分布式数据处理
Distributed Data Processing

陆桑璐

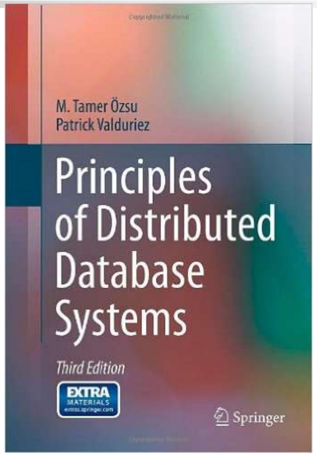
2018.3 – 2018.6

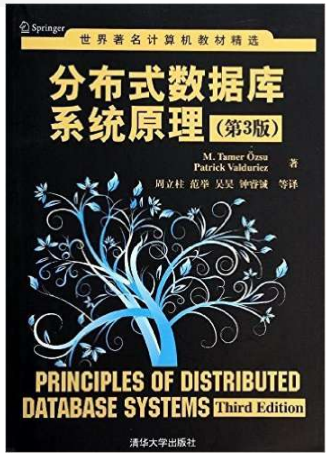
Course Information

- Lecture hour: Mar. ~ Jun., 2018
  - 星期一上午10:10 ~ 12:00
- Lecture location:
  - 仙II - 506
- Lecturer: 陆桑璐
  - Email: sanglu@nju.edu.cn
  - Phone: 89686757
  - Office: 仙林校区计算机楼615

Bibliography

Principles of Distributed Database Systems，
M. Tamer Özsu and Patrick Valduriez，Prentice Hall (1991, 1999, 2011)





- **Principles of Distributed Database Systems**分布式数据库系统原理**(**第三版**)**，周立柱等译 **(2014)**
- 分布式数据库管理系统实践，Saeed Rahimi，Frank Haug著，邱海艳等译（2014）
- 分布式数据库系统（原理与应用），申德荣、于戈等（2013）
- 分布式数据库系统及其应用**(**第二版**),** 邵佩英, 科学出版社(2005)
- 分布式数据处理**,** 谢立, 孙钟秀, 国防工业出版社(1990)

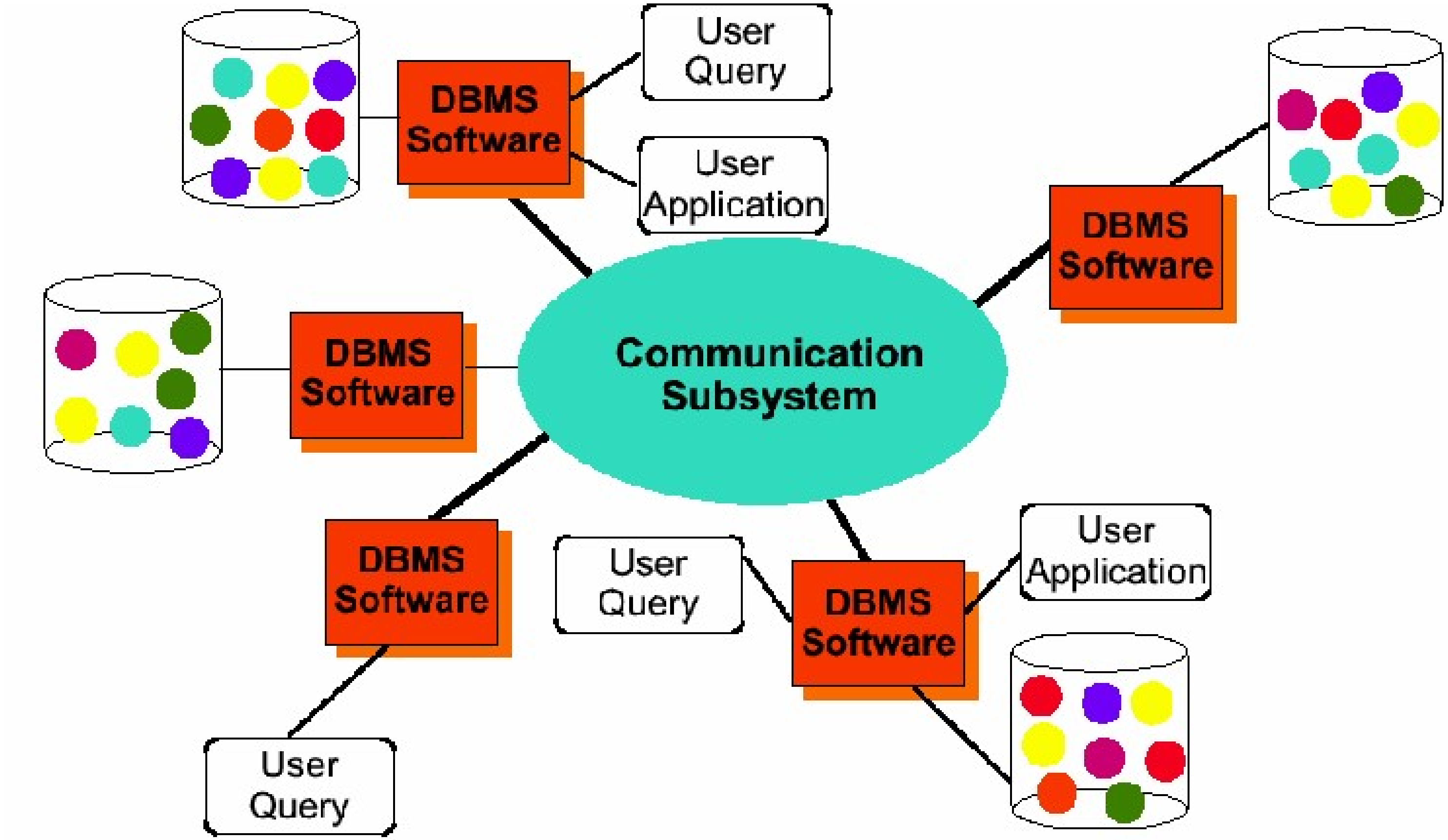
Course Objectives

- To introduction and explain the theory, algorithms, and methods that underly distributed DBMS
  - To address the principles of distributed data management
  - To address new and emerging issues

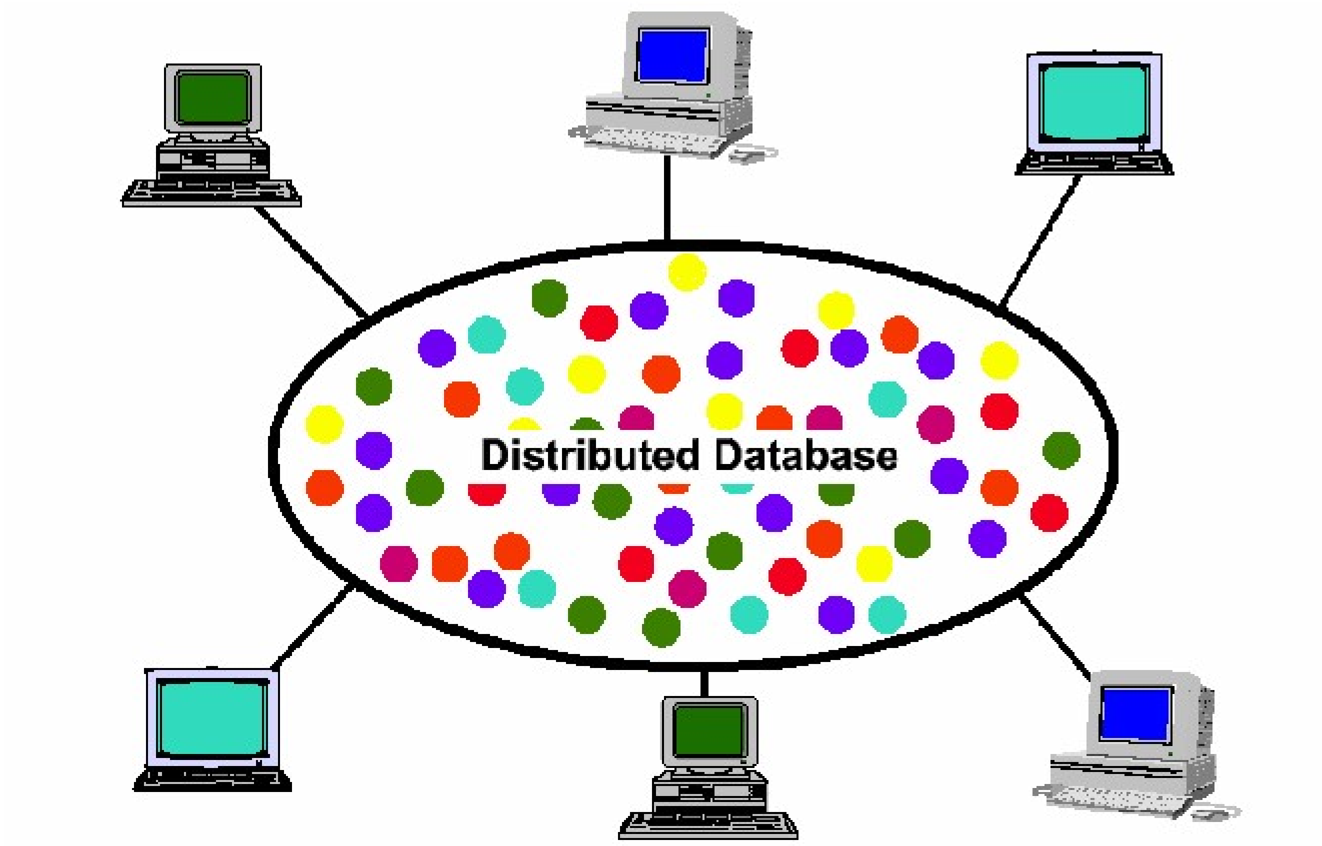
Outline

- Introduction
- Distributed DBMS Architecture
- Distributed DB Design
- Query Processing
- Transaction Management
- Current Issues

Distributed DBS



Distributed DBS – User View



Distributed Data Processing

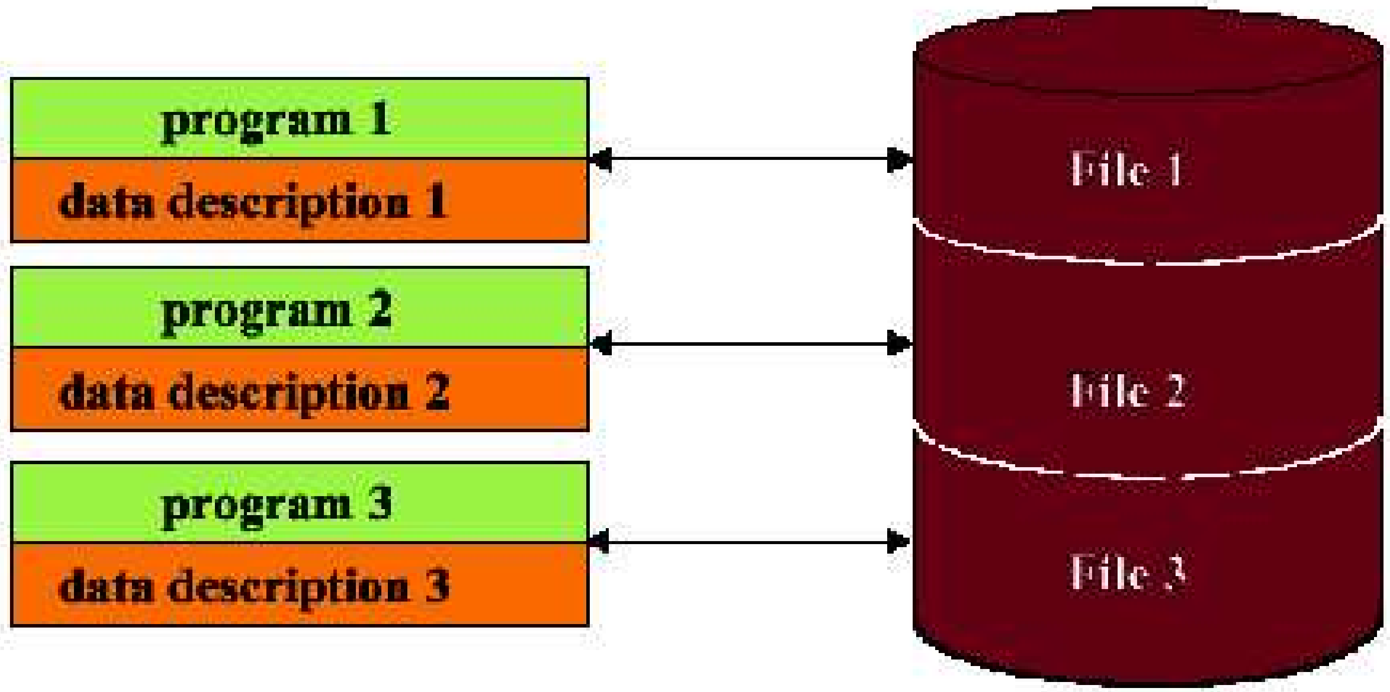
- <u>File System</u>

-><u>Database System</u>

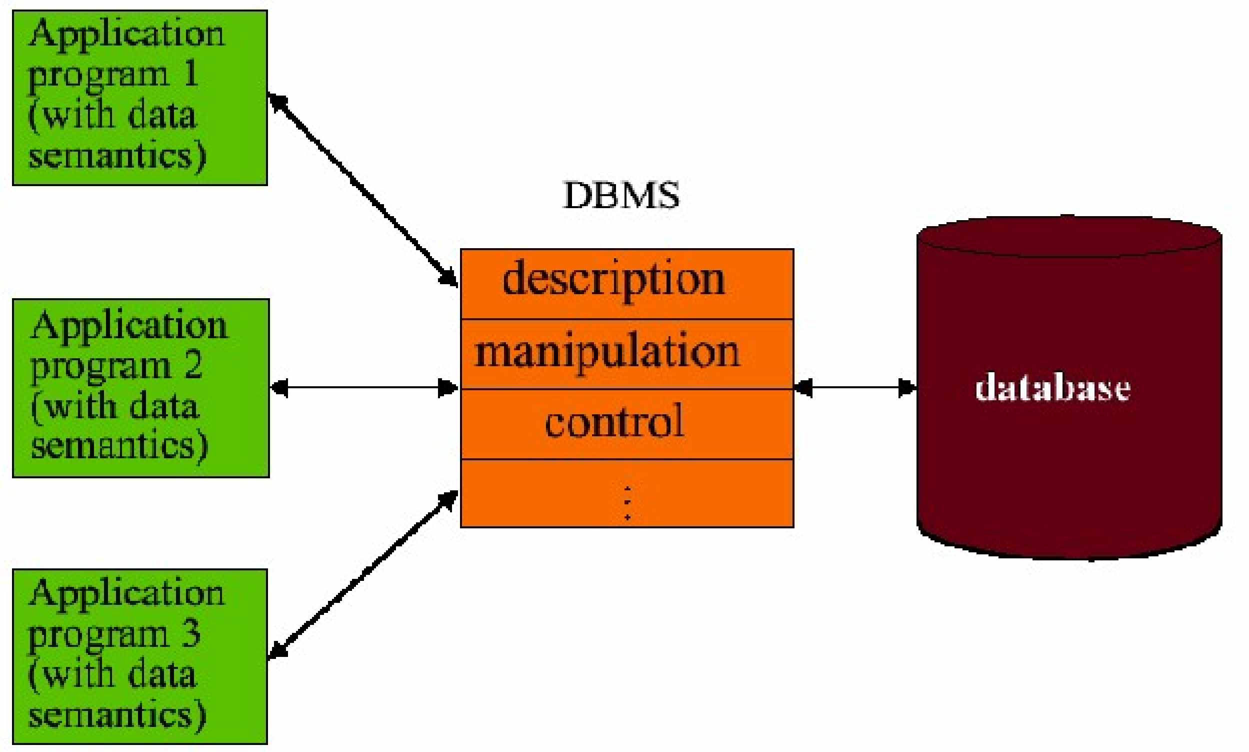
-><u>Distributed Database System</u> (DB+Network)

  - Technology push / Application pull

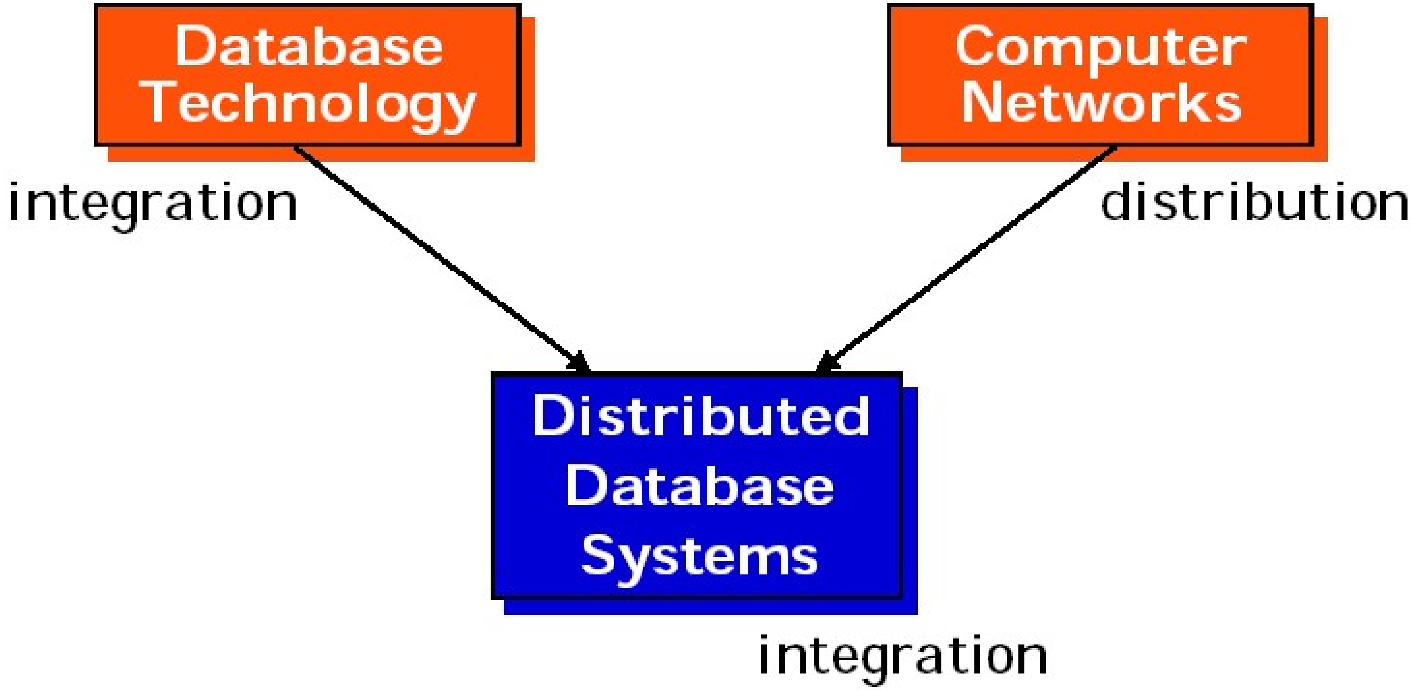
Traditional File Processing



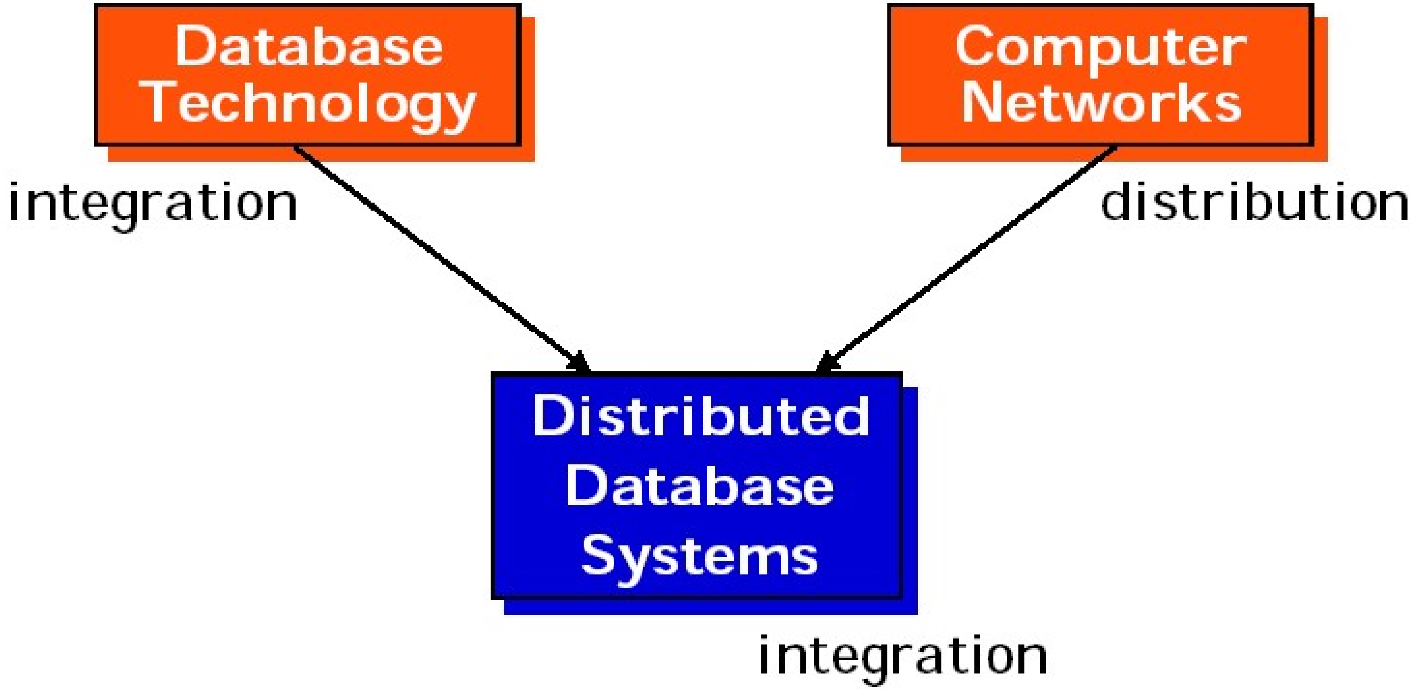
Database Processing



Distributed Database Systems



Integration ≠Centralization



integration ≠ centralization

Outline

- Introduction
- Distributed DBMS Architecture
- Distributed DB Design
- Query Processing
- Transaction Management
- Current Issues

1. Introduction

1.1 Distributed Processing

Distributed Processing (Computing) System

- It is a number of **autonomous** processing elements that are **interconnected** by a computer network and that **cooperate** in performing their assigned tasks.

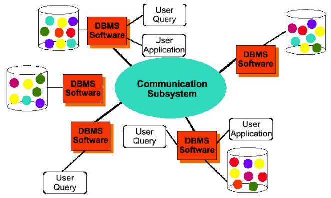
Why do we distribute at all?

- Current organizational structure
- Current applications
- Solve big and complicate problem (divide-and-conquer)

Distributed database systems

should be viewed within this framework and treated as tools that could make distributed processing easier and more efficient.

1.2 What is a Distributed Database System



What is a Distributed Database System?

- Distributed database system (DDBS) =

  DDB + DDBMS

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

Implicit Assumptions

- Data stored at a number of sites
  - each site *logically* consists of a single processor.
- Processors at different sites are interconnected by a computer network
  - no multiprocessors
- Distributed database is a database, not a collection of files
  - data logically related as exhibited in the users' access patterns
- D- DBMS is a full- fledged DBMS
  - not remote file system, not a TP system

1.3 Promises of DDBSs

Distributed DBMS Promises

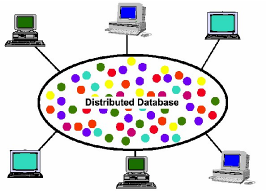
- **Transparent management**

  of distributed, fragmented, and replicated data
- Improved **reliability/ availability** through distributed transactions
- Improved **performance**
- Easier and more economical **system expansion**

Transparencies in a DDBMS

- **Transparency** is the separation of the higher level semantics of a system from the lower level implementation issues.

  In other words, a transparent system "hides" the implementation details from users.

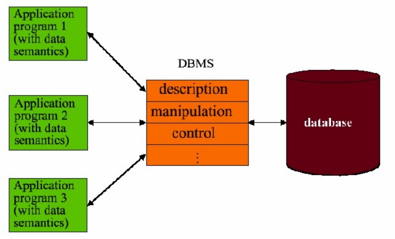


In a DDBMS, it should provide

full transparency

in the distributed environment

Data Independence



- Refers to the immunity of user applications to changes in the definition and organization of data, and vice versa.

- **Logical data independency**
  - The immunity of user applications to changes in the logical structure of the database
- **Physical data independency**
  - Hiding the details of the storage structure from user application

Network Transparency

- Also referred to as **distributed transparency**
  - Desirable to hide even the existence of the network, if possible

- Including:
  - Location transparency
    - The command used is independent of both the location of the data and the system on which an operation is carried out.
  - Naming transparency
    - A unique name is provided for each object in the database

e.g. cp <source file> <target file>

rcp <machine name: source file> < machine name: target file>

Fragmentation Transparency

- Horizontal fragmentation
- Vertical fragmentation
- Hybrid

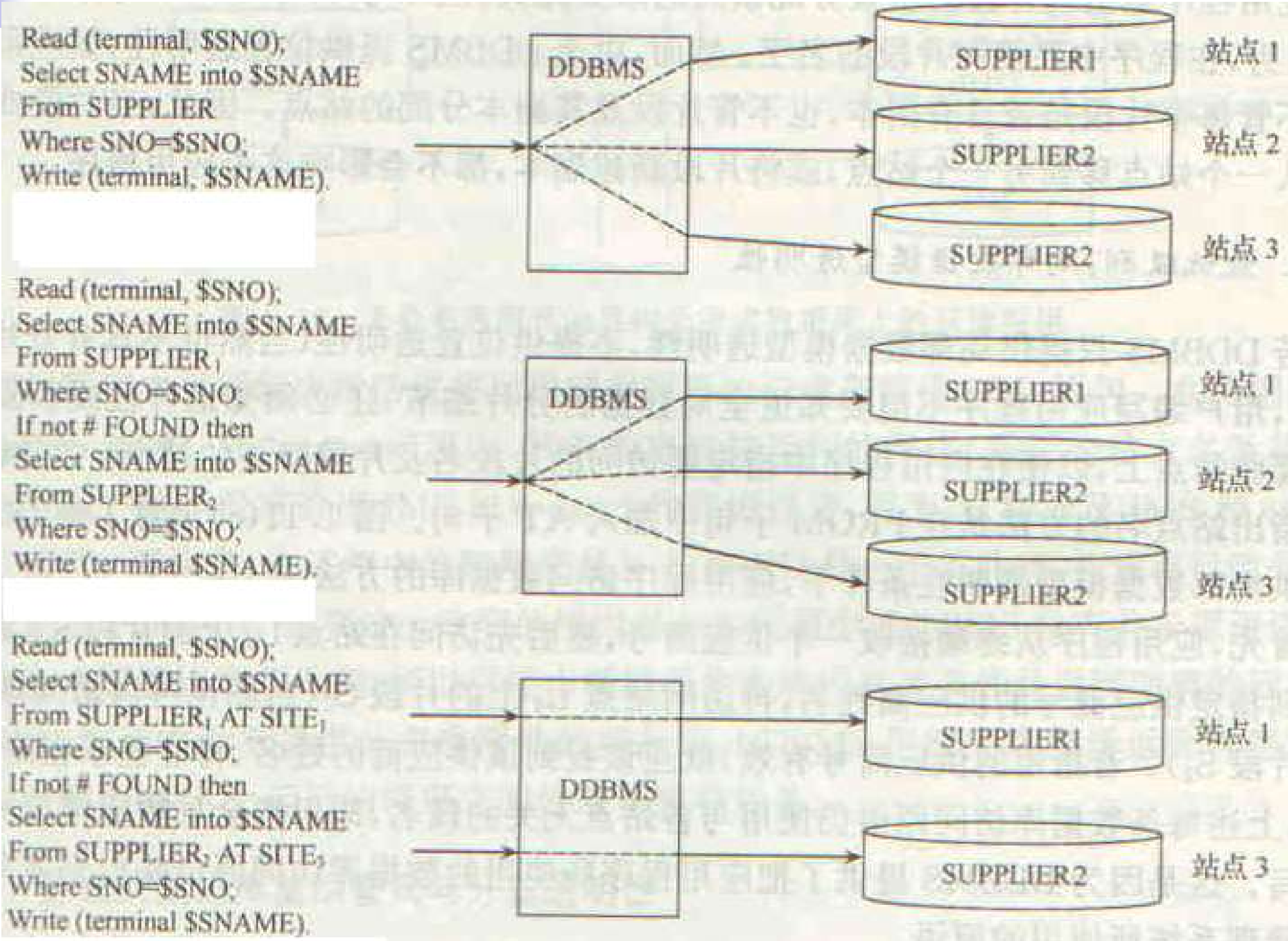
- **Fragmentation transparency** refers to the immunity of user to details of fragmentation
  - Handing user queries that were specified on entire relations but have to be performed on subrelations

Replication Transparency

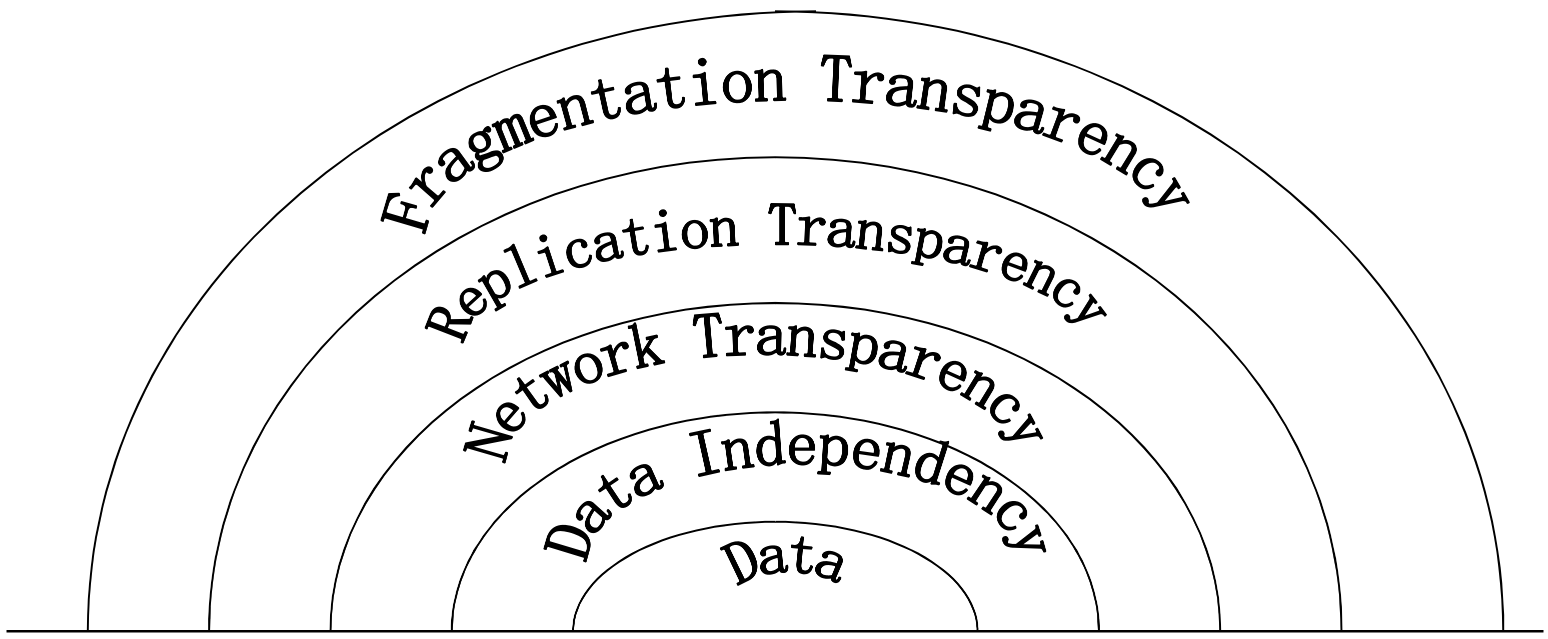
- Why to replicating data?
  - Performance, reliability and availability
- Problem: update

- **Replication transparency** refers to the immunity of user to existence of copies
  - User should act as if there is a single copy
  - Not the location of copies

Example (different layers)



Layers of Transparency



Example

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E7	P5	Engineer	23
E8	P3	Manager	40

PROJ

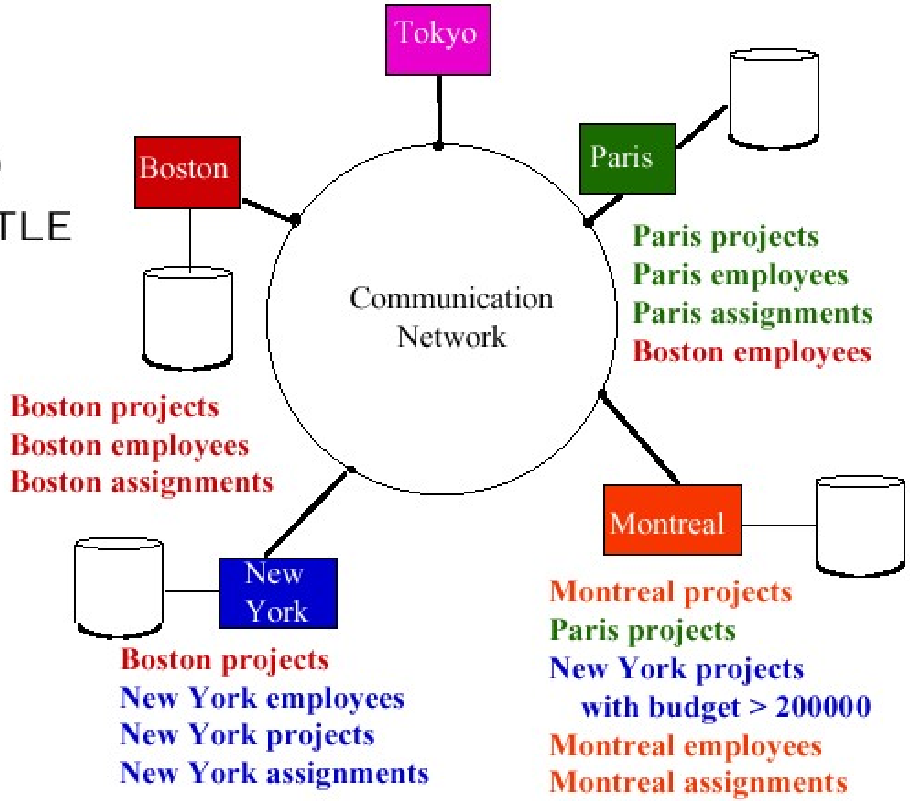
PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

Transparent Access

```
SELECT   ENAME,SAL
FROM     EMP,ASS,PAY
WHERE    DUR > 12
AND      EMP.ENO = ASS.ENO
AND      EMP.TITLE = PAY.TITLE
```



Who should provide transparency?

- Access layer
- Operating System layer
- DBMS layer

About full transparency

- After providing **full transparencies**, there would be **no difference** between database applications that would run on a centralized database and those that would run on a distributed database.
- However, the level of transparency is inevitably a compromise between **ease of use and the difficulty and overhead cost** of providing high levels of transparency

Potentially Improved Performance

- Proximity of data to its points of use
  - Contention reduced
  - Remote access delay reduced
- Parallelism in execution
  - Inter-query parallelism
  - Intra-query parallelism

Applications

- Manufacturing - especially multi-plant manufacturing
- Military command and control
- Corporate MIS
- Airlines
- Hotel chains

Any organization which has a decentralized organization structure

1.4 complicating Factors

Factors

- Data may be replicated in a distributed environment
- Some type of failure
- Synchronization of transactions on multiple sites

1.5 Problem Areas

Problem Areas

- Distributed Database Design
- Distributed Directory Management
- Distributed Query Processing
- Distributed Concurrency Control
- Distributed Deadlock Management
- Reliability of Distributed DBMS
- Operating System Support
- Heterogeneous Databases

Distributed Database Design

- How to distribute the database
- Replicated & non-replicated database distribution
- A related problem in directory management
- The general problem is NP-hard

Query Processing

- Convert user transactions to data manipulation instructions
- Optimization problem
- Min{cost = data transmission + local processing}
- General formulation is NP-hard

Concurrency Control

- Synchronization of concurrent accesses
- Consistency and isolation of transactions' effects
- Deadlock management

Reliability

- How to make the system resilient to failures
- Atomicity and durability

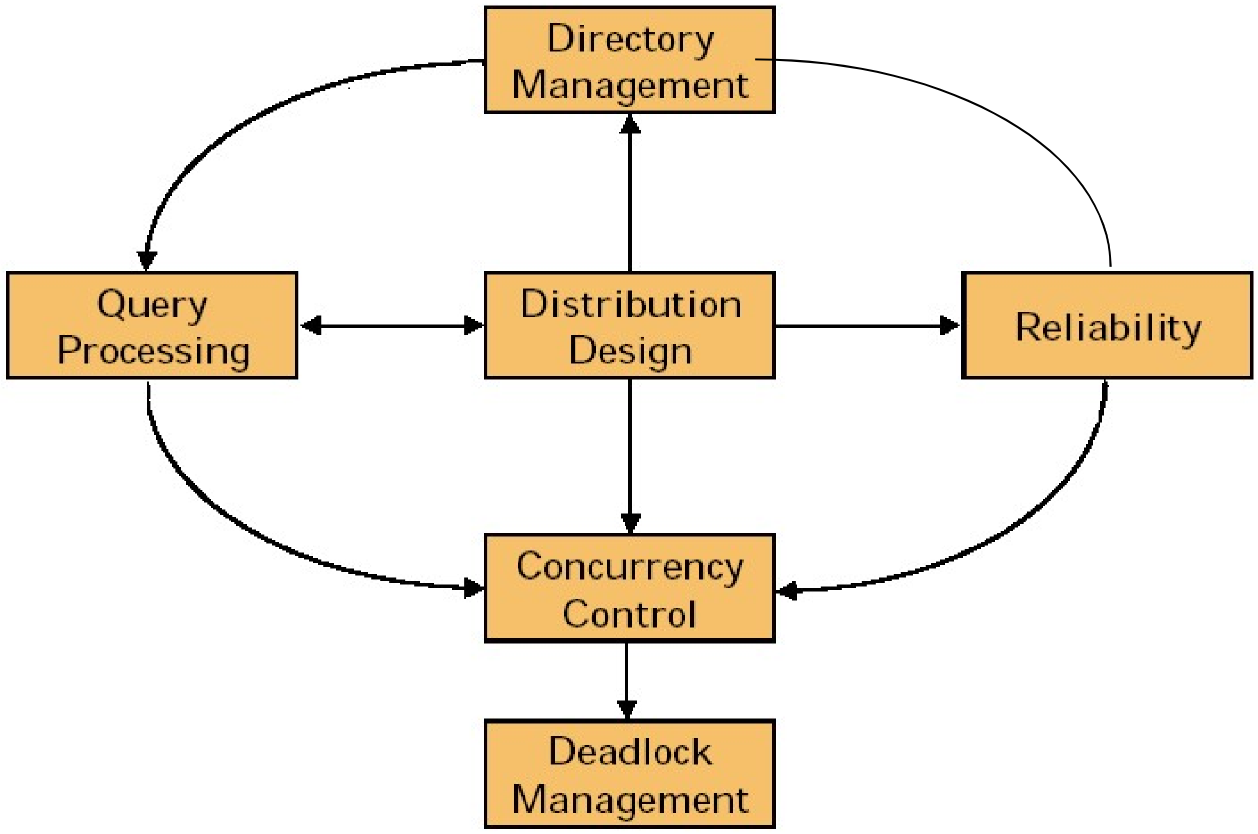
Operating System Support

- Operating system with proper support for database operations
- Dichotomy between general purpose processing requirements and database processing requirements

Open Systems and Interoperability

- Distributed Multidatabase Systems
- More probable scenario
- Parallel issues

Relationship between issues



1.6 Prototypes

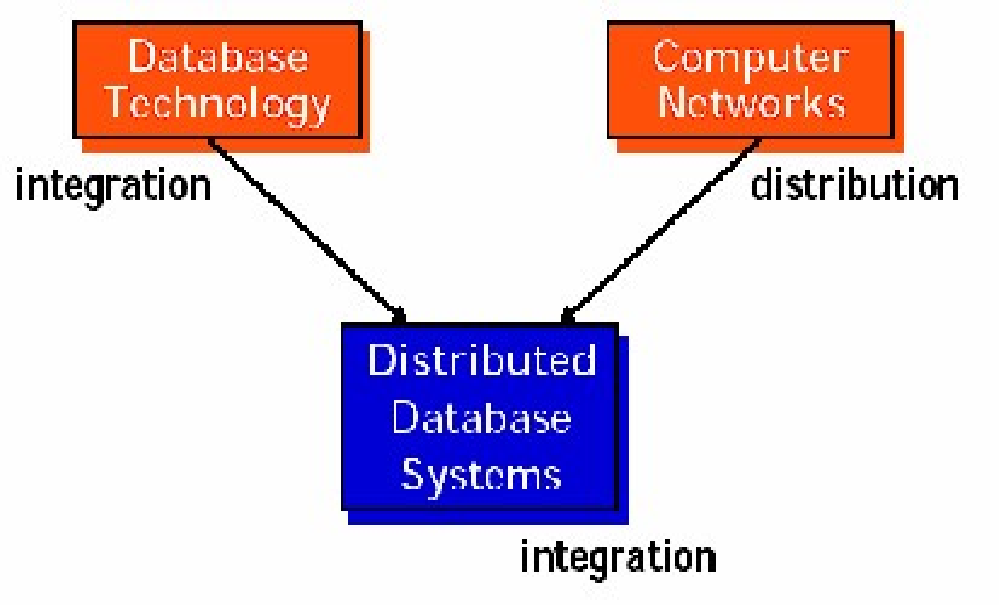
Prototypes

- SDD-1 (CCA)
- Distributed INGRES (UCB)
- R* (IBM)
- POREL
- SIRIUS
- LSZ (NJU)
- WDDBS(WHU)

1.7 Background

Background

- Overview of Rational DBMS
- Overview of Computer Networks



1.7.1 Overview of Relational DBMS

Why to choose relational model

- The mathematical foundation of the relational model
- Problems are easier to formulate
- A large number of relational systems are now on the market
- Most distributed database systems are also relational

Three powerful features

- Its data structure are simple
  - two-dimensional tables
- The relational model provides a solid foundation to data consistency
  - normalization and integrity rules
- The relational model allows the set-oriented manipulation of relations
  - rational algebra and relational calculus

Relational Database Concepts

- **Database**
  - A **structured collection of data** related to some real-life phenomena that we are trying to model
- **Relational Database**
  - One where the database structure is **in the form of tables**
  - Formally, a relation R defined over n sets $D_1, D_2, \ldots, D_n$ (not necessary distinct) is a set of multiple tuples (or simply tuple) $<d_1, d_2, \ldots, d_n>$ such that $d_1 \in D_1, d_2 \in D_2, \ldots, d_n \in D_n$.

Example

- Engineering Company
  - Engineers: ENG
  - Projects: PRO

  ENG(ENO,ENAME,TITLE,SAL,PNO,RESP,DUR)

  PRO(PNO,PNAME,BUDGET,LOC)
    - ENO: employee number
    - ENAME: employee name
    - TITLE: employee title in the company
    - SAL: employee salary
    - PNO: project number
    - RESP: responsibility within the project
    - DUR: duration of the assignment in months
    - PNAME: project name
    - BUDGET: project budget
    - LOC: Project location

Database Scheme

ENG

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR

PRO

PNO	PNAME	BUDGET	LOCATION

- **Key** – the minimum nonempty subset of its attributes such that the values of the attributes comprising the key uniquely identify each tuple of the relation. eg.ENG:(ENO,PNO), PRO:(PNO)
  - Prime attribute
- Prime key / Candidate key
- The number of attributes of a relation defines its **degree**,whereas the number of tuples of the relation defines its **cardinality**

Normalization

- **Normalization** is a step-by-step reversible process of replacing a given collection of relations by successive collections in which relations have a progressively ***simpler and more regular structure***

- Aim
  - To eliminate various *anomalies* (or undesirable aspects) of a relation in order to obtain “better” relations.

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR

Anomalies

- The following four problems might exist in a relation scheme:
  - **Repetition anomaly**
  - **Update anomaly**
  - **Insertion anomaly**
  - **Deletion anomaly**

Normalization process

Decomposition approach

- To starts with a single relation, called the universal relation, which contains all attributes (and probably anomalies) and iteratively reduces it.
- At each iteration, a relation is split into two or more relations of a higher **normal form**:
  - 1NF (Codd)
  - 2NF (Codd)
  - 3NF (Codd)
  - BCNF (Boyce-Codd)
  - 4NF (Fagin)
  - 5NF (Fagin)
- Requirements:
  - Lossless
  - Dependency preservation

Dependency structure

- The normal forms are based on certain dependency structures:
  - 1NF, 2NF, 3NF, BCNF:
    functional dependencies(FD)
  - 4NF: multivalued dependencies(MVD)
  - 5NF: projection-join dependencies(PJD)

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR

The result scheme

- EMP

ENO	ENAME	TITLE

- ASG

ENO	PNO	RESP	DUR

- PAY

TITLE	SAL

- PROJ

PNO	PNAME	BUDGET	LOCATION

Example Database

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E7	P5	Engineer	23
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

Integrity Rules

- Integrity rules are constraints that define consistent states of the database
- Structural constraints
  - Entity integrity

    Each attribute of the key is nonnull
  - Referential integrity

    A group of attributes in one relation is the key of another relation (example)
- Behavioral constraints
  - dependencies

Relational Data Language

- Relational algebra
  - Procedural in that the user is expected to specify, using certain high-level operators, how the result is to be obtained
- Relational calculus
  - Nonprocedural in that the user only specifies the relationship that should hold the result

Relational Algebra 1

- Selection (δ)
- Projection (Π)
- Union (∪)
- Set difference (-)
- Cartesian product (x)

example

Relational Algebra 2

- Intersection ($\cap$)
- θ -Join (∞_F)
- Nature join (∞_A)
- Semijoin ($\propto_F$)
  - The subset of the tuples of R that participate in the join of R with S
- Quotient

example

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E7	P5	Engineer	23
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

$\sigma_{\text{TITLE="Elect. Eng."}}(\text{EMP})$

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E6	L. Chu	Elect. Eng.

Figure 2.5. Result of Selection

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E7	P5	Engineer	23
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

$\Pi_{PNO,BUDGET}(PROJ)$

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000

Figure 2.6. Result of Projection

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E7	P5	Engineer	23
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

EMP x PAY

ENO	ENAME	EMP.TITLE	PAY.TITLE	SAL
E1	J. Doe	Elect. Eng.	Elect. Eng.	40000
E1	J. Doe	Elect. Eng.	Syst. Anal.	34000
E1	J. Doe	Elect. Eng.	Mech. Eng.	27000
E1	J. Doe	Elect. Eng.	Programmer	24000
E2	M. Smith	Syst. Anal.	Elect. Eng.	40000
E2	M. Smith	Syst. Anal.	Syst. Anal.	34000
E2	M. Smith	Syst. Anal.	Mech. Eng.	27000
E2	M. Smith	Syst. Anal.	Programmer	24000
E3	A. Lee	Mech. Eng.	Elect. Eng.	40000
E3	A. Lee	Mech. Eng.	Syst. Anal.	34000
E3	A. Lee	Mech. Eng.	Mech. Eng.	27000
E3	A. Lee	Mech. Eng.	Programmer	24000
E8	J. Jones	Syst. Anal.	Elect. Eng.	40000
E8	J. Jones	Syst. Anal.	Syst. Anal.	34000
E8	J. Jones	Syst. Anal.	Mech. Eng.	27000
E8	J. Jones	Syst. Anal.	Programmer	24000

Figure 2.7. Partial Result of Cartesian Product

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

EMP x PAY

ENO	ENAME	EMP.TITLE	PAY.TITLE	SAL
E1	J. Doe	Elect. Eng.	Elect. Eng.	40000
E1	J. Doe	Elect. Eng.	Syst. Anal.	34000
E1	J. Doe	Elect. Eng.	Mech. Eng.	27000
E1	J. Doe	Elect. Eng.	Programmer	24000
E2	M. Smith	Syst. Anal.	Elect. Eng.	40000
E2	M. Smith	Syst. Anal.	Syst. Anal.	34000
E2	M. Smith	Syst. Anal.	Mech. Eng.	27000
E2	M. Smith	Syst. Anal.	Programmer	24000
E3	A. Lee	Mech. Eng.	Elect. Eng.	40000
E3	A. Lee	Mech. Eng.	Syst. Anal.	34000
E3	A. Lee	Mech. Eng.	Mech. Eng.	27000
E3	A. Lee	Mech. Eng.	Programmer	24000
E8	J. Jones	Syst. Anal.	Elect. Eng.	40000
E8	J. Jones	Syst. Anal.	Syst. Anal.	34000
E8	J. Jones	Syst. Anal.	Mech. Eng.	27000
E8	J. Jones	Syst. Anal.	Programmer	24000

Figure 2.7. Partial Result of Cartesian Product

$EMP \bowtie_{EMP.TITLE=PAY.TITLE} PAY$

ENO	ENAME	TITLE	SAL
E1	J. Doe	Elect. Eng.	40000
E2	M. Smith	Analyst	34000
E3	A. Lee	Mech. Eng.	27000
E4	J. Miller	Programmer	24000
E5	B. Casey	Syst. Anal.	34000
E6	L. Chu	Elect. Eng.	40000
E7	R. Davis	Mech. Eng.	27000
E8	J. Jones	Syst. Anal.	34000

Figure 2.8. The Result of Join

$EMP \ltimes_{EMP.TITLE=PAY.TITLE} PAY$

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Analyst
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

Figure 2.9. The Result of Semijoin

Relational Calculus

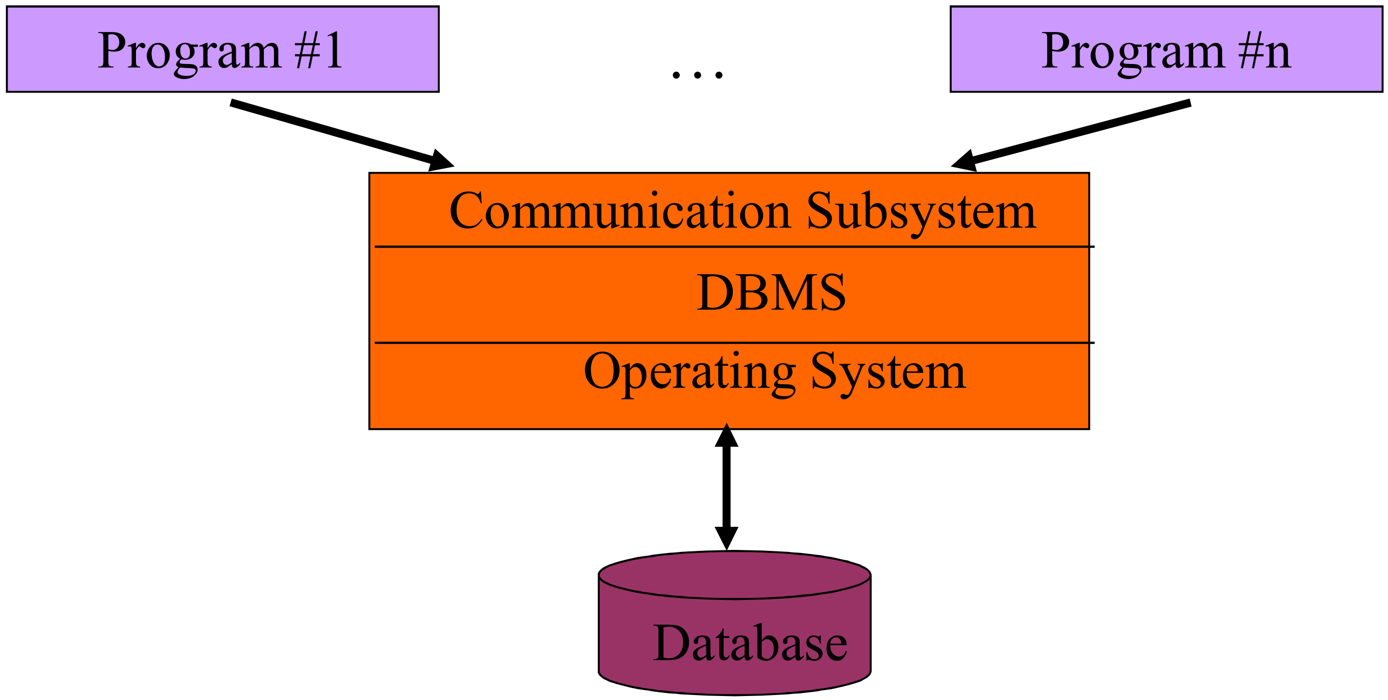
- Tuple relational calculus
  - SQL (Select-From-Where)
- Domain relational calculus
  - QBE

Interface with programming language

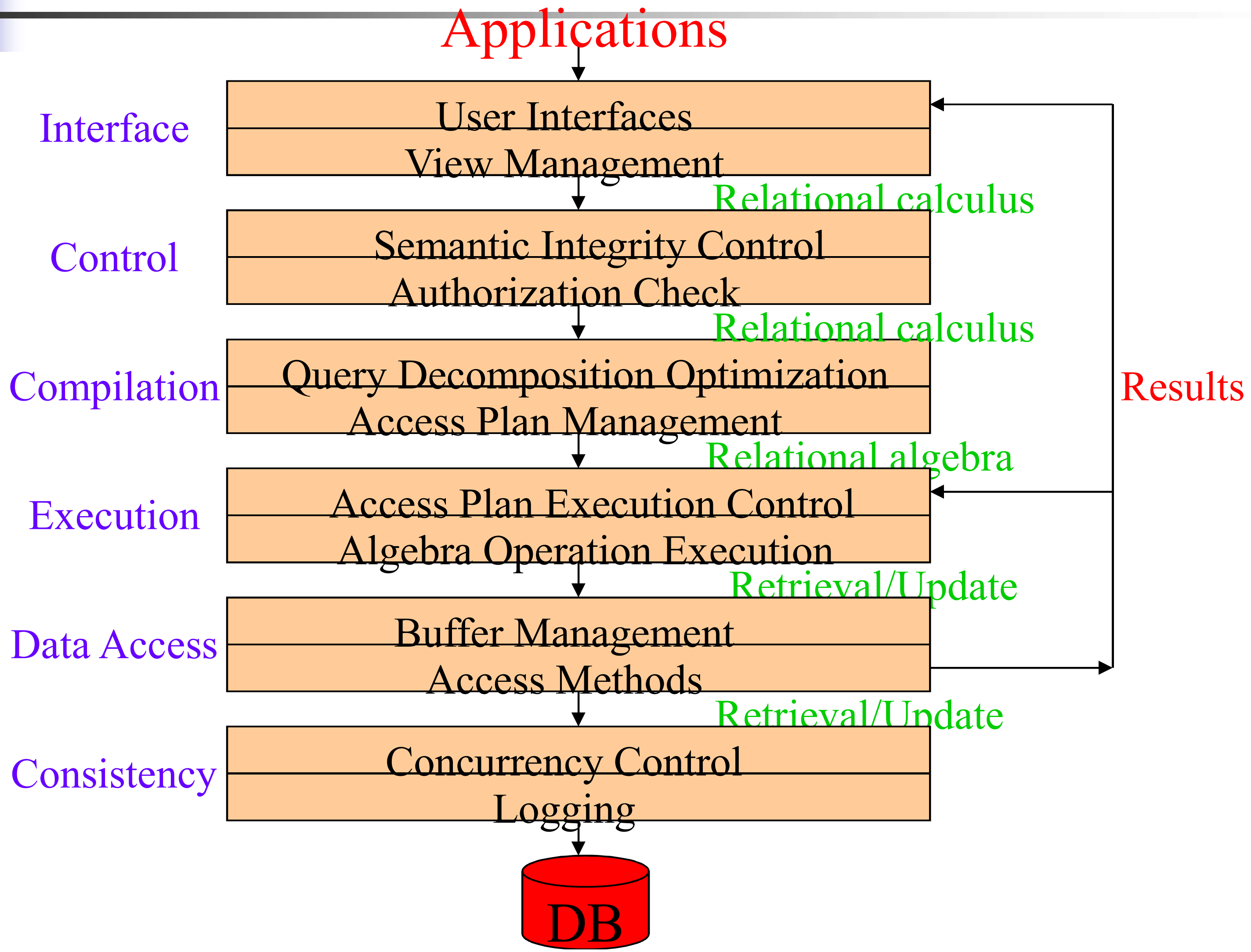
- Tightly coupled approach
  - The programming language and the database language are merged in a single language, eg. Pascal/R
- Loosely coupled approach
  - The programming language is a high-level language which does not know about the database concepts, and it is just simply extended with special commands that call the database system.

Relational DBMS

- Generic architecture of a centralized DBMS



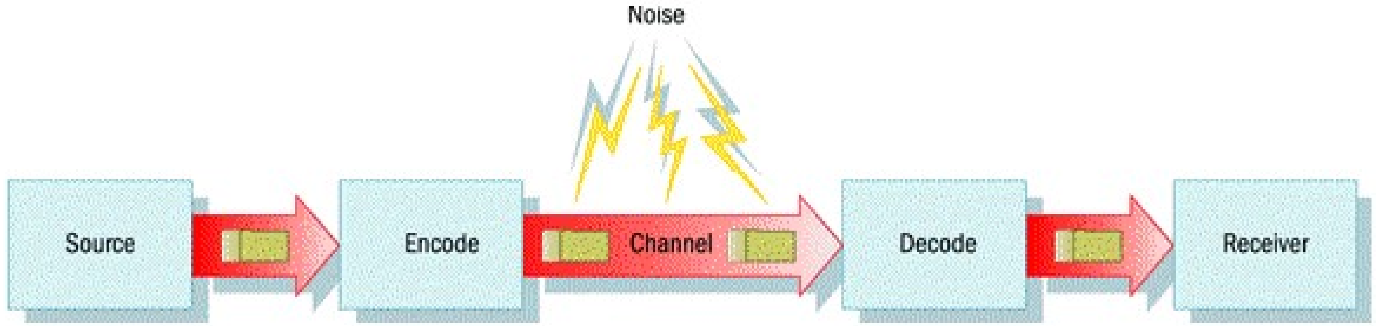
Functional layers of a relational DBMS



1.7.2 Overview of Computer Network

Shannon's Communication System Model

5. Noise, such as electrical interference, sometimes disrupts a transmission. The message can become garbled unless the communications system has the capability to check for errors and correct them.



1. The **source** originates or initiates the communication. The source might be a person or a computer.

2. The **message**, represented here by a folder, is the information that the source wants to communicate to the receiver. The message might be a document, picture, sound, or numeric data.

3. The message is **encoded** by changing its format into one that can be transmitted. In a modem communications system, a message typically is encoded into an electronic signal that can be sent over telephone lines, broadcast by radio waves, or transmitted as microwaves.

4. The encoded message travels by means of a **channel** or **communications link**. A communications link might include telephone wiring, fiber-optic cable, microwaves, or satellites.

6. The message is decoded at the end of the transmission. **Decoding** usually means reversing the coding process that occurred before the message was sent.

7. The **receiver** is the destination for the message. The receiver can be a person, a computer, or another communications device.

Types of Networks

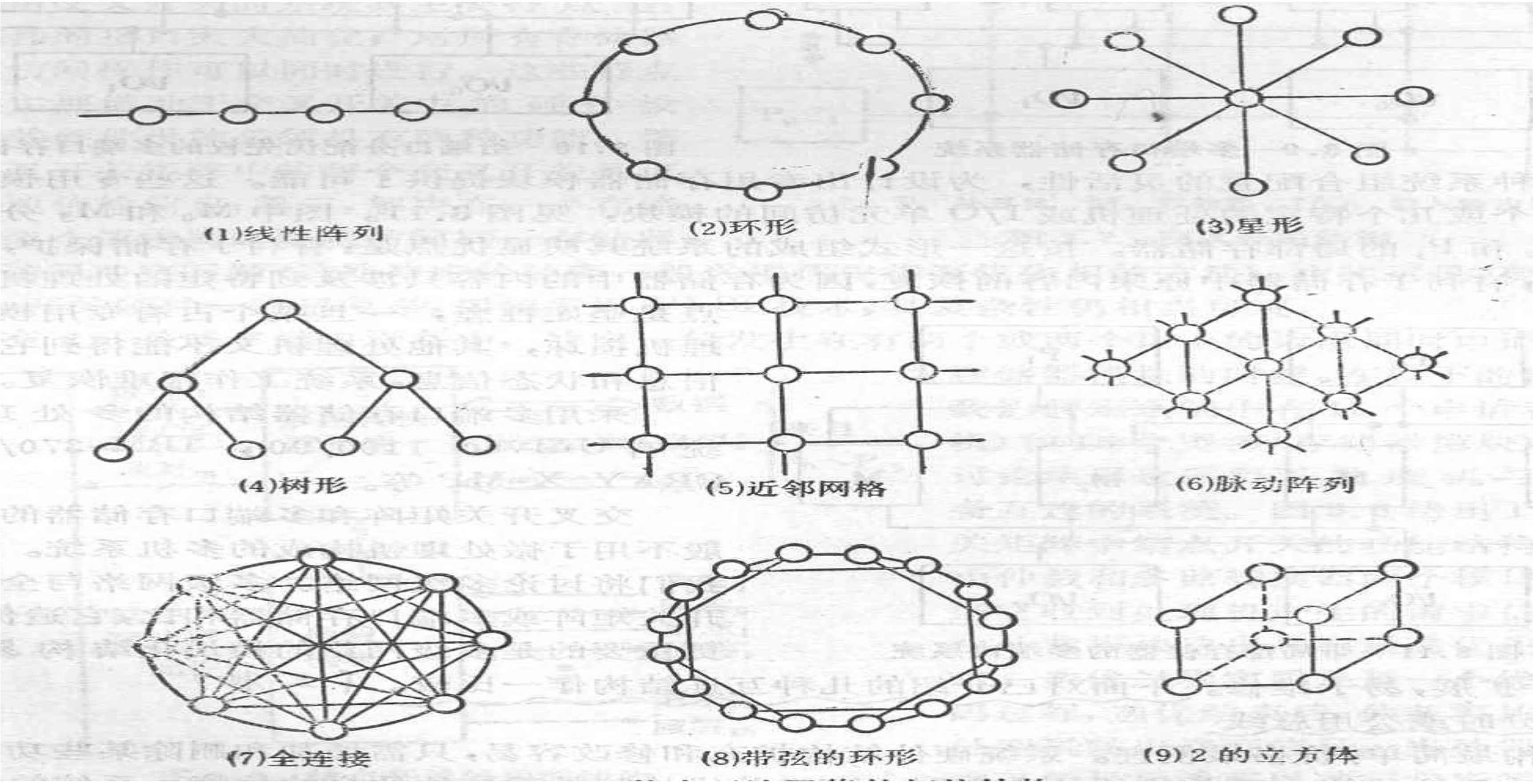
Criteria:

- Interconnection structure (Topology)
- Mode of transmission
- Geographic distribution (Scale)

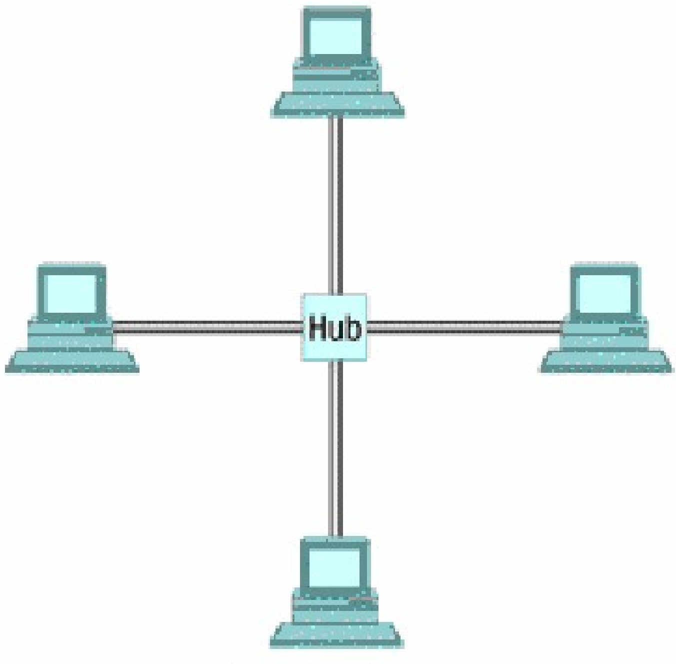
Topology

- Star
- Ring
- Bus
- Completely Connected
- Irregular

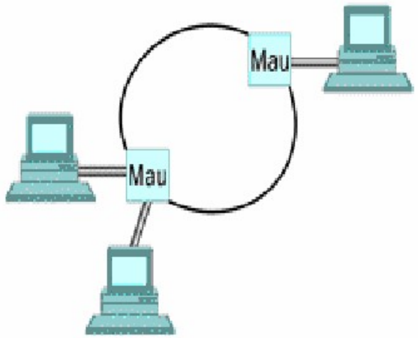
Network Topologies



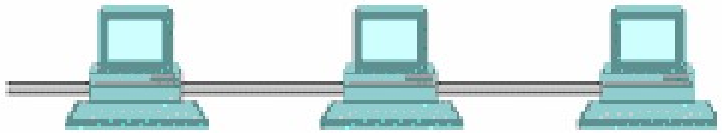
Network Topologies



Star topology



Ring topology



Bus topology

Communication schemes

- broadcast networks
  - broadcasting, multicasting
- Point-to-Point network
  - routing algorithm

Protocol Standards

- ISO/OSI RM
- TCP/IP
- IEEE 802

ISO/OSI RM

Application
Presentation
Session
Transport
Network
Data link
Physical

ISO/OSI - TCP/IP

OSI	TCP/IP
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data Link	Host-to-network
Physical	

IEEE 802

1. **802.3**

   CSMA/CD Bus network(Ethernet)

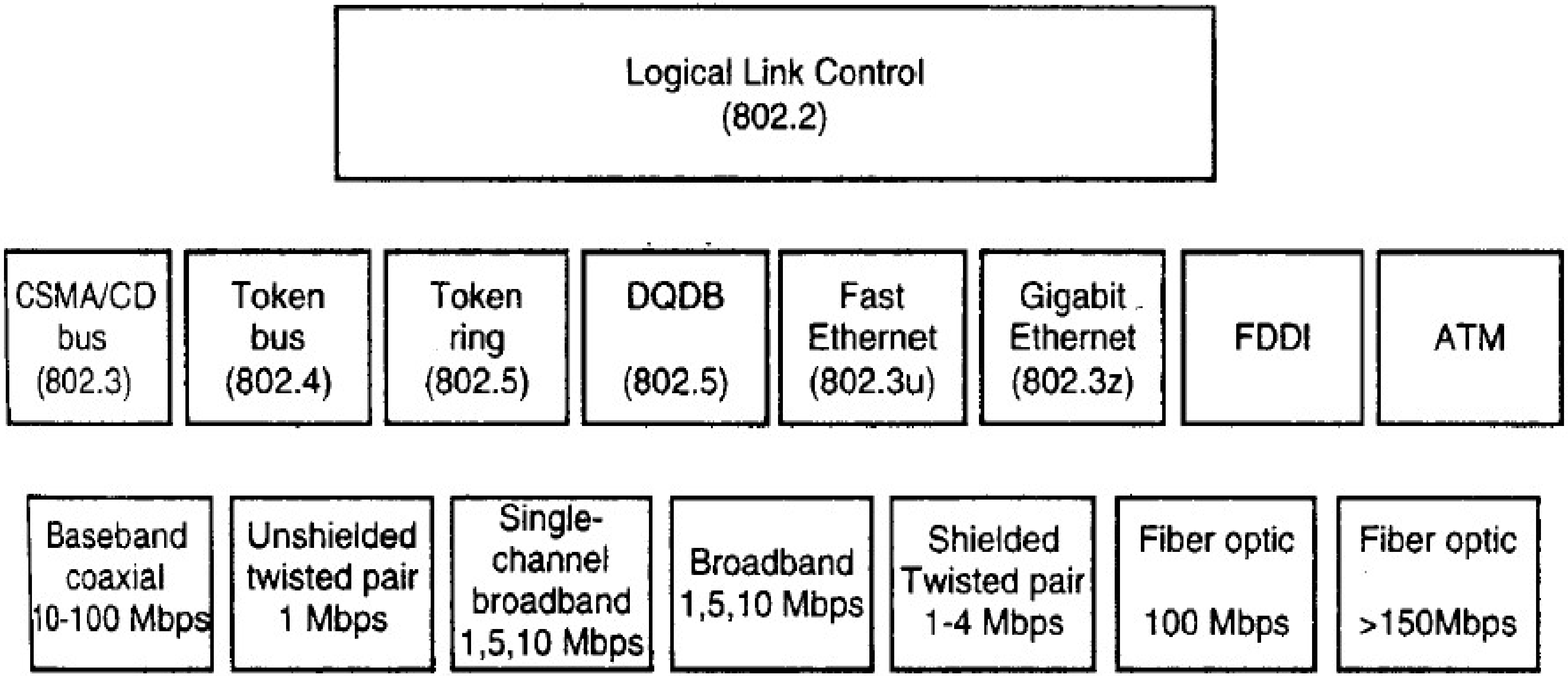
2. **802.4**

   Token Bus

3. **802.5**

   Token Ring

IEEE 802 Standard



Network Performance Metrics

Latency *and* ***bandwidth*** *are two fundamental metrics used to assess the performance*

- **Bandwidth**
  - the range of frequencies that a channel can carry.
    - The bandwidth of an **analog signal** is expressed in Hertz **(Hz);**
    - While the bandwidth of a **digital signal** is usually measured in bits per second **(bps).**
- **Communication latency**
  - the total time needed to transmit a message from a source node to a destination node; consists of
    - Software overhead
    - Channel delay
    - Routing delay
    - Contention delay

Effect of network

- Almost all the performance studies that we know assume a very simple network cost model, such as a fixed communication delay
- Although some trade-offs have been taken into account in design, such as the network speed vs. I/O speed in traditional WAN, they are mostly only in qualitative terms
- The effect of networks has to be more realistically taken into account in the design of DDBMSs, including significant details on the network infrastructure

Background

- Overview of Rational DBMS
  - Semi-Join
  - Referential Integrity
- Overview of Computer Networks

References

- M.T. Özsu and P. Valduriez. *Principles of Distributed Database Systems*. Prentice-Hall,1999.
- M.T. Özsu and P. Valduriez. "Distributed and Parallel Database Systems", In *Handbook of Computer Science and Engineering*, A. Tucker (ed.), CRC Press, 1997, pages 1093–1111 (Chapter 48).
- M.T. Özsu and P. Valduriez. "Distributed Database Systems: Where Are We Now?" *Computer*, August 1991, 24(8).
- M. T. Özsu. "The Push/Pull Effect - Can Distributed Database Technology Meet The Challenges of New Applications?" *Database Programming & Design*, April 1997.
- S. Ceri and G. Pelagatti. *Distributed Databases – Principles and Systems*. McGraw Hill, 1984.
- D.A. Bell and J.B. Grimson. *Distributed Database Systems*. Addison-Wesley, 1992.
- G. Gardarin and P. Valduriez, Relational Databases and Knowledge Bases. Addison-Wesley, 1989
- A. S. Tanenbaum. Computer Networks (3rd edition). Englewood Cliffs, N.J., Prentice-Hall, 1997