Discussion 6

(Query Optimization)

Matthew Deng

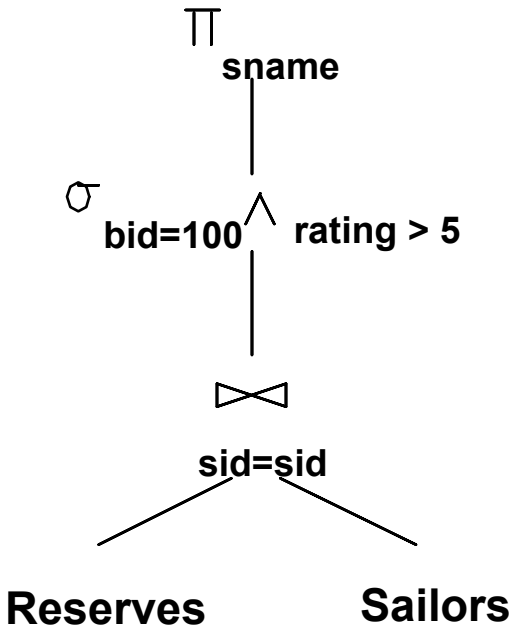
Query Optimization

Query Optimization

- What is the best way to run a query?

```
SELECT  S.sname
FROM  Reserves R, Sailors S
WHERE  R.sid=S.sid AND
   R.bid=100 AND S.rating>5
```

$$\pi_{(\text{sname})}\sigma_{(\text{bid}=100 \wedge \text{rating} > 5)}(\text{Reserves} \bowtie \text{Sailors})$$

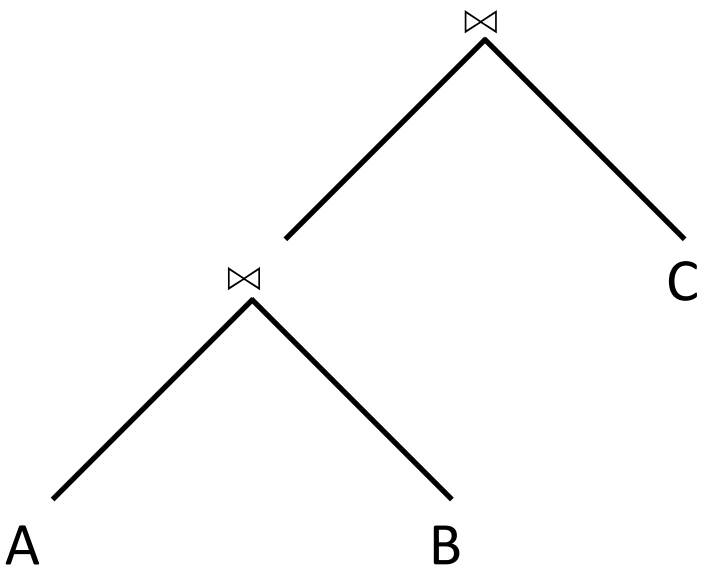


System R (Selinger) Optimizer

1. Plan Space
   - input
2. Cost Estimation
   - subroutine
3. Search Strategy
   - algorithm

Plan Space

- Left-deep trees only
- Various access methods
  - Equality vs. Range
  - Scan vs. index
  - Clustered vs. unclustered
- Various join algorithms
  - Index Nested Loop Join
  - Sort Merge Join
  - Chunk Nested Loop Join
  - Hash Join

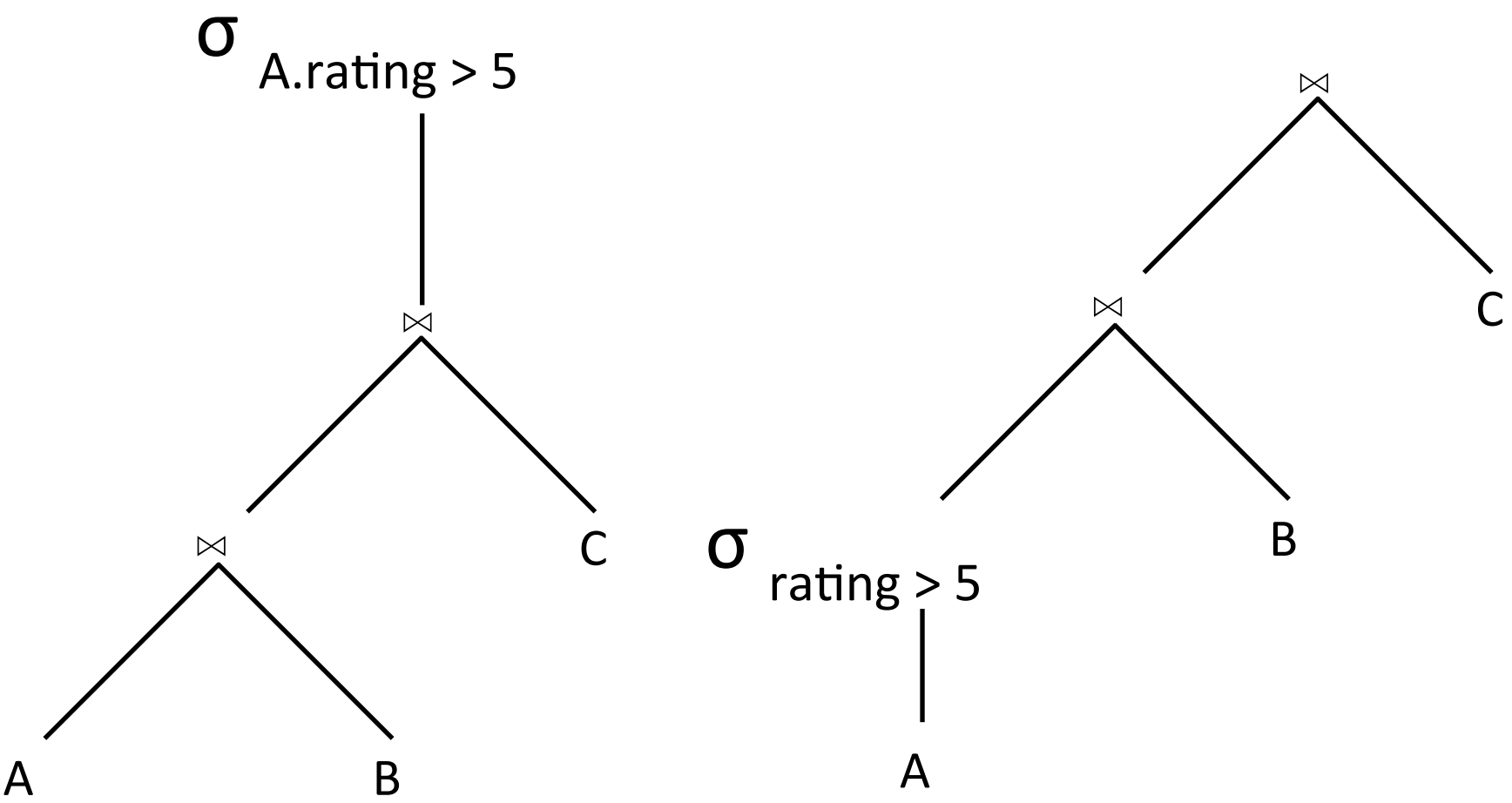


((A ⋈ B) ⋈ C)

(((A ⋈ B) ⋈ C) … ⋈ Z)

- Selections and projections as low as possible

Pushing Selections/Projections



Selections and projections as low as possible

Cost Estimation

- Estimate cost of individual tree nodes
- Estimate output cardinalities
  - Selectivity (reduction factor) estimation

Statistic	Meaning
Ntuples	Number of tuples in a table (Cardinality)
Npages	Number of disk pages in a table
Low	Minimum value in a column
High	Maximum value in a column
Nkeys	Number of distinct values in a column
IHeight	Height of an index
INPages	Number of disk pages in an index

Selectivity (Reduction Factor)

- Associated with each predicate term
- |output| / |input|
- Result cardinality = max output cardinality * $\Pi_{selectivity}$
  - Max output cardinality = $\Pi_{Ntuples}$

Selectivity Estimation

- Assuming values are uniformly distributed and terms are independent:

Term	Selectivity
col = value	1/NKeys(col)
col1 = col2	1/MAX(NKeys(col1), NKeys(col2))
col > value	(High(col) – value) / (High(col) – Low(col) + 1)

- If we missing information to estimate selectivity, assume selectivity = 1/10

Histograms

- Use for better selectivity estimatation
  - Not all distributions are uniform!
- Compare bucket by bucket
- Equiwidth
  - Good for normal distributions
  - Easy to implement
- Equidepth
  - Good for median, quantile, etc.
  - Handles outliers

Search Strategy

- Base case: single-relation plans
- Induction: multi-relation plans

Single-Relation Plans

- Sequential scan of file:
  - NPages(R)
- Index I on primary key matches selection
  - Height(I) + 1 for a B+ tree
- Selects on clustered index I:
  - (NPages(I) + NPages(R)) * $\Pi_{selectivity}$
- Selects on non-clustered index I:
  - (NPages(I) + NTuples(R)) * $\Pi_{selectivity}$

Interesting Orders

A relation has an interesting order if it is sorted by any of the following:

- ORDER BY attributes
  - Already ordered!
- GROUP BY attributes
  - Already grouped!
- Future join attributes
  - Already sorted!
  - Cheaper sort-merge join

Search Algorithm

- Step 1: Find the best 1-table access method.
- Step 2: Given the best 1-table method as the outer, find the best 2-table.
- …
- Step N: Given the best (N-1)-table method as the outer, find the best N-table.

Query Optimization Worksheet

Query Optimization Exercises

1. Name and describe the three factors of a System R (Selinger) Optimizer

Query Optimization Exercises

1. Name and describe the three factors of a System R (Selinger) Optimizer

- plan space
  - only left-deep join trees
  - avoid cartesian products
  - push selections and joins
  - interesting orders
- cost estimation
  - use cost formulas and size estimations
  - selectivity (reduction factor) estimation = |output| / |input|
- search algorithm
  - dynamic programming

Query Optimization Exercises

2. What is an interesting order?

Query Optimization Exercises

2. What is an interesting order?

A relation is said to have an interesting order if it is sorted by any of the following:

- ORDER BY attributes
- GROUP BY attributes
- downstream join attributes

Query Optimization Exercises

3. What are the best single-table plans (i.e., Phase 1)?

Query Optimization Exercises

3. What are the best single-table plans (i.e., Phase 1)?

Humans:

- File scan: 1000 IOs
- Index #4:
  - RF = 1200 / 50,000
  - Cost for unclustered index = (NPages(I) + NTuples(R)) * $\Pi RF_{matching}$
    = (20 + 50,000) * (1200 / 50000)
    = 1200 IOs
- Best single table plan: file scan

Query Optimization Exercises

3. What are the best single-table plans (i.e., Phase 1)?

Kitties:

- File scan: 100 IOs
- no other applicable indexes since no single-table predicates for Kitties
- Best single table plan: file scan

Query Optimization Exercises

3. What are the best single-table plans (i.e., Phase 1)?

Puppies:

- File scan: 50 IOs
- Index #2:
  - RF = 1/10 (because happiness can go from 1-10)
  - Cost for unclustered index = (NPages(I) + NTuples(R)) * $\Pi RF_{matching}$
    = (5 + 200) * (1/10)
    = 21 I/Os
- Index #3:
  - Index #3 is grouped on the owner, then by happiness. Since the only single table predicate we can consider is "P.happiness = 7", this is worse than an unclustered index, since it's not sorted on the search key
- Best single table plan: Index #2

Query Optimization Exercises

3. What are the best single-table plans (i.e., Phase 1)?

   Humans: File scan

   Kitties: File scan

   Puppies: Index #2

Query Optimization Exercises

4. List the pairs of tables the optimizer will consider for 2-way joins (i.e., Phase 2)?

Query Optimization Exercises

4. List the pairs of tables the optimizer will consider for 2-way joins (i.e., Phase 2)?

   Humans: File scan

   Kitties: File scan

   Puppies: Index #2 (Unclustered B+ Tree)

Query Optimization Exercises

4. List the pairs of tables the optimizer will consider for 2-way joins (i.e., Phase 2)?

   Humans[File scan] ⋈ Kitties

   Humans[File scan] ⋈ Puppies

   Kitties[File scan] ⋈ Humans

   Kitties[File scan] ⋈ Puppies

   Puppies[unclustered B+] ⋈ Humans

   Puppies[unclustered B+] ⋈ Kitties

Query Optimization Exercises

4. List the pairs of tables the optimizer will consider for 2-way joins (i.e., Phase 2)?

~~Humans[File scan] ⋈ Kitties~~

Humans[File scan] ⋈ Puppies

~~Kitties[File scan] ⋈ Humans~~

Kitties[File scan] ⋈ Puppies

Puppies[unclustered B+] ⋈ Humans

Puppies[unclustered B+] ⋈ Kitties

Query Optimization Exercises

4. List the pairs of tables the optimizer will consider for 2-way joins (i.e., Phase 2)?

   Humans[File scan] ⋈ Puppies

   Kitties[File scan] ⋈ Puppies

   Puppies[unclustered B+] ⋈ Humans

   Puppies[unclustered B+] ⋈ Kitties

Query Optimization Exercises

5. Which plans will be avoided?

Query Optimization Exercises

5. Which plans will be avoided?

   The (H ⋈ K) and (K ⋈ H) pairs will be avoided since they are cross-products. That is, they don't have a join predicate in the query.

Query Optimization Exercises

6. What would be the IO cost of doing Index Nested Loops Join using Puppies as the outer, with the optimal single table selection methods (see part 1)?

Query Optimization Exercises

6. What would be the IO cost of doing Index Nested Loops Join using Puppies as the outer, with the optimal single table selection methods (see question 3)?

cost = NPages(P) + $NTuples_{potential}$(P) * Lookup(p)

NPages(P) = 21 from q3 using Index #2.

$NTuples_{potential}$(P) ~20; 1/10(200 puppies) have happiness = 7

Lookup(p) ~ (5 + 400)*1/10 using Index #1 since we know K.cuteness = P.happiness

cost = 21 + 20* [(5 + 400)*1/10

= 831 I/Os

Query Optimization Exercises

7. Now with Kitties as the outer.

Query Optimization Exercises

7. Now with Kitties as the outer.

cost = NPages(K) + $NTuples_{potential}$ (K) * lookup(k)

NPages(K) = 100

$NTuples_{potential}$(K) = 400

Lookup(k): (15 + 50)*(1/10 owner)*(1/10 happiness) ~ 1 IO, using Index #3 and 1/10 owner from 1/Max(5, 10)

Cost = 100 + 400*1

= 500 I/Os