

Distributed Computing

- Definition
- A number of autonomous processing elements (not necessarily homogeneous) that are interconnected by a computer network and that cooperate in performing their assigned tasks.

Distributed Computing

- Synonymous terms
 - distributed function
 - distributed data processing
 - multiprocessors/multicomputers
 - satellite processing
 - backend processing
 - dedicated/special purpose computers
 - timeshared systems
 - functionally modular systems

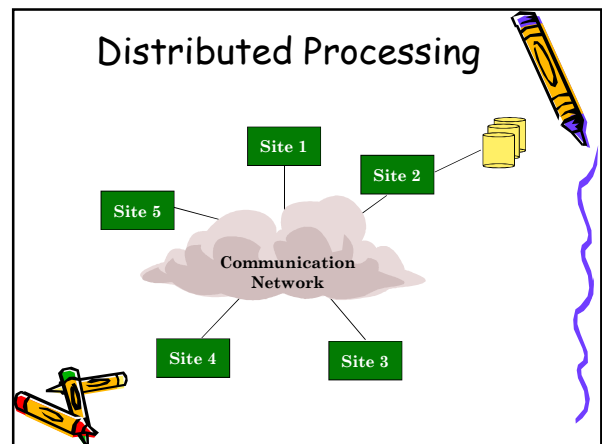
Distributed Database System

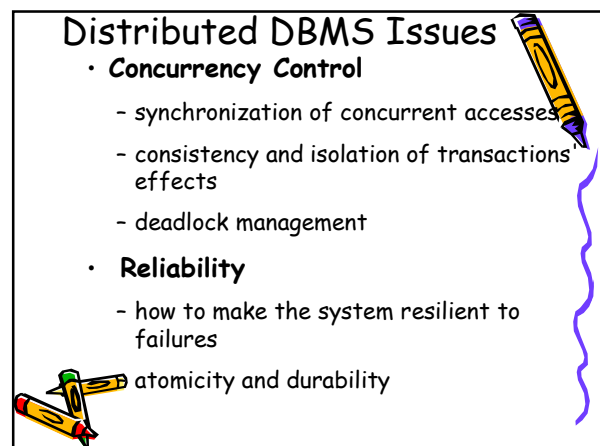
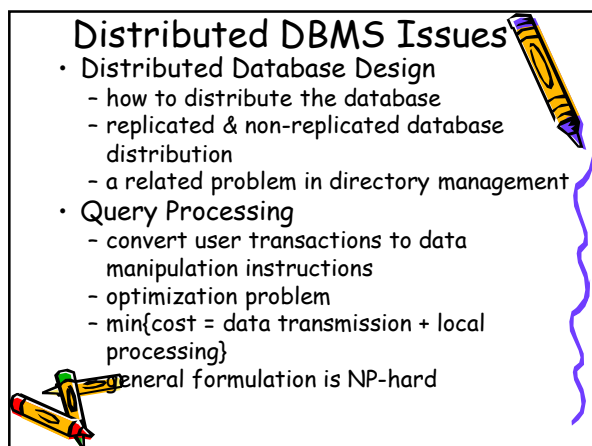
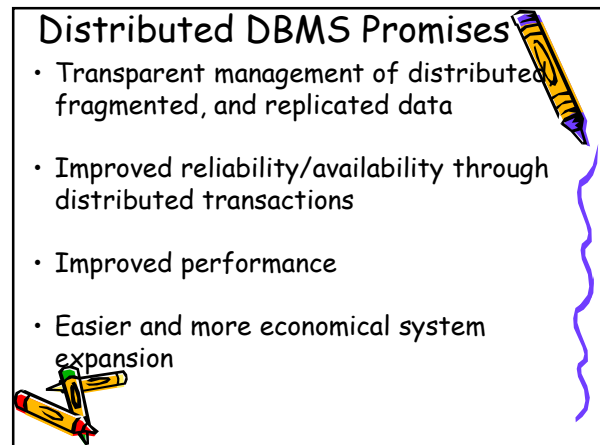
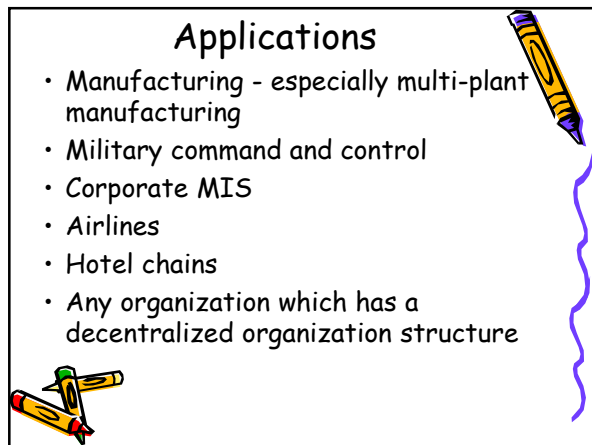
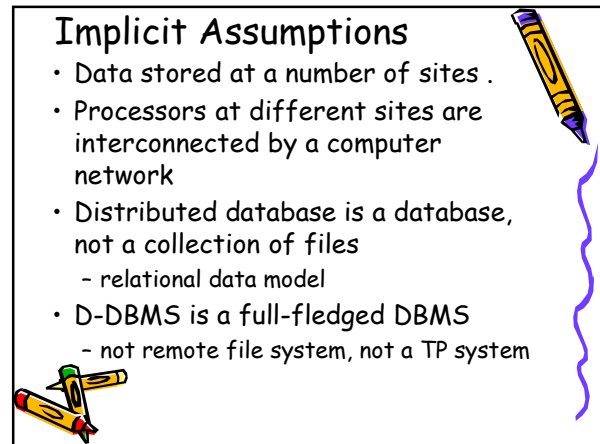
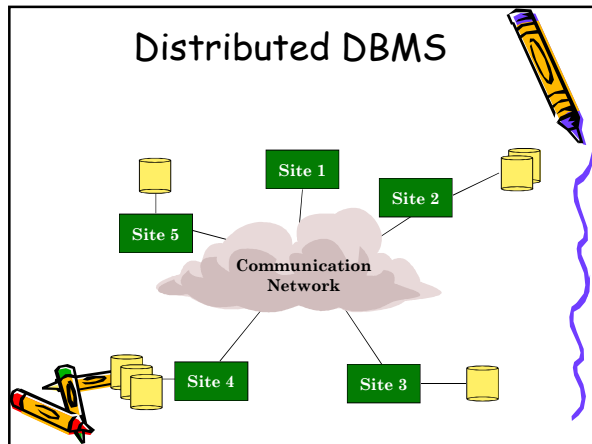
- A distributed database (DDB) is a collection of multiple, *logically interrelated* databases distributed over a *computer network*.
- A distributed database management system (D-DBMS) is the software that manages the DDB and provides an access mechanism that makes this distribution *transparent* to the users.

(DDBS) = DDB + D-DBMS

What is not a DDBS?

- A timesharing computer system
- A loosely or tightly coupled multiprocessor system
- Distributed Processing - this is a centralized database on a network node



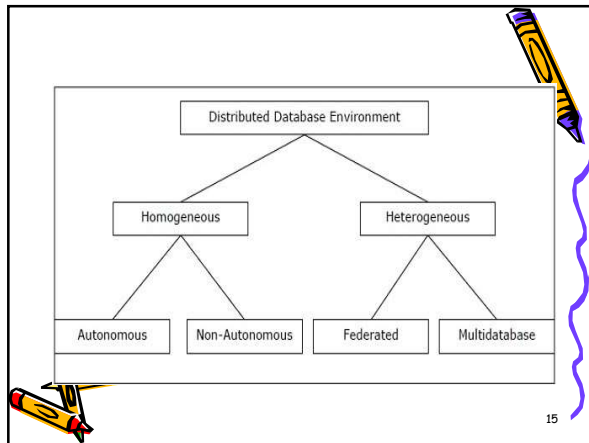


Functions of a DDBMS

- 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.
 - (For details refer to functions of DBMS)

Types of Distributed Database

- Dbs can be broadly classified into homogeneous and heterogeneous distributed database environments,
- each with further sub-divisions, as shown in the following illustration.



Homogeneous DDb

- all the sites use identical DBMS and operating systems.
- Its properties are
 - The sites use very similar software.
 - The sites use identical DBMS or DBMS from the same vendor.
 - Each site is aware of all other sites and cooperates with other sites to process user requests.
 - The database is accessed through a single interface as if it is a single database.

Types of Homogeneous DDb

- There are two types
 - **Autonomous** - Each database is independent that functions on its own.
 - They are integrated by a controlling application and use message passing to share data updates.
 - **Non-autonomous** - Data is distributed across the homogeneous nodes and a central or master DBMS co-ordinates data updates across the sites.

Heterogeneous DDb

- different sites have different operating systems, DBMS products and data models.
- Its properties are -
 - Different sites use dissimilar schemas and software.
 - The system may be composed of a variety of DBMSs like relational, network, hierarchical or object oriented.

Heterogeneous DDb

- Query processing is complex due to dissimilar schemas.
- Transaction processing is complex due to dissimilar software.
- A site may not be aware of other sites and so there is limited co-operation in processing user requests.



19

Types of Hetero DDb

- **Federated** - The heterogeneous database systems are independent in nature and integrated together so that they function as a single database system.
- **Un-federated** - The database systems employ a central coordinating module through which the databases are accessed.



20

DDBMS Architectures

- DDBMS architectures are developed depending on three parameters -
 - **Distribution** - It states the physical distribution of data across the different sites.
 - **Autonomy** - It indicates the distribution of control of the database system and the degree to which each constituent DBMS can operate independently.
 - **Heterogeneity** - It refers to the uniformity or dissimilarity of the data models, system components and databases.



21

Architectural Models

- Some of the common architectural models are -
 - Client - Server Architecture for DDBMS
 - Peer - to - Peer Architecture for DDBMS
 - Multi - DBMS Architecture



22

Client - Server

- This is a two-level architecture where the functionality is divided into servers and clients.
 - The server functions primarily encompass data management, query processing, optimization and transaction management.
 - Client functions include mainly user interface. However, they have some functions like consistency checking and transaction management.



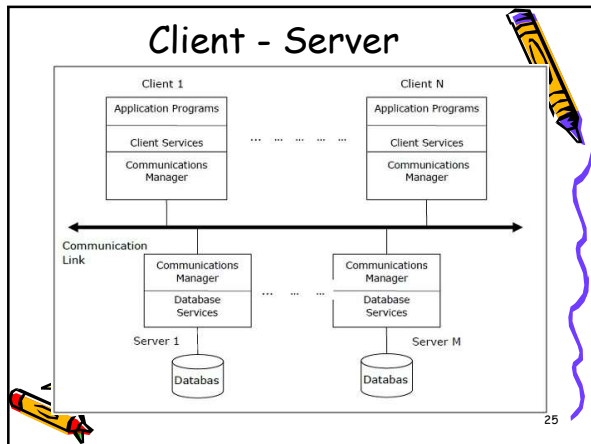
23

Client - Server

- The two different client - server architecture are -
 - Single Server Multiple Client
 - Multiple Server Multiple Client (shown in the following diagram)



24

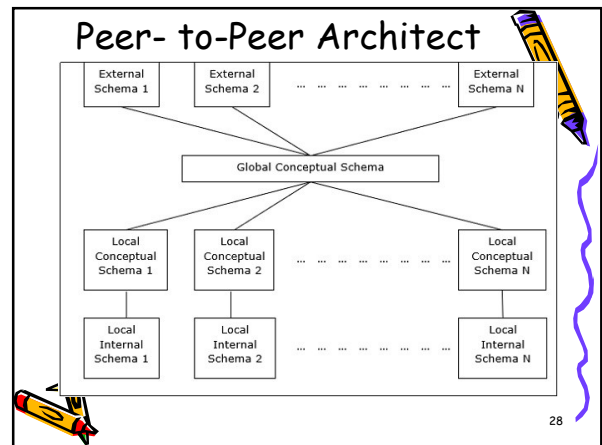


Peer- to-Peer Architect

- In these systems, each peer acts both as a client and a server for imparting database services.
- The peers share their resource with other peers and co-ordinate their activities.
- This architecture generally has four levels of schemas -

Peer- to-Peer Architect

- **Global Conceptual Schema** - Depicts the global logical view of data.
- **Local Conceptual Schema** - Depicts logical data organization at each site.
- **Local Internal Schema** - Depicts physical data organization at each site.
- **External Schema** - Depicts user view of data.



Multi - DBMS Architect


- This is an integrated database system formed by a collection of two or more autonomous database systems.
- Multi-DBMS can be expressed through six levels of schemas -
 - **Multi-database View Level** - Depicts multiple user views comprising of subsets of the integrated distributed database.

Multi - DBMS Architect

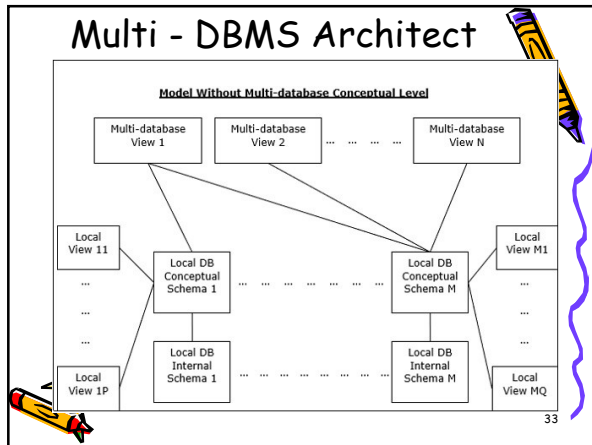
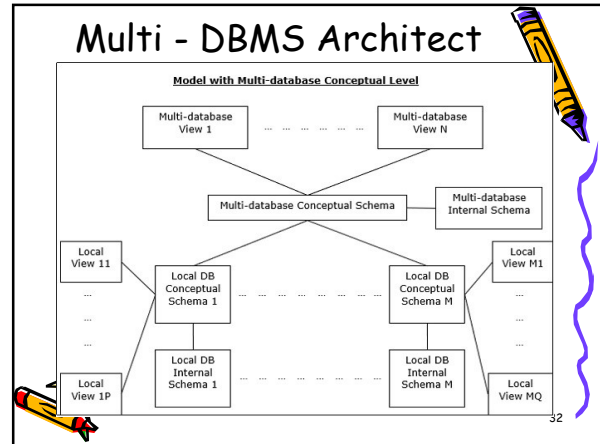
- **Multi-database Conceptual Level** - Depicts integrated multi-database that comprises of global logical multi-database structure definitions.
- **Multi-database Internal Level** - Depicts the data distribution across different sites and multi-database to local data mapping.
- **Local database View Level** - Depicts public view of local data.

Multi - DBMS Architect

- **Local database Conceptual Level** - Depicts local data organization at each site.
- **Local database Internal Level** - Depicts physical data organization at each site.
- There are two design alternatives for multi-DBMS -
 - Model with multi-database conceptual level.
 - Model without multi-database conceptual level.




31




Ddb Design

- **Fragmentation**
 - Relation may be divided into a number of sub-relations, which are then distributed
- **Replication**
 - Copy of fragment may be maintained at several sites.




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.
- **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.



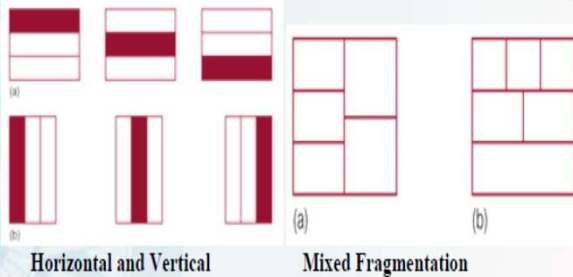
Correctness of Fragmentation

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

- 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.

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



Questions



46