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

Answer B     D
2      x

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

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

Answer B     D
2      x
• How do we execute query?

  - Do Cartesian product
  - Select tuples
  - Do projection

One idea

<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>a</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>x</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>10</td>
<td>20</td>
<td>y</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>x</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Bingo!
Got one...

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

Ex: Plan I

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

Plan II

Another idea:

OR: \[ \Pi_{B,D} \left( \sigma_{R.A = "c" \land S.E = 2 \land R.C = S.C} (R \times S) \right) \]
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
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
Example: Generating Relational Algebra

\[
\Pi_{\text{title}}
\sigma_{\text{birthdate} \text{ LIKE} \%1960'}
\text{StarsIn}
\text{IN}
\Pi_{\text{name}}
\text{starName}
\text{MovieStar}
\]

Fig. 7.15: An expression using a two-argument \( \sigma \), midway between a parse tree and relational algebra

Example: Logical Query Plan

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

Fig. 7.18: Applying the rule for \( \text{IN} \) conditions

Example: Improved Logical Query Plan

\[
\Pi_{\text{title}}
\sigma_{\text{starName} = \text{name}}
\text{StarsIn}
\Pi_{\text{name}}
\sigma_{\text{birthdate} \text{ LIKE} \%1960'}
\text{MovieStar}
\]

Fig. 7.20: An improvement on fig. 7.18.

Question: Push project to \text{StarsIn}?

Example: Estimate Result Sizes

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

Need expected size

Example: One Physical Plan

\[
\text{Hash join} 
\rightarrow \text{Parameters: join order, memory size, project attributes...}
\]

\[
\text{SEQ scan}
\rightarrow \text{Parameters: Select Condition...}
\]

\[
\text{index scan}
\]

\[
\text{StarsIn}
\rightarrow \text{MovieStar}
\]

Example: Estimate costs

\[
\text{L.Q.P}
\rightarrow \text{P1}
\rightarrow \text{P2}
\rightarrow \ldots
\rightarrow \text{Pn}
\rightarrow \text{C1}
\rightarrow \text{C2}
\rightarrow \ldots
\rightarrow \text{Cn}
\]

Pick best!
Textbook outline

Chapter 15
5 Algebra for queries  [bags vs sets]
- Select, project, join, ...
  [project list  a,a+b->x,...]
- Duplicate elimination, grouping, sorting
15.1 Physical operators
- Scan, sort, ...
15.2 - 15.6 Implementing operators +
    estimating their cost

Reading textbook - Chapters 15, 16

Optional:
- Sections 15.7, 15.8, 15.9 [15.7, 15.8]
- Sections 16.6, 16.7 [16.6, 16.7]
Optional: Duplicate elimination operator
grouping, aggregation operators

Query Optimization - In class order

- Relational algebra level (A)
- Detailed query plan level
  - Estimate Costs (B)
    • without indexes
    • with indexes
  - Generate and compare plans (C)

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.:

\[ \bowtie \]
\[ R \]
\[ S \]
\[ \bowtie \]
\[ S \]
\[ T \]

Rules: Selects

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

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

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

Note:

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

\[ \bowtie \]
\[ R \]
\[ S \]
\[ \bowtie \]
\[ S \]
\[ T \]
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,c,c,d\}
• Option 2  MAX
  RUS = \{a,a,b,b,b,c\}

Option 2 (MAX) makes this rule work:

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

Example: R = \{a,a,b,b,b,c\}
P1 satisfied by a,b;  P2 satisfied by b,c

“Sum” option makes more sense:

Senators (……)  Rep (……)
T1 = \pi_{yr, state} Senators;  T2 = \pi_{yr, state} Reps

<table>
<thead>
<tr>
<th>Yr</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>CA</td>
</tr>
<tr>
<td>16</td>
<td>CA</td>
</tr>
<tr>
<td>15</td>
<td>AZ</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

\[ \pi_{xy} (R) = \]
**Rules: Project**

Let: \( X = \text{set of attributes} \)
\( Y = \text{set of attributes} \)
\( XY = X \cup Y \)

\[ \pi_{xy} (R) = \pi_x [\pi_y (R)] \]

---

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

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

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

---

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

Some Rules can be Derived:

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

---

**Do one, others for homework:**

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

\[ [ (\sigma_p R) \bowtie S] \cup [R \bowtie (\sigma_q S)] \]
-> Derivation for first one:
\[ \sigma_{p \land q} (R \bowtie S) = \]

---

**Rules: \( \pi, \sigma \) combined**

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

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

---

Rules: \( \pi, \sigma \) combined

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

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

---

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

\[ \pi_{xy} (R \bowtie S) = \]
**Rules:** \( \pi, \bowtie \text{ combined} \)

Let \( x \) = subset of \( R \) attributes

\( y \) = subset of \( S \) attributes

\( z \) = intersection of \( R, S \) attributes

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

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

---

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

---

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

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

\[ z' = z \cup \{ \text{attributes used in } P \} \]

---

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

Similar...

E.g., \( \sigma_p (R X S) = ? \)

---

**Rules:** \( \sigma, U \) combined:

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

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

---

**Which are “good” transformations?**

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

Example: \( R(A,B,C,D,E) \) \( x=\{E\} \)
\[ P: (A=3) \land (B=\text{“cat”}) \]
\[ \pi_x \{ \sigma_p (R) \} \text{ vs. } \pi_{\sigma_p \{ \pi_{ABE}(R) \}} \]

But: What if we have A, B indexes?
\[ B = \text{“cat“} \quad \square \quad A=3 \quad \square \]
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>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></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

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

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

\[
S(W) = S(R_1) + S(R_2)
\]
Example

\[ W = \sigma_{z=val(R)} T(W) = \]

\[ R \quad A \quad B \quad C \quad D \]
\[ \text{cat} \quad 1 \quad 10 \quad a \]
\[ \text{cat} \quad 1 \quad 20 \quad b \]
\[ \text{dog} \quad 1 \quad 30 \quad a \]
\[ \text{dog} \quad 1 \quad 40 \quad c \]
\[ \text{bat} \quad 1 \quad 50 \quad d \]

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

Assumption:

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

Alternate Assumption:

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

Example

\[ W = \sigma_{z=val(R)} T(W) = \]

\[ R \quad A \quad B \quad C \quad D \]
\[ \text{cat} \quad 1 \quad 10 \quad a \]
\[ \text{cat} \quad 1 \quad 20 \quad b \]
\[ \text{dog} \quad 1 \quad 30 \quad a \]
\[ \text{dog} \quad 1 \quad 40 \quad c \]
\[ \text{bat} \quad 1 \quad 50 \quad d \]

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

What is probability this tuple will be in answer?

\[ W = \sigma_{z=val(R)} T(W) = \]

\[ A \quad B \quad C \quad D \]
\[ \text{cat} \quad 1 \quad 10 \quad a \]
\[ \text{cat} \quad 1 \quad 20 \quad b \]
\[ \text{dog} \quad 1 \quad 30 \quad a \]
\[ \text{dog} \quad 1 \quad 40 \quad c \]
\[ \text{bat} \quad 1 \quad 50 \quad d \]

Alternate assumption

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

\[ W = \sigma_{z=val(R)} T(W) = ? \]
Example

\[
W = \sigma_{z=\text{val}(R)}(R) \quad T(W) = ?
\]

Selection cardinality

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

\[
SC(R,A) = \frac{T(R)}{V(R,A)} \quad \text{if } V(R,A) > 0
\]

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

\[
T(W) = ?
\]

- 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>W=\sigma_z \geq 15 (R)</td>
<td>Max=20</td>
</tr>
</tbody>
</table>

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

\[ T(W) = f \times T(R) \]

Equivalently:

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

\[ T(W) = [f \times V(Z,R)] \times T(R) = f \times T(R) \]

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

<table>
<thead>
<tr>
<th>W</th>
<th>R_1 \bowtie R_2</th>
<th>X \cap Y = A</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_1</td>
<td>A</td>
<td>B</td>
</tr>
</tbody>
</table>
Case 2  \[ W = R_1 \bowtie R_2 \quad X \cap Y = A \]

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

“containment of value sets” Sec. 7.4.4

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

Take 1 tuple

1 tuple matches with \( T(R_2) \) tuples...

\[ T(W) = \frac{T(R_2) \times T(R_1)}{V(R_2, A)} \]

In general
\[ W = R_1 \bowtie R_2 \]

\[ T(W) = \frac{T(R_2) \times T(R_1)}{\max\{ V(R_1, A), V(R_2, A) \}} \]

Case 2  

With alternate assumption

Values uniformly distributed over domain

\[ \frac{T(R_2) \times T(R_1)}{\text{DOM}(R_2, A)} \text{ so} \]

\[ T(W) = \frac{T(R_2) \times T(R_1)}{\text{DOM}(R_2, A) \times \text{DOM}(R_1, A)} \]

Assume the same
In all cases:

\[ S(W) = S(R_1) + S(R_2) - S(A) \]

size of attribute \( A \)

Using similar ideas, we can estimate sizes of:

\[ \Pi_{A=a} (R) \]  Sec. 16.4.2 (same for either edition)

\[ \sigma_{A=a} (B=b) (R) \]  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 \( V(U, *) \) !!

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

\[
\begin{array}{c|c|c|c|c}
   & A & B & C & D \\
\hline
\text{cat} & 1 & 10 & 10 & \\
\text{cat} & 1 & 20 & 20 & \\
\text{dog} & 1 & 30 & 10 & \\
\text{dog} & 1 & 40 & 30 & \\
\text{bat} & 1 & 50 & 10 & \\
\end{array}
\]

\[ V(R1,A) = 3 \]

\[ V(R1,B) = 1 \]

\[ V(R1,C) = 5 \]

\[ V(R1,D) = 3 \]

\[ U = \sigma_{A=a} (R1) \]

Example

\[
\begin{array}{c|c|c|c|c}
   & A & B & C & D \\
\hline
\text{cat} & 1 & 10 & 10 & \\
\text{cat} & 1 & 20 & 20 & \\
\text{dog} & 1 & 30 & 10 & \\
\text{dog} & 1 & 40 & 30 & \\
\text{bat} & 1 & 50 & 10 & \\
\end{array}
\]

\[ V(R1,A) = 3 \]

\[ V(R1,B) = 1 \]

\[ V(R1,C) = 5 \]

\[ V(R1,D) = 3 \]

\[ U = \sigma_{A=a} (R1) \]

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

\[ V(D,U) \] ... 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**  \( U = R1(A,B) \bowtie R2(A,C) \)

\[
\begin{align*}
V(U,A) &= \min \{ V(R1, A), V(R2, A) \} \\
V(U,B) &= V(R1, B) \\
V(U,C) &= V(R2, C)
\end{align*}
\]

[called “preservation of value sets” in section 7.4.4]

**Example:**
\( Z = R1(A,B) \bowtie R2(B,C) \bowtie R3(C,D) \)

\[
\begin{align*}
&\text{R1} \quad T(R1) = 1000 \quad V(R1,A)=50 \quad V(R1,B)=100 \\
&\text{R2} \quad T(R2) = 2000 \quad V(R2,B)=200 \quad V(R2,C)=300 \\
&\text{R3} \quad T(R3) = 3000 \quad V(R3,C)=90 \quad V(R3,D)=500
\end{align*}
\]

**Partial Result:**  \( U = R1 \bowtie R2 \)

\[
\begin{align*}
T(U) &= \frac{1000 \times 2000}{200} \quad V(U,A) = 50 \\
&\quad \quad V(U,B) = 100 \quad V(U,C) = 300
\end{align*}
\]

**Z = U \bowtie R3**

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

**A Note on Histograms**

\[
\begin{align*}
\sigma_{A=val(R)} &= ?
\end{align*}
\]

number of tuples in \( R \) with \( A \) value in given range
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 I/Os — next...

• Generate and compare plans