**Class 18: Joins I**

**Summary**

<table>
<thead>
<tr>
<th>Tables</th>
<th>M</th>
<th>(N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows per page</td>
<td>Pr</td>
<td></td>
</tr>
</tbody>
</table>

**Selection**

(a) i) cheapest access path  
  ii) retrieve tuples  
  iii) apply remaining selection conditions  

(b) i) get rid of all matching indexes  
  ii) intersect them  
  iii) retrieve tuples & apply remaining sel. cond.

**Projection**

Sort + discard unwanted fields & duplicates  
hash + 

Cost: \( M + 2 \cdot T \leftarrow \text{pages after removing unwanted fields} \)

**Joins**

Nested-Loop Joins  
Sort-Merge Joins  
Hash Joins  
Remaining op (joins + agg)
SELECT * FROM R, S WHERE R.sid = S.sid
R \times S \text{ discuss as } \# I/Os \text{ discard output}

- Simple Nested-Loop Join

\[
\begin{align*}
\forall r & \in R \text{ - outer} \\
\forall s & \in S \text{ - inner} \\
& \text{ if } r_i = s_j \text{ then add } \langle r, s \rangle \text{ to the result}
\end{align*}
\]

\text{Cost}

\[
(M \cdot P_R) \cdot N + M = \left(1000 \cdot 100 \right) \cdot 500 + 1000 = 50,001,000 \text{ \# I/Os}
\]

\[
\begin{align*}
\# \text{rows of } R & : 1 \text{ I/O } \rightarrow 2 \text{ms} \\
R & : M=1000 \rightarrow 4 \text{MB} \\
S & : N=500 \rightarrow 2 \text{MB}
\end{align*}
\]

Swap R with S

\[
(N \cdot P_S) \cdot M + N = \left(40,000 \cdot 500 \right) \text{ I/Os}
\]

- Page-oriented Nested-Loop Join

\[
\begin{align*}
\forall \text{ page } br & \text{ in } R \\
\forall \text{ page } bs & \text{ in } S \\
\forall \text{ tuple } r & \text{ in } br \\
\forall \text{ tuple } s & \text{ in } bs \\
& \text{ if } r_i = s_j \text{ then add } \langle r, s \rangle \text{ to the result}
\end{align*}
\]
\[
\text{Cost}
\]
\[
M \times N + M = 1000 \times 500 + 1000 = 501,000 \rightarrow 17 \text{ min}
\]

smaller outer?
\[
N \times M + N = 500 \times 1000 + 500 = 500,500
\]

\underline{Index Nested Loop Join}

\[\text{tuple } r \text{ in } R\]

probe index to fetch \( s \) such that \( s_i = v_j \)

add \( L_r, s \) to result

\underline{Cost}

\[
M + M \cdot Pr \cdot \text{cost of finding matching tuples through the index}
\]

\[
\rightarrow \text{Hash index 1.2 1 I/Os}
\]

\[
\rightarrow \text{B}^*\text{-Tree 2-4 1 I/Os}
\]

Clustered \( \rightarrow 1 \) I/O per page of matching tuples

Unclustered \( \rightarrow 1 \) I/O per matching tuple

**Example 1**: hash idx on sid of \( S \)

Scan \( R \): (\( M \))

\[\text{At each tuple in } R\]

fetch data entry (1, 2)

goto file (1)

\[
M + M \cdot Pr \cdot (1.2 + 1) \rightarrow 1000 \times 100 \times (2.2) = 221,000 \rightarrow 17 \text{ min}
\]
Example 2: hash idx on sid of \( R \)

\[
\frac{M \cdot R_b}{N \cdot P_S} = \frac{100k}{40k} = 2.5 \text{ res. s} / \text{file}
\]

Scan \( S \) \((N)\)

\( S \) probe hash idx \((1,2)\)

Find matching tuples \( 2.5 \)

\( N = N \cdot P_S (1.2 + 2.5) = 500 + 500 \cdot 80 \cdot (3.7) = 148,500 \Rightarrow 5 \text{ min} \)

Block Nested Loop Joins

\( \rightarrow \) 1 page for streaming the inner \( S \)

1 page for output

\( K \) pages for holding blocks of \( R \) of outer \( R \)

\( \uparrow \) block of \( K \) pages of \( R \)

\( \uparrow \) page \( b_S \) in \( S \)

\( \uparrow \) tuple \( r \) in \( K \) pages of \( R \)

\( \uparrow \) tuple \( s \) in \( b_S \)

(If \( r_i = S_j \) add \( r_i, s \) to the result)

Cost

Scan outer \( R \): \( M \) I/0s

Scan inner for each block of \( R \)

\[
M + \frac{M}{K} \cdot N \rightarrow 1000 + \frac{500 \cdot 1000}{K} \text{ \( \leftarrow 100 \text{ pages} \)}
\]

\[
N + \frac{N}{K} \cdot M \rightarrow 500 + \frac{500 \cdot 1000}{K} \text{ \( \rightarrow 5500 \rightarrow 11 \text{ sec} \)}
\]
Sort - Merge Join

- Both sorted on the join attribute
  - Useful: 1) both or one relations sorted on join attr
  - 2) output should be sorted on join attr.
- Many duplicates may lead to backtracking

Cost: \( \text{Sort } R + \text{Sort } S + M + N \)

Worst case? \( MN \) if all is equal

Cost: \( (M+N) \cdot 2 \cdot \# \text{passes} + M + N \)

2 passes?

\[
\sqrt[2]{\frac{N}{B}} = B - 1 \approx \frac{N}{B} = B - 1 \Rightarrow B^2 - B - N = 0
\]

\[
B = \sqrt{N} + 1 = 33
\]

\[
\text{Cost} = (M+N) \cdot s = 1500 \cdot s = 7500 \text{ I/Os} \Rightarrow 159 \text{ sec}
\]

BNLJ w/ 33 buffers

\[
M + \frac{M-N}{B} = 1000 + \frac{500 \cdot 1000}{33} \leq \frac{1000 + 15151}{33}
\]

\[
N + \frac{M \cdot N}{K} = 500 + \frac{500 \cdot 1000}{33} \leq \frac{500 + 15151}{33}
\]

if \( h = 100 \) SMJ cannot do better than \( 7500 \)

BNLJ will do as low as \( 5500 \)
Refined Sort-Merge Join

assume \( B > \sqrt{M} \) and \( B > \sqrt{N} \)

after pass 0

\[
R \to \frac{M}{B} \text{ runs } \quad B > \sqrt{M} \Rightarrow \frac{1}{B} < \frac{1}{\sqrt{M}} \Rightarrow \frac{M}{B} < \sqrt{M} < B
\]

\[
S \to \frac{N}{B} \text{ runs } \quad B > \sqrt{N} \Rightarrow \frac{N}{B} < \sqrt{N} < B
\]

after pass 0 either \( R, S \) \( \# \) runs \( \leq B \)

consider using replacement sort

it results to runs with size \( \leq 2B \)

\# sorted runs after pass 0 using replacement sort

\[
R \to \frac{M}{2B} < \frac{B}{2} \quad S \to \frac{N}{2B} < \frac{B}{2}
\]

we allocate a buffer per sorted run per file

\[
\text{cost} = (M+N)^3
\]

Read \( R \to \text{writing } \leq B/2 \# \text{ runs of } R \quad 2M \)

Read \( S \to \text{writing } \leq B/2 \# \text{ runs of } S \quad 2N \)

Read \( R \) and \( S \) and merge on the fly: \( M+N \)

\[
(M+N)^3 = 4500 \text{ I/Os} \implies 9S
\]