

Information Integration

MEDIATORS

WAREHOUSING

ANSWERING QUERIES USING VIEWS

Information Integration

Information integration is the process of taking several databases and making the data in these sources work together as if they were a single database.

The integrated database may be

- Physical ("data warehouse")
- Virtual ("mediator") that may be queried even though it does not exist physically

Information-integration systems require special kinds of query-optimization techniques for their efficient operation.

Why Information Integration?

If we could put data always in a single database, there would be no need for information integration.

However, in the real world, matters are rather different..

- Databases are created independently, even if they later need to work together.
- The use of databases evolves, so we cannot design a database to support every possible future use.

Example Applications

1. Enterprise Information Integration: making separate DB's, all owned by one company, work together.
2. Scientific DB's, e.g., genome DB's.
3. Catalog integration: combining product information from all your suppliers.

Challenges

1. *Legacy databases* : DB's get used for many applications.
 - ◆ You can't change its structure for the sake of one application, because it will cause others to break.
2. *Incompatibilities (heterogeneity problem)*: Two, supposedly similar databases, will mismatch in many ways.

Examples: Incompatibilities

Lexical : `addr` in one DB is `address` in another.

Value mismatches : is a “BL” car the same color in each DB (blue versus black)? Is 20 degrees Fahrenheit or Centigrade?

Semantic : are “employees” in each database the same? What about consultants? Retirees? Contractors?

Query-Language heterogeneity : Relational database (SQL) versus XML (Xquery)

Data Type differences : Serial numbers might be represented as *string* in one source and *integer* in another source.

Examples: Schema Heterogeneity

One dealer might store cars in a single relation that look like:

- `Cars(serialNo, model, color, autoTrans, navi, ...)`

Another dealer might use a schema in which options are separated out into a second relation, such as:

- `Autos(serial, model, color)`
- `Options(serial, option)`

What Do You Do About It?

Grubby, handwritten translation at each interface.

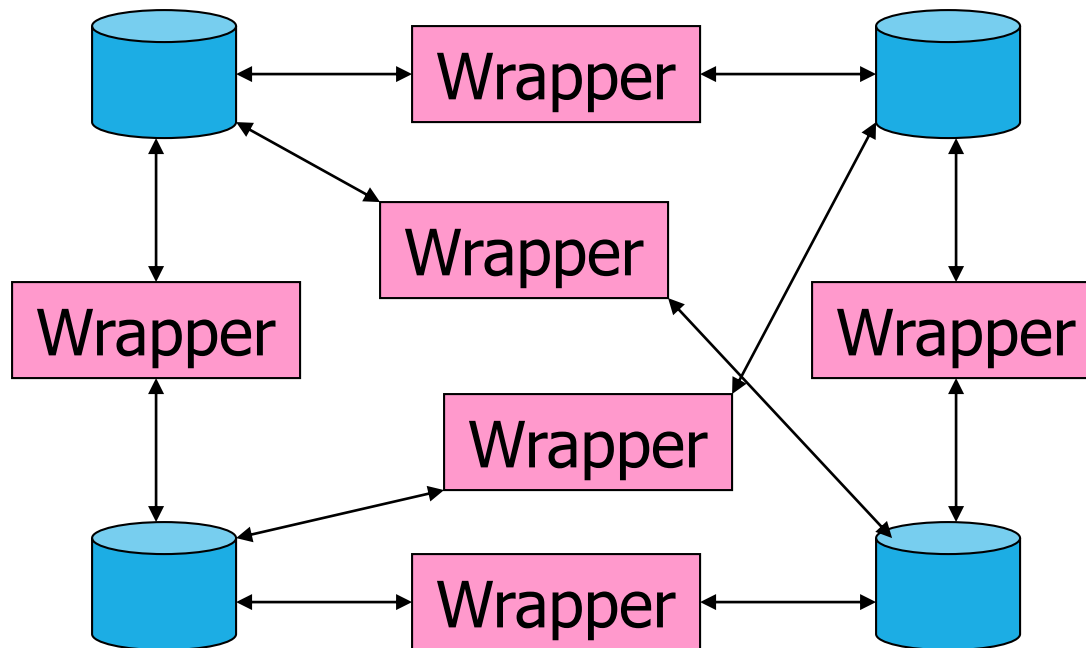
- Some research on automatic inference of relationships.

Wrapper (aka “adapter”) translates incoming queries and outgoing answers.

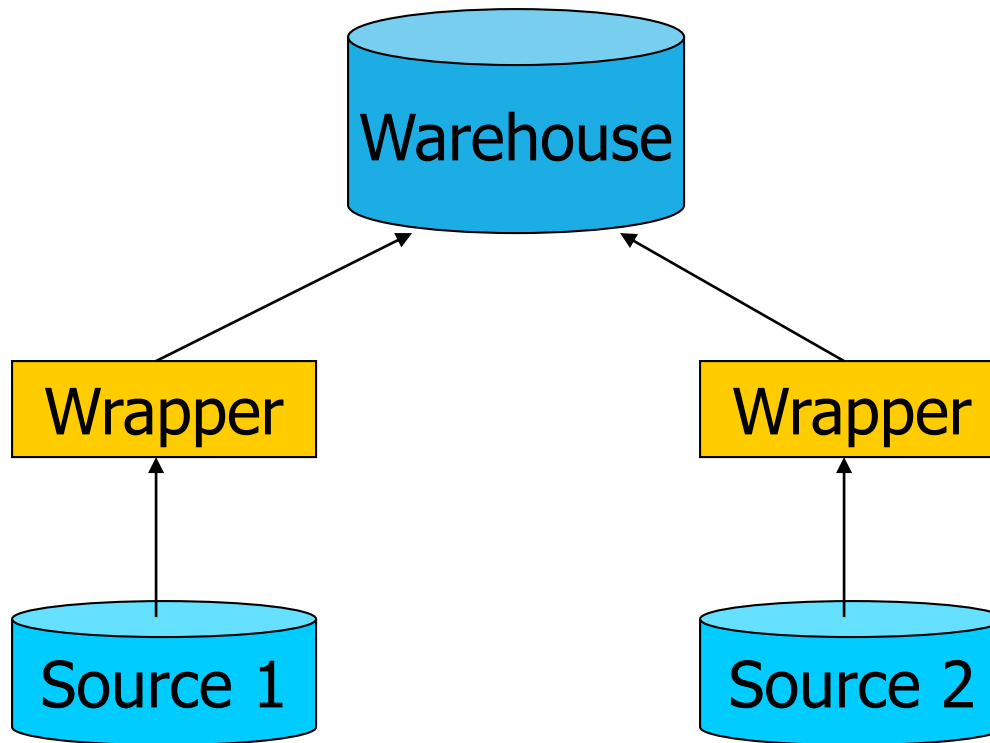
Integration Architectures

1. *Federation* : everybody talks directly to everyone else.
2. *Warehouse* : Sources are translated from their local schema to a global schema and copied to a central DB.
3. *Mediator* : *Virtual warehouse* --- turns a user query into a sequence of source queries.

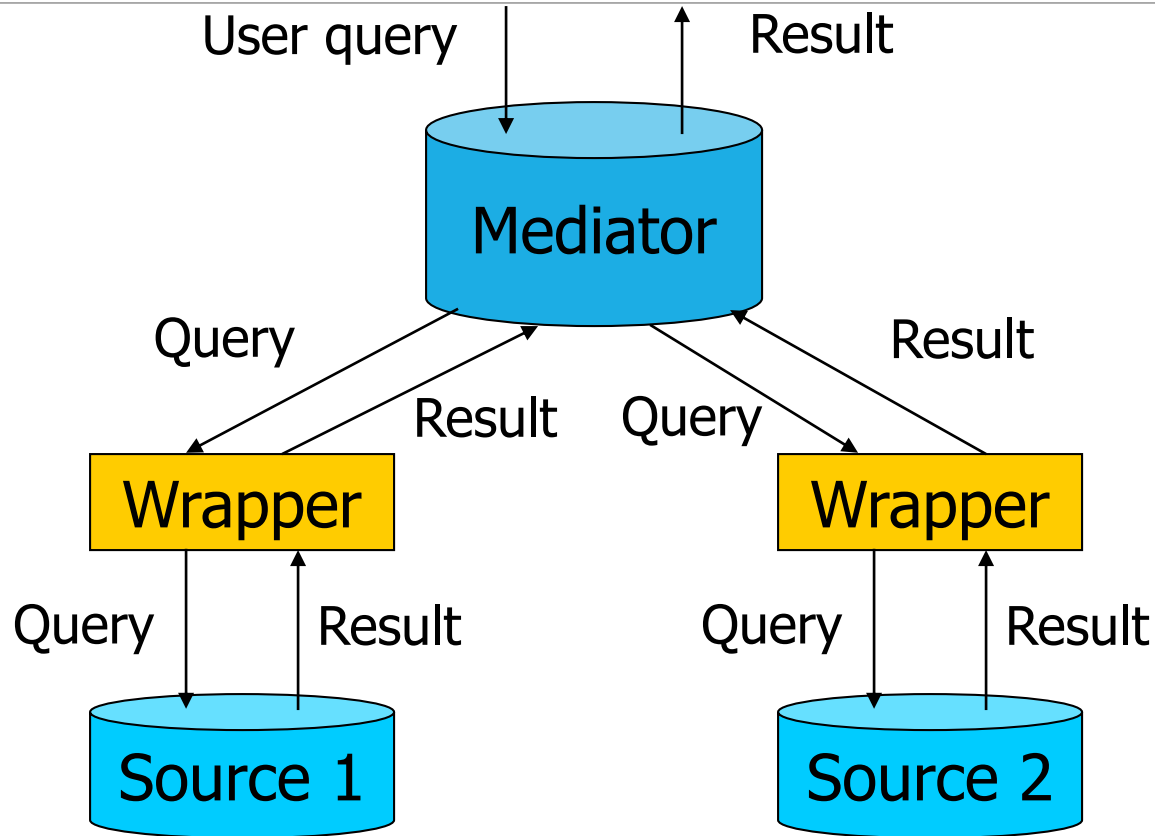
Federations



Warehouse Diagram



A Mediator



Schema Heterogeneity

One dealer might store cars in a single relation that look like:

- `Cars(serialNo, model, color, autoTrans, navi, ...)`

Another dealer might use a schema in which options are separated out into a second relation, such as:

- `Autos(serial, model, color)`
- `Options(serial, option)`

Example: Mediator

Suppose mediator integrates the same two automobile sources into a view that is a single relation with schema:

- AutosMed (serialNo, model, color, autoTrans, dealer)

Assume the user asks the mediator about red cars, with the query:

```
SELECT serialNo, model
FROM AutosMed
WHERE color = 'red';
```

Example: Mediator

The wrapper for Dealer 1 translates the query into the terms of the dealer's schema:

```
SELECT SerialNo, model
FROM Cars
WHERE color = 'red'
```

At the same time, the wrapper for Dealer 2 translates the same query into the schema of that dealer:

```
SELECT serial, model
FROM Autos
WHERE color = 'red';
```

The mediator takes union of these sets and returns the result to the user.

Mediation Approach

Mediator processes queries into steps executed at sources.

Example: Catalog Integration

Suppose Dell wants to buy a bus and a disk that share the same protocol.

Global schema: Buses (manf, model, protocol)
Disks (manf, model, protocol)

Local schemas: each bus or disk manufacturer has a (model, protocol) relation --- manf is implied.

Example: Global-as-View

Mediator might start by querying each bus manufacturer for model-protocol pairs.

- The wrapper would turn them into triples by adding the manf component.

Then, for each protocol returned, mediator queries disk manufacturers for disks with that protocol.

- Again, wrapper adds manf component.