	B005

Fragment S_{23}

staffNo	fName	lName	branchNo
SA9	Mary	Howe	B007

Figure 24.12 Mixed fragmentation of Staff.

Fragment P_3

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40	SG14
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14

Fragment P_4

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41

Fragment P_5

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9

Figure 24.13 Derived fragmentation of PropertyForRent based on Staff.

Date's 12 Rules for a DDBMS

0. Fundamental Principle

   To the user, a distributed system should look exactly like a nondistributed system.

1. Local Autonomy
2. No Reliance on a Central Site
3. Continuous Operation
4. Location Independence
5. Fragmentation Independence
6. Replication Independence

Date's 12 Rules for a DDBMS

7. Distributed Query Processing
8. Distributed Transaction Processing
9. Hardware Independence
10. Operating System Independence
11. Network Independence
12. Database Independence

- **Last four rules are ideals.**