Managing Inconsistency and Uncertainty in Databases

Renée J. Miller
University of Toronto

October 2, 2007
Bertinoro, Italy

Bertinoro Workshop on Information Integration (INFINT)

A Rosenthal Decomposition

Clio

Nice clean problem

Create mappings that ensure all target tgds (foreign keys) are satisfied.
Nice elegant solution.

Remainder

Managing data that violates key constraints
Need a solution…

UofT:DB Group

Contributors

- Ariel Fuxman, PhD Thesis
  - *Microsoft Search Labs*
  - ICDT05, SIGMOD05, VLDB05(demo), ICDE06
- Periklis Andritsos
  - *University of Trento*
  - ICDE06
- Elham Fazli and Jiang Du, MS students
- Diego Fuxman, Undergrad Thesis
- Oktie Hassanzadeh, PhD student

Dirty Databases

- The presence of dirty data is known to be a major problem in enterprises
  - "A quarter of data held in commercial enterprises is flawed" (Gartner 2004)
  - "Three quarters of financial service providers are making business decisions based on suboptimal data" (Gartner 2003)

Without getting a warning from their database systems

Querying a Dirty Database

- Fundamental assumption underlying traditional database systems
  - Data is clean and consistent
- If we break this assumption
  - Query results may not be meaningful
- Current Solution
  - Data Cleaning

Limitations of Data Cleaning

- Semi-automatic process
  - Requires highly-qualified domain experts
- Time consuming
  - May not be possible to wait until the database is clean

Our Proposal

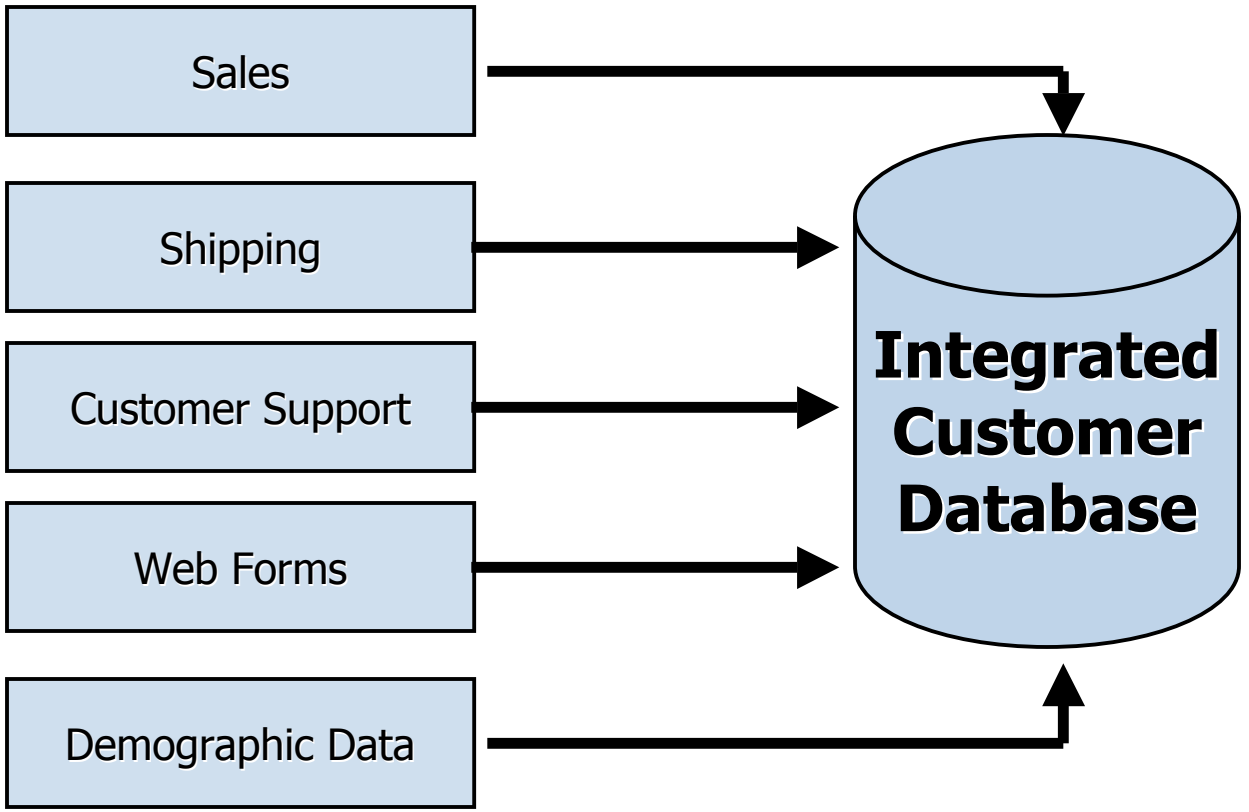
- A complementary approach to data cleaning
- Extend database systems to obtain meaningful answers from **potentially dirty** databases
  - Semantics of “dirty data”
  - Semantics of “meaningful answer”
  - Efficient query processing

Outline

- **Data Integration Example**
- Semantics
- Contributions
- Experimental Evaluation
- Future Challenges

A Data Integration Example

Integrating customer data…



Matching and Merging

- Matching is well supported by tools
  - Industrial tools (e.g., IBM's QualityStage, SQL Server Integration Services)
  - Research work - approximate joins
- Less tool support for merging

custid	*name*	*address*	…	*income*
Peter	Peter Yarrow	276 College Street	….	40K
	Paul Stookey	100 Bloor Street	…	400K
	Mary Travers	20 Union Street	…	110K

web

custid	*name*	*address*	…	*income*
Peter	P. Yarrow	276 College St.	….	200K
	P. Stookey	100 Bloor St.	…	400K
	M. Travers	20 Union St.	…	130K

sales

Merging (Resolution)

- Conflict resolution rules may be difficult to design
- True disagreement between sources

custid	*name*	*address*	…	*income*
Peter	Peter Yarrow	276 College Street	….	40K
	Paul Stookey	100 Bloor Street	…	400K
	Mary Travers	20 Union Street	…	110K

web

custid	*name*	*address*	…	*income*
Peter	Peter Yarrow	276 College Street	….	200K
	P. Stookey	100 Bloor St.	…	400K
	M. Travers	20 Union St.	…	130K

sales

Inconsistent Integrated Databases

In the absence of resolution rules…

SATISFY custid KEY

Web

custid	…	***income***
Peter	…	40K
Paul	…	400K
Mary	…	110K

Sales

custid	…	***income***
Peter	…	200K
Paul	…	400K
Mary	…	130K

VIOLATES custid KEY

Inconsistent Integrated Database

custid	…	***income***
Peter	…	40K
Peter	…	200K
Paul	…	400K
Mary	…	110K
Mary	…	130K

Querying an Inconsistent Database

Example: Offering a Platinum credit card…

q_1="Get customers who make more than 100K"

Peter,Paul,Mary

Are we sure that we want to offer a card to Peter?

custid	***income***	
Peter	40K	*web*
Peter	200K	*sales*
Paul	400K	*sales/web*
Mary	110K	*web*
Mary	130K	*sales*

Querying an Inconsistent Database

- **Aggressive:** Get customers who **possibly** make more than 100K
  - *Peter, Paul, Mary*
- **Conservative:** Get customers who **certainly** make more than 100K
  - *Paul, Mary*

custid	***income***	
Peter	40K	*web*
Peter	200K	*sales*
Paul	400K	*sales/web*
Mary	110K	*web*
Mary	130K	*sales*

Answer

Peter	possibly
Paul	certainly
Mary	certainly

Outline

- Data Integration Example
- **Semantics**
- Contributions
- Experimental Evaluation
- Future Challenges

Formal Semantics

- Models for incomplete data
  - [Imielinski Lipski 84]
- Querying incomplete data
  - [Imielinski Lipski 84, Abiteboul Duschka 98]
  - Possible world: “complete” database
- Querying inconsistent data
  - [Arenas Bertossi Chomicki 99]
  - Possible world: (maximal) “consistent” database
  - **Consistent answers**: conservative semantics

Consistent Answers

Inconsistent database

custid	***income***	
Peter	40K	*web*
Peter	200K	*sales*
Paul	400K	*sales/web*
Mary	110K	*web*
Mary	130K	*sales*

Key: *custid*

Repairs

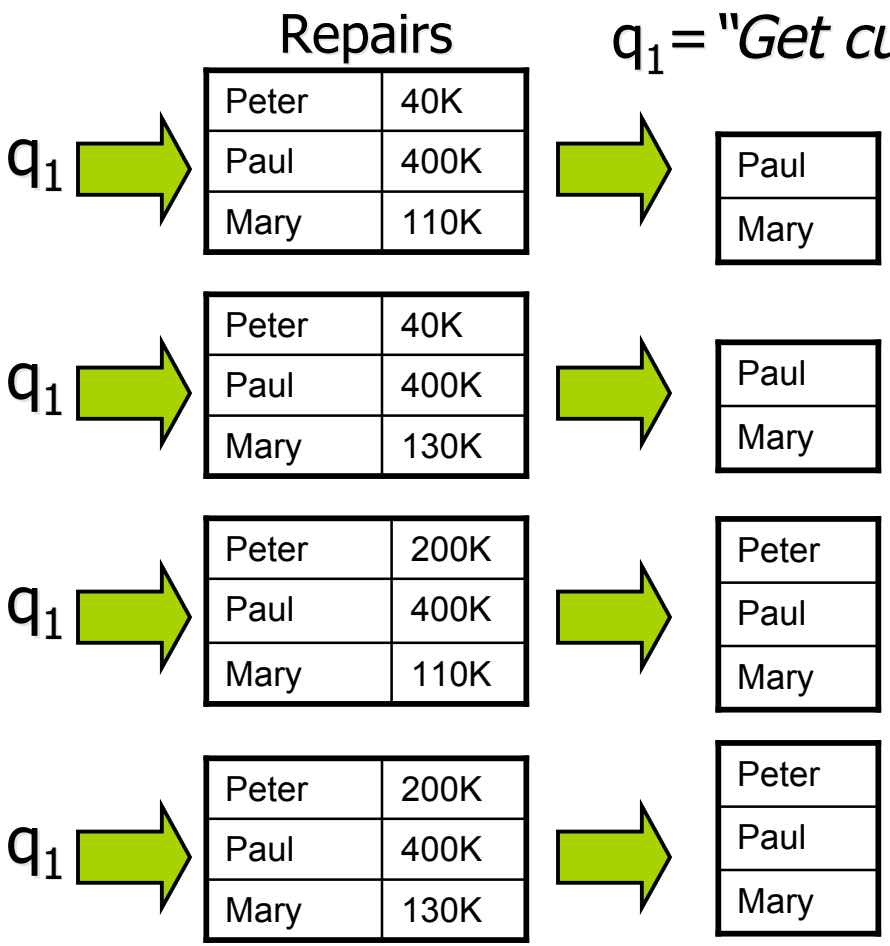
Peter	40K
Paul	400K
Mary	110K

Peter	40K
Paul	400K
Mary	130K

Peter	200K
Paul	400K
Mary	110K

Peter	200K
Paul	400K
Mary	130K

Consistent Query Answers



CONSISTENT ANSWER

Answer obtained no matter which repair we choose

{Paul, Mary}

Problem

- Potentially **HUGE** number of repairs!
- Plenty of negative results
  - [Chomicki et al 02, Arenas et al. 01, Cali et al 04]
- Few tractability results
  - [Arenas et al. 99, Arenas et al. 01]
- Computation based on logic programming
  - [Bravo and Bertossi 03, Eiter et al. 03, see Rosati talk]]
  - Consider rich sets of constraints
  - May be computationally expensive
  - Focus (mostly) small number conflicts

Bridging Theory and Practice

Are these ideas applicable to realistic scenarios?

YES!

Negative results do not rule out many realistic scenarios, such as:

- Key constraints
- Commonly used decision support queries

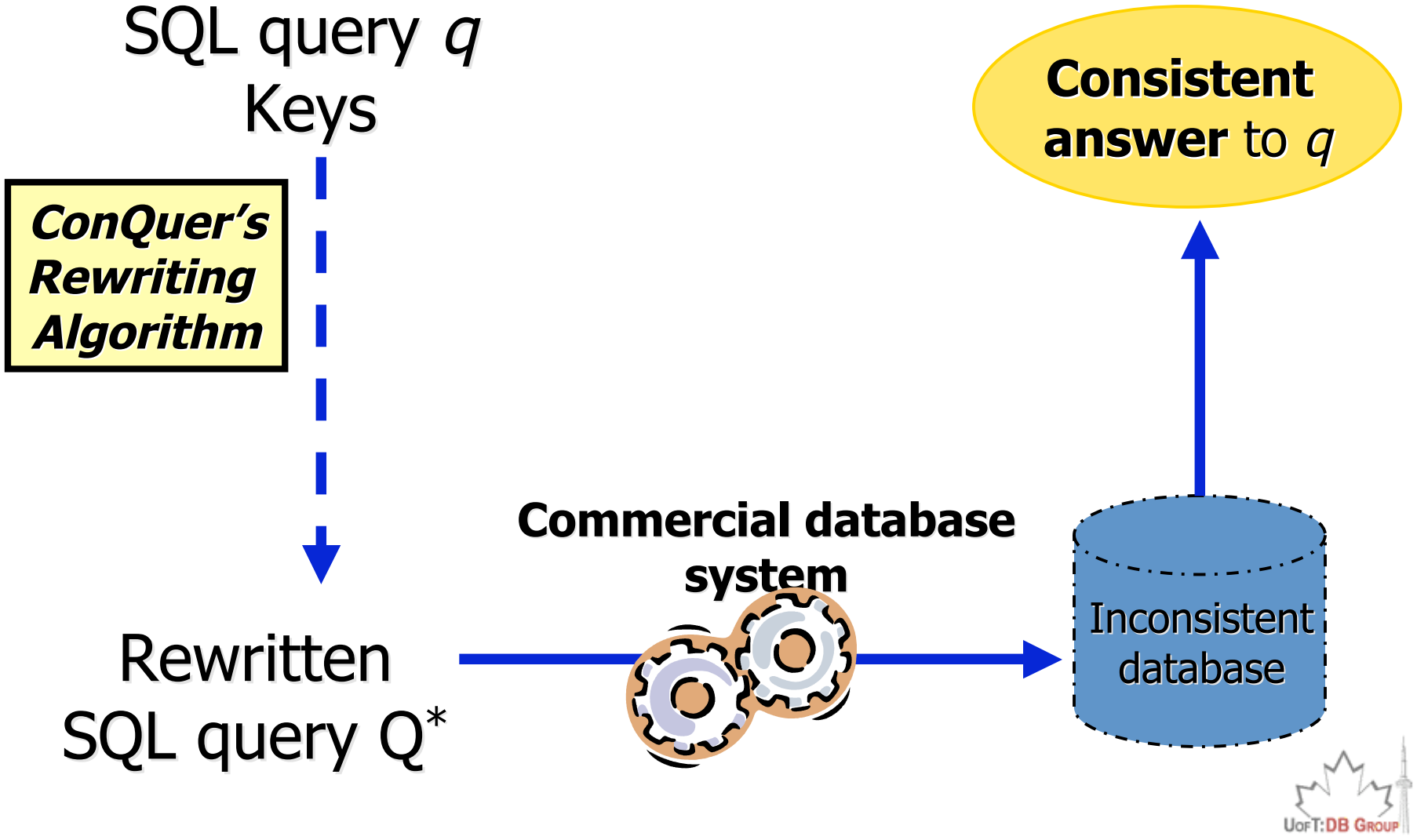
Outline

- Data Integration Example
- Semantics
- **Contributions**
- Experimental Evaluation
- Future Challenges

Contributions

- Formal characterization of a class of queries for which computing consistent answers is tractable **[ICDT05-JCSS07]**
- Algorithms for efficient computation of consistent answers using SQL rewriting **[SIGMOD05]**
- A probabilistic semantics that generalizes consistent answers **[ICDE06]**
- *ConQuer*, a system designed to compute consistent answers efficiently **[VLDB05 demo]**

ConQuer


SQL query *q*
Keys
Consistent answer to *q*
ConQuer's Rewriting Algorithm
Commercial database system
Rewritten SQL query Q*
Inconsistent database
UofT:DB Group

A Large Class of Queries

- ConQuer supports SQL queries with
  - Joins between keys
  - Joins between non-keys and keys
  - Arbitrary number of relations
  - Set and bag semantics
  - Grouping and aggregation
- Showed (experimentally) applicable to
  - Large instances (50G)
  - Large number of conflicts (50% dirty data)
- Restrictions
  - No joins between non-keys (unless in SELECT)
  - Acyclic non-key to key structure (no self joins)
  - Subqueries must be decorrelated

Uncertain Data

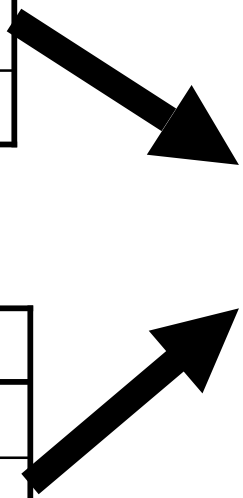
PROVENANCE INFORMATION
(e.g., source reputation)

0.3
Web

custid	…	*income*
Peter	…	40K
Paul	…	400K
Mary	…	110K

0.7
Sales

custid	…	*income*
Peter	…	200K
Paul	…	400K
Mary	…	130K



Integrated Database

custid	…	*income*	
Peter	…	40K	**0.3**
Peter	…	200K	**0.7**
Paul	…	400K	**1**
Mary	…	110K	**0.3**
Mary	…	130K	**0.7**

Probabilistic Semantics

- Probabilistic model based on the basic assumptions of repairs
  - Tuples that share the same key value do not appear in the same repair (mutually exclusive)
  - Tuples that do not share a key value are chosen independently
- Probability distribution over repairs
- Query results are assigned a “confidence” of being obtained from the clean database

Uncertain Inconsistent DB

- An uncertain inconsistent DB assumes
  - Alternatives for a key value are mutually exclusive
  - Tuples for different key values are independent
- Presented rewriting for computing consistent answers with probabilities [ICDE 06]
  - Again focus was on scalability/execution of rewriting
- Since our work such databases have also been considered by Dalvi and Suciu [PODS 07] invited talk
  - “Disjoint-independent” databases
  - Presents classification of queries that extends ours

Outline

- Data Integration Example
- Semantics
- Contributions
- **Experimental Evaluation**
- Future Challenges

Overview of Experimental Results

- Showed scalability of rewritings on
  - TPC-H queries
  - Up to 50G with varying degree and type of inconsistency
  - Overhead
    - Getting consistent and possible answers for most queries takes 2-3 times as long as getting possible answers alone

Outline

- Data Integration Example
- Semantics
- Contributions
- Experimental Evaluation
- **Future Challenges**

Future Challenge 0

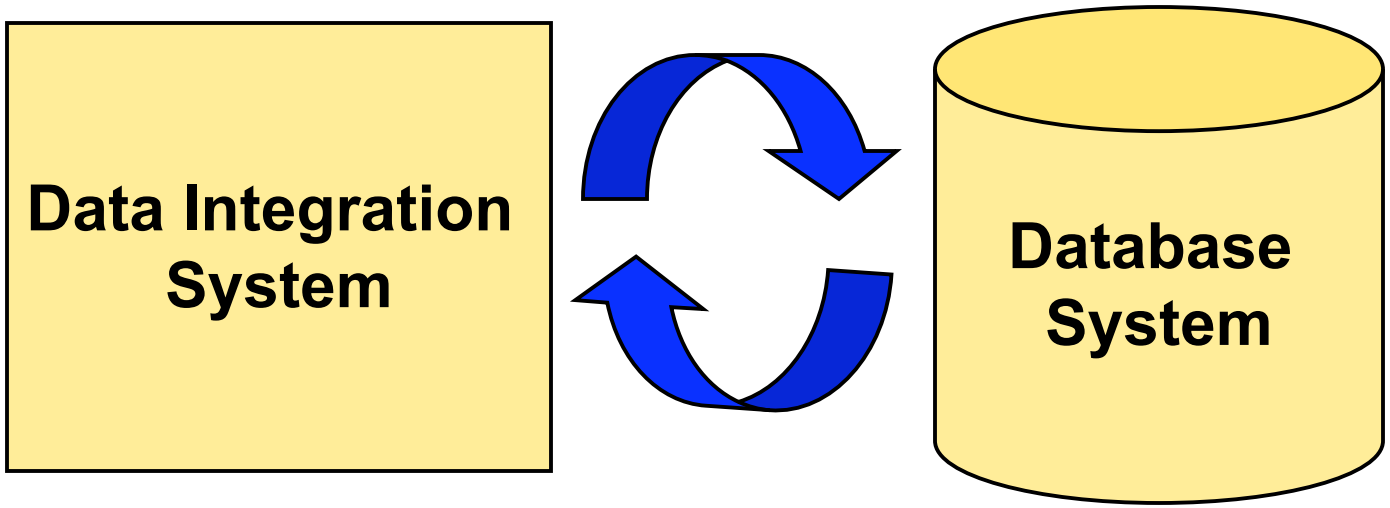
- Our approach does not work for ALL queries
  - Our rewriting is sound but it may “miss” some consistent answers for queries outside out class (that is for “hard” queries)
  - Hard queries are only hard for some instances
  - Can we approximate consistent answers for all queries? (Apply another Rosenthal decomposition…)

Future Challenge 1

- How do we meaningfully assign probabilities to alternative, inconsistent tuples?
  - Is it enough to use output of a similarity measure (or approximate join technique)?
  - What probability assignments lead to the most meaningful query answers?

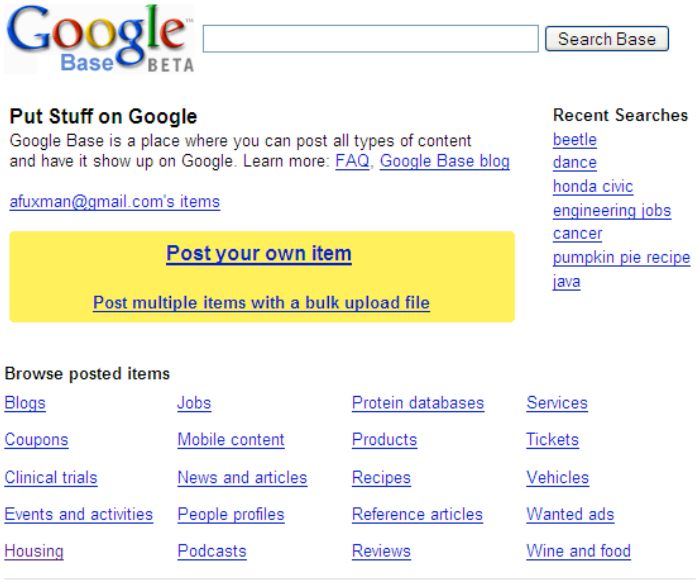
Future Challenge 2

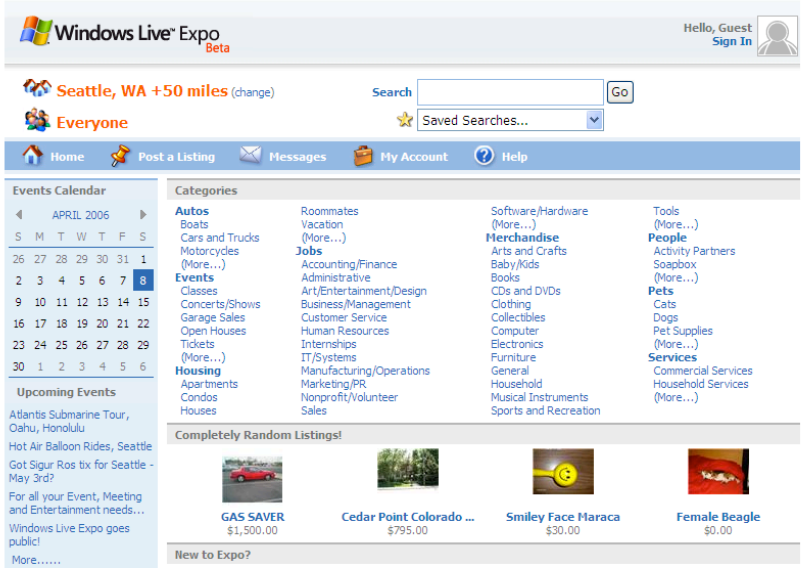
- Promote querying dirty data as an integral part of mainstream data integration and database systems



Future Challenge 3

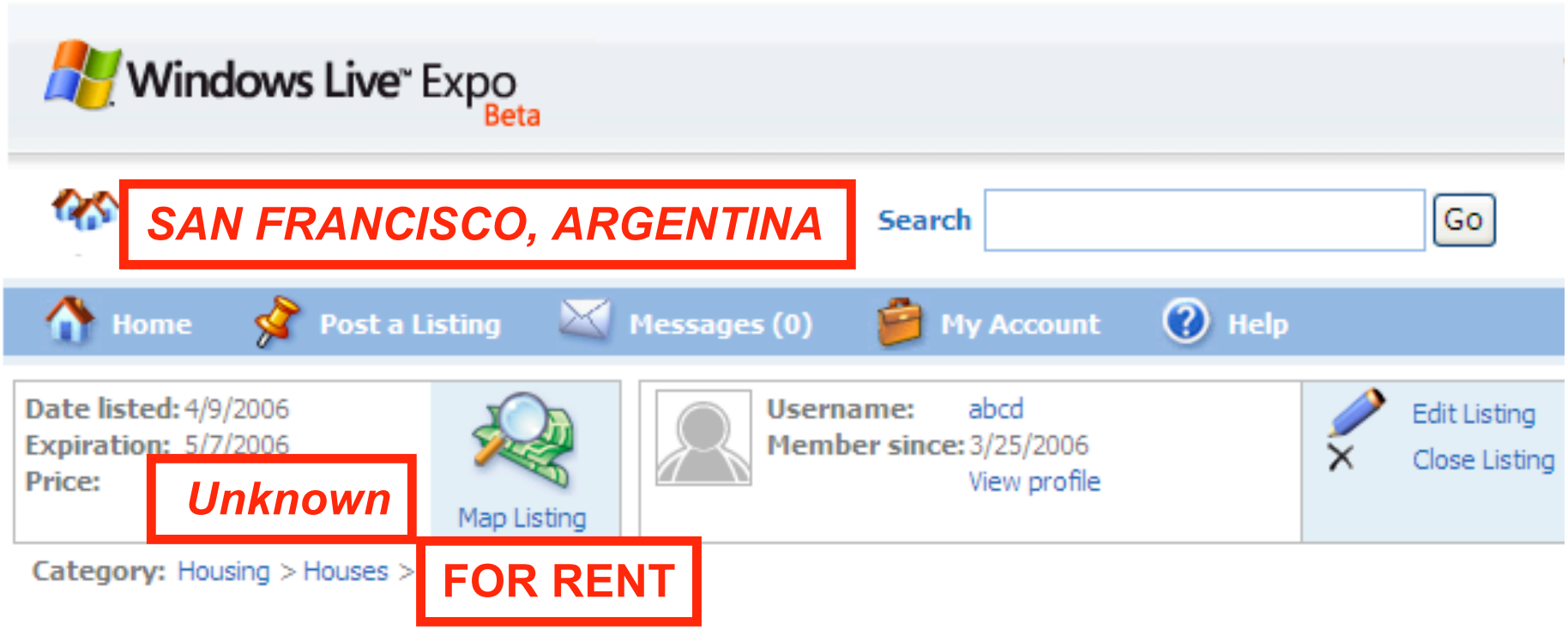
- Bring the paradigm of querying dirty data to new domains, including Web applications





UofT:DB Group

Future Challenge 3



UofT:DB Group

Questions?

Web Page:

www.cs.toronto.edu/~miller