Physical Optimization

Complete Example

Full Example

- Generating the process tree
- Physical optimization
  - Generation of alternatives
    - Data structures available
    - Alternative algorithms to execute an operation
  - Cost-based estimation per alternative
    - Cost formulas per access path and data structure available
      - Access path: all tuples, one tuple, several tuple
      - Data structures: table file, B+, clustered B+, hash index
    - The relevance of statistics in cost-based optimization
      - The database catalog
    - Estimation of statistics for the intermediate results
      - Selectivity factor (selections and joins)
      - Main hypotheses to propagate costs
        - Uniform distribution
        - Independence of attributes
  - Choose best alternative

Database Specification and Query

The tables have the following structures:

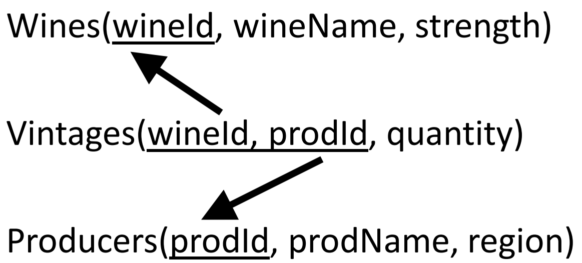
- Producers
  - Clustered by prodId
  - B^+ by region
- Wines
  - Clustered by wineId
- Vintages
  - Clustered by wineId and prodId

We have the following statistics (via database catalog):

- Tables (extra space due to being clustered needs to be added)
  - $|P|=10000$ $R_P=12$ $B_P=834$
  - $|W|= 5000$ $R_W=10$ $B_W=500$
  - $|V|= 100000$ $R_V=20$ $B_V=5000$
- Attributes
  - prodId, wineID and strength: length=5 bytes
  - ndist(region)=30
  - min(quantity)=10 max(quantity)=500
  - ndist(strength)=100

Moreover, we know that

- There are 500 useful bytes per disk block (use this to compute the size of intermediate results)
- Cost of accessing disk blocks is 1 second (D=1)
- Cost of CPU processing is negligible (C=0)
- The order of B-trees is 75 (d)
- The DBMS can use:
  - Block Nested Loops (with 6 memory pages, M=4)
  - Row Nested Loops
  - Sort Match (with 3 memory pages for sorting, M=2)



Wines(wineId, wineName, strength)

Vintages(wineId, prodId, quantity)

Producers(prodId, prodName, region)

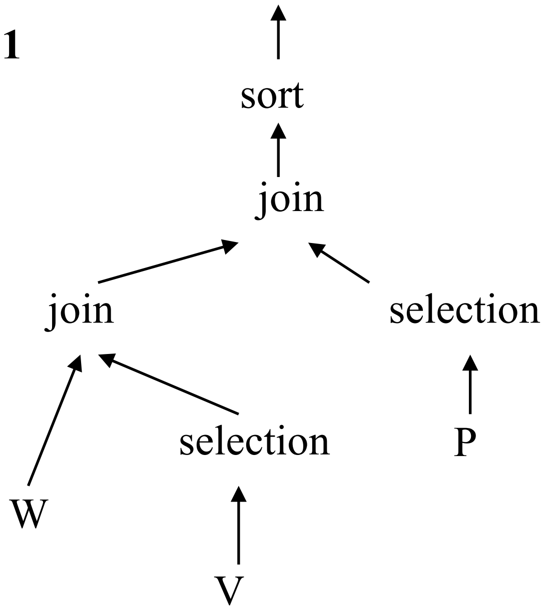
```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
          AND p.prodId=v.prodId
          AND p.region=”Priorat”
          AND v.quantity>100;
```

Input Process Trees

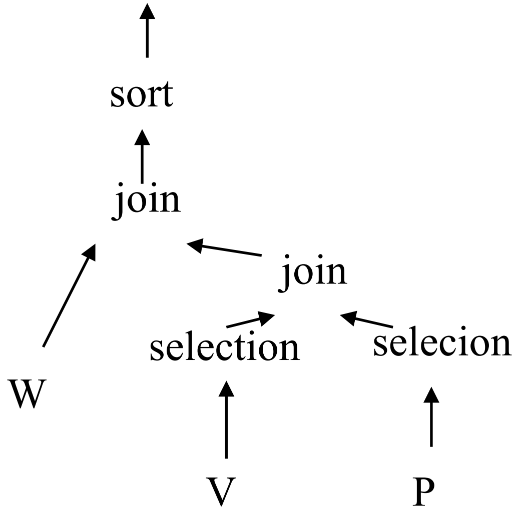
Process trees considered in this exercise:

```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
                AND p.prodId=v.prodId
                AND p.region=”Priorat”
                AND v.quantity>100;
```

PT1



PT2



ESTIMATION OF THE STATISTICS OF THE INTERMEDIATE RESULTS

Required statistics not available in the database catalog

Statistics of the Intermediate Results

```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
            AND p.prodId=v.prodId
            AND p.region="Priorat"
            AND v.quantity>100;
```

- □ **PT1/PT2** **(common for both trees)**
  - **Selection over V: V'**

    Record length V'= 5+5=10 bytes

    SF(quantity>100)=

    = (max(quantity)-100)/(max(quantity)-min(quantity)) =

    = 0.81632

    |V'|= SF*|V|= 0.81632*100,000= 81,632

    $R_{V'} = \lfloor 500/10 \rfloor$=50 records/block

    $B_{V'} = \lceil 81{,}632/50 \rceil$=1,633 blocks

  - **Selection over P: P'**

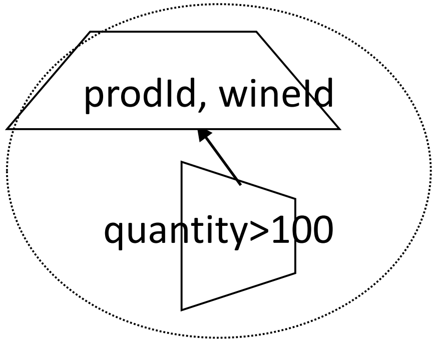
    Record length P'= 5 bytes

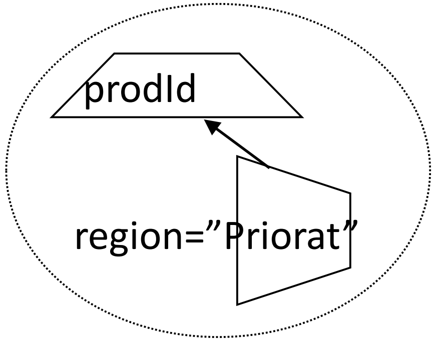
    SF(region="Priorat")=1/ndist(region)= 1/30

    |P'|= SF*|P|=10000/30=333

    $R_{P'} = \lfloor 500/5 \rfloor$=100 records/block

    $B_{P'} = \lceil 333/100 \rceil$=4 blocks





Statistics of the Intermediate Results

```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
          AND p.prodId=v.prodId
          AND p.region=”Priorat”
          AND v.quantity>100;
```

- **PT1**
  - **Join between W and V’: WV’**

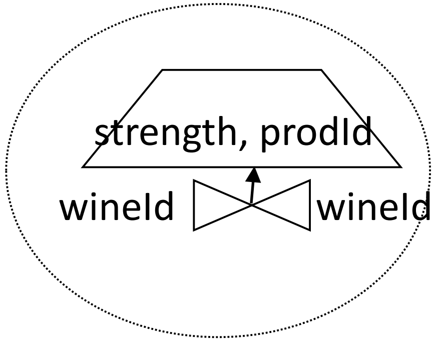
    Record length WV’ = 5+5 bytes

    SF = $1/|W| = 1/5000$

    $|WV'| = SF*|W|*|V'| = |V'| = 81{,}632$

    $R_{WV'} = \lfloor 500/10 \rfloor = 50$ records/block

    $B_{WV'} = \lceil 81{,}632/50 \rceil = 1{,}633$ blocks



  - **Join between WV’ and P’: WV’P’** (if quantity and region are independent)

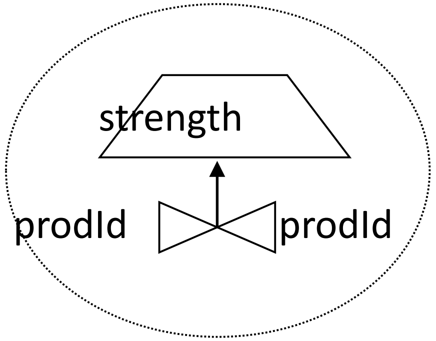
    Record length WV’P’ = 5 bytes

    $SF(WV'*P') = (1/|P'|)(333/10000) = 10^{-4}$

    $|WV'P'| = SF*|WV'|*|P'| = 10^{-4}*|WV'|*|P'| = 2{,}721$

    $R_{WV'P'} = \lfloor 500/5 \rfloor = 100$ records/block

    $B_{WV'P'} = \lceil 2721/100 \rceil = 28$ blocks



Statistics of the Intermediate Results

```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
          AND p.prodId=v.prodId
          AND p.region="Priorat"
          AND v.quantity>100;
```

□ **PT2**

- **Join between V' and P': V'P'** (if quantity and region independent)

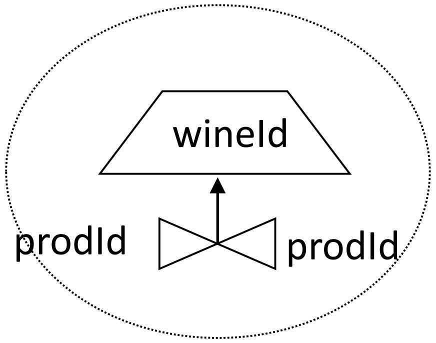
  Record length V'P' = 5 bytes

  $SF(V'*P') = (1/|P'|)(1/30) = 10^{-4}$

  $|V'P'| = SF*|V'|*|P'| = 10^{-4}*|V'|*|P'| = 2,721$

  $R_{V'P'} = \lfloor 500/5 \rfloor = 100$ records/block

  $B_{V'P'} = \lceil 2721/100 \rceil = 28$ blocks



- **Join between W and V'P': WV'P'**

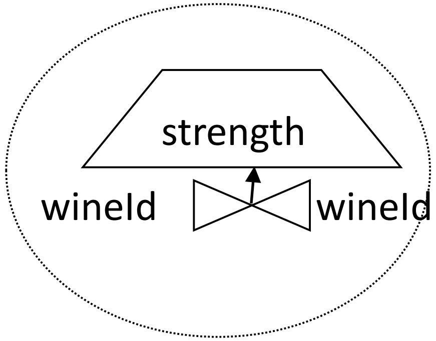
  Record length WV'P = 5 bytes

  $SF = 1/|W|$

  $|WV'P'| = FS*|W|*|V'P'| = |V'P'| = 2,721$

  $R_{WV'P'} = \lfloor 500/5 \rfloor = 100$ records/block

  $B_{WV'P'} = \lceil 2721/100 \rceil = 28$ blocks



Statistics of the Intermediate Results

```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
                AND p.prodId=v.prodId
                AND p.region=”Priorat”
                AND v.quantity>100;
```

❑ **PT1/PT2**

- **Final result: O**

  Record length O = 5 bytes

  |O| = ndist(strength) = 100

  $R_o = \lfloor 500/5 \rfloor = 100$ records/block

  $B_o = \lceil 100/100 \rceil = 1$ blocks

ESTIMATION OF COST PER OPERATION

Considering the available data structures, the access paths required and the alternative execution algorithms

Estimation of Cost Per Operation

```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
              AND p.prodId=v.prodId
              AND p.region=”Priorat”
              AND v.quantity>100;
```

- AP1/AP2
  - **Selection over V: V’**

    Available access paths: No index

    cost $_{Scan}(V')= \lceil 1.5B_V \rceil = \lceil 1.5*5{,}000 \rceil = 7{,}500$

    Choose Scan
  - **Selection over P: P’**

    Available access paths: B+ and No index

    cost $_{Scan}(P')= \lceil 1.5*B_P \rceil = \lceil 1.5*834 \rceil = 1{,}251$

    cost $_{B+}(P')=\lceil \log_{100}|P| \rceil - 1 + SF(region=\text{“Priorat”})*|P| + ((SF(region=\text{“Priorat”})*|P|-1)/100)$

    $=1+333+332/100=337$

    Choose B+
  - **Sort of WV’P’: O**

    $cost_{MergeSort}(O)=2B_{WV'P'}\cdot\lceil \log_M(B_{WV'P'})\rceil - B_{WV'P'}=2\cdot 28\cdot\lceil \log_2(28)\rceil - 28=252$

Estimation of Cost Per Operation

```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
              AND p.prodId=v.prodId
              AND p.region=”Priorat”
              AND v.quantity>100;
```

- **PT1**
  - **Join between W and V’: WV’**

    Available algorithms:

    Block Nested Loops

    $\lceil 1.5 \cdot B_W \rceil < B_{V'}$ (use commutative property of joins)

    cost $_{NestedLoop}$(WV’)= $\lceil 1.5B_W \rceil + \lceil 1.5B_W / M \rceil * B_{V'} =$

    $= \lceil 1.5*500 \rceil + \lceil 1.5*500/4 \rceil *1633 = 307{,}754$

    Row Nested Loops

    Yes, we do look for attributes of W

    V’ does not use extra space any more for being ordered

    cost $_{RowNestedLoops}$(WV’) $= B_{V'} + |V'|*(\lceil \log_{100}|W| \rceil - 1 + 1 + (1.5(k-1)/10)) =$

    $= 1{,}633+81{,}632*(\lceil \log_{100}5{,}000 \rceil - 1+1) = 164{,}897$

    Sort-Match

    W is ordered by wineID, V’ is still ordered by wineId and prodId

    cost $_{SortMatch}$(WV’) $= \lceil 1.5B_W \rceil + B_{V'} = \lceil 1.5*500 \rceil + 1{,}633 = 2{,}383$

    Choose Sort-Match

Even if shown for completeness, in this course you are not expected to be capable of computing the cost of each join algorithm

Estimation of Cost Per Operation

```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
              AND p.prodId=v.prodId
              AND p.region=”Priorat”
              AND v.quantity>100;
```

- **PT1**
  - **Join between WV’ and P’: WV’P’**

    Available algorithms:

    Block Nested Loops

    $B_{P'} < B_{WV'}$ (use commutative property of joins)

    cost $_{NestedLoop}$(WV’P’) = $B_{P'} + \lceil B_{P'} /M \rceil * B_{WV'} = 4 + \lceil 4/4 \rceil * 1{,}633 = 1{,}637$

    Sort Match

    Neither WV’ nor P’ are ordered by prodId

    cost $_{SortMatch}$(WV’P’) $= 2*B_{WV'}*\lceil \log_2 B_{WV'} \rceil + 2*B_{P'}*\lceil \log_2 B_{P'} \rceil + B_{WV'} + B_{P'} =$

    $= 2*1{,}633*11 + 2*4*2 + 1633 + 4 = 37{,}579$

    Choose Nested Loops

Even if shown for completeness, in this course you are not expected to be capable of computing the cost of each join algorithm

Estimation of Cost Per Operation

```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
            AND p.prodId=v.prodId
            AND p.region=”Priorat”
            AND v.quantity>100;
```

- **PT2**
  - **Join between V’ y P’: V’P’**

    Available algorithms:

    Block Nested Loops

    $B_{P'} < B_{V'}$ (use commutative property of joins)

    $\text{cost}_{\text{NestedLoop}}(V'P') = B_{P'} + \lceil B_{P'}/M \rceil * B_{V'} = 4 + \lceil 4/4 \rceil * 1{,}633 = 1{,}637$

    Sort Match

    Neither V’ nor P’ are ordered by prodId

    $\text{cost}_{\text{SortMatch}}(V'P') = 2*B_{V'}*\lceil \log_2 B_{V'} \rceil + 2*B_{P'}*\lceil \log_2 B_{P'} \rceil + B_{V'} + B_{P'} =$

    $= 2*1{,}633*11 + 2*4*2 + 1{,}633 + 4 = 37{,}579$

    Choose Nested Loops

Even if shown for completeness, in this course you are not expected to be capable of computing the cost of each join algorithm

Estimation of Cost Per Operation

- Phase 1: Alternatives generation
- Phase 2: Intermediate results estimation
- **Phase 3: Cost estimation for operation**
- Phase 4: Choose the best option

```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
              AND p.prodId=v.prodId
              AND p.region="Priorat"
              AND v.quantity>100;
```

- □ **PT2**
  - **Join between W y V'P': WV'P'**

    Available algorithms:

    Block Nested Loops

    $B_{V'P'} < \lceil 1.5B_W \rceil$ (use commutative property of joins)

    cost $_{NestedLoop}$(WV'P')= $B_{V'P'} + \lceil B_{V'P'} / M \rceil * \lceil 1.5B_W \rceil = 28 + \lceil 28/4 \rceil * \lceil 1.5*500 \rceil = 5278$

    Row Nested Loops

    Yes, we look for attributes of W

    cost $_{RowNestedLoops}$(WV'P')= $B_{V'P'} + |V'P'| * (\lceil \log_{100}|W| \rceil -1 + 1 + (1.5(k-1)/10)) =$

    $= 28+2,721* (\lceil \log_{100}5,000 \rceil -1+1) = 5,470$

    Sort-Match

    W is sorted by wineId, V'P' is not sorted by wineId

    cost $_{SortMatch}$(WV'P') $= 2B_{V'P'} \lceil \log_2 B_{V'P'} \rceil + \lceil 1.5B_W \rceil + B_{V'P'} =$

    $= 2*28* \lceil \log_2 28 \rceil + \lceil 1.5*500 \rceil + 28 = 1,058$

    Choose Sort-Match

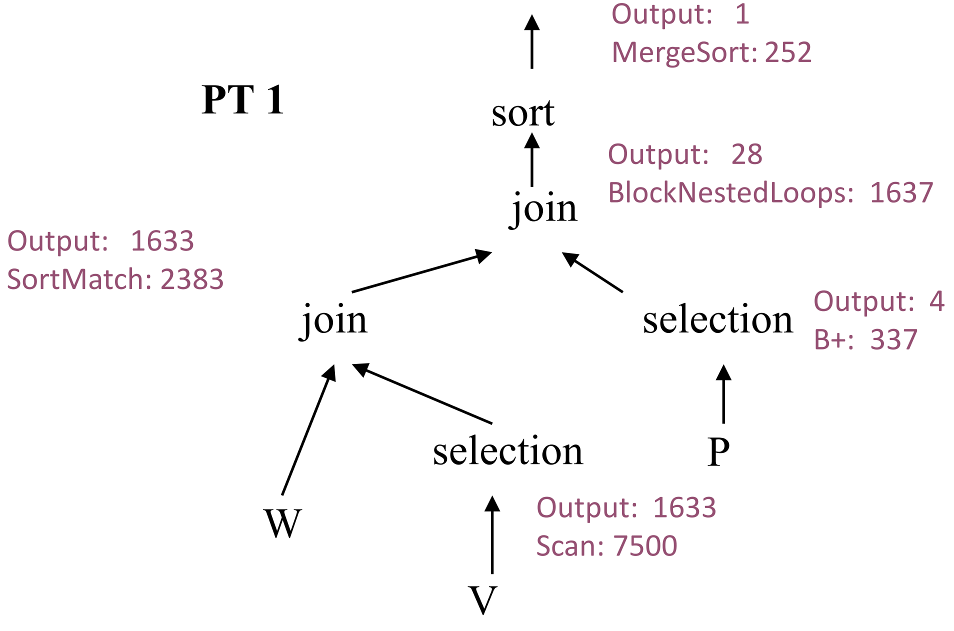
Even if shown for completeness, in this course you are not expected to be capable of computing the cost of each join algorithm

FINAL CHOICE

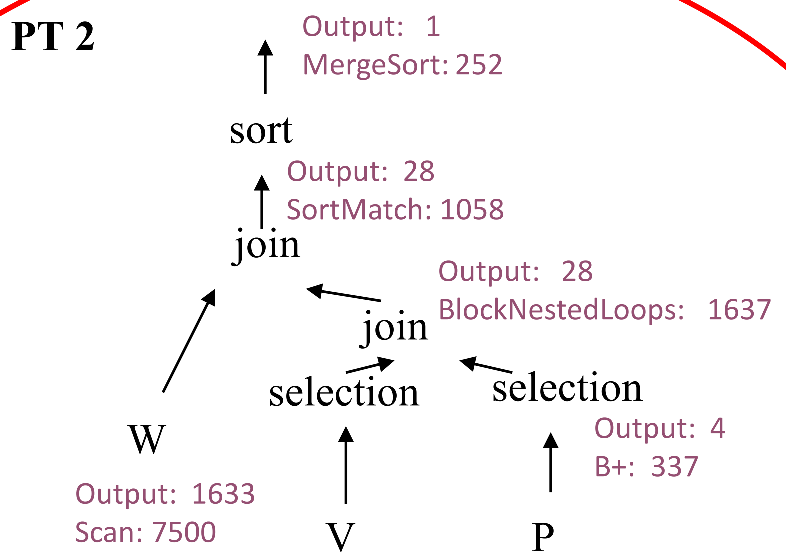
Alternative with the cheapest cost

Best Alternative

```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
            AND p.prodId=v.prodId
            AND p.region="Priorat"
            AND v.quantity>100;
```



Cost$_{PT1}$: 15,408



Cost$_{PT2}$: 12,478