Answering Queries Using Views: A Survey

Alon Y. Halevy etc

Qun Li

Introduction

- Given a Query Q and a set of View V
  - Is it possible to answer the query Q using only the answers to the views in V?
  - What is the maximal set of tuples in the answer of Q that we can obtain from the views?

Wide area of interest

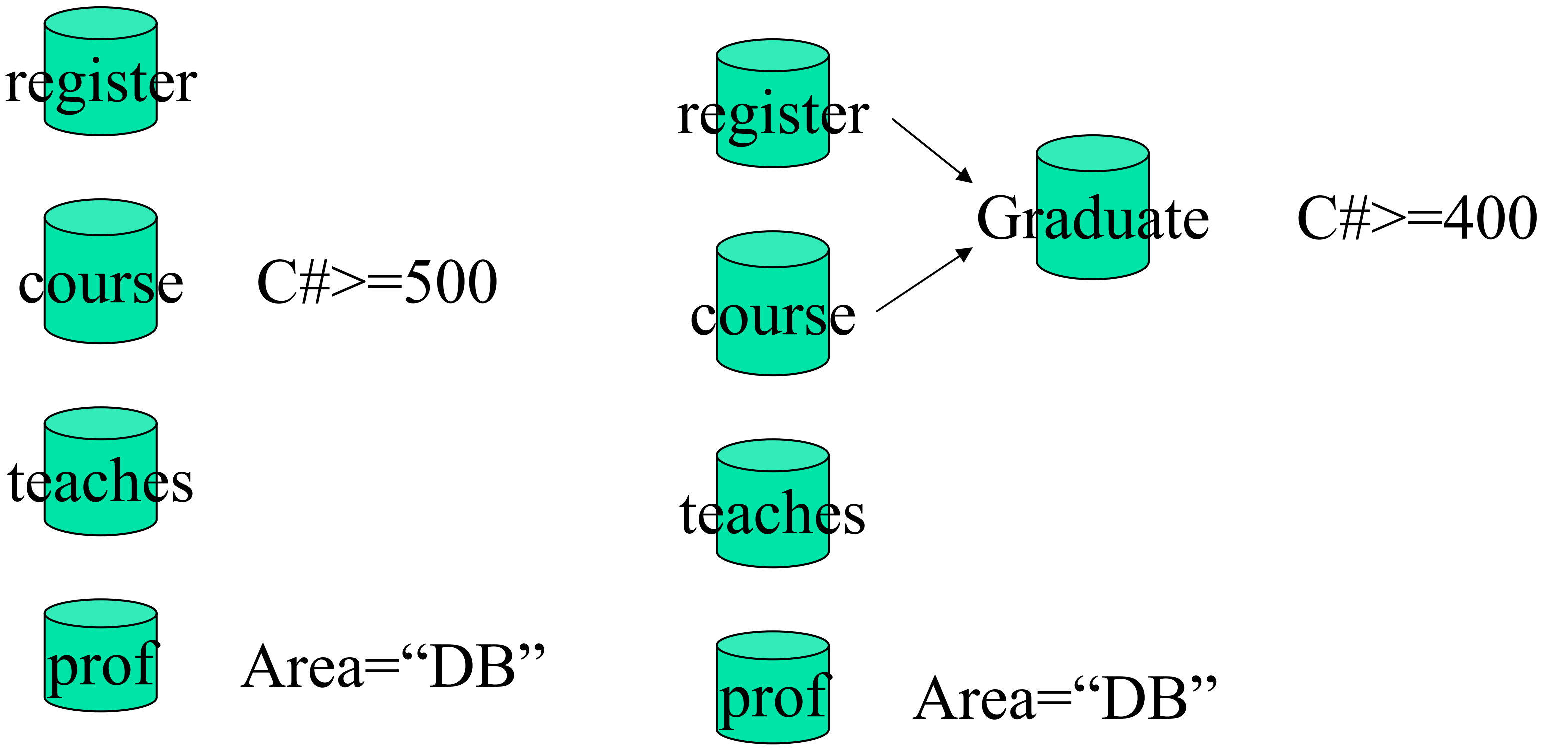
- Query optimization
- Maintenance of physical data independence
- Data warehouse design
- Data integration

Query optimization

- Speed up query processing
  - Grouping
  - Aggregation
- Not necessary always
  - Indexes
- Complete query results are desired

Example

Find students and course titles for students who registered In PH.D level classes taught by professors in DB area



Physical Data Independence

- Separation between the logical view of the data and the physical view of the data
  - Logical schema contains significant redundancy(OODB)
  - 1 to 1 ??
  - Physical representation has already been determined.
  - Logical view vs physical view

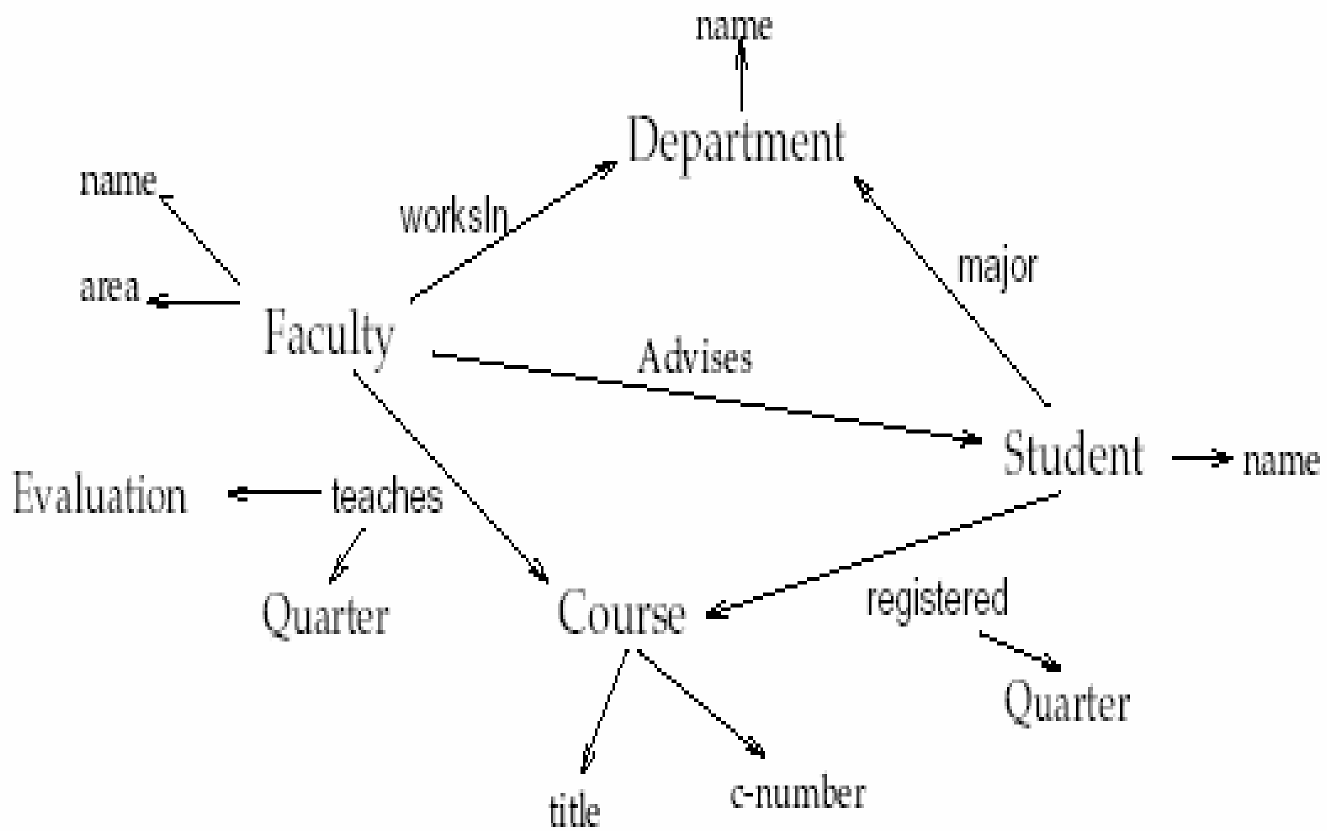
GMAP

Def_gmap G1 as B+ tree by
Given Student.name
Select Department
Where Student major Department

Def_gmap G2 as b+ tree by
Given Student.name
Select Course.c-number
Where Student registered Course

Def_gmap G3 as B+ tree by
Given Course.c-number
Select Department
Where Student registered Course and Student major Department.

GMAP



name
Department
name
worksIn
area
major
Faculty
Advises
Student
name
Evaluation
teaches
Quarter
Course
registered
Quarter
title
c-number

Data warehouse design

- Choose a set of views to materialize
- View
  - view are relations, except that they are not physically stored. they are computed only on-demand, and always up to date
- materialized views
  - used in date warehouses
  - precomputed offline (fast at runtime)
  - may have stale data (like cache)

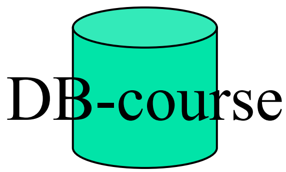
Data integration

- Multiple heterogeneous sources
  - Traditional databases
  - Legacy systems
  - Structured files
- Mediated schema
  - A Set of source description
  - Not stored anywhere
- Goal
  - Uniform query interface
  - User freedom
- Maximally contained answers is the only option sometimes

equivalent or maximal

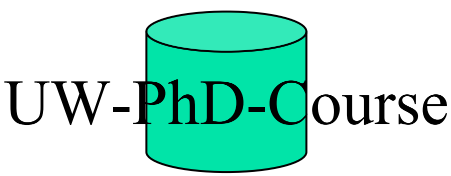
- Equivalent rewritings
  - In the contexts of query optimization
  - Maintenance of physical data independence
- Maximally-contained rewriting
  - In the context of data integration.
  - The available views may not contain the complete answer
  - The views may be incomplete(open world assumption)

Example



DB taught anywhere

Select prof
From DB-courses
Where univ="UW"



Select title, c-number
From DB_courses
Where univ="UW" and c-number>=400
UNION
Select title.c-number
From UW-phd-courses

AQUV for Data Integration

- May not be able to find an equivant rewriting
- Maximally-contained rewriting
- A union of several queries over the sources.

Algorithms

- Key challenge
  - Develop an algorithm that scales up in the number of views
  - Search through a possibly exponential number of rewritings
  - NP-Complete
- Bucket algorithm
- Inverse-rule algorithm
- MiniCon algorithm

Bucket Algorithm

- primarily used in the "Information Manifold" system
- exploits the content description to translate the original query into particular queries of each source
- Consider each sub-goal in isolation
- return the maximally contained rewritings of a query

Steps

- computes for each subgoal in the query the source that are relevant to it
- considers rewritings constructed by choosing one relevant source for every subgoal in the query, and checks whether its expansion is allowed.
- combine one element of every bucket
- expansion of a rewriting
- check whether the expansion is allowed.

Bucket Algorithm

- Combination step has several deficiencies
  - Need to perform a query containment test for every candidate rewriting
- Cartesian product of the buckets may still be rather large
- Does not scale up well.

Inverse-rule algorithm

- construct a set of rules that invert the view definition(skolem function)
- Invert view definitions:
  - Turn view tuples into "facts" in the database that can be used to reconstruct base tables and answer queries.
- conceptual simplicity and modularity
- Has to re-evaluate the views from their inverted definitions

Example

- Schema
  - Cites(a,b)
  - sameTopic(c,d)
- Query and views
  - Q1(x) :- cites(x,y),cites(y,x),sameTopic(x,y)
  - V4(a) :-cites(a,b),cites(b,a)
  - V5(c,d) :-sameTopic(c,d)
  - V6(f,h) :-cites(f,g),cites(g,h),sameTopic(f,g)

Example

- R1: cites(a,f1(a)):-V4(a)
- R2: cites(f1(a),a):-V4(a)
- R3: sameTopic(c,d) :-V5(c,d)
- R4: cites(f,f2(f,h)):-V6(f,h)
- R5: cites(f2(f,h),h):-V6(f,h)
- R6: sameTopic(f,f2(f,h):-V6(f,h)

Inverse-rule algorithm

- Attempt to find more effective methods to produce rewriting that do not require exhaustive search
- Inverse rules can be constructed ahead of time in polynomial time. Independent of a particular query
- Requires reconstructing the base tables(the performance advantage of using materialized views is lost

Mini-con

- Welcome STAN

Conclusion

- Using view to answer queries is an important problem.Especially for data integration.
- It is hard (NP-complete)
- Many issues remain open