Week 12

Query Processing

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>10</td>
<td>x</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>20</td>
<td></td>
<td>20</td>
<td>y</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>2</td>
<td>10</td>
<td></td>
<td>30</td>
<td>z</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>2</td>
<td>35</td>
<td></td>
<td>40</td>
<td>x</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>3</td>
<td>45</td>
<td></td>
<td>50</td>
<td>y</td>
<td>3</td>
<td></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?

- Do Cartesian product
- Select tuples
- Do projection

One idea

\[
\begin{array}{cccccc}
\hline
a & 1 & 10 & 10 & x & 2 \\
a & 1 & 10 & 20 & y & 2 \\
. & . & . & . & . & . \\
C & 2 & 10 & 10 & x & 2 \\
. & . & . & . & . & . \\
\end{array}
\]
Relational Algebra - can be used to describe plans...

Ex: Plan I

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

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

Another idea:

Plan II

\[ \Pi_{B,D} \]
\[ \sigma_{R.A=“c”} \]
\[ \sigma_{S.E=2} \]
\[ \bowtie \]
\[ \bowtie \text{ natural join} \]

\[ R \]
\[ S \]
**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
Overview of Query Optimization
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> <RelName> <Tuple> IN <Query>
    title StarsIn <Attribute> ( <Query> )
  starName <SFW>
    SELECT <SelList> FROM <FromList> WHERE <Condition>
    <Attribute> <RelName> <Attribute> LIKE <Pattern>
    name MovieStar birthDate '%'1960'
```

**Example: Generating Relational Algebra**

```
\Pi title
  \sigma
    StarsIn <condition>
      <tuple> IN \Pi name
        <attribute> \sigma birthdate LIKE '%1960'
          startName MovieStar
```

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

\[
\begin{align*}
\Pi_{\text{title}} \\
\sigma_{\text{starName}=\text{name}} \\
\times \\
\Pi_{\text{name}} \\
\sigma_{\text{birthdate LIKE } '1960'} \\
\text{MovieStar}
\end{align*}
\]

Applying the rule for IN conditions

**Example: Improved Logical Query Plan**

\[
\begin{align*}
\Pi_{\text{title}} \\
\text{starName}=\text{name} \\
\times \\
\Pi_{\text{name}} \\
\sigma_{\text{birthdate LIKE } '1960'} \\
\text{MovieStar}
\end{align*}
\]

Question: Push projection to StarsIn?
**Example:** Estimate Result Sizes

Need expected size

\[ \Pi \sigma \]

\[ \text{MovieStar} \]

**Example:** One Physical Plan

![Diagram of a physical plan](image)

Hash join

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

SEQ scan

index scan

Parameters: Select Condition,...


**Example: Estimate costs**

```
L.Q.P
  |   |
P1  P2  ...  Pn
  |   |
C1  C2  ...  Cn
```

Pick best!

**Outline**

**Algebra for queries**
- Select, project, join, ....
- Duplicate elimination, grouping, sorting

[bags vs sets]

[project list
a,a+b->x,...]

**Physical operators**
- Scan, sort, ...

**Implementing operators**
- estimating their cost
Parsing
Algebraic laws
Parse tree -> logical query plan
Estimating result sizes
Cost based optimization

**Query Optimization**

- 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 \bowtie S = S \bowtie R
(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)
**Note:**

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

```
  T   R
 / \ / \  =  / \ / \\
R   S  S  T
```

**Rules:** Natural joins & cross products & union

- $\text{R } \bowtie \text{ S } = \text{ S } \bowtie \text{ R}$
- $(\text{R } \bowtie \text{ S}) \bowtie \text{ T } = \text{ R } \bowtie (\text{ S } \bowtie \text{ T })$

- $\text{R } \times \text{ S } = \text{ S } \times \text{ R}$
- $(\text{R } \times \text{ S}) \times \text{ T } = \text{ R } \times (\text{ S } \times \text{ T })$

- $\text{R } \cup \text{ S } = \text{ S } \cup \text{ R}$
- $\text{R } \cup (\text{ S } \cup \text{ T }) = (\text{ R } \cup \text{ S }) \cup \text{ T}$
**Rules: Selects**

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

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

**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:

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

**Example:** \( R = \{a, a, b, b, b, c\} \)

- \( P_1 \) satisfied by \( a, b \); \( P_2 \) satisfied by \( b, c \)
- \( \sigma_{p_1 \lor p_2}(R) = \{a, a, b, b, b, c\} \)
- \( \sigma_{p_1}(R) = \{a, a, b, b, b\} \)
- \( \sigma_{p_2}(R) = \{b, b, b, c\} \)
- \( \sigma_{p_1}(R) \cup \sigma_{p_2}(R) = \{a, a, b, b, b, c\} \)

"Sum" option makes more sense:

Senators (......)    Rep (......)

\( T_1 = \pi_{yr, state} Senators; \quad T_2 = \pi_{yr, state} Reps \)

<table>
<thead>
<tr>
<th></th>
<th>Yr</th>
<th>State</th>
<th></th>
<th>Yr</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>97</td>
<td>CA</td>
<td></td>
<td>99</td>
<td>CA</td>
</tr>
<tr>
<td></td>
<td>99</td>
<td>CA</td>
<td></td>
<td>99</td>
<td>CA</td>
</tr>
<tr>
<td></td>
<td>98</td>
<td>AZ</td>
<td></td>
<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  
$XY = X U Y$

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

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

\[
\begin{align*}
\sigma_p (R \bowtie S) &= [\sigma_p (R)] \bowtie S \\
\sigma_q (R \bowtie S) &= R \bowtie [\sigma_q (S)]
\end{align*}
\]

**Rules:** \( \sigma + \bowtie \) combined (continued)

Some Rules can be Derived:

\[
\begin{align*}
\sigma_{p \land q} (R \bowtie S) &= \\
\sigma_{p \land q \land m} (R \bowtie S) &= \\
\sigma_{p \lor q} (R \bowtie S) &=
\end{align*}
\]
Do one, others for homework:

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

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

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

\[ [ (\sigma_p R) \bowtie S] \cup [R \bowtie (\sigma_q S)] \]

\[ \rightarrow \text{ Derivation for first one:} \]

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

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

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

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

Let \( x = \) 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

\[
\pi_{xy}(R \bowtie S) = \pi_{xy}\{[\pi_{xz}(R) \bowtie \pi_{yz}(S)]\}
\]
\[ \pi_{xy} \{ \sigma_p (R \bowtie S) \} = \]
\[ \pi_{xy} \{ \sigma_p [\pi_{xz'} (R) \bowtie \pi_{yz'} (S)] \} \]
\[ z' = z \cup \{ \text{attributes used in P} \} \]

**Rules for \( \sigma, \pi \) combined with \( X \)**

similar...

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

Example: \( R(A,B,C,D,E) \quad x = \{E\} \)
\[ P: (A=3) \land (B=\text{“cat”}) \]

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

**But** What if we have A, B indexes?

\( B = \text{“cat”} \) \[ \begin{array}{c}
\text{ pointers to matching tuples } \\
\end{array} \]
\( A = 3 \) \[ \begin{array}{c}
\text{ pointers to matching tuples } \\
\end{array} \]

Intersect pointers to get
**Bottom line:**

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

**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>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

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

\[ V(R,A) = 3 \quad V(R,C) = 5 \]

\[ V(R,B) = 1 \quad V(R,D) = 4 \]

Size estimates for \( W = R_1 \times R_2 \)

\[ T(W) = T(R_1) \times T(R_2) \]

\[ S(W) = S(R_1) + S(R_2) \]
**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)}$
**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

<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

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

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

$C=\text{val} \Rightarrow T(W) = (1/10)1 + (1/10)1 + ...$
$= (5/10) = 0.5$

$B=\text{val} \Rightarrow T(W) = (1/10)5 + 0 + 0 = 0.5$

$A=\text{val} \Rightarrow T(W) = (1/10)2 + (1/10)2 + (1/10)1$
$= 0.5$
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>

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

\[ W = \sigma_{Z=\text{val}(R)} \]

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

Selection cardinality

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

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

$T(W) = ?$

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

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

- 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>

$f = \frac{20-15+1}{20-1+1} = \frac{6}{20} = 0.3$ (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 \( W = R_1 \bowtie R_2 \)

Let \( x = \text{attributes of } R_1 \)
\( y = \text{attributes of } R_2 \)

**Case 1**  \[ X \cap Y = \emptyset \]

Same as \( R_1 \times R_2 \)
Case 2

$W = R_1 \Join R_2 \quad X \cap Y = A$

<table>
<thead>
<tr>
<th>R1</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2</td>
<td>A</td>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>

**Assumption:**

$V(R_1, A) \leq V(R_2, A) \implies$ Every A value in R1 is in R2

$V(R_2, A) \leq V(R_1, A) \implies$ Every A value in R2 is in R1

“containment of value sets”

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

Take 1 tuple

1 tuple matches with $\frac{T(R_2)}{V(R_2, A)}$ tuples...

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

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

[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

<table>
<thead>
<tr>
<th>R1</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R2</th>
<th>A</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

Case 2

Assume the same

In all cases:

\[
S(W) = S(R1) + S(R2) - S(A)
\]

size of attribute A
Using similar ideas, we can estimate sizes of:

\[ \Pi_{AB}(R) \] ....

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

R \bowtie S with common attrs. A,B,C

Union, intersection, diff, ....

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, *) !!
To estimate $V_s$

E.g., $U = \sigma_{A=a} (R1)$

Say $R1$ has attribs $A,B,C,D$

$V(U, A) =$
$V(U, B) =$
$V(U, C) =$
$V(U, D) =$

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>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>

$V(R1,A) = 3$
$V(R1,B) = 1$
$V(R1,C) = 5$
$V(R1,D) = 3$

$U = \sigma_{A=a} (R1)$

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

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

\[
\begin{align*}
V(U,A) &= 1 \\
V(U,B) &= V(R,B)
\end{align*}
\]

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

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

[“preservation of value sets”]
**Example:**

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

<table>
<thead>
<tr>
<th>Relation</th>
<th>T</th>
<th>V(R1,A)</th>
<th>V(R1,B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>1000</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>R2</td>
<td>2000</td>
<td>200</td>
<td>300</td>
</tr>
<tr>
<td>R3</td>
<td>3000</td>
<td>90</td>
<td>500</td>
</tr>
</tbody>
</table>

**Partial Result:** \[ U = R \Join S \]

\[
T(U) = \frac{1000 \times 2000}{200} = 50 \\
V(U,A) = 50 \\
V(U,B) = 100 \\
V(U,C) = 300
\]
\[
Z = U \times R^3
\]

\[
T(Z) = \frac{1000 \times 2000 \times 3000}{200 \times 300} \quad V(Z,A) = 50
\]
\[
V(Z,B) = 100 \quad V(Z,C) = 90 \quad V(Z,D) = 500
\]

**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      → occurs next...

• Generate and compare plans  ...Final step