COnnected KEywords

Grégory Smits, Olivier Pivert and Virginie Thion

IRISA, Campus de Beaulieu, 35042 Rennes, France
{gregory.smits, olivier.pivert, virginie.thion}@irisa.fr

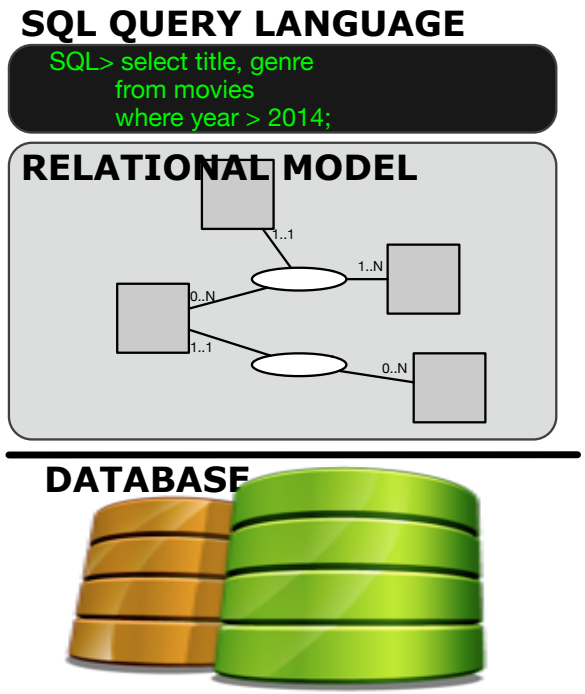
RCIS, May 13-15 2015, Athens, Greece

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES

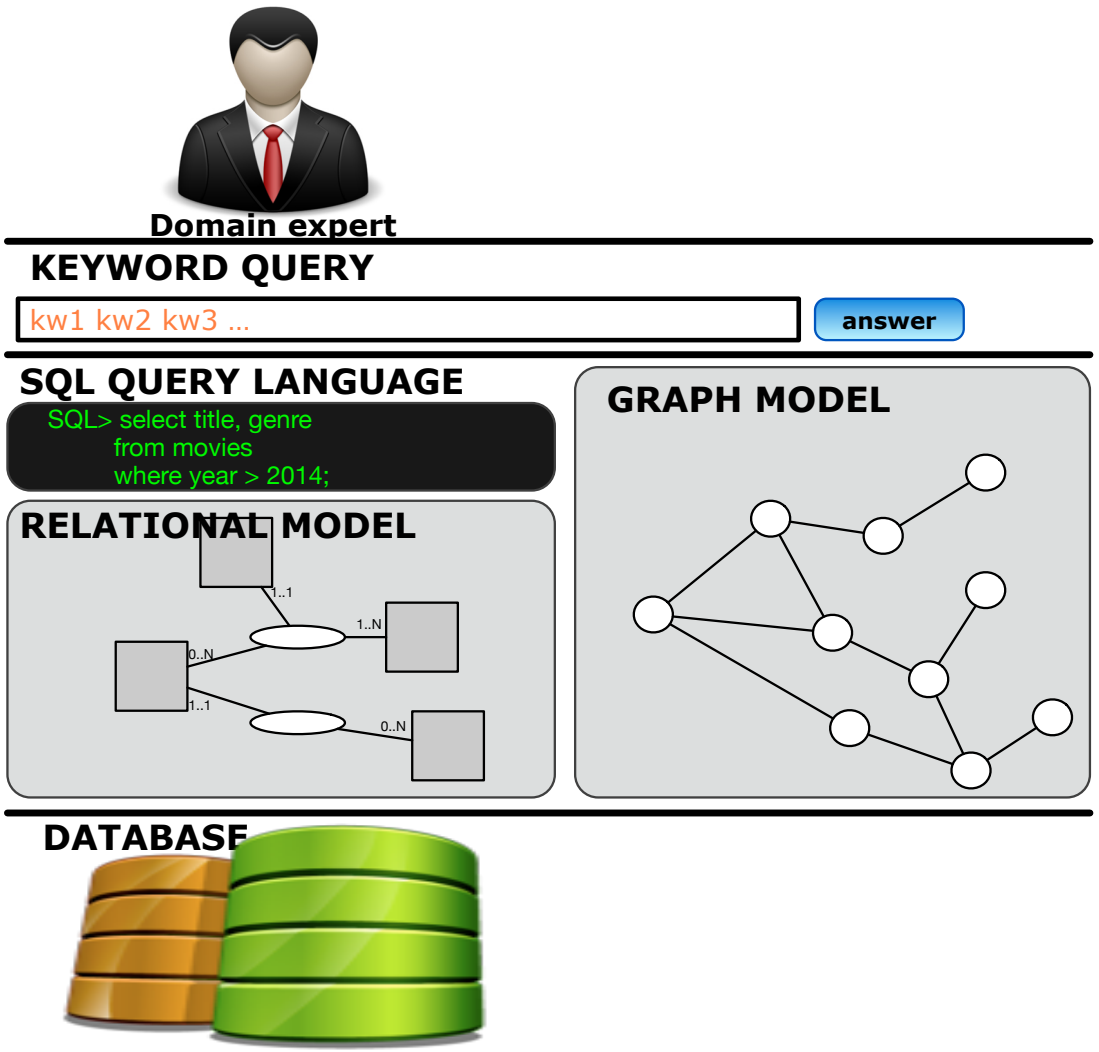
UMR IRISA

Giving Access to Structured Data

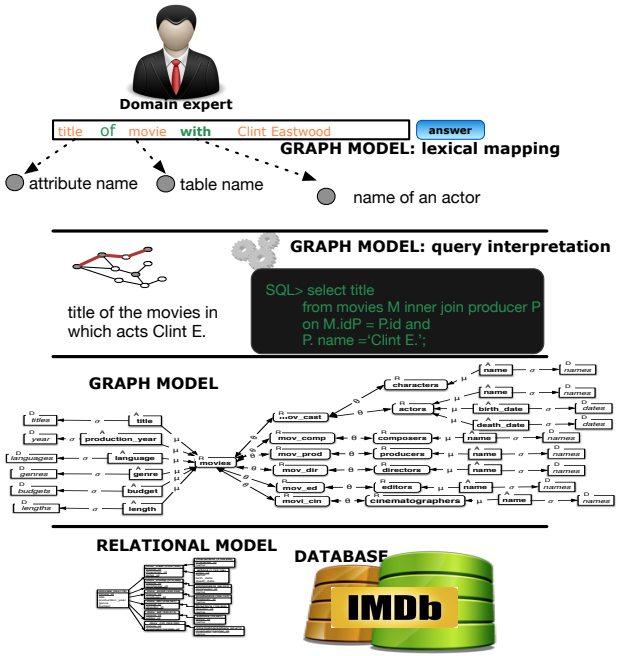
Domain expert



Giving Access to Structured Data



Motivations



More expressive query interface:

- keywords and connectives,
- constrained but rich query language,
  (SPJ queries, aggregates, some nested queries, ...)
  'movies Clint E. Gene H.'
  'genre of movies entitled Scarface'
  'movies genre is western released in 2014'

  'actors appearing in Dogma'
- tool to help domain experts:
  - query their corporate databases,
  - manage the COKE language.

Outline

1 Introduction
1.1 Problem
1.2 Motivating example

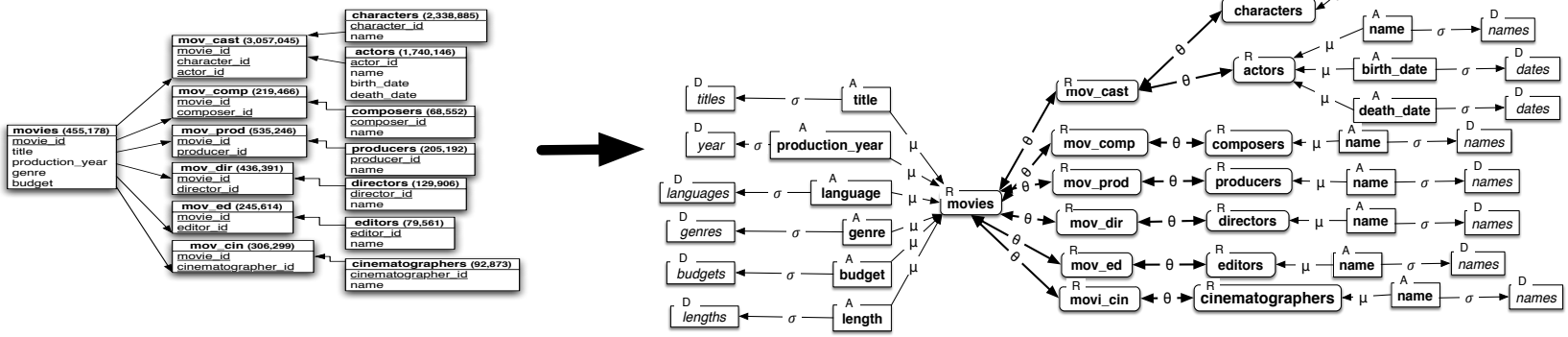
2 Data Model and Language
2.1 Schema Graph Model
2.2 Query Graph Model and Searchable Vocabulary
2.3 a Constrained Keyword-based Query Language

3 Cooperative Querying Approach
3.1 KW Query Definition
3.2 KW Query Interpretation
3.3 Translation into SQL

4 Experimentations

Translation of the Relational Schema into a Graph Schema

- capture the data semantics,
- have a more manageable model,
- make it possible to apply graph-based algorithms.



- **tables, attributes and definition domains** represented by **nodes**,
- **primary to foreign key constraints** form **links between table nodes**,
- **domain nodes** are **linked** to their **attribute nodes** themselves **linked** to their **table nodes**.

Query Graph Model: mapping between the model and SQL

Directed graph representing the elements that may appear in a user query:

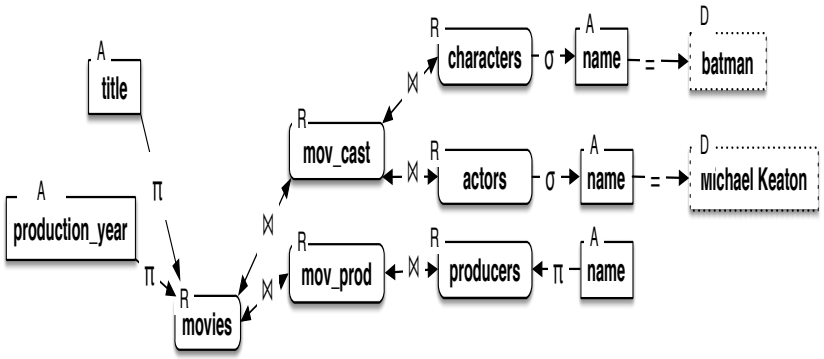
- table, attribute, domain and function nodes,
- projection, predicate, selection, join and transformation edges,
- subquery edges.

Definition

A query is covered by the COKE system if it forms a connected subgraph of the query graph.

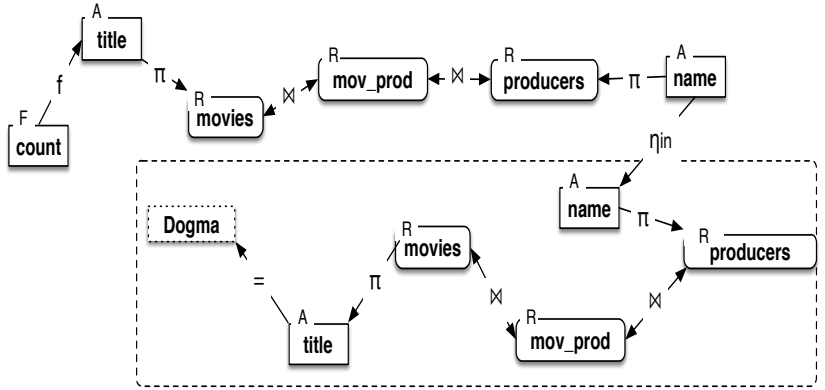
Query Graph Model: mapping between the model and SQL

'name of producers, title and production year of movies with Michael Keaton in the role of Batman'



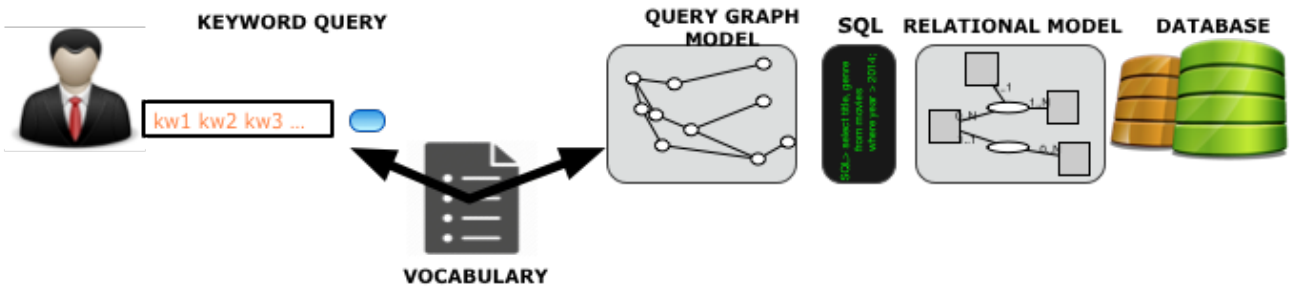

```
SQL> SELECT title, production_year, P.name
FROM producer P join movies M on M.pid = P.id
      join mov_cast MC on MC.mid = M.id
      join characters C on C.id = MC.cid
      join actors A on A.id = MC.aid
WHERE a.name = 'Keaton'
      and C.name = 'batman';
```

'number of movies produced by the producer of the movie entitled Dogma'




```
SQL> SELECT count(*)
FROM movies M join mov_prod MP on M.id = MP.mid
      join producers P on P.id = MP.pid
WHERE P.name = ( SELECT P.name
     FROM producers P2 join mov_prod MP2 on
     P2.id = MP2.id
     join movies M2 on M2.id = MP2.mid
     WHERE M2.title = 'Dogma');
```

Searchable Vocabulary: mapping keywords to elements of the query graph

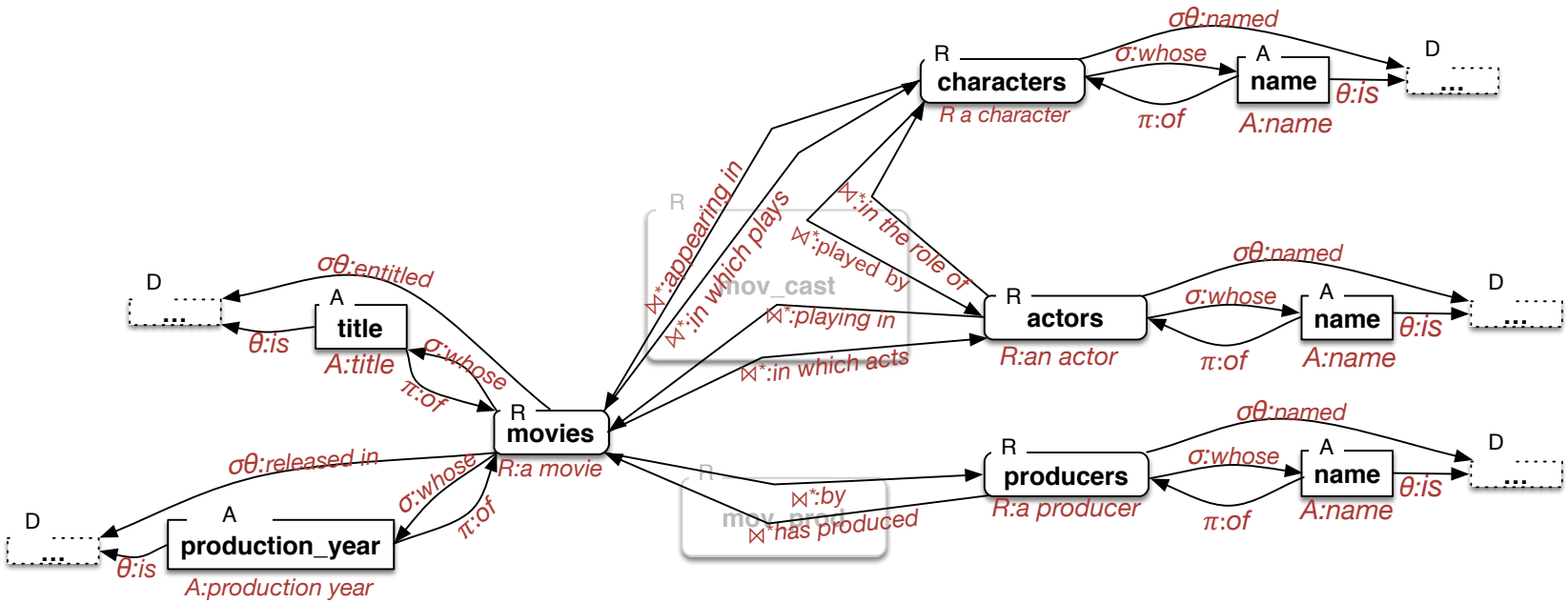


Definition

A keyword is a word or a group of words attached to a searchable element of the query graph:

- table, attribute, domain and function nodes,
- projection, predicate, join, selection and transformation edges,
- subqueries edges,
- and some paths (predicate + selection, joins, joins + selection).

Searchable Vocabulary: mapping keywords to elements of the query graph



Extract of the labelled query graph

COKE: a Constrained Keyword-based Query Language

A COKE query is a keyword-based expression of a selection-projection-join SQL query:

- with a restricted vocabulary,
- and a constrained syntax *(projection part + selection part)*.

Backus Naur form grammar:

COKE_query ::= {Projection_Statement}* {Selection_Statement}*

Projection_Statement ::= $F^f\ [e^f]\ A_l^k\ e^\pi\ R^k \mid F^f\ [e^f]\ R^k \mid [Attributes\ [e^\pi]]\ R^k \mid Attributes$

Attributes := $A_l^k\ \{,\ A_i^k\}*\ [and\ A_j^k]$

Selection_Statement ::= $\{e^\bowtie | p^{\bowtie^*}\}\ R^k\ e^\sigma\ A_l^k\ e^\theta\ D_l^k \mid A_l^k\ e^\pi\ R^k\ e^\theta\ D_l^k \mid \{e^\bowtie | p^{\bowtie^*}\}\ R^k\ p^{\sigma\theta}\ D_l^k \mid A_l^k\ e^\theta\ D_l^k \mid R^k\ p^{\sigma\theta}\ D_l^k \mid p^{\sigma\theta}\ D_l^k \mid p^{\bowtie^*\sigma\theta}\ D_l^k \mid R^k\ D_l^k \mid D_l^k \mid e^{\eta^{op}}\ D_l^k.$

The COKE Syntax Illustrated

COKE query = a projection part + a selection part

Projection statement:

- basic form: "movies", "title year movies", "number movies", etc.,
- more expressive syntax: "title, length and genre of movies", "number of movies", etc.

Selection statement (joins, restrictions, some nested queries):

- basic form: "Clint E. Gene H.", "actor Clint E.", "producer Clint E. actor Gene H.", etc.,
- more expressive syntax:. "by a producer whose name is Clint E.", "produced by Clint E. with Gene H.", "with an actor of Dogma", etc.

COKE: a Cooperative Querying Approach

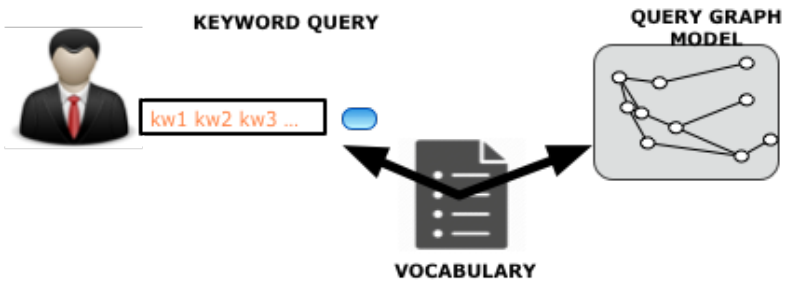
A constrained keyword-based query language with:

1. a restricted vocabulary,
2. a limited syntax,
3. and semantic expectations (connected subgraph of the query graph).

An interactive query construction and interpretation process to:

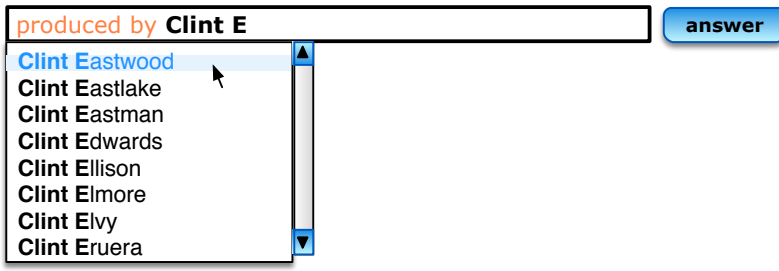
- help users express their information needs with covered queries,
- interactively determine the exact meaning of the queries.

Query Definition at the Lexical Level



Autocompletion and locked query field to ensure lexical coverage:

- index of the whole vocabulary.



Definition

The lexical analyzer (made easy by autocompletion) produces mappings between keywords and elements of the query graph.

Query Interpretation at the Syntactic Level

Parsing to identify valid projection and selection statements

Definition

A syntactic analysis is a mapping from adjacent keywords to projection/selection graph patterns.

COKE_query ::= {Projection_Statement}* {Selection_Statement}*
Projection_Statement ::= $F^f\ [e^f]\ A_l^k\ e^\pi\ R^k \mid F^f\ [e^f]\ R^k \mid [Attributes\ [e^\pi]]\ R^k \mid Attributes$
Attributes := $A_l^k\ \{,\ A_i^k\}*\ [and\ A_j^k]$

Selection_Statement ::= $\{e^{\bowtie} | p^{\bowtie^*}\}\ R^k\ e^\sigma\ A_l^k\ e^\theta\ D_l^k \mid A_l^k\ e^\pi\ R^k\ e^\theta\ D_l^k \mid \{e^{\bowtie} | p^{\bowtie^*}\}\ R^k\ p^{\sigma\theta}\ D_l^k \mid A_l^k\ e^\theta\ D_l^k \mid R^k\ p^{\sigma\theta}\ D_l^k \mid p^{\sigma\theta}\ D_l^k \mid p^{\bowtie^*\sigma\theta}\ D_l^k \mid R^k\ D_l^k \mid D_l^k \mid e^{\eta^{op}}\ D_l^k$.

Example:

$\langle$ ('title production year of movie', $\{A_l^k\}+\ e^\pi\ R^k$) ('with an actor named Mickael Keaton', $e^{\bowtie}\ R\ p^{\sigma\theta}\ D$) ('in the role of Batman', $p^{\bowtie^*\sigma\theta} D_l^k$)$\rangle$.

Interactive Syntactic Disambiguation

Some syntactic patterns may be ambiguous or uncovered:

- ask for a confirmation of the right interpretation,
  - expansion of ambiguous patterns.
- show syntactically uncovered parts of the query.

budget genre name birth date entitled avatar language in 2009 **answer**

Information to be displayed

- budget and genre of a movie ☑
- name and birth date of an actor ☑
- name of a producer ☐
- name of a director ☐
- name of a character ☐
- name of a composer ☐
- name of a cinematographer ☐
- name of an editor ☐

Satisfying the criteria

- entitled avatar ☑
- production year of a movie is 2009 ☑
- birth date of an actor is 2009 ☐
- death date of an actor is 2009 ☐
- name of a character is 2009 ☐

What do you mean by …

- language

Discovering the Meaning of the Query

Definition

A query is semantically consistent if its interpretation forms a connected subgraph of the query graph.

In case of isolated statements (e.g. 'title of movies name Clint E.'), re-establish the semantic consistency:

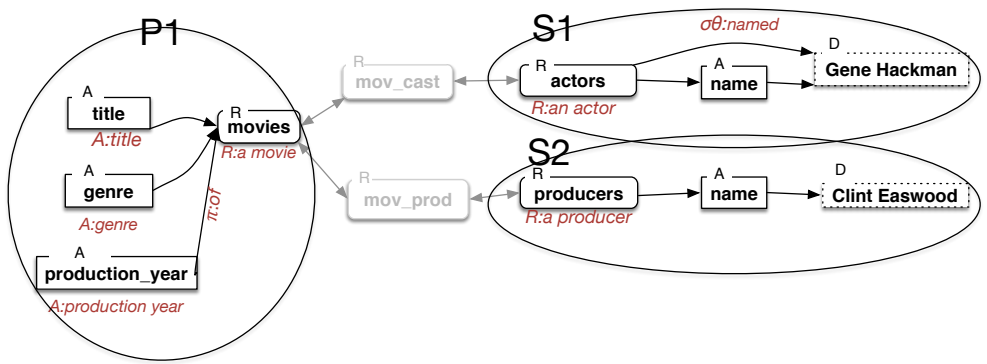
- identify the subject (relation node) of each statement,
  - projection statement $\rightarrow$ the concerned relation,
  - selection statement $\rightarrow$ the relation on which a restriction is applied (selection edge).
- determine meaningful joins,
  - link an isolated subject node to a previously activated subject,
  - favor the shortest join path between two subjects.

Discovering the Meaning of the Query (example)

"title genre and production year of movies actor named Gene H. producer Clint E."

Subject relation nodes: $\{movies, actors, producers\}$.

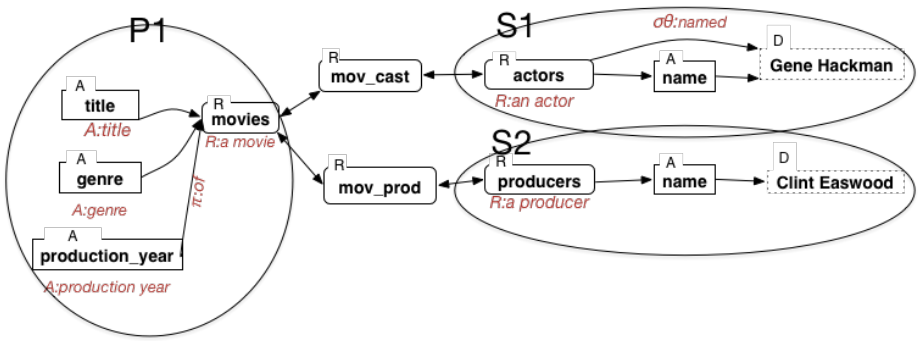
- projection statement,
  P^1 "title, genre and production year of movies"
- selection statements,
  S^1 "actor named Gene H."
  S^2 "producer Clint E."



Translation to SQL

Depth-first traversal of the connected subgraph:

- starting with projection statements,
- translation rules for each graph pattern,
- incremental filling of the *SELECT*, *FROM* and *WHERE* clauses.




```
SQL> SELECT movie.title, movies.genre, movie.production year
FROM movies, mov cast, actors, mov prod, producers
WHERE movies.idM = mov cast.idM and mov cast.idA = actors.idA and
actors.name = 'Gene Hackman' and movies.idM = mov prod.idM and
mov prod.idP = producers.idP and producers.name = 'Clint Eastwood';
```

Translation rules:

- $A\ \pi\ R\rightarrow$ append A to *SELECT* and R to *FROM*,
- $R^1 \bowtie R^2$ append a natural join between R^1 and R^2 to *FROM*,
- $A\ \theta\ D$ append $A\theta D$ to *WHERE*.

Experimentations

Implementation on top of the MusicBrainz DB:

- manual definition of the query graph and vocabulary.

Expressivity assessment:

- translation of the QALD corpus (100 NL queries):
  - 81 queries successfully translated into COKE queries,
  - 8 concern data not available in the DB version of MusicBrainz,
  - 11 uncovered (6 Boolean queries and 5 relying on indirect knowledge).

A first attempt of precision assessment (precision 100%):

- translation by a "COKE" queries expert,
- using unambiguous statements, no join deduction,
- basic queries (80/81 queries contain 1 projection and 1 selection).

Efficiency assessment:

- NL approach: avg. 31.6s, COKE: avg. 0.65s.

Conclusion and Perspectives

A keyword-based query approach taking connectives into account:

- constrained keyword-based language,
- interactive query definition and disambiguation strategy.

A tool to help domain experts query their corporate DBs

Perspectives:

- make COKE queries closer to natural language,
- automatic acquisition of the vocabulary,
- handle more complex queries (aggregates, groups),
- real users feedbacks in a CRM applicative context.