Distributed DBMS

- Concepts
- Advantages and disadvantages of distributed databases.
- Functions of DDBMS.
- Distributed database design.

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.

Concepts

- Distributed database system (DDBS) = DDB + D–DBMS
- A distributed database system consists of loosely coupled sites that share no physical component
- Database systems that run on each site are independent of each other
- Transactions may access data at one or more sites

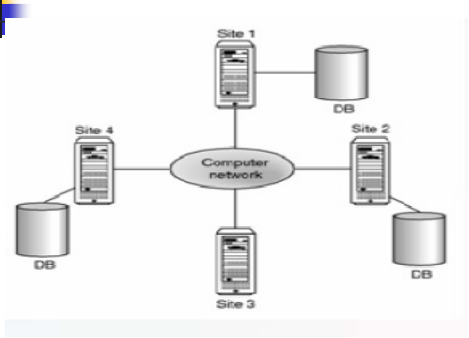
Concepts

- Collection of logically-related shared data.
- Data split into fragments.
- Fragments may be replicated.
- Fragments/replicas allocated to sites.

Concepts

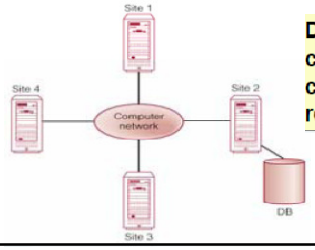
- Sites linked by a communications network.
- Data at each site is under control of a DBMS.
- DBMSs handle local applications autonomously.
- Each DBMS participates in at least one global application.

Distributed DBMS



Distributed Processing

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



Data physically centralised, even if it can be accessed remotely.

Advantages of DDBMSs

- Organizational Structure
- 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
  - Resembles a centralised DB, but data is distributed across a number of sites in a network
  - All sites use same DBMS product.
  - Has multiple data collections
  - It integrates multiple data resources
  - Much easier to design and manage.

Heterogeneous DDBMS

- Sites may run Different hardware, DBMS products, Data model or Combination of above
- Occurs when sites have implemented their own databases and integration is considered later.

Heterogeneous DDBMS

- Complete local autonomy
- Translations required to allow for:
  - Different hardware.
  - Different DBMS products.
  - Different hardware and different DBMS products.

Federated Database System

- Cross between distributed and centralized DBMS
- Distributed for global users and
- Centralized for local users

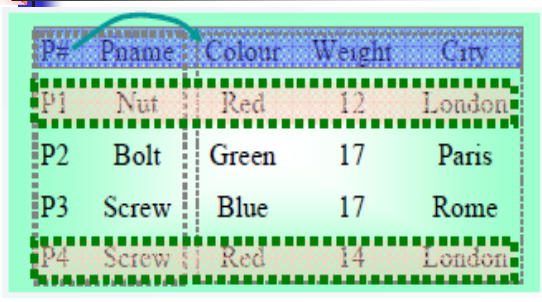
Functions of a DDBMS

- Expect DDBMS to have at least the functionality of a centralized DBMS.
- Also to have following functionality:
  - Extended communication services.
  - Extended Data Dictionary.
  - Distributed query processing.
  - Extended concurrency control.
  - Extended recovery services.

Distributed Database Design

- 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.

Distributed Database Design



P#	Pname	Colour	Weight	City
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London

Fragmentation

- Definition and allocation of fragments carried out strategically to achieve:
  - Locality of Reference
  - Improved Reliability and Availability
  - Improved Performance
  - Balanced Storage Capacities and Costs
  - Minimal Communication Costs.

Correctness of Fragmentation

- Completeness
  - If relation R is decomposed into fragments R1, R2, ... Rn, each data item that can be found in R must appear in at least one fragment.

Correctness of Fragmentation

- Reconstruction
  - Must be possible to define a relational operation that will reconstruct R from the fragments.
  - Reconstruction for horizontal fragmentation is Union operation and Join for vertical .

Correctness of Fragmentation

- Disjointness
  - If data item di appears in fragment Ri, then it should not appear in any other fragment.
  - Exception:
    - vertical fragmentation, where primary key attributes must be repeated to allow reconstruction.

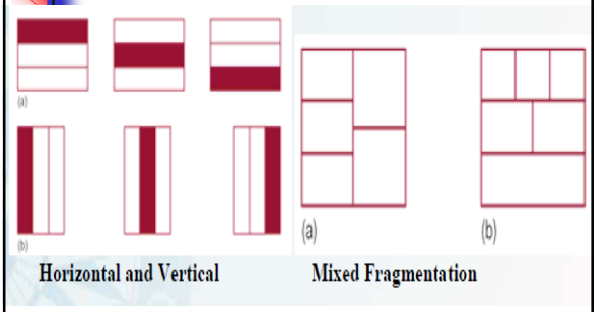
Correctness of Fragmentation

- For horizontal fragmentation, data item is a tuple
- For vertical fragmentation, data item is an attribute.

Types of Fragmentation

- Four types of fragmentation:
  - Horizontal
  - Vertical
  - Mixed
  - Derived.
- Other possibility is no fragmentation:
  - If relation is small and not updated frequently, may be better not to fragment relation.

Fragmentation



Fragmentation

- Reconstructing the original relation from the vertical fragments is done via a suitable ***join*** operation, &
- from horizontal fragments via the ***union*** operation.

Fragmentation Example

- *EMPLOYEE (Empnum, Name, Deptnum, Salary, Taxes)*
- Horizontal fragmentation
  - EMPLOYEE1 = $\sigma_{Empnum<=3}$ EMPLOYEE
  - EMPLOYEE2 = $\sigma_{Empnum>3}$ EMPLOYEE
  - Reconstruction requires a union:
    - EMPLOYEE = EMPLOYEE1 $\cup$ EMPLOYEE2

Fragmentation Example

- Vertical fragmentation:
  - EMPLOYEE1 = $\Pi_{EmpNum,Name}$ EMPLOYEE
  - EMPLOYEE2 = $\Pi_{EmpNum,DeptName,Salary,Tax}$ EMPLOYEE
  - Reconstruction requires an Equijoin on key values (natural join).
    - EMPLOYEE = EMPLOYEE1 $\bowtie$ EMPLOYEE2

Fragmentation

- Horizontal
  - Consists of a subset of the tuples of a relation.
  - Defined using Selection operation of relational algebra.
- Vertical
  - Consists of a subset of attributes of a relation.
  - Defined using Projection operation of relational algebra.

Fragmentation

- Mixed
  - Consists of a horizontal fragment that is vertically fragmented, or a vertical fragment that is horizontally fragmented.
  - Defined using *Selection* and *Projection* operations of relational algebra
-

Fragmentation

- Derived
  - A horizontal fragment that is based on horizontal
  - fragmentation of a parent relation.
  - Ensures that fragments that are frequently joined together are at same site.
  - Defined using *Semi-Join* operation of relational algebra.

Data Allocation

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

Data Allocation

- Centralized
  - Consists of single database and DBMS stored at one site with users distributed across the network. ***(not a true distribution)!***
- Partitioned
  - Database partitioned into disjoint fragments, each fragment assigned to one site.

Data Allocation

- Complete Replication
  - Consists of maintaining complete copy of database at each site.
- Selective Replication
  - Combination of partitioning, replication, and centralization

Database Replication

- Functionality of DDBMS is attractive but protocols & algorithms are complex and can cause problems that may outweigh advantages.
- Alternative and more simplify approach to data distribution is DB Replication

Database Replication

- Replication server:
  - Every major database vendor has replication solution.
- Database Replication:
  - the process of copying and maintaining database objects, such as relations, in multiple databases that make up a distributed database system.

Benefits of Database Replication

- Availability
- Reliability
- Performance
- Load Reduction
- Disconnected Computing
- Support Multiple Users
- Support Advanced Applications