信息技术前沿课

孟小峰

email: xfmeng@ruc.edu.cn

http://www.ccf-dbs.org.cn/idke/xfmeng

信息技术前沿课

- 数据库技术的发展
- 挑战性问题
- 目前的研究工作
  - Web数据管理
  - XML数据库
  - 移动数据管理
- 博士论文的要求

COMPUTERWORLD

- Taming Data Chaos **Editor's Note: The Knowledge Center Special Report on data management will help you cope with the motley collection of data you've been accumulating.**
  The Story So Far **In the history of database and "business intelligence" software, users such as BF Goodrich and Procter & Gamble played a major role.**
  Merging Data Silos **Cleansing and combining data from various databases is hard work. But it could save your CRM, ERP and supply chain projects.**

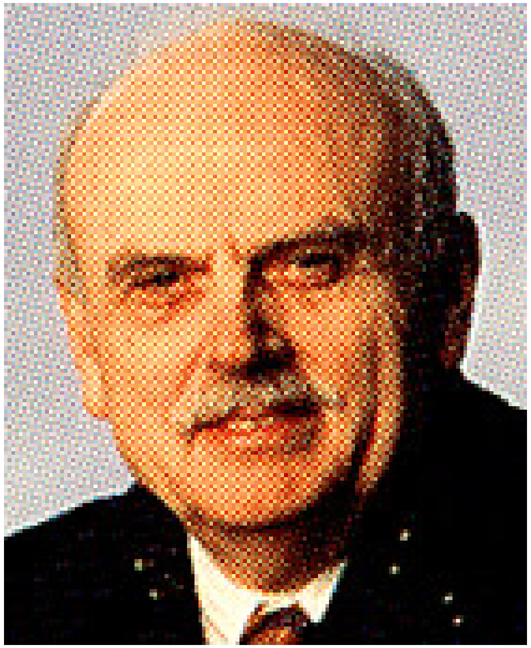
And now, on with the story. . . .

- **1951:** The Univac uses magnetic tape as well as punched cards for data storage.
- **1956:** IBM introduces first magnetic hard disk drive in its Model 305 RAMAC.
- **1961:** Charles Bachman at GE develops the first database management system, IDS

1951: Univac uses magnetic tape as well as punched cards for data storage.

And now, on with the story. . . .

- **1969:** Edgar F. “Ted” Codd invents the relational database.
- **1973:** Cullinane, led by John J. Cullinane, ships IDMS, a network-model database for IBM mainframes.
- **1976:** Honeywell ships Multics Relational Data Store, the first commercial relational database.



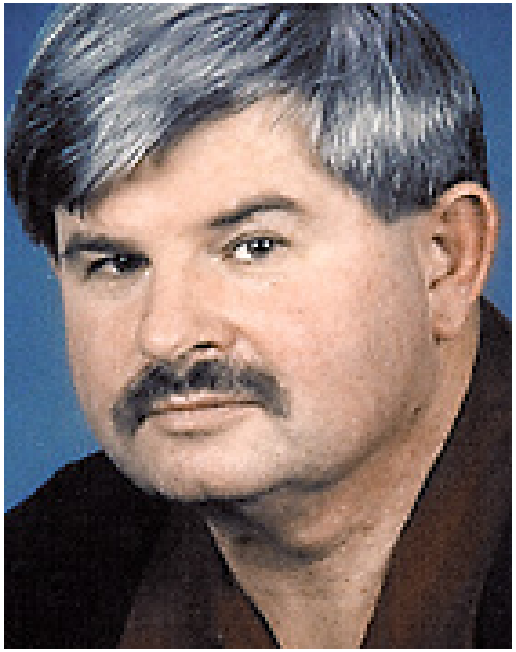
1969: Edgar F. "Ted" Codd invents the relational database.

C. J. Date

- C.J.Date是关系数据库技术领域中非常著名的独立撰稿人，学者和顾问。现在在加利福尼亚的海得斯堡工作。
- 参与了IBM公司的SQL/DS和DB2两大产品的技术规划和设计。他于1983年5月离开IBM公司
- 30多年来，Date 先生一直活跃在数据库领域中。他是最早认识到Codd在关系模型方面所做的开创性贡献的学者之一
  - 《数据库系统导论》，7ed,
  - 《对象关系数据库基础：第三次宣言》（1998）
  - 他的著作被翻译为多种语言并广为传播，如：中文，荷兰语，法语，德语，希腊语，意大利语，日语，朝鲜语，波兰语，葡萄牙语，俄语 ，西班牙语和盲人用的布利叶文字。

And now, on with the story. . . .

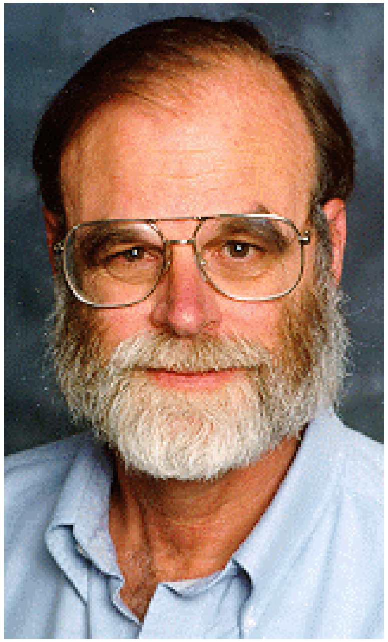
- **1979:** Oracle introduces the first commercial SQL relational database management system.
- **1983:** IBM introduces DB2.
- **1985:** The first business intelligence system is designed for Procter & Gamble.
- **1991:** W.H. “Bill” Inmon publishes Building the Data Warehouse.



1991: W.H. "Bill" Inmon publishes Building the Data Warehouse.

And now, on with the story. . . .

- **1992:** *Transaction Processing* published.
- **From 1997**
  - Web computing & Databases
  - Grid computing & Databases
  - Pervasive computing & Databases
  - ……



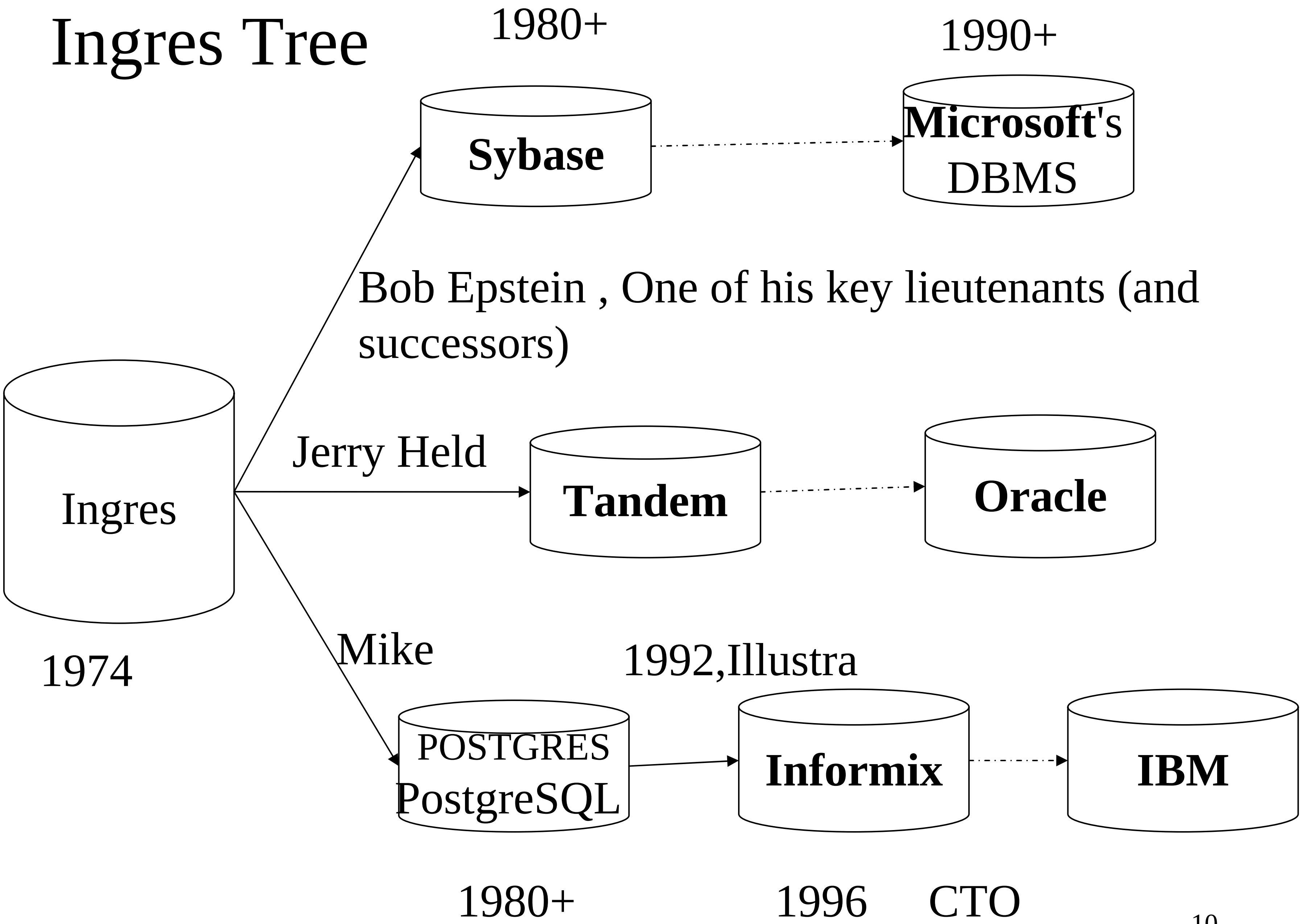
1992 Jim published the *Transaction Processing*

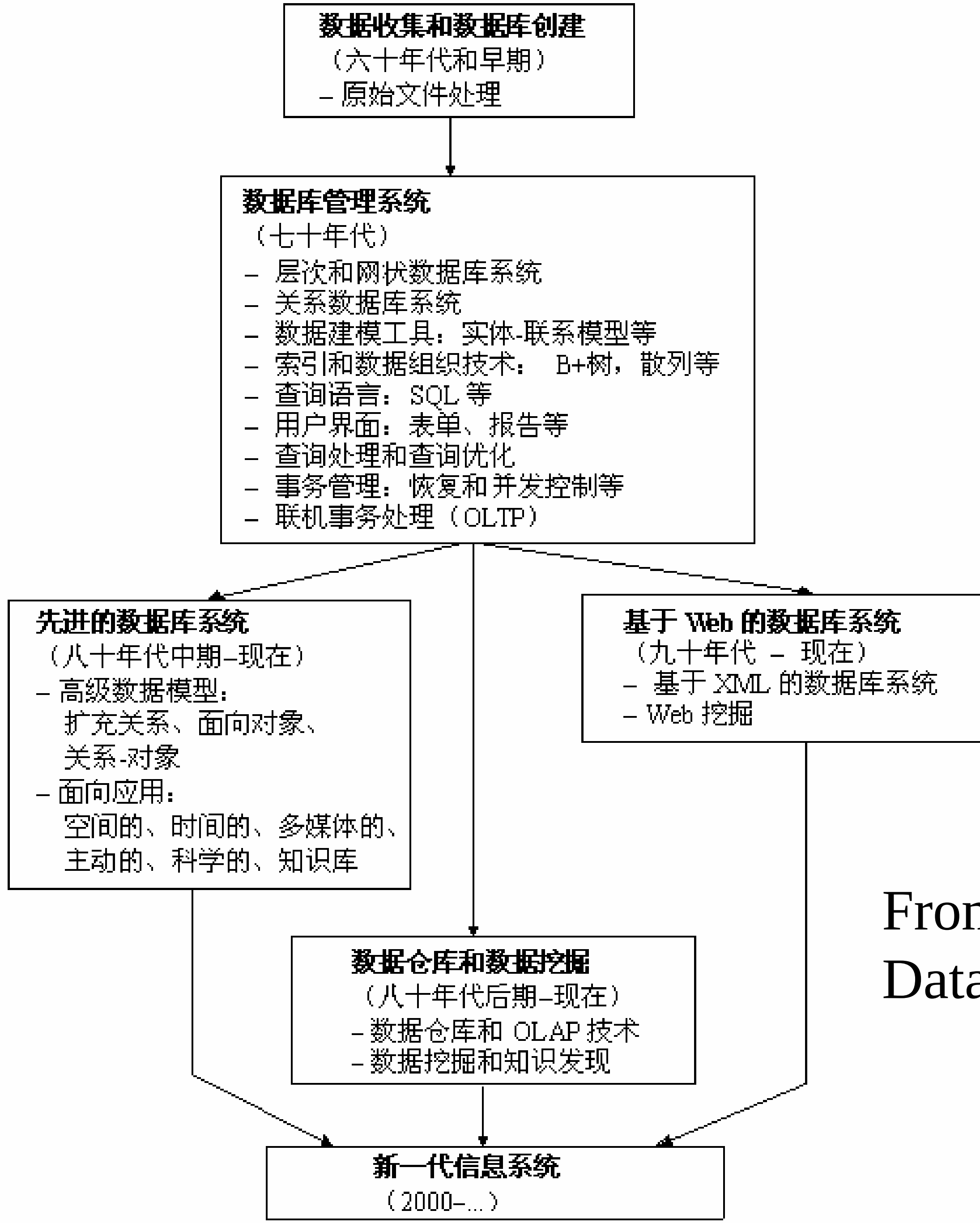
The greatest living contributor to database technology

- INGRES research project at UC Berkely Along with Eugene Wong and grad student Jerry Held
- Spun off the company later known as **Ingres,** Oracle's chief direct competitor in its early years
- He retired from UC Berkeley in 2000 and is currently an adjunct professor of computer science at MIT

Michael Stonebraker, co-inventor of the relational database.

Ingres Tree


1980+
1990+
Sybase
Microsoft's DBMS
Bob Epstein , One of his key lieutenants (and successors)
Jerry Held
Ingres
Tandem
Oracle
1974
Mike
1992,Illustra
POSTGRES
PostgreSQL
Informix
IBM
1980+
1996
CTO



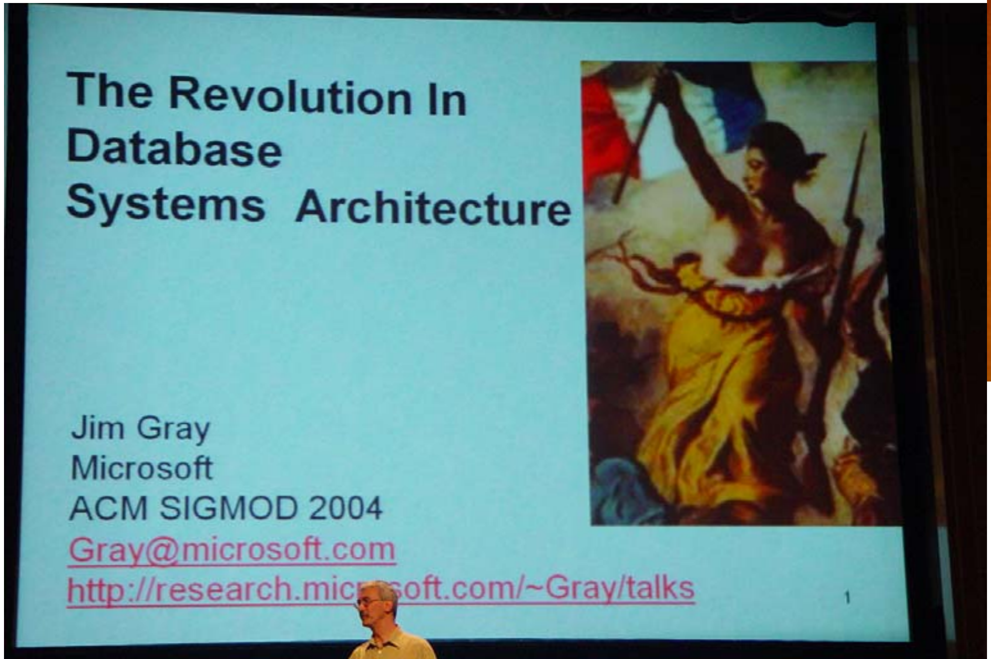
From: Han Jiawei, Data Mining

数据库的挑战:
Senior database researcher Meeting

- Senior database researchers have gathered every few years to assess the state of database research and to recommend problems and problem areas deserve additional focus.
  - Laguna Beach, Calif. in 1989 [1]
  - Palo Alto, Calif. ("Lagunita") in 1990 [2] and 1995 [3]
  - Cambridge, Mass. in 1996 [4]
  - Asilomar, Calif. in 1998 [5]
  - Lowell, Mass . In 2003

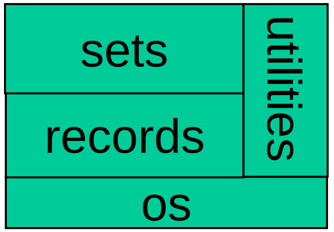
对数据库技术发展的思考

- Jim Gray在SIGMOD2004年会的主题发言
  - 数据库体系结构面临革命性变革

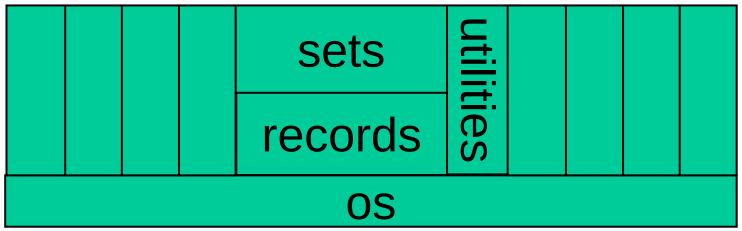


DB Systems evolved to be **containers for information services** develop, deploy, and execution environment

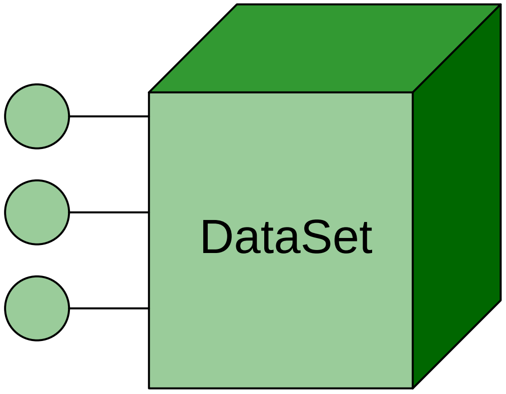
- The classic DBMS model



- The current situation
  - + Programming Languages
  - + Triggers and queues
  - + Replication, Pub/sub
  - + Extract-Transform-Load
  - + Text, Time, Space
  - + Cubes, Data mining
  - + XML, XQuery
  - + Many more extensions comming

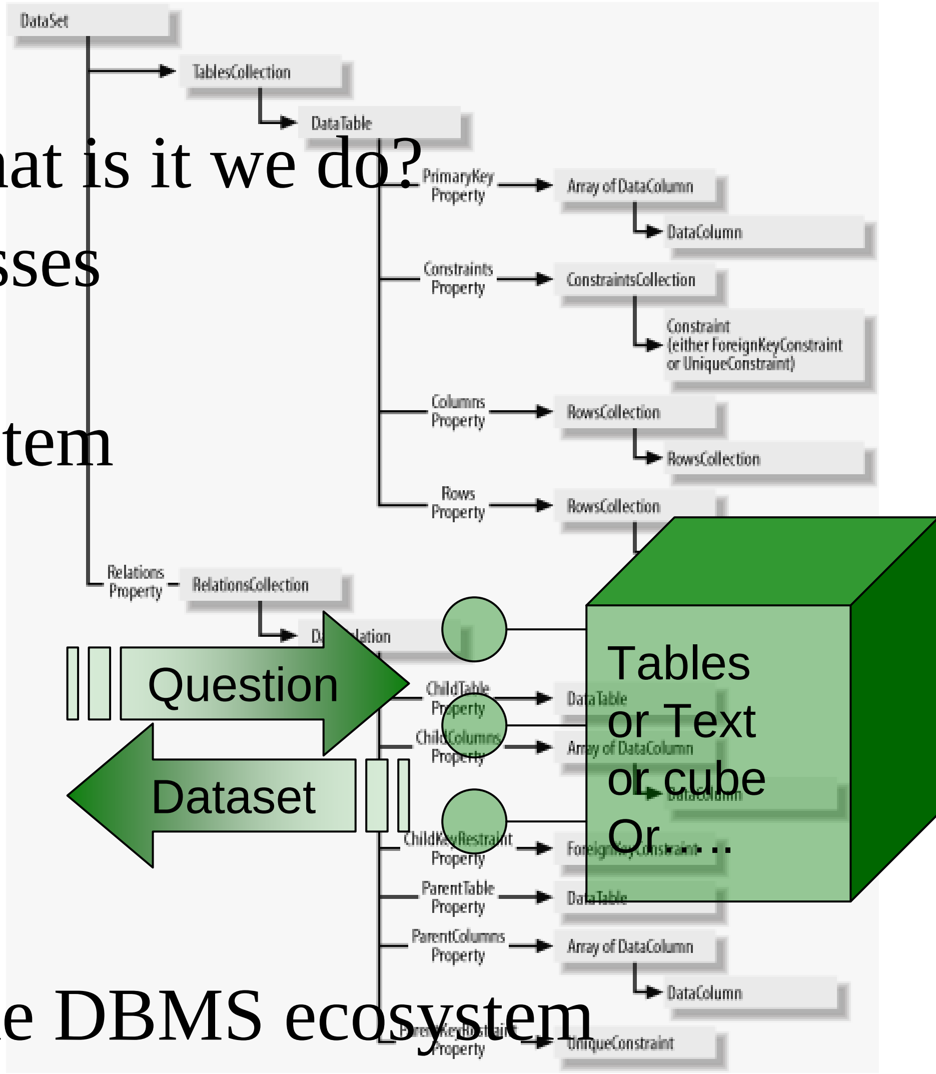


- DBMS is an ecosystem
  OO is the key structuring strategy:
  - Everything is a class
  - Database is a complex object
  - Core object is DataSet
  - Classes publish/consume them
  - Depends on strong Object Model



- Many of the concepts you pioneered are now mainstream.

Ask not "How to add objects to databases", Ask "What kind of object is a database?"



Q: Given an object model, what is it we do?

A: RecordSet and DataSet classes
and their methods

This is the basis for the ecosystem

- Distributed DB
- Extensible DB
- Interoperable DB
- ….

This was implicit in ODBC
but is now explicit within the DBMS ecosystem

Input: Command (any language)

Output: Dataset

Code and Data: Separated at Birth

COBOL

– IDENTIFICATION: document

AUTHOR, PROGRAM-ID, INSTALLATION,
SOURCE-COMPUTER, OBJECT-COMPUTER,
SPECIAL-NAMES, FILE-CONTROL, I-O-CONTROL,
DATE-WRITTEN, DATE-COMPILED,
SECURITY.

– ENVIRONMENT: OS

CONFIGURATION SECTION.
INPUT-OUTPUT SECTION.

– DATA: Files/Records

FILE SECTION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
REPORT SECTION.
SCREEN SECTION.

– PROCEDURE: code

"them"

"us"

CODASYL - DBTG

COnference on DAta SYstems Languages
Data Base Task Group

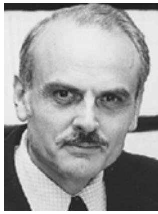
Defined DDL for a network data model

Set-Relationship semantics
Cursor Verbs

Isolated from procedures.

No encapsulation

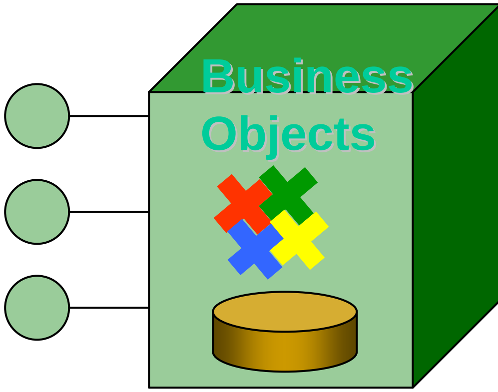
Klaus Wirth: *Algorithms + Data Structures = Programs*

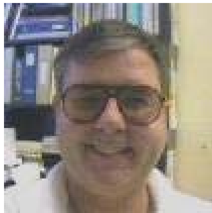


The Object-Relational World

marry programming languages and DBMSs

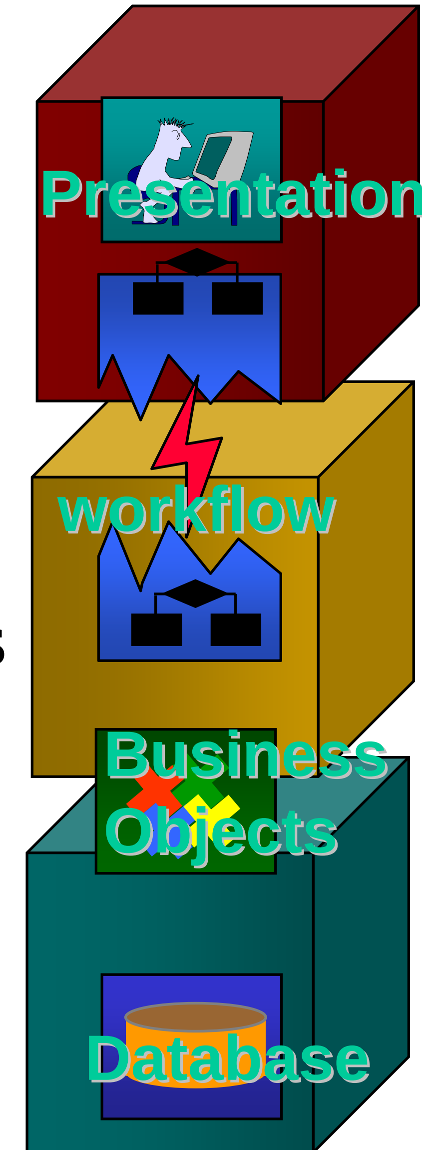
- Stored procedures evolve to "real" languages Java, C#,.. With real object models.
- Data encapsulated: a class with methods
- Classes may be persistent
- Records are vectors of objects
- Tables are enumerable & indexable
- Opaque or transparent types
- Set operators on transparent classes
- Transactions:
  - Preserve invariants
  - A composition strategy
  - An exception strategy
- **Ends Inside-DB Outside-DB dichotomy**





What's Outside?
Classic: Three Tier Computing

- Clients do presentation, gather input
- Do some workflow (script)
- Send high-level requests to ORB (Object Request Broker)
- ORB dispatches workflows and business objects -- proxies for client, orchestrate flows & queues
- Server-side workflow invokes distributed business objects to execute task
- Business object read/write database

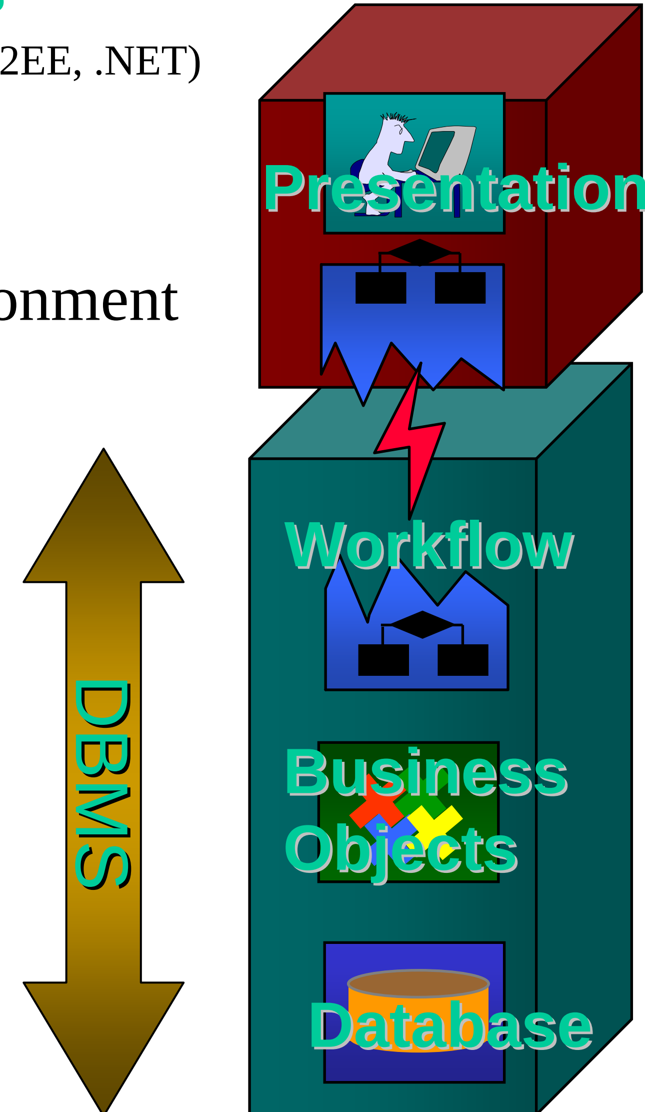


DBMS is Web Service!

Client/server is back; the revenge of TP-lite

Intelligence migrated to clients

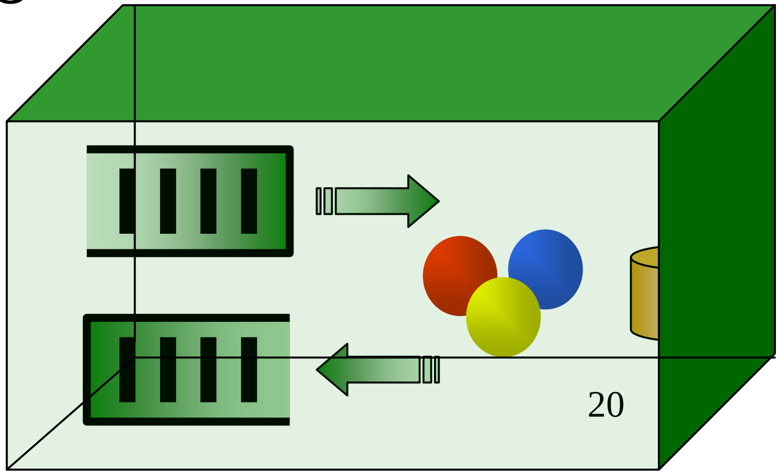
- Web servers and runtimes (Apache, IIS, J2EE, .NET) displaced TP monitors & ORBS
  - Give persistent objects
  - Holistic programming model & environment
- Web services (soap, wsdl, xml) are displacing current brokers
- DBMS listening to Port 80 publishing WSDL, DISCO, Servicing SOAP calls.
  DBMS is a web service
- Basis for distributed systems.
- A consequence of OR DBMS



Queues, Transactions, Workflows

- The world is loosely connected via
  Queued messages
- Queues are databases.
- Queues: workflow basis
- Queues: the first class to add to an OR DBMS
- Queues fire triggers.
  Active databases
- Queues cohabit with DBMS

Workflow:
Script
Execute
Administer &
Expedite
all built on queues



Text, Temporal, and Spatial Data Access

- **Q**: What comes after queues?

  A: Basic types: text, time, space,…
- Great application of OR technology
- Key idea:
  table valued functions == indices

  An index is a table, organized differently

  Query executor uses index to map:

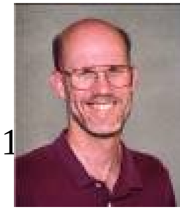
  Key → set (aka sequence of rows)
- Table valued function can do this map

  Optimizer can use it.
- +extras: cost function, cardinality,…
- BIG DEAL: Approximate answers:
  Rank and Support

```
select Title, Abstract, Rank
from Books join
    FreeTextTable(Title,
                  Abstract,
                  'XML semistructured') T
on  BookID = T.Key
```

```
select galaxy, distance
from GetNearbyObjEQ(22,37)
```

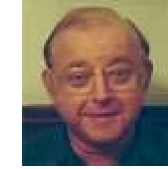
```
select store, holiday, sum(sales)
from Sales join
    HolidayDates(2004) T
on Sales.day = T.day
group by store, holiday
```



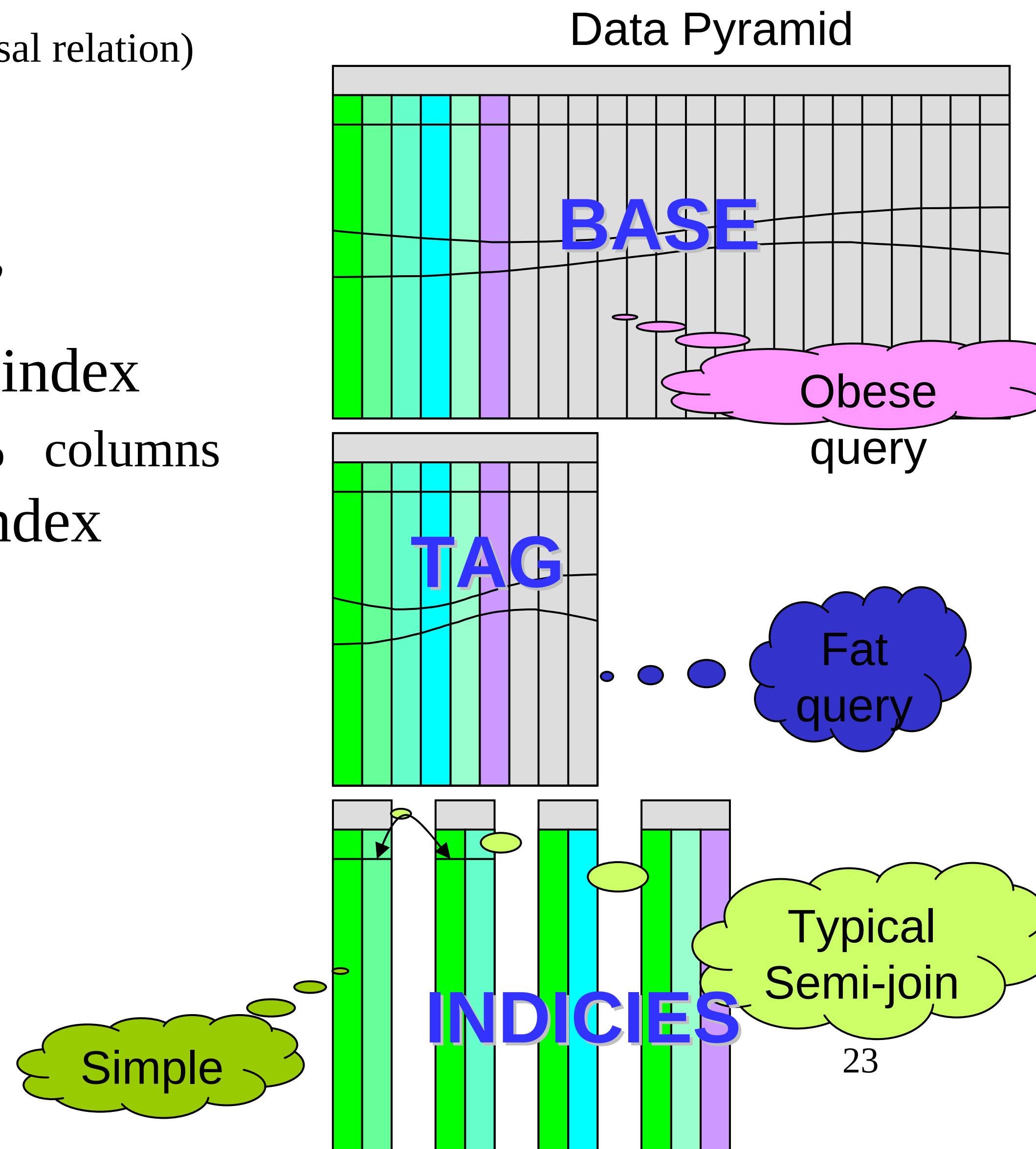
What's new here?

- DBMS have tight-integration with language classes (Java, C#, VB,..)
- The DB is a class
- You can add classes to DB.
  Adding indices is "easy"
  If you have a new idea.
- Now have solid Queue systems
  Adding workflow is "easy"
  If you have a new idea.

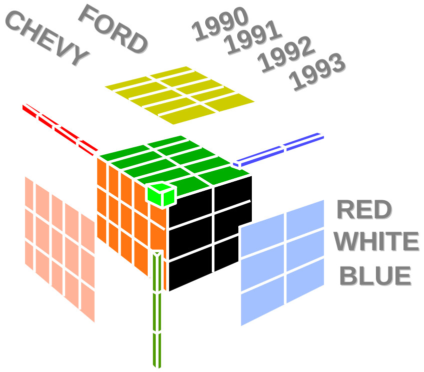
Column Stores & Row Stores



- Users see fat base tables (universal relation)
- Conceptually simple
  but use only some columns
- To avoid reading useless data,
  Do vertical partitions
  Define 10% popular columns index
- Make many skinny indices 1% columns
- Query engine uses covering index
- Much faster read
  slower insert/update
- MANY! optimizations
  (bitmaps, compression,..).
- Column stores automate
  all this,
  see Adabase, Model204 and…
- Challenge: Automate design.



Cubes

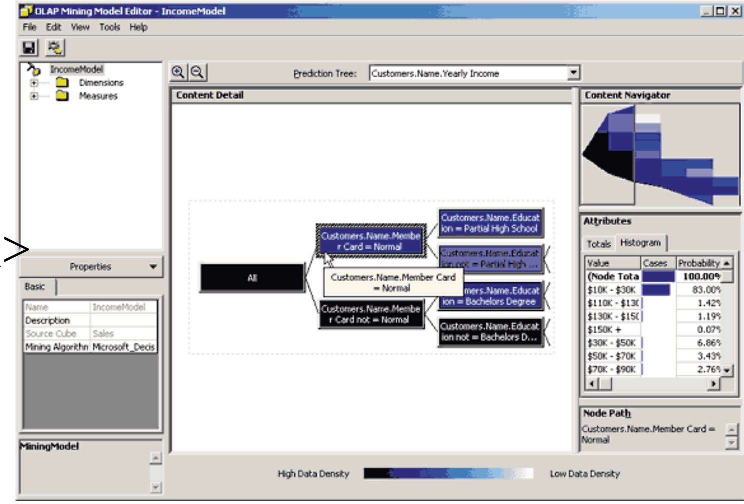


- Data cubes now standard
- Cube stores cohabit with row stores
  ROLAP + MOLAP
  (relational +multidimensional online analytic processing)
- Dimension, Measure, Operator concepts highly evolved beyond snowflake schema
- MDX is very powerful
- very sophisticated algorithms
- A big part of the ecosystem

```
SELECT &lt;axis_spec&gt;
FROM   &lt;cube_spec&gt;
WHERE  &lt;slicer_spec&gt;
```


Data Mining and Machine Learning

- Tasks: classification, association, prediction
- Tools: Decision trees, Bayes, apriori, clustering, regression, neural net,…
- now unified with DBs
  - Create table T (x,y,z,u,v,w)
    Learn “x,y,z” from “u,v,w” using <algorithm>
  - Train T with data.
  - Then can ask:
    - Probability x,y,z,u,v,w
    - What are the u,v,w probabilities given x,y,z
  - Example: Learn height from age.
- Anyone with a data mining algorithm has full access to the DBMS infrastructure.
- Challenge: Better learning algorithms.



DM – DB Synergy

Create the model:

```
CREATE MINING MODEL  HeightFromAgeSex
                    ( ID long key,
                      Gender   text discrete,
                      Age long continuous,
                      Height long continuous PREDICT)
                      USING Decision_Trees
```

learn height
from Gender + Age

Train a data mining model:

```
INSERT INTO Height
          SELECT  ID, Gender, Age, Height
          FROM People
```

DB verbs to
drive Modeler

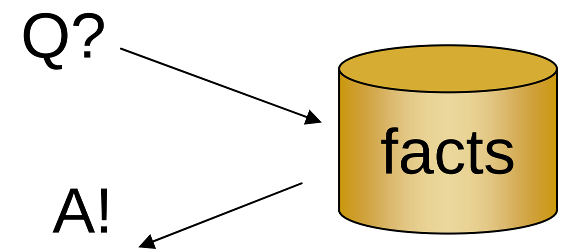
Predict height from model:

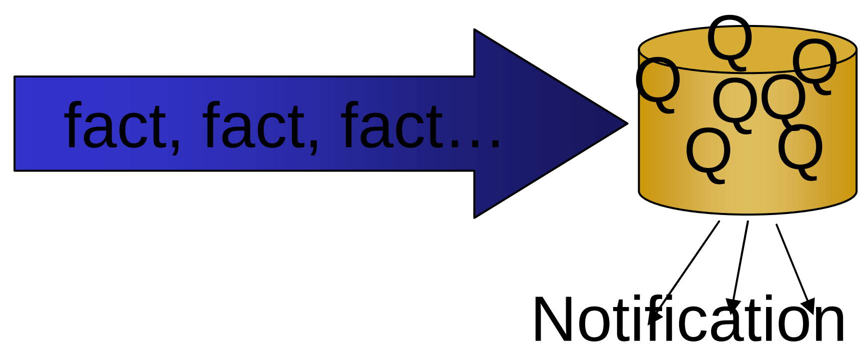
```
SELECT height,
          PredictProbability(height)
FROM Height PREDICTION JOIN New
          ON   New.Gender = Height.Gender
          AND New.Age       = Height.Age
```

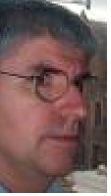
Probabilistic
Reasoning

Stream Processing and Sensor Processing

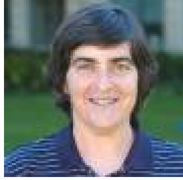
- Traditionally:
  Query billions of facts
- Streams:
  millions of queries one new fact
  - New protein compare to all DNA
  - Change in price or time
- Implications
  - New relational operators
  - New programming style
  - Streams in products:
    - Queries represented as records
    - New kind of query optimizer.
- Sensor networks
  - push queries out to sensors.
  - Simpler programming model
  - Optimizes power & bandwidth



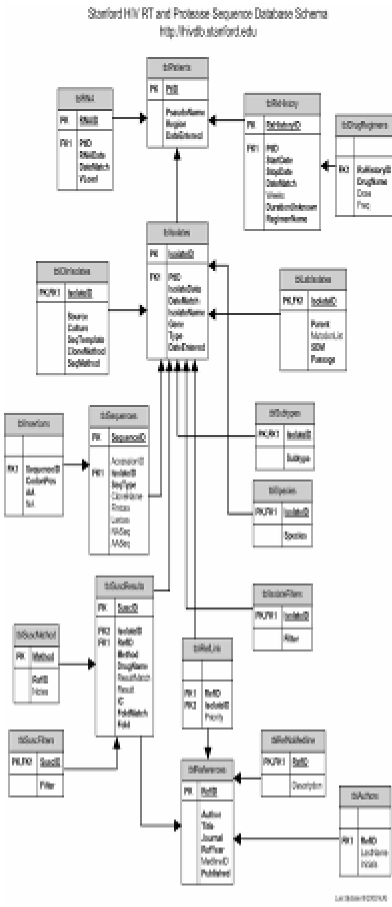


Semi-Structured Data

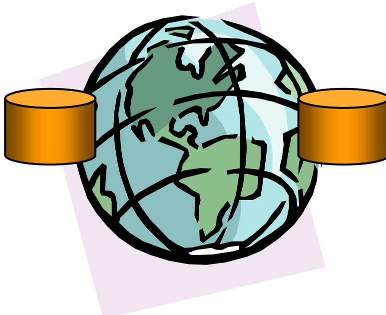
 

- *"Everyone starts with the same schema: <stuff/>."*
  Then they refine it." J. Widom
- We are a "strong schema" community
- That has pros-and-cons.
- Files <stuff/> and XML <<foo/> <bar/>>
  are here to stay. *Get over it!*
- File directories are becoming databases;
  - Pivot on any attribute
  - Folders are standing queries.
  - Freetext+schema search (better precision/recall)
- XSD (xml schema) and xQuery are transitional;
  But we have to do them to get to the real answer.
- Challenge: figure out what comes after XSD+xQuery



Publish-Subscribe, Replication Extract-Transform-Load (ETL)

- Data has many users
- Replicas for availability and/or performance (e.g. directories.)
- Mobile users do local updates synchronize replicas later.
- Classic Warehouse
  - Replicate to data warehouse
  - Data marts subscribe to publications
- Disaster Recovery wants geoplex
- Many different algorithms:
  - transactions, 1-safe, snapshot, merge, log ship,…
  - Each algorithm seems to be best for something.
- ETL has become a major application & component
  - Data loading
  - Data scrubbing
  - Publish/subscribe workflows.



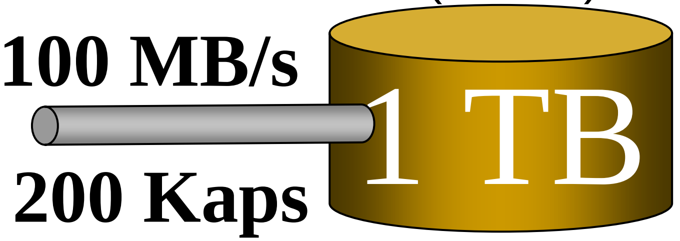
Late Binding in Query Plans

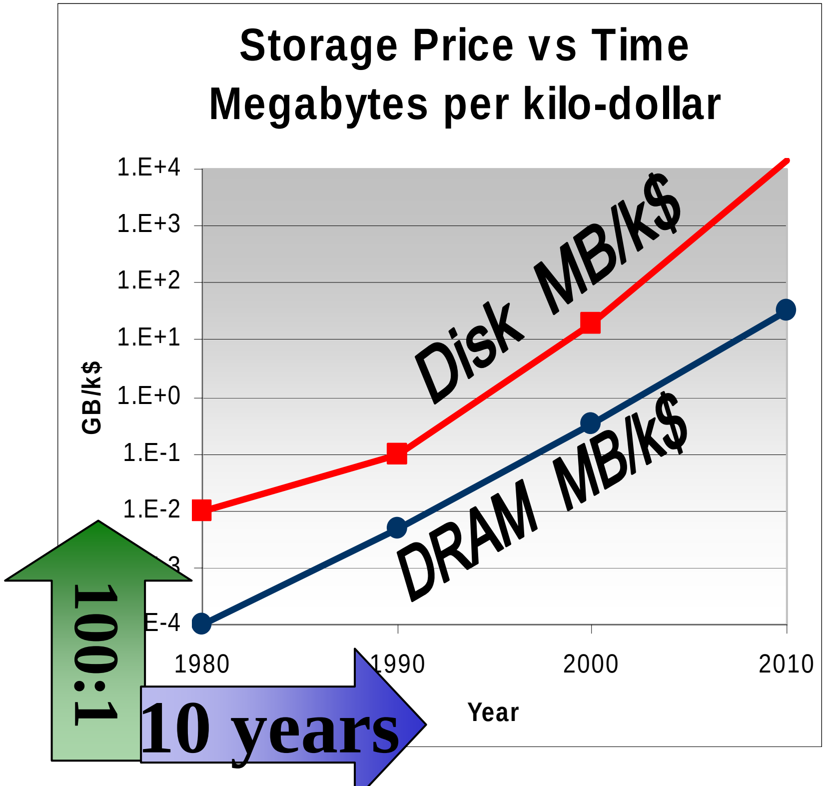

- Cost based query optimizers are great!
  when they guess right.
- But if it guessed 1 minute and the query has been running for a day…
- If system is busy plan is different
- Better strategy: Have query optimizer learn
  - from previous queries
  - From previous instances of this query
  - From this query
  - From environment.
- As a person who has waited days for a query to complete – I think this VERY important (!)

Massive Memory, Massive Latency

- RAM costs ~ 100k$...300k$/**TeraByte**
- 64 bit addressing everywhere
- Latency a problem
- NUMA latency a problem
- Checkpoint 1TB?
  Restart 1TB?
  Scan 1TB
- OK,
  now how about 100TB?
- Challenge: Algorithms for Massive Main Memory

the absurd disk is (almost) here





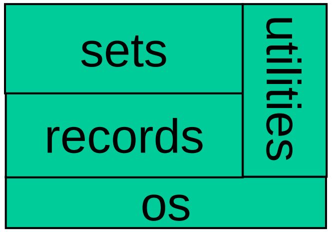
Smart Objects: Databases Everywhere

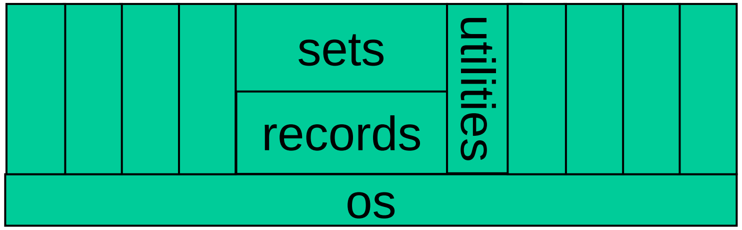
- Phones, PDAs, Cameras,… have small DBs.
- Disk drives have enough cpu, memory to run a full-blown DBMS.
- All these devices want-need to share data.
- They need an Esperanto.
- It is the DBMS ecosystem language.

Self Managing & Always Up

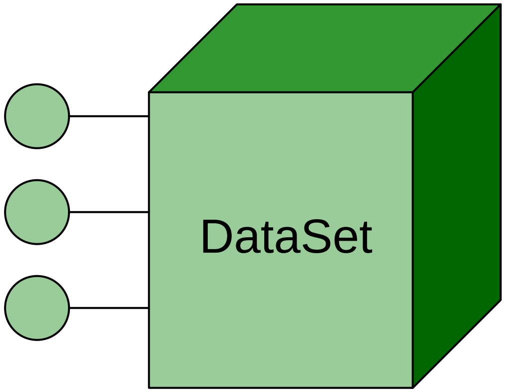
- People costs have always exceeded IT capital.
- But now that hardware is “free” …
- Self-managing self-configuring self-healing is key.
- Also self-organizing and
- No DBAs for cell phones or cameras.
- Requires a modular software architecture
  - Clear and simple knobs on modules
  - Software manages these knobs

Restatement: DB Systems evolved to be **containers for information services** develop, deploy, and execution environment





- DBMS is an ecosystem
  Key structuring strategy:
  – Everything is a class
  – Database is a complex object
  – Core object is DataSet
- This architecture uses many of your ideas
- The architecture lets you add your new ideas.



The Lowell Database Research Meeting， Lowell Massachusetts, 4-6 May 2003

- This meeting focuses on :
  - information storage, organization, management, and access and it is driven by new applications, technology trends, new synergies with related fields, and innovation within the field itself

From: The Lowell Database Research Self Assessment，Lowell, 2003

The Lowell Database Research Meeting

- Attendees at the Lowell Workshop were:
- Serge Abiteboul, Rakesh Agrawal, Phil Bernstein, Mike Carey, Stefano Ceri, Bruce Croft, David DeWitt, Mike Franklin, Hector Garcia Molina, Dieter Gawlick, Jim Gray, Laura Haas, Alon Halevy, Joe Hellerstein, Yannis Ioannidis, Martin Kersten, Michael Pazzani, Mike Lesk, David Maier, Jeff Naughton, Hans Schek, Timos Sellis, Avi Silberschatz, Mike Stonebraker, Rick Snodgrass, Jeff Ullman, Gerhard Weikum, Jennifer Widom, and Stan Zdonik. Slides and some detailed notes from the event are at http://research.microsoft.com/~gray/lowell/.

From: The Lowell Database Research Self Assessment , Lowell, 2003

Topics(1) Agenda

- **How are IR and structured data going to get together**?
  - Not clear how to do web crawling of structured data. Not clear how two communities can leverage each other. Where do we go from here? Is semi-structured data the answer?
- **Infoglut**
  - What do we do about the overwhelming amount of information that shows up on our desktop. How do we pick a needle out of a haystack? Is MyGoogle the answer? Other standards on the horizon? Super-duper UDDI?
- **What is the future of XML?**
  - This is the year of XML. (VLDB last year has more than 10 XML papers among 70+ papers)
  - Will it be anything more than an intergalactic data interchange language? What about Xquery and XSD? Is there any research here?

From: The Lowell Database Research Self Assessment , Lowell, 2003

Topics(2)

- **Will federated databases ever go anywhere**?
  - Federated data bases have never gone anywhere. Moreover, they are currently used primarily as ETL tools. Will Liquid Data go anywhere? Is there any hope for dealing with semantic heterogeneity. Is there anything to peer-to-peer data bases that is not in federated data bases?
- **Data Mining**
  - Is there anything here, other than 2nd rate statistics? How do you answer the query "tell me something interesting, that I don't know already"?
- **Stream processing**
  - Is there any meat here? Why can't this whole area be done by traditional middleware? Why are the current proposals so complex? Does anybody really need quality of service. How do you support mixed environments where some data is transactional and some can be forgotten?

From: The Lowell Database Research Self Assessment , Lowell, 2003

Reports:
Next Generation Infrastructure

- Integration of Text, Data, Code, and Streams
- Information Fusion
- Sensor Data and Sensor Networks
- Multimedia Queries
- Reasoning about Uncertain Data
- Personalization
- Data Mining
- Self Adaptation
- Privacy
- New User Interfaces
- Trustworthy Systems
- One-Hundred-Year Storage
- Query Optimization

From: The Lowell Database Research Self Assessment , Lowell, 2003

对数据库技术发展的思考（1）

- 在成熟的关系DBMS之后，DBMS已经研究的没有问题了？
  - VLDB2000
    - 会议的主题是“Broadening the Database Field”
    - 会议的论文分为“core database technology”和“information systems infrastructures ”

对数据库技术发展的思考（2）

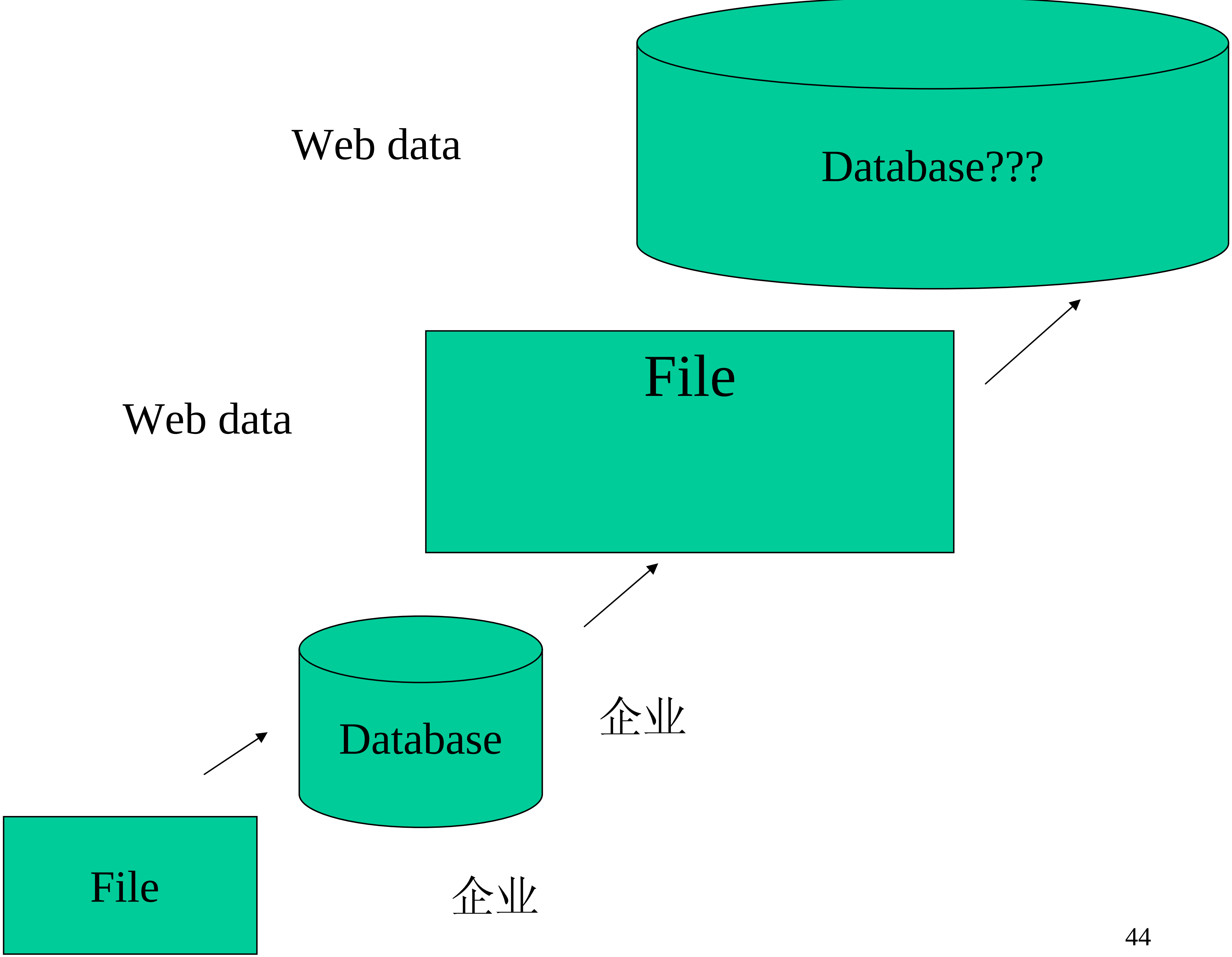
- 关注信息系统架构的创新性数据库管理问题
- 在Web大背景下新的处理要求在那里？
- “泛数据”研究
  - X-data: XML data, streaming data, …
  - X-computing: grid data, sensor data, p2p data, ubiquitous/pervasive data, …

对数据库技术发展的思考（3）

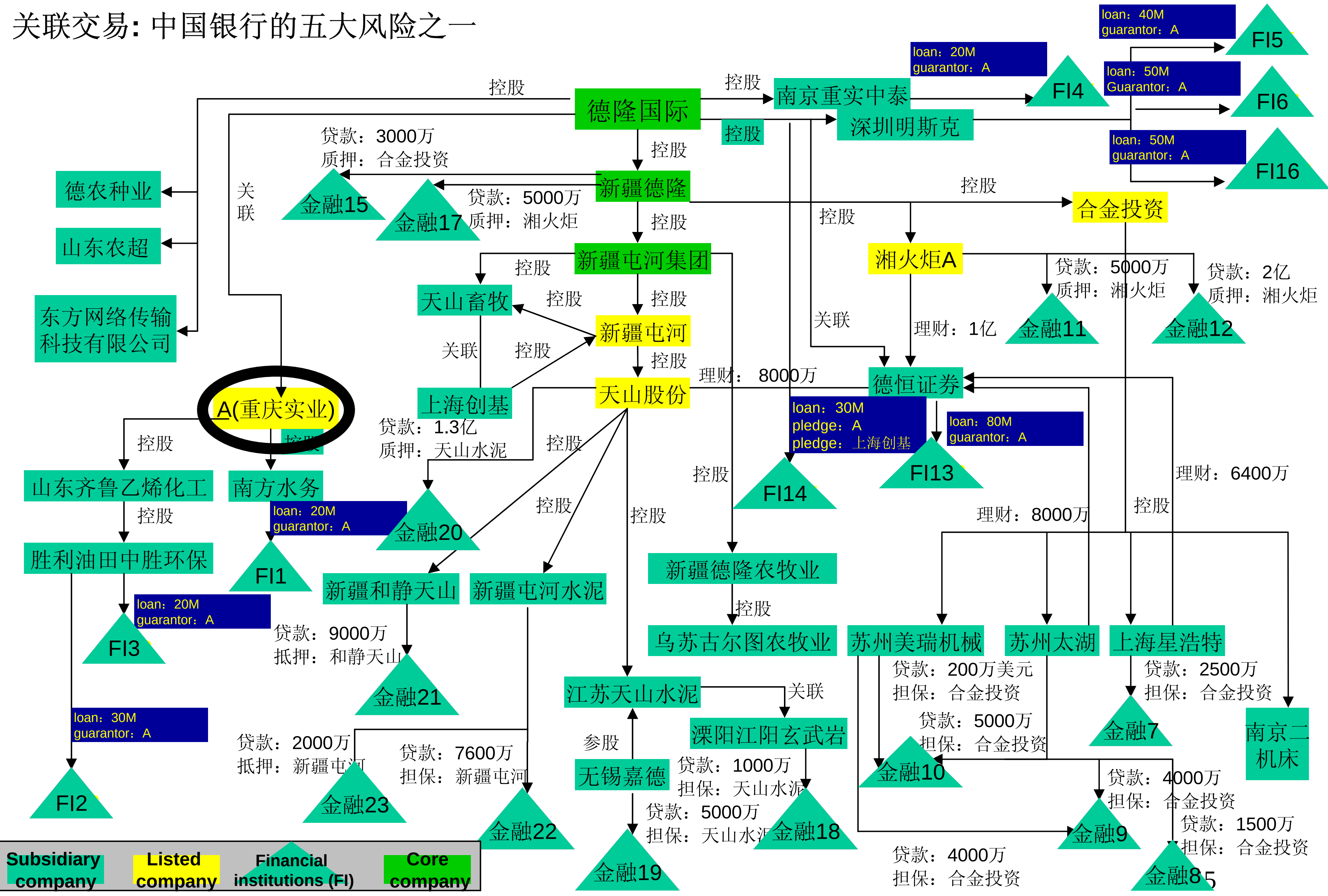
- 追求原始创新技术
  - 高水平的原型系统
  - 高质量的论文成果
- 如何寻找原始创新
  - 现实的应用需求
  - 客观、真实的问题描述
  - 。。。。
- 我们有没有找准、定义清楚我们的问题？

对数据库技术发展的思考（4）

- 什么是现实的应用需求？
  - 以数据库为例：
    - 企业数据的高效组织管理---------DBMS
  - Database System vs. File System
  - 现实一个令人震惊的事件
    - 关联交易----德隆事件
  - 现实的E-Catalog问题

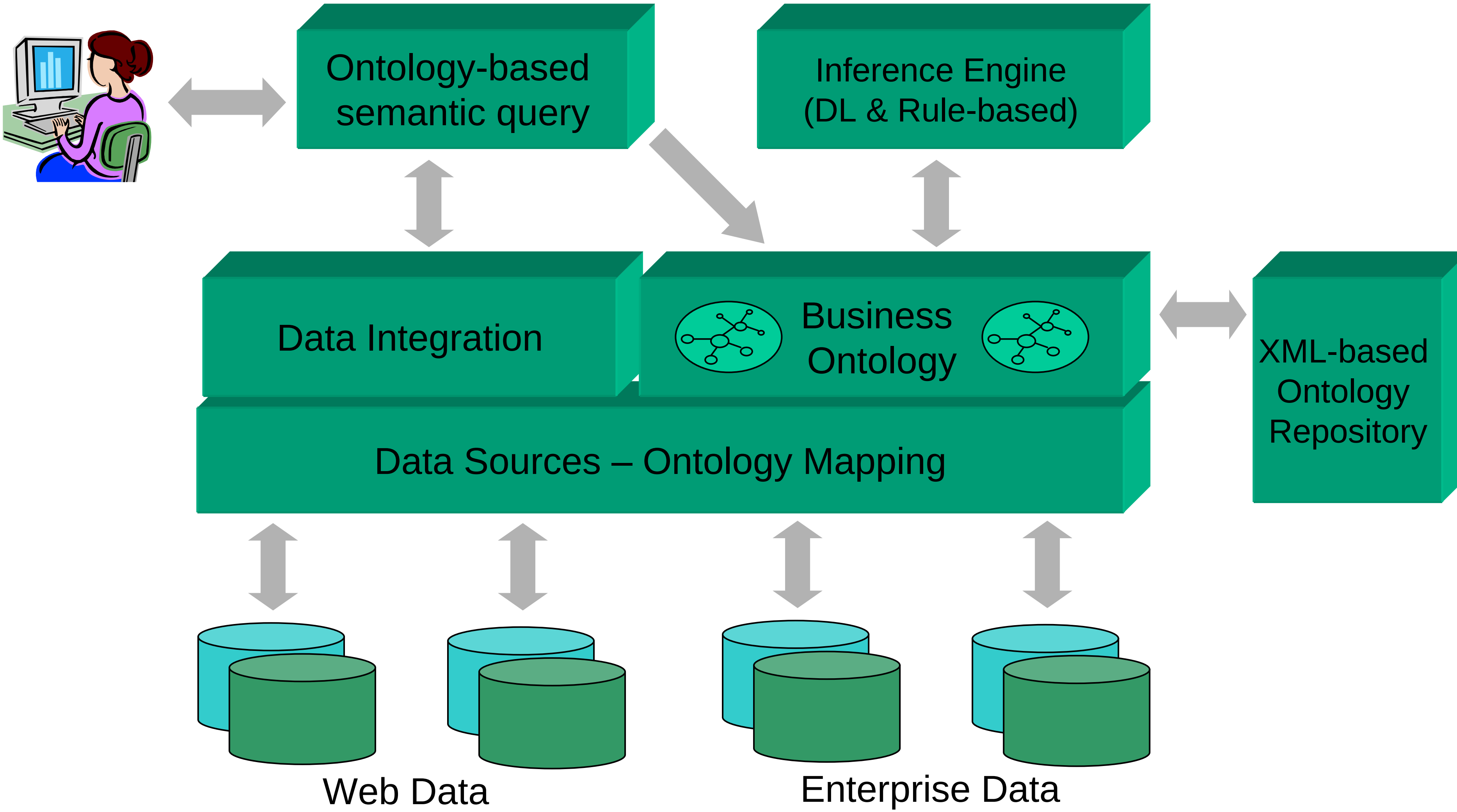

Web data
Database???
File
Web data
Database
企业
File
企业

关联交易: 中国银行的五大风险之一



来源: *财经*. No.12, 2004

Architecture



What is E-Catalog?

- An e-catalog usually contains the descriptions for many products
- Each product has its own set of attributes
  - T-shirt: size, style, color, price
  - TV set: brand, view-type, signal-type, screen-size, price
- The total number of attributes across all products can be huge
- E-catalog should efficiently support users to search for products of interest via constraints on attributes
  - Find all the OIDs of T-shirts with size='M' and price<$25

Schemas for E-catalog

- Horizontal Schema: one big "fat" table *H(OID, A1, A2, ..., An)*
  - Conceptually easy
  - Too many columns: impossible for real RDBMS
  - Very sparse: a lot of null values, resulting in poor query processing
  - High processing and maintenance cost for product changes
- Binary Schema: each attribute corresponds to one table *Bi(OID, Ai)*
  - Dense
  - A lot of joins are involved in search query: poor query performance

Horizontal Schema

OID	A1	A2	A3	A4	A5
1	v1	v2	v3		v4
2	v5	v6	v7		v8
3	v9	v10			v11
4			v12	v13	v14
5			v15		v16
6				v17	v18

Binary Schema

OID	A1
1	v1
2	v5
3	v9

OID	A2
1	v2
2	v6
3	v10

OID	A3
1	v3
2	v7
4	v12
5	v15

OID	A4
4	v13
6	v17

OID	A5
1	v4
2	v8
3	v11
4	v14
5	v16
6	v18

Vertical Schema

OID	AttrName	Value
1	A1	v1
1	A2	v2
1	A3	v3
1	A5	v4
2	A1	v5
2	A2	v6
2	A3	v7
2	A5	v8
3	A1	v9
3	A2	v10
3	A5	v11
4	A3	v12
4	A4	v13
4	A5	v14
5	A3	v15
5	A5	v16
6	A4	v17
6	A5	v18

Schemas for E-catalog *(continued)*

- Vertical Schema: one big "skinny" table
  V(OID, attribute_name, attribute_value)

This is the schema used in many commercial e-commerce systems!

  - Advantages
    - High Flexibility
    - Ease of schema evolution
    - Low storage overhead (dense)
  - Disadvantages
    - Writing SQL queries against V is cumbersome
    - A lot of joins are involved in search queries: query performance is no better than binary schema

Typical E-catalog Query: Parametric Search (Search Products via Constraints)

```
SELECT OID
FROM H
WHERE (Ai1 not null)  AND (bound on Ai1) AND
      (Ai2 not null)  AND (bound on Ai2) AND
                 .....
      (Aik not null)  AND (bound on Aik)
```

Related Work

- Agrawal et al. [VLDB 2001]
  - creating a logical horizontal view on top of the vertical schema
  - query rewrite algorithms to convert relational algebra operators against the horizontal view to that against the vertical table
  - queries against vertical schema performs NO BETTER than against binary schema in most cases

Parametric Search Against Vertical Schema: Why Is It So Slow?

- The key reason: the cost-based optimizer of current RDBMS is not designed for vertical schema.
- The statistics (histogram information) is misleading when using vertical schema: it contains aggregated statistics of heterogeneous attributes from different product categories.

```
SELECT OID
FROM T-shirt
WHERE size='M' AND
      color='Purple' AND
      $45<price<$50
```

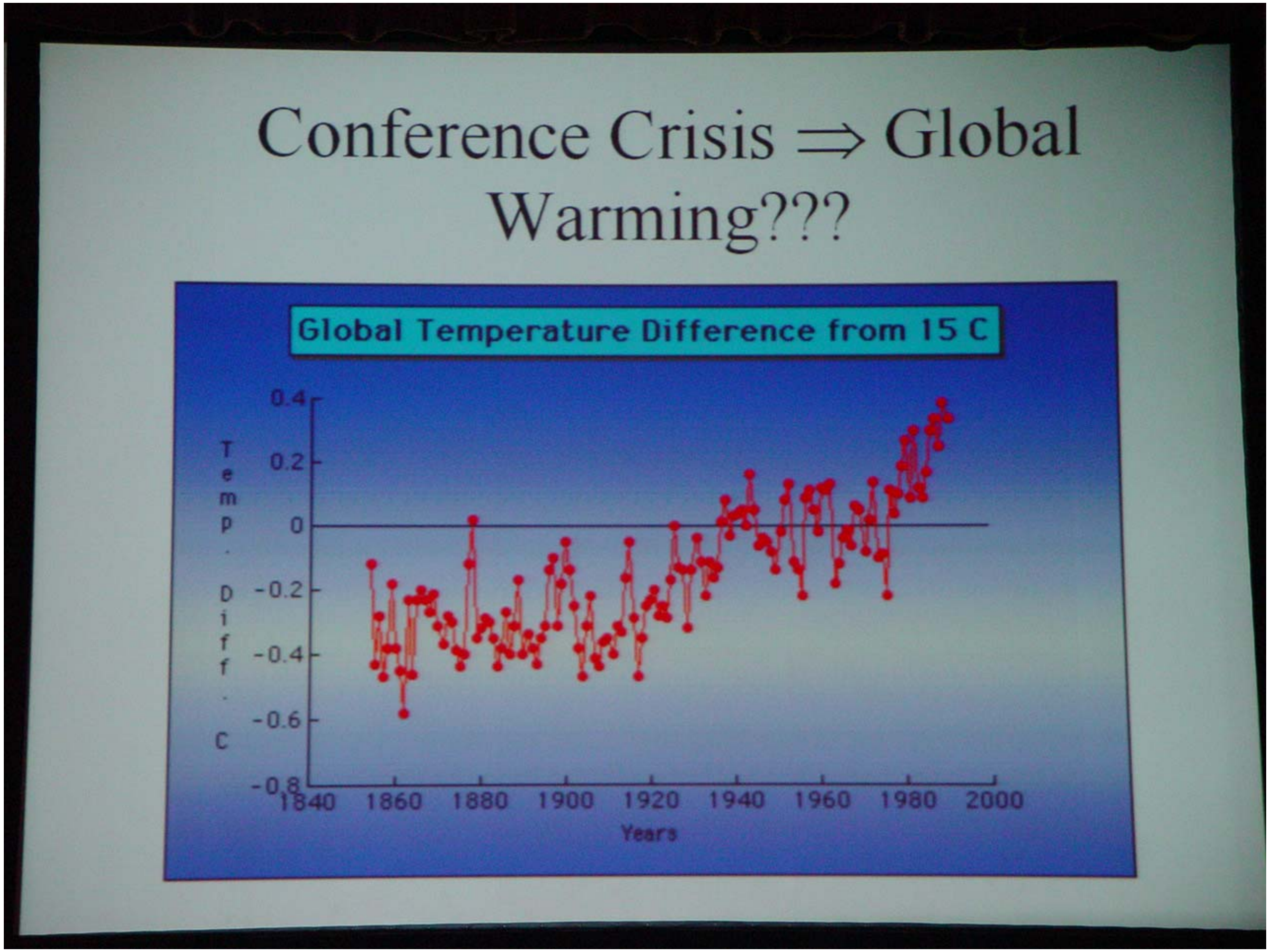
Q1

```
SELECT OID
FROM V
WHERE AttrName='size' AND Value='M'
INTERSECT
SELECT OID
FROM V
WHERE AttrName='color' AND Value='Purple'
INTERSECT
SELECT OID
FROM V
WHERE AttrName='price' AND $45<Value<$50
```

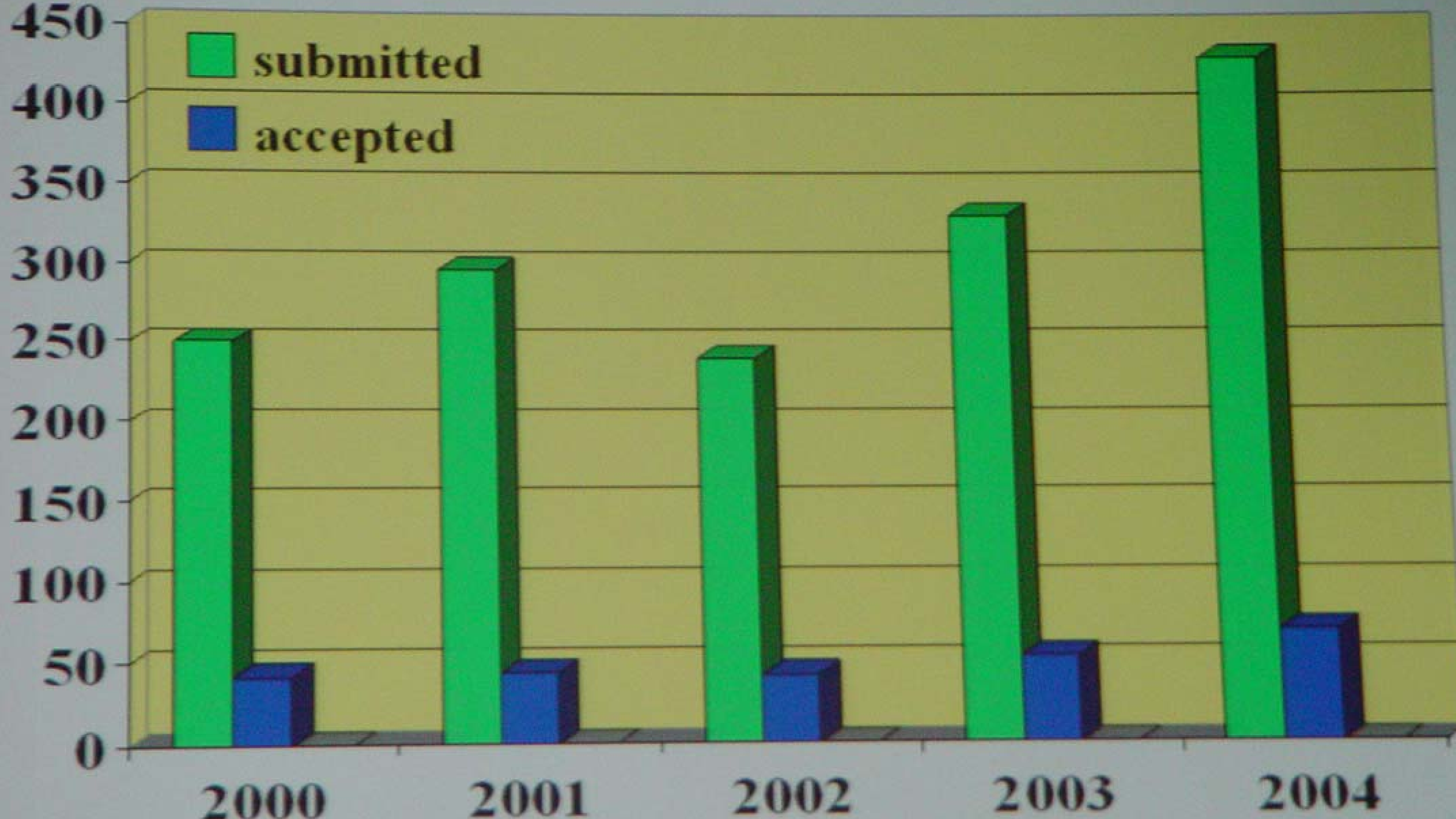
Q2

数据库的学术研究

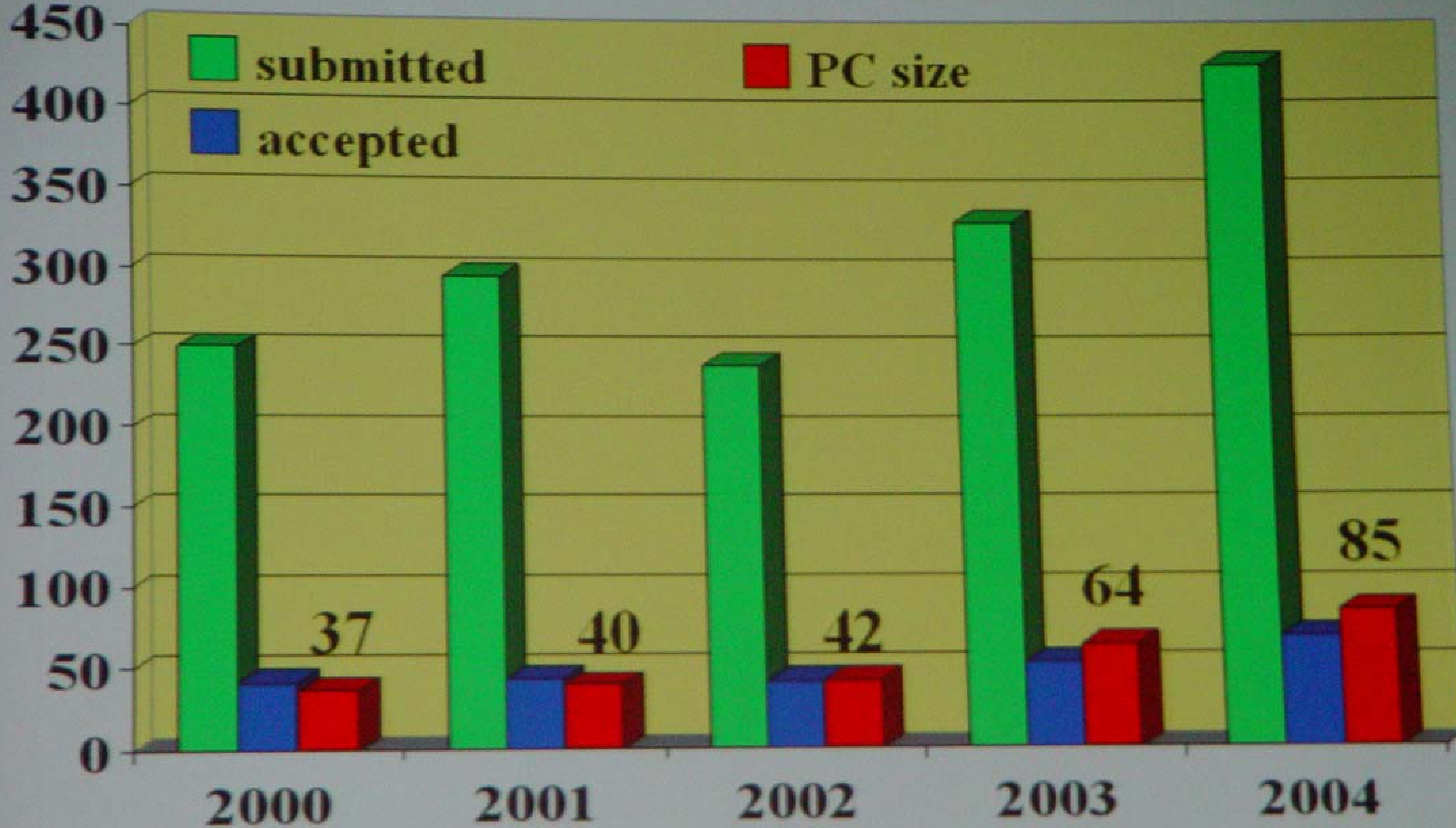
- 学术会议
  - VLDB, SIGMOD/PODS, ICDE
  - EDBT, DASFAA, KDD
  - WebDB, WIDM, MDM, SSDBM
  - WAIM, APWeb, PAKDD
  - NDBC, ADB(澳大利亚), 巴西,英国,印度
- 一个非传统的会议
  - CIDR: Conference on Innovative Data Systems Research


Conference Crisis ⇒ Global Warming???
Global Temperature Difference from 15 C
Temp. Diff. C
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
1840
1860
1880
1900
1920
1940
1960
1980
2000
Years

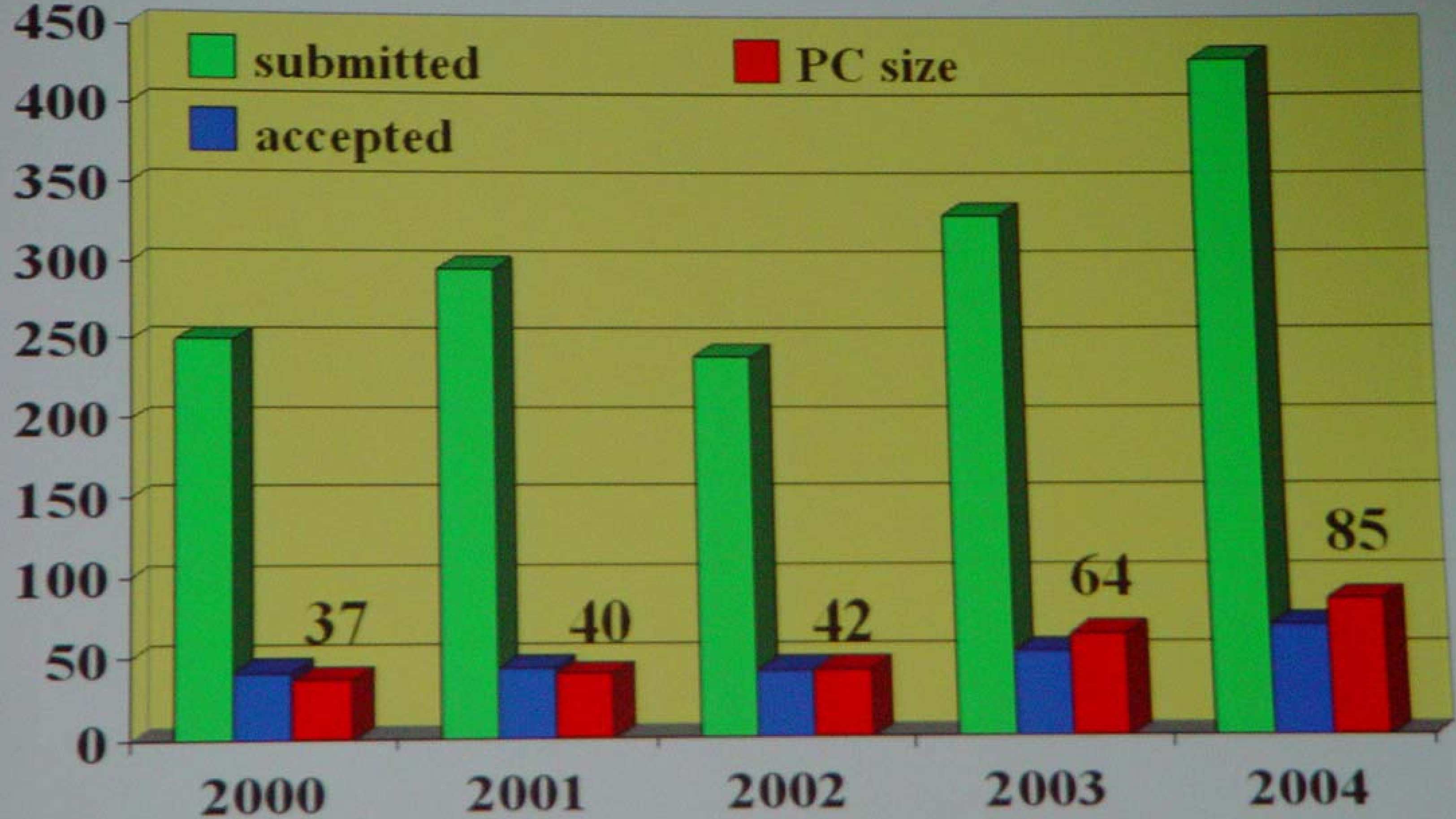
Trends & Challenges



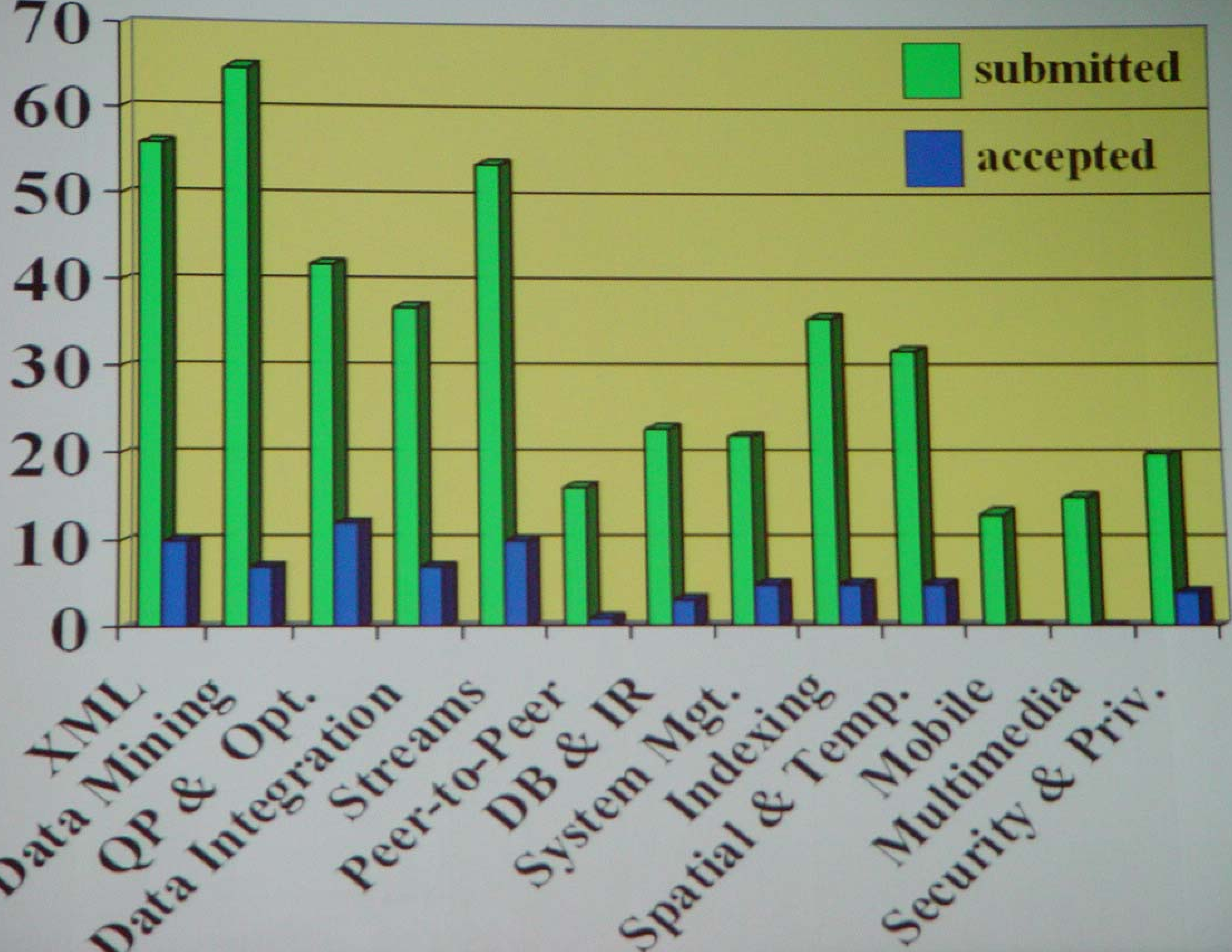
Trends & Challenges



Trends & Challenges



Topic Statistics 2004



Long-Term Statistics (1)

Most frequent keywords in conference program

1990 Atlantic City	query, object, rule
1995 San Jose	query, object, parallel
1998 Seattle	query, mining, similarity
1999 Philadelphia	query, clustering, efficient
2000 Dallas	mining, query, index
2001 Santa Barbara	query, efficient, XML
2002 Madison	query, XML, caching
2003 San Diego	XML, query, stream

e: 97.24 % of all statistics are made up

Long-Term Statistics (1)

Most frequent keywords in conference program

1990 Atlantic City	**query, object, rule**
1995 San Jose	**query, object, parallel**
1998 Seattle	**query, mining, similarity**
1999 Philadelphia	**query, clustering, efficient**
2000 Dallas	**mining, query, index**
2001 Santa Barbara	**query, efficient, XML**
2002 Madison	**query, XML, caching**
2003 San Diego	**XML, query, stream**
2004 Paris	**query, XML, efficient**

Note: 97.24 % of all statistics are made up

References

- Philip A. Bernstein, Umeshwar Dayal, David J. DeWitt, Dieter Gawlick, Jim Gray, Matthias Jarke, Bruce G. Lindsay, Pete C. Lockemann, David Maier, Erich J. Neuhold, Andreas Reuter, Lawrence A. Rowe, Hans-Jörg Schek, Joachim W. Schmidt, Michael Schrefl, and Michael Stonebraker: Future Directions in DBMS Research - The Laguna Beach Participants. SIGMOD Record 18(1): 17-26 (1989)
- Abraham Silberschatz, Michael Stonebraker, and Jeffrey D. Ullman: Database Systems: Achievements and Opportunities. CACM 34(10): 110-120 (1991)
- Abraham Silberschatz, Michael Stonebraker, and Jeffrey D. Ullman: Database Research; Achievements and Opportunities into the 21st Century. SIGMOD Record 25(1): 52-63 (1996)
- Abraham Silberschatz, Stanley B. Zdonik, et al: Strategic Directions in Database Systems — Breaking Out of the Box. ACM Computing Surveys 28(4): 764-778 (Dec. 1996).
- Philip A. Bernstein, Michael L. Brodie, Stefano Ceri, David J. DeWitt, Michael J. Franklin, Hector Garcia-Molina, Jim Gray, Gerald Held, Joseph M. Hellerstein, H. V. Jagadish, Michael Lesk, David Maier, Jeffrey F. Naughton, Hamid Pirahesh, Michael Stonebraker, and Jeffrey D. Ullman: The Asilomar Report on Database Research. SIGMOD Record 27(4): 74-80 (1998)