COL362/632 Introduction to Database Management Systems

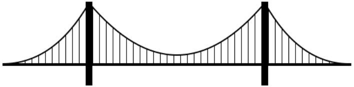
Query Optimization – Plan Enumeration

Kaustubh Beedkar

Department of Computer Science and Engineering
Indian Institute of Technology Delhi

Declarative Programming

What?



Query Optimization

How?

Imperative Programs

WIKIPEDIA
The Free Encyclopedia

Create account Log in

Patricia Selinger

3 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

Patricia G. Selinger is an American computer scientist and IBM Fellow, best known for her work on relational database management systems.

Selinger in 2014

Education [edit]

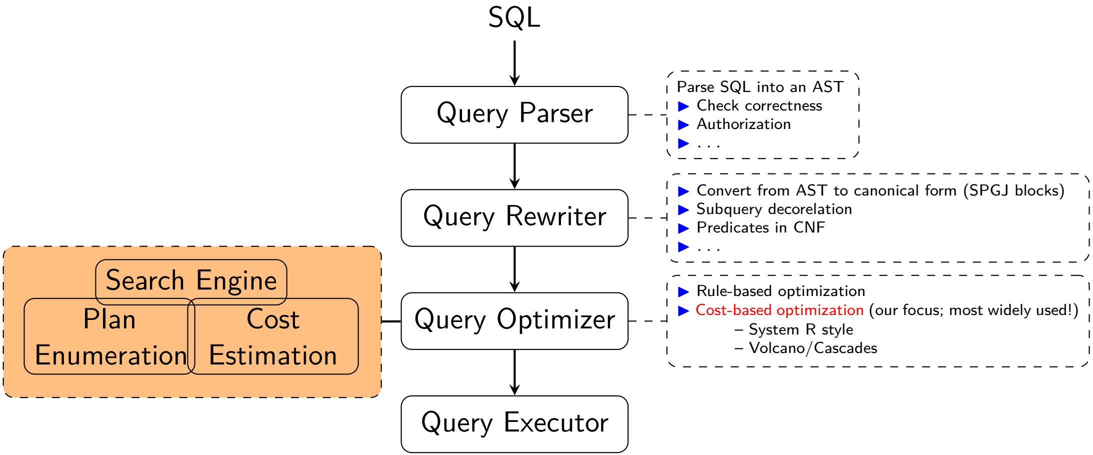
She received A.B. (1971), S.M. (1972), and Ph.D. (1975) degrees in applied mathematics from Harvard University.[1]

Biography [edit]

She played a fundamental role in the development of System R, a pioneering relational database implementation, and wrote the canonical paper on relational query optimization.[2] She is a pioneer in relational database management and inventor of the technique of cost-based query optimization. She was a key member of the original System R team that created the first relational database research prototype.[3][4][5] The dynamic programming algorithm for determining join order proposed in that paper still forms the basis for most of the query optimizers used in modern relational systems. She also established and led IBM's Database Technology Institute, considered one of the most successful examples of a fast technology pipeline from research to development and personally has technical contributions in the areas of database optimization, data parallelism, distributed data, and unstructured data management.[6][7] Before her retirement from IBM, she was the Vice President of Data Management Architecture and Technology at IBM.

Dr. Selinger was appointed an IBM Fellow in 1994, IBM's highest technical recognition, and is an ACM Fellow (2009) and a Fellow of the American Academy of Arts and Sciences.[8][9] She was also elected a member of the National Academy of Engineering (1999) for leadership and contributions to relational database technology.

Query Parsing & Optimization



The Plan Space

What plans will produce **equivalent** answers?

Algebraic Equivalences

Rules for selections (σ)

1. cascade

   $$\sigma_{c_1 \wedge c_2 \wedge, \ldots, \wedge c_n}(R) \equiv \sigma_{c_1}(\sigma_{c_2}(\ldots(\sigma_{c_n}(R))))$$

2. selections can commute

   $$\sigma_{c_1}(\sigma_{c_2}(R)) \equiv \sigma_{c_2}(\sigma_{c_1}(R))$$

3. selection pushdown

   $$\sigma_c(R \bowtie S) \equiv (R \bowtie \sigma_c(S))$$

Algebraic Equivalences

Rules for projections (π)

1. cascade

   $$\pi_{a_1}(R) \equiv \pi_{a_1}(\pi_{a_1,a_2}(\pi_{\ldots}(\pi_{a_1,a_2,\ldots,a_n}(R))))$$

2. pushdown

   $$\pi_{c,d}(R \bowtie_{R.a=S.b} S) \equiv \pi_{c,d}(\pi_{a,c}(R(a,c)) \bowtie_{R.a=S.b} \pi_{b,d}(S(b,d)))$$

Algebraic Equivalences

Rules for cross product ($\times$)

1. Associative

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

2. Commutative

   $R \times S \equiv S \times R$

Algebraic Equivalences

Rules for joins ($\bowtie$)

- $R \bowtie (S \bowtie T) \equiv (R \bowtie S) \bowtie T$ //not always true!
  - Consider $R(A, B)$, $S(B, C)$, and $T(A, D)$
    $(R \bowtie_{R.B=S.B} S) \bowtie_{A=T.A} T \equiv R \bowtie_{R.B=B \wedge R.A=A} (S \times T)$
  - Consider $R(A, B)$, $S(A, C)$, and $T(B, C)$
    $R \bowtie (S \bowtie T) \equiv (R \bowtie S) \bowtie T$

  Join reordering

  - Consider $R(A, C)$, $S(A, B)$, and $T(B, D)$ and $R.A = S.A$ and $S.B = T.B$

$$\begin{aligned}(R \bowtie_{A=A} S) \bowtie_{B=B} T &\equiv R \bowtie_{A=A} (S \bowtie_{B=B} T) \\ &\equiv (R \times T) \bowtie_{A=A \wedge B=B} S \\ &\equiv R \bowtie_{A=A} (T \bowtie_{B=B} S)\end{aligned}$$

Physical Equivalences

- Scans and selections
  - File Scan
  - Index Scan
- Equi Joins
  - Nested Loop Join
  - Block Nested Loop Join
  - Sort Merge Join
  - Hash Join
- Theta Joins
  - Nested Loop Join
  - Block Nested Loop Join
- Processing
  - Materialization
  - Pipelining

Example

Supplier(sid, name, city, state)
PartSupplier(sid,pno,quantity)

```
select name from Supplier S, PartSupplier PS where S.sid = PS.sid and
PS.pno=2 and S.city = 'Udaipur' and S.state='RJ'
```

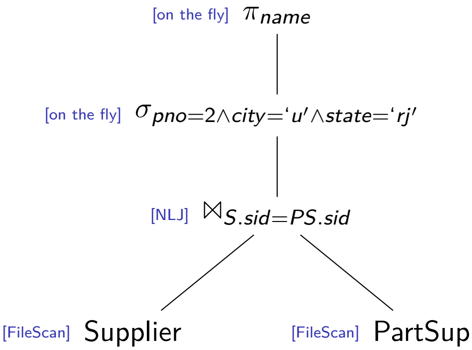
Catalogue

- $n_S = 100,000$, $n_{PS} = 200,000$
- $b_S = 5000$, $b_{PS} = 1000$
- $V(S, city) = 20$, $V(S, state) = 10$, $V(PS, pno) = 250$
- PS has clustered index on (pno)
- S has non-clustered index on (sid)
- $B = 11$ //buffer pages available

Candiate Plan 1

Catalogue

- $n_S = 100,000$
- $n_{PS} = 200,000$
- $b_S = 5000$
- $b_{PS} = 1000$
- $V(S, city) = 20,\ V(S, state) = 10,\ V(PS, pno) = 250$
- PS has clustered index on (pno)
- S has non-clustered index on (sid)
- $B = 11$ //buffer pages available



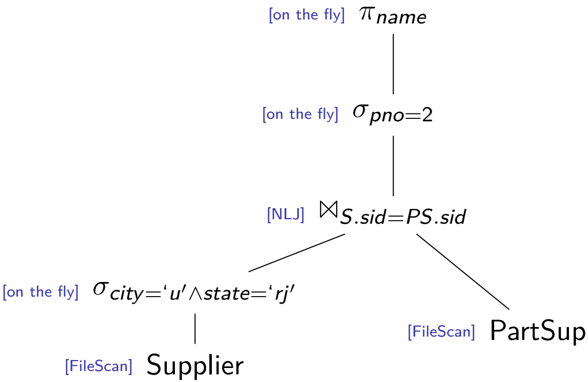
- Scan cost(Supplier) $= 5000$
- Scan cost(PartSupp) $= 1000$

- Plan cost $= 5000 + (5000 \times 1000)$
  $=$5,005,000 I/Os

Candiate Plan 2

Catalogue

- $n_S = 100,000$
- $n_{PS} = 200,000$
- $b_S = 5000$
- $b_{PS} = 1000$
- $V(S, city) = 20,\ V(S, state) = 10,\ V(PS, pno) = 250$
- PS has clustered index on (pno)
- S has non-clustered index on (sid)
- $B = 11$ //buffer pages available



- Scan cost(Supplier) $= 5000$
- Scan cost(PartSupp) $= 1000$

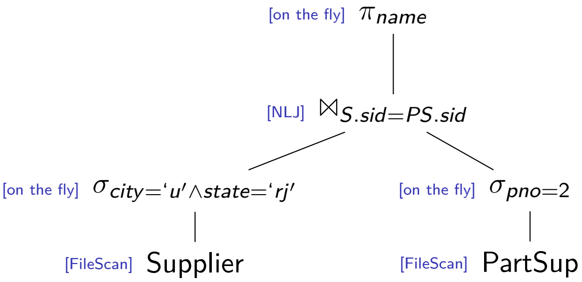
- cardinality after filter on Supplier $= \frac{100,000}{20 \times 10} = 500$
- $\#blocks = \left\lceil \frac{500}{20} \right\rceil = 25$ (20tuples/block)

- Plan cost $= 5000 + (25 \times 1000) = 30,000$ I/Os

Candiate Plan 3

Catalogue

- $n_S = 100,000$ ▶ $n_{PS} = 200,000$ ▶ $b_S = 5000$ ▶ $b_{PS} = 1000$ ▶ $V(S, city) = 20,\ V(S, state) = 10,\ V(PS, pno) = 250$ ▶ PS has clustered index on (pno)
- S has non-clustered index on (sid) ▶ $B = 11$ //buffer pages available

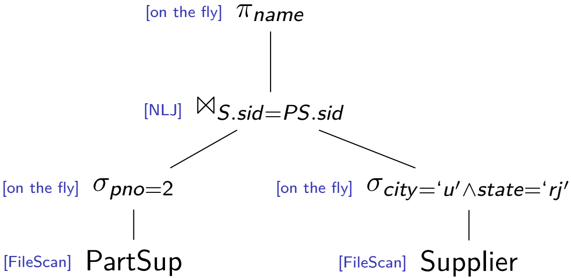


- Scan cost(Supplier) $= 5000$
- Scan cost(PartSupp) $= 1000$
- cadinality after filter on PartSupp $= \frac{200{,}000}{250} = 800$
- #blocks $= \left\lceil \frac{800}{200} \right\rceil = 4$ 200tuples/block
- Plan cost $= 5000 + (25 \times 1000) = 30{,}000$ I/Os

Candiate Plan 4

Catalogue

- $n_S = 100,000$ ▶ $n_{PS} = 200,000$ ▶ $b_S = 5000$ ▶ $b_{PS} = 1000$ ▶ $V(S, city) = 20,\ V(S, state) = 10,\ V(PS, pno) = 250$ ▶ PS has clustered index on (pno)
- S has non-clustered index on (sid) ▶ $B = 11$ //buffer pages available



- Scan cost(Supplier) $= 5000$
- Scan cost(PartSupp) $= 1000$

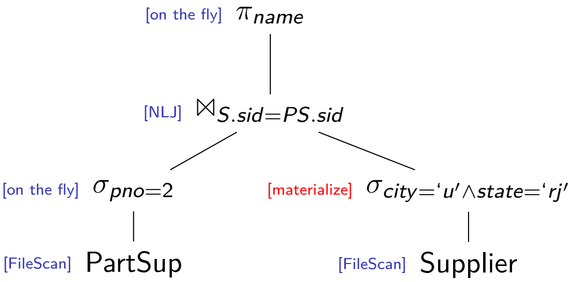
- #blocks after $\sigma(S) = 25$
- #blocks after $\sigma(PS) = 4$

- Plan cost $= 1000 + (4 \times 5000) =$ 21,000 I/Os

Candiate Plan 5

Catalogue

- $n_S = 100,000$ ▶ $n_{PS} = 200,000$ ▶ $b_S = 5000$ ▶ $b_{PS} = 1000$ ▶ $V(S, city) = 20,\ V(S, state) = 10,\ V(PS, pno) = 250$ ▶ PS has clustered index on (pno)
- S has non-clustered index on (sid) ▶ $B = 11$ //buffer pages available



- Scan cost(Supplier) $= 5000$
- Write cost$(\sigma(S)) = 25$
- Scan cost(PartSupp) $= 1000$

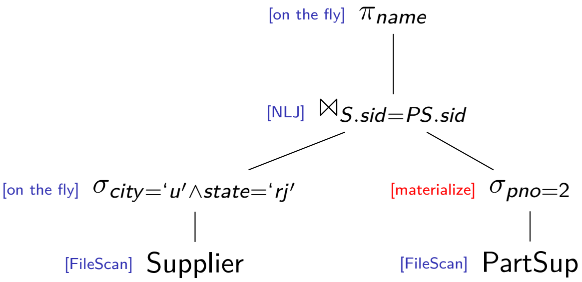
- #blocks after $\sigma(S) = 25$
- #blocks after $\sigma(PS) = 4$

- Plan cost
  $= 1000 + (4 \times 25) + 5000 + 25$
  $= 6125$ I/Os

Candiate Plan 6

Catalogue

- $n_S = 100,000$ ▶ $n_{PS} = 200,000$ ▶ $b_S = 5000$ ▶ $b_{PS} = 1000$ ▶ $V(S, city) = 20,\ V(S, state) = 10,\ V(PS, pno) = 250$ ▶ PS has clustered index on (pno)
- S has non-clustered index on (sid) ▶ $B = 11$ //buffer pages available



- Scan cost(Supplier) $= 5000$
- Scan cost(PartSupp) $= 1000$
- Write cost$(\sigma(PS)) = 4$

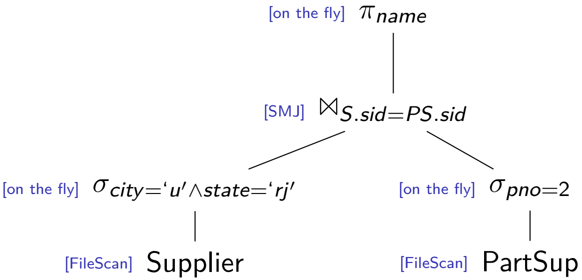
- #blocks after $\sigma(S) = 25$
- #blocks after $\sigma(PS) = 4$

- Plan cost
  $= 5000 + (25 \times 4) + 1000 + 4$
  $= 6104$ I/Os

Candiate Plan 7

Catalogue

- $n_S = 100,000$ ▶ $n_{PS} = 200,000$ ▶ $b_S = 5000$ ▶ $b_{PS} = 1000$ ▶ $V(S, city) = 20,\ V(S, state) = 10,\ V(PS, pno) = 250$ ▶ PS has clustered index on (pno)
- S has non-clustered index on (sid) ▶ $B = 11$ //buffer pages available



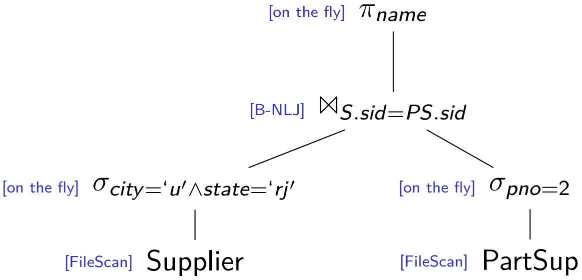
- Plan cost
  $= 5000 + 75 + 1000 + 4 + 29$
  $= 6108$ I/Os

- Scan cost(Supplier) $= 5000$
- Scan cost(PartSupp) $= 1000$
- #blocks after $\sigma(S) = 25$
- #blocks after $\sigma(PS) = 4$
- Sort cost(Supplier)
  - pass $0 = 25$ (only write cost)
  - #passes $= \left\lceil log_{10} \left\lceil \frac{25}{11} \right\rceil \right\rceil = 1$
  - cost $= 25 + (2 \times 25 \times 1) = 75$
- Sort cost(PartSupp)
  - pass $0 = 4$ (only write cost)
  - #passes $= \left\lceil log_{10} \left\lceil \frac{4}{11} \right\rceil \right\rceil = 0$
  - cost $= 4 + (2 \times 4 \times 0) = 4$
- Merge cost $= 25 + 4 = 29$

Candiate Plan 8

Catalogue

▶ $n_S = 100,000$ ▶ $n_{PS} = 200,000$ ▶ $b_S = 5000$ ▶ $b_{PS} = 1000$ ▶ $V(S, city) = 20$, $V(S, state) = 10$, $V(PS, pno) = 250$ ▶ PS has clustered index on (pno) ▶ S has non-clustered index on (sid) ▶ $B = 11$ //buffer pages available



- Scan cost(Supplier) $= 5000$
- Scan cost(PartSupp) $= 1000$

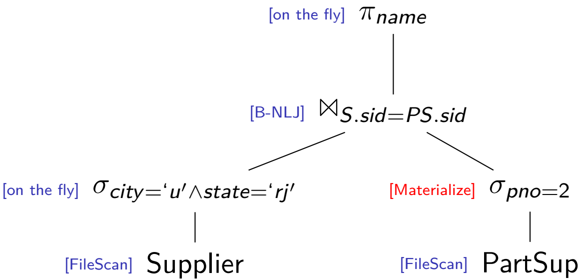
- blocking unit for $\sigma(S)$ $= \left\lceil \frac{25}{11-2} \right\rceil = 3$

- Plan cost $= 5000 + (3 \times 1000)$ $=$ 8000 I/Os

Candiate Plan 9

Catalogue

- $n_S = 100,000$
- $n_{PS} = 200,000$
- $b_S = 5000$
- $b_{PS} = 1000$
- $V(S, city) = 20$, $V(S, state) = 10$, $V(PS, pno) = 250$
- PS has clustered index on (pno)
- S has non-clustered index on (sid)
- $B = 11$ //buffer pages available

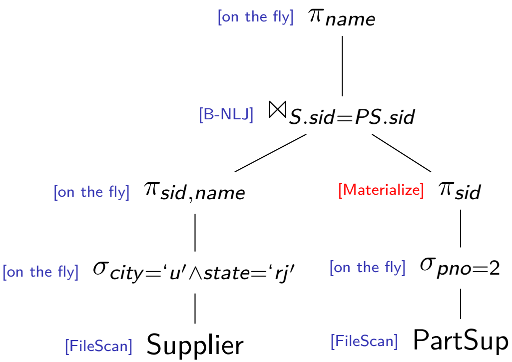


- Scan cost(Supplier) $= 5000$
- Scan cost(PartSupp) $= 1000$
- Write cost($\sigma(PS)$) $= 4$
- blocking unit for $\sigma(S)$ $= \left\lceil \frac{25}{11-2} \right\rceil = 3$
- Plan cost $= 5000 + (3 \times 4) + 1000 + 4$ $= 6016$ I/Os

Candiate Plan 10

Catalogue

- $n_S = 100,000$ ▶ $n_{PS} = 200,000$ ▶ $b_S = 5000$ ▶ $b_{PS} = 1000$ ▶ $V(S, city) = 20$, $V(S, state) = 10$, $V(PS, pno) = 250$ ▶ PS has clustered index on (pno)
- S has non-clustered index on (sid) ▶ $B = 11$ //buffer pages available



- Scan cost(Supplier) $= 5000$
- Scan cost(PartSupp) $= 1000$

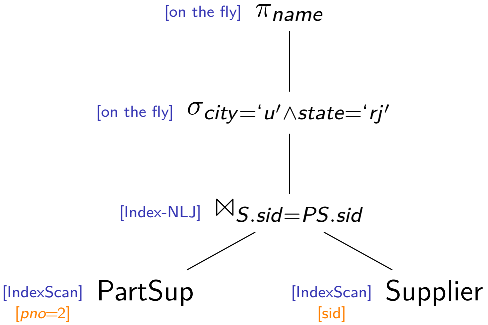
- #blocks after $\sigma(S) = 25$
- #blocks after $\pi_{sid,name} = 2$
- #blocks after $\sigma(PS) = 4$
- #blocks after $\pi_{sid} = 2$
- Write cost$(\pi_{sid}(\sigma(PS)) = 2$

- Plan cost
  $= 5000 + (2 \times 2) + 1000 + 2$
  $=$ 6006 I/Os

Candiate Plan 11

Catalogue

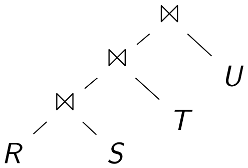
- $n_S = 100,000$ ▶ $n_{PS} = 200,000$ ▶ $b_S = 5000$ ▶ $b_{PS} = 1000$ ▶ $V(S, city) = 20$, $V(S, state) = 10$, $V(PS, pno) = 250$ ▶ PS has clustered index on (pno)
- S has non-clustered index on (sid) ▶ $B = 11$ //buffer pages available



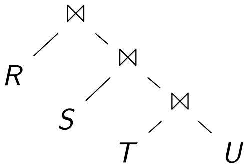
- Index scan cost(PartSupp) $= 4$ (clustering index on pno)
- #tuples after $\sigma(PS) = 800$
- Index cost(Supplier) $= 1$ (non-clustering index: clustering/non-clustering index does not matter here as sid is key!)
- Plan cost $= 4 + (800 \times 1) = 804$ I/Os

Common Tree Shapes

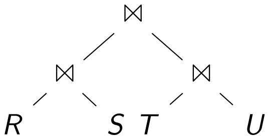
- Left-deep tree



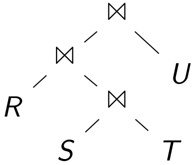
- Right-deep tree



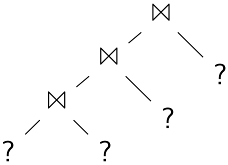
- Bushy tree



- Zigzag tree



Search Space of Left-deep Plans



- Number of left-deep trees with n relations $= n!$

Catalan Numbers

- The number of binary trees with n leaf nodes $= \mathcal{C}_{n-1}$

$$\mathcal{C}_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{(n+1)!n!} \qquad \binom{2n}{n} = \frac{(2n)!}{n!n!}$$

$$\mathcal{C}_0 = 1$$

$$\mathcal{C}_{n+1} = \sum_{i=0}^{n} \mathcal{C}_i \mathcal{C}_{n-i} \text{ for } n \geq 0$$

Search Space for Bushy Plans

- Space of bushy plans $\supseteq$ Space of Left-deep plans

$$
\begin{aligned}
&= n!\mathcal{C}_{n-1} \\
&= \frac{(2n-2)!}{(n-1)!}
\end{aligned}
$$

Summary

- Plan Space can be huge, even for simple queries!
- Query optimization problem is NP-hard
- Number of plans grows exponentially with query complexity
  - Average query has 1000s possible join orders
  - 10s of physical operators
  - couple of access methods
  - combined with algebraic equivalences

  $\Longrightarrow$ millions of alternatives!

Goetz Graefe

[...] "if you have 17 join algorithms in your system, chances are you'll hardly ever pick the optimal one. In fact, you should be happy if you always pick a good one. And it's unlikely to be the case."