Ullman et al. : Database System Principles

Notes 6: Query Processing

Query Processing

Q $\rightarrow$ 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

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

$\Pi_{B,D}$

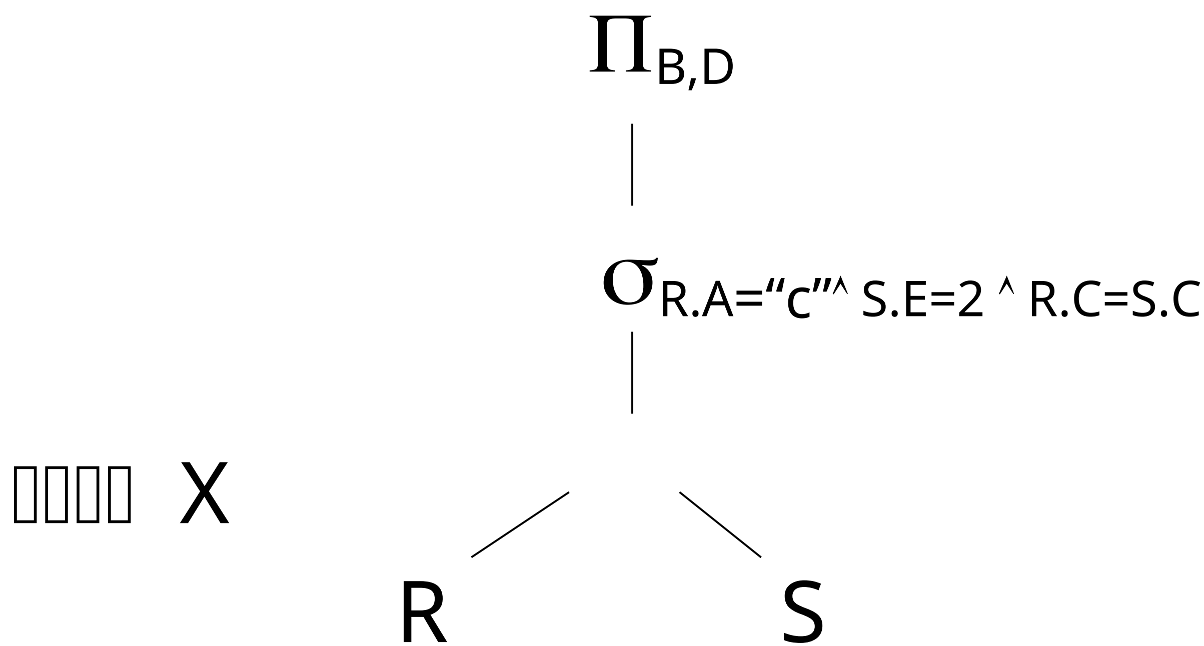
$\sigma_{R.A=\text{“c”} \wedge S.E=2 \wedge R.C=S.C}$

 X

R S

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

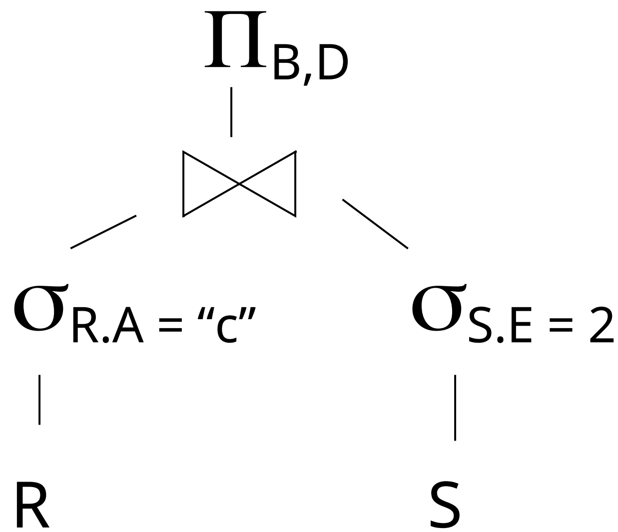
Ex: Plan I



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

$\bowtie$

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

I_1 **A**="c"

<c,2,10>

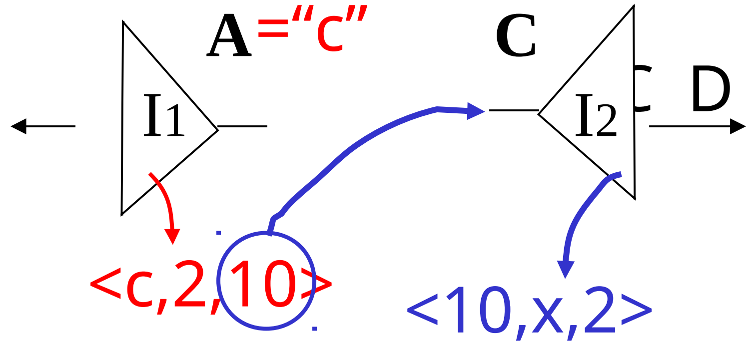
C I_2

S

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

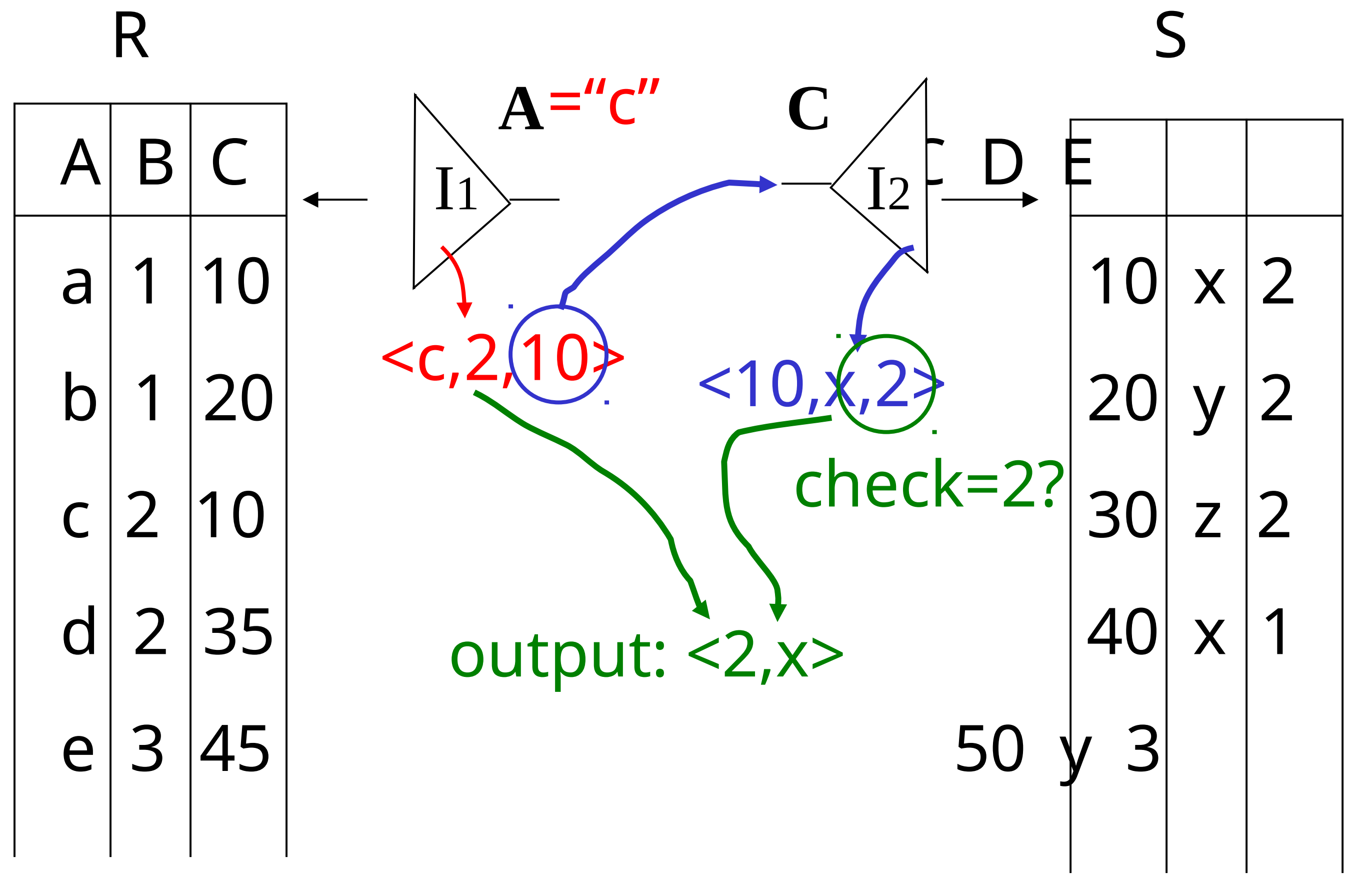
R

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

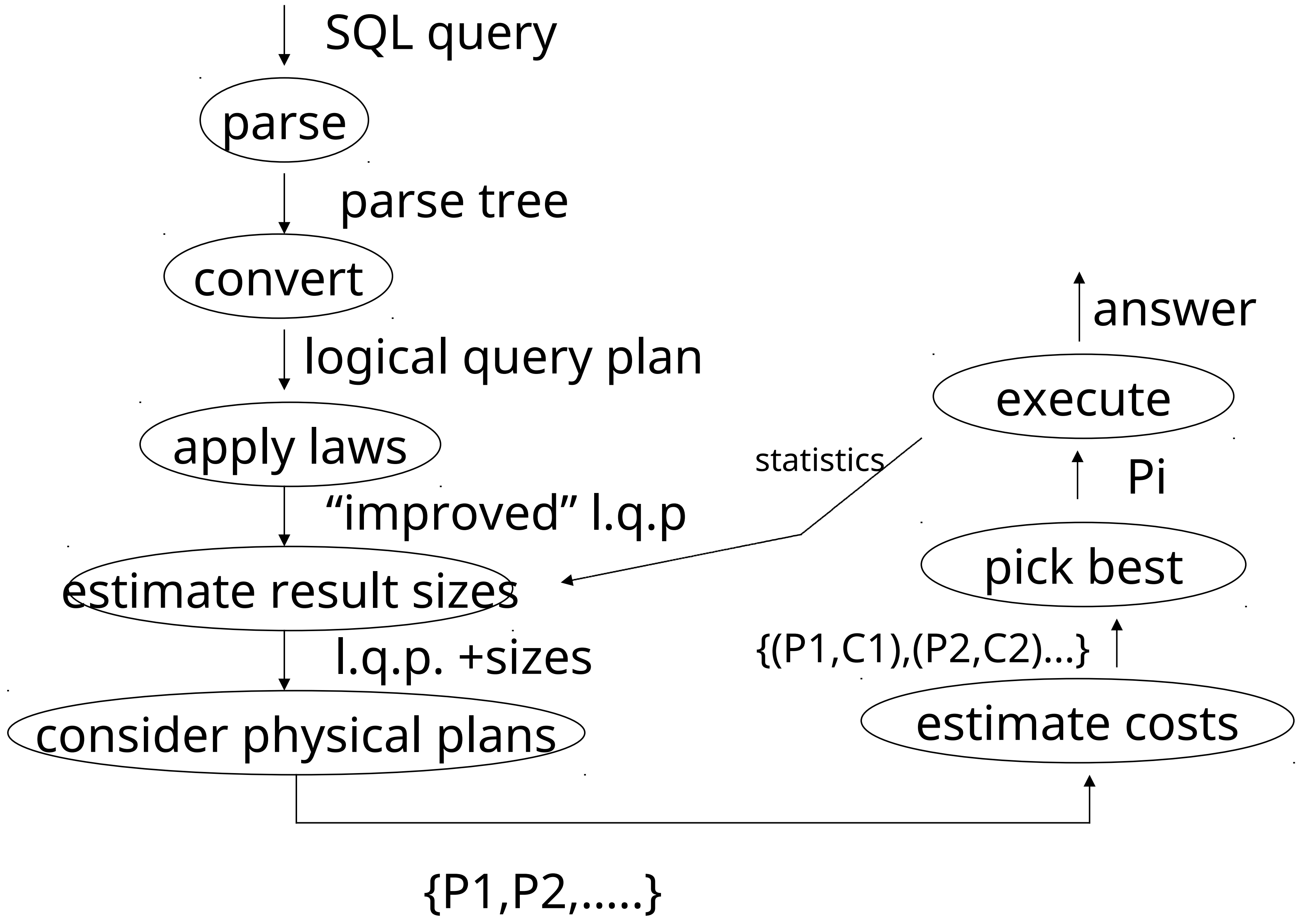


S

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


R
S
A B C
a 1 10
b 1 20
c 2 10
d 2 35
e 3 45
A="c"
C
I1
I2
C D E
10 x 2
20 y 2
30 z 2
40 x 1
50 y 3
<c,2,10>
<10,x,2>
check=2?
output: <2,x>

Overview of Query Optimization

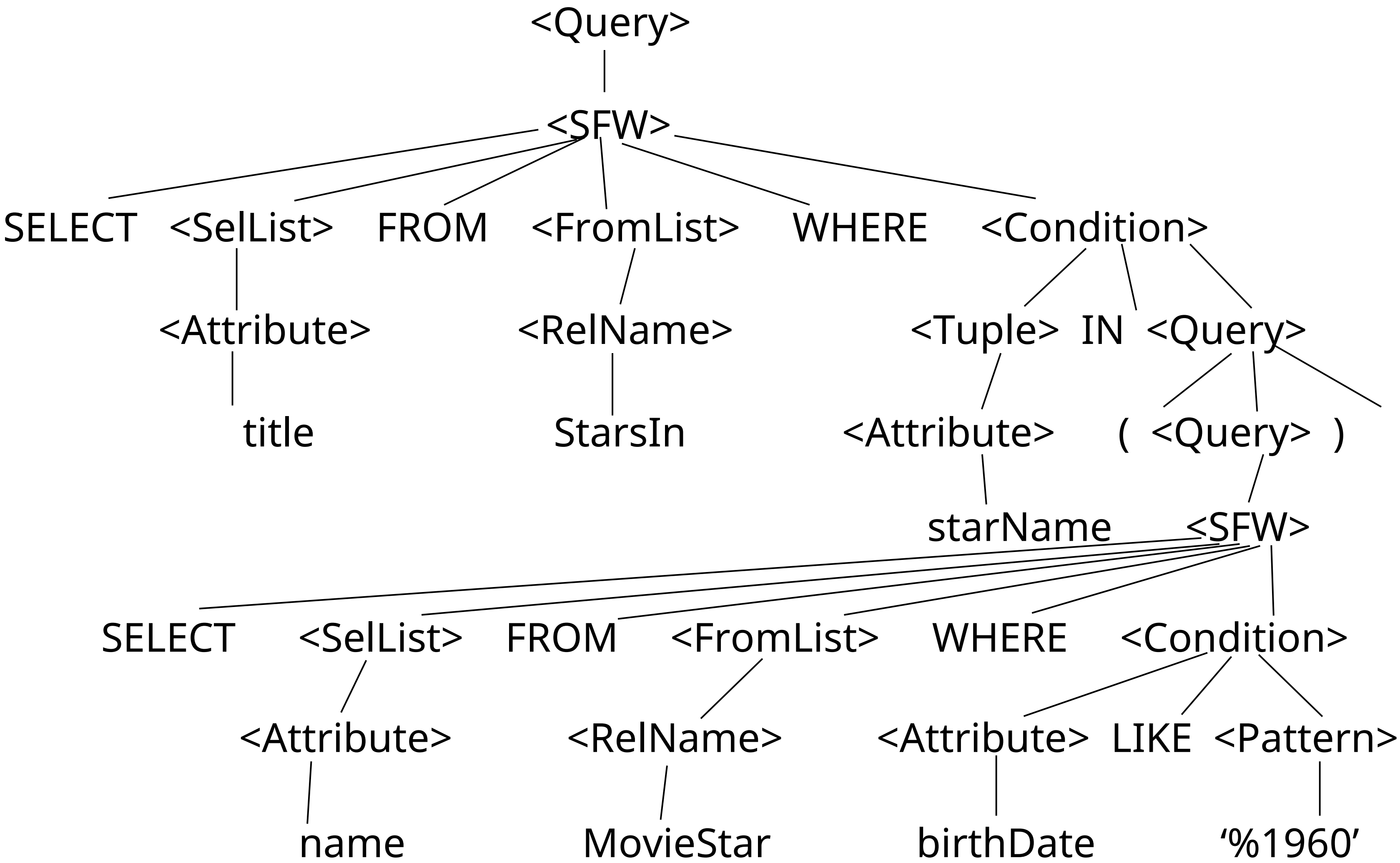

SQL query
parse
parse tree
convert
logical query plan
apply laws
"improved" l.q.p
estimate result sizes
l.q.p. +sizes
consider physical plans
{P1,P2,.....}
estimate costs
{(P1,C1),(P2,C2)...}
pick best
Pi
execute
answer
statistics

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

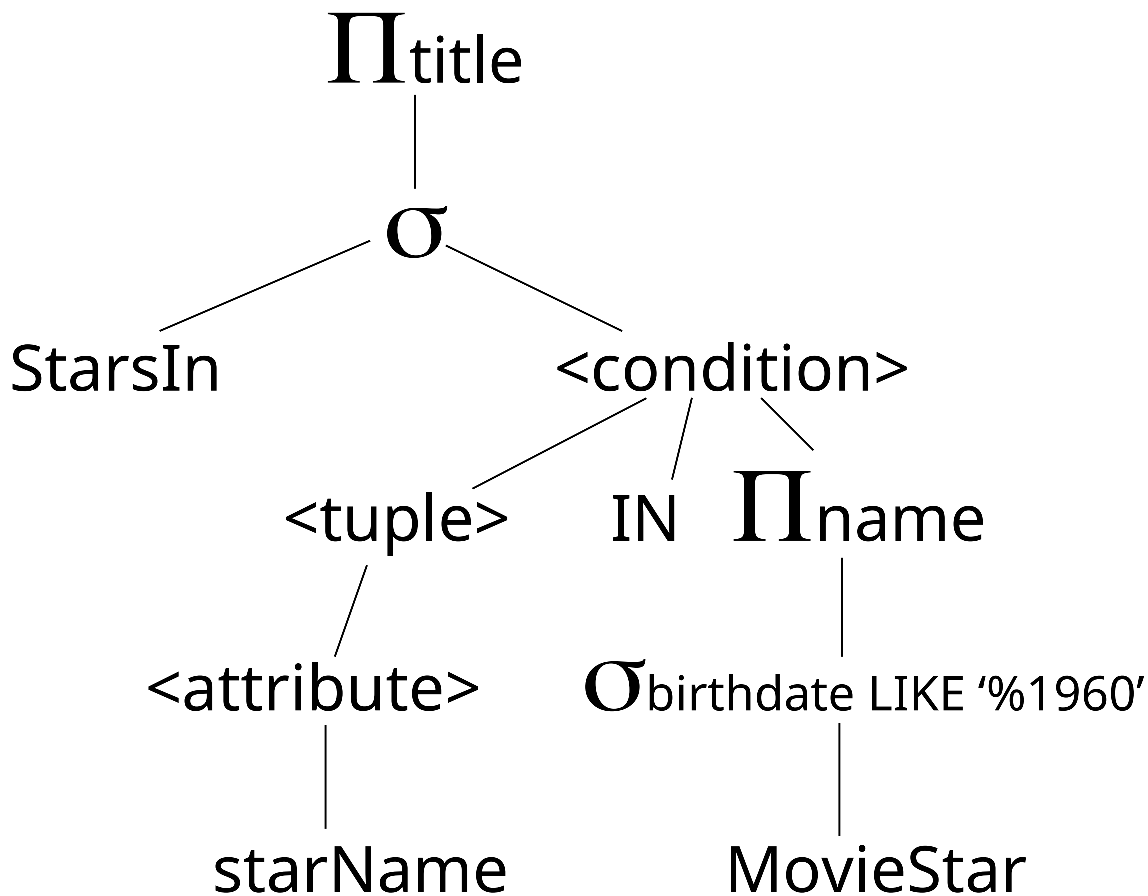


Fig. 7.15: An expression using a two-argument σ, midway between a parse tree and relational algebra

Example: Logical Query Plan

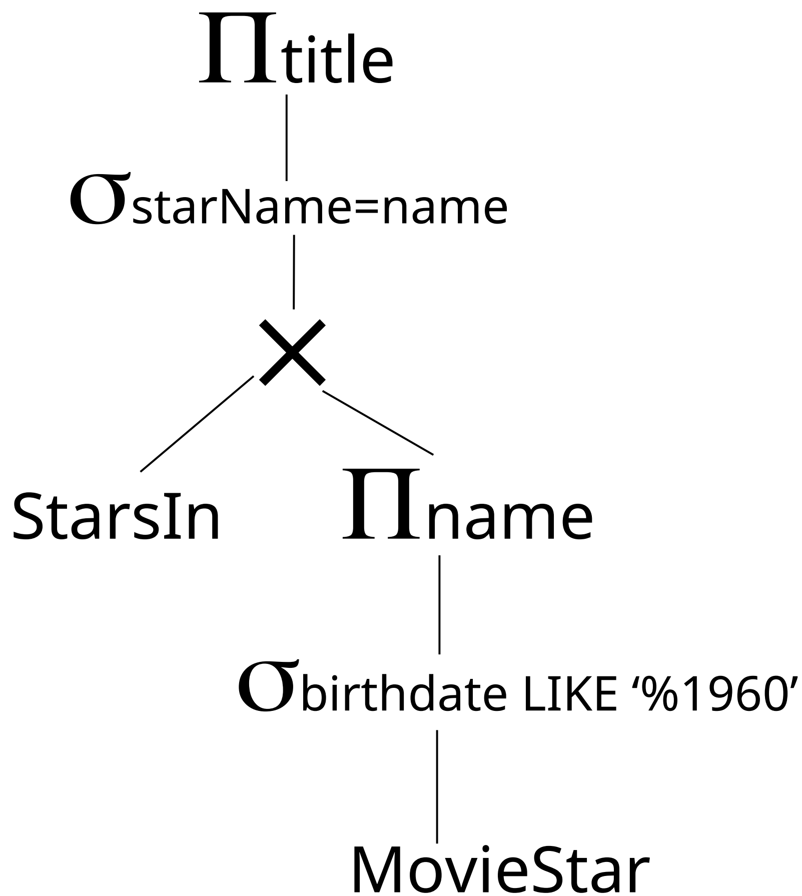
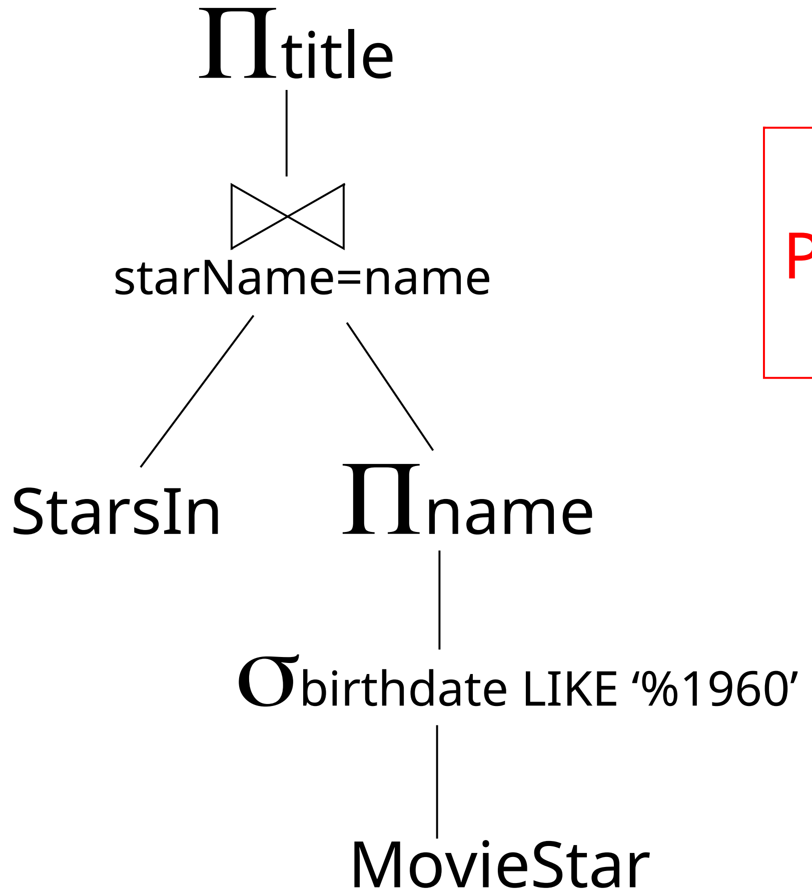


Fig. 7.18: Applying the rule for IN conditions

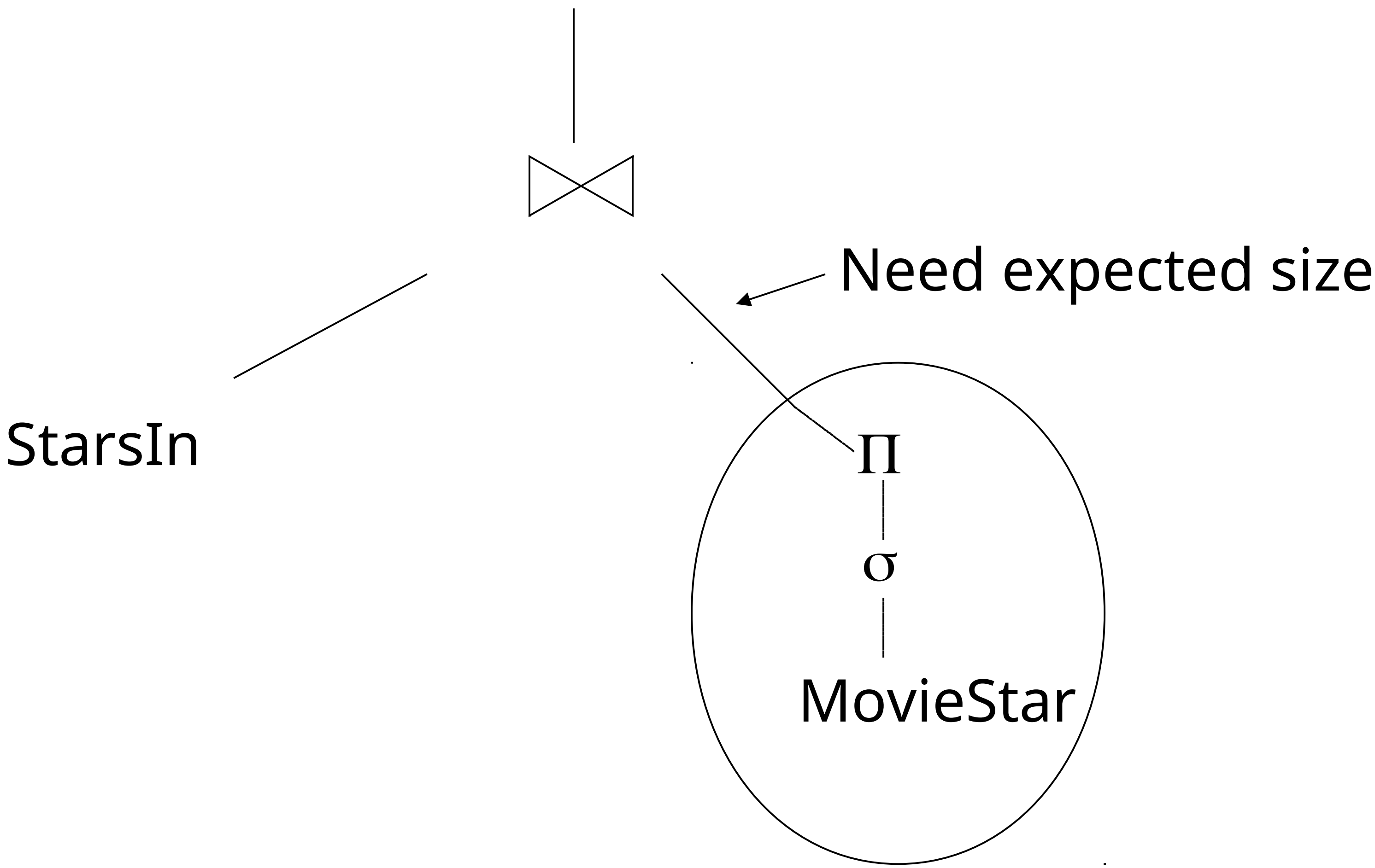
Example: Improved Logical Query Plan



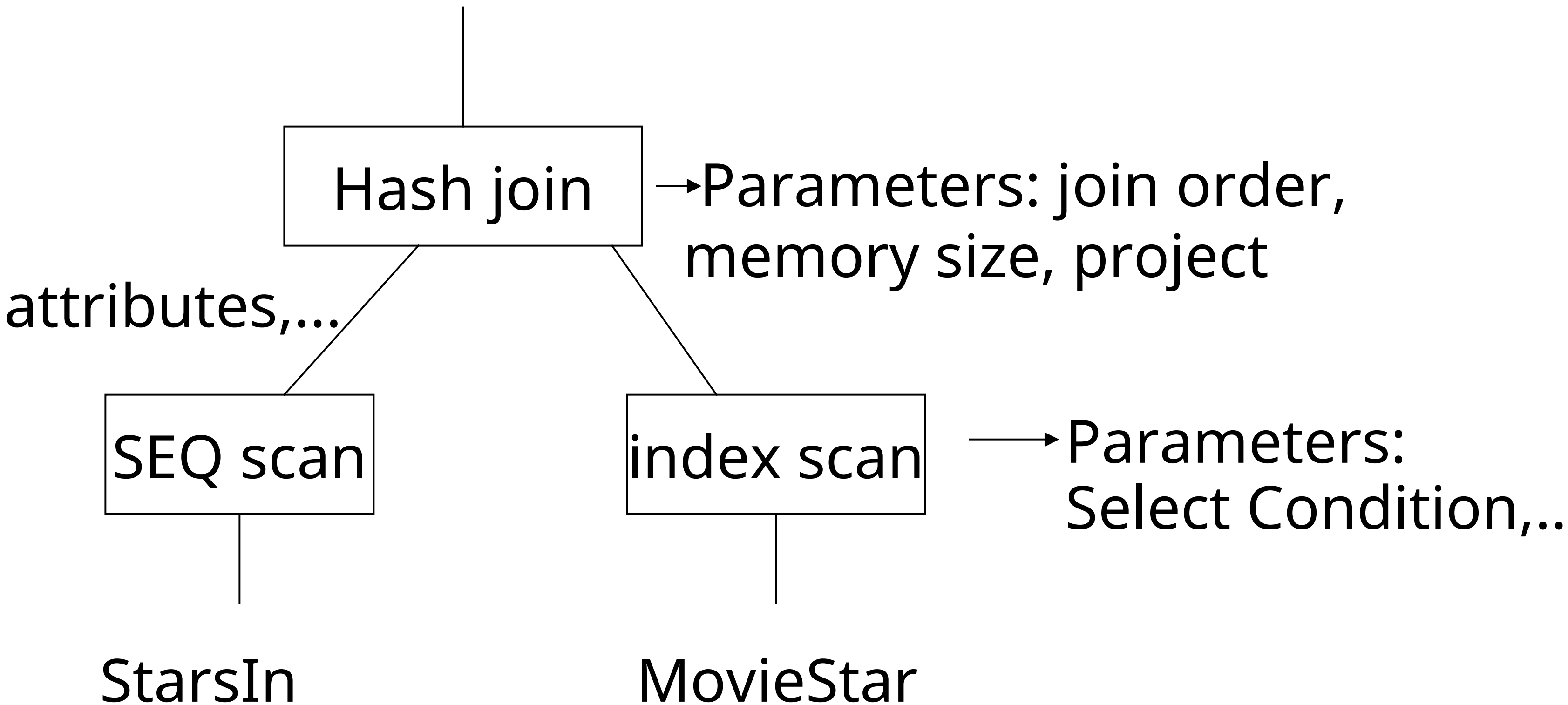
Question:
Push project to
StarsIn?

Fig. 7.20: An improvement on fig. 7.18.

Example: Estimate Result Sizes



Example: One Physical Plan



<u>Example:</u> Estimate costs

L.Q.P

P1 P2 .... Pn

C1 C2 .... Cn

Pick best!

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?

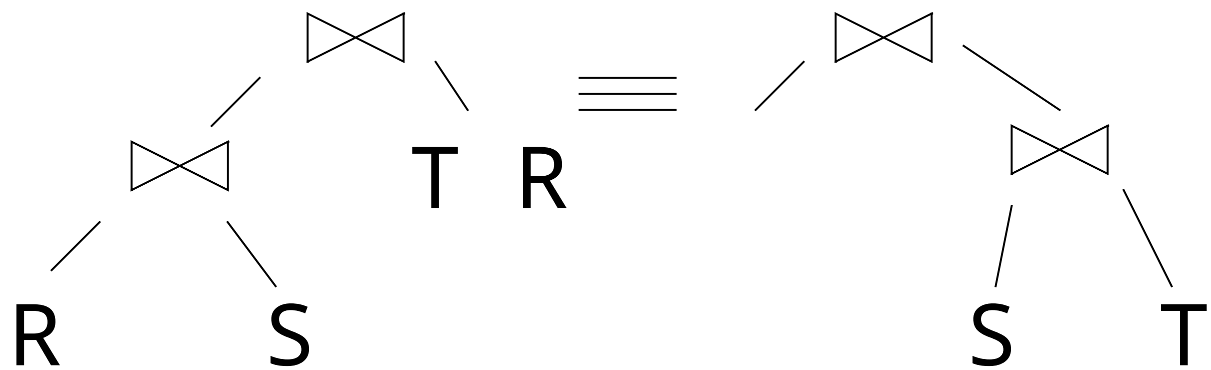
<u>Rules:</u> Natural joins & cross products & union

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

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

Note:

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



Rules: Natural joins & cross products & union

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

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

$$R \times S = S \times R$$

$$(R \times S) \times T = R \times (S \times T)$$

$$R \cup S = S \cup R$$

$$R \cup (S \cup T) = (R \cup S) \cup T$$

Rules: Selects

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

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

<u>Rules:</u> Project

Let: X = set of attributes

Y = set of attributes

XY = X U Y

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

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

Some Rules can be Derived:

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

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

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

Rules: π,σ combined

Let x = subset of R attributes

z = attributes in predicate P

(subset of R attributes)

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

Rules: π, $\bowtie$ 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 σ, π combined with X

similar...

e.g., σ_p (R X S) = ?

Rules σ, $\cup$ 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?

- $\sigma_{p1 \wedge p2}(R) \rightarrow \sigma_{p1}[\sigma_{p2}(R)]$
- $\sigma_{p}(R \bowtie S) \rightarrow [\sigma_{p}(R)] \bowtie S$
- $R \bowtie S \rightarrow S \bowtie R$
- $\pi_{x}[\sigma_{p}(R)] \rightarrow \pi_{x}\{\sigma_{p}[\pi_{xz}(R)]\}$

Conventional wisdom: do projects early

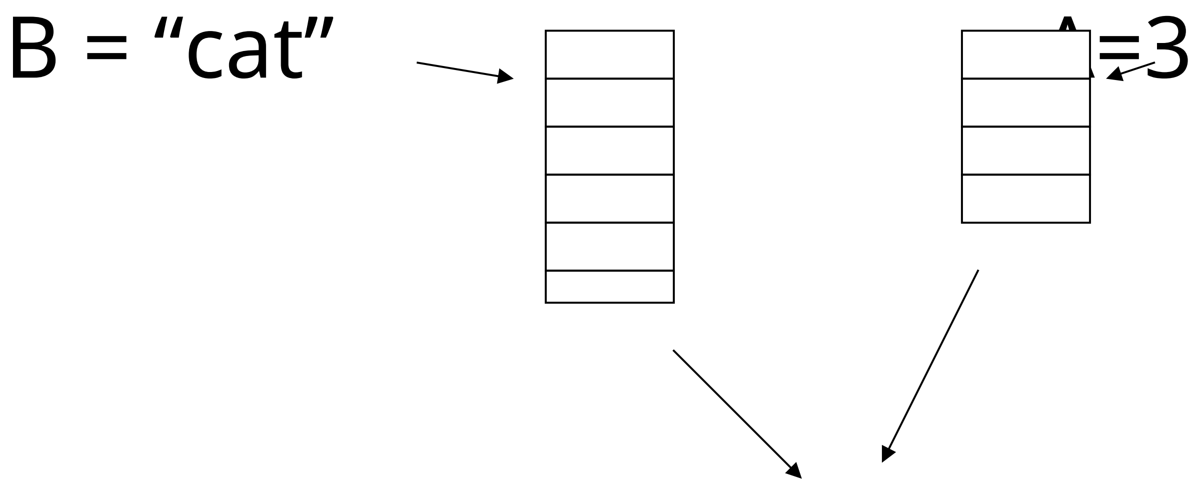
Example: R(A,B,C,D,E)   x={E}

P: (A=3) $\wedge$ (B="cat")

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

Usually good: early selections

But What if we have A, B indexes?



Intersect pointers to get
pointers to matching tuples

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
- L(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
- b: block size
- bf(R) (blocking factor): # of tuples in a block

  bf(R) = b/L(R)

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 L(R) = 37

V(R,A) = 3 V(R,C) = 5

V(R,B) = 1 V(R,D) = 4

Size estimates for $W = R \times S$

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

$L(W) = L(R) + L(S)$

$bf(W) = b/(L(R)+L(S))$

$B(W) = T(R)*T(S)/bf(W) =$

$= T(R)*T(S)*L(S)/b + T(S)*T(R)*L(R)/b =$

$= T(R)*T(S)/bf(S) + T(S)*T(R)/bf(R) =$

$= T(R)*B(S) + T(S)*B(R)$

<u>Size estimate</u> for $W = \sigma_{A=a}(R)$

$L(W) = L(R)$

$T(W) = ?$

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

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

Selection cardinality

SC(R,A) = average # records that satisfy
equality condition on R.A

SC(R,A) = T(R) / V(R,A)

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

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

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

Size estimate for $W = R1 \bowtie R2$

Let x = attributes of R1

y = attributes of R2

$\boxed{\text{Case 1}}$

$$X \cap Y = \emptyset$$

Same as R1 x R2

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

R1	A	B	C

R2	A	D

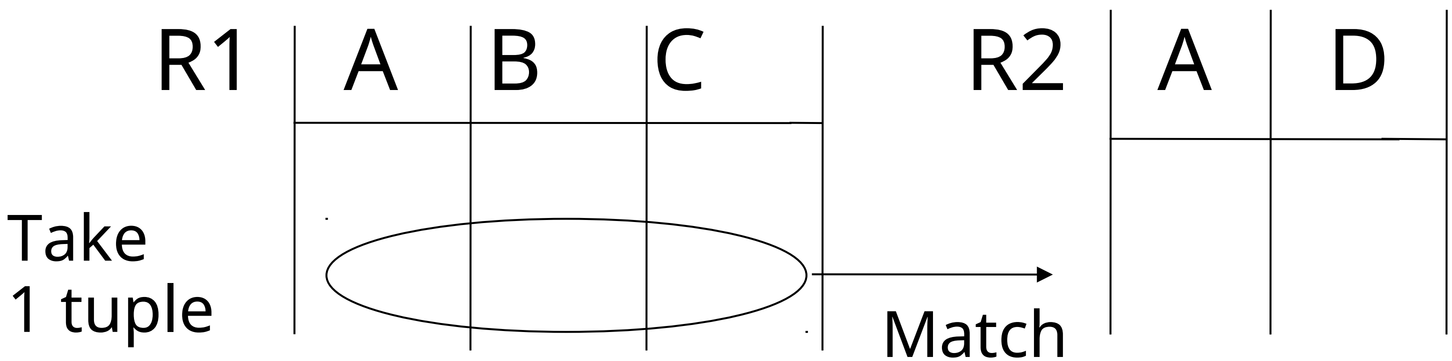
Assumption:

$V(R1,A) \leq V(R2,A) \Rightarrow$ 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"

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



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

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

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

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

[A is common attribute]

In general $W = R1 \bowtie R2$

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

Size Estimation Summary (1/2)

$\sigma_{A=v}(R)$ SC(R,A) (--> SC(R,A) = T(R) / V(R,A))

$\sigma_{A\le v}(R)$ $T(R)*\frac{v-\min(A,R)}{\max(A,R)-\min(A,R)}$

$\sigma_{\theta_1\wedge\theta_2\wedge\ldots\wedge\theta_n}(R)$ multiplying probabilities

$$T(R)*[(sc_1/T(R))*(sc_2/T(R))*\ldots(sc_n/T(R))]$$

$\sigma_{\theta_1\vee\theta_2 v\ldots\vee\theta_n}(R)$ probability that a record satisfy none of θ:

$$[(1-sc_1/T(R))*(1-sc_2/T(R))*\ldots*(1-sc_n/T(R))]$$

$$T(R)*(1-[(1-sc_1/T(R))*(1-sc_2/T(R))*\ldots*(1-sc_n/T(R))])$$

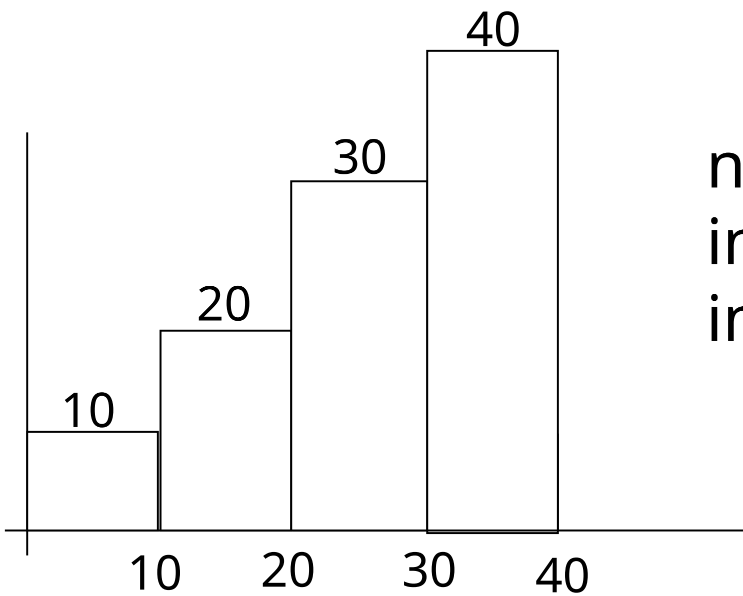
Size Estimation Summary(2/2)

- R x S

  T(RxS) = T(R)*T(S)

- R $\bowtie$ S
  - R ∩ S = ∅: T(R) * T(S)
  - R ∩ S key for R: maximum output size is T(S)
  - R ∩ S foreign key for R: T(S)
  - R ∩ S = {A}, neither key of R nor S
    - T(R) * T(S) / V(S,A)
    - T(R) * T(S) / V(R,A)

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?)