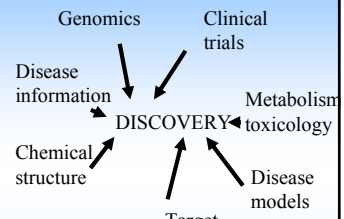
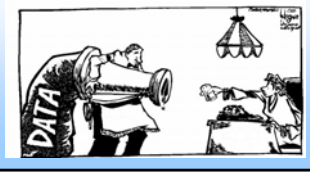
Integration of databanks

Accessing multiple data sources



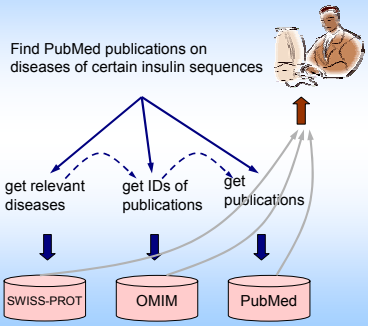


Access to multiple databanks- Problems

- Users need good knowledge on where the required information is stored and how it can be accessed
- Representation of an entity in different databanks can be different.

  Same name in different databanks can refer to different entities.

Queries over multiple databanks

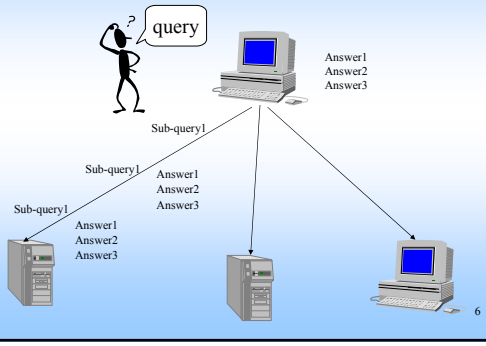


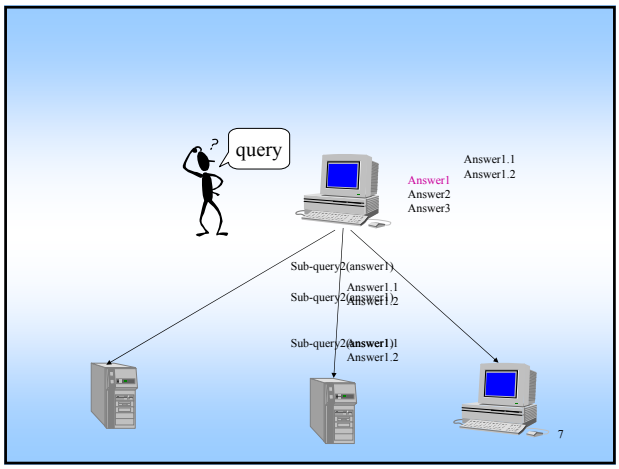
- Find
- Divide & order
- Execute
- Combine

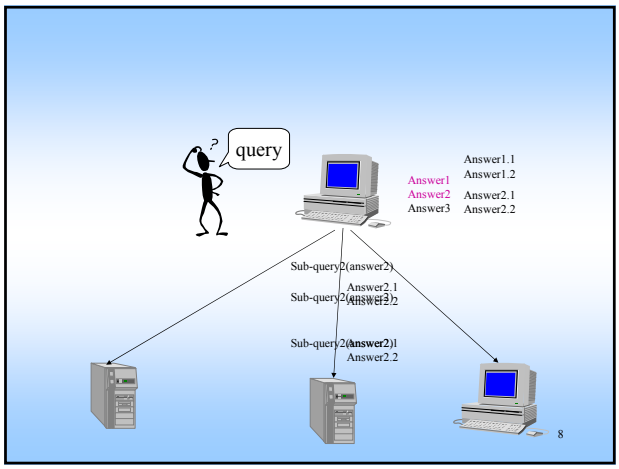
Access to multiple databanks - steps

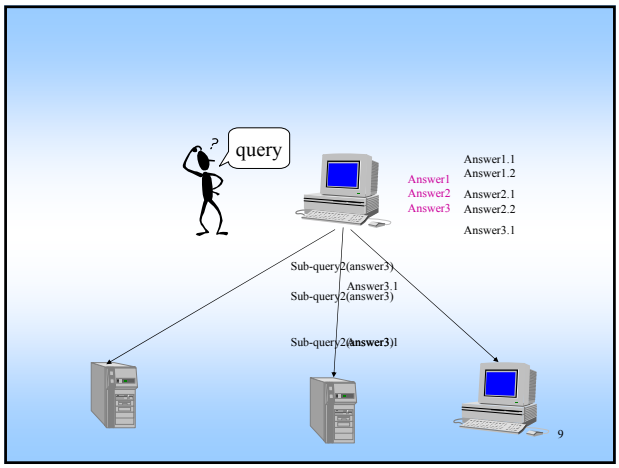
- Decide which databanks should be used
- Divide query into sub-queries to the databanks
- Decide in which order to send sub-queries to the databanks
- Send sub-queries to the databanks - use the terminology of the databanks
- Merge results from the databanks to an answer for the original query

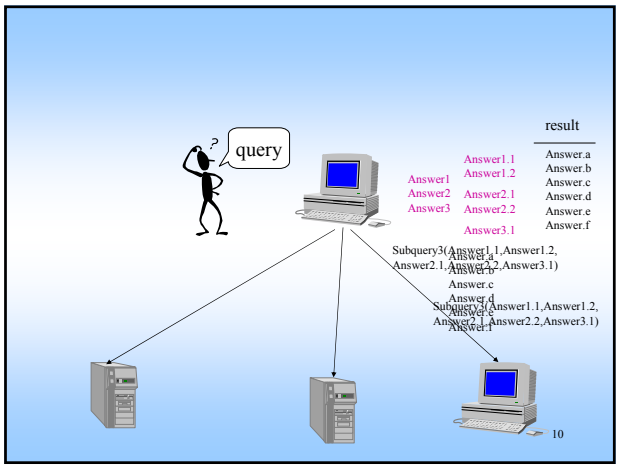
→ mistake in any step can lead to inefficient processing of the query or failure to get a result



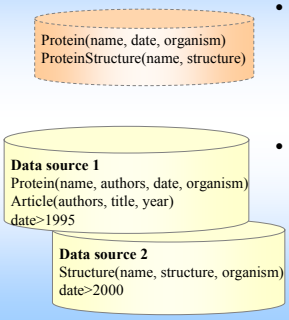








Problem formulation



- Databank properties
  - Autonomous databanks
  - Different data models
  - Differences in terminology
  - Overlapping, redundant data
- Integration aims to provide transparent access to multiple heterogenous databanks
  - uniform query language
  - uniform representation of results

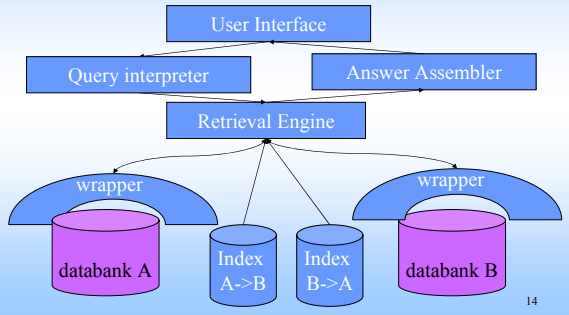
Methods for integration

- Link driven federations
  - Explicit links between databanks.
- Warehousing
  - Data is downloaded, filtered, integrated and stored in a warehouse. Answers to queries are taken from the warehouse.
- Mediation or View integration
  - A global schema is defined over all databanks.

Link driven federations

- Creates explicit links between databanks
- query: get interesting results and use web links to reach related data in other databanks
- systems: SRS, Entrez

Link driven federations



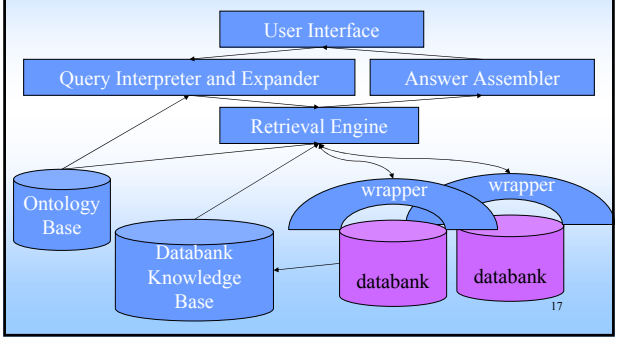
Link driven federations

- Advantages
  - complex queries
  - fast
- Disadvantages
  - require good knowledge
  - syntax based
  - terminology problem not solved

View integration

- Define a global schema over the databanks
- high level query language
- systems: BioKleisli, K2, TAMBIS, BioTRIFU

View integration



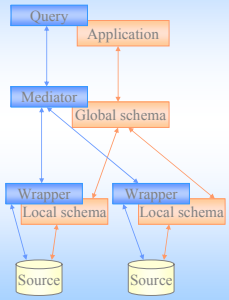
View integration

- Advantages
  - complex queries
  - requires less knowledge
  - solution for terminology problem
  - semantics based

View integration

- Disadvantages
  - more computation
  - view maintenance

Mediation



- Query problem
  How to answer queries expressed using the global schema.
- Modeling problem
  How to model the global schema, databanks och mappings.

Queries

- Queries use the global schema
- Conjunctive queries
  - select-project-join queries

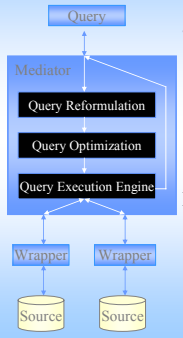
head if body/subgoals

p(X,Z) :- a(X,Y), b(Y,Z)

q(name, structure) :- Protein(name, 2001, 'human'),
ProteinStructure(name, structure)

- Mediator reformulates queries in terms of a set of queries that use the local schema. Equivalence and containment of queries needs to be preserved.
  - Q1 is contained in Q2
    if the result of Q1 is a subset of the result of Q2.

Mediator

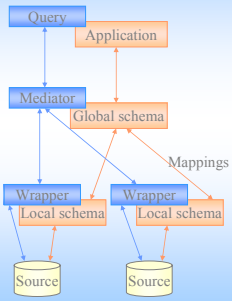


- Mediator is responsible for query processing
  - reformulation of queries decide query plan
  - query optimization
  - execution of query plan, assemble results into final answer

Issues:

- Semantically correct reformulation
- Access only relevant databanks

Knowledge



- Description of databank content
  - global schema (domain model/ontology)
  - local schema (databank model)
- Information for integration
  - mapping
- Capabilities
  - attributes and constraints
  - processing capabilities
  - completeness
  - cost of query answering
  - reliability
- Used for
  - selection of relevant databanks
  - query plan formulation
  - query plan optimization

Mapping

- Relation between domain and databank content

Global schema:
Protein(name, date, organism)
ProteinStructure(name, structure)

Data source local schema:
DS1(name, authors, date, organism)
DS2(name, structure, organism)

- Global as view
  The global schema is defined in terms of source terminology

Protein(name, date, organism) :- DS1(name, authors, date, organism)
ProteinStructure(name, structure) :- DS2(name, structure, organism)

Mapping

- Relation between domain and databank content

Global schema:
Protein(name, date, organism)
ProteinStructure(name, structure)

Data source local schema:
DS1(name, authors, date, organism)
DS2(name, structure, organism)

  - Local as view
    The sources are defined in terms of the global schema.

DS1(name, authors, date, organism) -: Protein(name, date, organism), date >1995
DS2(name, structure, organism) -: Protein(name, date, organism), ProteinStructure(name, structure), date >2000

Query processing in GAV

Query: give name and structure for human proteins with date '2001'.
q(name, structure) :- Protein(name, 2001, 'human'), ProteinStructure(name, structure)

GAV: Protein(name, date, organism) :- DS1(name, authors, date, organism)
ProteinStructure(name, structure) :- DS2(name, structure, organism)

- No explicit representation of databank content
- Mapping gives direct information about which data satisfies the global schema.
- Query is processed by expanding the query atoms according to their definitions.

New query: q(name, structure) :-
DS1(name, authors, 2001, 'human'), DS2(name, structure, organism)

Query processing in LAV

Query: give name and structure for human proteins with date '2001'.
q(name, structure) :- Protein(name, 2001, 'human'), ProteinStructure(name, structure)

LAV:
DS1(name, authors, date, organism) -: Protein(name, date, organism), date >1995
DS2(name, structure, organism) -: Protein(name, date, organism), ProteinStructure(name, structure), date >2000

- Mapping does not give direct information about which data satisfies the global schema.
- To answer the query it needs to be inferred how the mappings should be used.

Query processing in LAV

Query: give name and structure for human proteins with date '2001'.
q(name, structure) :- Protein(name, 2001, 'human'), ProteinStructure(name, structure)

LAV:
DS1(name, authors, date, organism) -: Protein(name, date, organism), date >1995
DS2(name, structure, organism) -: Protein(name, date, organism), ProteinStructure(name, structure), date >2000

- Bucket algoritm (Information Manifold)
  - For each sub-goal in query create bucket of relevant views.
  - Define rewritings of query. Each rewriting consists of one conjunct from every bucket. Check whether the resulting conjunction is contained in the query.
  - The result is the union of the rewritings.

New query: q(name, structure) :-
DS1(name, authors, 2001, 'human'), DS2(name, structure, organism)

Comparison GAV - LAV

- Global as view
  - Clear how databanks interact
  - When a databank is added, the global schema can change
  - Query processing is easy
- Local as view
  - Each databank is specified in isolation
  - Easy to add databanks
  - Easier to specify constraints on the contents of sources
  - Query processing requires reasoning

Capabilities

- Most common capabilities describe attributes
  - f - free, attribute can be specified or not
  - b - bound, a value must be specified for the attribute, all values are permitted
  - u - unspecified, not permitted to specify a value for the attribute
  - c[S] - value should be one of the values in finite set S
  - o[S] - value is not specified or one of the values in finite set S

DS1: (name, authors, date, organism) f f b c[human mouse]

View integration

Research issues:
- knowledge representation
- creation and maintenance of global schema

Ontology Base
Databank Knowledge Base
wrapper
wrapper
databank
databank

View integration

Research issues:
- knowledge representation
- creation of ontologies
- merging of ontologies

Ontology Base
Databank Knowledge Base
wrapper
wrapper
databank
databank

View integration

User Interface
Query Interpreter and Expander
Answer Assembler
Retrieval Engine

Research issues:
- unified query language
- knowledge representation
- strategies for query expansion
- query optimization

Ontology Base

View integration

Research issue:
- semi-automatic generation of wrappers

Ontology Base
Databank Knowledge Base
wrapper
wrapper
databank
databank

Integration - exercises

Integration systems

Discuss MedMaker, Information Manifold, SIMS and InfoSleuth

- What does the system focus on?
- Describe the query language.
- Describe the knowledge representation.
- Describe the query processing approach.
- Describe the system architecture.
- What are the advantages of the system?
- Interesting features?