CS 186 Discussion #7

Query Optimization

Logistics

- Midterm 1 Regrades due 10/23

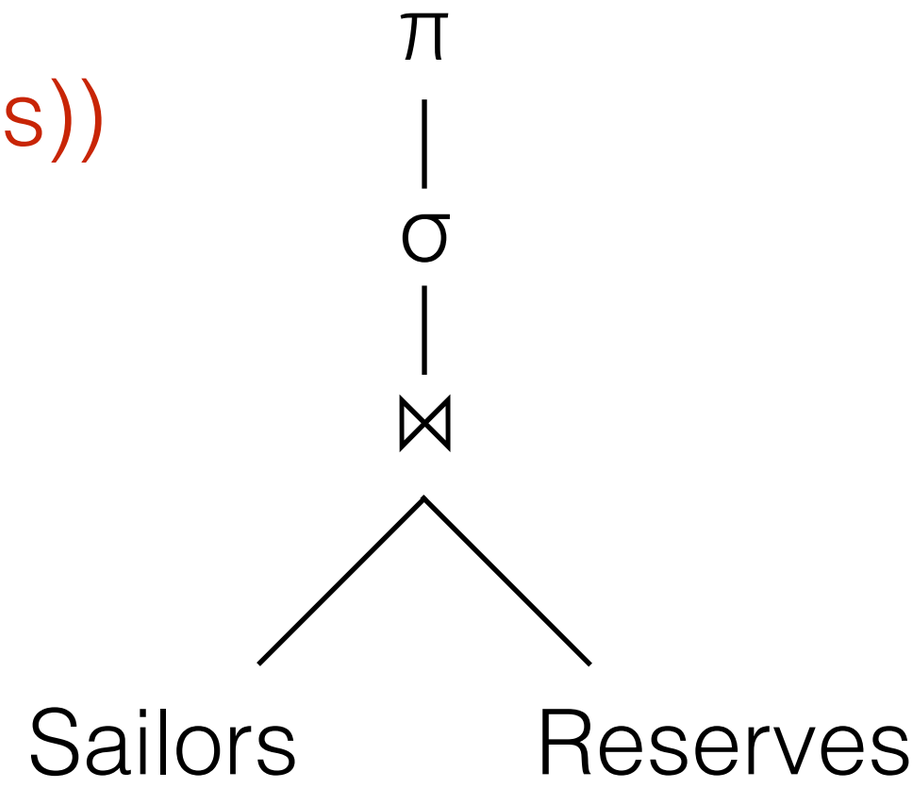
- Feedback Survey on Piazza

- Homework 4 coming soon
  - Covers Query Optimization

Query Optimization

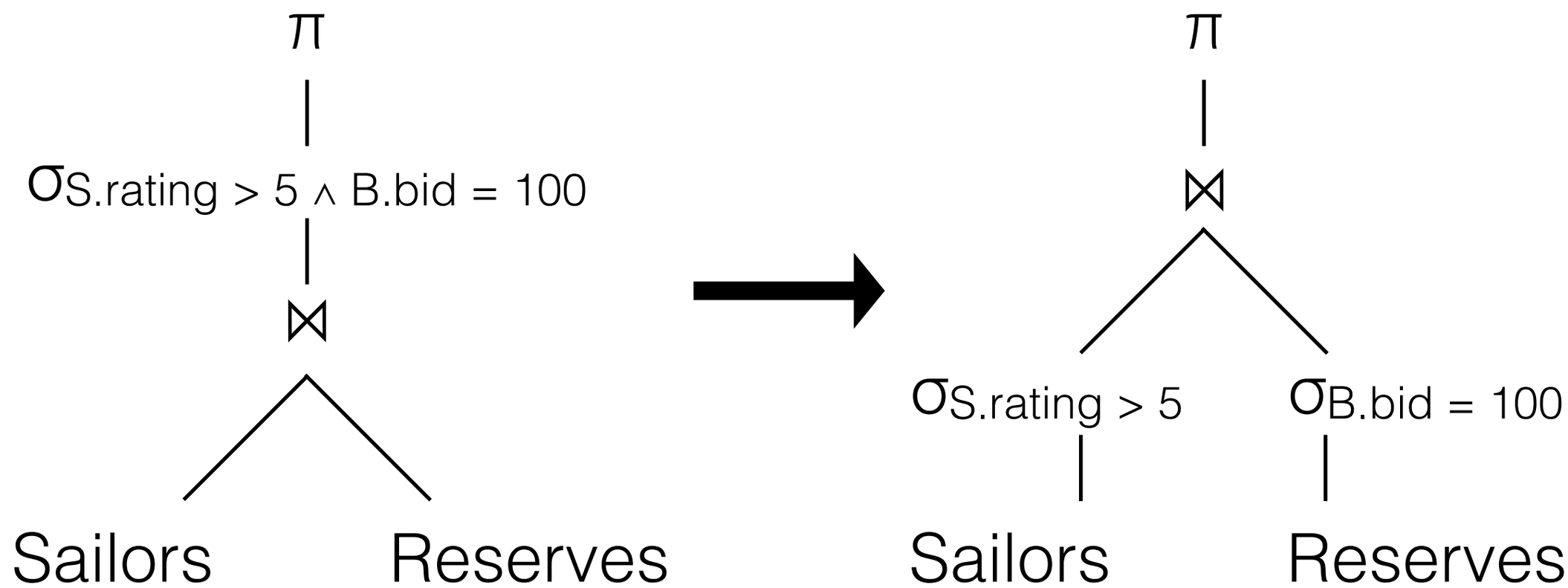
- Represent relational algebra with trees
- Order of operators affects IOs and resource usage

π...(σ...(Sailors ⋈... Reserves))



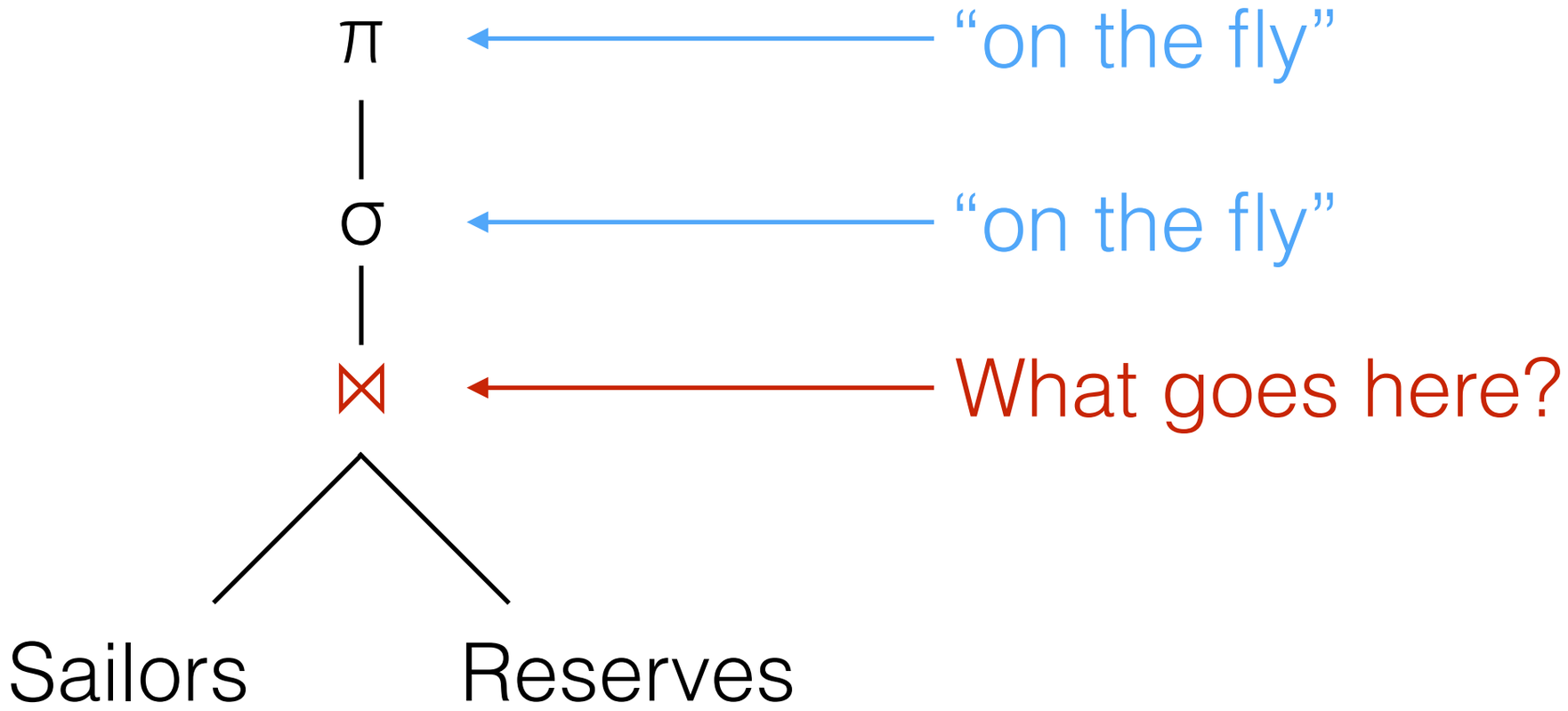
Alternate Plans

- Push selects/projects down (Why?)
- Use temp files and indexes (INLJ)



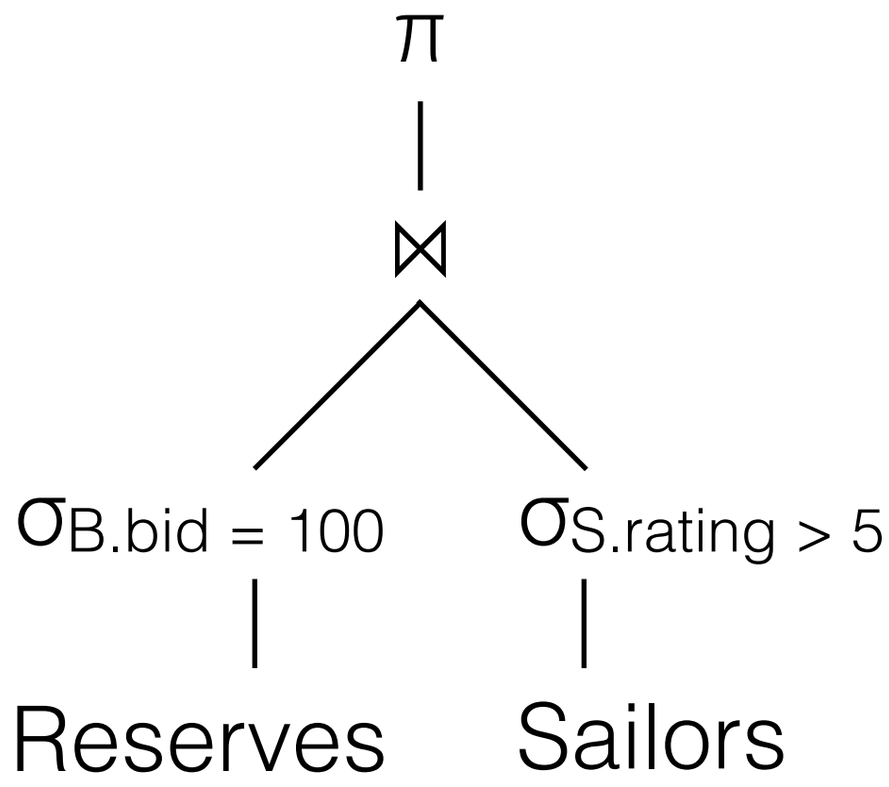
IO Costs

- Selection and projection are done “on the fly”



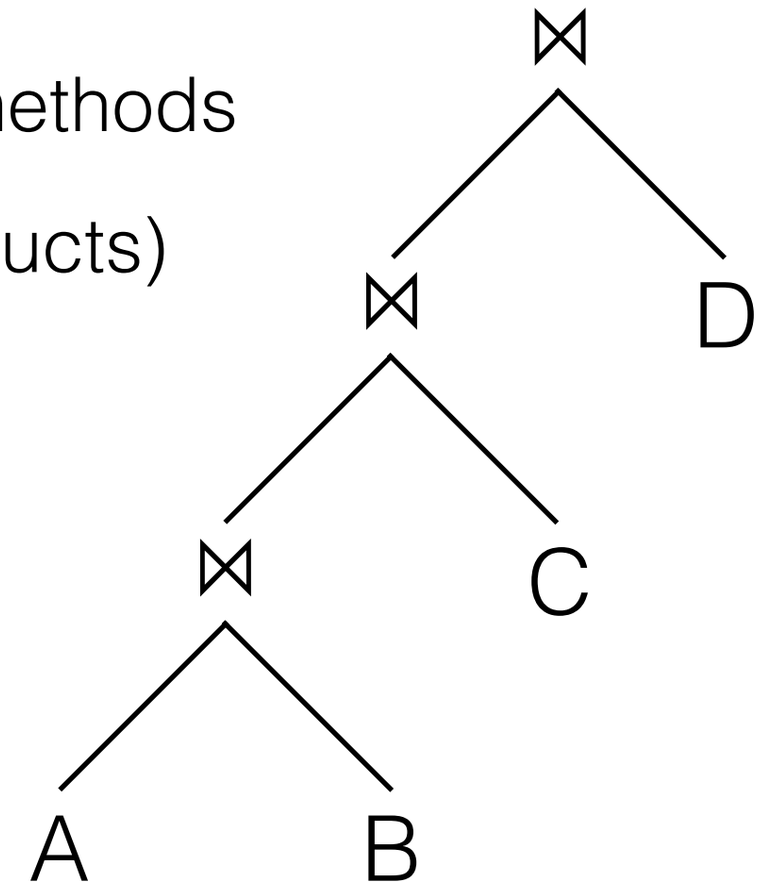
Quick Calculation Review

- From lecture:
  - B = 5
  - [R] = 1000, 100 boats
  - [S] = 500, 10 ratings
- Cost of a CNLJ?
  - Size of the temp file?



System R

1. Prune plan spaces
   - Only consider left-deep plans
     - But consider all join orders and methods
   - Ignore costly subtrees (avoid x products)
   - Push selections/projections
   - Handle interesting orders



System R

2. Make cost estimations for all operators
    - Use system catalog info and selectivity to estimate sizes
    - Cost = [# IOs] + CPU-factor * [# tuples]
    - Selectivity = |output| / |input|
        - Also called Reduction Factor (RF)
    - Result size = [# tuples] * [product of selectivities]
        - # tuples depends on join, cross product, etc.

RF Estimations

- Assuming *uniform distribution* and *independence*
- If *column = value*:

  RF = 1/NKeys(R)

- If *column1 = column2*:

  RF = 1/max(NKeys(R1), NKeys(R2))

- If *col > value*:

  RF = (High(I) - value)/(High(I) - Low(I) + 1)

System R

3. Search for cheapest plan
    - Dynamic Programming

Things to remember

- Cost is exponential in the number of tables
- Only match plans for join conditions, and after predicates have been applied
  - Avoid cross products
- Handle “interesting orders” as a final step
  - ORDER BY, GROUP BY, aggregates, downstream join attributes

Passes

- Pass 1: Best single-relation plan for each relation
- Pass 2: Best join between two single-relation plans
- Pass 3: Best join between a two-relation plan and a single-relation plan
- …
- Pass N: Best join between an (N – 1) relation plan with Nth relation

Single-Relation Estimates

- Equality selection on key of B+ tree T:
  - Cost = Height(T) + 1
- Multiple selects on clustered index:
  - Cost = (NPages(T) + NPages(R)) * [prod. of RFs]
- Multiple selects on unclustered index:
  - Cost = (NPages(T) + NTuples(R)) * [prod. of RFs]
- Sequential scan of file:
  - Cost = NPages(R)

Worksheet

- Kitties: (kid, cuteness [1-10], owner [10]); [K] = 100, |K| = 400
- Puppies: (pid, yappiness [1-10], owner [5]); [P] = 50, |P| = 200
- Humans: (hid, age [1-100]); [H] = 1000, |H| = 50000

- Unclustered tree on K.cuteness [5 pages]
- Unclustered tree on P.yappiness [5 pages]
- Clustered tree on (P.owner, P.yappiness) [15 pages]
- Unclustered tree on H.hid [20 pages]

- Join K, P, H
  - with predicates:
    - H.hid < 1200, P.yappiness = 7
    - K.owner = P.owner, P.owner = H.hid, P.yappiness = K.cuteness

3. Best Single-table Plans

- What info do we have for Humans H?
  - Humans: (hid, age [1-100]); [H] = 1000, |H| = 50000
  - Unclustered tree on H.hid [20 pages]
  - Predicates: H.hid < 1200

3. Best Single-table Plans

- What info do we have for Humans H?
  - Humans: (hid, age [1-100]); [H] = 1000, |H| = 50000
  - Unclustered tree on H.hid [20 pages]
  - Predicates: H.hid < 1200
- What plans can we use?

3. Best Single-table Plans

- What info do we have for Humans H?
  - Humans: (hid, age [1-100]); [H] = 1000, |H| = 50000
  - Unclustered tree on H.hid [20 pages]
  - Predicates: H.hid < 1200
- What plans can we use?
  - File Scan
  - Unclustered index + predicate!

3. Best Single-table Plans

- What info do we have for Humans H?
  - Humans: (hid, age [1-100]); [H] = 1000, |H| = 50000
  - Unclustered tree on H.hid [20 pages]
  - Predicates: H.hid < 1200

- What plans can we use?
  - File Scan = 1000 IOs
  - Unclustered index + predicate!
    - RF = |output| / |input| = 1200 / 50000
    - Cost = (NPages(T) + NTups(R)) * [prod. of RFs]
      = (20 + 50000) * 1200 / 50000 = 1200 IOs

3. Best Single-table Plans

- What do we have for Kitties K?
  - Kitties: (kid, cuteness [1-10], owner [10]);
    - [K] = 100, |K| = 400
  - Unclustered tree on K.cuteness [5 pages]
- Only one option… (why?)
  - File scan = 100 IOs

3. Best Single-table Plans

- What do we have for Puppies P?
  - Puppies: (pid, yappiness [1-10], owner [5]);
    - [P] = 50, |P| = 200
  - Unclustered tree on P.yappiness [5 pages]
  - Clustered tree on (P.owner, P.yappiness) [15 pages]
  - Predicates: P.yappiness = 7
- File scan
- Unclustered
- Clustered

3. Best Single-table Plans

- What do we have for Puppies P?
  - Puppies: (pid, yappiness [1-10], owner [5]);
    - [P] = 50, |P| = 200
  - Unclustered tree on P.yappiness [5 pages]
  - Clustered tree on (P.owner, P.yappiness) [15 pages]
  - Predicates: P.yappiness = 7
- File scan = 50 IOs
- Unclustered = 1/10 * (5 + 200) = 21 IOs
- Clustered = :(

4. Two-way joins

- What pairs are considered?

4. Two-way joins

- What pairings of relations are considered?
  - $K_{filescan} \bowtie P$
  - $P_{unclustered} \bowtie K$
  - $K_{filescan} \bowtie H$
  - $H_{filescan} \bowtie K$
  - $P_{unclustered} \bowtie H$
  - $H_{filescan} \bowtie P$

5. Two-way joins

- What pairings of relations are ignored?
  - $K_{filescan} \bowtie P$
  - $P_{unclustered} \bowtie K$
  - $K_{filescan} \bowtie H$
  - $H_{filescan} \bowtie K$
  - $P_{unclustered} \bowtie H$
  - $H_{filescan} \bowtie P$

5. Two-way joins

- What pairings of relations are ignored?
  - $K_{filescan} \bowtie P$
  - $P_{unclustered} \bowtie K$
  - $K_{filescan} \bowtie H$
  - $H_{filescan} \bowtie K$
  - $P_{unclustered} \bowtie H$
  - $H_{filescan} \bowtie P$

What pairs of relations are chosen?

6. Cost of an INLJ

- $P_{unclustered} \bowtie$ Kitties
  - Single-table predicate: P.yappiness = 7
  - Join predicates:
    - K.owner = P.owner
    - P.yappiness = K.cuteness
- Cost of an INLJ?
  - NPages(P) + NTuples(P) * (index lookup for K)

6. Cost of an INLJ

- $P_{unclustered} \bowtie$ Kitties
  - Single-table predicate: P.yappiness = 7
  - Join predicates:
    - K.owner = P.owner
    - P.yappiness = K.cuteness
- Cost of an INLJ?
  - NPages(P) + NTuples(P) * (index lookup for K)
  - 21 + 200/10 * [(5 + 400) * 1/10] = 831 IOs

7. Cost of an INLJ

- $Kitties_{filescan} \bowtie Puppies$
  - Join predicates:
    - K.owner = P.owner
    - P.yappiness = K.cuteness
  - Cost of an INLJ?
    - NPages(K) + NTuples(K) * (index lookup for P)

7. Cost of an INLJ

- $Kitties_{filescan} \bowtie Puppies$
  - Join predicates:
    - K.owner = P.owner
    - P.yappiness = K.cuteness
  - Cost of an INLJ?
    - NPages(K) + NTuples(K) * (index lookup for P)
    - 100 + 400 * [(15 + 50) * 1/10 * 1/10] = 500 IOs