

CSE544

Data Management

Lecture14

Query Optimization – Part 3

Announcements

- HW3 due on Friday
- Review 6 due on Monday:
 - Read the entire paper
 - Will discuss in class
- Project: MAJOR milestone next Friday!

Query Optimization

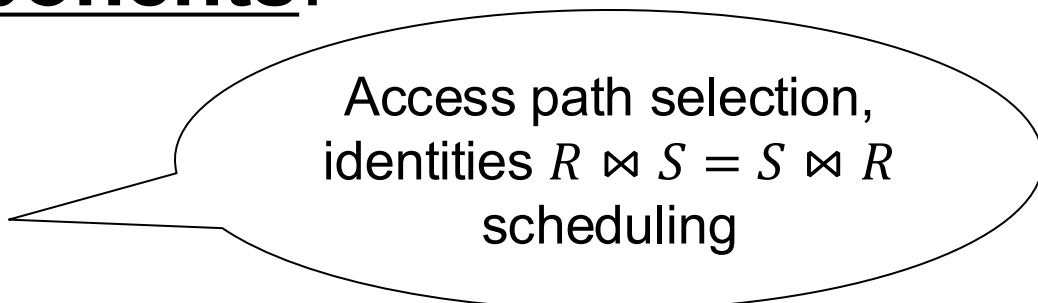
Three major components:

1. Search space
2. Cardinality and cost estimation
3. Plan enumeration algorithms

Query Optimization

Three major components:

1. Search space



Access path selection,
identities $R \bowtie S = S \bowtie R$
scheduling

2. Cardinality and cost estimation

3. Plan enumeration algorithms

Query Optimization

Three major components:

1. Search space

Access path selection,
identities $R \bowtie S = S \bowtie R$
scheduling

2. Cardinality and cost estimation

Today

3. Plan enumeration algorithms

Query Optimization

Three major components:

1. Search space

Access path selection,
identities $R \bowtie S = S \bowtie R$
scheduling

2. Cardinality and cost estimation

Today

3. Plan enumeration algorithms

Wednesday

Cost Estimation

Problem: compute the cost of each candidate query plan

Two steps

- Estimate **cardinality** of all intermediate results
- Compute the **cost** using these estimates

Cost Estimation

Problem: compute the cost of each candidate query plan

Two steps

- Estimate **cardinality** of all intermediate results
- Compute the **cost** using these estimates



Hard



Easier

Cardinality Estimation

Problem: given statistics on base tables and a query, estimate size of the answer

Very difficult, because:

- Need to do it very fast
- Need to use very little memory

Cardinality Estimation

- Traditional method:
 - Simple probability space on the set of possible tuples
- Alternate methods:
 - Based on ML
 - Based on sampling
 - Upper bounds -- next week

Traditional Cardinality Estimation

Statistics on Base Data

- Number of tuples (cardinality) $T(R)$
- Number of physical pages $B(R)$
- Indexes, number of keys in the index $V(R,a)$

Statistics on Base Data

- Number of tuples (cardinality) $T(R)$
- Number of physical pages $B(R)$
- Indexes, number of keys in the index $V(R,a)$
- Histogram on single attribute (1d)
- Histogram on two attributes (2d)

Computed periodically, often using sampling

Probability Space

```
SELECT *                               --- attrs A1, A2, ..., An
FROM R1, R2, ..., Rm
WHERE condition
```

Probability Space

```
SELECT *                               --- attrs A1, A2, ..., An
FROM R1, R2, ..., Rm
WHERE condition
```

Probability space:

- Outcomes: $R_1 \times R_2 \times \cdots \times R_m$
- Probability: $p(A_1, A_2, \dots, A_n)$

Cardinality estimate: $Est = p(condition) \cdot T(R_1) \cdots T(R_m)$

Probability Space

```
SELECT *                --- attrs A1, A2, ..., An
FROM R1, R2, ..., Rm
WHERE condition
```

Probability space:

- Outcomes: $R_1 \times R_2 \times \cdots \times R_m$
- Probability: $p(A_1, A_2, \dots, A_n)$

Cardinality estimate: $Est = p(condition) \cdot T(R_1) \cdots T(R_m)$

Goal: compute $p(condition)$

Assumptions

- Uniformity
- Independence
- Containment of values
- Preservation of values

Selectivity Factors

$p(pred)$ is called selectivity factor θ

$$\text{Est}(\sigma_{pred}(R)) = \theta_{pred} * T(R)$$

$$\text{Est}(R \bowtie_{A=B} S) = \theta_{A=B} * T(R) * T(S)$$

Selectivity Factors

Uniformity+Containment of values

$$\sigma_{A=c}(R)$$

Selectivity Factors

Uniformity+Containment of values

$$\sigma_{A=c}(R)$$

Equality:

- $\theta_{A=c} = 1/V(R,A)$

Selectivity Factors

Uniformity+Containment of values

$$\sigma_{A=c}(R)$$

Equality:

- $\theta_{A=c} = 1/V(R,A)$

$$\sigma_{c1 < A < c2}(R)$$

Range:

- $\theta_{c1 < A < c2} = (c2 - c1)/(\max(R,A) - \min(R,A))$

Selectivity Factors

Uniformity+Containment of values

Equality:

- $\theta_{A=c} = 1/V(R,A)$

$$\sigma_{A=c}(R)$$

$$\sigma_{c1 < A < c2}(R)$$

Range:

- $\theta_{c1 < A < c2} = (c2 - c1)/(\max(R,A) - \min(R,A))$

$$\sigma_{A=c \text{ and } B=d}(R)$$

Selectivity Factors

Uniformity+Containment of values

$$\sigma_{A=c}(R)$$

Equality:

- $\theta_{A=c} = 1/V(R,A)$

$$\sigma_{c1 < A < c2}(R)$$

Range:

- $\theta_{c1 < A < c2} = (c2 - c1) / (\max(R,A) - \min(R,A))$

$$\sigma_{A=c \text{ and } B=d}(R)$$

Independence assumption

- $\theta_{\text{pred1 and pred2}} = \theta_{\text{pred1}} * \theta_{\text{pred2}} = 1/V(R,A) * 1/V(R,B)$

Supplier(sid, sname, scity, sstate)

$T(\text{Supplier}) = 100000$
 $V(\text{Supplier}, \text{scity}) = 2000$
 $V(\text{Supplier}, \text{sstate}) = 50$

Example

```
SELECT *  
FROM Supplier  
WHERE scity = 'Seattle'
```

Est = **????**

Supplier(sid, sname, scity, sstate)

$T(\text{Supplier}) = 100000$
 $V(\text{Supplier}, \text{scity}) = 2000$
 $V(\text{Supplier}, \text{sstate}) = 50$

Example

```
SELECT *  
FROM Supplier  
WHERE scity = 'Seattle'
```

$$\text{Est} = 1/2000 * 100000 = 50$$

Supplier(sid, sname, scity, sstate)

$T(\text{Supplier}) = 100000$
 $V(\text{Supplier}, \text{scity}) = 2000$
 $V(\text{Supplier}, \text{sstate}) = 50$

Example

```
SELECT *  
FROM Supplier  
WHERE scity = 'Seattle'
```

$$\text{Est} = 1/2000 * 100000 = 50$$

```
SELECT *  
FROM Supplier  
WHERE sstate = 'WA'
```

$$\text{Est} = 1/50 * 100000 = 2000$$

Supplier(sid, sname, scity, sstate)

$T(\text{Supplier}) = 100000$
 $V(\text{Supplier}, \text{scity}) = 2000$
 $V(\text{Supplier}, \text{sstate}) = 50$

Example

```
SELECT *  
FROM Supplier  
WHERE scity = 'Seattle'
```

$$\text{Est} = 1/2000 * 100000 = 50$$

```
SELECT *  
FROM Supplier  
WHERE sstate = 'WA'
```

$$\text{Est} = 1/50 * 100000 = 2000$$

```
SELECT *  
FROM Supplier  
WHERE scity = 'Seattle'  
and sstate = 'WA'
```

$$\text{Est} = \text{????}$$

Supplier(sid, sname, scity, sstate)

$T(\text{Supplier}) = 100000$
 $V(\text{Supplier}, \text{scity}) = 2000$
 $V(\text{Supplier}, \text{sstate}) = 50$

Example

```
SELECT *  
FROM Supplier  
WHERE scity = 'Seattle'
```

$$\text{Est} = 1/2000 * 100000 = 50$$

```
SELECT *  
FROM Supplier  
WHERE sstate = 'WA'
```

$$\text{Est} = 1/50 * 100000 = 2000$$

Independence

```
SELECT *  
FROM Supplier  
WHERE scity = 'Seattle'  
and sstate = 'WA'
```

$$\begin{aligned}\text{Est} &= 1/2000 * 1/50 * 100000 \\ &= 1\end{aligned}$$

Supplier(sid, sname, scity, sstate)

$T(\text{Supplier}) = 100000$
 $V(\text{Supplier}, \text{scity}) = 2000$
 $V(\text{Supplier}, \text{sstate}) = 50$

Example

```
SELECT *  
FROM Supplier  
WHERE scity = 'Seattle'
```

$$\text{Est} = 1/2000 * 100000 = 50$$

```
SELECT *  
FROM Supplier  
WHERE sstate = 'WA'
```

$$\text{Est} = 1/50 * 100000 = 2000$$

Independence

```
SELECT *  
FROM Supplier  
WHERE scity = 'Seattle'  
and sstate = 'WA'
```

$$\text{Est} = 1/2000 * 1/50 * 100000 = 1$$

Very wrong

Why?

$$R \bowtie_{R.A=S.B} S$$

Selectivity Factors

Join

- $\theta_{R.A=S.B} = 1 / (\text{MAX}(V(R,A), V(S,B))$

Why? Will explain next...

$$R \bowtie_{R.A=S.B} S$$

Selectivity Factors

Containment of values: if $V(R,A) \leq V(S,B)$, then the set of A values of R is included in the set of B values of S

- Note: this indeed holds when A is a foreign key in R, and B is a key in S

$$R \bowtie_{R.A=S.B} S$$

Selectivity Factors

Assume $V(R,A) \leq V(S,B)$

- Tuple t in R joins with $T(S)/V(S,B)$ tuples in S
- Hence $\text{Est}(R \bowtie_{A=B} S) = T(R) T(S) / V(S,B)$

$$R \bowtie_{R.A=S.B} S$$

Selectivity Factors

Assume $V(R,A) \leq V(S,B)$

- Tuple t in R joins with $T(S)/V(S,B)$ tuples in S
- Hence $\text{Est}(R \bowtie_{A=B} S) = T(R) T(S) / V(S,B)$

In general:

- $\text{Est}(R \bowtie_{A=B} S) = T(R) T(S) / \max(V(R,A), V(S,B))$
- $\theta_{R.A=S.B} = 1 / (\max(V(R,A), V(S,B)))$

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

T(Supplier) = 100000
V(Supplier, sid) = 100000

Example

```
SELECT *  
FROM Supplier x, Supply y  
WHERE x.sid = y.sid
```

T(Supply) = 5000000
V(Supply, sid) = 80000

What is the **exact** output size?

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

$T(\text{Supplier}) = 100000$
 $V(\text{Supplier}, \text{sid}) = 100000$

Example

$T(\text{Supply}) = 5000000$
 $V(\text{Supply}, \text{sid}) = 80000$

```
SELECT *  
FROM Supplier x, Supply y  
WHERE x.sid = y.sid
```

What is the **exact** output size?

$T(\text{Supply}) = 5000000$

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

$T(\text{Supplier}) = 100000$
 $V(\text{Supplier}, \text{sid}) = 100000$

Example

$T(\text{Supply}) = 5000000$
 $V(\text{Supply}, \text{sid}) = 80000$

```
SELECT *  
FROM Supplier x, Supply y  
WHERE x.sid = y.sid
```

What is the **exact** output size?

$T(\text{Supply}) = 5000000$

$$Est(\text{Supplier} \bowtie \text{Supply}) = \frac{T(\text{Supplier})T(\text{Supply})}{\max(V(\text{Supplier}, \text{sid}), V(\text{Supply}, \text{sid}))}$$

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

$T(\text{Supplier}) = 100000$
 $V(\text{Supplier}, \text{sid}) = 100000$

Example

$T(\text{Supply}) = 5000000$
 $V(\text{Supply}, \text{sid}) = 80000$

```
SELECT *  
FROM Supplier x, Supply y  
WHERE x.sid = y.sid
```

What is the **exact** output size?

$T(\text{Supply}) = 5000000$

$$Est(\text{Supplier} \bowtie \text{Supply}) = \frac{T(\text{Supplier})T(\text{Supply})}{\max(V(\text{Supplier}, \text{sid}), V(\text{Supply}, \text{sid}))}$$

Always larger
Why?

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

$T(\text{Supplier}) = 100000$
 $V(\text{Supplier}, \text{sid}) = 100000$

Example

$T(\text{Supply}) = 5000000$
 $V(\text{Supply}, \text{sid}) = 80000$

```
SELECT *  
FROM Supplier x, Supply y  
WHERE x.sid = y.sid
```

What is the **exact** output size?

$T(\text{Supply}) = 5000000$

$$Est(\text{Supplier} \bowtie \text{Supply}) = \frac{T(\text{Supplier})T(\text{Supply})}{\max(V(\text{Supplier}, \text{sid}), V(\text{Supply}, \text{sid}))}$$

$$= \frac{T(\text{Supplier})T(\text{Supply})}{V(\text{Supplier}, \text{sid})}$$

Always larger
Why?

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

$T(\text{Supplier}) = 100000$
 $V(\text{Supplier}, \text{sid}) = 100000$

Example

$T(\text{Supply}) = 5000000$
 $V(\text{Supply}, \text{sid}) = 80000$

```
SELECT *  
FROM Supplier x, Supply y  
WHERE x.sid = y.sid
```

What is the **exact** output size?

$T(\text{Supply}) = 5000000$

$$Est(\text{Supplier} \bowtie \text{Supply}) = \frac{T(\text{Supplier})T(\text{Supply})}{\max(V(\text{Supplier}, \text{sid}), V(\text{Supply}, \text{sid}))}$$

$$= \frac{T(\text{Supplier})T(\text{Supply})}{V(\text{Supplier}, \text{sid})} = T(\text{Supply})$$

Always larger
Why?

Final Assumption

Preservation of values:

For any other attribute C:

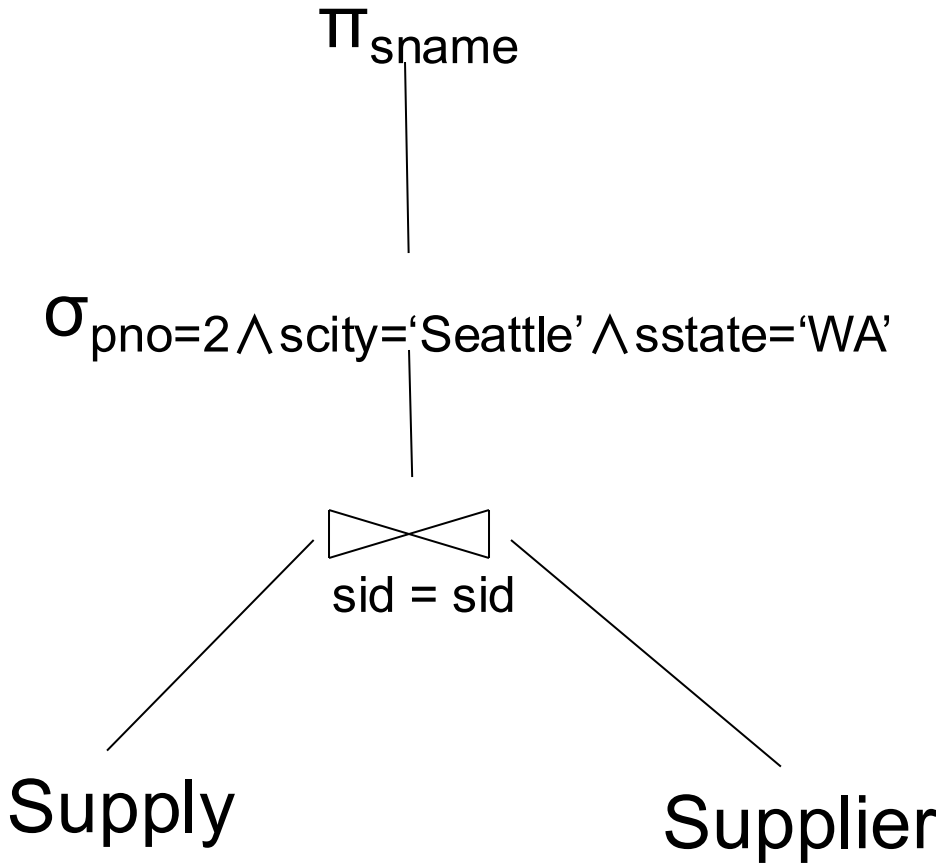
- $V(R \bowtie_{A=B} S, C) = V(R, C)$ or
- $V(R \bowtie_{A=B} S, C) = V(S, C)$
- This is needed higher up in the plan

Computing the Cost of a Plan

- Estimate **cardinalities** bottom-up
- Estimate **cost** by using estimated cardinalities
- Example next...

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Logical Query Plan 1



```
SELECT sname
FROM Supplier x, Supply y
WHERE x.sid = y.sid
      and y.pno = 2
      and x.scity = 'Seattle'
      and x.sstate = 'WA'
```

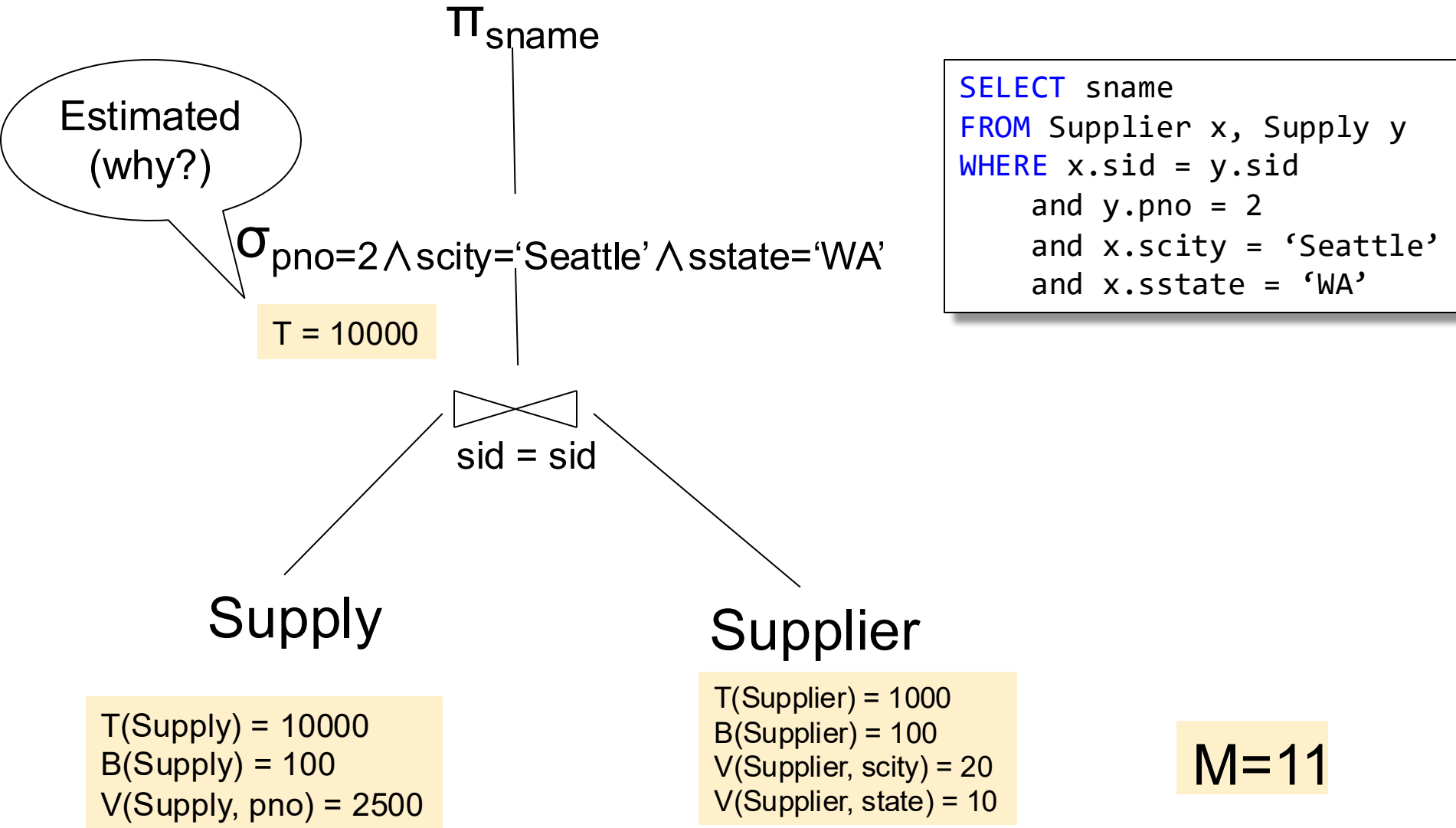
T(Supply) = 10000
B(Supply) = 100
V(Supply, pno) = 2500

T(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10

M=11

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

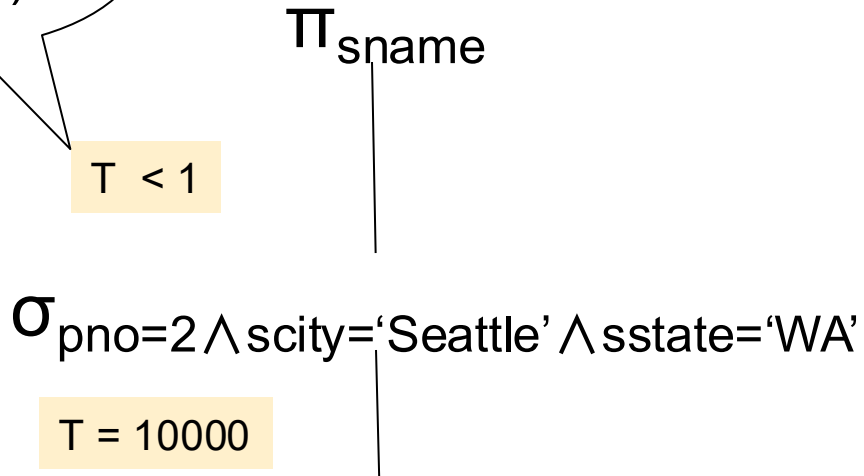
Logical Query Plan 1



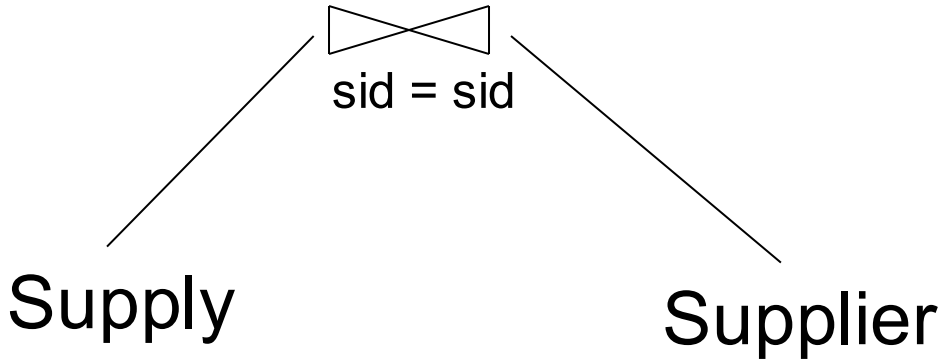
Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Estimated
(why?)

Logical Query Plan 1



```
SELECT sname
FROM Supplier x, Supply y
WHERE x.sid = y.sid
      and y.pno = 2
      and x.scity = 'Seattle'
      and x.sstate = 'WA'
```



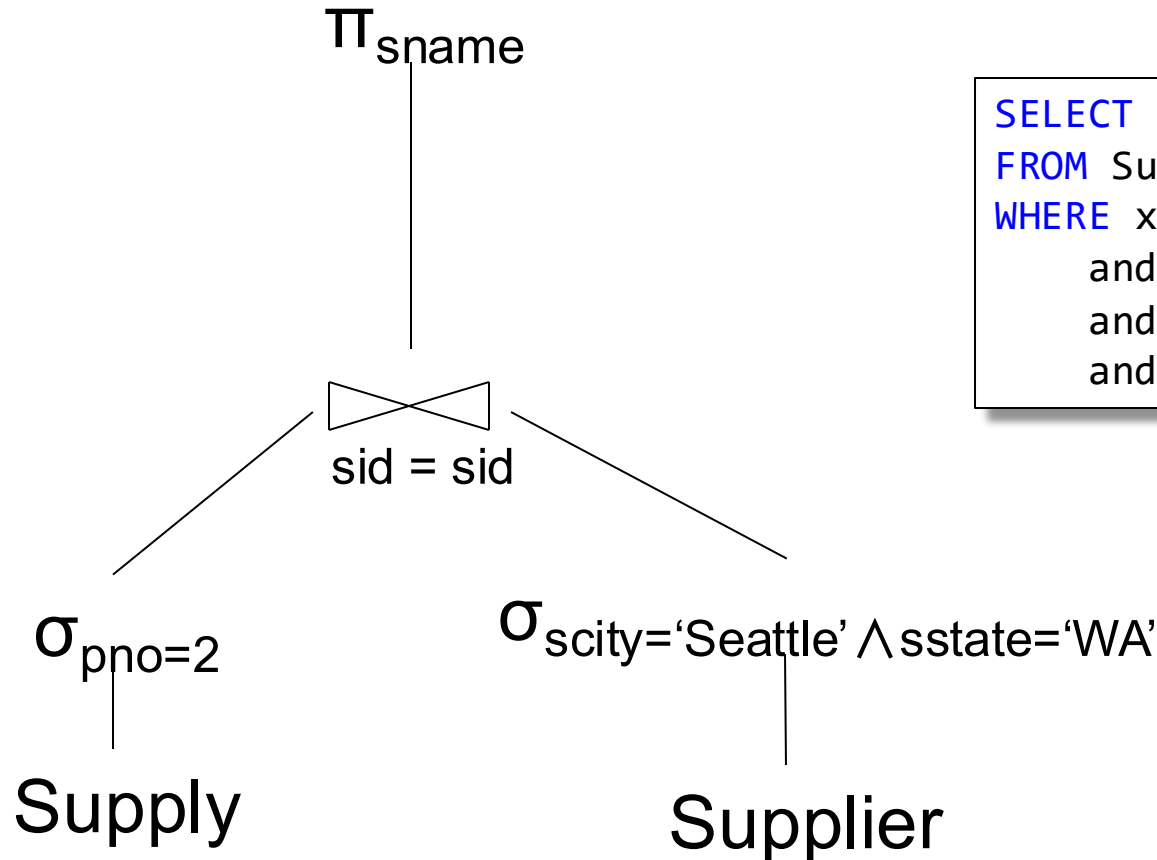
T(Supply) = 10000
B(Supply) = 100
V(Supply, pno) = 2500

T(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10

M=11

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Logical Query Plan 2



```
SELECT sname
FROM Supplier x, Supply y
WHERE x.sid = y.sid
      and y.pno = 2
      and x.scity = 'Seattle'
      and x.sstate = 'WA'
```

T(Supply) = 10000
B(Supply) = 100
V(Supply, pno) = 2500

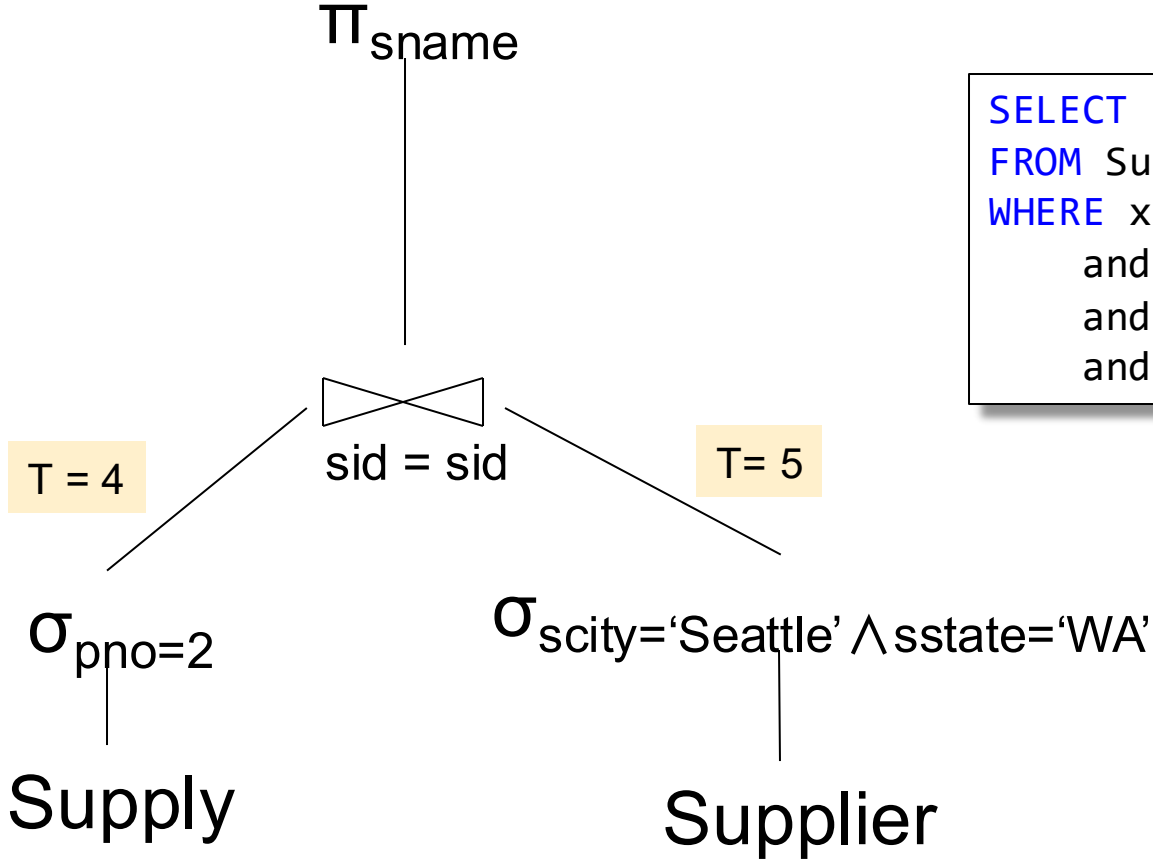
T(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10

M=11

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Logical Query Plan 2

```
SELECT sname
FROM Supplier x, Supply y
WHERE x.sid = y.sid
      and y.pno = 2
      and x.scity = 'Seattle'
      and x.sstate = 'WA'
```



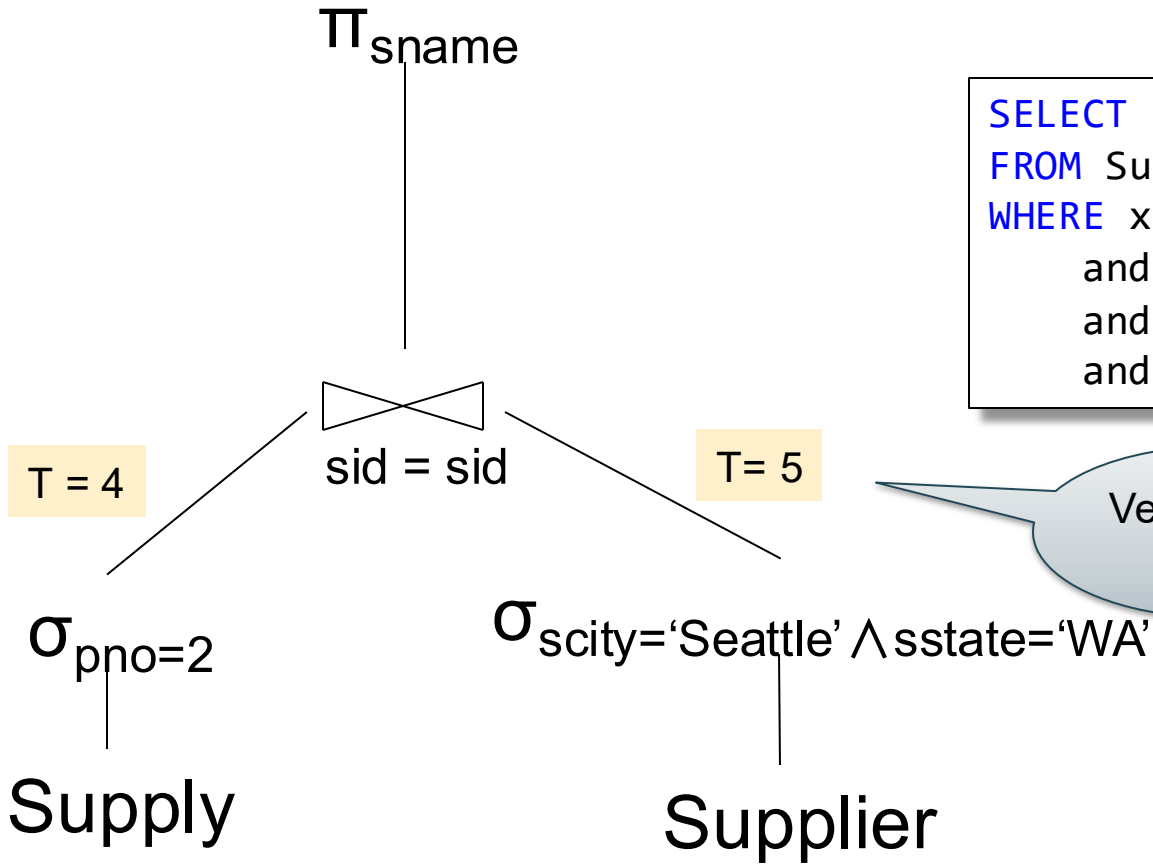
T(Supply) = 10000
B(Supply) = 100
V(Supply, pno) = 2500

T(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10

M=11

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Logical Query Plan 2



```
SELECT sname
FROM Supplier x, Supply y
WHERE x.sid = y.sid
      and y.pno = 2
      and x.scity = 'Seattle'
      and x.sstate = 'WA'
```

Very wrong!
Why?

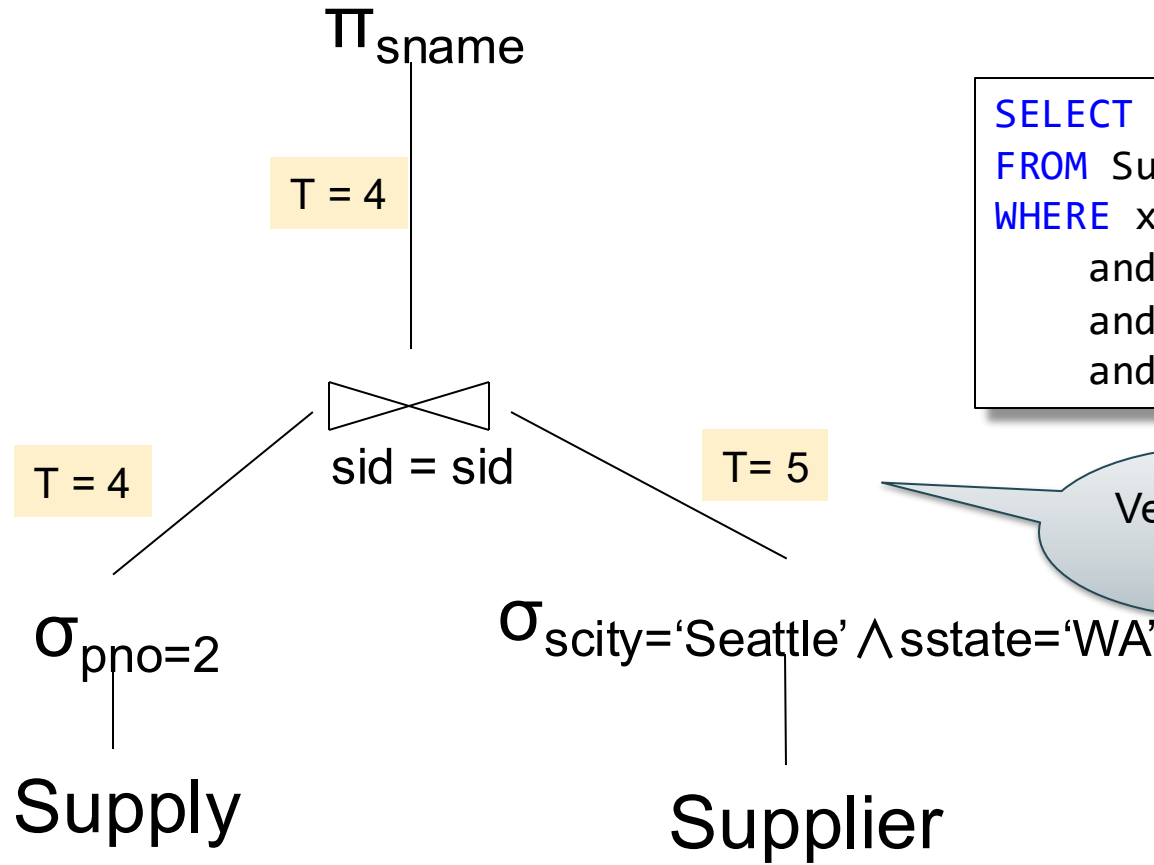
T(Supply) = 10000
B(Supply) = 100
V(Supply, pno) = 2500

T(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10

M=11

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Logical Query Plan 2



```
SELECT sname
FROM Supplier x, Supply y
WHERE x.sid = y.sid
      and y.pno = 2
      and x.scity = 'Seattle'
      and x.sstate = 'WA'
```

Very wrong!
Why?

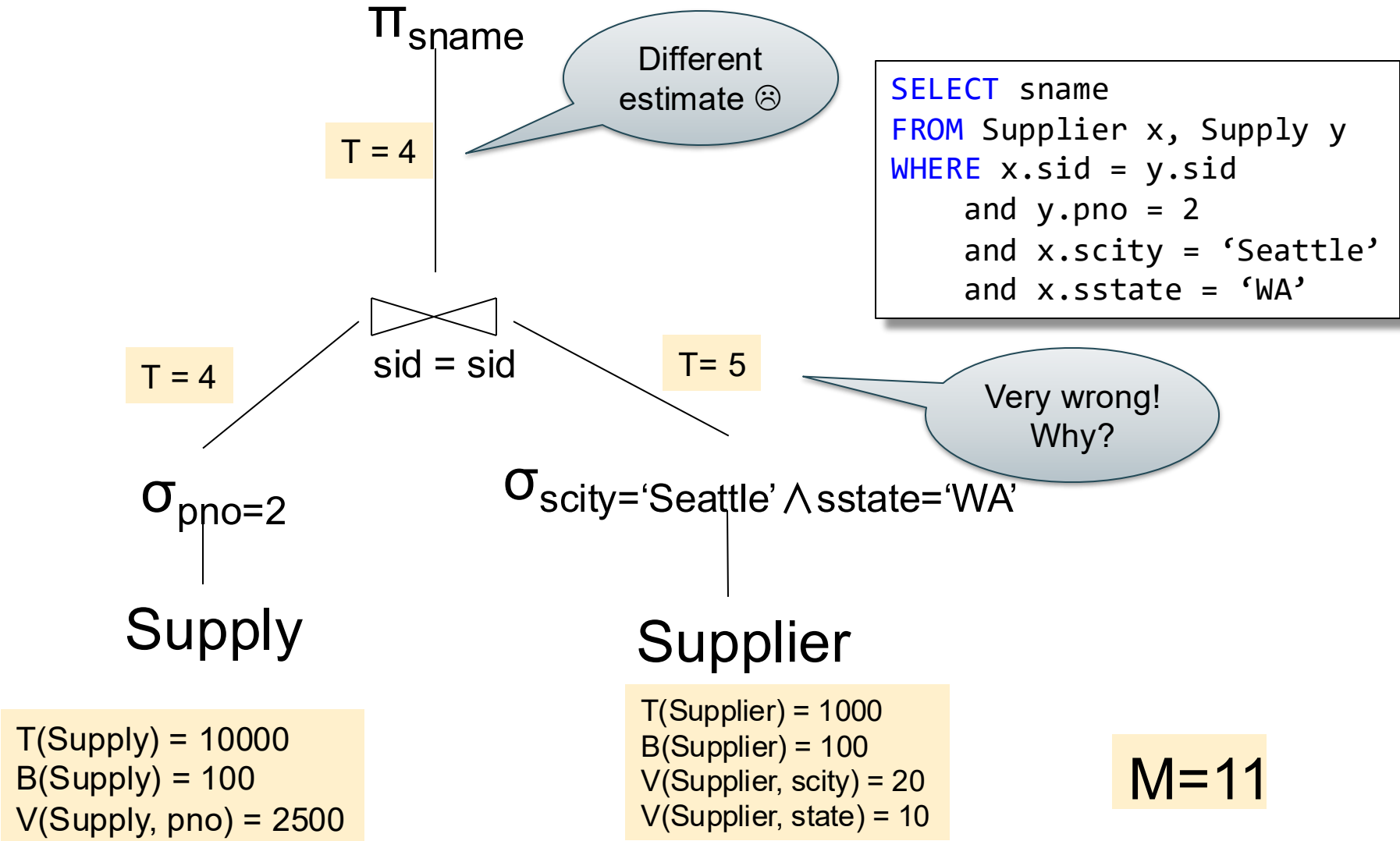
T(Supply) = 10000
B(Supply) = 100
V(Supply, pno) = 2500

T(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10

M=11

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Logical Query Plan 2



Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Physical Plan 1

π_{sname}

T < 1

$\sigma_{pno=2 \wedge scity='Seattle' \wedge sstate='WA'}$

T = 10000

Total cost:

$$B(\text{Supply}) + B(\text{Supply})B(\text{Supplier})/(M-1)$$



sid = sid

Block nested loop join

Scan

Supply

Scan

Supplier

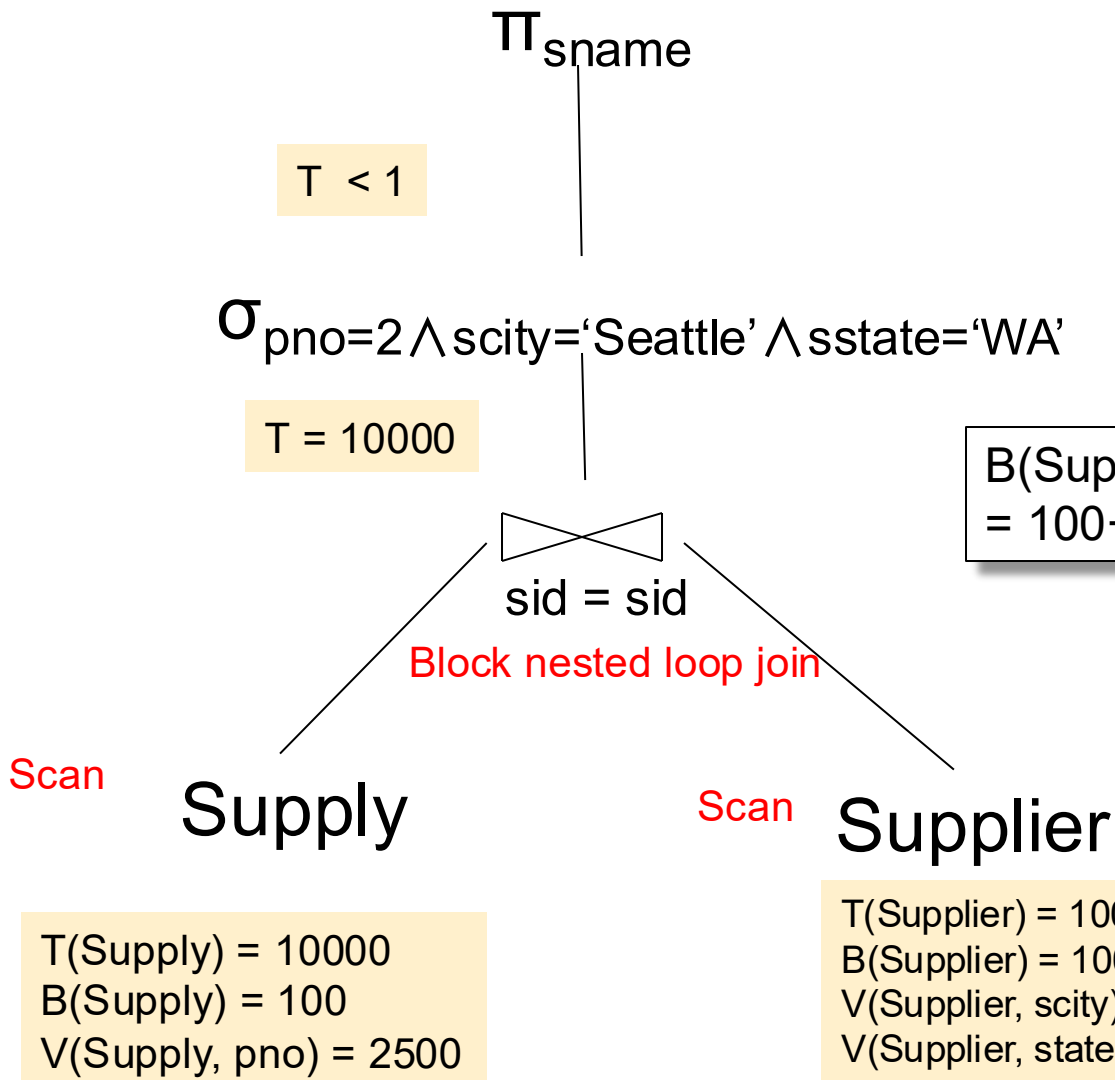
T(Supply) = 10000
B(Supply) = 100
V(Supply, pno) = 2500

T(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10

M=11

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Physical Plan 1



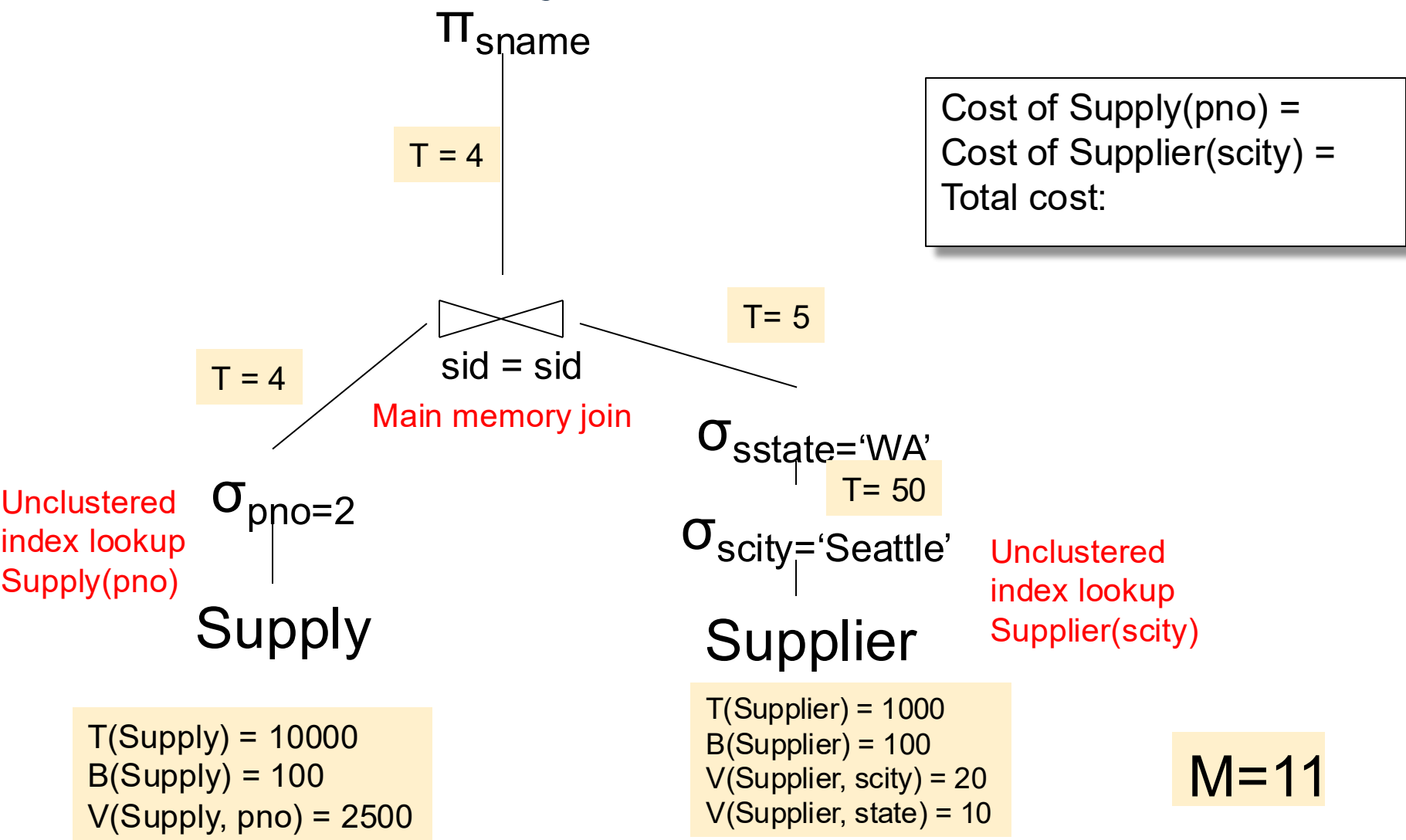
Total cost:

$$B(\text{Supply}) + B(\text{Supply})B(\text{Supplier})/(M-1) = 100 + 100 \cdot 100 / 10 = 1100$$

M=11

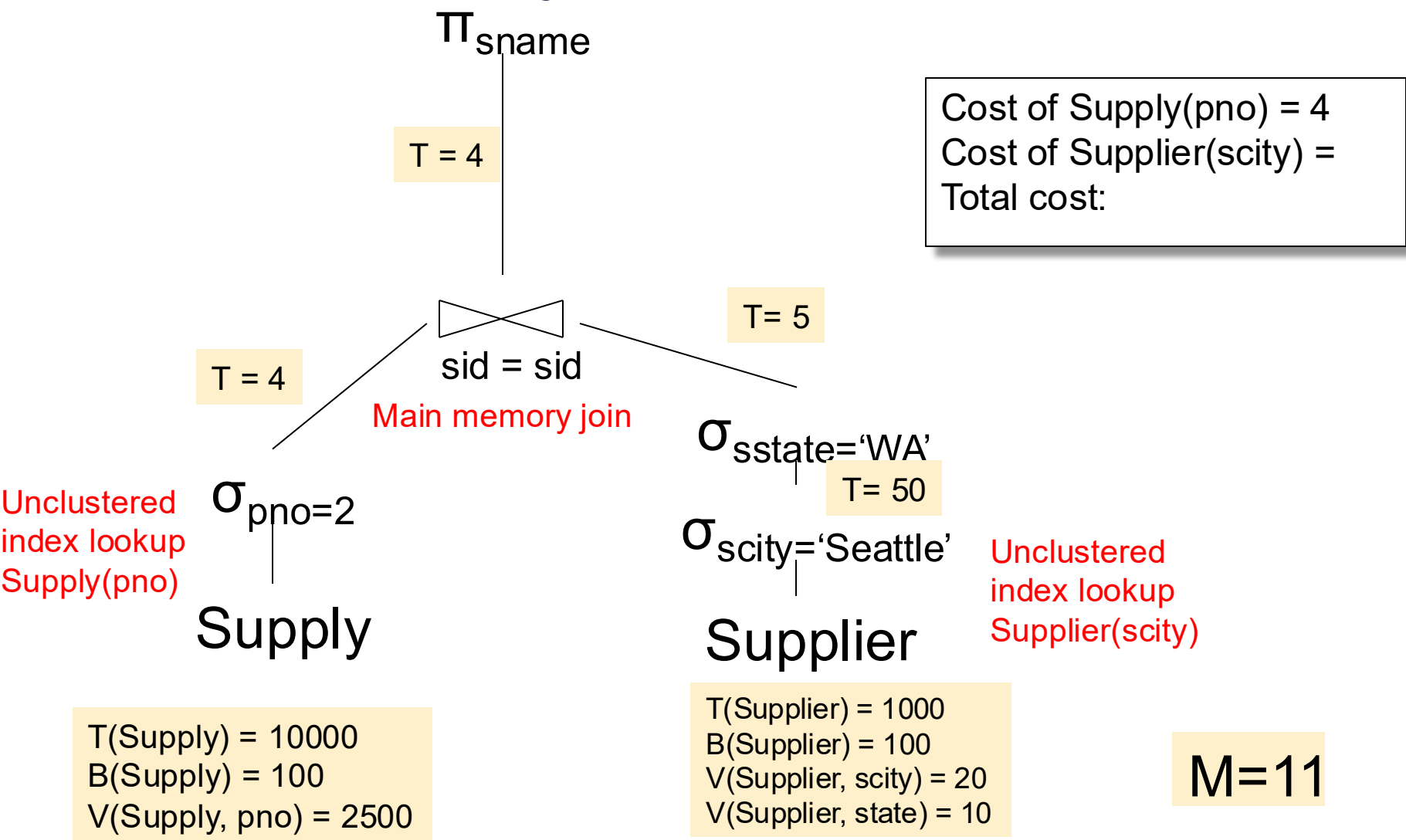
Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Physical Plan 2



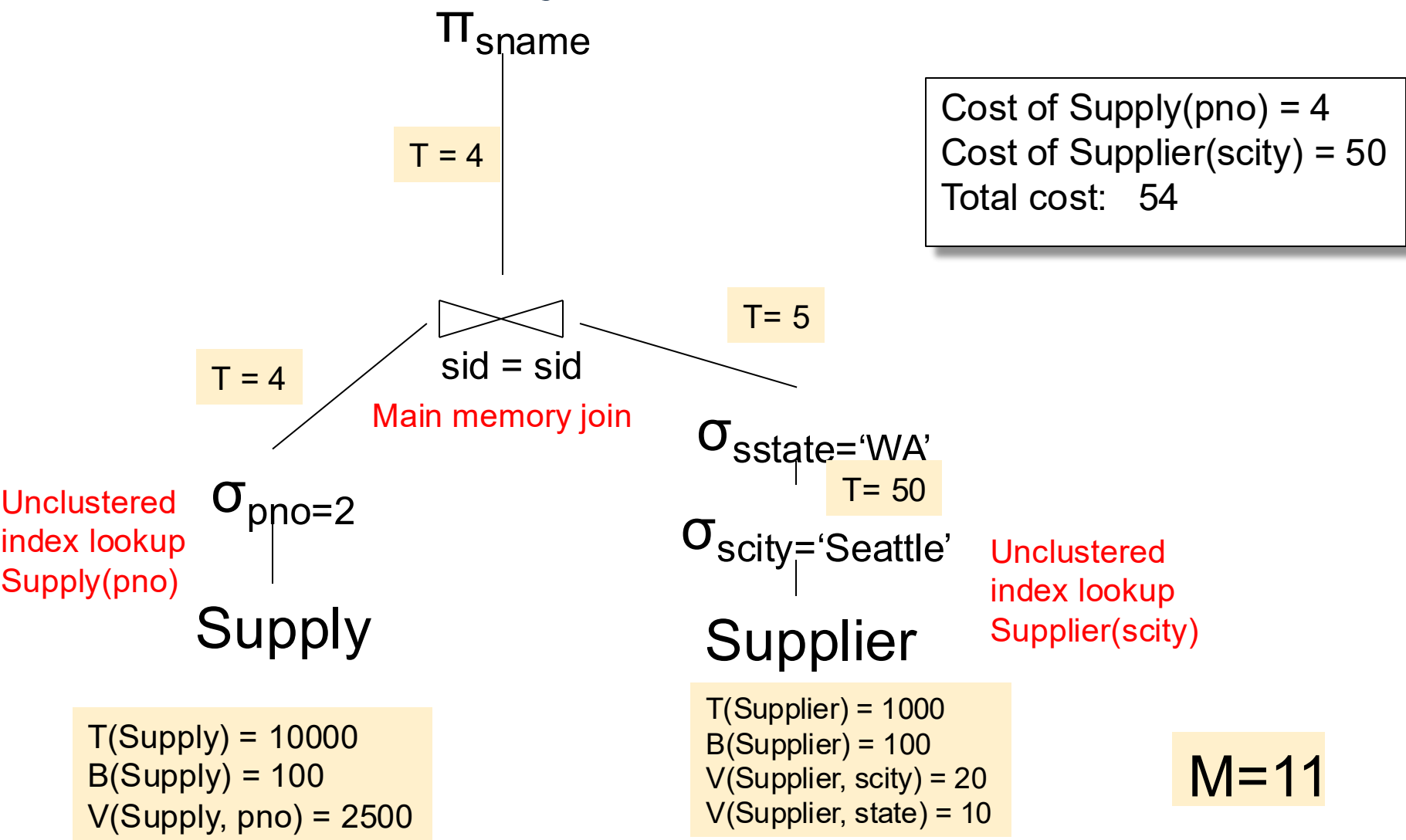
Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Physical Plan 2



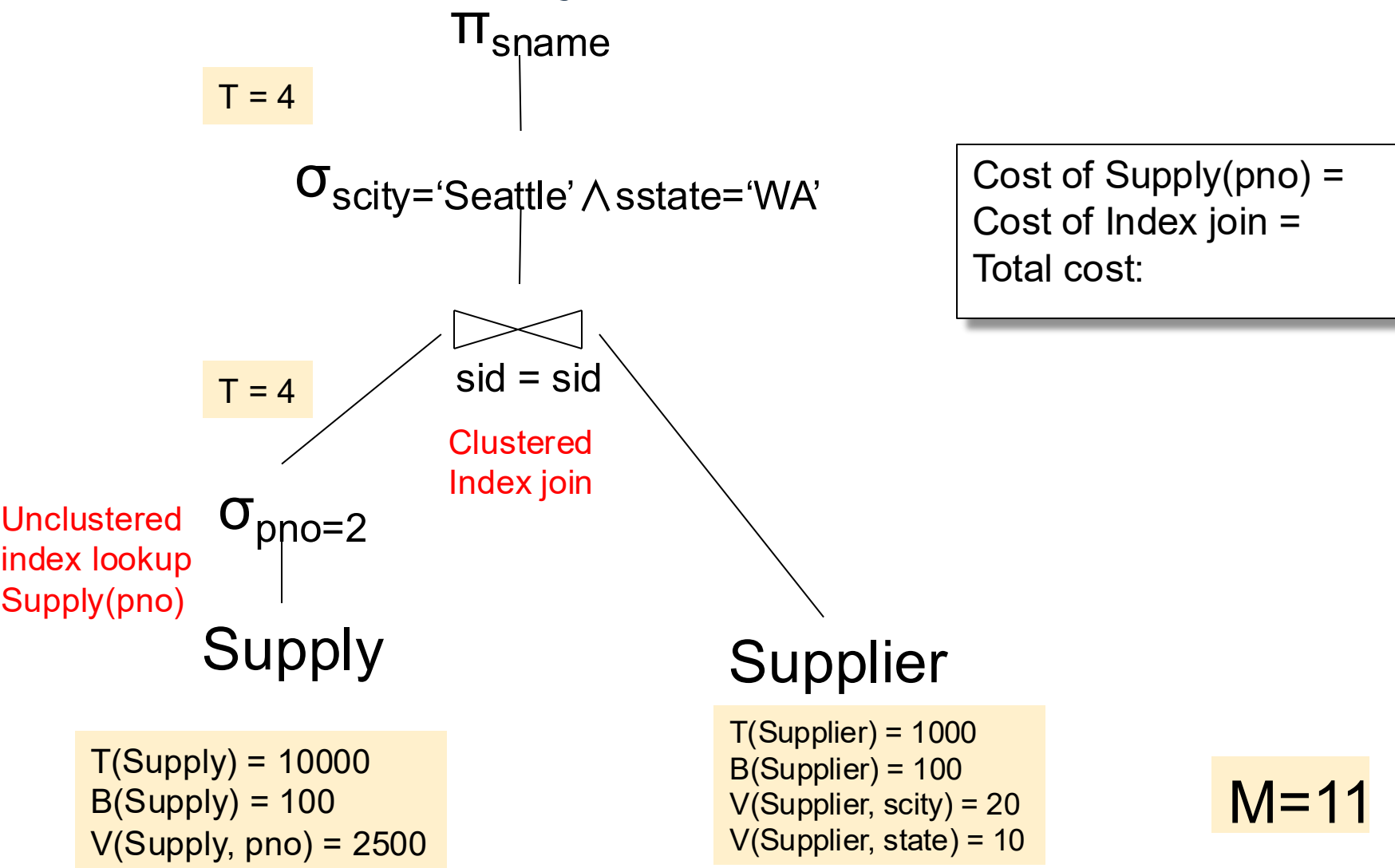
Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Physical Plan 2



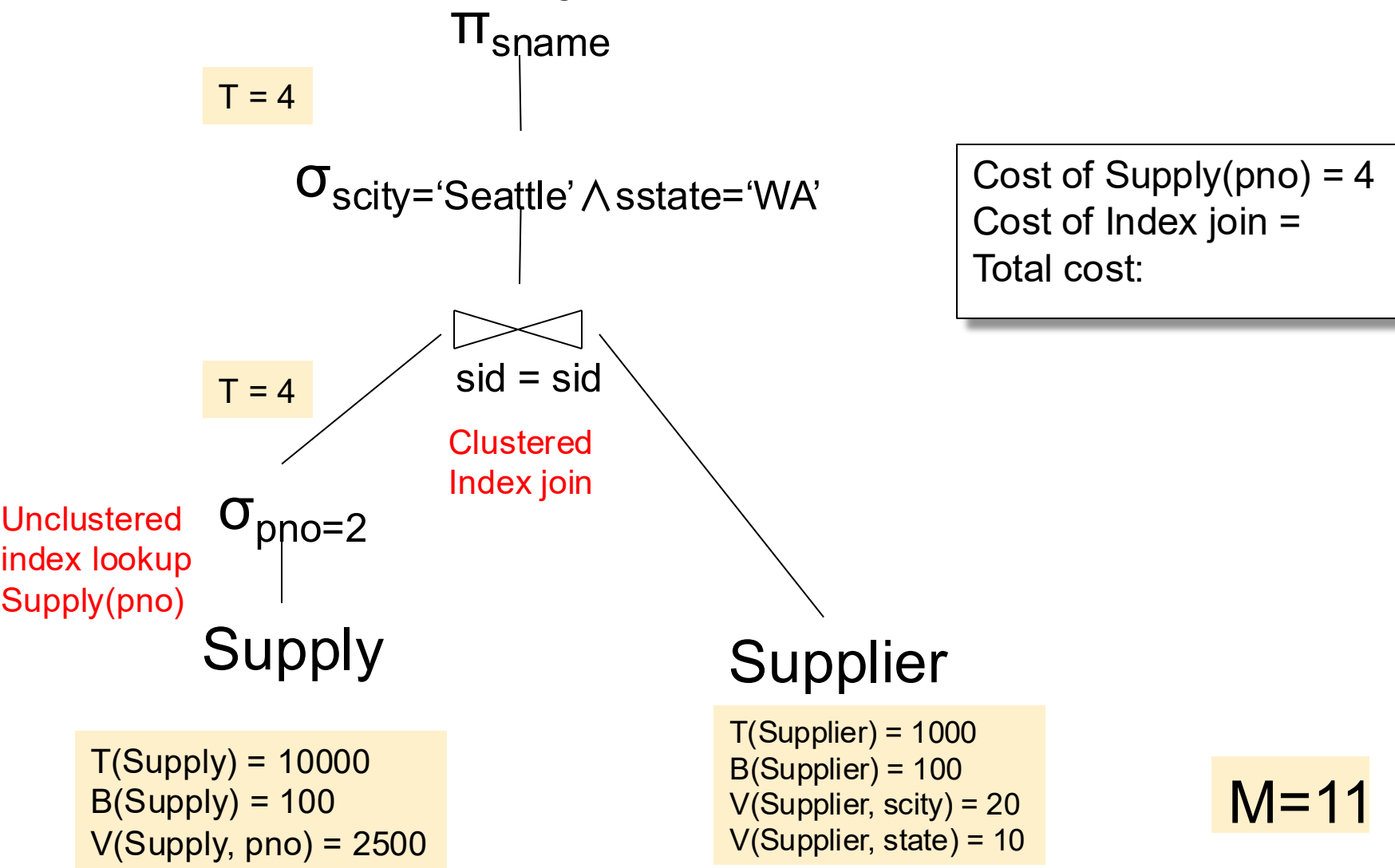
Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Physical Plan 3



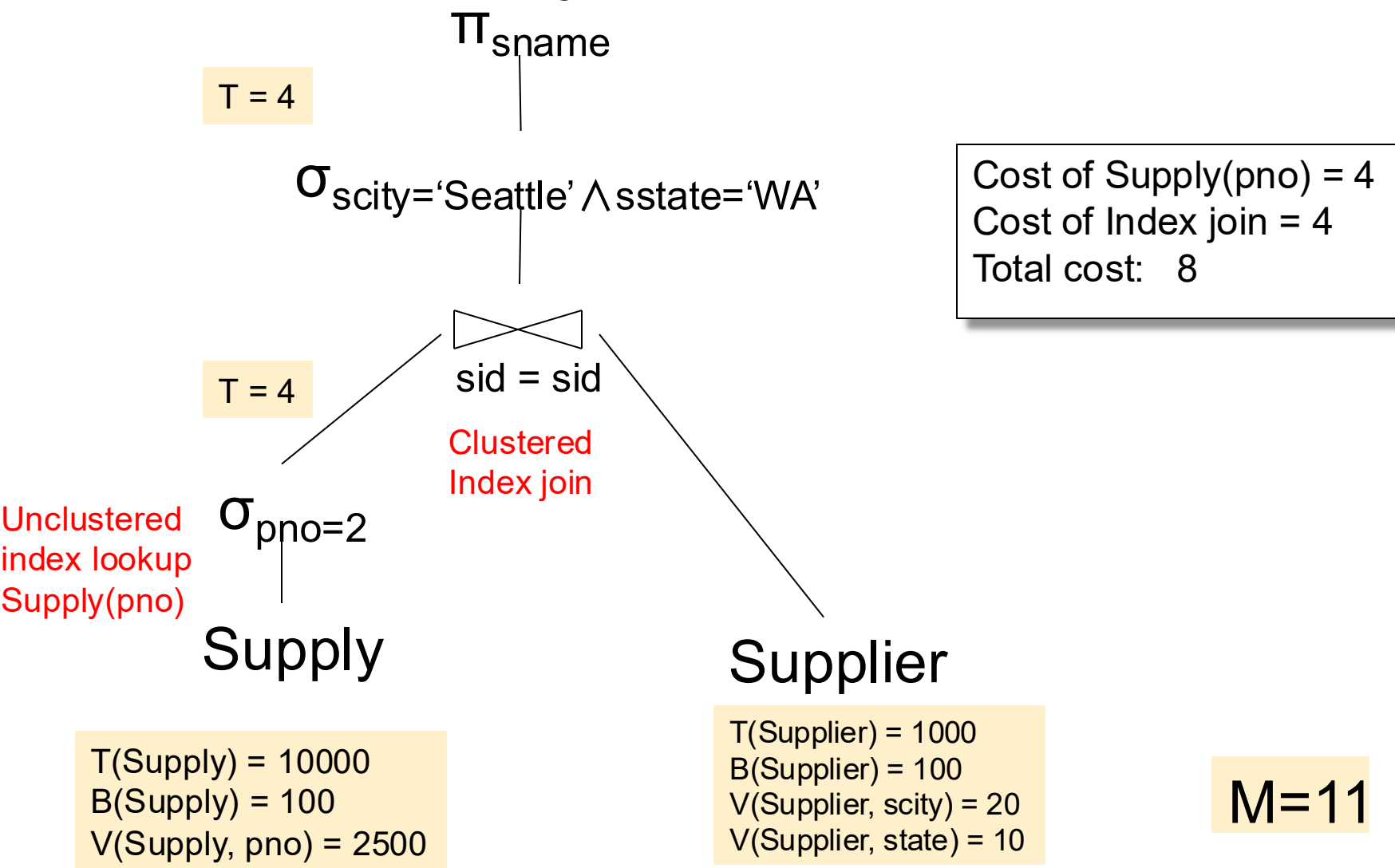
Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Physical Plan 3



Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

Physical Plan 3



Discussion

- We considered only IO cost; systems also need to estimate the CPU cost; SEE PAPER
- We assumed that all index pages were in memory: sometimes we need to add the cost of fetching index pages from disk

Histograms

- 1d Histograms mitigate uniformity assumption
- 2d Histograms mitigate independence assumption

1d-Histograms

- Histogram on $R.A$ refines $T(R)$, $V(R,A)$
- Each bucket: $T(\text{bucket})$, $V(\text{bucket}, A)$

1d-Histograms

Employee(ssn, name, age)

$T(\text{Employee}) = 25000$, $V(\text{Employee}, \text{age}) = 50$

$\sigma_{\text{age}=48}(\text{Employee}) = ?$

1d-Histograms

Employee(ssn, name, age)

$T(\text{Employee}) = 25000$, $V(\text{Employee}, \text{age}) = 50$

$\sigma_{\text{age}=48}(\text{Employee}) = ?$

Estimate: $T(\text{Employee}) / V(\text{Employee}, \text{age}) = 500$

1d-Histograms

Employee(ssn, name, age)

$T(\text{Employee}) = 25000$, $V(\text{Employee}, \text{age}) = 50$

$\sigma_{\text{age}=48}(\text{Employee}) = ?$

Estimate: $T(\text{Employee}) / V(\text{Employee}, \text{age}) = 500$

Age:	0..20	20..29	30-39	40-49	50-59	> 60
T =	200	800	5000	12000	6500	500

1d-Histograms

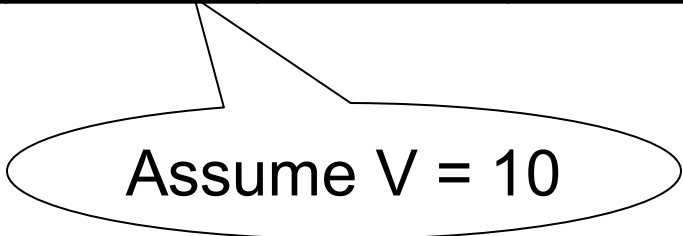
Employee(ssn, name, age)

$T(\text{Employee}) = 25000$, $V(\text{Employee}, \text{age}) = 50$

$\sigma_{\text{age}=48}(\text{Employee}) = ?$

Estimate: $T(\text{Employee}) / V(\text{Employee}, \text{age}) = 500$

Age:	0..20	20..29	30-39	40-49	50-59	> 60
T =	200	800	5000	12000	6500	500



Assume $V = 10$

1d-Histograms

Employee(ssn, name, age)

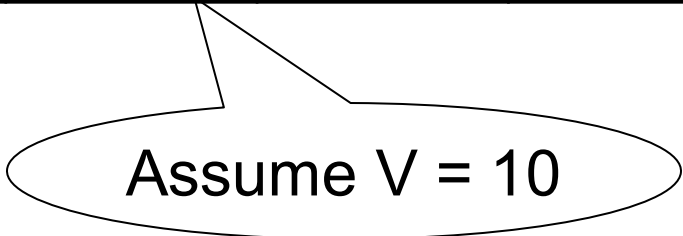
$T(\text{Employee}) = 25000$, $V(\text{Employee}, \text{age}) = 50$

$\sigma_{\text{age}=48}(\text{Employee}) = ?$

~~Estimate: $T(\text{Employee}) / V(\text{Employee}, \text{age}) = 500$~~

Age:	0..20	20..29	30-39	40-49	50-59	> 60
T =	200	800	5000	12000	6500	500

Estimate: $12000/10 = 1200$



Assume $V = 10$

1d-Histograms

Employee(ssn, name, age)

$T(\text{Employee}) = 25000$, $V(\text{Employee}, \text{age}) = 50$

$\sigma_{\text{age}=48}(\text{Employee}) = ?$

~~Estimate: $T(\text{Employee}) / V(\text{Employee}, \text{age}) = 500$~~

Age:	0..20	20..29	30-39	40-49	50-59	> 60
T =	200	800	5000	12000	6500	500
V =	3	10	7	6	5	4

Estimate: $12000/10 = 1200$

1d-Histograms

Employee(ssn, name, age)

$T(\text{Employee}) = 25000$, $V(\text{Employee}, \text{age}) = 50$

$\sigma_{\text{age}=48}(\text{Employee}) = ?$

~~Estimate: $T(\text{Employee}) / V(\text{Employee}, \text{age}) = 500$~~

Age:	0..20	20..29	30-39	40-49	50-59	> 60
T =	200	800	5000	12000	6500	500
V =	3	10	7	6	5	4

Estimate: ~~$12000/10 = 1200$~~ $12000/6 = 2000$

Types of 1d-Histograms

- Eq-Width
- Eq-Depth
- Compressed: store outliers separately
- V-Optimal histograms

Employee(ssn, name, age)

Types of 1d-Histograms

Eq-width:

Age:	0..20	20..29	30-39	40-49	50-59	> 60
Tuples	200	800	5000	12000	6500	500

Eq-depth:

Age:	0..32	33..41	42-46	47-52	53-58	> 60
Tuples	1800	2000	2100	2200	1900	1800

V-Optimal: next

V-Optimal Histogram

- $\text{domain}(A) = \{v_1, \dots, v_n\}$

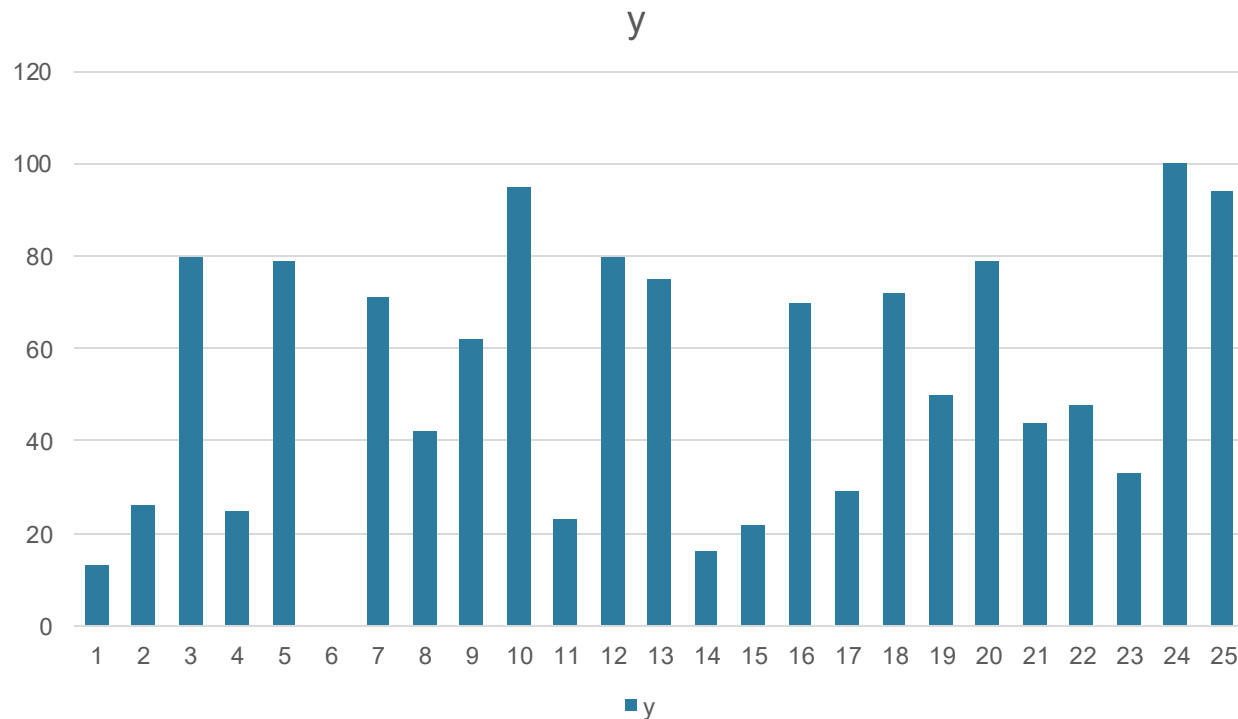
V-Optimal Histogram

- $\text{domain}(A) = \{v_1, \dots, v_n\}$
- Have budget of $b+1$ buckets for R.A:
$$-\infty < d_1 < d_2 < \dots < d_b < \infty$$

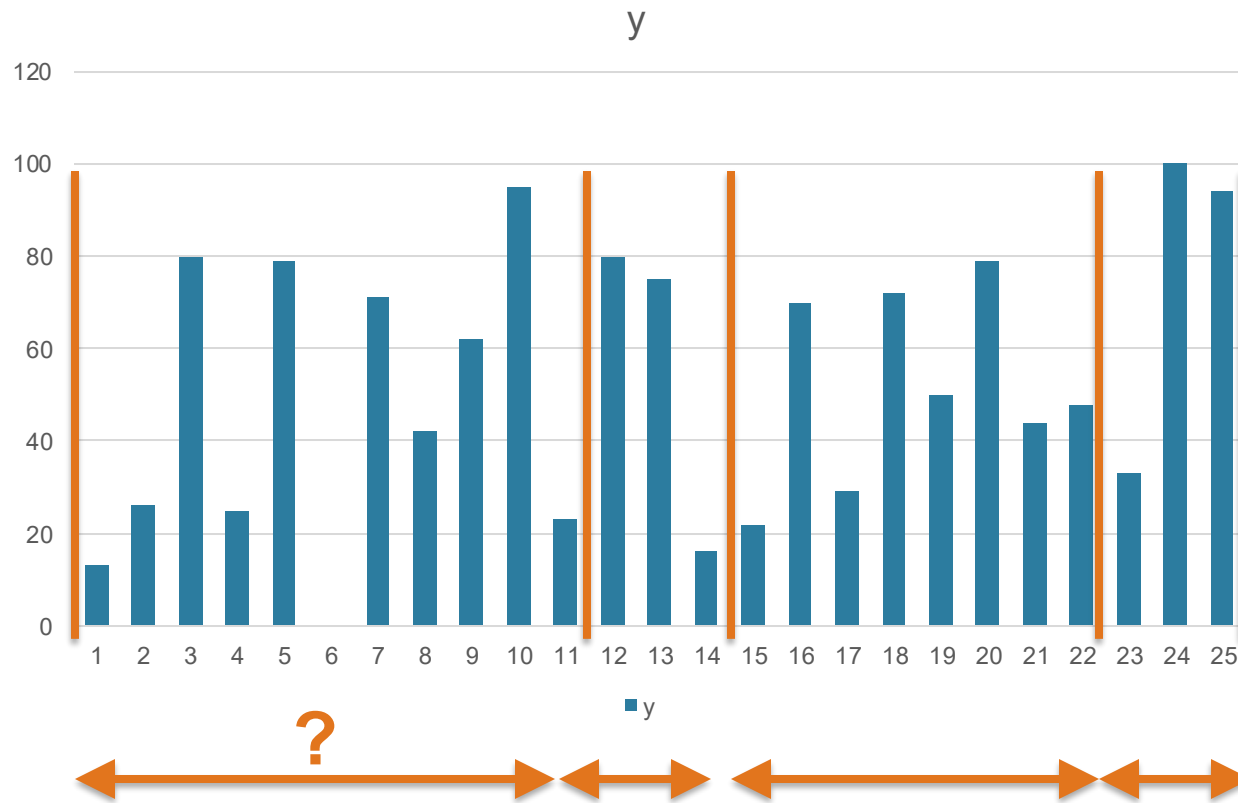
V-Optimal Histogram

- $\text{domain}(A) = \{v_1, \dots, v_n\}$
- Have budget of $b+1$ buckets for R.A:
$$-\infty < d_1 < d_2 < \dots < d_b < \infty$$
- Choose $d_1, \dots, d_b$ to minimize error of all queries $\sigma_{A=v}(R)$, for $v \in \text{Domain}(A)$

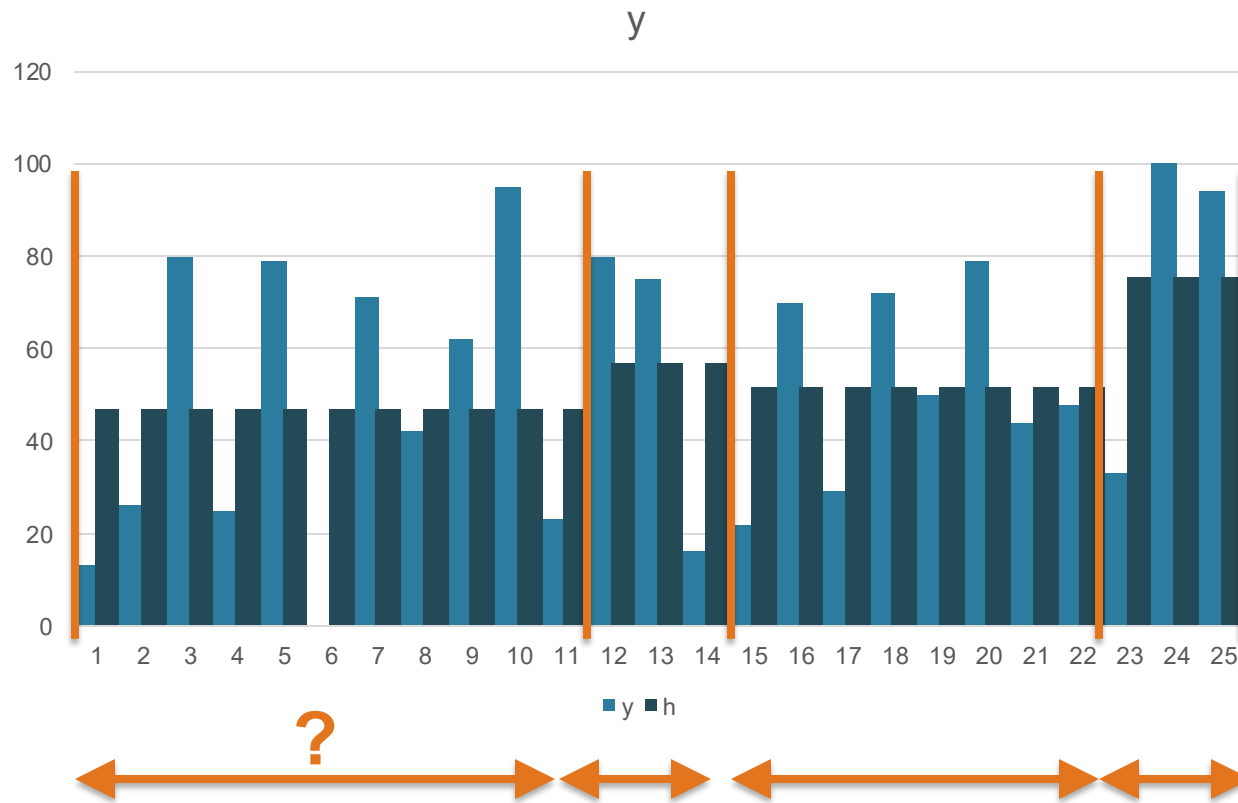
V-Optimal Histogram



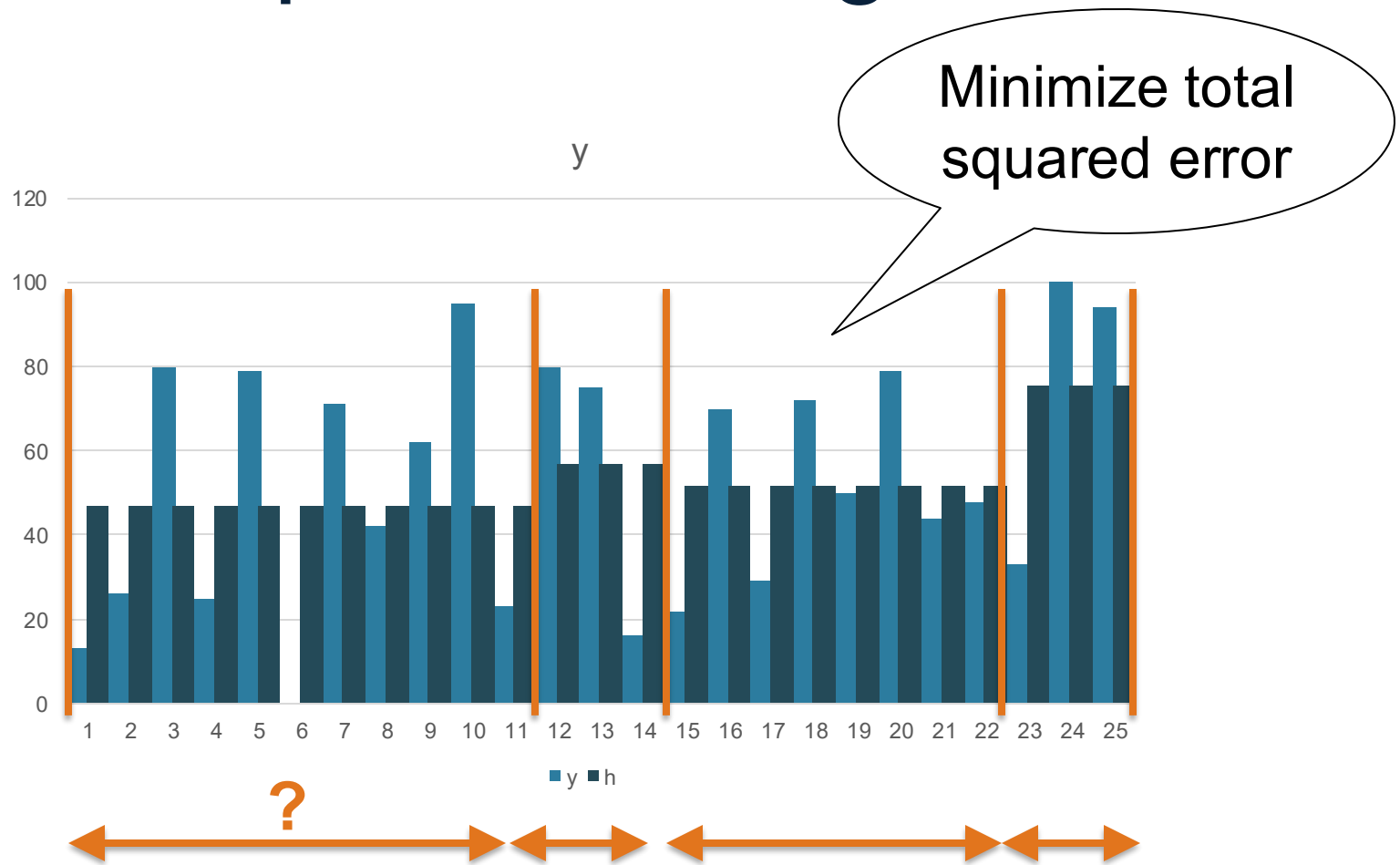
V-Optimal Histogram



V-Optimal Histogram



V-Optimal Histogram



V-Optimal Histogram

- Error:

$$\sum_{v \in \text{Domain}(A)} \left(|\sigma_{A=v}(R)| - \text{est}_{\text{Hist}}(\sigma_{A=v}(R)) \right)^2$$

- Bucket boundaries = $\text{argmin}_{\text{Hist}}(\text{Error})$
- Dynamic programming

Discussion: 1d-Histograms

- All systems support some forms
- Most Common Values (MCV): separate
- Histograms need to be small to reside in main memory: 1000 – 10000 buckets
- Recomputed periodically, often from samples. E.g Postgres ANALYSE

2d-Histograms

```
SELECT *  
FROM Supplier  
WHERE scity = 'Seattle'  
and sstate = 'WA'
```

	AL-AR	CA-FL	...	TX-WA	WV-WY
Ab..Co					
...					
Sa...Tu					

Independence only
needed within the bucket

T(bucket),
V(bucket,scity)
V(bucket,sstate)

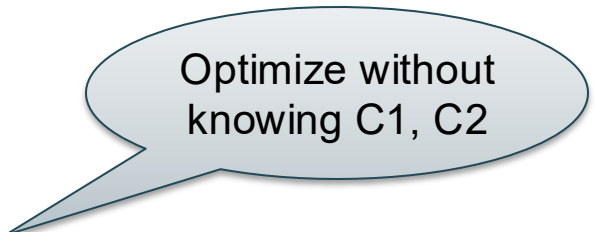
Issues with 2d-Histograms

- Limited # of dividers: $\sqrt{1000} - \sqrt{10000}$
- Too many 2d-histograms: $n(n-1)/2$
- Problem: how do we estimate $P(A,B,C)$ given all 1d and 2d-histograms?
- Few systems support 2d-histograms

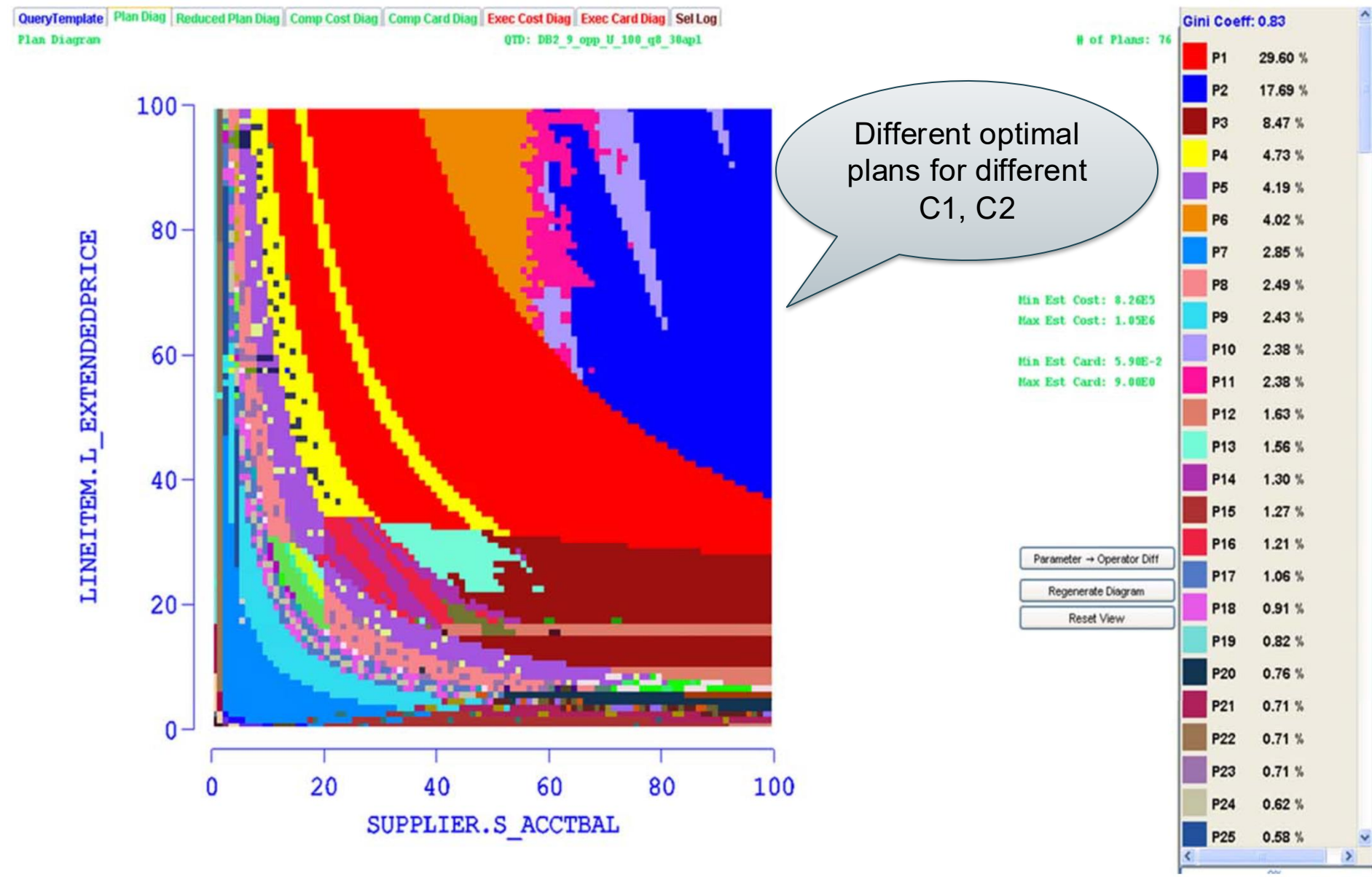
Yet Another Difficulty

- SQL Queries issued from applications
- Optimized once using **prepare** statement, executed repeatedly
- Constants in the query are not known at optimization time

```
select
  o_year, sum(case when nation = 'BRAZIL' then volume else 0 end) / sum(volume)
from
  (select YEAR(o_orderdate) as o_year,
    l_extendedprice * (1 - l_discount) as volume,
    n2.n_name as nation
  from part, supplier, lineitem, orders,
    customer, nation n1, nation n2, region
  where p_partkey = l_partkey and s_suppkey = l_suppkey
    and l_orderkey = o_orderkey and o_custkey = c_custkey
    and c_nationkey = n1.n_nationkey
    and n1.n_regionkey = r_regionkey
    and r_name = 'AMERICA'
    and s_nationkey = n2.n_nationkey
    and o_orderdate between '1995-01-01'
    and '1996-12-31'
    and p_type = 'ECONOMY ANODIZED STEEL'
    and s_acctbal ≤ C1 and l_extendedprice ≤ C2 ) as all_nations
group by o_year order by o_year
```



Optimize without
knowing C1, C2



Traditional CE -- Summary

- Used by all DBMS
- Assumptions: uniformity, independence,...
- 1d-histograms; sometimes also 2d
- Often underestimate (why?)

Alternate CE Methods: ML, Sampling

Upper bounds: next week

ML-Based CE

- A trend in research: use some ML model to do cardinality estimation
- Main hope: remove independence and uniformity assumptions
- Data-driven or Query-driven

Data-Driven Estimators

- Train a generative ML model that represents $p(A,B,C,\dots)$ from the database instance
- Map query Q to hyperrectangle
- Problem: space is over all attributes
 - “Full outer join”
 - Lots of complications to make this work

Query-Driven Estimators

- Training set is a set of pairs $(Q, |Q|)$
- All predicates in the WHERE condition need to be featurized, embedded
- Train a discriminative model for $\text{Est}(Q)$

Discussion

Problems with ML-based CE:

- Large model size: 1MB – 1GB
- Inference time varies
- Updates, change in skewness, correlations
- Not explainable

Consensus: not ready for production

Sampling-Based CE

- Main idea: use a sample of the database to estimate the output size
- Warning: different probability space!
 - Sampler v.s. data distribution
- Offline Sampling or Online Sampling

Offline Sampling

- Compute uniform sample $R_{sample} \subseteq R$
- Horwitz-Thompson:

$$Est(Q(DB)) = \frac{|R|}{|R_{sample}|} \cdot \frac{|S|}{|S_{sample}|} \cdots |Q(DB_{sample})|$$

- The missing tuple problem: $Q(DB_{sample}) = 0$
- High variance

Online Sampling

- To estimate $R \bowtie S$:
 - Sample a tuple from R
 - Sample a **matching tuple** from S
 - Repeat
- Need to use index on S
- **WanderJoin**: next slides show how to estimate $R \bowtie S \bowtie T$ using random walk

R

A	B
...	1
...	2
...	1
...	3
...	4
...	2
...	4

R

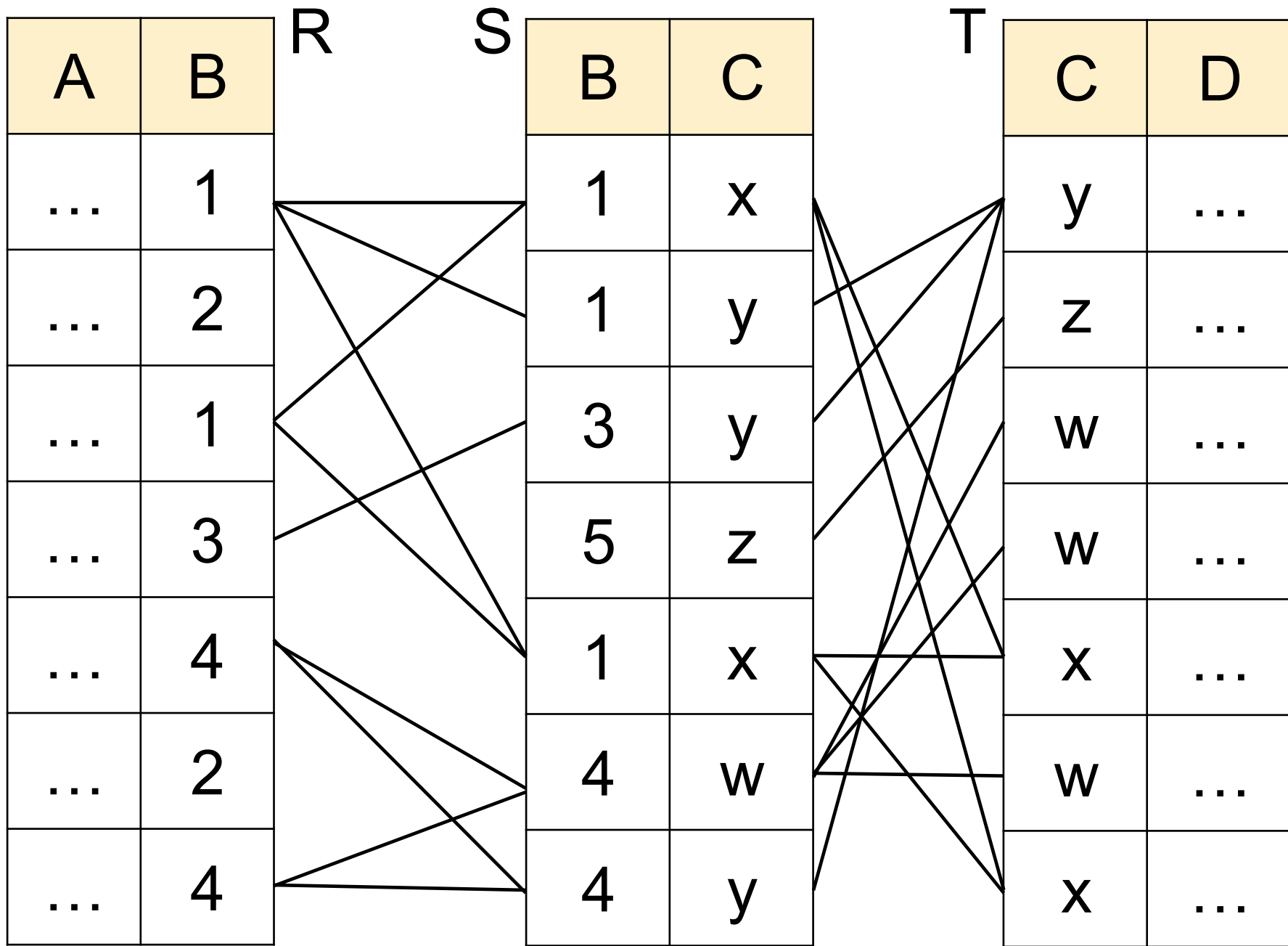
A	B
...	1
...	2
...	1
...	3
...	4
...	2
...	4

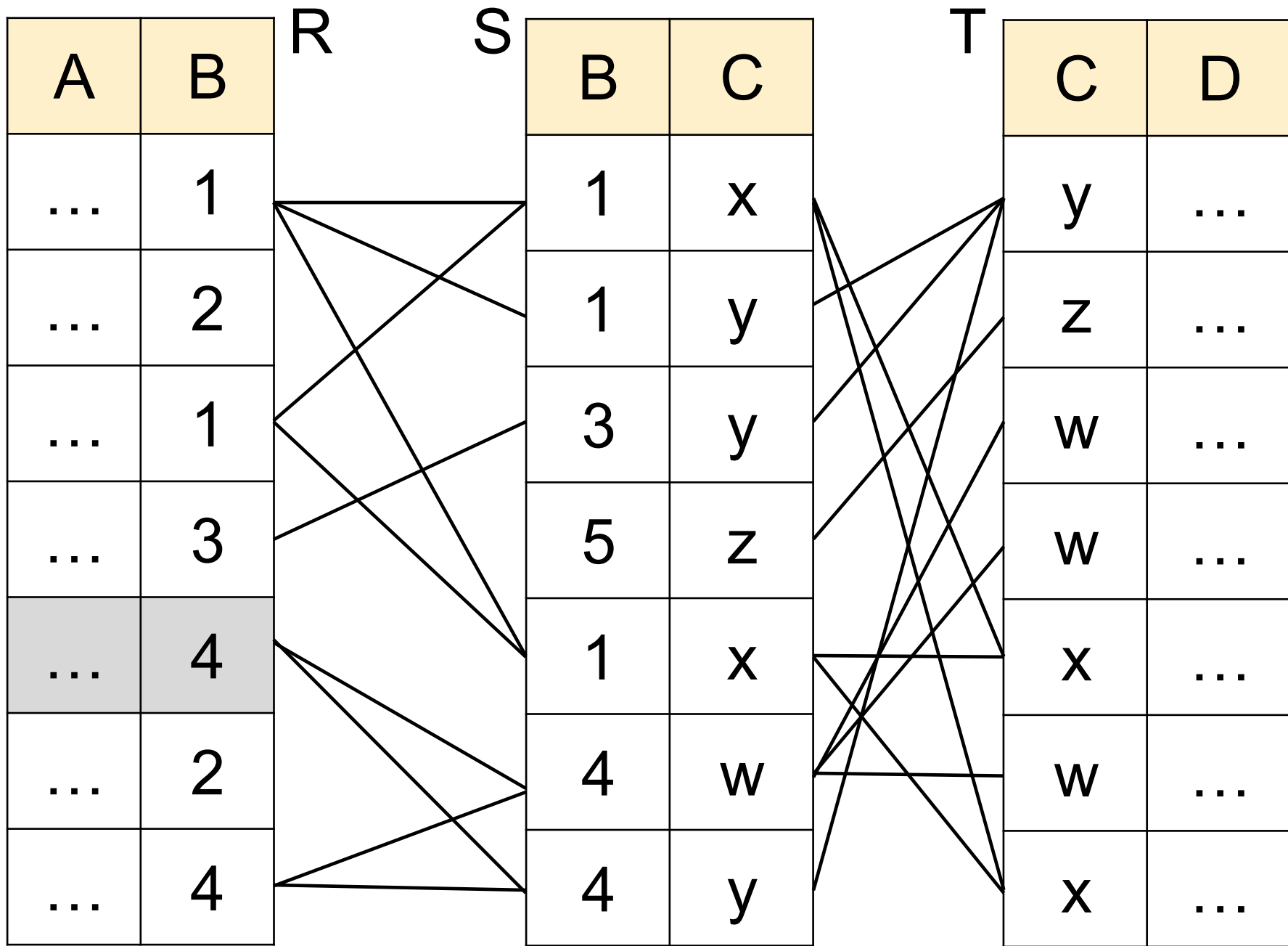
S

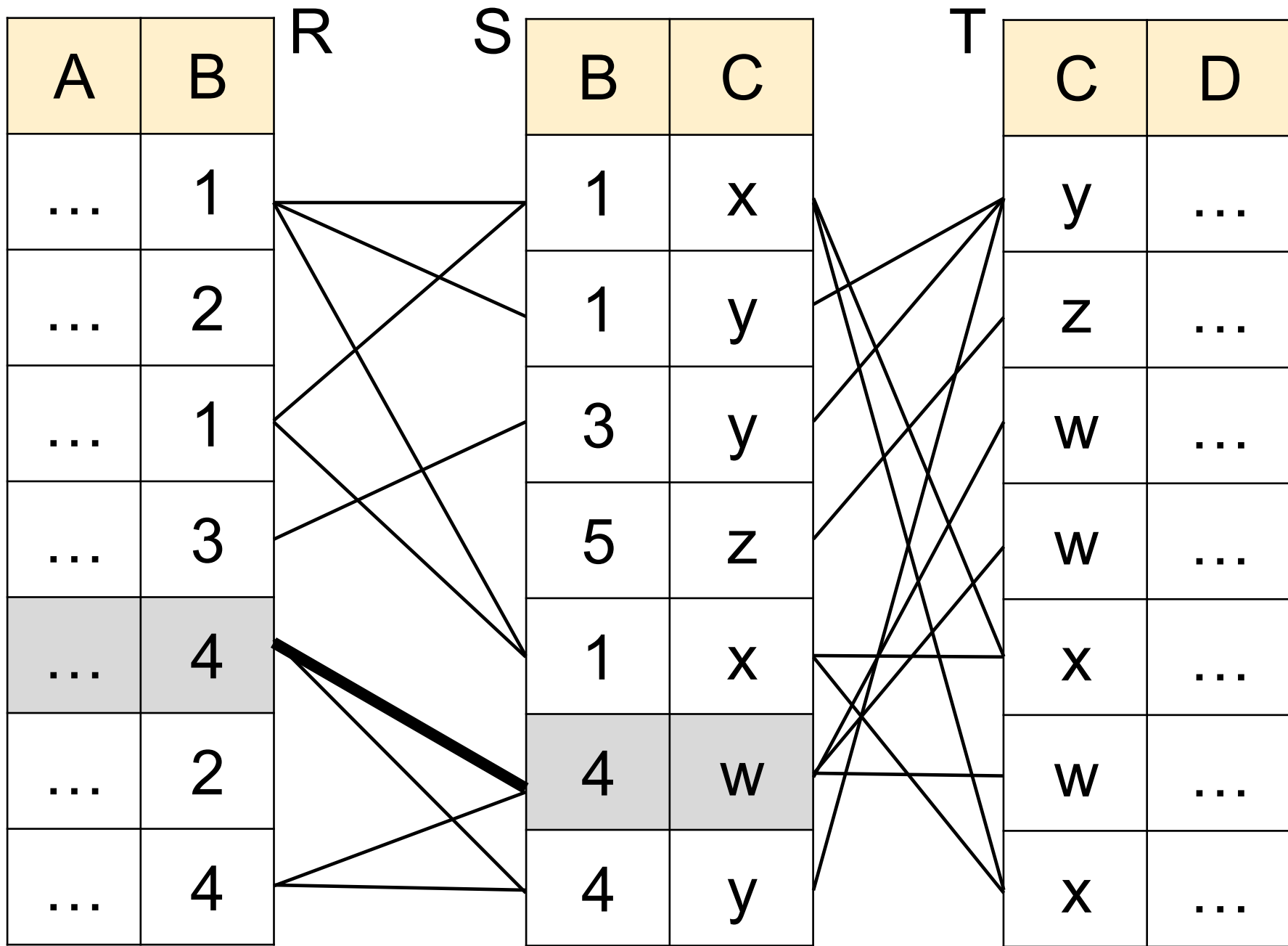
B	C
1	x
1	y
3	y
5	z
1	x
4	w
4	y

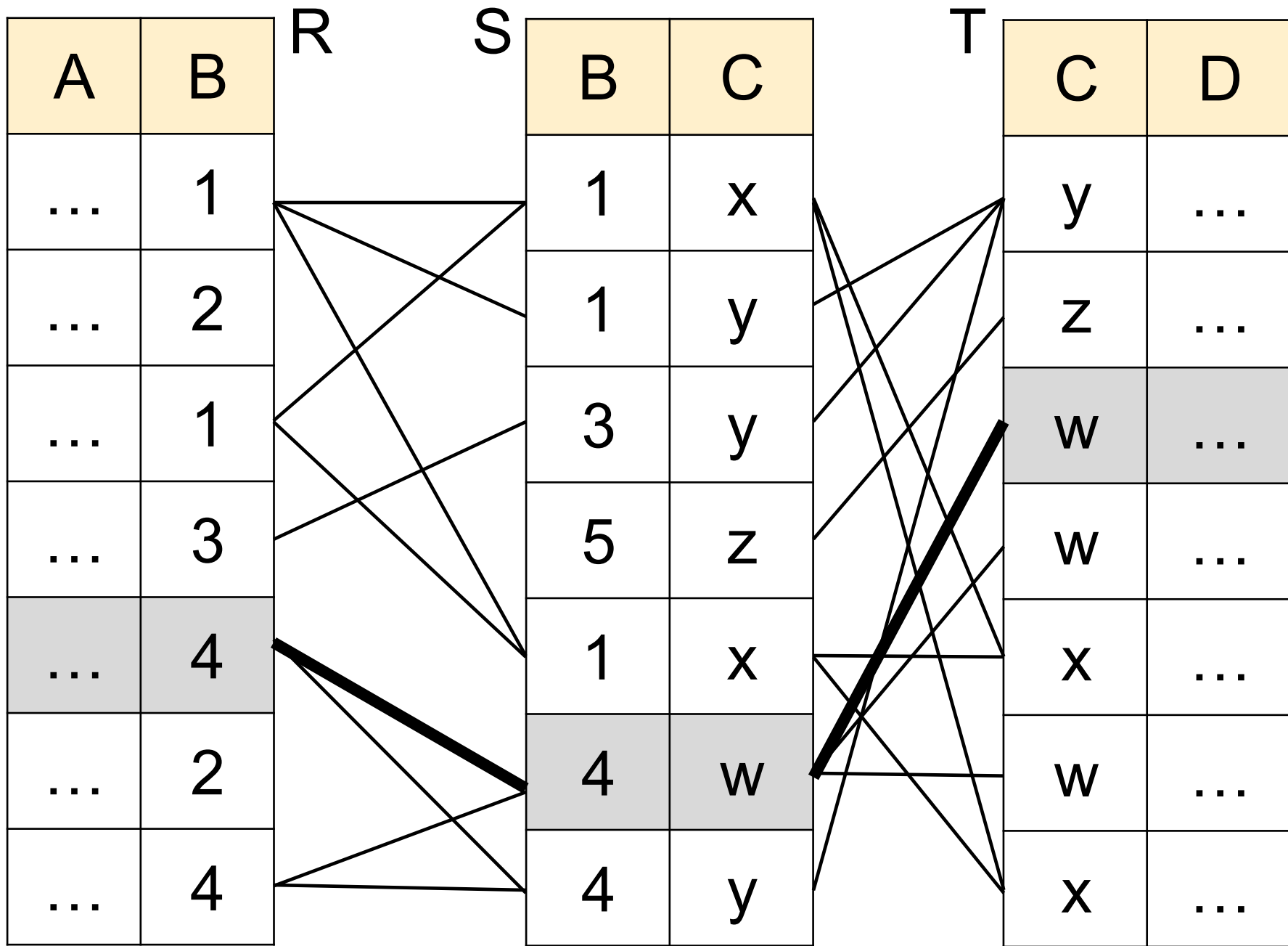
T

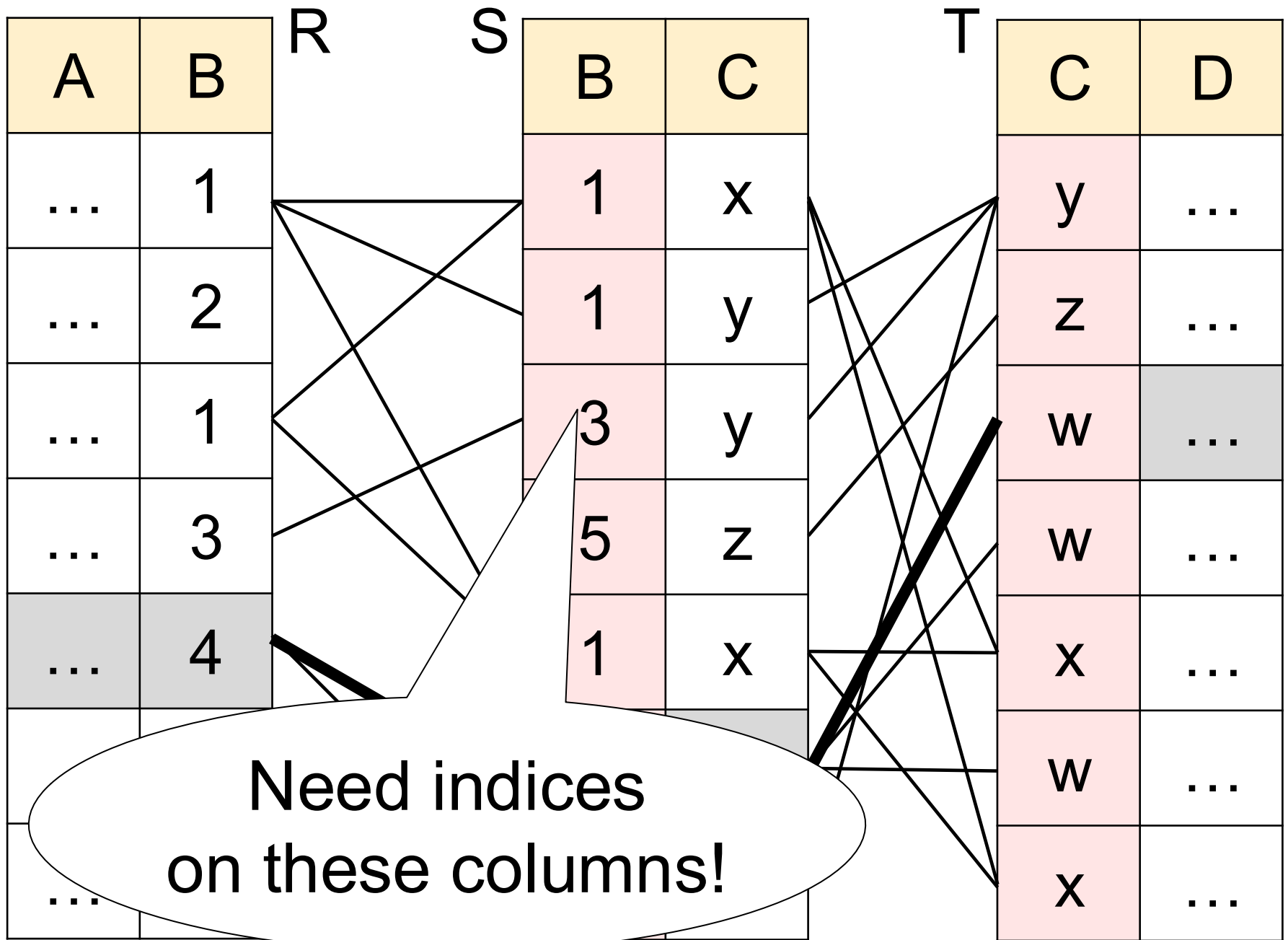
C	D
y	...
z	...
w	...
w	...
x	...
w	...
x	...

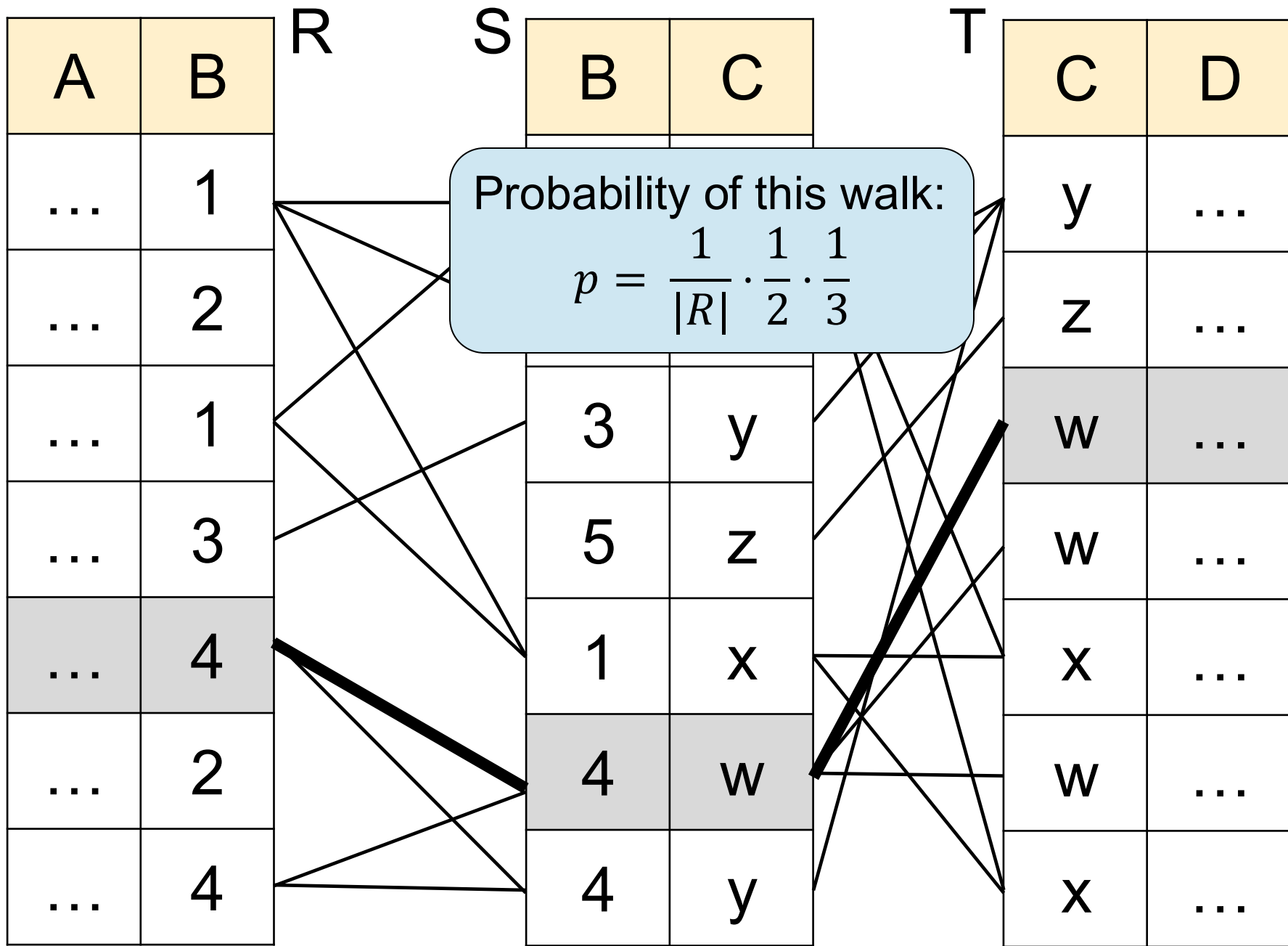


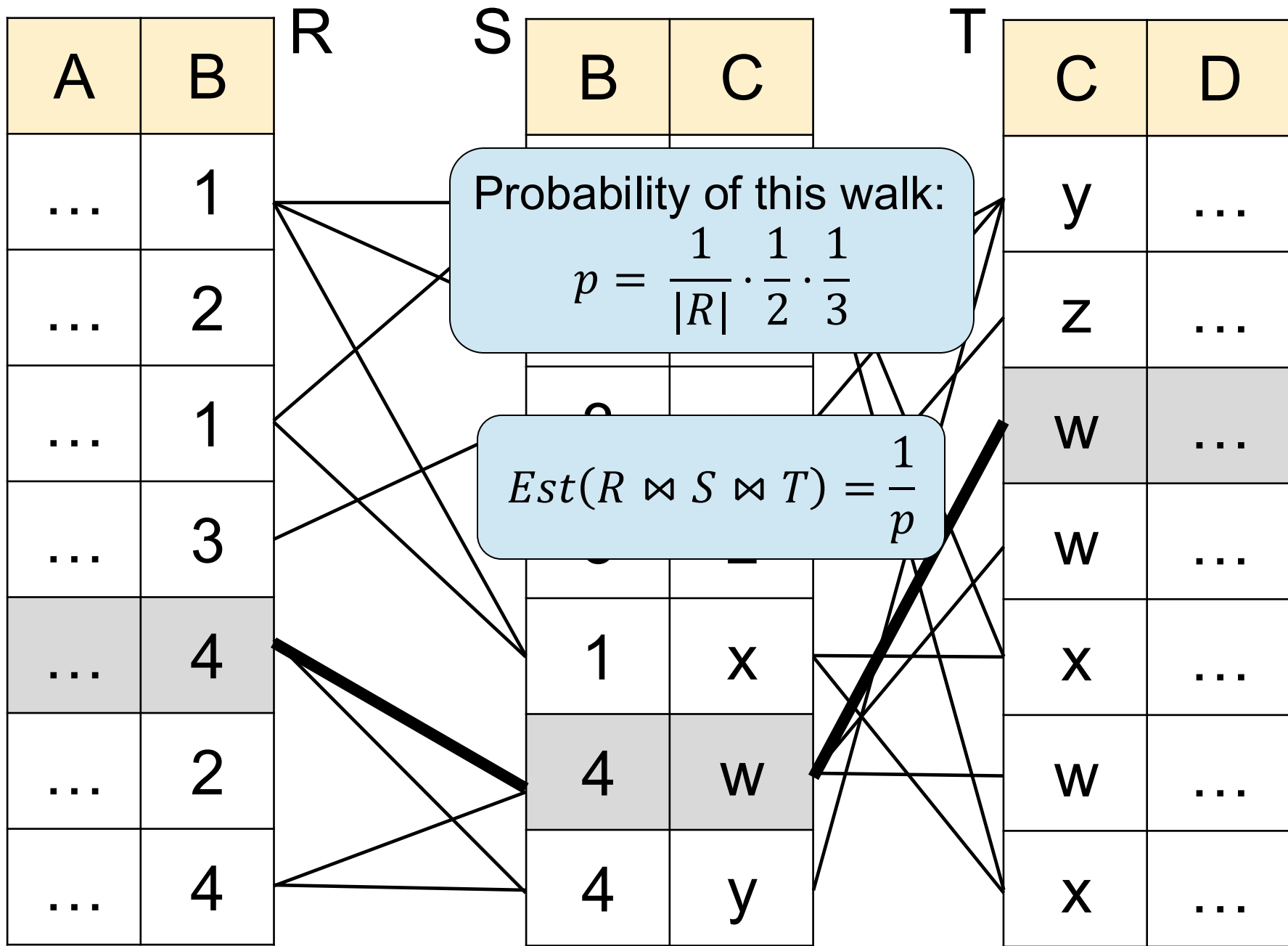


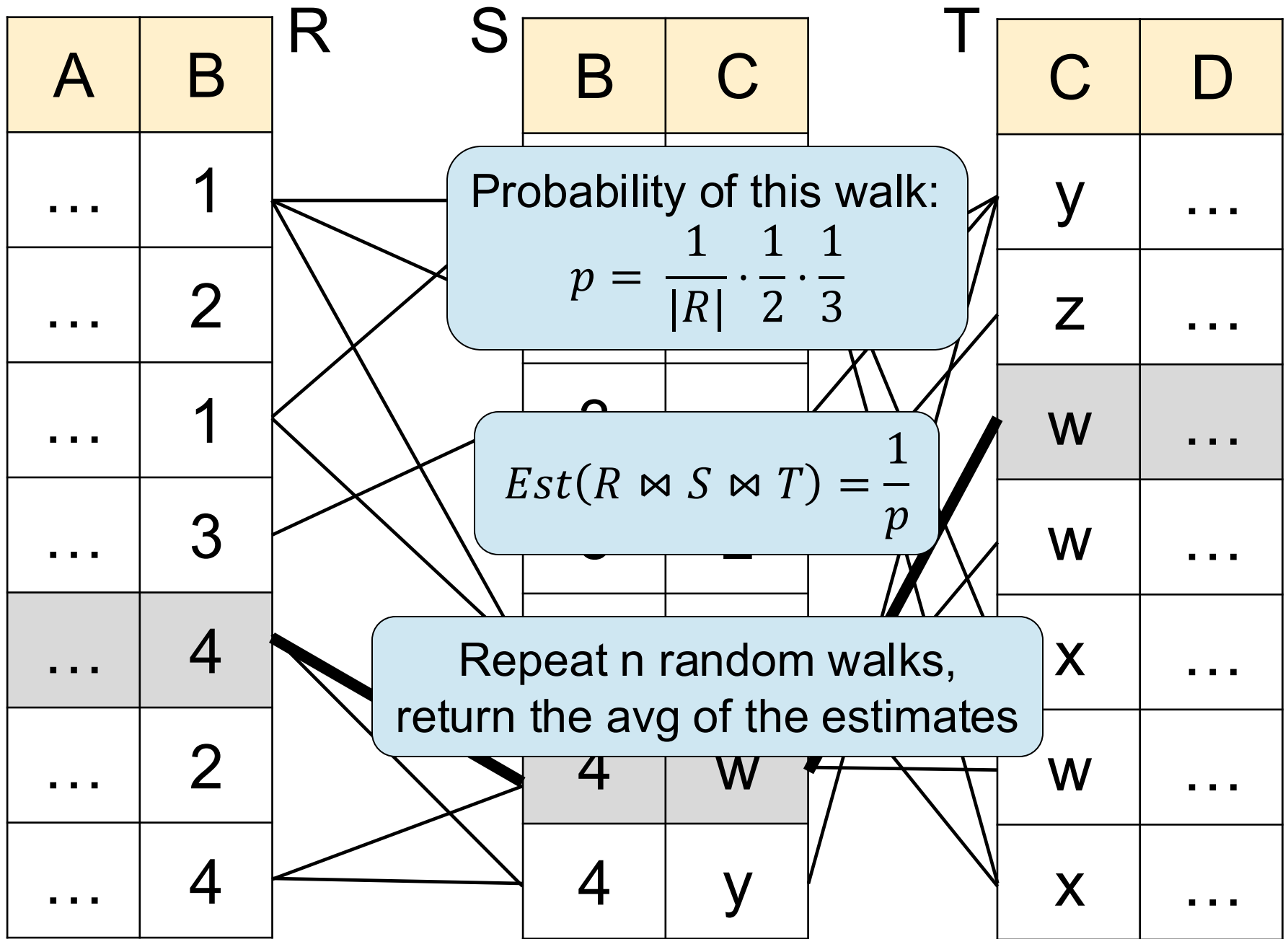












Discussion

- Sampling based CE have best precision
- However:
 - Offline samples need to be too large
 - Online samples require access to indices
- Some systems use offline sample for single-table predicates only