Operator Types

• Stateful versus stateless operators
  – Select is stateless
  – Join is stateful

• Blocking versus non-blocking operators
  – Select is non-blocking
  – Agg functions are blocking

• Pipelined versus non-pipelined operators
  – Select is pipelinable
  – What about Join? (see next slide)
Join?

• Join: Revelation is that it depends on the implementation strategy chosen for an operator
  – Iteration-join: pipelinable
  – Merge-sort join: blocking
  – Index join: pipelinable
  – Hash join: blocking
Costing of a complete plan

• We went over an example query plan

• Important: first we classify operators as pipelined or not-pipelined
• If pipelined, then for stateless operators the IO cost is zero (for example, for Select or Project)
Costing of a Complete Query Plan

• What about a Select? How is it implemented?
• If in middle of plan, pipeline it (one tuple at a time iteration)
• If at leaf of plan, identify any potential index to use index-lookup to implement the Select
• *If index available, cost of implemention the select operator is equal to cost of an index lookup*
Costing of a complete plan

- Main idea:
  - Determine # of distinct values – \( V(R,a) \)
  - Determine physical impl. Strategies per operator
  - Then, compute IO costs for each operator
  - Then, sum up all costs.
  - Done.
How to generate that ‘good’ Physical Plan?

Many alternate search algorithms are possible:
1. Exhaustive listing of all possible plans
2. Dynamic programming
3. Branch and bound
4. Greedy bottom-up plan construction

NOTE: often only left-deep trees are being considered to keep the search space small.