Query Optimization

Exercise Session 3

Bernhard Radke

November 21

Homework: Task 1

```
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 $|R1|$, $|R2|$, domains of $R1.x$, $R2.y$,

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:

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: $\frac{1}{|R1|}$

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:

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: $\frac{1}{|R1.x|}$

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

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:

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: $\frac{1}{max(|R1|,|R2|)}$

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: $\frac{1}{max(|R1|,|R2|)}$
- if only x is the key:

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: $\frac{1}{max(|R1|,|R2|)}$
- if only x is the key: $\frac{1}{|R1|}$

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: $\frac{1}{max(|R1|,|R2|)}$
- if only x is the key: $\frac{1}{|R1|}$
- if both x and y are not the keys:

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: $\frac{1}{max(|R1|,|R2|)}$
- if only x is the key: $\frac{1}{|R1|}$
- if both x and y are not the keys: $\frac{1}{max(|R1.x|,|R2.y|)}$

Selectivity estimation

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

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

Selectivity estimation

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

The selectivity of $\sigma_{R1.x>c}$ is $\frac{max(R1.x)-c}{max(R1.x)-min(R1.x)}$

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 $|R1|$, $max(R1.x)$, $min(R1.x)$, $R1.x$ 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 $\frac{c2-c1}{max-min}$

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. $\frac{100{,}000}{100} \Rightarrow R$

Homework: Task 3

- Time to read one block:
  $T_b = avg.seek + (100 \frac{1}{\text{transfer speed}}) = 0.02s$

Homework: Task 3

- Time to read one block:
  $T_b = avg.seek + (100\frac{1}{\text{transfer speed}}) = 0.02s$
- Read 1 block from R, join it with S:

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

Homework: Task 3

- Time to read one block:
  $T_b = avg.seek + (100\frac{1}{\text{transfer speed}}) = 0.02s$
- 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:

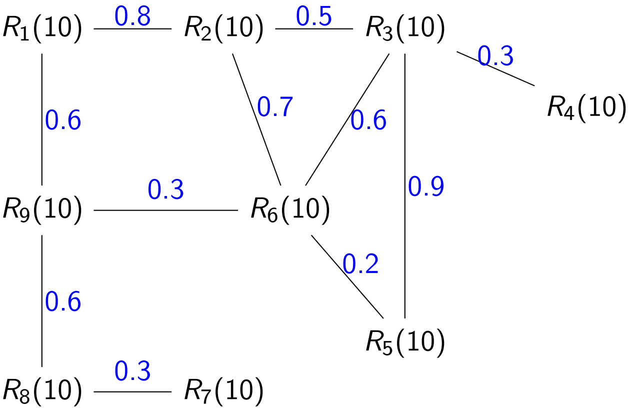
$$T_{BNLJ} = \frac{\text{\#pages in R}}{\text{block size}}(10s) \approx 100s$$

Greedy operator ordering

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

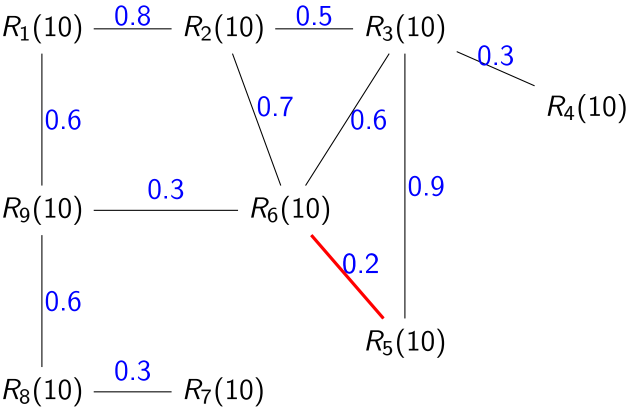
Generates bushy trees!

Example

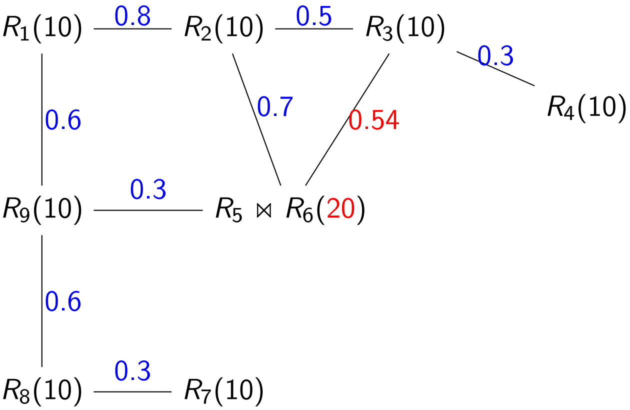


$R_1(10)$
0.8
$R_2(10)$
0.5
$R_3(10)$
0.3
$R_4(10)$
0.6
0.7
0.6
0.9
$R_9(10)$
0.3
$R_6(10)$
0.2
0.6
$R_5(10)$
$R_8(10)$
0.3
$R_7(10)$

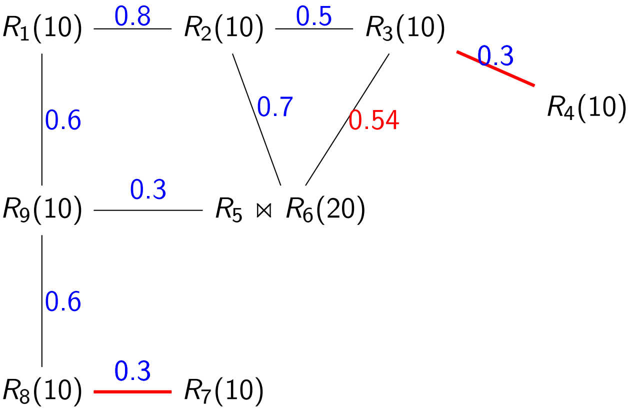
Example - step 1



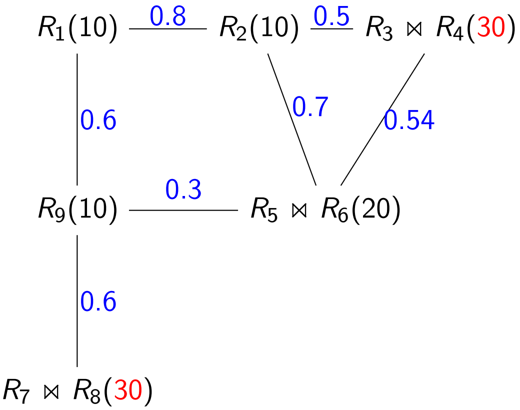
Example - after step 1



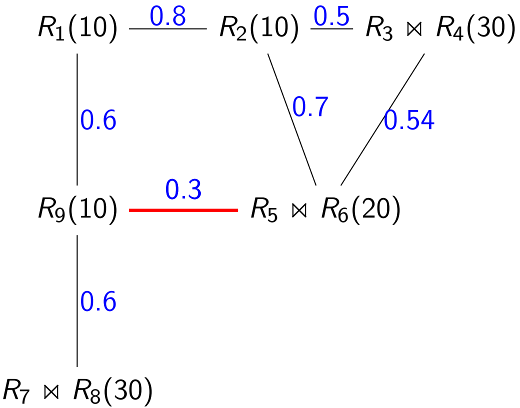
Example - step 2



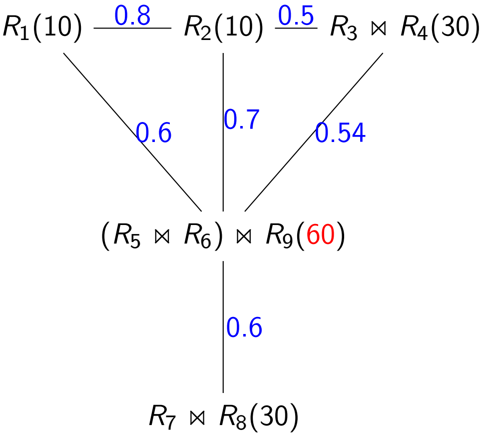
Example- after step 2



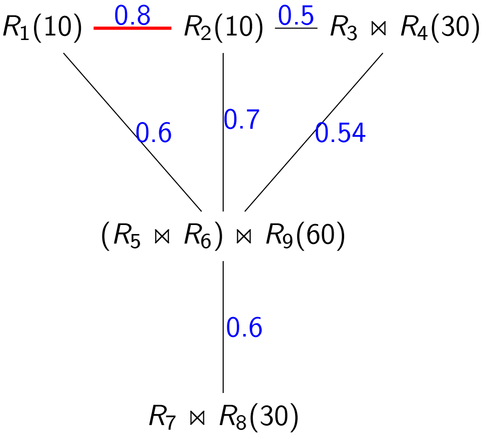
Example - step 3



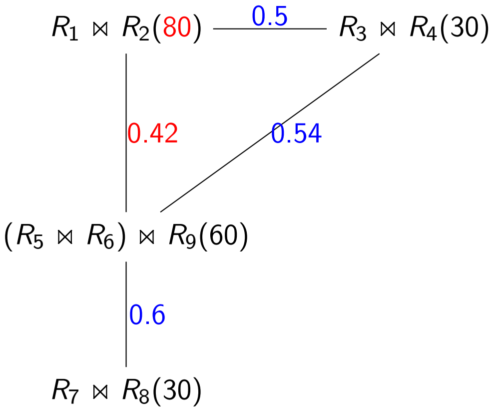
Example - after step 3



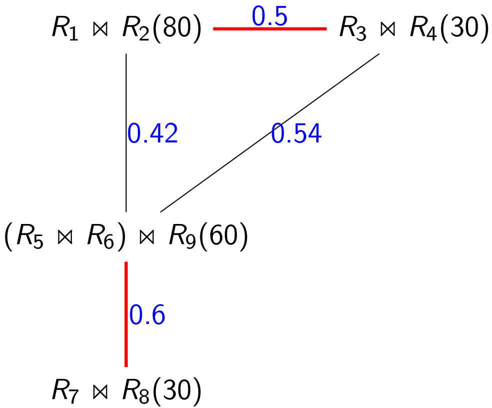
Example - step 4



Example - after step 4



Example - step 5



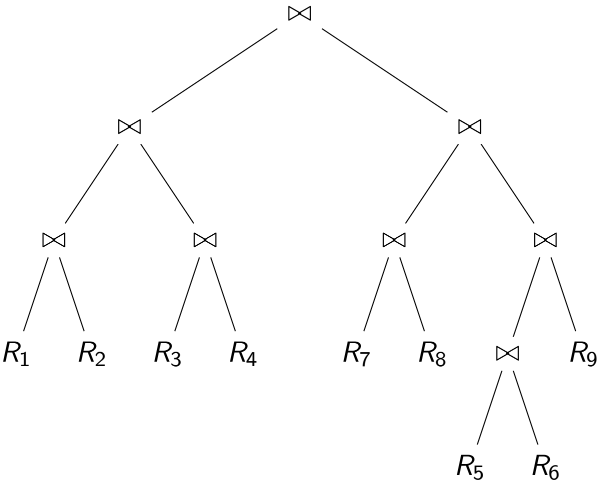
Example - after step 5

$$(R_1 \bowtie R_2) \bowtie (R_3 \bowtie R_4)(1200)$$

0.2268

$$(R_7 \bowtie R_8) \bowtie ((R_5 \bowtie R_6) \bowtie R_9)(1080)$$

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 $rank(c_1) > rank(c_2)$ in the subtree rooted at r
4. merge chains in the subtree rooted at r, rank is ascending
5. repeat 1

IKKBZ (informally)

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 $r(S) = \frac{T(S) - 1}{C(S)}$

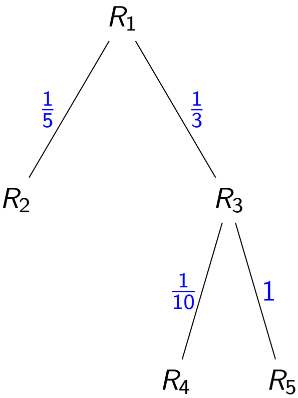
Understanding IKKBZ

- what is the rank?
- when is $(R_1 \bowtie R_2) \bowtie R_3$ cheaper than $(R_1 \bowtie R_3) \bowtie R_2$?

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 $r(R_2) < r(R_3)$!

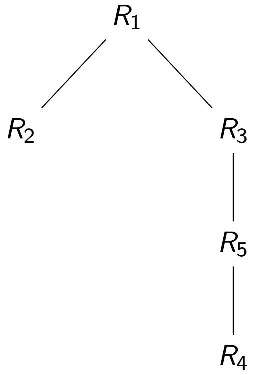
IKKBZ - example



Relation	n	s	C	T	*rank*
2	20	$\frac{1}{5}$	4	4	$\frac{3}{4}$
3	30	$\frac{1}{3}$	10	10	$\frac{9}{10}$
4	50	$\frac{1}{10}$	5	5	$\frac{4}{5}$
5	2	1	2	2	$\frac{1}{2}$

IKKBZ - example

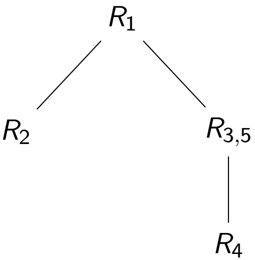
Subtree R_3: merging,
$rank(R_5) < rank(R_4)$



Relation	n	s	C	T	*rank*
2	20	$\frac{1}{5}$	4	4	$\frac{3}{4}$
3	30	$\frac{1}{3}$	10	10	$\frac{9}{10}$
4	50	$\frac{1}{10}$	5	5	$\frac{4}{5}$
5	2	1	2	2	$\frac{1}{2}$

IKKBZ - example

Subtree R_1:
$rank(R_3) > rank(R_5)$,
normalizing



Relation	n	s	C	T	*rank*
2	20	$\frac{1}{5}$	4	4	$\frac{3}{4}$
3	30	$\frac{1}{3}$	10	10	$\frac{9}{10}$
4	50	$\frac{1}{10}$	5	5	$\frac{4}{5}$
5	2	1	2	2	$\frac{1}{2}$
3,5	60	$\frac{1}{3}$	30	20	$\frac{19}{30}$

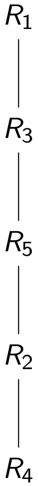
IKKBZ - example

Subtree R_1: merging

Relation	n	s	C	T	*rank*
2	20	$\frac{1}{5}$	4	4	$\frac{3}{4}$
3	30	$\frac{1}{15}$	10	10	$\frac{9}{10}$
4	50	$\frac{1}{10}$	5	5	$\frac{4}{5}$
5	2	1	2	2	$\frac{1}{2}$
3,5	60	$\frac{1}{15}$	30	20	$\frac{19}{30}$

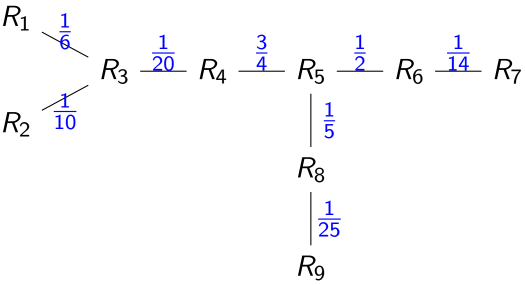
IKKBZ - example

Denormalizing



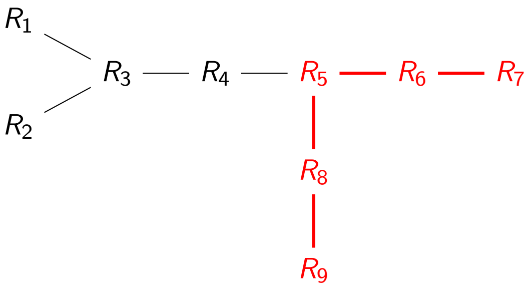
Relation	n	s	C	T	*rank*
2	20	$\frac{1}{5}$	4	4	$\frac{3}{4}$
3	30	$\frac{1}{15}$	10	10	$\frac{9}{10}$
4	50	$\frac{1}{10}$	5	5	$\frac{4}{5}$
5	2	1	2	2	$\frac{1}{2}$
3,5	60	$\frac{1}{3}$	30	20	$\frac{3}{30}$

IKKBZ - another example



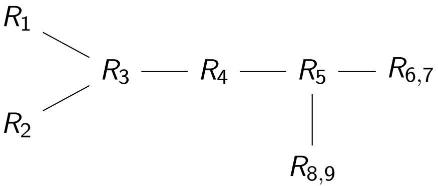
- $|R_1| = 30$
- $|R_2| = 100$
- $|R_3| = 30$
- $|R_4| = 20$
- $|R_5| = 10$
- $|R_6| = 20$
- $|R_7| = 70$
- $|R_8| = 100$
- $|R_9| = 100$

IKKBZ



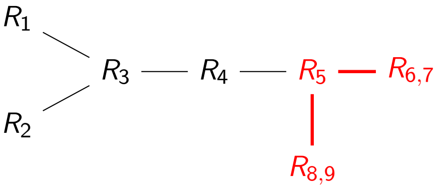
- $r(R_2) = \frac{9}{10} = 0.9$
- $r(R_3) = \frac{4}{5} = 0.8$
- $r(R_4) = 0$
- $r(R_5) = \frac{13}{15} \approx 0.86$
- $r(R_6) = \frac{9}{10} = 0.9$
- $r(R_7) = \frac{4}{5} = 0.8$
- $r(R_8) = \frac{19}{20} = 0.95$
- $r(R_9) = \frac{3}{4} = 0.75$

IKKBZ



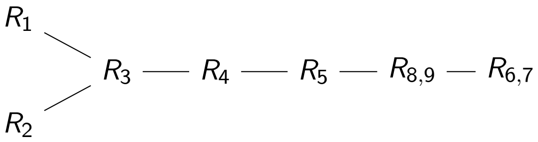
- $C(R_{8,9}) = 100$
- $T(R_{8,9}) = 80$
- $r(R_{8,9}) = \frac{79}{100} = 0.79$
- $C(R_{6,7}) = 60$
- $T(R_{6,7}) = 50$
- $r(R_{6,7}) = \frac{49}{60} \approx 0.816$

IKKBZ



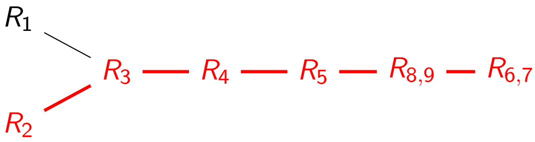
- $C(R_{8,9}) = 100$
- $T(R_{8,9}) = 80$
- $r(R_{8,9}) = \frac{79}{100} = 0.79$
- $C(R_{6,7}) = 60$
- $T(R_{6,7}) = 50$
- $r(R_{6,7}) = \frac{49}{60} \approx 0.816$

IKKBZ



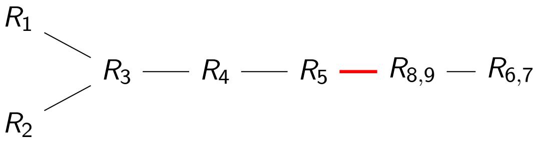
- $r(R_{8,9}) < r(R_{6,7})$

IKKBZ



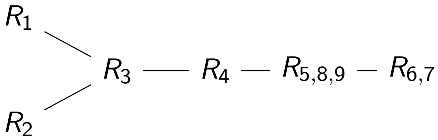
- $r(R_4) = 0$
- $r(R_5) = \frac{13}{15} \approx 0.86$
- $r(R_{8,9}) = \frac{79}{100} = 0.79$
- $r(R_{6,7}) = \frac{49}{60} \approx 0.81$

IKKBZ



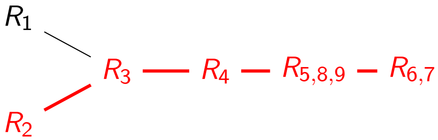
- $r(R_5) = \frac{13}{15} \approx 0.86$
- $r(R_{8,9}) = 0.79$

IKKBZ



- $n_{5,8,9} = 800$
- $C_{5,8,9} = \frac{1515}{2}$
- $T_{5,8,9} = 600$
- $r(R_{5,8,9}) = \frac{1198}{1515} \approx 0.79$
- $r(R_{6,7}) \approx 0.816$

IKKBZ



- $r(R_2) = \frac{9}{10}$
- $r(R_3) = 0.8$
- $r(R_4) = 0$
- $r(R_{5,8,9}) = \frac{1198}{1515} \approx 0.79$
- $r(R_{6,7}) \approx 0.816$

IKKBZ

$$R_1 \text{ —— } R_3 \text{ —— } R_4 \text{ — } R_{5,8,9} \text{ — } R_{6,7} \text{ —— } R_2$$

IKKBZ

$$R_1 \text{ --- } R_3 \text{ --- } R_4 \text{ --- } R_5 \text{ --- } R_8 \text{ --- } R_9 \text{ --- } R_6 \text{ --- } R_7 \text{ --- } R_2$$

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