Information Integration

Problem: Related data exists in many places and in many forms. They talk about the same things, but differ in the model, schema, or terminology.

Goal: To provide a uniform interface to a multitude of data sources.

Three Approaches

1. Federated Databases: The sources are independent, but one source can call the other.
2. Warehousing: Make copies of information at each data source centrally.
   - Reconstruct data at regular intervals (daily/weekly/monthly), but it is never up-to-date.
3. Mediation: Create a view of all information, but do not make copies.
   - Answer queries by sending appropriate queries to the sources.

Mediator Approach

Users pose queries in terms of a mediated schema.

There must be some description of the relationship between the source relations and the mediated schema.

The query processor must be able to reformulate a query posed in terms of the mediated schema into a query against the source schemas.

Use a restricted form of first-order logic;

Conjunctive Queries

\[ q(X) : = e_1(X_1), \ldots, e_n(X_n) \]

where \( e_1, \ldots, e_n \) are database relations, and \( X_1, \ldots, X_n \) are database relations and \( X_1, \ldots, X_n \) are tuples of variables and constants.

Queries with unions are expressed by multiple rules with the same head predicate.

A view refers to a named query, and it is said to be materialized if its results are stored in the database.

Query Containment and Equivalence

A query \( Q_1 \) is said to be contained in a query \( Q_2 \), denoted \( Q_1 \sqsubseteq Q_2 \) if for any database \( D \), \( Q_1(D) \subseteq Q_2(D) \).

How do we express the equivalence of two queries.

The Problem

- Need a description of the relation between the source relations and the global relations. Two main approaches.
- Need to rewrite the user query expressed in the mediated schema into a query expressed in the source schema.

So, given such a query \( Q \), find a query \( Q' \) that uses only the source relations, such that:
- \( Q' \models Q \)
- \( Q' \) provides all possible answers to \( Q \) given the sources
Global as View

GAV: For each relation $R$ in the mediated schema, we write a query over the source relations specifying how to obtain $R$'s tuples from the sources.

Example: We have two sources DB1 and DB2 containing titles actors and years of movies

$\text{MovieActor}(\text{title}, \text{actor}) \leftarrow\text{DB1}(\text{id}, \text{title}, \text{actor}, \text{year})$

$\text{MovieActor}(\text{title}, \text{actor}) \leftarrow\text{DB2}(\text{id}, \text{title}, \text{actor}, \text{year})$

If we then add a third source DB3 that provides movie reviews, we might add:

$\text{MovieReview}(\text{title}, \text{review}) \leftarrow\text{DB1}(\text{id}, \text{title}, \text{actor}, \text{year}) \text{ AND } \text{DB3}(\text{id}, \text{review})$

The second clause is clearly redundant.

Queries

Find reviews for movies starring Marlon Brando:

$q(\text{title}, \text{review}) :=$

$\text{MovieActor}(\text{title}, \text{Brando}) \text{ AND } \text{MovieActor}(\text{title}, \text{review})$. 

Unfolding the descriptions of $\text{MovieActor}$ and $\text{MovieReview}$ will yield the following queries over the source relations:

$q(\text{title}, \text{review}) :=$

$\text{DB1}(\text{id}, \text{title}, \text{Brando}, \text{year}) \text{ AND } \text{DB3}(\text{id}, \text{review})$

$q(\text{title}, \text{review}) :=$

$\text{DB1}(\text{id}, \text{title}, \text{Brando}, \text{year}) \text{ AND } \text{DB2}(\text{id}, \text{title}, \text{Brando}, \text{year}) \text{ AND } \text{DB3}(\text{id}, \text{review})$

The second clause is clearly redundant.

Comparison

• GAV
  – Query Reformulation is very simple.
  – Adding sources is more difficult.

• LAV
  – Adding sources is easy.
  – Query reformulation is difficult.

Systems

• GAV
  – TSIMMIS (Stanford)
  – HERMES (University of Maryland)

• LAV
  – Information Manifold (AT&T)
  – InfoMaster (Stanford)
  – Tukwila (University of Washington)