

# CSE 444: Database Internals

## Lecture 10 Query Optimization (part 1)

# Reminders

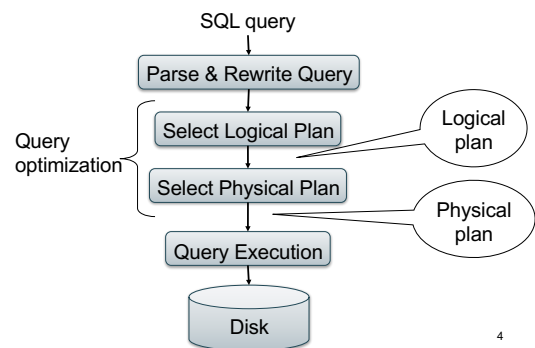
- HW2 is due tonight
- 5th year master's reading is due tonight
- Lab2 EXTENDED to WEDNESDAY

Know how to compute the cost of a plan

Next: Find a good plan automatically?

This is the role of the query optimizer

# Query Optimization Overview



# What We Already Know...

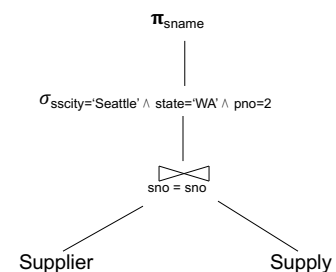
Supplier (sno, sname, scity, sstate)  
Part (pno, pname, psize, pcolor)  
Supply (sno, pno, price)

For each SQL query....

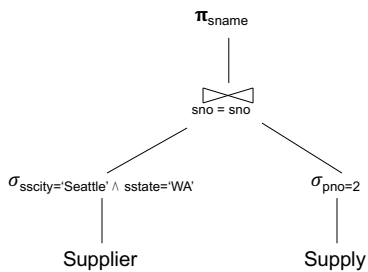
```
SELECT S.sname
FROM Supplier S, Supply U
WHERE S.scity='Seattle' AND S.sstate='WA'
AND S.sno = U.sno
AND U.pno = 2
```

There exist many logical query plan...

# Example Query: Logical Plan 1



## Example Query: Logical Plan 2



CSE 444 - Winter 2017

7

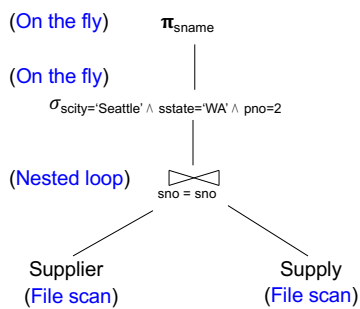
## What We Also Know

- For each logical plan...
- There exist many physical plans

CSE 444 - Winter 2017

8

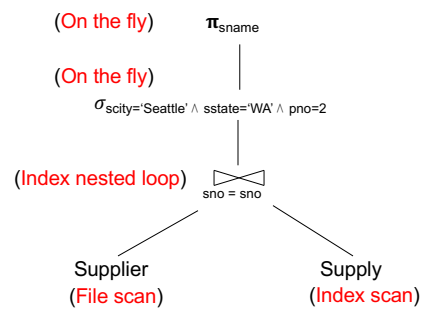
## Example Query: Physical Plan 1



CSE 444 - Winter 2017

9

## Example Query: Physical Plan 2



CSE 444 - Winter 2017

10

## Query Optimizer Overview

- **Input:** A logical query plan
- **Output:** A good physical query plan
- **Basic query optimization algorithm**
  - Enumerate alternative plans (logical and physical)
  - Compute estimated cost of each plan
    - Compute number of I/Os
    - Optionally take into account other resources
  - Choose plan with lowest cost
  - This is called cost-based optimization

CSE 444 - Winter 2017

11

## Lessons

- No magic “best” plan: depends on the data
- In order to make the right choice
  - Need to have **statistics** over the data
  - The B's, the T's, the V's
  - Commonly (and in SimpleDB): histograms over base data

CSE 444 - Winter 2017

12

## Outline

- Search space
- Algorithm for enumerating query plans

CSE 444 - Winter 2017

13

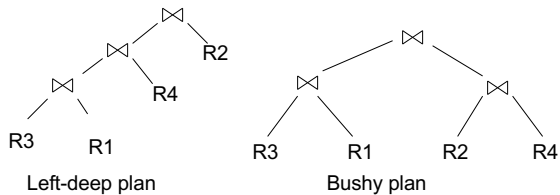
## Relational Algebra Equivalences

- Selections
  - Commutative:  $\sigma_{c_1}(\sigma_{c_2}(R))$  same as  $\sigma_{c_2}(\sigma_{c_1}(R))$
  - Cascading:  $\sigma_{c_1 \wedge c_2}(R)$  same as  $\sigma_{c_2}(\sigma_{c_1}(R))$
- Projections
  - Cascading
- Joins
  - Commutative:  $R \bowtie S$  same as  $S \bowtie R$
  - Associative:  $R \bowtie (S \bowtie T)$  same as  $(R \bowtie S) \bowtie T$

CSE 444 - Winter 2017

14

## Left-Deep Plans, Bushy Plans, and Linear Plans



Linear plan: One input to each join is a relation from disk  
Can be either left or right input

CSE 444 - Winter 2017

15

## Commutativity, Associativity, Distributivity

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

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

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

CSE 444 - Winter 2017

16

## Laws Involving Selection

$$\sigma_{C \text{ AND } C'}(R) = \sigma_C(\sigma_{C'}(R)) = \sigma_C(R) \cap \sigma_{C'}(R)$$

$$\sigma_{C \text{ OR } C'}(R) = \sigma_C(R) \cup \sigma_{C'}(R)$$

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

$$\sigma_C(R - S) = \sigma_C(R) - S$$

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

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

Assuming C on attributes of R

CSE 444 - Winter 2017

17

## Example: Simple Algebraic Laws

- Example:  $R(A, B, C, D), S(E, F, G)$ 
  - $\sigma_{F=3}(R \bowtie_{D=E} S) = ?$
  - $\sigma_{A=5 \text{ AND } G=9}(R \bowtie_{D=E} S) = ?$

CSE 444 - Winter 2017

18

## Example: Simple Algebraic Laws

- Example:  $R(A, B, C, D), S(E, F, G)$   
 $\sigma_{F=3}(R \bowtie_{D=E} S) = R \bowtie_{D=E} \sigma_{F=3}(S)$   
 $\sigma_{A=5 \text{ AND } G=9}(R \bowtie_{D=E} S) = \sigma_{A=5}(R) \bowtie_{D=E} \sigma_{G=9}(S)$

CSE 444 - Winter 2017

19

## Laws Involving Projections

$$\Pi_M(R \bowtie S) = \Pi_M(\Pi_P(R) \bowtie \Pi_Q(S))$$

$$\Pi_M(\Pi_N(R)) = \Pi_M(R)$$

*/\* note that  $M \subseteq N$  \*/*

- Example  $R(A,B,C,D), S(E, F, G)$   
 $\Pi_{A,B,G}(R \bowtie_{D=E} S) = \Pi_{A,B,G}(\Pi_{A,B,C,D}(R) \bowtie_{D=E} \Pi_{E,F,G}(S))$

CSE 444 - Winter 2017

20

## Laws Involving Projections

$$\Pi_M(R \bowtie S) = \Pi_M(\Pi_P(R) \bowtie \Pi_Q(S))$$

$$\Pi_M(\Pi_N(R)) = \Pi_M(R)$$

*/\* note that  $M \subseteq N$  \*/*

- Example  $R(A,B,C,D), S(E, F, G)$   
 $\Pi_{A,B,G}(R \bowtie_{D=E} S) = \Pi_{A,B,G}(\Pi_{A,B,C,D}(R) \bowtie_{D=E} \Pi_{E,F,G}(S))$

CSE 444 - Winter 2017

21

## Laws involving grouping and aggregation

$$\gamma_{A, \text{agg}(D)}(R(A,B) \bowtie_{B=C} S(C,D)) = \gamma_{A, \text{agg}(D)}(R(A,B) \bowtie_{B=C} (\gamma_{C, \text{agg}(D)} S(C,D)))$$

CSE 444 - Winter 2017

22

## Laws involving grouping and aggregation

$$\delta(\gamma_{A, \text{agg}(B)}(R)) = \gamma_{A, \text{agg}(B)}(R)$$

$$\gamma_{A, \text{agg}(B)}(\delta(R)) = \gamma_{A, \text{agg}(B)}(R)$$

*if agg is "duplicate insensitive"*

Which of the following are "duplicate insensitive" ?  
sum, count, avg, min, max

CSE 444 - Winter 2017

23

## Laws Involving Constraints

Product(pid, pname, price, cid)  
Company(cid, cname, city, state)

Foreign key

$$\Pi_{\text{pid, price}}(\text{Product} \bowtie_{\text{cid=cid}} \text{Company}) = \Pi_{\text{pid, price}}(\text{Product})$$

CSE 444 - Winter 2017

24

## Search Space Challenges

- Search space is huge!
  - Many possible equivalent trees
  - Many implementations for each operator
  - Many access paths for each relation
    - File scan or index + matching selection condition
- Cannot consider ALL plans
  - Heuristics: only partial plans with “low” cost

CSE 444 - Winter 2017

25

## Outline

- Search space
- Algorithm for enumerating query plans

CSE 444 - Winter 2017

26

## Key Decisions

### Logical plan

- What logical plans do we consider (left-deep, bushy?); *Search Space*
- Which algebraic laws do we apply, and in which context(s)?; *Optimization rules*
- In what order do we explore the search space?; *Optimization algorithm*

CSE 444 - Winter 2017

27

## Key Decisions

### Physical plan

- What physical operators to use?
- What access paths to use (file scan or index)?
- Pipeline or materialize intermediate results?

These decisions also affect the *search space*

CSE 444 - Winter 2017

28

## Two Types of Optimizers

- Heuristic-based optimizers:
  - Apply greedily rules that always improve plan
    - Typically: push selections down
  - Very limited: no longer used today
- Cost-based optimizers:
  - Use a cost model to estimate the cost of each plan
  - Select the “cheapest” plan
  - We focus on cost-based optimizers

CSE 444 - Winter 2017

29

## Three Approaches to Search Space Enumeration

- Complete plans
- Bottom-up plans
- Top-down plans

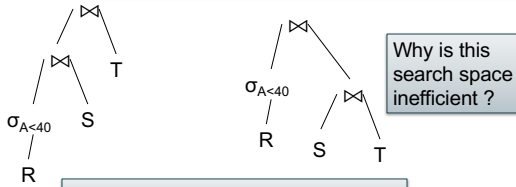
CSE 444 - Winter 2017

30

## Complete Plans

R(A,B)  
S(B,C)  
T(C,D)

```
SELECT *
FROM R, S, T
WHERE R.B=S.B and S.C=T.C and R.A<40
```



Answer: No way to do early pruning

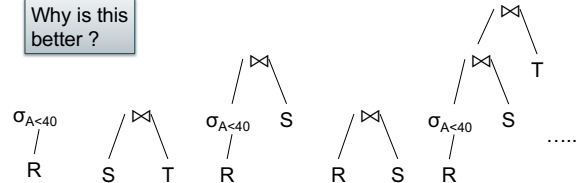
31

## Bottom-up Partial Plans

R(A,B)  
S(B,C)  
T(C,D)

```
SELECT *
FROM R, S, T
WHERE R.B=S.B and S.C=T.C and R.A<40
```

Why is this better?



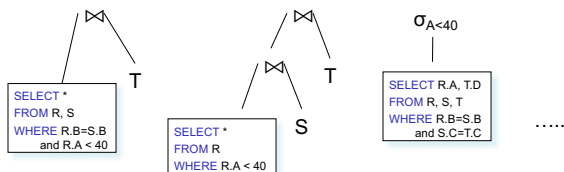
We will prune bad plans for sub-expressions

32

## Top-down Partial Plans

R(A,B)  
S(B,C)  
T(C,D)

```
SELECT *
FROM R, S, T
WHERE R.B=S.B and S.C=T.C and R.A<40
```



CSE 444 - Winter 2017

33

## Two Types of Plan Enumeration Algorithms

- Dynamic programming (in class)
  - Based on System R (aka Selinger) style optimizer[1979]
  - Limited to joins: *join reordering algorithm*
  - Bottom-up
- Rule-based algorithm (will not discuss)
  - Database of rules (=algebraic laws)
  - Usually: dynamic programming
  - Usually: top-down

CSE 444 - Winter 2017

34