CSE 562
Database Systems

Query Processing: Semantic Optimization

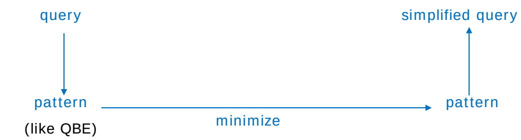
Slides are based or modified from originals by
Victor Vianu

cse@buffalo

Exact Minimization of # of Joins

Possible for large set of queries:

- Basic, unnested SQL queries
- One-stage QBE queries
- Relational algebra with π, σ, $\bowtie$
- Relational calculus with $\exists$, $=$, $\wedge$
- Basic idea:



Minimization for a Simple Case

- Relational algebra with:
  - one input relation (can be extended to several)
  - π, $\bowtie$
  - σ_{cond} where cond is A = const
    - → rspj query
- First step: write query in QBE style, using a pattern → tableau
  - e.g. tableau for $\pi_{AB}(R) \bowtie \pi_{BC}(R)$, where R: ABC, is:

A	B	C	
a	b	c_1	pattern
a_1	b	c	pattern
a	b	c	answer

Answer to a Tableau Query: Example

The answer to the tableau query

A	B	C
a	b	c_1
a_1	b	c
a	b	c

applied to

A	B	C
0	1	2
3	1	4
5	6	7

is:

A	B	C
0	1	4
3	1	2
0	1	2
3	1	4
5	6	7

For instance, the tuple <3,1,2> is obtained by the following mapping:

A	B	C
a	b	c_1
a_1	b	c
a	b	c

A	B	C
0	1	2
3	1	4
5	6	7

a → 3
b → 1
c_1 → 4
a_1 → 0
c → 2

Tableaux

- Tableau over R:<S, t> where
  - S is the set of rows over R, with variables or constants
  - t is the "answer" row

  t is over a subset of att(R)

  t(A) is a variable or a constant

- Terminology
  - variables in t: distinguished (free)
    - denoted a, b, c
  - other variables: non-distinguished (quantified)
    - denoted a_1 a_2 ... b_1 b_2 ... c_1 c_2

Example

- Directors who are also actors
- QBE:

movie	title	dir	actor
		_d	
			_d

answer	dir
I.	_d

- Tableau:

	title	dir	actor	
S	t_1	d	a_1	
S	t_2	d_1	d	
t		d		answer row

From rspj-Algebra Queries to Tableaux

Example

R: ABC

q: $\pi_{AC}(\pi_{AB}(R) \bowtie \pi_{BC}(\sigma_{A=5}(\pi_{AB}(R)) \bowtie \pi_{AC}(R)))$

Relational calculus (caution: use a different variable with each quantifier to avoid ambiguity!):

$\exists b_2[\exists c_1(R(ab_2c_1)) \wedge \exists a_1(\exists c_2(R(a_1b_2c_2)) \wedge a_1=5$
$\wedge\ b_1(R(a_1b_1c)))]$

Prenex form (move $\exists$ to left):

$\exists a_1b_1b_2c_1c_2[R(ab_2c_1) \wedge R(a_1b_2c_2) \wedge R(a_1b_1c) \wedge a_1=5]$

Replace a_1 by 5:

$\exists b_1b_2c_1c_2[R(ab_2c_1) \wedge R(5,b_2c_2) \wedge R(5,b_1c)]$

Example (cont.)

Tableau:

A	B	C
a	b_2	c_1
5	b_2	c_2
5	b_1	c
a		c

Note: # rows in tableau = 1 + # joins

Note: like QBE query

R	A	B	C
	_a	$\_b_2$	
	5	$\_b_2$	
	5		_c

answer	A	C
I.	_a	_c

Minimizing Tableau

- <S, t>: tableau
- Definition
  - Mapping **f** on variables is a homomorphism on <S, t> iff:
    - **f**(t) = t
    - **f**(c) = c if c is constant
    - **f**(S) $\subseteq$ S (every row is mapped to an existing row in S)

Theorem: <**f**(S), t> is equivalent to <S, t>

  - Therefore, a row r$\notin$**f**(S) is redundant
  - **Minimization algorithm**: eliminate redundant rows until no longer possible

Example

- Tableau <S, t>

A	B	C	
a	b_1	c_1	
a_1	b	c_1	
a	b_2	c_2	S
a_2	b_2	c	
a_2	b_1	c	
a	b	c	t

- Let **f** be defined by $c_2 \rightarrow c_1$
  $b_2 \rightarrow b_1$
  all other variables stay unchanged

Example (cont.)

- **f**(<S, t>):

A	B	C
a	b_1	c_1
a_2	b_1	c
a_1	b	c_1
a	b	c

No redundant rows: MINIMAL

- Fact: all minimal equivalent tableaux are isomorphic!
  So the algorithm yields the minimum possible number of rows (and joins).

From Tableau Back to Algebra

Example

A	B	C
a	b	c_1
a_1	b	c
a	b	c

- Write domain calculus query for the tableau:
  $\exists c_1 \exists a_1 \ (R(abc_1) \wedge R(a_1bc))$
- Translate to algebra:
  $\pi_{ABC} [\ \delta_{C\rightarrow C1}(R) \bowtie \delta_{A\rightarrow A1}(R)]$
- Renaming can **always** be avoided by quantifying (and projecting) as early as possible:
  $\exists c_1 \exists a_1 \ (R(abc_1) \wedge R(a_1bc))$ is the same as
  $\exists c_1 \ (R(abc_1)) \wedge \exists a_1 \ (R(a_1bc))$ which translates to
  $\pi_{AB}(R) \bowtie \pi_{BC}(R)$

Example

R: ABC

q: $\pi_{AB}(\sigma_{B=5}(R)) \bowtie \pi_{BC}(\pi_{AB}(R) \bowtie \pi_{AC}(\sigma_{B=5}(R)))$

Tableau:

A	B	C
a	5	c_1
a_1	5	c_2
a_1	5	c
a	5	c

Minimization
$c_2 \rightarrow c$

↓

A	B	C
a	5	c_1
a_1	5	c
a	5	c

↓

$\sigma_{B=5}(\pi_{AB}(R) \bowtie \pi_{BC}(R))$ — Minimized query

Functional Dependencies

- **Dependencies:** statements about properties of valid data
  - e.g.: "Every student is a person"
    - inclusion dependency
  - "Each employee works in no more than one department"
    - NAME → DEPARTMENT
    - functional dependency
- Use of dependencies:
  - check data integrity
  - query optimization
  - schema design → "normal forms"

Functional Dependencies

- Functional dependency over R:
  - expression $X \rightarrow Y$ where $X, Y \subseteq att(R)$
- A relation R satisfies $X \rightarrow Y$ iff whenever two tuples in R agree on X, they also agree on Y

e.g.

SCHEDULE	THEATER	TITLE
	la jolla	killer tomatoes
	hillcrest	tango

Satisfies THEATER → TITLE

SCHEDULE	THEATER	TITLE
	la jolla	killer tomatoes
	hillcrest	tango
	hillcrest	splendor

Violates THEATER → TITLE, satisfies TITLE → THEATER

Using FDs in Query Optimization

- Example: R: ABC with $B \rightarrow C$
  - query $\pi_{AB}(R) \bowtie \pi_{BC}(R)$
- Fact: if R satisfies $B \rightarrow C$ then
  - $\pi_{AB}(R) \bowtie \pi_{BC}(R) = R$
  - Why: tableau of query is

A	B	C
a	b	c_1
a_1	b	c
a	b	c

  - If $<a, b, c_1> \in R$ and $<a_1, b, c> \in R$ then $c_1 = c$ since R satisfies $B \rightarrow C$
  - So, tableau is equivalent to

A	B	C
a	b	c
a	b	c

$\equiv R$

The Chase

- In general: can simplify tableau <S, t> over R if R satisfies a set F of FDs
- Algorithm: The Chase
  - Input: tableau <S, t>, set F of FDs
  - Output: tableau $CHASE_F$<S, t> on all relations satisfying F
- Note: assume without loss of generality that FDs in F are of the form $X \to A$ where A is one attribute

The Chase

- Repeat until no change
  - For each $X \to A$ in F do
    - For each t_1, t_2 in S such that $t_1(X) = t_2(X)$, $t_1(A) \neq t_2(A)$ do
      - if $t_1(A)$, $t_2(A)$ are non-distinguished then replace one by the other in S
      - if $t_1(A)$ distinguished, $t_2(A)$ non-distinguished then replace $t_2(A)$ by $t_1(A)$ in S
      - if $t_1(A)$ is constant, $t_2(A)$ is variable then replace $t_2(A)$ by $t_1(A)$ in S
      - if $t_1(A)$ is constant, $t_2(A)$ is constant then STOP and output $\varnothing$

Optimization of RSPJ Queries with FDs

- q over R, set of FDs F over R
  - build tableau <S, t> of q
  - compute $CHASE_F$<S, t>
  - minimize $CHASE_F$<S, t>
  - construct rspj query from minimal tableau

Example

- R: ABC F = $\{B \to A\}$
- $q = \pi_{BC}(\sigma_{A=5}(R)) \bowtie \pi_{AB}(R)$

<S, t>:

A	B	C
5	b	c
a	b	c_1
a	b	c

CHASE<S, t>:

A	B	C
5	b	c
5	b	c_1
5	b	c

MIN:

A	B	C
5	b	c
5	b	c

RSPJ: $\sigma_{A=5}(R)$

Example

– R: ABC F = {B → A}, $q = \pi_{BC}(\sigma_{A=5}(R)) \bowtie \pi_{AB}(\sigma_{A=6}(R))$

<S, t>:

A	B	C
5	b	c
6	b	c_1
6	b	c

CHASE<S, t>: ∅

QUERY: ∅

Example

– R: ABC F = {A → B},
$q = \pi_{AB}(R) \bowtie \pi_{A}(\sigma_{B=5}(R)) \bowtie \pi_{AB}(\pi_{AC}(R) \bowtie \pi_{BC}(R))$

<S, t>:

A	B	C
a	b	c_1
a	b_1	c_2
a_1	b	c_2
a	5	c_3
a	b	

CHASE<S, t>:

A	B	C
a	5	c_1
a	5	c_2
a_1	5	c_2
a	5	c_3
a	5	

MIN:

A	B	C
a	5	c_1
a	5	

RSPJ: $\pi_{AB}(\sigma_{B=5}(R))$