Example of physical optimization (I)

- The tables have the following structures:
  - Producers
    - Clustered by prodId
    - B^+ by region
  - Wines
    - Clustered by wineId
  - Vintages
    - Clustered by wineId and prodId

Wines(<u>wineId</u>, wineName, strength)

Vintages(<u>wineId, prodId</u>, quantity)

Producers(<u>prodId</u>, prodName, region)

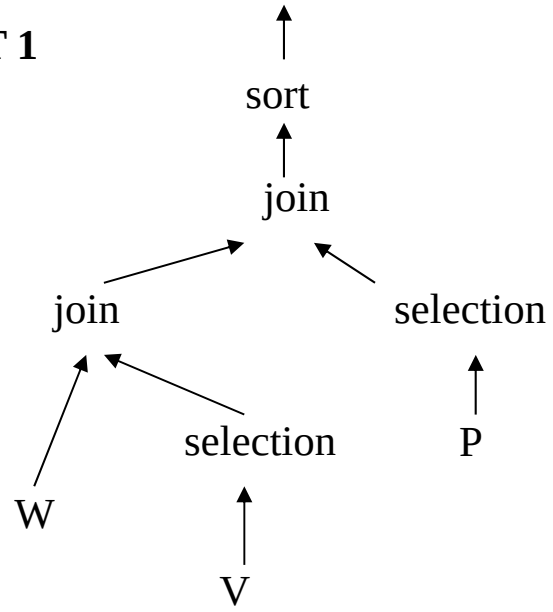
- We have the following statistics:
  - Tables (extra space due to being clustered needs to be added)
    - $|P|=10000$ $R_P=12$ $B_P=834$
    - $|W|=5000$ $R_W=10$ $B_W=500$
    - $|V|=100000$ $R_V=20$ $B_V=5000$
  - Attributes
    - prodId, wineID and strength: length=5 bytes
    - ndist(region)=30
    - min(quantity)=10 max(quantity)=500
    - ndist(strength)=100
- Moreover, we know that
  - There are 500 useful bytes per intermediate disk block
  - Each table is in a different file (there is no Clustered Structure)
  - 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
  - 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)
  - We will not change the order of operations coming from syntactic optimization

Example of physical optimization (II)

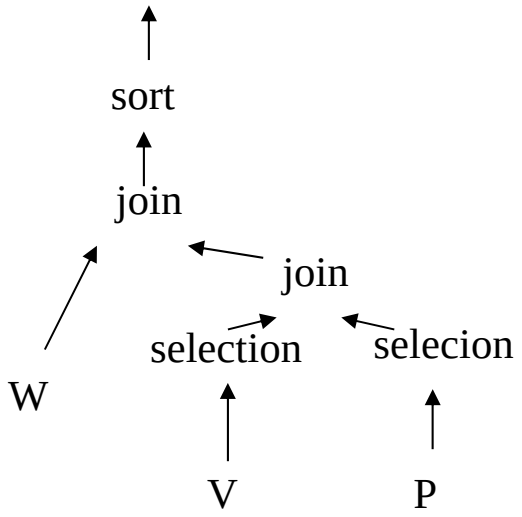
- **Phase 1: Alternatives generation**
- Phase 2: Intermediate results estimation
- Phase 3: Cost estimation for each algorithm
- Phase 4: Choose the best option

```
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;
```

PT 1



PT 2



Example of physical optimization (III)

- Phase 1: Alternatives generation
- **Phase 2: Intermediate results estimation**
- Phase 3: Cost estimation for each algorithm
- 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;
```

PT1/PT2

- **Selection over V: V'**

  Record length V'= 5+5=10 bytes

  SF(quantity>100)=

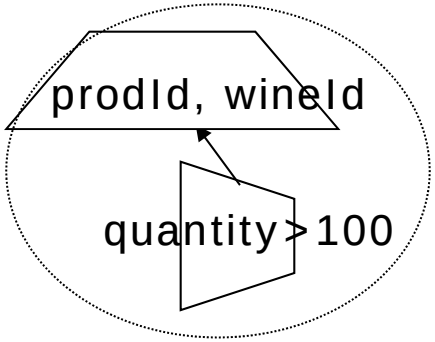
  = (max(quantity)-100)/(max(quantity)-min(quantity)) =

  = 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'**

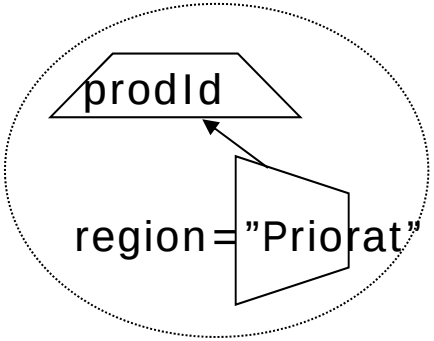
  Record length P'= 5 bytes

  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



Example of physical optimization (IV)

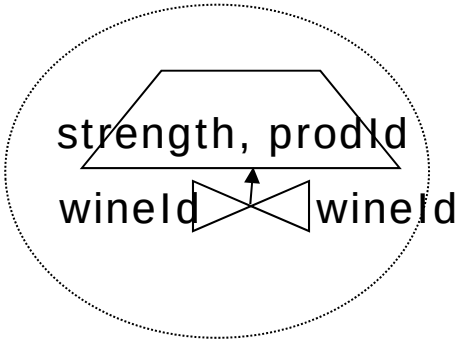
- Phase 1: Alternatives generation
- **Phase 2: Intermediate results estimation**
- Phase 3: Cost estimation for each algorithm
- Phase 4: Choose the best option

```
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;
```

PT1

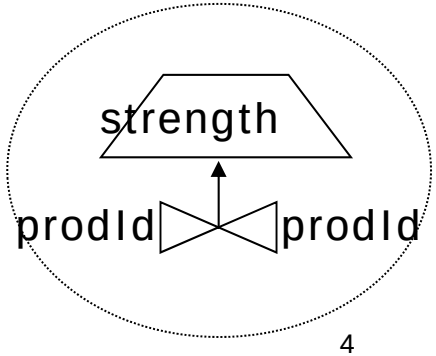
- **Join between W and V': WV'**

  Record length WV' = 5+5 bytes
  SF = 1/|W| = 1/5000
  |WV'| = SF*|W|*|V'|= |V'|= 81,632
  $R_{WV'}$ = ⌊500/10⌋ = 50 records/block
  $B_{WV'}$ = ⌈81,632/50⌉ = 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'|)*(1/30)= 10^{-4}
  |WV'P'| = SF*|WV'|*|P'|= 10^{-4}*|WV'|*|P'|= 2,721
  $R_{WV'P'}$ = ⌊500/5⌋ = 100 records/block
  $B_{WV'P'}$ = ⌈2721/100⌉ = 28 blocks



Example of physical optimization (V)

- Phase 1: Alternatives generation
- **Phase 2: Intermediate results estimation**
- Phase 3: Cost estimation for each algorithm
- 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 V' and P': V'P'** (if quantity and region independent)

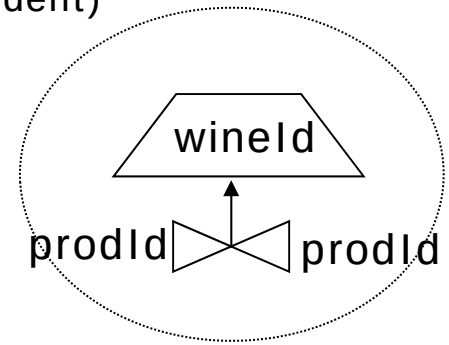
    Record length V'P' = 5 bytes

    SF(V'*P') = (1/30)*(1/|P'|) = 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'**

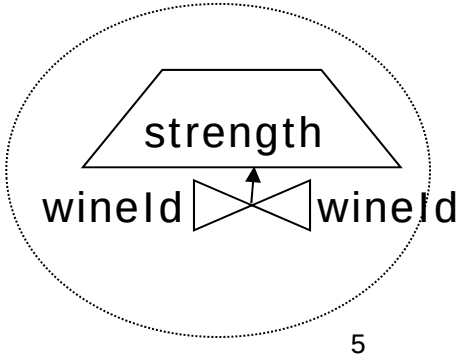
    Record length WV'P = 5 bytes

    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



Example of physical optimization (VI)

- Phase 1: Alternatives generation
- **Phase 2: Intermediate results estimation**
- Phase 3: Cost estimation for each algorithm
- 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;
```

- **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

Example of physical optimization (VII)

- Phase 1: Alternatives generation
- Phase 2: Intermediate results estimation
- **Phase 3: Cost estimation for each algorithm**
- 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;
```

- 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

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

    $cost_{B+}(P') = \lceil \log_{100}|P| \rceil - 1 + SF(region="Priorat")*|P| + ((SF(region="Priorat")*|P|-1)/100)$

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

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

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

Example of physical optimization (VIII)

- Phase 1: Alternatives generation
- Phase 2: Intermediate results estimation
- **Phase 3: Cost estimation for each algorithm**
- 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;
```

- **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

Example of physical optimization (IX)

- Phase 1: Alternatives generation
- Phase 2: Intermediate results estimation
- **Phase 3: Cost estimation for each algorithm**
- 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;
```

- **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

Example of physical optimization (X)

- Phase 1: Alternatives generation
- Phase 2: Intermediate results estimation
- **Phase 3: Cost estimation for each algorithm**
- 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 V' y P': V'P'**

    Available algorithms:

    Block Nested Loops

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

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

    Sort Match

    Neither V' nor P' are ordered by prodId

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

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

    Choose Nested Loops

Example of physical optimization (XI)

- Phase 1: Alternatives generation
- Phase 2: Intermediate results estimation
- **Phase 3: Cost estimation for each algorithm**
- 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

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

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

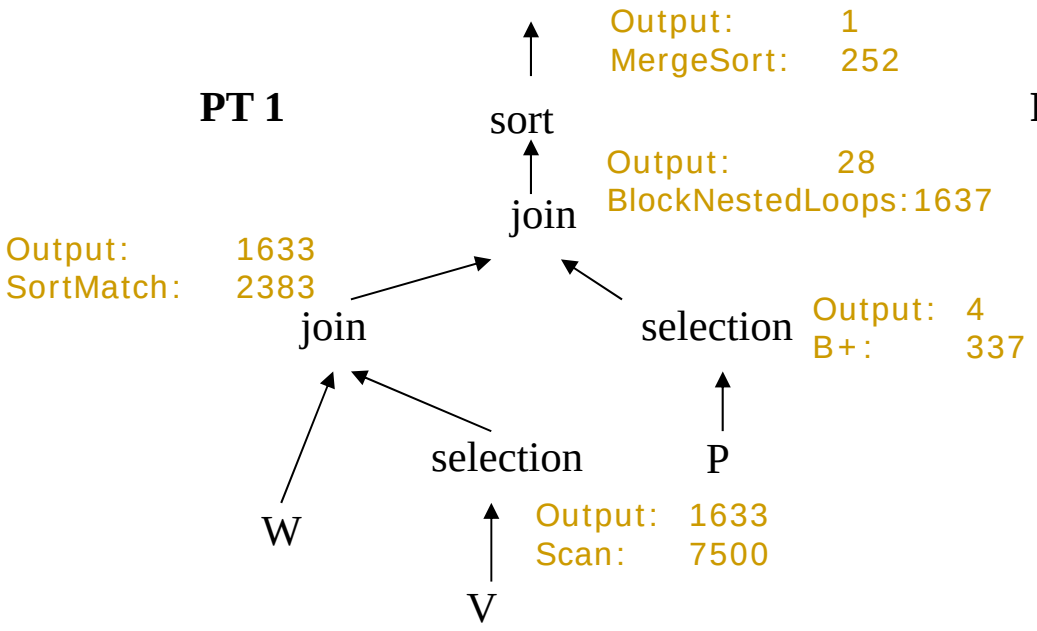
    Choose Sort-Match

Example of physical optimization (XII)

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

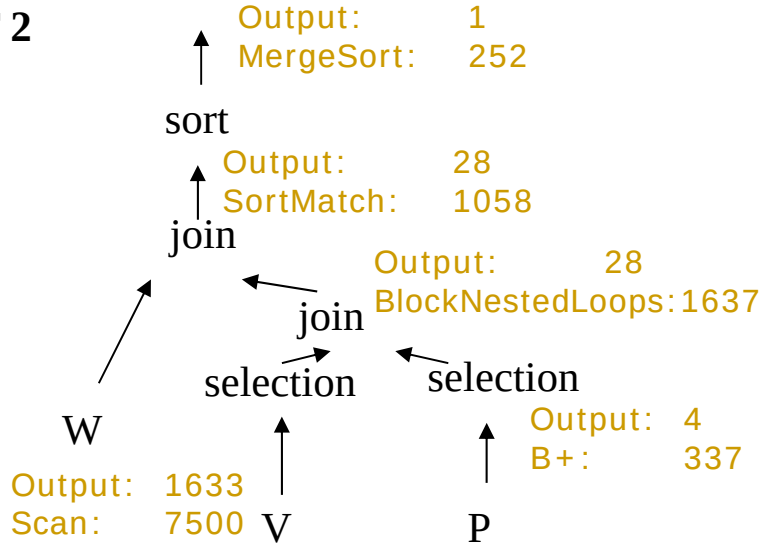
```
SELECT DISTINCT w.strength
FROM wines w, producers p, vintages v
WHERE v.wineId=w.wineId
        AND p.prodId=v.prodId
        AND p.region="Priorat"
        AND v.quantity>100;
```

PT 1



$Cost_{PT1}$: 15,408

PT 2



$Cost_{PT2}$: 12,478