Query Optimization
Exercise Session 3

Bernhard Radke

November 21
select *
from lineitem l, orders o, customers c
where l.l_orderkey=o.o_orderkey
    and o.o_custkey=c.c_custkey
    and c.c_name='Customer#000014993'.
Homework: Task 2

We know $|R_1|$, $|R_2|$, domains of $R_1.x$, $R_2.y$,
We know $|R_1|$, $|R_2|$, domains of $R_1.x$, $R_2.y$, (that is, $|R_1.x|$, $|R_2.y|$), and whether $x$ and $y$ are keys or not. The selectivity of $\sigma_{R_1.x=c}$ is...

- if $x$ is the key:
We know $|R_1|$, $|R_2|$, domains of $R_1.x$, $R_2.y$, (that is, $|R_1.x|$, $|R_2.y|$), and whether $x$ and $y$ are keys or not. The selectivity of $\sigma_{R_1.x=c}$ is...

- if $x$ is the key: $\frac{1}{|R_1|}$
We know $|R_1|, |R_2|$, domains of $R_1.x, R_2.y$, (that is, $|R_1.x|, |R_2.y|$), and whether $x$ and $y$ are keys or not. The selectivity of $\sigma_{R_1.x=c}$ is...

- if $x$ is the key: $\frac{1}{|R_1|}$
- if $x$ is not the key:
Homework: Task 2

We know $|R1|$, $|R2|$, domains of $R1.x$, $R2.y$, (that is, $|R1.x|$, $|R2.y|$), and whether $x$ and $y$ are keys or not.

The selectivity of $\sigma_{R1.x=c}$ is...

- if $x$ is the key: $\frac{1}{|R1|}$
- if $x$ is not the key: $\frac{1}{|R1.x|}$
We know $|R1|$, $|R2|$, $|R1.x|$, $|R2.y|$, and whether $x$ and $y$ are keys or not.
First, the size of $R1 \times R2$ is
Homework: Task 2

We know \(|R1|, |R2|, |R1.x|, |R2.y|\), and whether \(x\) and \(y\) are keys or not.
First, the size of \(R1 \times R2\) is \(|R1||R2|\)
The selectivity of \(\bowtie_{R1.x=R2.y}\) is...
  ▶ if both \(x\) and \(y\) are the keys:
We know $|R_1|$, $|R_2|$, $|R_1.x|$, $|R_2.y|$, and whether $x$ and $y$ are keys or not.

First, the size of $R_1 \times R_2$ is $|R_1||R_2|

The selectivity of $\bowtie_{R_1.x=R_2.y}$ is...

- if both $x$ and $y$ are the keys: $\frac{1}{\max(|R_1|,|R_2|)}$
We know $|R_1|$, $|R_2|$, $|R_1.x|$, $|R_2.y|$, and whether $x$ and $y$ are keys or not.
First, the size of $R_1 \times R_2$ is $|R_1||R_2|
The selectivity of $\bowtie_{R_1.x=R_2.y}$ is...

- if both $x$ and $y$ are the keys: $\frac{1}{\max(|R_1|,|R_2|)}$
- if only $x$ is the key:
We know $|R_1|$, $|R_2|$, $|R_1.x|$, $|R_2.y|$, and whether $x$ and $y$ are keys or not.
First, the size of $R_1 \times R_2$ is $|R_1||R_2|$
The selectivity of $\bowtie_{R_1.x=R_2.y}$ is...
- if both $x$ and $y$ are the keys: $\frac{1}{\max(|R_1|,|R_2|)}$
- if only $x$ is the key: $\frac{1}{|R_1|}$
Homework: Task 2

We know $|R_1|$, $|R_2|$, $|R_1.x|$, $|R_2.y|$, and whether $x$ and $y$ are keys or not.
First, the size of $R_1 \times R_2$ is $|R_1||R_2|$
The selectivity of $\bowtie_{R_1.x=R_2.y}$ is...

- if both $x$ and $y$ are the keys: $\frac{1}{\max(|R_1|,|R_2|)}$
- if only $x$ is the key: $\frac{1}{|R_1|}$
- if both $x$ and $y$ are not the keys:
We know $|R_1|$, $|R_2|$, $|R_1.x|$, $|R_2.y|$, and whether $x$ and $y$ are keys or not.

First, the size of $R_1 \times R_2$ is $|R_1||R_2|

The selectivity of $\bowtie_{R_1.x=R_2.y}$ is...

- if both $x$ and $y$ are the keys: $\frac{1}{\max(|R_1|,|R_2|)}$
- if only $x$ is the key: $\frac{1}{|R_1|}$
- if both $x$ and $y$ are not the keys: $\frac{1}{\max(|R_1.x|,|R_2.y|)}$
Selectivity estimation

We know $|R_1|$, $\max(R_1.x)$, $\min(R_1.x)$, $R_1.x$ is arithmetic.

The selectivity of $\sigma_{R_1.x > c}$ is
Selectivity estimation

We know $|R_1|$, $\max(R_1.x)$, $\min(R_1.x)$, $R_1.x$ is arithmetic.

The selectivity of $\sigma_{R_1.x > c}$ is $\frac{\max(R_1.x) - c}{\max(R_1.x) - \min(R_1.x)}$. 
Selectivity estimation

We know $|R1|$, $\max(R1.x)$, $\min(R1.x)$, $R1.x$ is arithmetic.

The selectivity of $\sigma_{R1.x>c}$ is

$$\frac{\max(R1.x) - c}{\max(R1.x) - \min(R1.x)}$$

The selectivity of $\sigma_{c1<R1.x<c2}$ is
Selectivity estimation

We know $|R_1|$, $\max(R_1.x)$, $\min(R_1.x)$, $R_1.x$ is arithmetic.

The selectivity of $\sigma_{R_1.x > c}$ is $\frac{\max(R_1.x) - c}{\max(R_1.x) - \min(R_1.x)}$

The selectivity of $\sigma_{c_1 < R_1.x < c_2}$ is $\frac{c_2 - c_1}{\max - \min}$
Homework: Task 3

- $|R| = 1,000$ pages, $|S| = 100,000$ pages
- 1 page - 50 tuples, 1 block - 100 pages
- avg. access = 10 ms, transfer speed = 10,000 pages/sec
- Time for blocked nested loops join = ?
Homework: Task 3

- $|R| = 1,000$ pages, $|S| = 100,000$ pages
- 1 page - 50 tuples, 1 block - 100 pages
- avg. access = 10 ms, transfer speed = 10,000 pages/sec
- Time for blocked nested loops join = ?
- choose left argument: $R$ vs. $S$, $\frac{1,000}{100}$ vs. $\frac{100,000}{100}$ $\Rightarrow R$
- Time to read one block:
  \[ T_b = \text{avg}.\text{seek} + \left(100 \frac{1}{\text{transfer speed}}\right) = 0.02s \]
Homework: Task 3

- Time to read one block:
  \[ T_b = \text{avg.seek} + \left(100 \cdot \frac{1}{\text{transfer speed}}\right) = 0.02s \]
- Read 1 block from \( R \), join it with \( S \):
  \[ T_b + \text{time to read } S \approx 10s \]
Time to read one block:

\[ T_b = \text{avg.seek} + \left(100 \frac{1}{\text{transfer speed}}\right) = 0.02s \]

Read 1 block from \( R \), join it with \( S \):

\[ T_b + \text{time to read } S \approx 10s \]

Repeat it for every block in \( R \):

\[ T_{BNLJ} = \frac{\#\text{pages in } R}{\text{block size}} (10s) \approx 100s \]
Greedy operator ordering

- take the query graph
- find relations $R_1, R_2$ such that $|R_1 \Join R_2|$ is minimal and merge them into one node
- repeat until the query graph has more than one node

Generates bushy trees!
Example

\[ R_1(10) \rightarrow 0.8 \rightarrow R_2(10) \rightarrow 0.5 \rightarrow R_3(10) \rightarrow 0.3 \rightarrow R_4(10) \]

\[ R_9(10) \rightarrow 0.3 \rightarrow R_6(10) \rightarrow 0.2 \rightarrow R_5(10) \]

\[ R_8(10) \rightarrow 0.3 \rightarrow R_7(10) \]
Example - step 1

R_1(10) → 0.8 → R_2(10) → 0.5 → R_3(10) → 0.3 → R_4(10)

R_9(10) → 0.3 → R_6(10) → 0.6 → R_5(10)

R_8(10) → 0.3 → R_7(10)
Example - after step 1

\[
\begin{array}{c}
R_1(10) \quad 0.8 \quad R_2(10) \quad 0.5 \quad R_3(10) \\
| \\
0.6 \\
| \\
R_9(10) \quad 0.3 \quad R_5 \times R_6(20) \\
| \\
0.6 \\
| \\
R_8(10) \quad 0.3 \quad R_7(10) \\
| \\
0.3 \\
| \\
R_4(10) \\
\end{array}
\]
Example - step 2

\[ R_1(10) \xrightarrow{0.8} R_2(10) \xrightarrow{0.5} R_3(10) \xrightarrow{0.3} R_4(10) \]
\[ R_9(10) \xrightarrow{0.3} R_5 \times R_6(20) \]
\[ R_8(10) \xrightarrow{0.3} R_7(10) \]

0.8, 0.5, 0.3, 0.6, 0.3, 0.7, 0.54, 0.6, 0.3, 0.6, 0.3, 0.7, 0.54
Example- after step 2

\[ R_1(10) \xrightarrow{0.8} R_2(10) \xrightarrow{0.5} R_3 \ltimes R_4(30) \]

\[ R_9(10) \xrightarrow{0.3} R_5 \ltimes R_6(20) \xrightarrow{0.7} 0.54 \]

\[ R_7 \ltimes R_8(30) \]
Example - step 3

\[ R_1(10) \overset{0.8}{\longrightarrow} R_2(10) \overset{0.5}{\longrightarrow} R_3 \rtimes R_4(30) \]

\[ R_9(10) \overset{0.3}{\longrightarrow} R_5 \rtimes R_6(20) \]

\[ R_7 \rtimes R_8(30) \]
Example - after step 3

\[
R_1(10) \stackrel{0.8}{\rightarrow} R_2(10) \stackrel{0.5}{\rightarrow} R_3 \bowtie R_4(30)
\]

\[
(R_5 \bowtie R_6) \bowtie R_9(60)
\]

\[
R_7 \bowtie R_8(30)
\]
Example - step 4

\[ R_1(10) \quad 0.8 \quad R_2(10) \quad 0.5 \quad R_3 \bowtie R_4(30) \]

\[ (R_5 \bowtie R_6) \bowtie R_9(60) \]

\[ R_7 \bowtie R_8(30) \]
Example - after step 4

\[ R_1 \bowtie R_2(80) \overset{0.5}{\longrightarrow} R_3 \bowtie R_4(30) \]

\[ (R_5 \bowtie R_6) \bowtie R_9(60) \]

\[ R_7 \bowtie R_8(30) \]
Example - step 5

\[ R_1 \ltimes R_2(80) \; \begin{array}{c} 0.5 \end{array} \; R_3 \ltimes R_4(30) \]

\[ (R_5 \ltimes R_6) \ltimes R_9(60) \]

\[ R_7 \ltimes R_8(30) \]
Example - after step 5

\[(R_1 \Join R_2) \Join (R_3 \Join R_4)(1200)\]

\[\downarrow 0.2268\]

\[(R_7 \Join R_8) \Join ((R_5 \Join R_6) \Join R_9)(1080)\]
Example - result
IKKBZ (informally)

Query graph $Q$ is acyclic. Pick a root node, turn it into a tree. Run the following procedure for every root node, select the cheapest plan

Input: rooted tree $Q$

1. if the tree is a single chain, stop
2. find the subtree (rooted at $r$) all of whose children are chains
3. normalize, if $c_1 \rightarrow c_2$, but $\text{rank}(c_1) > \text{rank}(c_2)$ in the subtree rooted at $r$
4. merge chains in the subtree rooted at $r$, rank is ascending
5. repeat 1
For every relation $R_i$ we keep

- cardinality $n_i$
- selectivity $s_i$ — the selectivity of the incoming edge from the parent of $R_i$
- cost $C(R_i) = n_i s_i$ (or 0, if $R_i$ is the root)
- rank $r_i = \frac{n_i s_i - 1}{n_i s_i}$

Moreover,

- $C(S_1 S_2) = C(S_1) + T(S_1) C(S_2)$
- $T(S) = \prod_{R_i \in S} (s_i n_i)$
- rank of a sequence $r(S) = \frac{T(S) - 1}{C(S)}$
Understanding IKKBZ

- what is the rank?
- when is \((R_1 \Join R_2) \Join R_3\) cheaper than \((R_1 \Join R_3) \Join R_2\)?
Understanding IKKBZ

- what is the rank?
- when is \( (R_1 \bowtie R_2) \bowtie R_3 \) cheaper than \( (R_1 \bowtie R_3) \bowtie R_2 \)?
- if \( r(R_2) < r(R_3) \)!
IKKBZ - example

\[
\begin{array}{c}
R_1 \\
\frac{1}{5} & \frac{1}{3} \\
R_2 & R_3 \\
\frac{1}{10} & 1 \\
R_4 & R_5
\end{array}
\]

<table>
<thead>
<tr>
<th>Relation</th>
<th>n</th>
<th>s</th>
<th>C</th>
<th>T</th>
<th>rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>20</td>
<td>$\frac{1}{5}$</td>
<td>4</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>$\frac{1}{3}$</td>
<td>10</td>
<td>10</td>
<td>$\frac{9}{10}$</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>$\frac{1}{10}$</td>
<td>5</td>
<td>5</td>
<td>$\frac{4}{5}$</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>$\frac{1}{2}$</td>
</tr>
</tbody>
</table>

IKKBZ - example

Subtree $R_3$: merging, $\text{rank}(R_5) < \text{rank}(R_4)$

$$
\begin{array}{c}
R_1 \\
| \\
R_2 \\
| \\
R_3 \\
|
R_5 \\
| \\
R_4
\end{array}
$$

<table>
<thead>
<tr>
<th>Relation</th>
<th>n</th>
<th>s</th>
<th>C</th>
<th>T</th>
<th>rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>20</td>
<td>$\frac{1}{5}$</td>
<td>4</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>$\frac{1}{3}$</td>
<td>10</td>
<td>10</td>
<td>$\frac{9}{10}$</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>$\frac{1}{10}$</td>
<td>5</td>
<td>5</td>
<td>$\frac{4}{5}$</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>$\frac{1}{2}$</td>
</tr>
</tbody>
</table>
Subtree $R_1$:
$\text{rank}(R_3) > \text{rank}(R_5)$, normalizing

$$R_1 \xrightarrow{R_3,5} R_2 \xrightarrow{R_4}$$

<table>
<thead>
<tr>
<th>Relation</th>
<th>n</th>
<th>s</th>
<th>C</th>
<th>T</th>
<th>rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>20</td>
<td>$\frac{1}{5}$</td>
<td>4</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>$\frac{1}{3}$</td>
<td>10</td>
<td>10</td>
<td>$\frac{9}{10}$</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>$\frac{1}{10}$</td>
<td>5</td>
<td>5</td>
<td>$\frac{4}{5}$</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>3,5</td>
<td>60</td>
<td>$\frac{1}{3}$</td>
<td>30</td>
<td>20</td>
<td>$\frac{19}{30}$</td>
</tr>
</tbody>
</table>
Subtree $R_1$: merging

<table>
<thead>
<tr>
<th>Relation</th>
<th>n</th>
<th>s</th>
<th>C</th>
<th>T</th>
<th>rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>20</td>
<td>$\frac{1}{5}$</td>
<td>4</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>$\frac{1}{15}$</td>
<td>10</td>
<td>10</td>
<td>$\frac{9}{10}$</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>$\frac{1}{10}$</td>
<td>5</td>
<td>5</td>
<td>$\frac{4}{5}$</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>3,5</td>
<td>60</td>
<td>$\frac{1}{15}$</td>
<td>30</td>
<td>20</td>
<td>$\frac{19}{30}$</td>
</tr>
</tbody>
</table>
IKKBZ - example

### Denormalizing

\[
\begin{align*}
R_1 & \\
R_3 & \\
R_5 & \\
R_2 & \\
R_4 & \\
\end{align*}
\]

<table>
<thead>
<tr>
<th>Relation</th>
<th>n</th>
<th>s</th>
<th>C</th>
<th>T</th>
<th>rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>20</td>
<td>(\frac{1}{5})</td>
<td>4</td>
<td>4</td>
<td>(\frac{3}{4})</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>(\frac{1}{15})</td>
<td>10</td>
<td>10</td>
<td>(\frac{9}{10})</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>(\frac{1}{10})</td>
<td>5</td>
<td>5</td>
<td>(\frac{4}{5})</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>(\frac{1}{2})</td>
</tr>
<tr>
<td>3,5</td>
<td>60</td>
<td>(\frac{1}{3})</td>
<td>30</td>
<td>20</td>
<td>(\frac{3}{30})</td>
</tr>
</tbody>
</table>
IKKBZ - another example

$R_1 \quad \frac{1}{6}$

$R_2 \quad \frac{1}{10}$

$R_3 \quad \frac{1}{20} \quad R_4 \quad \frac{3}{4} \quad R_5 \quad \frac{1}{2} \quad R_6 \quad \frac{1}{14} \quad R_7$

$R_8 \quad \frac{1}{5}$

$R_9 \quad \frac{1}{25}$

$|R_1| = 30$

$|R_2| = 100$

$|R_3| = 30$

$|R_4| = 20$

$|R_5| = 10$

$|R_6| = 20$

$|R_7| = 70$

$|R_8| = 100$

$|R_9| = 100$
\[
\begin{align*}
\r(R_2) &= \frac{9}{10} = 0.9 \\
\r(R_3) &= \frac{4}{5} = 0.8 \\
\r(R_4) &= 0 \\
\r(R_5) &= \frac{13}{15} \approx 0.86 \\
\r(R_6) &= \frac{9}{10} = 0.9 \\
\r(R_7) &= \frac{4}{5} = 0.8 \\
\r(R_8) &= \frac{19}{20} = 0.95 \\
\r(R_9) &= \frac{3}{4} = 0.75
\end{align*}
\]
\[ C(R_{8,9}) = 100 \]
\[ T(R_{8,9}) = 80 \]
\[ r(R_{8,9}) = \frac{79}{100} = 0.79 \]
\[ C(R_{6,7}) = 60 \]
\[ T(R_{6,7}) = 50 \]
\[ r(R_{6,7}) = \frac{49}{60} \approx 0.816 \]
\[
\begin{align*}
C(R_{8,9}) & = 100 \\
T(R_{8,9}) & = 80 \\
r(R_{8,9}) & = \frac{79}{100} = 0.79 \\
R_{8,9} & \ \\
C(R_{6,7}) & = 60 \\
T(R_{6,7}) & = 50 \\
r(R_{6,7}) & = \frac{49}{60} \approx 0.816
\end{align*}
\]
IKKBZ

\begin{itemize}
\item \( r(R_{8,9}) < r(R_{6,7}) \)
\end{itemize}
$r(R_4) = 0$

$\Rightarrow r(R_5) = \frac{13}{15} \approx 0.86$

$\Rightarrow r(R_{8,9}) = \frac{79}{100} = 0.79$

$\Rightarrow r(R_{6,7}) = \frac{49}{60} \approx 0.81$
\[ r(R_5) = \frac{13}{15} \approx 0.86 \]
\[ r(R_{8,9}) = 0.79 \]
\[ n_{5,8,9} = 800 \]
\[ C_{5,8,9} = \frac{1515}{2} \]
\[ T_{5,8,9} = 600 \]
\[ r(R_{5,8,9}) = \frac{1198}{1515} \approx 0.79 \]
\[ r(R_{6,7}) \approx 0.816 \]
\[ r(R_2) = \frac{9}{10} \]
\[ r(R_3) = 0.8 \]
\[ r(R_4) = 0 \]
\[ r(R_{5,8,9}) = \frac{1198}{1515} \approx 0.79 \]
\[ r(R_{6,7}) \approx 0.816 \]
$R_1 \rightarrow R_3 \rightarrow R_4 \rightarrow R_{5,8,9} \rightarrow R_{6,7} \rightarrow R_2$
IKKBZ

$R_1 \quad R_3 \quad R_4 \quad R_5 \quad R_8 \quad R_9 \quad R_6 \quad R_7 \quad R_2$
IKKBZ-based heuristics

What if $Q$ has cycles?

- Observation 1: the answer of the query, corresponding to any subgraph of the query graph, is a superset of the answer to the original query.
- Observation 2: a very selective join is more likely to be influential in choosing the order than a non-selective join.
IKKBZ-based heuristics

What if $Q$ has cycles?

- Observation 1: the answer of the query, corresponding to any subgraph of the query graph, is a superset of the answer to the original query
- Observation 2: a very selective join is more likely to be influential in choosing the order than a non-selective join

Choose the minimum spanning tree (minimize the product of the edge weights), compute the total order, compute the original query.
Homework: Task 1 (10 points)

Selectivity estimation continues...

- Our estimations (prev. homework) are far from perfect
- Construct specific examples (database schema, concrete instances of relations and selections/joins), where our estimations are very "bad"
- "Bad" – means that for some queries (give examples of SQL queries) the logical plan will be suboptimal (w.r.t $C_{out}$), if we use these estimations
- In other words, bad estimations mislead the optimizer and it outputs a clearly suboptimal plan
- Two examples (one for selections, one for joins)
Homework: Task 2 (5 points)

- Give an example query instance where the optimal join tree (using $C_{out}$) is bushy and includes a cross product.
- Note: the query graph should be connected!
Homework: Task 3 (15 points)

- Using the program from the first exercise as a basis, implement a program that
  - parses SQL queries
  - translates them into tinydb execution plans
  - and executes the query.

- Note: a canonical translation of the joins is fine, but push all predicates of the form attr = const down to the base relations.
Info

- Slides and exercises: http://db.in.tum.de/teaching/ws1617/queryopt/
- Send any questions, comments, solutions to exercises etc. to radke@in.tum.de
- Exercises due: 9 AM, November 21