Query Optimization

Query Processing

Q → Query Plan

Focus: Relational System

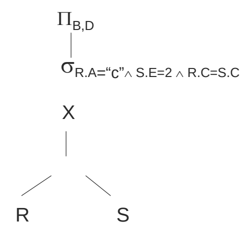
- Others?

Example

Select B,D
From R,S
Where R.A = “c” $\wedge$ S.E = 2 $\wedge$ R.C=S.C

Relational Algebra - can be used to describe plans...

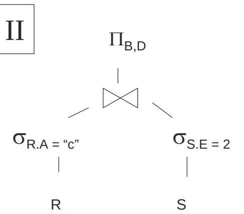
Ex: Plan I



OR: $\Pi_{B,D}\,[\sigma_{R.A=\text{"c"} \wedge S.E=2 \wedge R.C = S.C}\,(R X S)]$

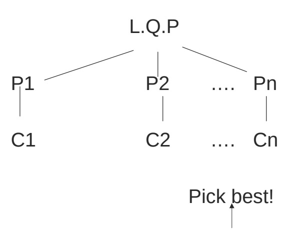
Another idea:

Plan II



$\bowtie$ natural join

Example: Estimate costs



Relational algebra optimization

- Transformation rules
  (preserve equivalence)
- What are good transformations?

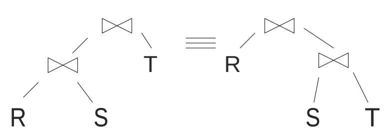
Rules: Natural joins & cross products & union

$R \bowtie S \quad = \quad S \bowtie R$

$(R \bowtie S) \bowtie T \quad = R \bowtie (S \bowtie T)$

Note:

- Carry attribute names in results, so order is not important
- Can also write as trees, e.g.:



Rules: Natural joins & cross products & union

$R \bowtie S = S \bowtie R$

$(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$

R x S = S x R

(R x S) x T = R x (S x T)

R U S = S U R

R U (S U T) = (R U S) U T

Rules: Selects

$\sigma_{p1 \wedge p2}(R) = \sigma_{p1} [\sigma_{p2} (R)]$

$\sigma_{p1 \vee p2}(R) = [\sigma_{p1} (R)] \cup [\sigma_{p2} (R)]$

Rules: Project

Let: X = set of attributes

Y = set of attributes

XY = X U Y

$\pi_{xy} (R) =$

~~$\pi_x [\pi_y (R)]$~~

Rules: σ + ⋈ combined

Let p = predicate with only R attribs
q = predicate with only S attribs
m = predicate with only R,S attribs

$\sigma_p (R \bowtie S) = [\sigma_p (R)] \bowtie S$

$\sigma_q (R \bowtie S) = R \bowtie [\sigma_q (S)]$

Rules: σ + ⋈ combined (continued)

Some Rules can be Derived:

$\sigma_{p \wedge q} (R \bowtie S) =$

$\sigma_{p \wedge q \wedge m} (R \bowtie S) =$

$\sigma_{p \vee q} (R \bowtie S) =$

$\sigma_{p \wedge q} (R \bowtie S) = [\sigma_p (R)] \bowtie [\sigma_q (S)]$

$\sigma_{p \wedge q \wedge m} (R \bowtie S) =$

$\sigma_m \left[(\sigma_p R) \bowtie (\sigma_q S)\right]$

$\sigma_{p \vee q} (R \bowtie S) =$

$\left[(\sigma_p R) \bowtie S\right] \cup \left[R \bowtie (\sigma_q S)\right]$

Rules: π, σ combined

Let x = subset of R attributes

z = attributes in predicate P

(subset of R attributes)

$\pi_x[\sigma_{p}(R)] = \pi_x\{\sigma_p[\pi_{xz}(R)]\}$

Rules: $\pi, \bowtie$ combined

Let x = subset of R attributes

y = subset of S attributes

z = intersection of R,S attributes

$\pi_{xy}(R \bowtie S) = \pi_{xy}\{[\pi_{xz}(R)] \bowtie [\pi_{yz}(S)]\}$

$\pi_{xy}\{\sigma_p(R \bowtie S)\} =$

$\pi_{xy}\{\sigma_p[\pi_{xz'}(R) \bowtie \pi_{yz'}(S)]\}$

z' = z U {attributes used in P}

Rules for σ, π combined with X

similar...

e.g., $\sigma_p(R \times S) = ?$

Rules $\sigma, \cup$ combined:

$\sigma_p(R \cup S) = \sigma_p(R) \cup \sigma_p(S)$

$\sigma_p(R - S) = \sigma_p(R) - S = \sigma_p(R) - \sigma_p(S)$

Which are "good" transformations?

- $\sigma_{p1 \wedge p2}(R) \rightarrow \sigma_{p1}[\sigma_{p2}(R)]$
- $\sigma_p(R \bowtie S) \rightarrow [\sigma_p(R)] \bowtie S$
- $R \bowtie S \rightarrow S \bowtie R$
- $\pi_x[\sigma_p(R)] \rightarrow \pi_x\{\sigma_p[\pi_{xz}(R)]\}$

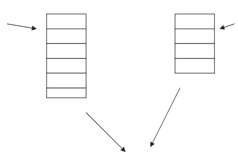
Conventional wisdom: do projects early

Example: R(A,B,C,D,E) x={E}
P: (A=3) $\wedge$ (B="cat")

$\pi_x\{\sigma_p (R)\}$ vs. $\pi_E\{\sigma_p\{\pi_{ABE}(R)\}\}$

But What if we have A, B indexes?

B = "cat" A=3



Intersect pointers to get pointers to matching tuples

Bottom line:

- No transformation is always good
- Usually good: early selections

• Estimating cost of query plan

(1) Estimating size of results
(2) Estimating # of IOs

Estimating result size

- Keep statistics for relation R
  - T(R) : # tuples in R
  - S(R) : # of bytes in each R tuple
  - B(R): # of blocks to hold all R tuples
  - V(R, A) : # distinct values in R for attribute A

Example

R

A	B	C	D
cat	1	10	a
cat	1	20	b
dog	1	30	a
dog	1	40	c
bat	1	50	d

A: 20 byte string
B: 4 byte integer
C: 8 byte date
D: 5 byte string

T(R) = 5 S(R) = 37
V(R,A) = 3 V(R,C) = 5
V(R,B) = 1 V(R,D) = 4

Size estimates for W = R1 x R2

$T(W) = T(R1) \times T(R2)$

$S(W) = S(R1) + S(R2)$

Size estimate for W = $\sigma_{A=a}(R)$

$S(W) = S(R)$

$T(W) = ?$

Example

R

A	B	C	D
cat	1	10	a
cat	1	20	b
dog	1	30	a
dog	1	40	c
bat	1	50	d

V(R,A)=3
V(R,B)=1
V(R,C)=5
V(R,D)=4

$W = \sigma_{z=val}(R) \quad T(W) = \frac{T(R)}{V(R,Z)}$

Assumption:

Values in select expression Z = val
are uniformly distributed
over possible V(R,Z) values.

Alternate Assumption:

Values in select expression Z = val
are uniformly distributed
over domain with DOM(R,Z) values.

Example

R

A	B	C	D
cat	1	10	a
cat	1	20	b
dog	1	30	a
dog	1	40	c
bat	1	50	d

Alternate assumption
V(R,A)=3 DOM(R,A)=10
V(R,B)=1 DOM(R,B)=10
V(R,C)=5 DOM(R,C)=10
V(R,D)=4 DOM(R,D)=10

$W = \sigma_{z=val}(R)$ T(W) = ?

C=val ⇒ T(W) = (1/10)1 + (1/10)1 + ...
= (5/10) = 0.5

B=val ⇒ T(W)= (1/10)5 + 0 + 0 = 0.5

A=val ⇒ T(W)= (1/10)2 + (1/10)2 + (1/10)1
= 0.5

Example

R

A	B	C	D
cat	1	10	a
cat	1	20	b
dog	1	30	a
dog	1	40	c
bat	1	50	d

Alternate assumption
V(R,A)=3 DOM(R,A)=10
V(R,B)=1 DOM(R,B)=10
V(R,C)=5 DOM(R,C)=10
V(R,D)=4 DOM(R,D)=10

$$W = \sigma_{z=val}(R) \quad T(W) = \frac{T(R)}{DOM(R,Z)}$$

Selection cardinality

SC(R,A) = average # records that satisfy equality condition on R.A

$$SC(R,A) = \begin{cases} \dfrac{T(R)}{V(R,A)} \\ \dfrac{T(R)}{DOM(R,A)} \end{cases}$$

What about W = $\sigma_{z \geq val}$ (R) ?

T(W) = ?

- Solution # 1:
  T(W) = T(R)/2

- Solution # 2:
  T(W) = T(R)/3

- Solution # 3: Estimate values in range

Example R

	Z

Min=1 ↕ Max=20

V(R,Z)=10

W= $\sigma_{z \geq 15}$ (R)

$f = \frac{20-15+1}{20-1+1} = \frac{6}{20}$ (fraction of range)

T(W) = f × T(R)

Equivalently:

f×V(R,Z) = fraction of distinct values

$T(W) = [f \times V(Z,R)] \times \frac{T(R)}{V(Z,R)} = f \times T(R)$

Size estimate for W = R1 ⋈ R2

Let x = attributes of R1

y = attributes of R2

Case 1

$X \cap Y = \emptyset$

Same as R1 x R2

1/14/2005 Yan Huang - CSCI5330 Database Implementation – Query Optimization

---

Case 2

W = R1 ⋈ R2 $X \cap Y = A$

R1 | A | B | C R2 | A | D

Assumption:

V(R1,A) $\leq$ V(R2,A) $\Rightarrow$ Every A value in R1 is in R2

V(R2,A) $\leq$ V(R1,A) $\Rightarrow$ Every A value in R2 is in R1

"containment of value sets"

1/14/2005 Yan Huang - CSCI5330 Database Implementation – Query Optimization

---

Computing T(W) when V(R1,A) $\leq$ V(R2,A)

R1 | A | B | C R2 | A | D

Take 1 tuple → Match

1 tuple matches with $\frac{T(R2)}{V(R2,A)}$ tuples...

so $T(W) = \frac{T(R2)}{V(R2, A)} \times T(R1)$

1/14/2005 Yan Huang - CSCI5330 Database Implementation – Query Optimization

- $V(R1,A) \leq V(R2,A)$ $T(W) = \dfrac{T(R2)\,T(R1)}{V(R2,A)}$

- $V(R2,A) \leq V(R1,A)$ $T(W) = \dfrac{T(R2)\,T(R1)}{V(R1,A)}$

[A is common attribute]

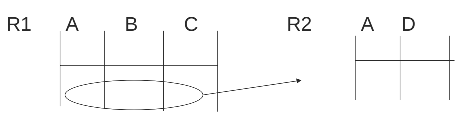
<u>In general</u> $W = R1 \bowtie R2$

$$T(W) = \frac{T(R2)\,T(R1)}{\max\{ V(R1,A), V(R2,A) \}}$$

Case 2

with alternate assumption

Values uniformly distributed over domain



This tuple matches T(R2)/DOM(R2,A) so

$$T(W) = \frac{T(R2)\,T(R1)}{DOM(R2, A)} = \frac{T(R2)\,T(R1)}{DOM(R1, A)}$$

Assume the same

In all cases:

S(W) = S(R1) + S(R2) - S(A)

size of attribute A