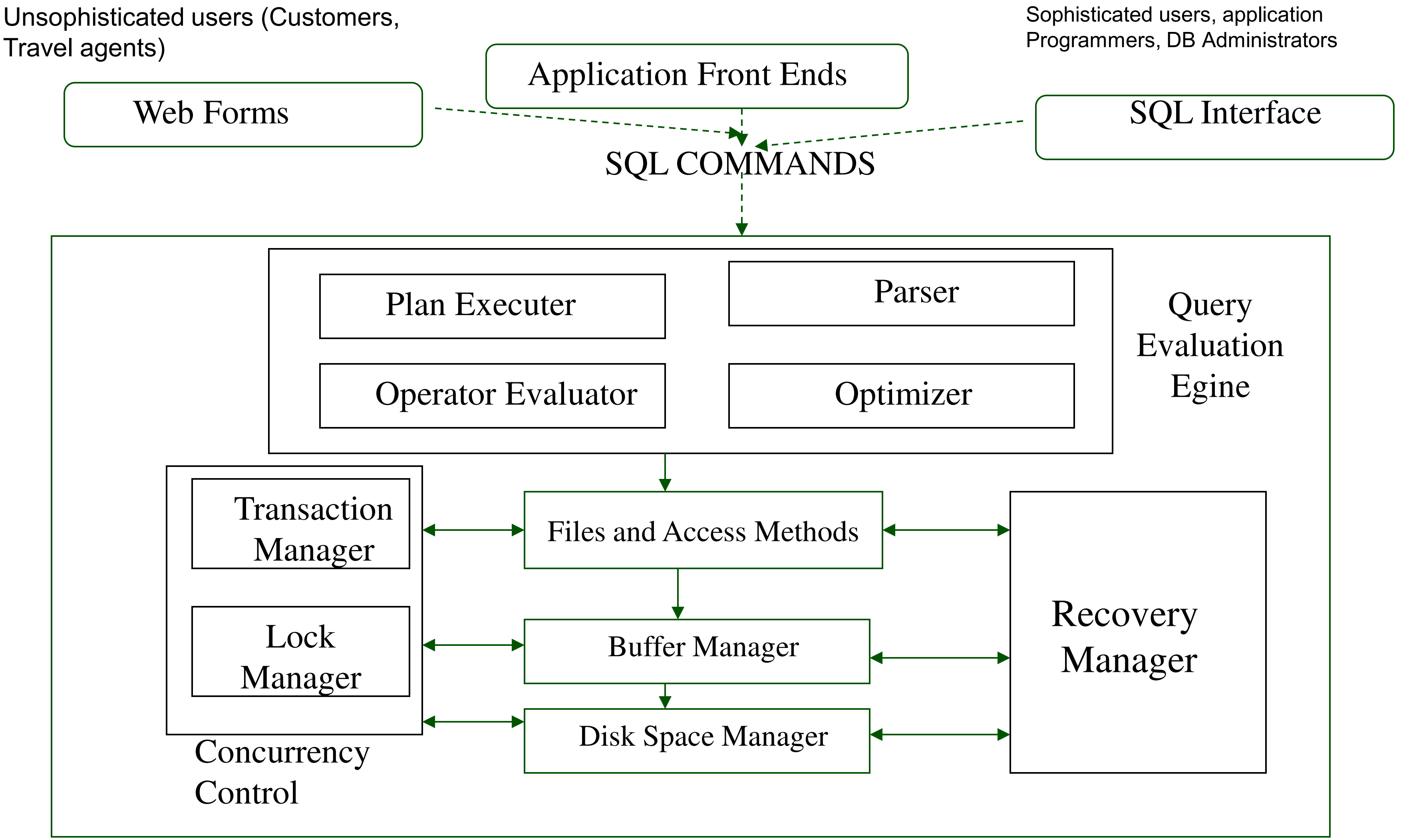
Additional Topics

Refer

1. Chapter 16 of the text book
2. Raghu Ramakrishnan’s book

Architecture Of DBMS



Unsophisticated users (Customers, Travel agents)

Sophisticated users, application Programmers, DB Administrators

Web Forms

Application Front Ends

SQL Interface

SQL COMMANDS

Plan Executer

Parser

Operator Evaluator

Optimizer

Query Evaluation Egine

Transaction Manager

Lock Manager

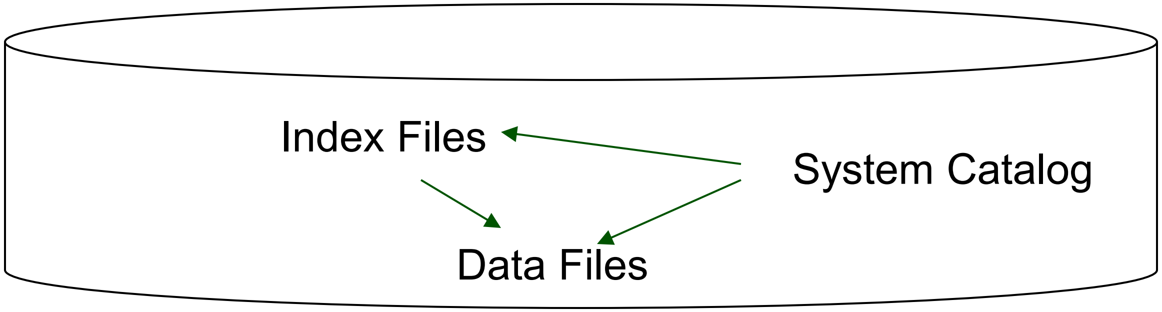
Concurrency Control

Files and Access Methods

Buffer Manager

Disk Space Manager

Recovery Manager



Index Files

System Catalog

Data Files

Outline

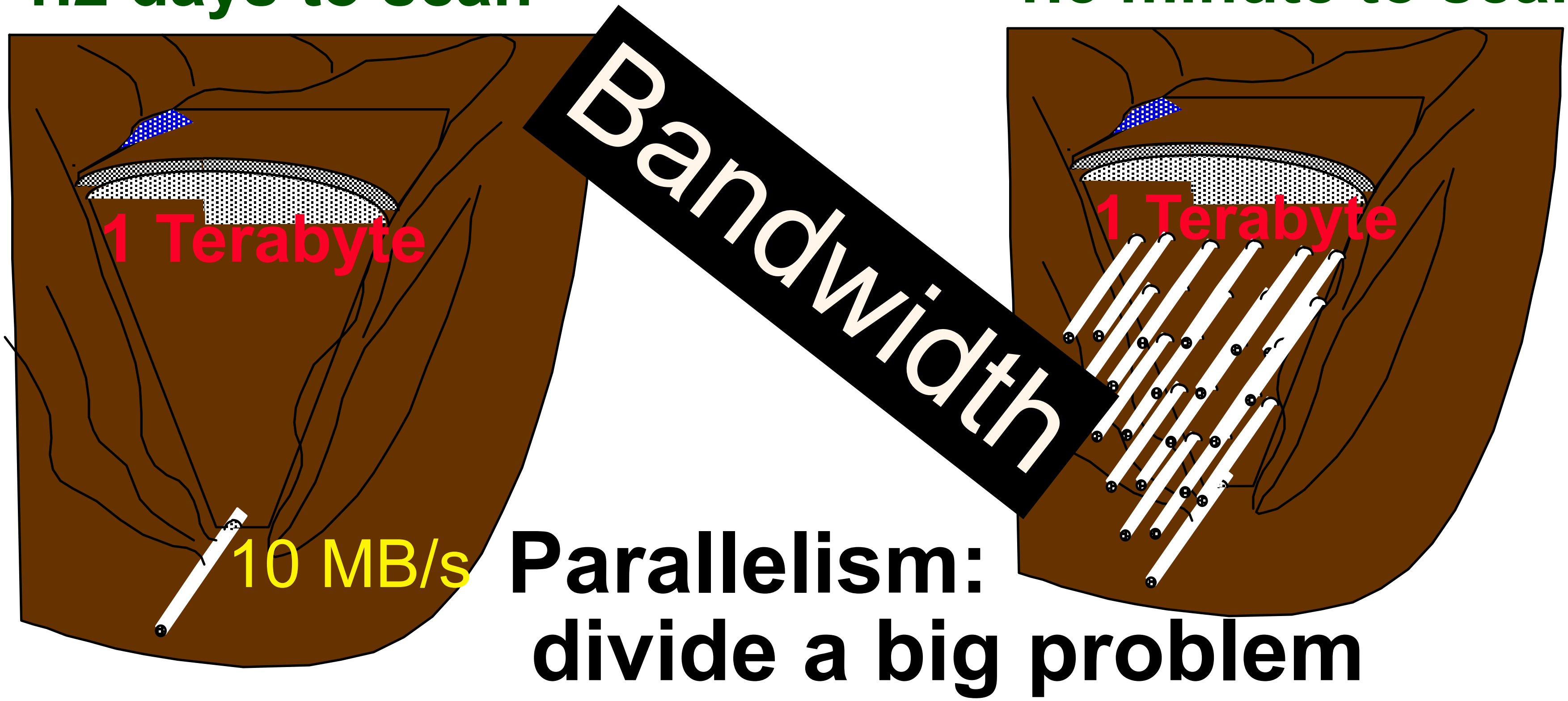
- **Parallel Databases**
- Distributed Databases
- Object-database systems
- Deductive database systems
- Data warehousing and decision support
- Data mining
- Information Retrieval and XML data
- Spatial data management
- Other topics:
    - Advanced transaction processing,
    - Mobile databases
    - Main Memory databases
    - Multimedia databases
    - GIS
    - Temporal databases
    - Biological databases
    - Information visualization

Parallel databases

Why Parallel Access To Data?

At 10 MB/s
1.2 days to scan

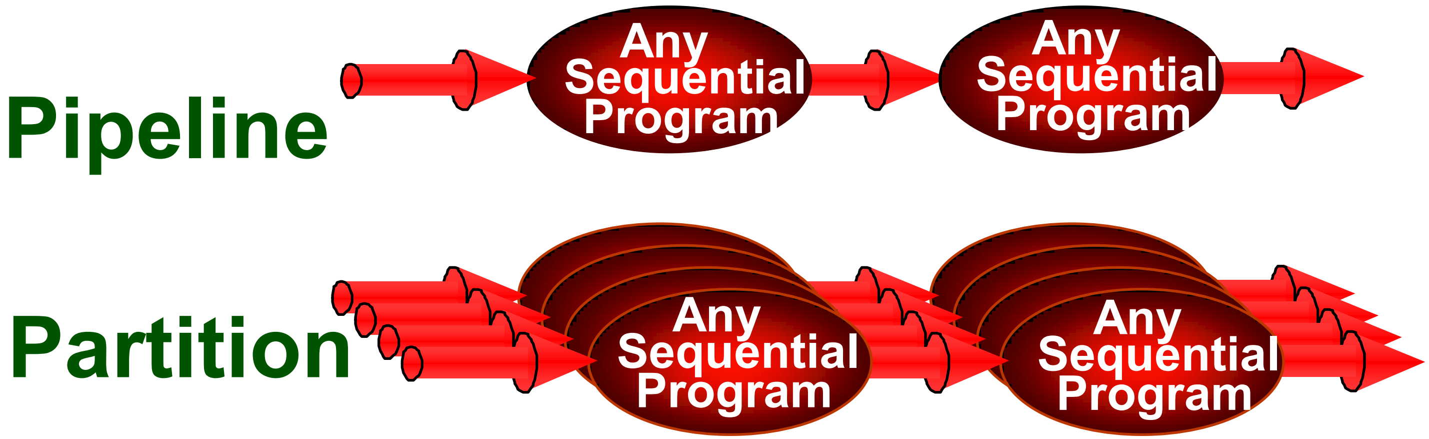
1,000 x parallel
1.5 minute to scan.



Parallelism:
divide a big problem
into many smaller ones
to be solved in parallel.

Parallel DBMS: Intro

- Parallelism is natural to DBMS processing
  - *Pipeline parallelism:* many machines each doing one step in a multi-step process.
  - *Partition parallelism:* many machines doing the same thing to different pieces of data.
  - **Both are natural in DBMS!**



Pipeline

Any Sequential Program

Any Sequential Program

Partition

Any Sequential Program

Any Sequential Program

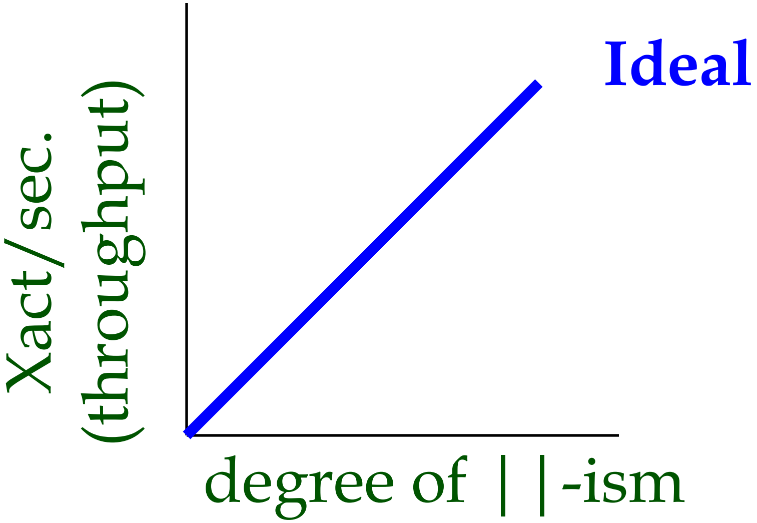
outputs split N ways, inputs merge M ways

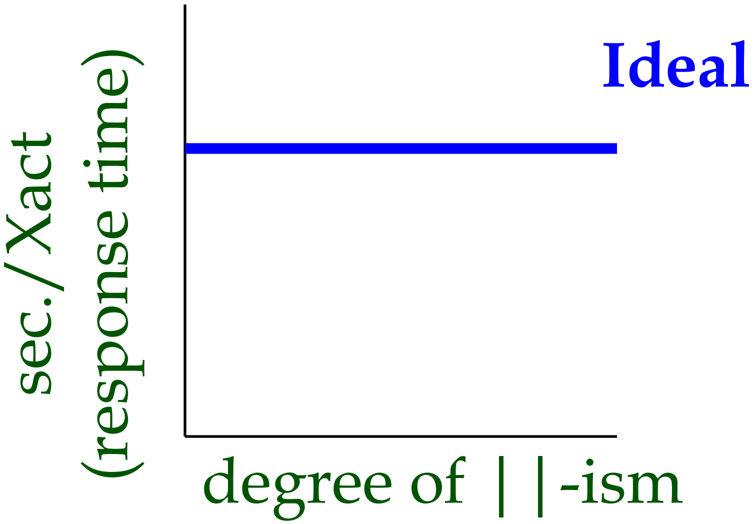
DBMS: **The** || Success Story

- DBMSs are the most (only?) successful application of parallelism.
  - Teradata, Tandem vs. Thinking Machines, KSR..
  - Every major DBMS vendor has some || server
  - Workstation manufacturers now depend on || DB server sales.
- Reasons for success:
  - Bulk-processing (= partition ||-ism).
  - Natural pipelining.
  - Inexpensive hardware can do the trick!
  - Users/app-programmers don’t need to think in ||

Some || Terminology

- Speed-Up
  - More resources means proportionally less time for given amount of data.
- Scale-Up
  - If resources increased in proportion to increase in data size, time is constant.



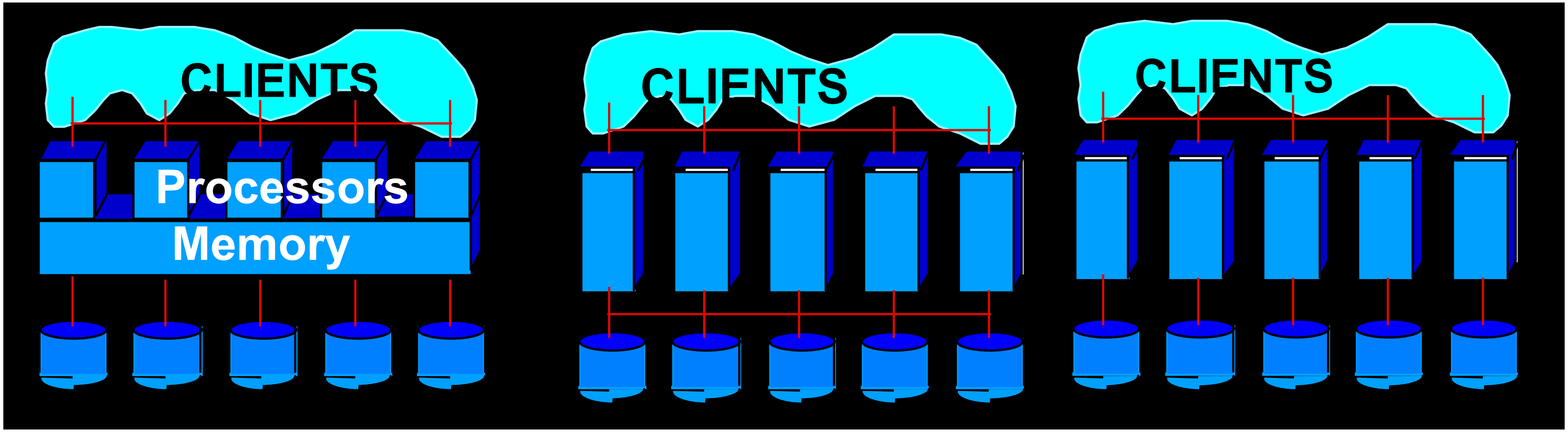


Architecture Issue: Shared What?

Shared Memory (SMP)

Shared Disk

Shared Nothing (network)



Easy to program
Expensive to build
Difficult to scaleup

Hard to program
Cheap to build
Easy to scaleup

Sequent, SGI, Sun

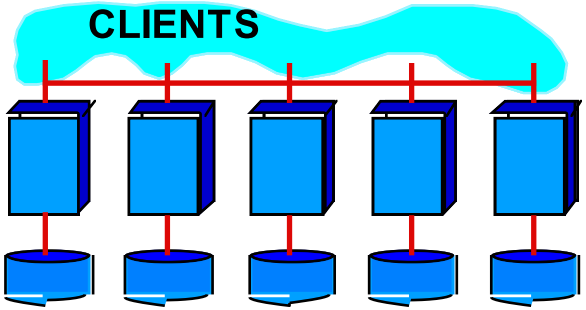
VMScluster, Sysplex

Tandem, Teradata, SP2

What Systems Work This Way
(as of 9/1995)

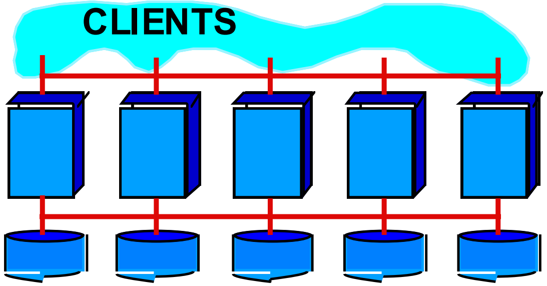
Shared Nothing

Teradata:	400 nodes
Tandem:	110 nodes
IBM / SP2 / DB2:	128 nodes
Informix/SP2	48 nodes
ATT & Sybase	? nodes



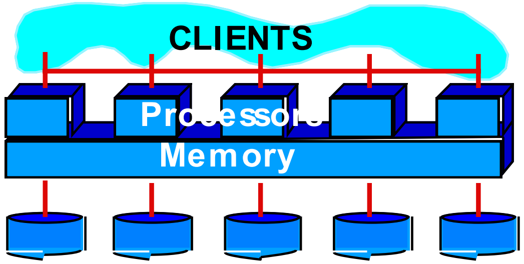
Shared Disk

Oracle	170 nodes
DEC Rdb	24 nodes



Shared Memory

Informix	9 nodes
RedBrick	? nodes



Different Types of DBMS ||-ism

- Intra-operator parallelism
  - get all machines working to compute a given operation (scan, sort, join)
- Inter-operator parallelism
  - each operator may run concurrently on a different site (exploits pipelining)
- Inter-query parallelism
  - different queries run on different sites

Parallelizing Individual Operations

- Bulk loading
- Parallel scanning
- Parallel Sorting
- Joins:
  - Parallel Nested loop join
  - Parallel Sort merge join
  - Parallel Hash join
- Parallel query optimization
- Transaction logging and concurrency control is additional issue

Outline

- Parallel Databases
- **Distributed Databases**
- Object-database systems
- Deductive database systems
- Data warehousing and decision support
- Data mining
- Information Retrieval and XML data
- Spatial data management
- Other topics:
    - Advanced transaction processing,
    - Mobile databases
    - Main Memory databases
    - Multimedia databases
    - GIS
    - Temporal databases
    - Biological databases
    - Information visualization

Introduction to distributed database system

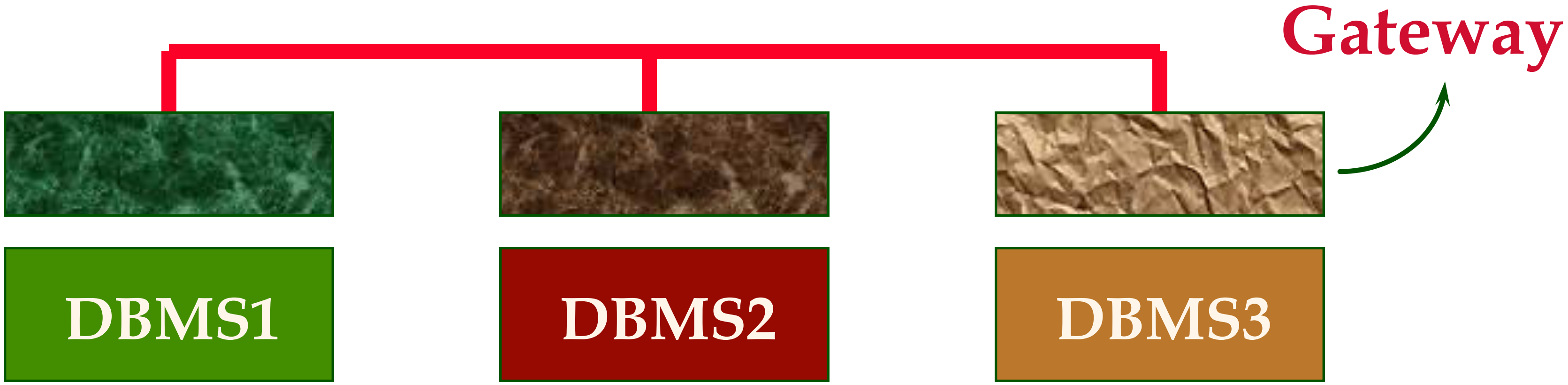
- Data is stored at several sites, each managed by a DBMS that can run independently.
- Distributed Data Independence: Users should not have to know where data is located (extends Physical and Logical Data Independence principles).
- Distributed Transaction Atomicity: Users should be able to write Xacts accessing multiple sites just like local Xacts.

Recent Trends

- Users have to be aware of where data is located, i.e., Distributed Data Independence and Distributed Transaction Atomicity are not supported.
- These properties are hard to support efficiently.
- For globally distributed sites, these properties may not even be desirable due to administrative overheads of making location of data transparent.

Types of Distributed Databases

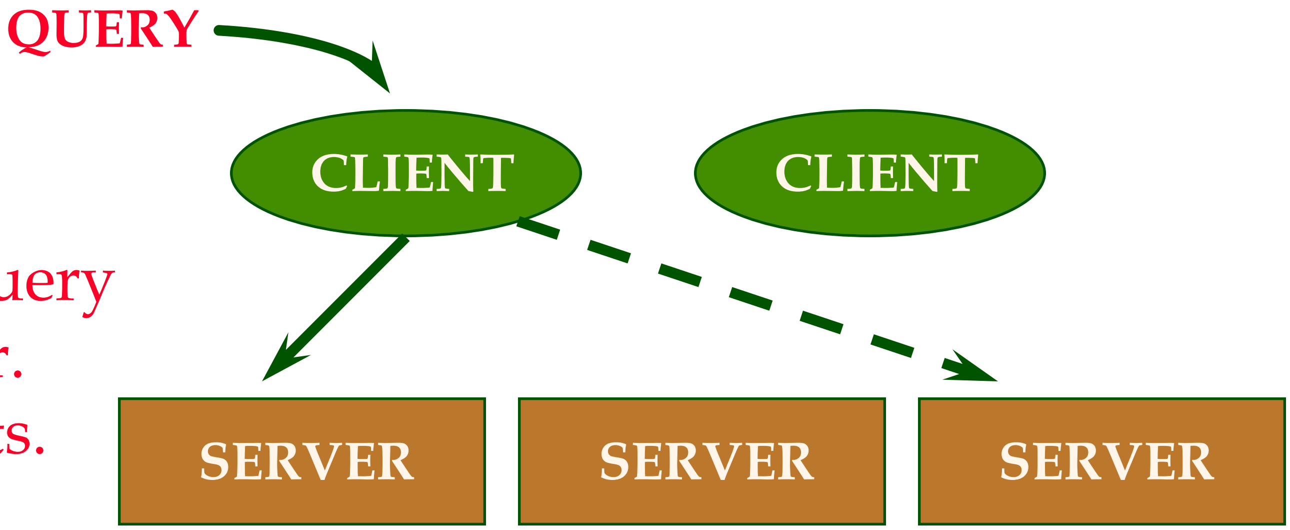
- Homogeneous: Every site runs same type of DBMS.
- Heterogeneous: Different sites run different DBMSs (different RDBMSs or even non-relational DBMSs).



Distributed DBMS Architectures

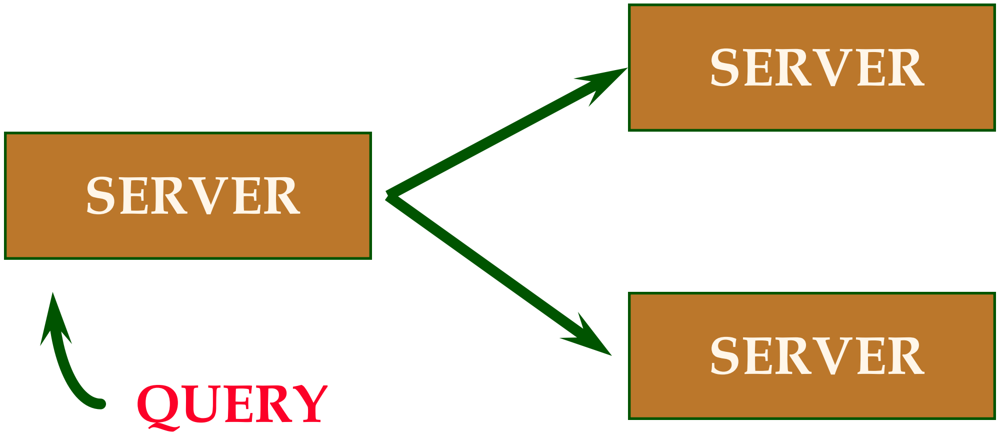
- Client-Server

  Client ships query to single site. All query processing at server.
  - *Thin* vs. *fat* clients.
  - Set-oriented communication, client side caching.



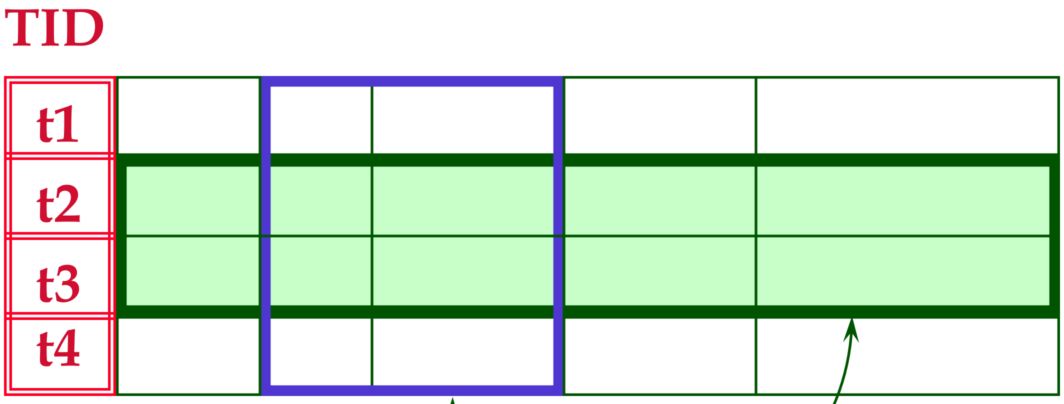
- Collaborating-Server

  Query can span multiple sites.

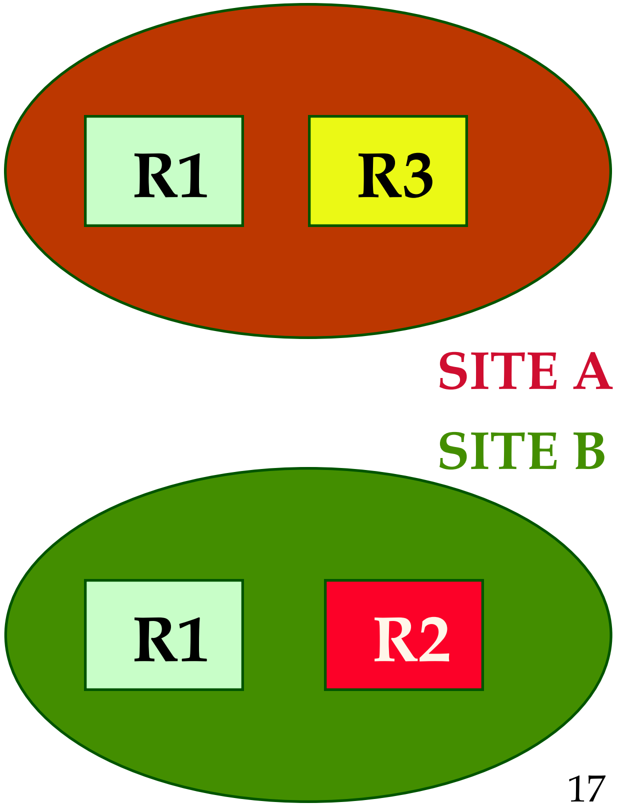


- Middleware

Storing Data



- Fragmentation
  - Horizontal: Usually disjoint.
  - Vertical: Lossless-join; tids.
- Replication
  - Gives increased availability.
  - Faster query evaluation.
  - Synchronous vs. Asynchronous.
    - Vary in how current copies are.



DDBMS topics

- Distributed catalog management
- Distributed query processing
  - Semi-joins, Bloomjoins
- Distributed query optimization
  - Cost-based optimization (Include communication cost)
- Updating/transaction processing on distributed data
  - Synchronous replication: all copies are updated together
  - Asynchronous replication: only primary copy is updated secondary copies are updated later.
- Distributed concurrency control
  - Centralized
  - Primary copy
  - Fully distribued
  - Distributed deadlock
- Distributed recovery
  - Two-phase commit/three phase commit

Outline

- Parallel Databases
- Distributed Databases
- **Object-database systems**
- Deductive database systems
- Data warehousing and decision support
- Data mining
- Information Retrieval and XML data
- Spatial data management
- Other topics:
  - Advanced transaction processing,
  - Mobile databases
  - Main Memory databases
  - Multimedia databases
  - GIS
  - Temporal databases
  - Biological databases
  - Information visualization

Object Database Systems

- Relational database model (integers, dates and string data types) is inadequate for several applications
  - CAD/CAM, multimedia, document management
- So, DBMS should support for complex data types.
- Object-database systems have developed along two district lines.
  - Object-oriented database systems
    - Object Database Management Group has developed a standard data model (ODM) and object query language (OQL)
  - Object-relational database systems
    - Extend relational database systems to support complex data types
    - Provide as a bridge between relational and object oriented paradigms.
    - SQL:1999 standard incorporates support for object-relational model of data.
    - Storing and retrieving of BLOOBs

New Data Types

- User-defined data types
- Inheritance
- Object Identity

Outline

- Parallel Databases
- Distributed Databases
- Object-database systems
- **Deductive database systems**
- Data warehousing and decision support
- Data mining
- Information Retrieval and XML data
- Spatial data management
- Other topics:
  - Advanced transaction processing,
  - Mobile databases
  - Main Memory databases
  - Multimedia databases
  - GIS
  - Temporal databases
  - Biological databases
  - Information visualization

Motivation

- SQL-92 cannot express some queries:
  - Are we running low on any parts needed to build a ZX600 sports car?
  - What is the total component and assembly cost to build a ZX600 at today's part prices?
- Can we extend the query language to cover such queries?
  - Yes, by adding recursion.

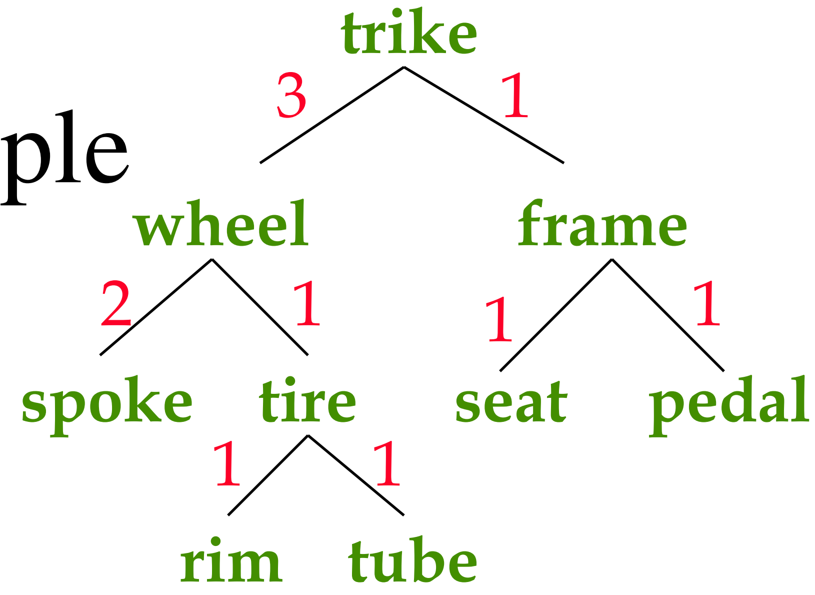
Datalog

- SQL queries can be read as follows:

“If some tuples exist in the From tables that satisfy the Where conditions, then the Select tuple is in the answer.”

- Datalog is a query language that has the same if-then flavor:
  - New: The answer table can appear in the From clause, i.e., be defined recursively.
  - Prolog style syntax is commonly used.

Example



trike

3

1

wheel

frame

2

1

1

1

spoke

tire

seat

pedal

1

1

rim

tube

part	subpart	number
trike	wheel	3
trike	frame	1
frame	seat	1
frame	pedal	1
wheel	spoke	2
wheel	tire	1
tire	rim	1
tire	tube	1

Assembly instance

- Find the components of a trike?
- We can write a relational algebra query to compute the answer on ***the given instance of Assembly*.**
- But there is no R.A. (or SQL-92) query that computes the answer on ***all Assembly instances***.

The Problem with R.A. and SQL-92

- Intuitively, we must join Assembly with itself to deduce that trike contains spoke and tire.
  - Takes us one level down Assembly hierarchy.
  - To find components that are one level deeper (e.g., rim), need another join.
  - To find all components, need as many joins as there are levels in the given instance!
- For any relational algebra expression, we can create an Assembly instance for which some answers are not computed by including more levels than the number of joins in the expression!

A Datalog Query that Does the Job

```
Comp(Part, Subpt) :- Assembly(Part, Subpt, Qty).
Comp(Part, Subpt) :- Assembly(Part, Part2, Qty),
                     Comp(Part2, Subpt).
```

head of rule **implication** **body of rule**

Can read the second rule as follows:

"**For all** values of Part, Subpt and Qty,
if there is a tuple (Part, Part2, Qty) in Assembly
and a tuple (Part2, Subpt) in Comp,
then there must be a tuple (Part, Subpt) in Comp."

Outline

- Parallel Databases
- Distributed Databases
- Object-database systems
- Deductive database systems
- **Data warehousing and decision support**
- Data mining
- Information Retrieval and XML data
- Spatial data management
- Other topics:
  - Advanced transaction processing,
  - Mobile databases
  - Main Memory databases
  - Multimedia databases
  - GIS
  - Temporal databases
  - Biological databases
  - Information visualization

Introduction

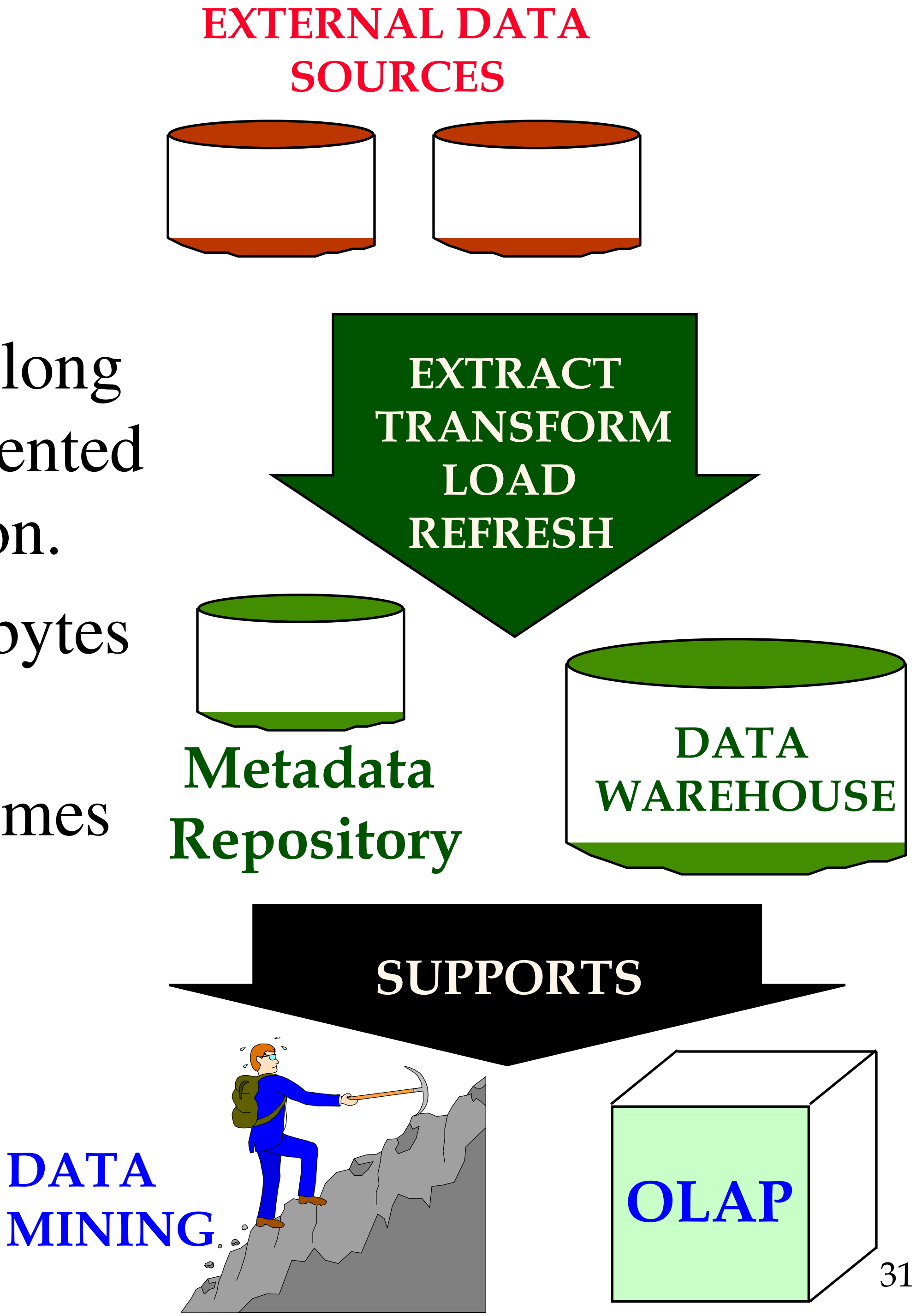
- Increasingly, organizations are analyzing current and historical data to identify useful patterns and support business strategies.
- Emphasis is on complex, interactive, exploratory analysis of very large datasets created by integrating data from across all parts of an enterprise; data is fairly static.
  - Contrast such **On-Line Analytic Processing (OLAP)** with traditional **On-line Transaction Processing (OLTP)**: mostly long queries, instead of short update Xacts.

Three Complementary Trends

- Data Warehousing: Consolidate data from many sources in one large repository.
  - Loading, periodic synchronization of replicas.
  - Semantic integration.
- OLAP:
  - Complex SQL queries and views.
  - Queries based on spreadsheet-style operations and "multidimensional" view of data.
  - Interactive and "online" queries.
- Data Mining: Exploratory search for interesting trends and anomalies. (Another lecture!)

Data Warehousing

- Integrated data spanning long time periods, often augmented with summary information.
- Several gigabytes to terabytes common.
- Interactive response times expected for complex queries; ad-hoc updates uncommon.



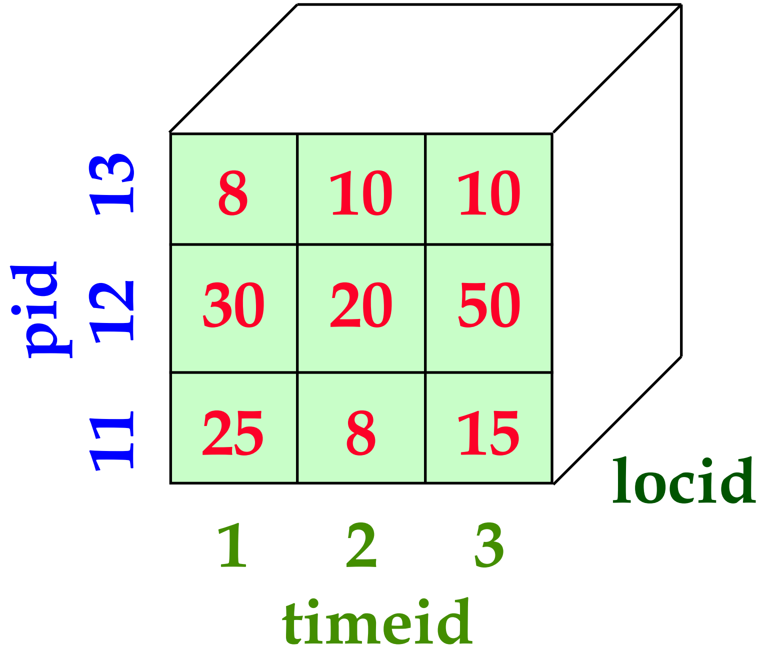
Warehousing Issues

- Semantic Integration: When getting data from multiple sources, must eliminate mismatches, e.g., different currencies, schemas.
- Heterogeneous Sources: Must access data from a variety of source formats and repositories.
  - Replication capabilities can be exploited here.
- Load, Refresh, Purge: Must load data, periodically refresh it, and purge too-old data.
- Metadata Management: Must keep track of source, loading time, and other information for all data in the warehouse.

Multidimensional Data Model: Data cube

- Collection of numeric measures, which depend on a set of dimensions.
  - E.g., measure **Sales**, dimensions **Product** (key: pid), **Location** (locid), and **Time** (timeid).

Slice locid=1 is shown:



pid	timeid	locid	sales
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35

• • •

MOLAP vs ROLAP

- Multidimensional data can be stored physically in a (disk-resident, persistent) array; called MOLAP systems. Alternatively, can store as a relation; called ROLAP systems.
- The main relation, which relates dimensions to a measure, is called the fact table. Each dimension can have additional attributes and an associated dimension table.
  - E.g., **Products(pid, pname, category, price)**
  - Fact tables are *much* larger than dimensional tables.

OLAP Queries

- Drill-down: The inverse of roll-up.
  - E.g., Given total sales by state, can drill-down to get total sales by city.
  - E.g., Can also drill-down on different dimension to get total sales by product for each state.
- Pivoting: Aggregation on selected dimensions.
  - E.g., Pivoting on Location and Time yields this **cross-tabulation**:

	WI	CA	Total
1995	63	81	144
1996	38	107	145
1997	75	35	110
Total	176	223	339

- Slicing and Dicing: Equality and range selections on one or more dimensions.

Summary

- Decision support is an emerging, rapidly growing subarea of databases.
- Involves the creation of large, consolidated data repositories called data warehouses.
- Warehouses exploited using sophisticated analysis techniques: complex SQL queries and OLAP "multidimensional" queries (influenced by both SQL and spreadsheets).
- New techniques for database design, indexing, view maintenance, and interactive querying need to be supported.

Outline

- Parallel Databases
- Distributed Databases
- Object-database systems
- Deductive database systems
- Data warehousing and decision support
- **Data mining**
- Information Retrieval and XML data
- Spatial data management
- Other topics:
  - Advanced transaction processing,
  - Mobile databases
  - Main Memory databases
  - Multimedia databases
  - GIS
  - Temporal databases
  - Biological databases
  - Information visualization

Definition

Data mining is the exploration and analysis of large quantities of data in order to discover valid, novel, potentially useful, and ultimately understandable patterns in data.

Valid: The patterns hold in general.

Novel: We did not know the pattern beforehand.

Useful: We can devise actions from the patterns.

Understandable: We can interpret and comprehend the patterns.

Why Use Data Mining Today?

Human analysis skills are inadequate:

- Volume and dimensionality of the data
- High data growth rate

Availability of:

- Data
- Storage
- Computational power
- Off-the-shelf software
- Expertise

An Abundance of Data

- Supermarket scanners, POS data
- Preferred customer cards
- Credit card transactions
- Direct mail response
- Call center records
- ATM machines
- Demographic data
- Sensor networks
- Cameras
- Web server logs
- Customer web site trails

Evolution of Database Technology

- 1960s: IMS, network model
- 1970s: The relational data model, first relational DBMS implementations
- 1980s: Maturing RDBMS, application-specific DBMS, (spatial data, scientific data, image data, etc.), OODBMS
- 1990s: Mature, high-performance RDBMS technology, parallel DBMS, terabyte data warehouses, object-relational DBMS, middleware and web technology
- 2000s: High availability, zero-administration, seamless integration into business processes
- 2010: Sensor database systems, databases on embedded systems, P2P database systems, large-scale pub/sub systems, ???

Much Commercial Support

- Many data mining tools
  - http://www.kdnuggets.com/software
- Database systems with data mining support
- Visualization tools
- Data mining process support
- Consultants

Why Use Data Mining Today?

Competitive pressure!

“The secret of success is to know something that nobody else knows.”

Aristotle Onassis

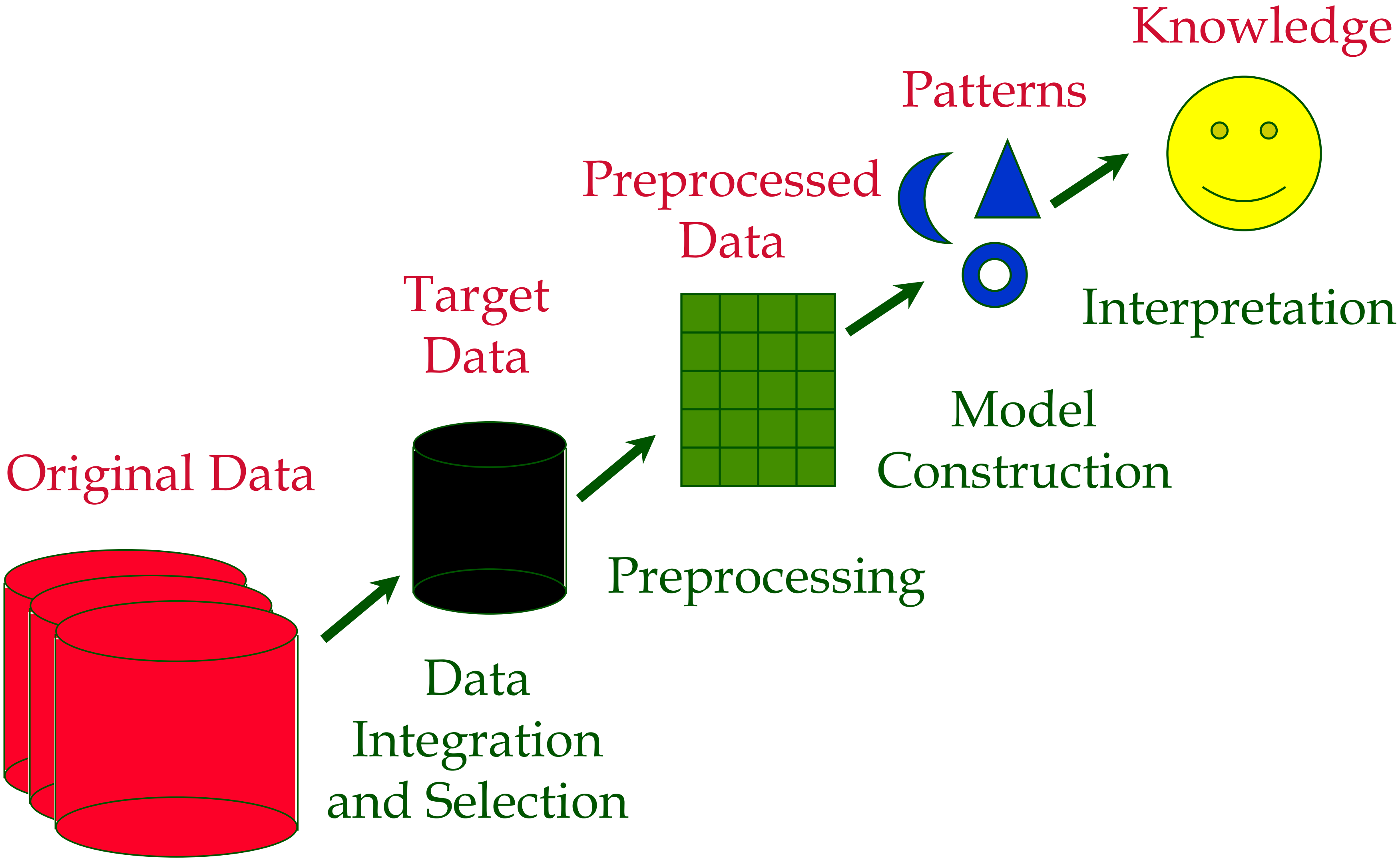
- Competition on service, not only on price (Banks, phone companies, hotel chains, rental car companies)
- Personalization, CRM
- The real-time enterprise
- “Systemic listening”
- Security, homeland defense

The Knowledge Discovery Process

Steps:

1. Identify business problem
2. Data mining
3. Action
4. Evaluation and measurement
5. Deployment and integration into businesses processes

Preprocessing and Mining



Original Data

Data Integration and Selection

Target Data

Preprocessing

Preprocessed Data

Model Construction

Patterns

Interpretation

Knowledge

Data Mining Techniques

- Supervised learning
  - Classification and regression
- Unsupervised learning
  - Clustering
- Dependency modeling
  - Associations, summarization, causality
- Outlier and deviation detection
- Trend analysis and change detection

Outline

- Parallel Databases
- Distributed Databases
- Object-database systems
- Deductive database systems
- Data warehousing and decision support
- Data mining
- **Information Retrieval and XML data**
- Spatial data management
- Other topics:
  - Advanced transaction processing,
  - Mobile databases
  - Main Memory databases
  - Multimedia databases
  - GIS
  - Temporal databases
  - Biological databases
  - Information visualization

Introduction

- 1940
  - We have vast amount of text information
  - Speedy access is becoming difficult
  - Relevant information gets ignored
    - Duplication of work and effort
  - IR is simple: library
    - Given the store of the documents and person formulates a question
    - The answer is a set of documents satisfying the information need expressed by his question
    - Solution: Read all the documents in the store and retain the relevant documents and discard all the others.
    - However, where is the Time ?

Introduction

- After the advent of computers
  - Many thought that computers will read the entire document collection to extract the relevant documents
    - Storing the documents is easy
  - But, reading like human is difficult
      - Attempt to duplicate the human with software process.
- Why
  - Reading by human being involves attempting to extract information both syntactic and semantic and using it to determine the **relevance** of the document.
  - Knowing to extract information is not sufficient.
  - But, we should know how to use it to decide relevance.
    - Slow progress in modern linguistics
    - Machine translation

Introduction

- Important notion in IR
  - Retrieve all the relevant documents; at the same time retrieve as few of the non-relevant as possible.
- Intellectually it is possible for a human to establish the relevance of a document to a query.
- For a computer to do this, we need to construct a model within which relevant decisions can be quantified.
- Research in IR
  - Model building to identify relevant documents

Information Retrieval System

- An information retrieval system does not inform (I.e., change the knowledge of) the user on the subject of his inquiry. It merely informs on the existence (or non existence) and whereabouts of documents relating to his request.
- Distinguishing properties of data retrieval (DBMS) and information retrieval

	Data Retrieval	**Information Retrieval**
Matching	Exact match	Partial match, best match
Inference	Deduction	Induction
Model	Deterministic	Probabilistic
Classification	Monothetic	Polythetic
Query language	Artificial	Natural
Query specification	Complete	Incomplete
Items wanted	Matching	Relevant
Error response	Sensitive	Insensitive

Introduction to Semistructured Data and XML

How the Web is Today

- HTML documents
  - often generated by applications
  - consumed by humans only
  - easy access: across platforms, across organizations
- No application interoperability:
  - HTML not understood by applications
    - screen scraping brittle
  - Database technology: client-server
    - still vendor specific

New Universal Data Exchange Format: XML

A recommendation from the W3C

- XML = data
- XML generated by applications
- XML consumed by applications
- Easy access: across platforms, organizations

Paradigm Shift on the Web

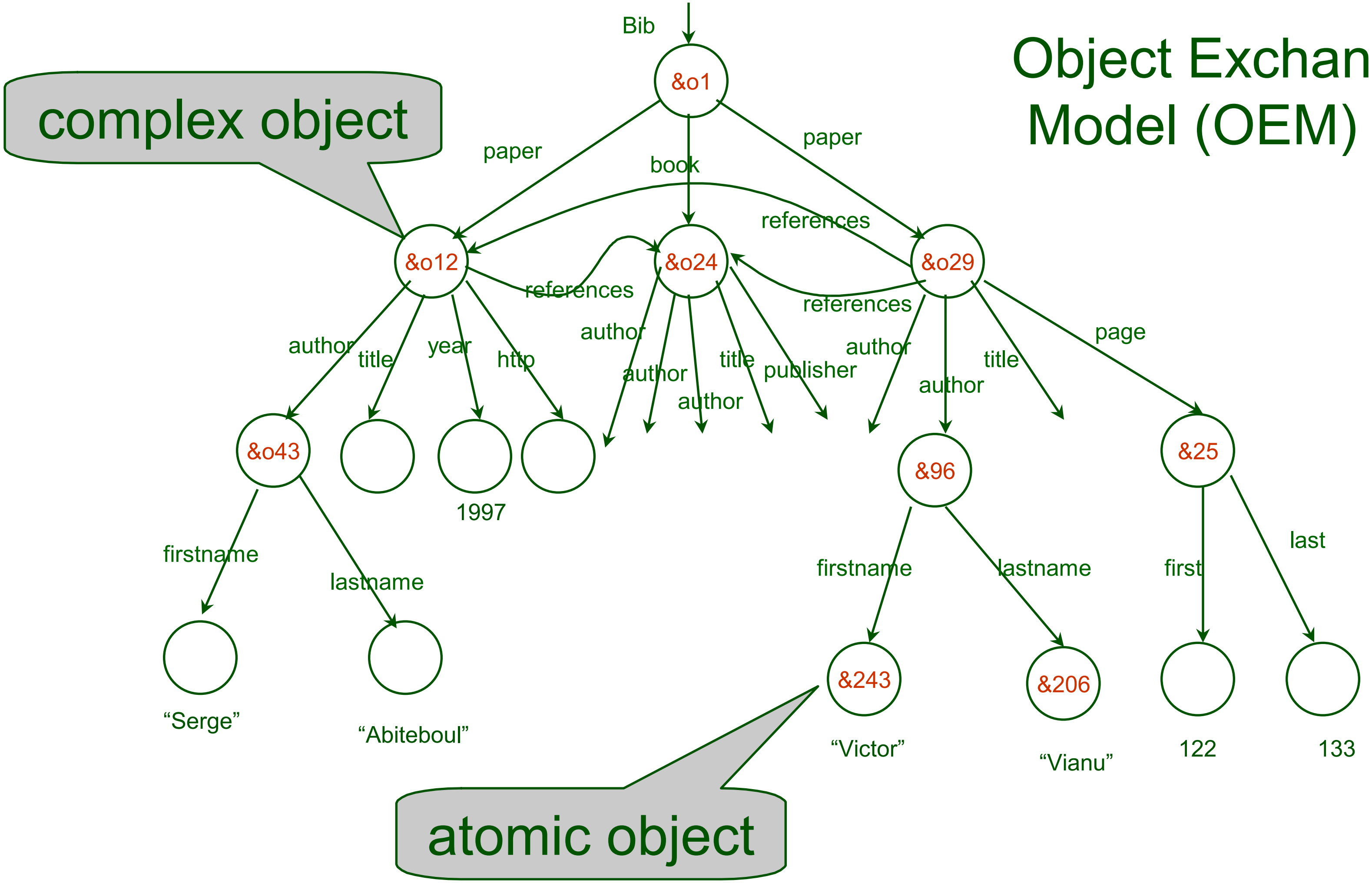
- From documents (HTML) to data (XML)
- From information retrieval to data management
- For databases, also a paradigm shift:
  - from relational model to semistructured data
  - from data processing to data/query translation
  - from storage to transport

Semistructured Data

Origins:

- Integration of heterogeneous sources
- Data sources with non-rigid structure
  - Biological data
  - *Web data*

The Semistructured Data Model



complex object

atomic object

Object Exchange Model (OEM)

Syntax for Semistructured Data

```
Bib: &o1 { paper: &o12 { … },
           book:  &o24 { … },
           paper: &o29
              { author: &o52 "Abiteboul",
                author: &o96 { firstname: &243 "Victor",
                               lastname: &o206 "Vianu"},
                title: &o93 "Regular path queries with constraints",
                references: &o12,
                references: &o24,
                pages: &o25 { first: &o64 122, last: &o92 133}
              }
          }
```

Observe: Nested tuples, set-values, oids!

Syntax for Semistructured Data

May omit oids:

```
{ paper: { author: “Abiteboul”,
           author: { firstname: “Victor”,
                     lastname: “Vianu”},
           title: “Regular path queries …”,
           page: { first: 122, last: 133 }
         }
}
```

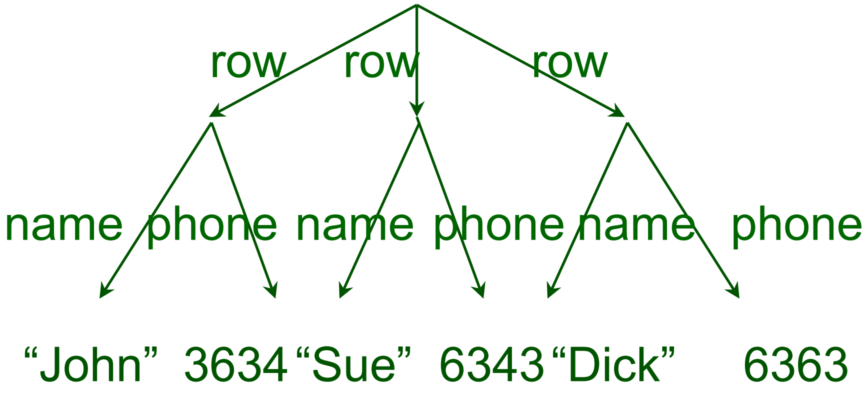
Characteristics of Semistructured Data

- Missing or additional attributes
- Multiple attributes
- Different types in different objects
- Heterogeneous collections

Self-describing, irregular data, no a priori structure

Comparison with Relational Data

name	phone
John	3634
Sue	6343
Dick	6363



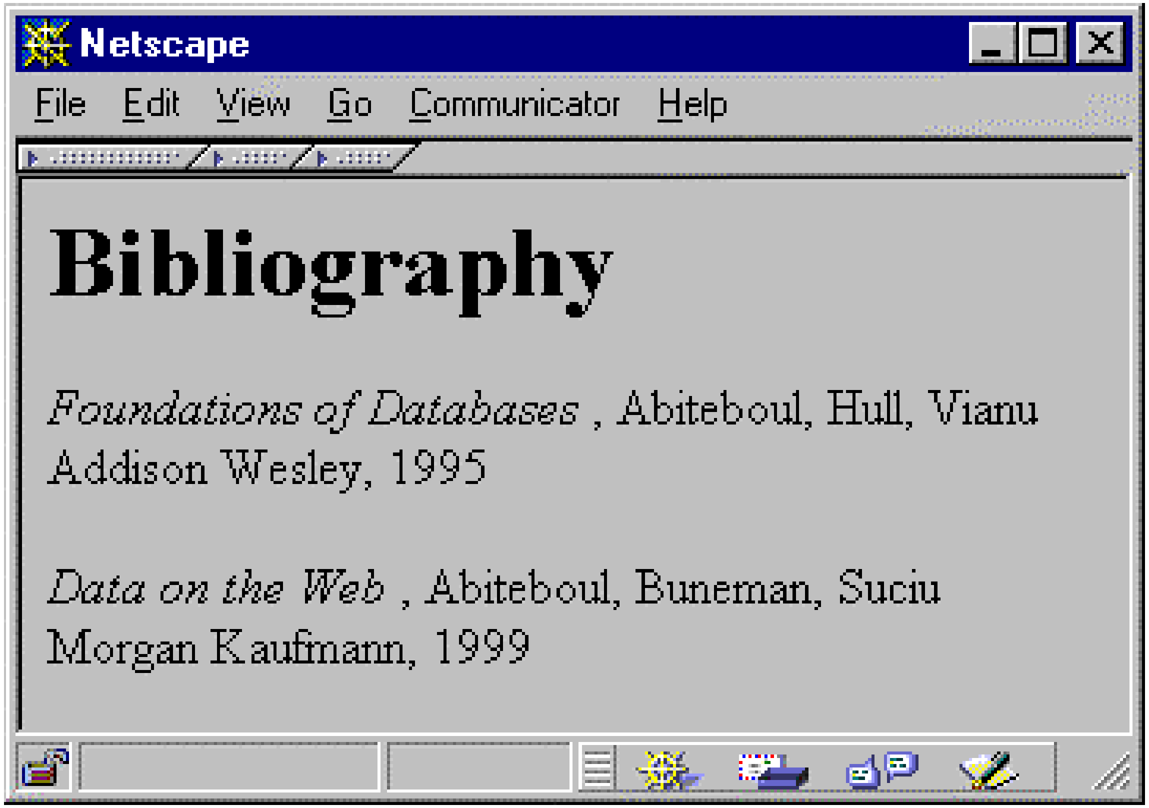
```
{ row: { name: “John”, phone: 3634 },
  row: { name: “Sue”,   phone: 6343 },
  row: { name: “Dick”,  phone: 6363 }
}
```

XML

- A W3C standard to complement HTML
- Origins: Structured text SGML
  - Large-scale electronic publishing
  - Data exchange on the web
- Motivation:
  - HTML describes presentation
  - XML describes content
- http://www.w3.org/TR/2000/REC-xml-20001006 (version 2, 10/2000)

$$\text{HTML4.0} \in \text{XML} \subset \text{SGML}$$

From HTML to XML



Bibliography

Foundations of Databases , Abiteboul, Hull, Vianu
Addison Wesley, 1995

Data on the Web , Abiteboul, Buneman, Suciu
Morgan Kaufmann, 1999

HTML describes the presentation

HTML

```
<h1> Bibliography </h1>
<p> <i> Foundations of Databases </i>
        Abiteboul, Hull, Vianu
        <br> Addison Wesley, 1995
<p> <i> Data on the Web </i>
        Abiteboul, Buneman, Suciu
        <br> Morgan Kaufmann, 1999
```

XML

```
<bibliography>
        <book>   <title> Foundations… </title>
                 <author> Abiteboul </author>
                 <author> Hull </author>
                 <author> Vianu </author>
                 <publisher> Addison Wesley </publisher>
                 <year> 1995 </year>
        </book>
        …
</bibliography>
```

XML describes the content

Why are we DB’ers interested?

- It’s data.
- Proof by Google:
  - database+XML – 1,940,000 pages.
- Database issues:
  - How are we going to model XML? (graphs).
  - How are we going to query XML? (XQuery)
  - How are we going to store XML (in a relational database? object-oriented? native?)
  - How are we going to process XML efficiently? (many interesting research questions!)

Outline

- Parallel Databases
- Distributed Databases
- Object-database systems
- Deductive database systems
- Data warehousing and decision support
- Data mining
- Information Retrieval and XML data
- **Spatial data management**
- Other topics:
  - Advanced transaction processing,
  - Mobile databases
  - Main Memory databases
  - Multimedia databases
  - GIS
  - Temporal databases
  - Biological databases
  - Information visualization

Types of Spatial Data

- Point Data
  - Points in a multidimensional space
  - E.g., *Raster data* such as satellite imagery, where each pixel stores a measured value
  - E.g., Feature vectors extracted from text
- Region Data
  - Objects have spatial extent with location and boundary
  - DB typically uses geometric approximations constructed using line segments, polygons, etc., called *vector data*.

Types of Spatial Queries

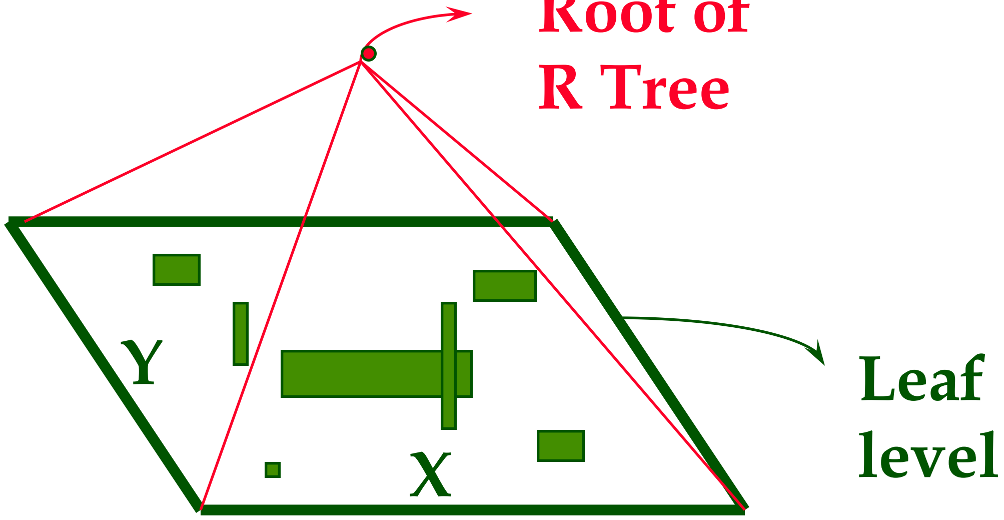
- Spatial Range Queries
  - *Find all cities within 50 miles of Madison*
  - Query has associated region (location, boundary)
  - Answer includes ovelapping or contained data regions
- Nearest-Neighbor Queries
  - *Find the 10 cities nearest to Madison*
  - Results must be ordered by proximity
- Spatial Join Queries
  - *Find all cities near a lake*
  - Expensive, join condition involves regions and proximity

Applications of Spatial Data

- Geographic Information Systems (GIS)
  - E.g., ESRI's ArcInfo; OpenGIS Consortium
  - Geospatial information
  - All classes of spatial queries and data are common
- Computer-Aided Design/Manufacturing
  - Store spatial objects such as surface of airplane fuselage
  - Range queries and spatial join queries are common
- Multimedia Databases
  - Images, video, text, etc. stored and retrieved by content
  - First converted to *feature vector* form; high dimensionality
  - Nearest-neighbor queries are the most common

Multi-dimensional Indexing: The R-Tree

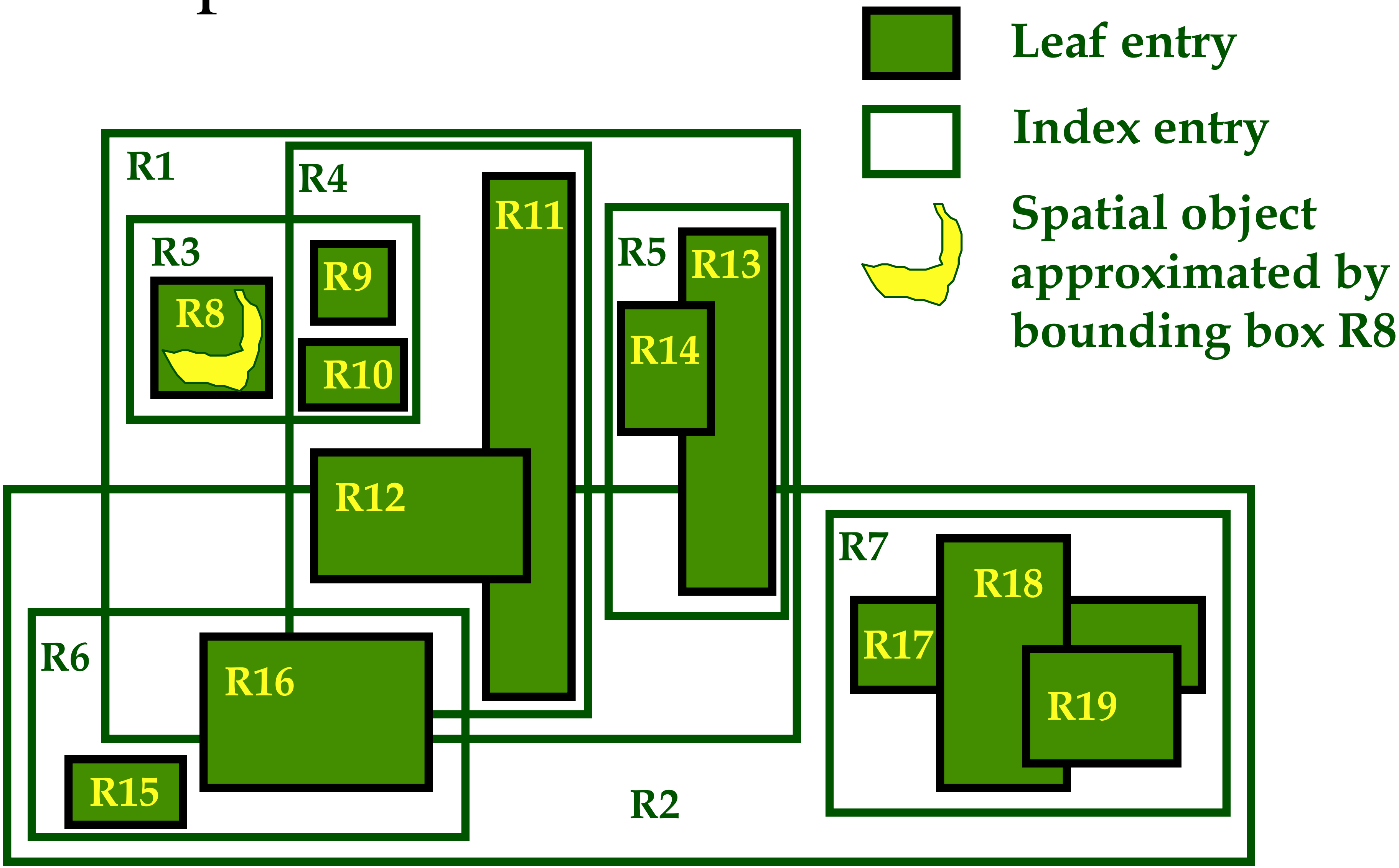
- The R-tree is a tree-structured index that remains balanced on inserts and deletes.
- Each key stored in a leaf entry is intuitively a box, or collection of intervals, with one interval per dimension.
- Example in 2-D:



R-Tree Properties

- Leaf entry = < n-dimensional box, rid >
  - This is Alternative (2), with *key value* being a box.
  - Box is the tightest bounding box for a data object.
- Non-leaf entry = < n-dim box, ptr to child node >
  - Box covers all boxes in child node (in fact, subtree).
- All leaves at same distance from root.
- Nodes can be kept 50% full (except root).
  - Can choose a parameter *m* that is <= 50%, and ensure that every node is at least *m*% full.

Example of an R-Tree

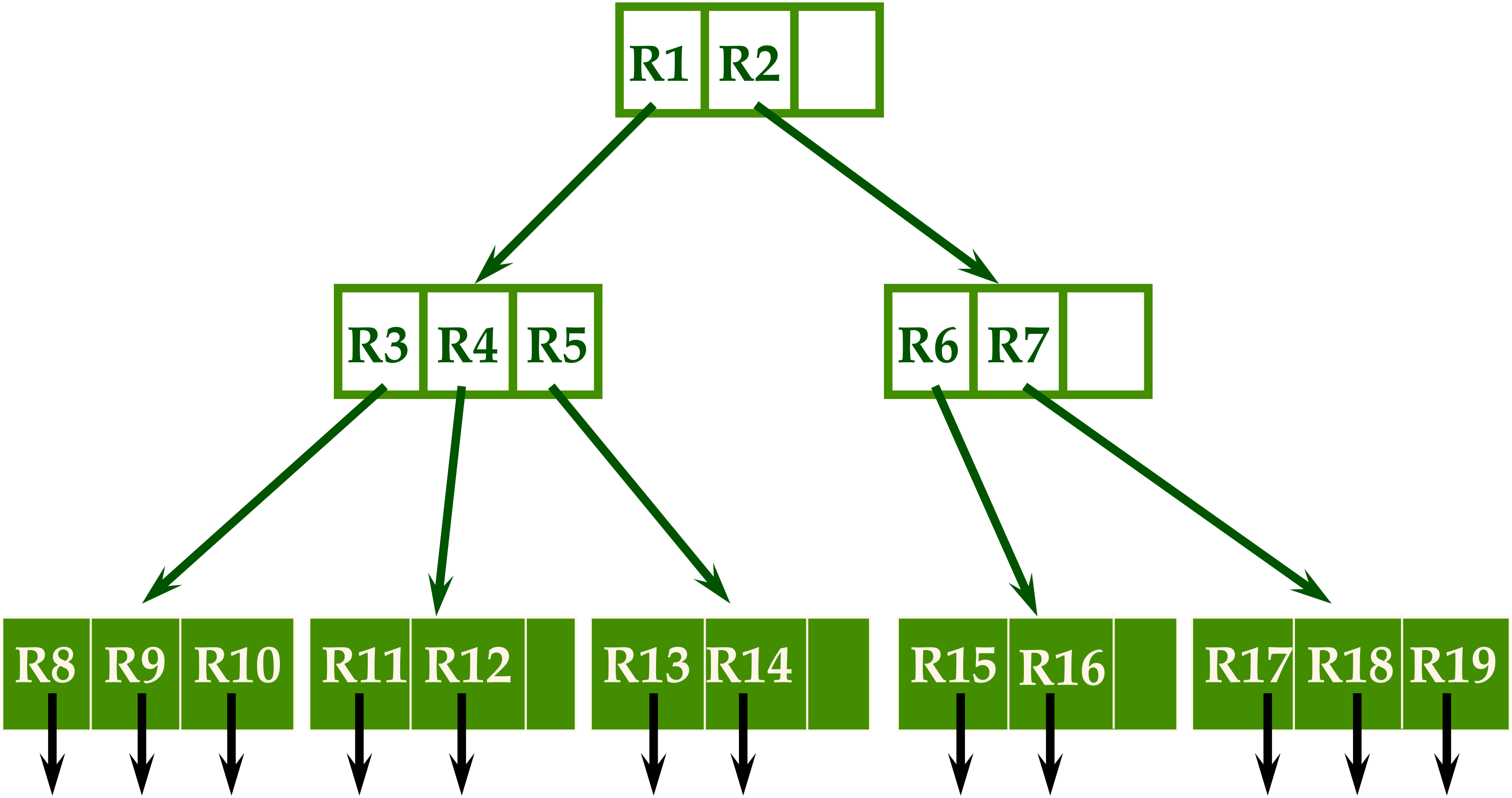


Leaf entry

Index entry

Spatial object approximated by bounding box R8

Example R-Tree (Contd.)



Outline

- Parallel Databases
- Distributed Databases
- Object-database systems
- Deductive database systems
- Data warehousing and decision support
- Data mining
- Information Retrieval and XML data
- Spatial data management
- **Other topics:**
  - **Advanced transaction processing,**
  - **Mobile databases**
  - **Main Memory databases**
  - **Multimedia databases**
  - **GIS**
  - **Temporal databases**
  - **Biological databases**
  - **Information visualization**

Other Topics

- TP monitors
  - Glues together the services of several resource managers and provides application programmers ACID properties.
- New Transaction Models
  - Multi-level transactions
  - Nested transactions
  - Example: Transaction in CAD environment
- Real-time DBMS
  - Transactions must be executed within a deadline (Soft and hard).
- Data integration
  - Users wants to access data from more than one resource
    - XML is used
- Mobile databases
  - Different from distributed DBMS
  - Bandwidth is 10 times less than Internet
  - User's locations constantly change
  - Can access multiple database severs using single transaction

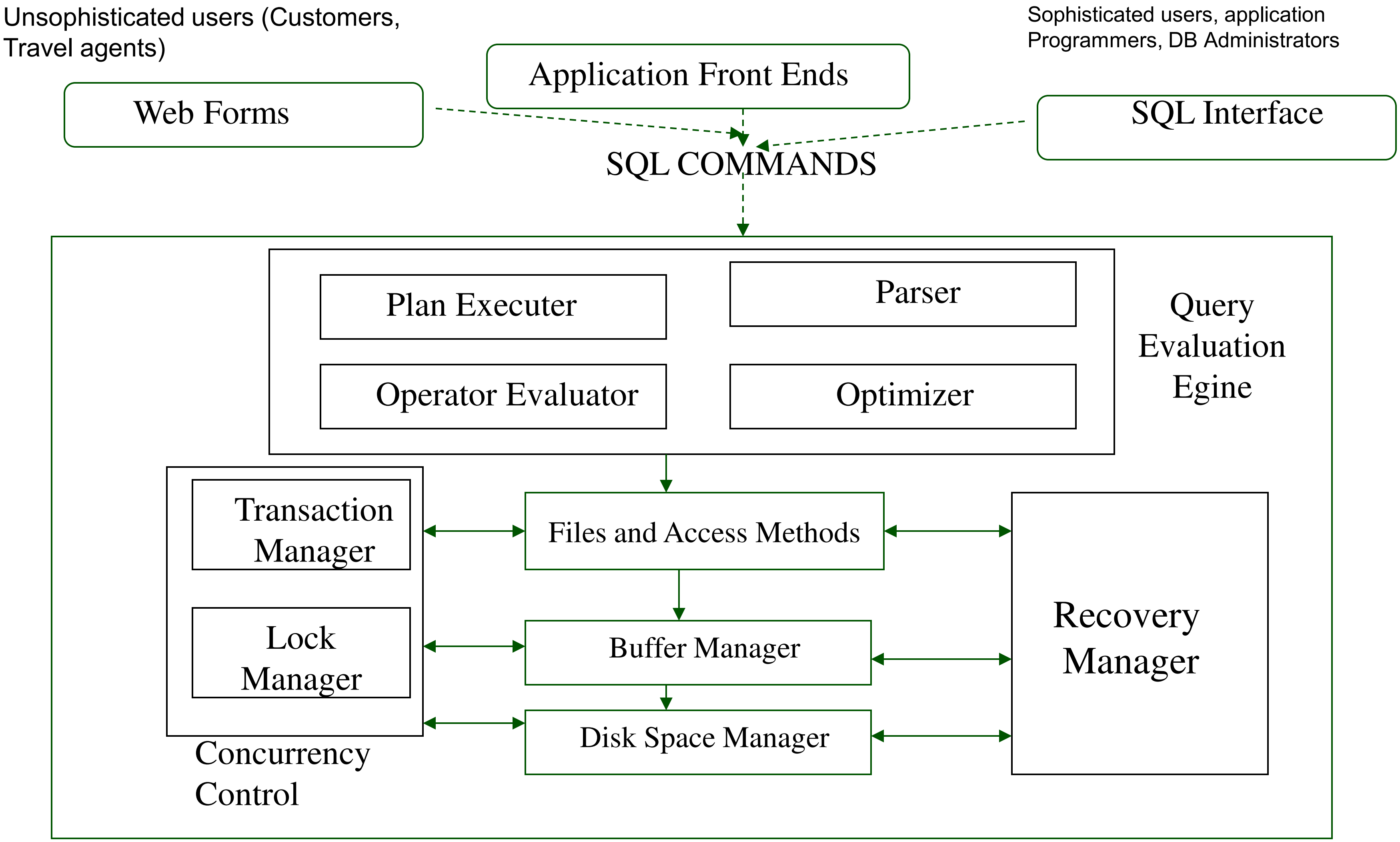
Other Topics…

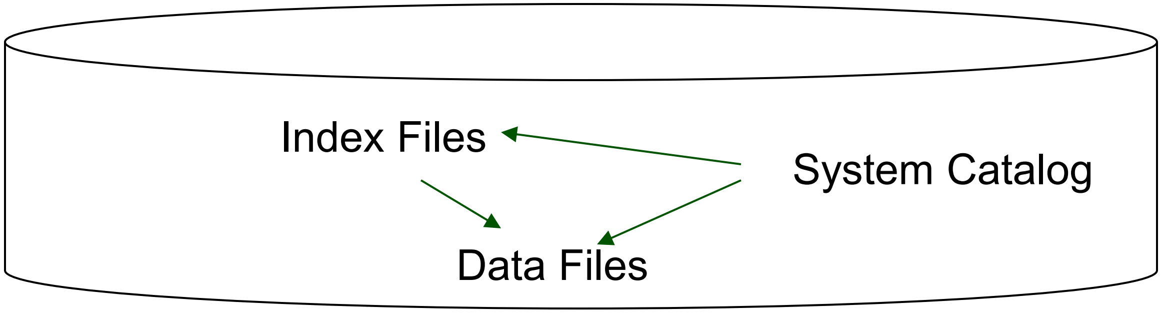
- Main Memory databases
  - Can buy enough main memory to hold entire database.
  - For recovery, DISK should be involved.
  - All optimization algorithms change.
  - Page-oriented data structures become less important
- Multimedia databases
  - Data is very large collections of multiple media: image, audio, video, text, sequence data
  - Content-based retrieval.
    - Users should specify selection conditions based on the contents of multimedia objects.
    - Example: Find all other images similar to the given image.
  - Managing the repositories of large objects
    - Objects are very large; compressions techniques should be employed.
  - Video on demand
    - Video must be deliveed to the user’s computer in real time, reliably and inexpensively.

Other Topics…

- GIS
  - GIS contain spatial information about states, countries, streets, lakes, rivers and other features support applications to combine such spatial information with non-spatial data
  - Applications
    - Vehicle navigation aids.
    - With GPS the car's location can be pin-pointed by accessing the database of local maps.
- Temporal databases
  - Data record have a valid time (valid in the real world) period.
  - From transactions and to flields
- Biological databases
  - Bioinformatics
    - Intersection of biology and Computer Science
  - Data has two interesting characteristics
    - Loosely structured data
    - Sequence data
      - DNS sequences
- Information Visualization
  - Present big results visually

Architecture Of DBMS


Unsophisticated users (Customers, Travel agents)
Sophisticated users, application Programmers, DB Administrators
Web Forms
Application Front Ends
SQL Interface
SQL COMMANDS
Plan Executer
Parser
Operator Evaluator
Optimizer
Query Evaluation Egine
Transaction Manager
Lock Manager
Concurrency Control
Files and Access Methods
Buffer Manager
Disk Space Manager
Recovery Manager


Index Files
System Catalog
Data Files