CS 245: Database System Principles

Notes 6: Query Processing

Hector Garcia-Molina

Query Processing

Q → Query Plan

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

R

A	B	C
a	1	10
b	1	20
c	2	10
d	2	35
e	3	45

S

C	D	E
10	x	2
20	y	2
30	z	2
40	x	1
50	y	3

R

A	B	C
a	1	10
b	1	20
c	2	10
d	2	35
e	3	45

S

C	D	E
10	x	2
20	y	2
30	z	2
40	x	1
50	y	3

Answer

B	D
2	x

- How do we execute query?

One idea

- Do Cartesian product
- Select tuples
- Do projection

RXS

R.A	R.B	R.C	S.C	S.D	S.E
a	1	10	10	x	2
a	1	10	20	y	2
.					
.					
C	2	10	10	x	2
.					
.					

RXS

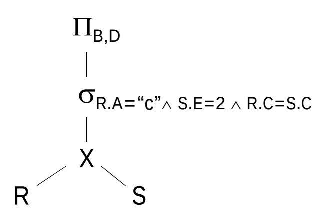
R.A	R.B	R.C	S.C	S.D	S.E
a	1	10	10	x	2
a	1	10	20	y	2
.					
.					
C	2	10	10	x	2
.					
.					

Bingo!

Got one...

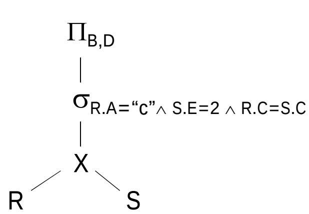
Relational Algebra - can be used to describe plans...

Ex: Plan I



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

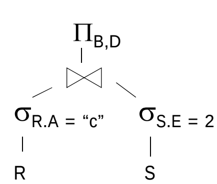
Ex: Plan I



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

Another idea:

Plan II



$\bowtie$ natural join

R

A	B	C
a	1	10
b	1	20
c	2	10
d	2	35
e	3	45

σ (R)

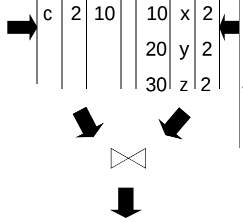
A	B	C
c	2	10

σ(S)

C	D	E
10	x	2
20	y	2
30	z	2

S

C	D	E
10	x	2
20	y	2
30	z	2
40	x	1
50	y	3



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

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 $\neq$ 2

(4) Join matching R,S tuples, project B,D attributes and place in result

R

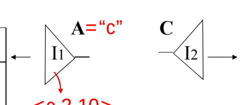
A	B	C
a	1	10
b	1	20
c	2	10
d	2	35
e	3	45

S

C	D	E
10	x	2
20	y	2
30	z	2
40	x	1
50	y	3

R

A	B	C
a	1	10
b	1	20
c	2	10
d	2	35
e	3	45

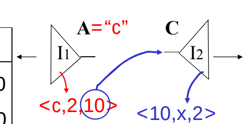


S

C	D	E
10	x	2
20	y	2
30	z	2
40	x	1
50	y	3

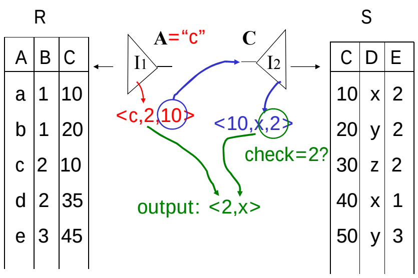
R

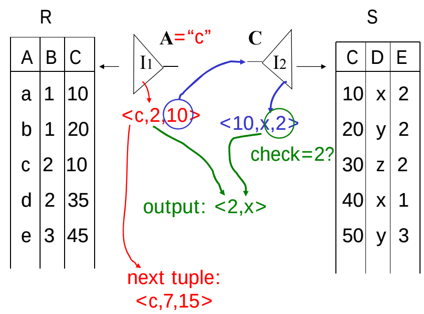
A	B	C
a	1	10
b	1	20
c	2	10
d	2	35
e	3	45



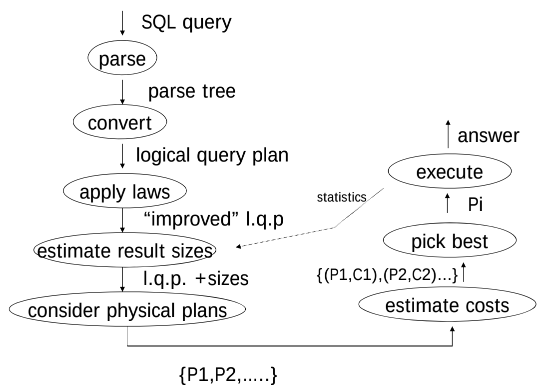
S

C	D	E
10	x	2
20	y	2
30	z	2
40	x	1
50	y	3





Overview of Query Optimization

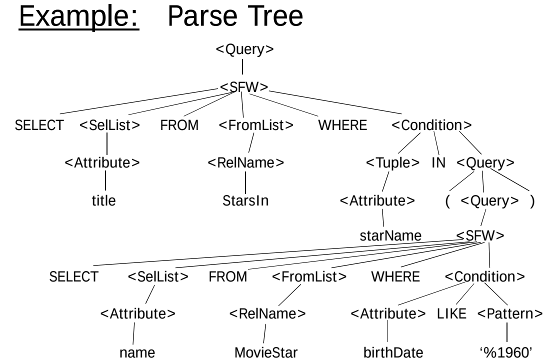


Example: SQL query

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

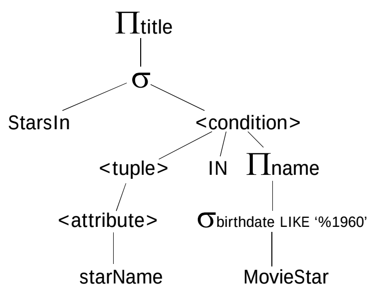


Fig. 7.15: An expression using a two-argument σ, midway between a parse tree and relational algebra

Example: Logical Query Plan

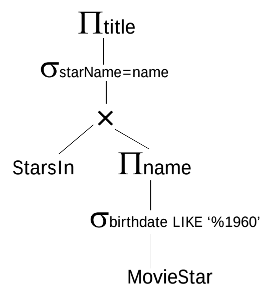
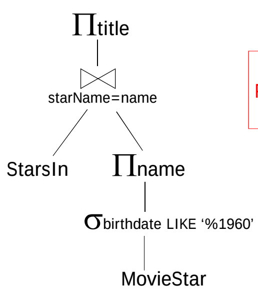


Fig. 7.18: Applying the rule for IN conditions

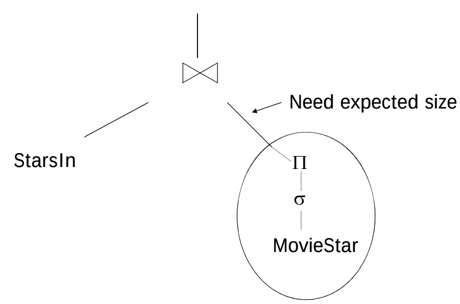
Example: Improved Logical Query Plan



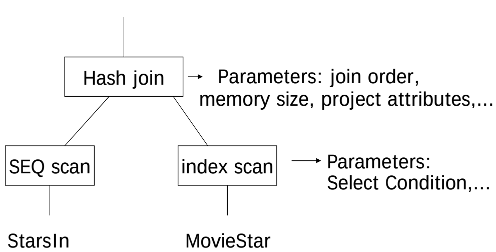
Question: Push project to StarsIn?

Fig. 7.20: An improvement on fig. 7.18.

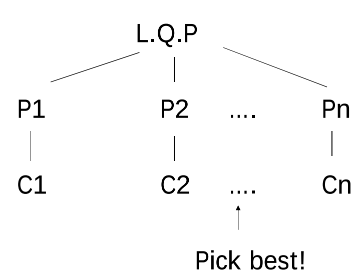
Example: Estimate Result Sizes



Example: One Physical Plan



Example: Estimate costs



Textbook outline

Chapter 15

5 [Ch 5] Algebra for queries [bags vs sets]

- Select, project, join, ... [project list a,a+b->x,...]
- Duplicate elimination, grouping, sorting

15.1 [15.1] Physical operators

- Scan,sort, ...

15.2 - 15.6 [15.2-15.6] Implementing operators + estimating their cost

Chapter 16

16.1[16.1] Parsing

16.2[16.2] Algebraic laws

16.3[16.3] Parse tree -> logical query plan

16.4[16.4] Estimating result sizes

16.5-7[16.5-7] Cost based optimization

Reading textbook - Chapters 15, 16

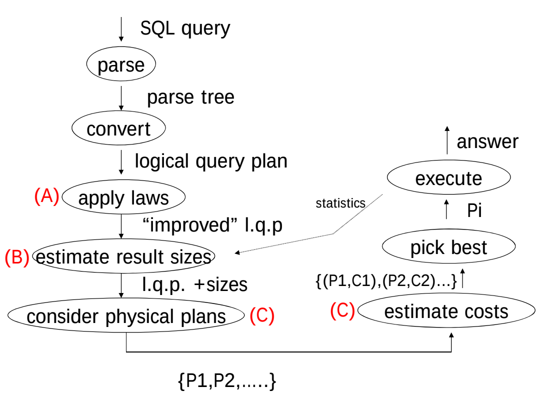
Optional:

- 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?

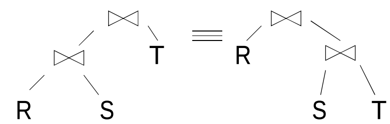
Rules: Natural joins & cross products & union

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

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

Note:

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



Rules: Natural joins & cross products & union

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

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

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

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

$\sigma_{p1 v p2}(R) =$

Rules: Selects

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

$\sigma_{p1 v p2}(R) = \quad [\, \sigma_{p1} \, (R)] \; U \; [\, \sigma_{p2} \, (R)]$

Bags vs. Sets

R = {a,a,b,b,b,c}

S = {b,b,c,c,d}

RUS = ?

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:

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

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

P1 satisfied by a,b; P2 satisfied by b,c

Option 2 (MAX) makes this rule work:

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

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

P1 satisfied by a,b; P2 satisfied by b,c

$\sigma_{p1 \vee p2}(R)$ = {a,a,b,b,b,c}

$\sigma_{p1}(R)$ = {a,a,b,b,b}

$\sigma_{p2}(R)$ = {b,b,b,c}

$\sigma_{p1}(R) \cup \sigma_{p2}(R)$ = {a,a,b,b,b,c}

"Sum" option makes more sense:

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

T1 = $\pi_{yr,state}$ Senators; T2 = $\pi_{yr,state}$ Reps

T1

Yr	State
14	CA
16	CA
15	AZ

T2

Yr	State
16	CA
16	CA
15	CA

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

$\pi_{xy}(R)$ =

Rules: Project

Let: X = set of attributes
Y = set of attributes
XY = X U Y

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

Rules: Project

Let: X = set of attributes
Y = set of attributes
XY = X U Y

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

Rules: $\sigma + \bowtie$ combined

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

$\sigma_p(R \bowtie S) =$

$\sigma_q(R \bowtie S) =$

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

Some Rules can be Derived:

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

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

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

Do one, others for homework:

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

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

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

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

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

--> Derivation for first one:

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

--> Derivation for first one:

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

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

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

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

<u>Rules:</u> π,σ combined

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

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

<u>Rules:</u> π,σ combined

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

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

<u>Rules:</u> π,σ combined

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

$\pi_x[\sigma_p (R)] = \pi_x \{\sigma_p [\overset{\pi_{xz}}{\cancel{\pi_x}} (R)]\}$

<u>Rules:</u> $\pi, \bowtie$ combined

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

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

Rules: π, $\bowtie$ combined

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

π_{xy} (R $\bowtie$ S) =

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

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

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

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

z' = z U {attributes used in P }

Rules for σ, π combined with X

similar...

e.g., σ_p (R X S) = ?

Rules σ, U combined:

σ_p(R U S) = σ_p(R) U σ_p(S)

σ_p(R - S) = σ_p(R) - S = σ_p(R) - σ_p(S)

Which are "good" transformations?

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

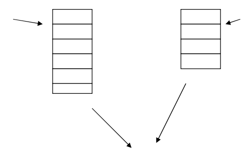
Conventional wisdom:
do projects early

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

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

$\boxed{\text{But}}$ 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

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

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

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) =

S(W) =

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

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

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

what is probability this tuple will be in answer?

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

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

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

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

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

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

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

what is probability this tuple will be in answer?

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

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

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

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

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

$T(W) = ?$

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

$T(W) = ?$

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

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 $\updownarrow$ Max=20

$V(R,Z)=10$

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

- Solution # 3: Estimate values in range

Example R

	Z

Min=1 $\updownarrow$ 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)

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

Equivalently:

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

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

<u>Size estimate</u> for $W = R1 \bowtie R2$

Let x = attributes of R1

y = attributes of R2

<u>Size estimate</u> for $W = R1 \bowtie R2$

Let x = attributes of R1

y = attributes of R2

Case 1

$X \cap Y = \emptyset$

Same as R1 x R2

Case 2 $W = R1 \bowtie R2$ $X \cap Y = A$

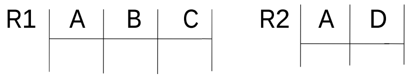
R1

A	B	C

R2

A	D

$\boxed{\text{Case 2}}$ $W = R1 \bowtie R2$ $X \cap Y = A$



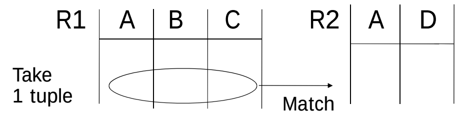
Assumption:

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

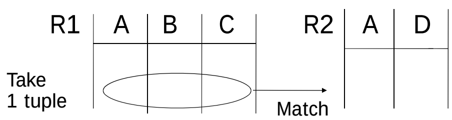
$V(R2,A) \leq V(R1,A) \Rightarrow$ Every A value in R2 is in R1

"containment of value sets" Sec. 7.4.4

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



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



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

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

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

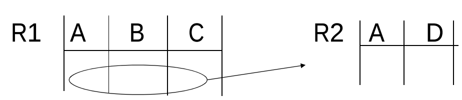
[A is common attribute]

In general $W = R1 \bowtie R2$

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

$\boxed{\text{Case 2}}$ 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

Using similar ideas,
we can estimate sizes of:

Π_{AB} (R) ..... Sec. 16.4.2 (same for either edition)

$\sigma_{A=a \wedge B=b}$ (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. $W = [\underbrace{\sigma_{A=a}(R1)}_{\text{Treat as relation U}}] \bowtie R2$

$T(U) = T(R1)/V(R1,A)$ $S(U) = S(R1)$

Also need V (U, *) !!

To estimate Vs

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

R1

A	B	C	D
cat	1	10	10
cat	1	20	20
dog	1	30	10
dog	1	40	30
bat	1	50	10

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

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

Example

R1

A	B	C	D
cat	1	10	10
cat	1	20	20
dog	1	30	10
dog	1	40	30
bat	1	50	10

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 $U = R1(A,B) \bowtie R2(A,C)$

$V(U,A) = \min\{ V(R1, A), V(R2, A) \}$
$V(U,B) = V(R1, B)$
$V(U,C) = V(R2, C)$

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

Example:

$Z = R1(A,B) \bowtie R2(B,C) \bowtie R3(C,D)$

$\boxed{R1}$	$T(R1) = 1000$	$V(R1,A)=50$	$V(R1,B)=100$
$\boxed{R2}$	$T(R2) = 2000$	$V(R2,B)=200$	$V(R2,C)=300$
$\boxed{R3}$	$T(R3) = 3000$	$V(R3,C)=90$	$V(R3,D)=500$

Partial Result: $U = R1 \bowtie R2$

$T(U) = \frac{1000 \times 2000}{200}$

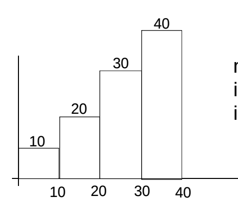
$V(U,A) = 50$
$V(U,B) = 100$
$V(U,C) = 300$

$Z = U \bowtie R3$

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

$V(Z,A) = 50$
$V(Z,B) = 100$
$V(Z,C) = 90$
$V(Z,D) = 500$

A Note on Histograms



number of tuples in R with A value in given range

$\sigma_{A=val}(R) = ?$

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