ALTERNATE MODELS AND SPECIALIZED APPLICATIONS

COMP163: Database Management Systems

ALTERNATE MODELS

- Active Databases
- Temporal and Spatial Database / GIS
- Deductive Databases
- Distributed Databases
- XML
- NoSQL

SPECIALIZED APPLICATIONS

- Big Data / Cloud
- Information Retrieval / Web Search
- Data Mining
- Data Warehousing / OLAP
- Mobile Databases
- Multimedia Databases
- Genome Data Management

ALTERNATE MODELS

ACTIVE DATABASES

- Active Behavior = ability to react to events
- dimensions:
  - event
  - condition
  - action
  - execution model
  - management
- Application Areas
  - composite objects
  - integrity constraints, business rules
  - derived data

TEMPORAL DATABASES

- A **temporal database** is a database management system with built-in time aspects, e.g. a temporal data model and a temporal version of structured query language.
- More specifically the temporal aspects usually include valid-time and transaction-time. These attributes go together to form bitemporal data.
- ***Valid time*** denotes the time period during which a fact is true with respect to the real world.
- ***Transaction time*** is the time period during which a fact is stored in the database.

TUPLES WITH VALID TIME FRAMES

EMP_VT

Name	Ssn	Salary	Dno	Supervisor_ssn	Vst	Vet
Smith	123456789	25000	5	333445555	2002-06-15	2003-05-31
Smith	123456789	30000	5	333445555	2003-06-01	Now
Wong	333445555	25000	4	999887777	1999-08-20	2001-01-31
Wong	333445555	30000	5	999887777	2001-02-01	2002-03-31
Wong	333445555	40000	5	888665555	2002-04-01	Now
Brown	222447777	28000	4	999887777	2001-05-01	2002-08-10
Narayan	666884444	38000	5	333445555	2003-08-01	Now

. . .

DEPT_VT

Dname	Dno	Manager_ssn	Vst	Vet
Research	5	888665555	2001-09-20	2002-03-31
Research	5	333445555	2002-04-01	Now

VALID TIME & TRANSACTION TIME

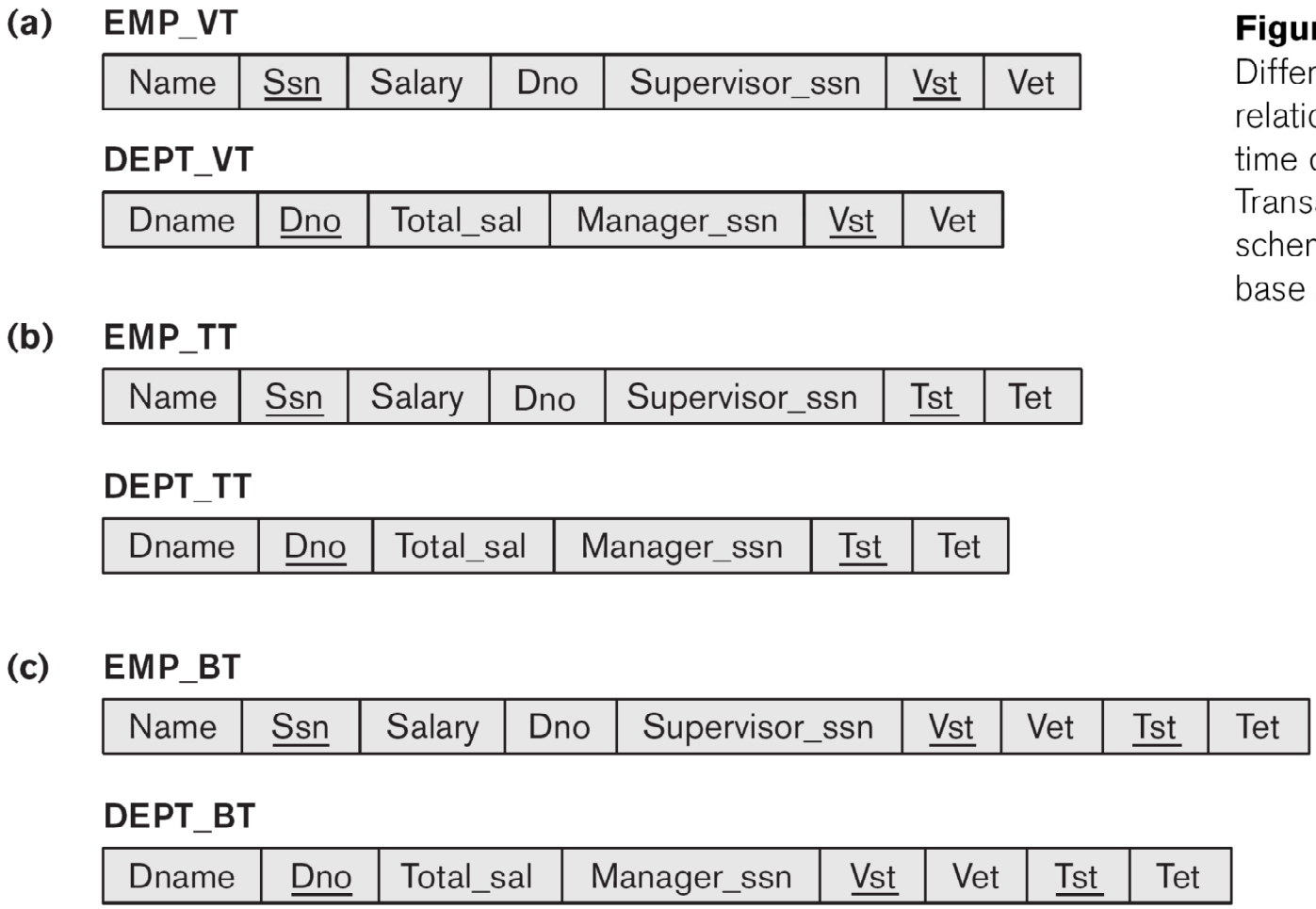


Figure 26.7
Different types of temporal relational databases. (a) Valid time database schema. (b) Transaction time database schema. (c) Bitemporal database schema.

SPATIAL DATABASES

- offers *spatial data types*
- supports *spatial indexing*
  - find all cities in Bavaria
- supports *spatial joins*
  - for each river, find all cities within 50 KM
- managing space → large collections of simple geometric objects
- 2D: geography (GIS), VLSI design
- 3D: astronomy, brain maps, molecules

SPATIAL DATABASE INDEXES

- **R-trees**
  - Technique for typical spatial queries
  - Group objects close in spatial proximity on the same leaf nodes of a tree structured index
  - Internal nodes define areas (rectangles) that cover all areas of the rectangles in its subtree.
- **Quad trees**
  - Divide subspaces into equally sized areas

GIS

- Specialization of spatial database systems
  - also has temporal aspects
  - highly dependent on complex range queries

DEDUCTIVE DATABASES

- logic programming + persistence
- prolog → datalog
- facts and rules
- inference engine

DATALOG – RULE AND QUERIES

(a)

Facts

SUPERVISE(franklin, john).
SUPERVISE(franklin, ramesh).
SUPERVISE(franklin, joyce).
SUPERVISE(jennifer, alicia).
SUPERVISE(jennifer, ahmad).
SUPERVISE(james, franklin).
SUPERVISE(james, jennifer).
. . .

Rules

SUPERIOR(*X, Y*) :– SUPERVISE(*X, Y*).
SUPERIOR(*X, Y*) :– SUPERVISE(*X, Z*), SUPERIOR(*Z, Y*).
SUBORDINATE(*X, Y*) :– SUPERIOR(*Y, X*).

Queries

SUPERIOR(james, *Y*)?
SUPERIOR(james, joyce)?

(b)

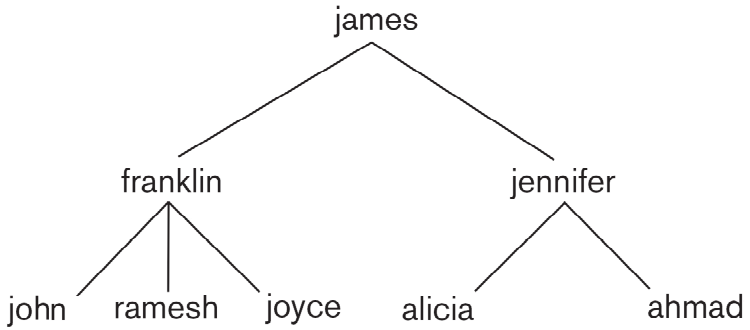


Figure 26.11
(a) Prolog notation.
(b) The supervisory tree.

DISTRIBUTED DATABASES

- data and processing (server) reside on multiple computers
  - transparent replication/distribution
  - increased reliability and availability
  - improved performance
  - easier expansion
- federated database system
  - shared global schema
- multidatabase system
  - interactively constructs shared schema

DISTRIBUTED TRANSACTIONS

- ensuring transaction properties is difficult, since multiple machines are involved
- 2 phase commit:
  - phase 1:
    - coordinator sends “prepare to commit” to all participants
    - participants force-write all logs
    - participants indicate “ready to commit” or “cannot commit”
  - phase 2:
    - if all participants are ready to commit, coordinator sends commit command
    - if any participant cannot commit, coordinator sends abort command

STRUCTURED, SEMI-STRUCTURED, UNSTRUCTURED DATA

- structured data: fits a predefined format
  - relation schema
  - Java class
- semi-structured data: structure is flexible, but can be described at any particular time
  - structure is embedded in the data
  - aka self-describing data
- unstructured data: no discernable structure
  - raw text

XML

- XML has become popular for dealing with semi-structured data
- creates a hierarchical structure
- schema may be embedded with data or separate
- allows for dynamic databases where structure changes more quickly than can be handled with schema modifications
- persistence and queries can be specialized for XML structures
  - XPATH, XQUERY

SEMI-STRUCTURE DATA

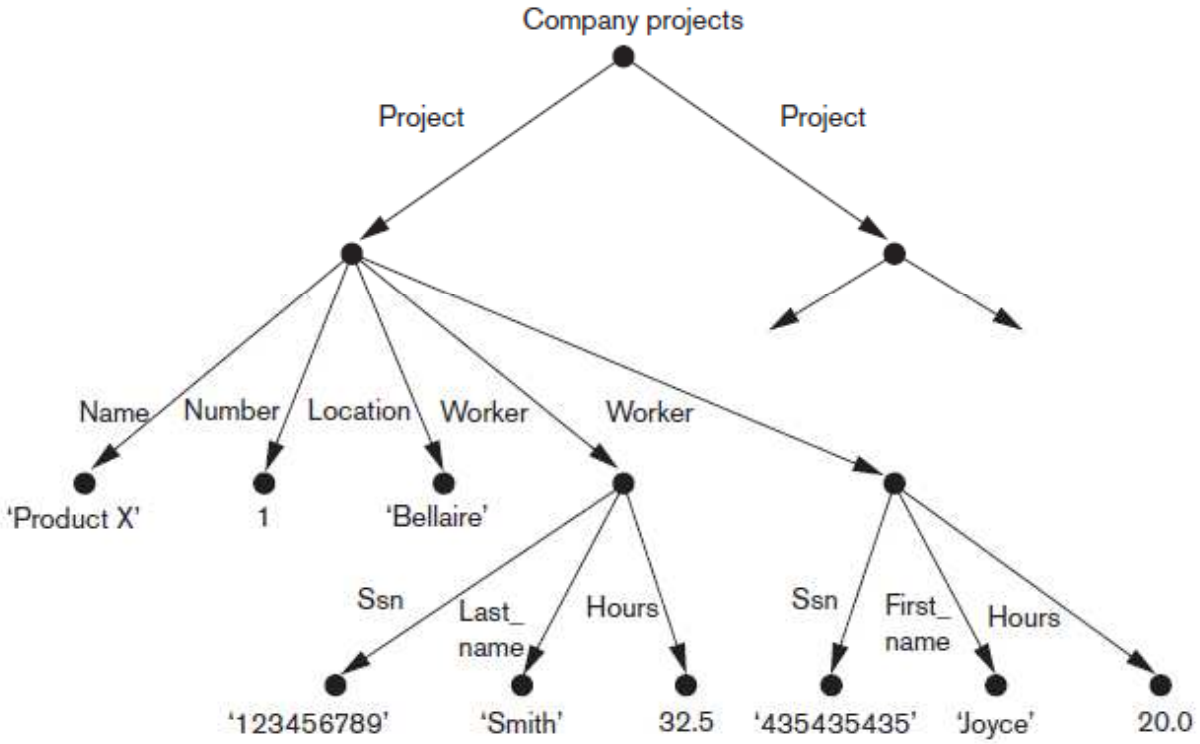


Figure 12.1
Representing semistructured data as a graph.

XML REPRESENTATION

```
<?xml version="1.0" standalone="yes"?>
<projects>

 <project>
  <Name>ProductX</Name>
  <Number>1</Number>
  <Location>Bellaire</Location>
  <DeptNo>5</DeptNo>
  <Worker>
   <SSN>123456789</SSN>
   <LastName>Smith</LastName>
   <hours>32.5</hours>
  </Worker>
  <Worker>
   <SSN>453453453</SSN>
   <FirstName>Joyce</FirstName>
   <hours>20.0</hours>
  </Worker>
 </project>
```

```
 <project>
  <Name>ProductY</Name>
  <Number>2</Number>
  <Location>Sugarland</Location>
  <DeptNo >5</DeptNo >
  <Worker>
   <SSN>123456789</SSN>
   <hours>7.5</hours>
  </Worker>
  <Worker>
   <SSN>453453453</SSN>
   <hours>20.0</hours>
  </Worker>
  <Worker>
   <SSN>333445555</SSN>
   <hours>10.0</hours>
  </Worker>
 </project>

...

</projects>
```

NO SQL

- designed for large, distributed data sets
- primarily read-only
- limited ACID requirements
  - http://www.techrepublic.com/blog/10things/10-things-you-should-know-about-nosql-databases/1772
  - http://en.wikipedia.org/wiki/NoSQL
  - http://gigaom.com/cloud/facebook-trapped-in-mysql-fate-worse-than-death/
  - http://gigaom.com/cloud/facebook-shares-some-secrets-on-making-mysql-scale/

SPECIALIZED APPLICATIONS

BIG DATA

- Past: difficult to find sufficient data to for analysis / problem solving
- Future: more data and information than we can manage
- Big data leads to new ways to investigate problems
  - old way: carefully design experiments to collect data to support/refute hypothesis
  - new way: throw more data at the problem
- White House Big Data initiative:
  - http://www.whitehouse.gov/sites/default/files/microsites/ostp/big_data_press_release.pdf

CLOUD

- from perspective of user/owner of data:
  - data's location is unknown
  - maintenance of system is provided service
  - hardware is irrelevant

INFORMATION RETRIEVAL

Table 27.1 A Comparison of Databases and IR Systems

Databases	IR Systems
■ Structured data	■ Unstructured data
■ Schema driven	■ No fixed schema; various data models (e.g., vector space model)
■ Relational (or object, hierarchical, and network) model is predominant	■ Free-form query models
■ Structured query model	■ Rich data operations
■ Rich metadata operations	■ Search request returns list or pointers to documents
■ Query returns data	■ Results are based on approximate matching and measures of effectiveness (may be imprecise and ranked)
■ Results are based on exact matching (always correct)	

GENERIC IR PIPELINE

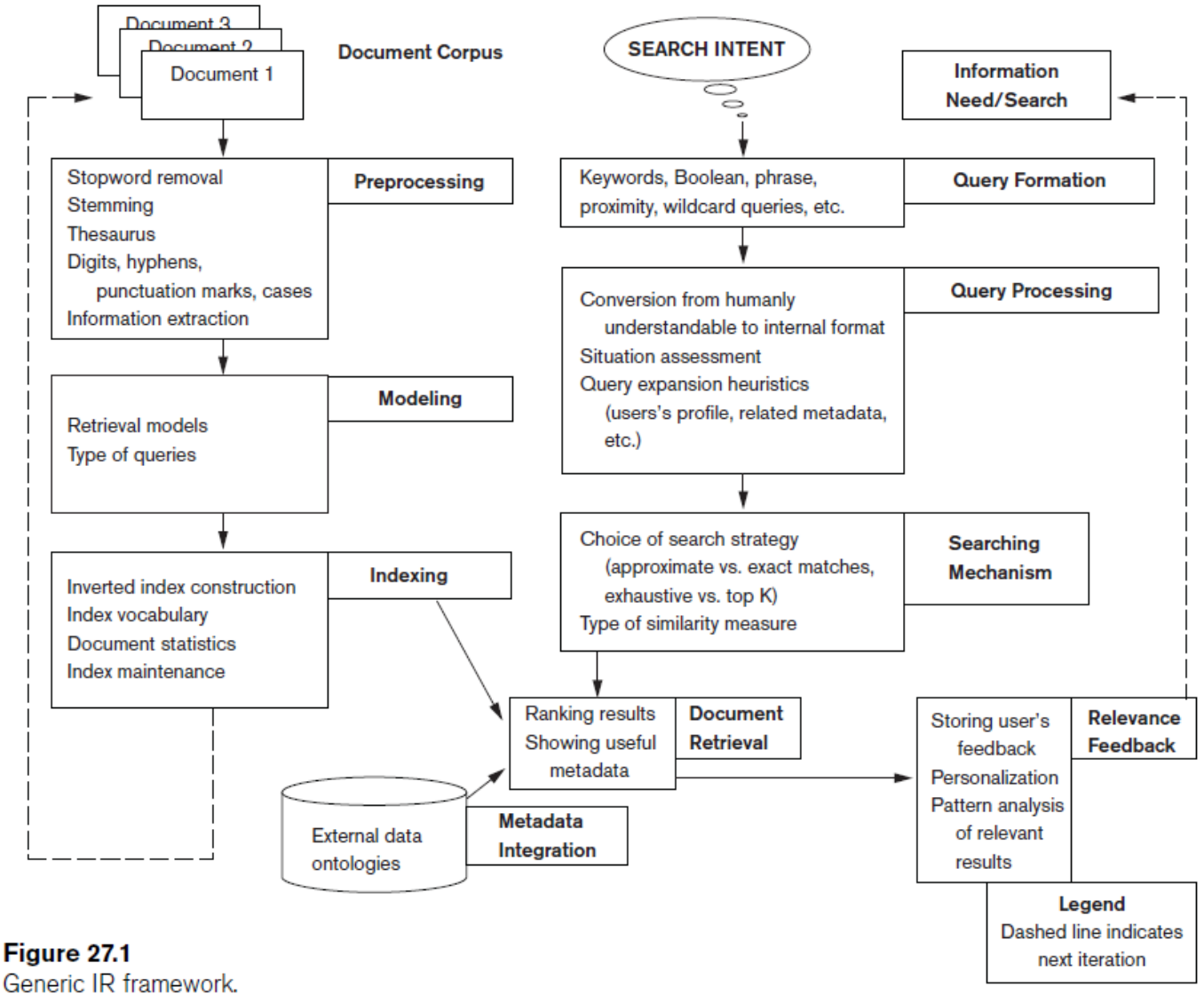


Figure 27.1
Generic IR framework.

GENERIC IR PIPELINE (CONT'D.)

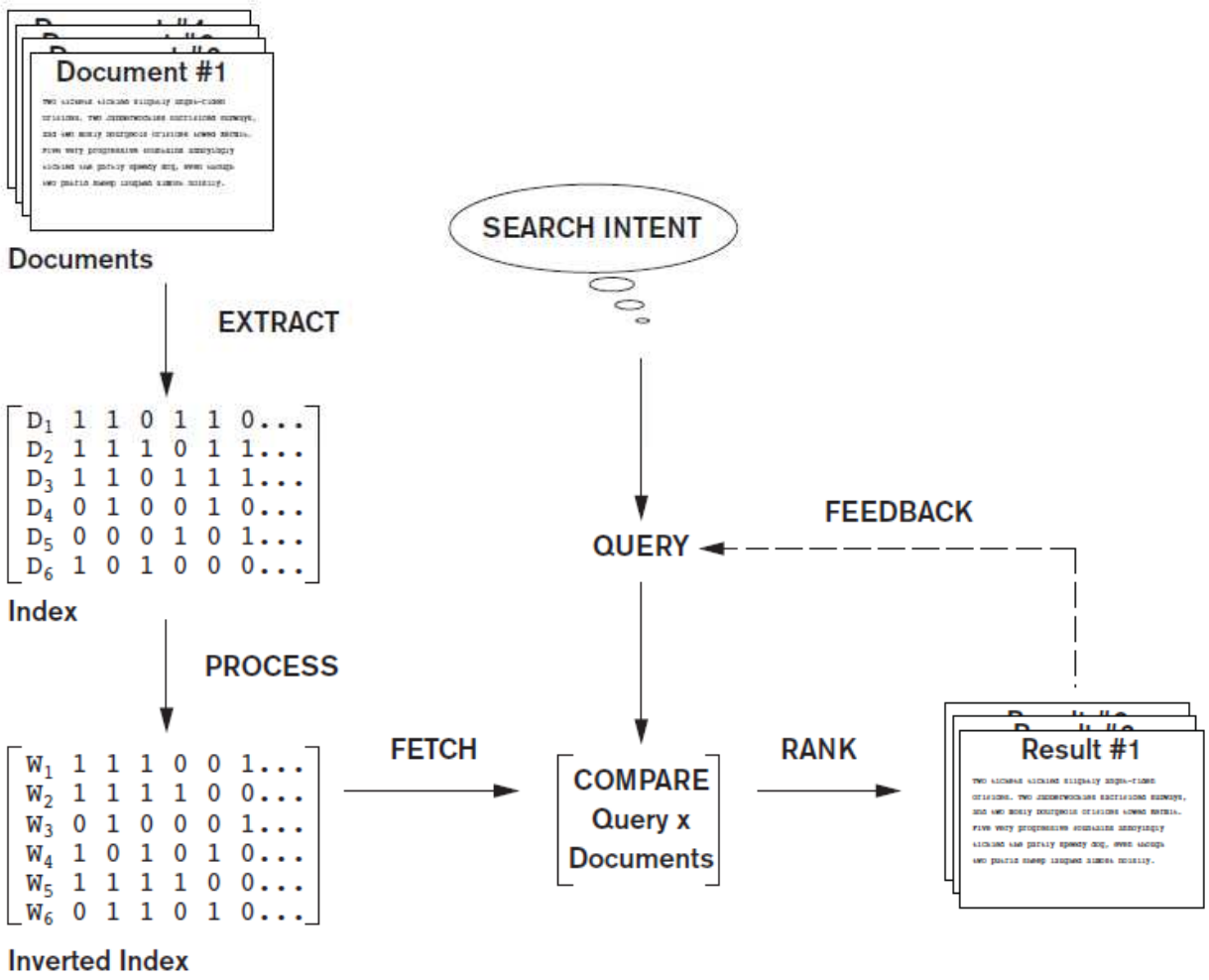


Figure 27.2
Simplified IR process pipeline.

DATA MINING

- KDD: Knowledge Discovery in Databases
  - 1. data selection
  - 2. data cleansing
  - 3. enrichment
  - 4. data transformation or encoding
  - 5. data mining
  - 6. reporting

DATA MINING

- Mining discovers
  - association rules
  - sequential patterns
  - classification hierarchies
  - patterns within time series
  - clustering
- Goals
  - prediction
  - identification
  - classification
  - optimization
- Methods
  - Sequential pattern analysis
  - Time Series Analysis
  - Regression
  - Neural Networks
  - Genetic Algorithms

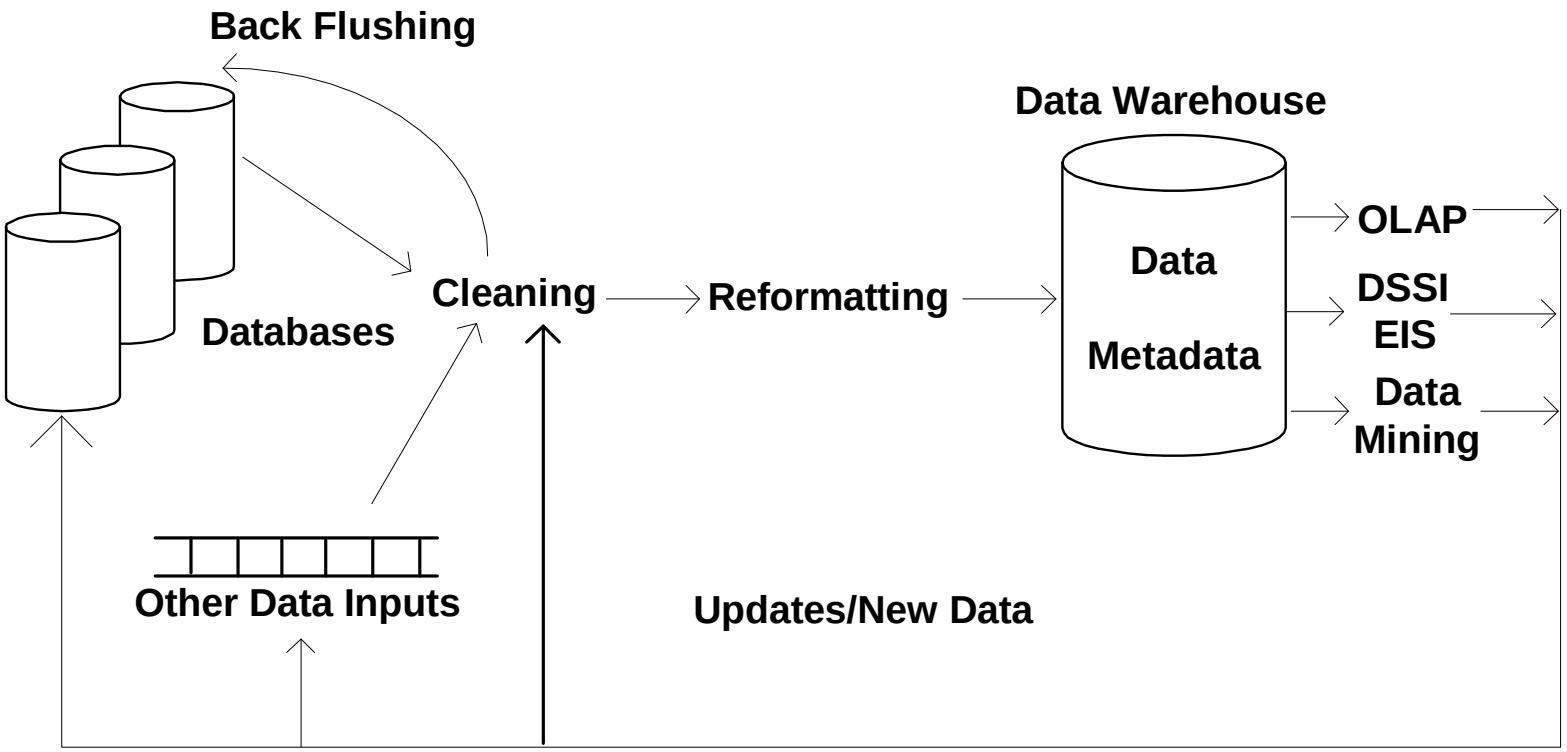
DATA WAREHOUSING

- Data Warehouses are
  - very large
  - multi-dimensional (i.e. temporal)
  - not transactional
  - the result of KDD step 4

OLAP

- Online Analytic Processing
  - analysis of complex data from a data warehouse
  - makes use of the multi-dimensional structure of the warehouse
  - provides the tools to knowledge workers

CONCEPTUAL STRUCTURE OF DATA WAREHOUSE



WAREHOUSE V.TRADITIONAL DB

- Data Warehouses are mainly optimized for appropriate data access.
  - Traditional databases are transactional and are optimized for both access mechanisms and integrity assurance measures.
- Data warehouses emphasize more on historical data as their main purpose is to support time-series and trend analysis.
- Compared with transactional databases, data warehouses are *nonvolatile*.
- In transactional databases transaction is the mechanism change to the database. By contrast information in data warehouse is relatively coarse grained and refresh policy is carefully chosen, usually incremental.

CHARACTERISTICS OF DATA WAREHOUSES

- Multidimensional conceptual view
- Generic dimensionality
- Unlimited dimensions and aggregation levels
- Unrestricted cross-dimensional operations
- Dynamic sparse matrix handling
- Client-server architecture
- Multi-user support
- Accessibility
- Transparency
- Intuitive data manipulation
- Consistent reporting performance
- Flexible reporting

MULTIMEDIA DATABASES

- Applications:
  - repositories
  - presentation
  - collaboration
- issues
  - modeling: complex objects with “hidden” semantics
  - indexing
  - storage: generally very large objects, not suitable for storage as records in files
  - queries and retrievals:
    What to match? What to return?

GENOME DATA MANAGEMENT

- data characteristics
  - highly complex
  - highly variable
  - rapidly changing schema
  - multiple representations of same data
  - generally read-only
  - most users are not database savvy
  - context dependent
  - representation of complex queries is important