Query Optimization

Query Processing

Q → Query Plan

Focus: Relational System

- Others?

Example

Select B,D
From R,S
Where R.A = "c" ∧ S.E = 2 ∧ R.C=S.C
Relational Algebra can be used to describe plans...

Ex: Plan I

\[ \Pi_{B,D} \sigma_{R.A = 'c'} \land S.E = 2 \land R.C = S.C } \]

OR: \[ \Pi_{B,D} [\sigma_{R.A = 'c'} \land S.E = 2 \land R.C = S.C } (RXS)] \]

Another idea:

Plan II

\[ \Pi_{B,D} \sigma_{R.A = 'c'} \land \sigma_{S.E = 2} ] \]

natural join

Example: Estimate costs

L.Q.P

P1 P2 ..., Pn

C1 C2 ..., Cn

Pick best!
Relational algebra optimization

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

Rules: Natural joins & cross products & union

\[ R \bowtie S = S \bowtie R \]
\[ (R \bowtie S) \bowtie T = R \bowtie (S \bowtie T) \]

Note:
- Carry attribute names in results, so order is not important
- Can also write as trees, e.g.:
Rules: Natural joins & cross products & union

\[ R \bowtie S = S \bowtie R \]
\[ (R \bowtie S) \bowtie T = R \bowtie (S \bowtie T) \]
\[ R \times S = S \times R \]
\[ (R \times S) \times T = R \times (S \times T) \]
\[ R \cup S = S \cup R \]
\[ R \cup (S \cup T) = (R \cup S) \cup T \]

Rules: Selects

\[ \sigma_{p_1 \land p_2}(R) = \sigma_{p_1 \land \sigma_{p_2}(R)} \]
\[ \sigma_{p_1 \lor p_2}(R) = \sigma_{p_1}(R) \cup \sigma_{p_2}(R) \]

Rules: Project

Let: \( X = \) set of attributes
\( Y = \) set of attributes
\( XY = X \cup Y \)
\[ \pi_{XY}(R) = \pi_{X}(\pi_{Y}(R)) \]
Let $p$ = predicate with only $R$ attribs
$q$ = predicate with only $S$ attribs
$m$ = predicate with only $R,S$ attribs

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

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

Some Rules can be Derived:

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

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

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

$$\sigma_{p\land m} (R \bowtie S) = \sigma_m [(\sigma_p R) \bowtie (\sigma_q S)]$$

$$\sigma_{p\land q} (R \bowtie S) = [\sigma_p (R \bowtie S)] \cup [R \bowtie (\sigma_q S)]$$
Rules: π, σ combined

Let x = subset of R attributes
z = attributes in predicate P
(subset of R attributes)

\[ \pi_x[\sigma_p(R)] = \pi_x[\sigma_p(\pi_x(R))] \]

Let x = subset of R attributes
y = subset of S attributes
z = intersection of R, S attributes

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

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

\[ z' = z \cup \{\text{attributes used in } P\} \]
Rules for \(\sigma, \pi\) combined with \(X\)

similar...

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

\[\sigma_p (R \cup S) = \sigma_p (R) \cup \sigma_p (S)\]

\[\sigma_p (R - S) = \sigma_p (R) - S = \sigma_p (R) - \sigma_p (S)\]

Rules \(\sigma, U\) 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?

- \(\sigma_{p1 \land p2} (R) \rightarrow \sigma_{p1} [\sigma_{p2} (R)]\)
- \(\sigma_p (R \bowtie S) \rightarrow [\sigma_p (R)] \bowtie S\)
- \(R \bowtie S \rightarrow S \bowtie R\)
- \(\pi_x [\sigma_p (R)] \rightarrow \pi_x \{\sigma_p [\pi_{xz} (R)]\}\)
Conventional wisdom: do projects early

Example: $R(A,B,C,D,E)$  \( x=E \)

\[ P: (A=3) \land (B=\text{"cat"}) \]

\[ \pi_x \{ \sigma_p (R) \} \text{ vs. } \pi_E \{ \sigma_p(\pi_{AB}E(R)) \} \]

But what if we have A, B indexes?

\( B = \text{"cat"}, A=3 \)

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

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>1</td>
<td>10</td>
<td>a</td>
</tr>
<tr>
<td>cat</td>
<td>1</td>
<td>20</td>
<td>b</td>
</tr>
<tr>
<td>dog</td>
<td>1</td>
<td>30</td>
<td>a</td>
</tr>
<tr>
<td>dog</td>
<td>1</td>
<td>40</td>
<td>c</td>
</tr>
<tr>
<td>bat</td>
<td>1</td>
<td>50</td>
<td>d</td>
</tr>
</tbody>
</table>

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 = σA=a (R)

\[ S(W) = S(R) \]
\[ T(W) = ? \]

Example

\[ R \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat 1</td>
<td>10</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>cat 1</td>
<td>20</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>dog 1</td>
<td>30</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>dog 1</td>
<td>40</td>
<td>c</td>
<td></td>
</tr>
<tr>
<td>bat 1</td>
<td>50</td>
<td>d</td>
<td></td>
</tr>
</tbody>
</table>

\[ W = \sigma z=val(R) (R) \]
\[ T(W) = \frac{T(R)}{V(R,z)} \]
Assumption:
Values in select expression \( Z = \text{val} \) are uniformly distributed over possible \( V(R,Z) \) values.

Alternate Assumption:
Values in select expression \( Z = \text{val} \) are uniformly distributed over domain with \( \text{DOM}(R,Z) \) values.

Example
\[
\begin{array}{cccc}
\text{A} & \text{B} & \text{C} & \text{D} \\
\text{cat} & 1 & 10 & a \\
\text{cat} & 1 & 20 & b \\
\text{dog} & 1 & 30 & a \\
\text{dog} & 1 & 40 & c \\
\text{bat} & 1 & 50 & d \\
\end{array}
\]

Alternate assumption
\[
\begin{align*}
V(R,A) &= 3 & \text{DOM}(R,A) &= 10 \\
V(R,B) &= 1 & \text{DOM}(R,B) &= 10 \\
V(R,C) &= 5 & \text{DOM}(R,C) &= 10 \\
V(R,D) &= 4 & \text{DOM}(R,D) &= 10 \\
\end{align*}
\]

\[ W = \sigma_{z=\text{val}(R)}(R) \quad T(W) = ? \]
\[ C = \text{val} \Rightarrow T(W) = (\frac{1}{10})^1 + (\frac{1}{10})^1 + \ldots \]
\[ = (\frac{5}{10}) = 0.5 \]

\[ B = \text{val} \Rightarrow T(W) = (\frac{1}{10})^5 + 0 + 0 = 0.5 \]

\[ A = \text{val} \Rightarrow T(W) = (\frac{1}{10})^2 + (\frac{1}{10})^2 + (\frac{1}{10})^1 \]
\[ = 0.5 \]

**Example**

<table>
<thead>
<tr>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>1</td>
<td>10</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>cat</td>
<td>1</td>
<td>20</td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>dog</td>
<td>1</td>
<td>30</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>dog</td>
<td>1</td>
<td>40</td>
<td>c</td>
<td>c</td>
</tr>
<tr>
<td>bat</td>
<td>1</td>
<td>50</td>
<td>d</td>
<td>d</td>
</tr>
</tbody>
</table>

Alternate assumption

\[ V(R, A) = 3 \quad \text{DOM}(R, A) = 10 \]
\[ V(R, B) = 1 \quad \text{DOM}(R, B) = 10 \]
\[ V(R, C) = 5 \quad \text{DOM}(R, C) = 10 \]
\[ V(R, D) = 4 \quad \text{DOM}(R, D) = 10 \]

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

**Selection cardinality**

\[ SC(R, A) = \text{average # records that satisfy equality condition on } R.A \]
\[ = \frac{T(R)}{V(R, A)} \]

\[ SC(R, A) = \left\{ \begin{array}{l}
\frac{T(R)}{V(R, A)} \\
\frac{T(R)}{\text{DOM}(R, A)}
\end{array} \right. \]
What about $W = \sigma_{z \geq \text{val}}(R)$?

$T(W) = ?$

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

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

**Solution #3:** Estimate values in range

*Example* $R$

<table>
<thead>
<tr>
<th>$Z$</th>
<th>$V(R,Z)=10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Min}=1$</td>
<td>$W = \sigma_{z \geq 15}(R)$</td>
</tr>
<tr>
<td>$\text{Max}=20$</td>
<td></td>
</tr>
</tbody>
</table>

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

$T(W) = f \times T(R)$

*Equivalently:*

$f \times V(R,Z) = \text{fraction of distinct values}$

$T(W) = \left[f \times V(Z,R)\right] \times T(R) = f \times T(R) \times V(Z,R)$
Size estimate for \( W = R_1 \bowtie R_2 \)

Let \( x \) = attributes of \( R_1 \)
\( y \) = attributes of \( R_2 \)

**Case 1**
\[ X \cap Y = \emptyset \]
Same as \( R_1 \times R_2 \)

**Case 2**
\[ W = R_1 \bowtie R_2 \quad X \cap Y = A \]

Assumption:
\[ V(R_1,A) \leq V(R_2,A) \Rightarrow \text{Every A value in } R_1 \text{ is in } R_2 \]
\[ V(R_2,A) \leq V(R_1,A) \Rightarrow \text{Every A value in } R_2 \text{ is in } R_1 \]

"containment of value sets"

**Computing** \( T(W) \) **when** \( V(R_1,A) \leq V(R_2,A) \)

Take 1 tuple

1 tuple matches with \( T(R_2) \) tuples...
\[ V(R_2,A) \]

so
\[ T(W) = \frac{T(R_2)}{V(R_2,A)} \times T(R_1) \]
\[ V(R1,A) \leq V(R2,A) \quad T(W) = \frac{T(R2) T(R1)}{V(R2,A)} \]

\[ V(R2,A) \leq V(R1,A) \quad T(W) = \frac{T(R2) T(R1)}{V(R1,A)} \]

[A is common attribute]

In general
\[ W = R1 \times R2 \]
\[ T(W) = \frac{T(R2) T(R1)}{\max\{V(R1,A), V(R2,A)\}} \]

Case 2

with alternate assumption

Values uniformly distributed over domain

This tuple matches \( T(R2) / \text{DOM}(R2,A) \) so
\[ T(W) = \frac{T(R2) T(R1)}{\text{DOM}(R2,A)} = \frac{T(R2) T(R1)}{\text{DOM}(R1,A)} \]
Assume the same
In all cases:
\[ S(W) = S(R1) + S(R2) - S(A) \]

size of attribute A