Heterogeneous Databases

Problem: Data “grows up” in different silos
- Independent systems
- Designed for different purposes
- No thought of sharing/interaction

Sources
- Company mergers
- Small-scale systems that grow
  - E.g., Access databases
- Cross-border
Heterogeneous Distributed Databases

- Many database applications require data from a variety of preexisting databases located in a heterogeneous collection of hardware and software platforms.
- Data models may differ (hierarchical, relational, etc.).
- Transaction commit protocols may be incompatible.
- Concurrency control may be based on different techniques (locking, timestamping, etc.).
- System-level details almost certainly are totally incompatible.
- A **multidatabase system** is a software layer on top of existing database systems, which is designed to manipulate information in heterogeneous databases.
  - Creates an illusion of logical database integration without any physical database integration.

Approaches

- **Database conversion**
  - Merge into a single format/model/system
    - Single schema
    - Single system, but separate schemas
  - *Often used with data warehousing*
- **Multidatabase**
  - Utilize existing, separated databases
  - Tools or APIs allowing interoperation
Advantages

- Preservation of investment in existing
  - hardware
  - system software
  - Applications
- Local autonomy and administrative control
- Allows use of special-purpose DBMSs
- Step towards a unified homogeneous DBMS
  - Full integration into a homogeneous DBMS faces
    - Technical difficulties and cost of conversion
    - Organizational/political difficulties
      - Organizations do not want to give up control on their data
      - Local databases wish to retain a great deal of autonomy

Unified View of Data

- Agreement on a common data model
  - Typically the relational model
- Agreement on a common conceptual schema
  - Different names for same relation/attribute
  - Same relation/attribute name means different things
- Agreement on a single representation of shared data
  - E.g. data types, precision,
  - Character sets
    - ASCII vs EBCDIC
    - Sort order variations
- Agreement on units of measure
- Variations in names
  - E.g. Köln vs Cologne, Mumbai vs Bombay
Problems to be Solved

- System Interoperability
  - Communications
  - Transaction Management
- Schema Integration
  - Schema matching
  - Schema mapping
- Record Linkage

System Interoperability

- Easiest: Common API (JDBC, ODBC)
  - Typically designed for client/server
  - “One-way” view of interoperation
- Distributed database APIs
  - Typically vendor-specific, may not be open
- Wrappers
  - Make one system “look like” another
Schema Matching

- Inconsistent terminology
  - Synonyms, homonyms
- No direct mappings
- Lack of documentation

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Birthdate</th>
<th>Course</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>7239</td>
<td>Chris Clifton</td>
<td>33/13/1872</td>
<td>CS34800</td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Last Name</th>
<th>First Name</th>
<th>ID</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clifton</td>
<td>Chris</td>
<td>348</td>
<td>MWF 11:30</td>
</tr>
</tbody>
</table>

© 2016 Christopher W. Clifton
Schema Mapping

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Birthdate</th>
<th>Course</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>7239</td>
<td>Chris Clifton</td>
<td>33/13/1872</td>
<td>CS34800</td>
<td>A</td>
</tr>
</tbody>
</table>

- **Rules for mapping**
  - Name = First Name ‘ ‘ Last Name
- **Wrappers**
  - Build with views?

Record Linkage

- **Identifying when records refer to the same entity**
  - Mismatch (e.g., different keys)
  - Data inconsistencies
    - Noise
    - intentional
Problems Interact

• Linked records can suggest schema mappings
• Knowledge of relationships within database can influence both
  – Keys
  – Functional Dependencies

Mediator Systems

- **Mediator** systems are systems that integrate multiple heterogeneous data sources by providing an integrated global view, and providing query facilities on global view
  - Unlike full fledged multidatabase systems, mediators generally do not bother about transaction processing
  - But the terms mediator and multidatabase are sometimes used interchangeably
  - The term **virtual database** is also used to refer to mediator/multidatabase systems
Problems often inexact

- Matches, mappings, linkage may be context dependent
  - “The same” isn’t always clear or consistent
- Often amenable to machine learning
  - Inexact answers
- Open area of research