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
<table>
<thead>
<tr>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>S</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td>10</td>
<td>x</td>
<td>2</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>20</td>
<td></td>
<td></td>
<td>20</td>
<td>y</td>
<td>2</td>
</tr>
<tr>
<td>c</td>
<td>2</td>
<td>10</td>
<td></td>
<td></td>
<td>30</td>
<td>z</td>
<td>2</td>
</tr>
<tr>
<td>d</td>
<td>2</td>
<td>35</td>
<td></td>
<td></td>
<td>40</td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>e</td>
<td>3</td>
<td>45</td>
<td></td>
<td></td>
<td>50</td>
<td>y</td>
<td>3</td>
</tr>
</tbody>
</table>

**Answer**

<table>
<thead>
<tr>
<th>B</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>x</td>
</tr>
</tbody>
</table>
• How do we execute query?

One idea

- Do Cartesian product
- Select tuples
- Do projection
<table>
<thead>
<tr>
<th>RXS</th>
<th>R.A</th>
<th>R.B</th>
<th>R.C</th>
<th>S.C</th>
<th>S.D</th>
<th>S.E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>x</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>1</td>
<td>10</td>
<td>20</td>
<td>y</td>
<td>2</td>
</tr>
</tbody>
</table>

Bingo! Got one...

- C
- \[\ell\]
- \[\ell\]
- \[\ell\]
Relational Algebra - can be used to describe plans...

Ex: Plan I

\[ \Pi_{B,D} \left( \sigma_{\text{R.A}=\text{"c"} \land S.E=2 \land R.C=S.C} (RXS) \right) \]

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

Plan II

\[ \Pi_{B,D} \]

\[ \sigma_{R.A = "c"} \]

\[ \sigma_{S.E = 2} \]

\[ R \]

\[ S \]

natural join
Plan III

Use R.A and S.C Indexes

(1) Use R.A index to select R tuples with R.A = “c”

(2) For each R.C value found, use S.C index to find matching tuples

(3) Eliminate S tuples S.E ≠ 2

(4) Join matching R,S tuples, project B,D attributes and place in result
### R
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>c</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>d</td>
<td>2</td>
<td>35</td>
</tr>
<tr>
<td>e</td>
<td>3</td>
<td>45</td>
</tr>
</tbody>
</table>

### S
<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>x</td>
<td>2</td>
</tr>
<tr>
<td>20</td>
<td>y</td>
<td>2</td>
</tr>
<tr>
<td>30</td>
<td>z</td>
<td>2</td>
</tr>
<tr>
<td>40</td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>50</td>
<td>y</td>
<td>3</td>
</tr>
</tbody>
</table>

**A = “c”**

- **I₁**
  - Input: <c, 2, 10>
  - Output: <2, x>
  - Next tuple: <c, 7, 15>

- **I₂**
  - Check = 2?
  - Input: <10, x, 2>
  - Output: <2, x>

**C**

- <10, x, 2>
Overview of Query Optimization
SQL query

parse

parse tree

convert

logical query plan

apply laws

“improved” l.q.p

estimate result sizes

l.q.p. + sizes

consider physical plans

execute

answer

pick best

estimate costs

{(P1,C1),(P2,C2)…}

{P1,P2,…..}
Example: SQL query

```sql
SELECT title
FROM StarsIn
WHERE starName IN (
    SELECT name
    FROM MovieStar
    WHERE birthdate LIKE '%%1960'
);

(Find the movies with stars born in 1960)
Example: Parse Tree

```
<Query>
  <SFW>
    SELECT <SelList> FROM <FromList> WHERE <Condition>
    <Attribute> title <RelName> StarsIn <Tuple> IN ( <Query> )
    <Attribute> starName <SFW>
    SELECT <SelList> FROM <FromList> WHERE <Condition>
    <Attribute> name <RelName> MovieStar <Attribute> birthDate LIKE <Pattern> '%1960'
```
Example: Generating Relational Algebra

\[ \Pi_{\text{title}} \]

\[ \sigma \]

\[ \text{StarsIn} \]

\[ \langle \text{condition} \rangle \]

\[ \langle \text{tuple} \rangle \]

\[ \text{IN} \]

\[ \Pi_{\text{name}} \]

\[ \langle \text{attribute} \rangle \]

\[ \sigma \text{birthdate LIKE '%1960'} \]

\[ \text{starName} \]

\[ \text{MovieStar} \]

Fig. 16.14: An expression using a two-argument \( \sigma \), midway between a parse tree and relational algebra
Example: Logical Query Plan

\[
\Pi_{\text{title}} \left( \sigma_{\text{starName}=\text{name}} (\prod_{\text{name}} (\sigma_{\text{birthdate LIKE } '%1960'} (\text{StarsIn}) \times \Pi_{\text{name}} (\text{MovieStar}))))\right)
\]

Fig. 16.16: Applying the rule for IN conditions
Example: Improved Logical Query Plan

$$\Pi_{\text{title}}$$

$$\sigma_{\text{birthdate} \text{ LIKE '1960'}}$$

$$\Pi_{\text{name}}$$

$$\sigma_{\text{starName} = \text{name}}$$

$$\Pi_{\text{name}}$$

$$\Pi_{\text{title}}$$

Question: Push project to StarsIn?

Fig. 16.21: The effect of query rewriting.
Example: Estimate Result Sizes

StarsIn

Need expected size

MovieStar

\( \Pi \sigma \)
Example: One Physical Plan

Hash join

SEQ scan
StarsIn

index scan
MovieStar

Parameters: join order, memory size, project attributes, ...

Parameters: Select Condition, ...
Example: Estimate costs

L.Q.P

P1      P2      ....      Pn

C1      C2      ....      Cn

Pick best!
Textbook outline

Chapter 15
15.1 Physical operators
   - Scan, sort, ...
15.2-15.9 Implementing operators + estimating their cost
Chapter 16

16.1 Parsing
16.2 Algebraic laws
16.3 Parse tree -> logical query plan
16.4 Estimating result sizes
16.5-16.7 Cost based optimization
Reading textbook - Chapters 15, 16

Optional: 15.7-15.9, 16.6-16.7
Optional: Duplicate elimination operator grouping, aggregation operators
Query Optimization - In class order

• Relational algebra level
• Detailed query plan level
  – Estimate Costs
    • without indexes
    • with indexes
  – Generate and compare plans
Relational algebra optimization

• Transformation rules (preserve equivalence)
• What are good transformations?
Rules: Natural joins & cross products & union

R ⋈ S = S ⋈ R
(R ⋈ S) ⋈ T = R ⋈ (S ⋈ 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

$$\sigma_{p_1 \wedge p_2}(R) = \sigma_{p_1} \left[ \sigma_{p_2}(R) \right]$$

$$\sigma_{p_1 \lor p_2}(R) = \left[ \sigma_{p_1}(R) \right] \cup \left[ \sigma_{p_2}(R) \right]$$
Bags vs. Sets

R = \{a,a,b,b,b,c\}
S = \{b,b,c,c,d\}
RUS = ?

• Option 1    SUM
  RUS = \{a,a,b,b,b,b,b,c,c,c,d\}

• Option 2    MAX
  RUS = \{a,a,b,b,b,c,c,d\}
Option 2 (MAX) makes this rule work:

$$\sigma_{p_1 \vee p_2}(R) = \sigma_{p_1}(R) \cup \sigma_{p_2}(R)$$

Example: $R = \{a, a, b, b, b, c\}$

- $P_1$ satisfied by $a, b$; $P_2$ satisfied by $b, c$

$$\sigma_{p_1 \vee p_2}(R) = \{a, a, b, b, b, c\}$$

$$\sigma_{p_1}(R) = \{a, a, b, b, b\}$$

$$\sigma_{p_2}(R) = \{b, b, b, c\}$$

$$\sigma_{p_1}(R) \cup \sigma_{p_2}(R) = \{a, a, b, b, b, c\}$$
"Sum" option makes more sense:

Senators (…….)

T1 = $\pi_{\text{yr, state}}$ Senators;

<table>
<thead>
<tr>
<th>Yr</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>CA</td>
</tr>
<tr>
<td>99</td>
<td>CA</td>
</tr>
<tr>
<td>98</td>
<td>AZ</td>
</tr>
</tbody>
</table>

Rep (…….)

T2 = $\pi_{\text{yr, state}}$ Reps

<table>
<thead>
<tr>
<th>Yr</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>CA</td>
</tr>
<tr>
<td>99</td>
<td>CA</td>
</tr>
<tr>
<td>98</td>
<td>CA</td>
</tr>
</tbody>
</table>

Union?
Executive Decision

-> Use “SUM” option for bag unions
-> Some rules cannot be used for bags
Rules: Project

Let: $X =$ set of attributes

$Y =$ set of attributes

$XY = X \cup Y$

$\pi_{xy}(R) = \pi_x[\pi_y(R)]$
Rules: $\sigma + \bigotimes$ combined

Let $p = \text{predicate with only R attribs}$
$q = \text{predicate with only S attribs}$
$m = \text{predicate with only R,S attribs}$

$$\sigma_p (R \bowtie S) = [\sigma_p (R)] \bowtie S$$
$$\sigma_q (R \bowtie S) = R \bowtie [\sigma_q (S)]$$
Some Rules can be Derived:

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

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

\( \sigma_{p \lor q} (R \bowtie S) = \)
Do one:

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

\[ \sigma_{p \land q \land m} (R \bowtie S) = \]

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

\[ \sigma_{pvq} (R \bowtie S) = \]

\[ \left[ (\sigma_p R) \bowtie S \right] \cup \left[ R \bowtie (\sigma_q S) \right] \]
--> Derivation for first one:

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

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

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

$$[\sigma_p (R)] \bowtie [\sigma_q (S)]$$
Rules: $\pi, \sigma$ combined

Let $x = \text{subset of } R \text{ attributes}$

$z = \text{attributes in predicate } P$
(\text{subset of } R \text{ attributes})

$$\pi_x[\sigma_p (R)] = \pi_x \{ \sigma_p [\pi_x (R)] \}$$
Rules: \( \pi, \bowtie \) combined

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

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

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

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

similar...

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

- $\sigma_{p_1 \land p_2} (R) \rightarrow \sigma_{p_1} [\sigma_{p_2} (R)]$

- $\sigma_p (R \bowtie S) \rightarrow [\sigma_p (R)] \bowtie S$

- $R \bowtie S \rightarrow S \bowtie R$

- $\pi_x [\sigma_p (R)] \rightarrow \pi_x \{\sigma_p [\pi_{xz} (R)]\}$
Conventional wisdom: do projections early

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

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

$\pi_x \{\sigma_p (R)\} \quad \text{vs.} \quad \pi_E \left\{\sigma_p \{\pi_{ABE}(R)\}\right\}$
But What if we have A, B indexes?

B = “cat”

Intersect pointers to get pointers to matching tuples
Bottom line:

• No transformation is always good
• Usually good: early selections
In textbook: more transformations

- Eliminate common sub-expressions
- Other operations: duplicate elimination
Outline - Query Processing

- Relational algebra level
  - transformations
  - good transformations
- Detailed query plan level
  - estimate costs
  - generate and compare plans
• 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></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></td>
</tr>
<tr>
<td>cat</td>
<td>1</td>
<td>20</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>dog</td>
<td>1</td>
<td>30</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>dog</td>
<td>1</td>
<td>40</td>
<td>c</td>
<td></td>
</tr>
<tr>
<td>bat</td>
<td>1</td>
<td>50</td>
<td>d</td>
<td></td>
</tr>
</tbody>
</table>

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

\[ T(R) = 5 \quad S(R) = 37 \]

\[ V(R,A) = 3 \]

\[ V(R,B) = 1 \]

\[ V(R,C) = 5 \]

\[ V(R,D) = 4 \]
Size estimates for $W = R1 \times R2$

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

$S(W) = S(R1) + S(R2)$
Size estimate for \( W = \sigma_{A=a}(R) \)

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

\[ T(W) = ? \]
Example

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

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

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

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

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

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

Alternate assumption:

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

$$W = \sigma_{z = \text{val}(R)} \ T(W) = ?$$
C=val \implies T(W) = (1/10)1 + (1/10)1 + \ldots \\
= (5/10) = 0.5

B=val \implies T(W) = (1/10)5 + 0 + 0 = 0.5

A=val \implies T(W) = (1/10)2 + (1/10)2 + (1/10)1 \\
= 0.5
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>

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**

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

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

$T(W) = ?$

- Solution #1:
  
  $T(W) = \frac{T(R)}{2}$

- Solution #2:
  
  $T(W) = \frac{T(R)}{3}$
• Solution # 3: Estimate values in range

Example

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

\[ f = \frac{20-15+1}{20-1+1} = \frac{6}{20} \quad \text{(fraction of range)} \]

\[ T(W) = f \times T(R) \]
Equivalently:
\[ f \times V(R, Z) = \text{fraction of distinct values} \]
\[ T(W) = \left[ f \times V(Z, R) \right] \times T(R) = f \times \frac{T(R)}{V(Z, R)} \]
Size estimate for $W = R_1 \boxtimes R_2$

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

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

Assumption:
\[
V(R_1,A) \leq V(R_2,A) \quad \Rightarrow \quad \text{Every A value in R1 is in R2}
\]
\[
V(R_2,A) \leq V(R_1,A) \quad \Rightarrow \quad \text{Every A value in R2 is in R1}
\]

“containment of value sets”  Sec. 7.4.4
Computing $T(W)$ when $V(R1,A) \leq V(R2,A)$

Take 1 tuple

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

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

• \( V(R_2,A) \leq V(R_1,A) \quad T(W) = \frac{T(R_2) \cdot T(R_1)}{V(R_1,A)} \)

[A is common attribute]
In general \[ W = R_1 \bowtie R_2 \]

\[
T(W) = \frac{T(R_2) \cdot T(R_1)}{\max\{ V(R_1,A), V(R_2,A) \}}
\]
Case 2 with alternate assumption

Values uniformly distributed over domain

\[ \begin{array}{ccc}
R1 & A & B & C \\
R2 & A & D \\
\end{array} \]

This tuple matches \( T(R2)/\text{DOM}(R2,A) \) so

\[ T(W) = \frac{T(R2) \cdot T(R1)}{\text{DOM}(R2, A)} = \frac{T(R2) \cdot T(R1)}{\text{DOM}(R1, A)} \]

Assume the same
In all cases:

\[ S(W) = S(R1) + S(R2) - S(A) \]
Using similar ideas, we can estimate sizes of:

\[ \Pi_{AB}(R) \] ….. Sec. 16.4.2

\[ \sigma_{A=a \land B=b}(R) \] .... Sec. 16.4.3

\[ R \bowtie S \] with common attribs. A,B,C
  Sec. 16.4.5

Union, intersection, diff, .... Sec. 16.4.7
Note: for complex expressions, need intermediate T, S, V results.

E.g. \( W = \sigma_{A=a} (R1) \bowtie R2 \)

Treat as relation \( U \)

\[ T(U) = T(R1)/V(R1,A) \quad S(U) = S(R1) \]

Also need \( V(U, \ast) \)!!
To estimate $V_s$

E.g., $U = \sigma_{A=a}(R_1)$

Say $R_1$ has attrs $A,B,C,D$

$V(U, A) =$

$V(U, B) =$

$V(U, C) =$

$V(U, D) =$
Example

R1

<table>
<thead>
<tr>
<th></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>10</td>
<td></td>
</tr>
<tr>
<td>cat</td>
<td>1</td>
<td>20</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>dog</td>
<td>1</td>
<td>30</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>dog</td>
<td>1</td>
<td>40</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>bat</td>
<td>1</td>
<td>50</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

V(R1,A) = 3
V(R1,B) = 1
V(R1,C) = 5
V(R1,D) = 3
U = \sigma_{A=a}(R1)

V(U,A) = 1  V(U,B) = 1  V(U,C) = \frac{T(R1)}{V(R1,A)}

V(D,U) ... somewhere in between
Possible Guess \[ U = \sigma_{A=a} (R) \]

\[ V(U,A) = 1 \]
\[ V(U,B) = V(R,B) \]
For Joins \[ U = R_1(A,B) \bowtie R_2(A,C) \]

\[
\begin{align*}
V(U,A) & = \min \{ V(R_1, A), V(R_2, A) \} \\
V(U,B) & = V(R_1, B) \\
V(U,C) & = V(R_2, C)
\end{align*}
\]

[called “preservation of value sets” in section 16.4.4]
Example:

\[ Z = R_1(A,B) \Join R_2(B,C) \Join R_3(C,D) \]

<table>
<thead>
<tr>
<th>R1</th>
<th>T(R1) = 1000 V(R1,A)=50 V(R1,B)=100</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2</td>
<td>T(R2) = 2000 V(R2,B)=200 V(R2,C)=300</td>
</tr>
<tr>
<td>R3</td>
<td>T(R3) = 3000 V(R3,C)=90 V(R3,D)=500</td>
</tr>
</tbody>
</table>
Partial Result: \( U = R \bowtie S \)

\[
T(U) = \frac{1000 \times 2000}{200} \quad V(U,A) = 50
\]

\[
V(U,B) = 100 \\
V(U,C) = 300
\]
\[ Z = U \otimes R3 \]

\[ T(Z) = \frac{1000 \times 2000 \times 3000}{200 \times 300} \]

\[ V(Z,A) = 50 \]
\[ V(Z,B) = 100 \]
\[ V(Z,C) = 90 \]
\[ V(Z,D) = 500 \]
Summary

• Estimating size of results is an “art”

• Don’t forget:
  Statistics must be kept up to date...
  (cost?)
Outline

• Estimating cost of query plan
  – Estimating size of results → done!
  – Estimating # of IOs ← next...

• Generate and compare plans