Query Optimization: Exercise

Session 4

Bernhard Radke

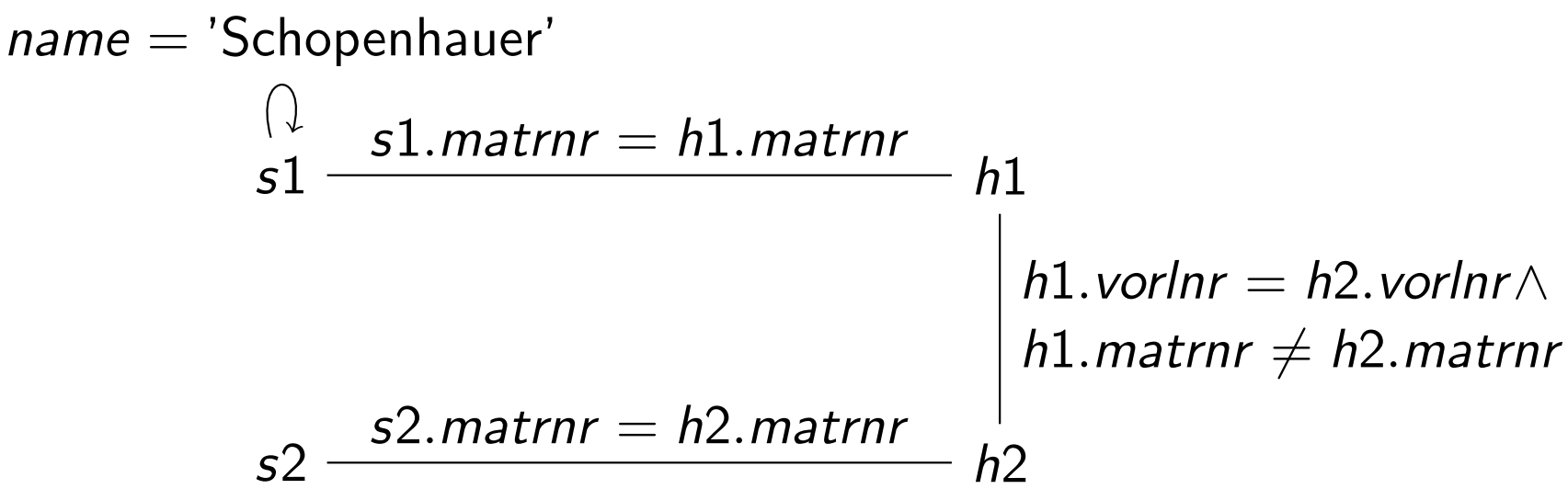
November 12, 2018

Homework

Exercise 1

```
select s2.name
from studenten s1, hoeren h1, hoeren h2, studenten s2
where s1.name='Schopenhauer' and s1.matrnr=h1.matrnr
  and h1.vorlnr=h2.vorlnr and h2.matrnr=s2.matrnr
  and h1.matrnr<>h2.matrnr
```

```
select s2.name
from studenten s1, hoeren h1, hoeren h2, studenten s2
where s1.name='Schopenhauer' and s1.matrnr=h1.matrnr
  and h1.vorlnr=h2.vorlnr and h2.matrnr=s2.matrnr
  and h1.matrnr<>h2.matrnr
```



Exercise 2

Exercise 2

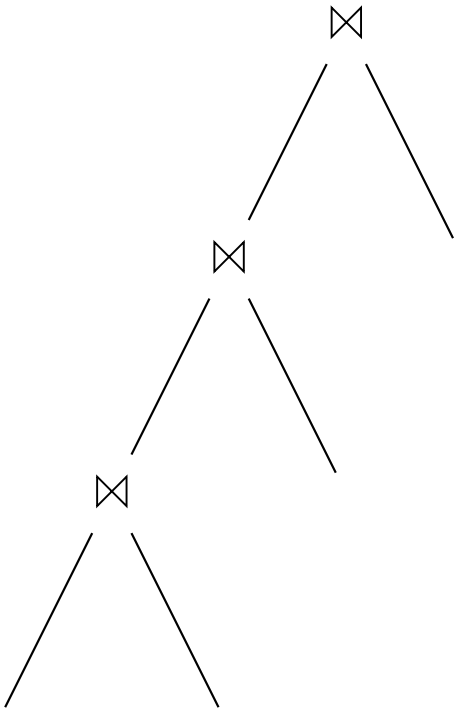
- When is a cross product beneficial?

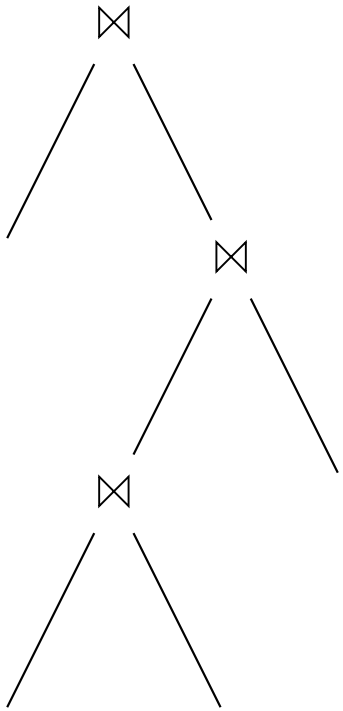
Exercise 2

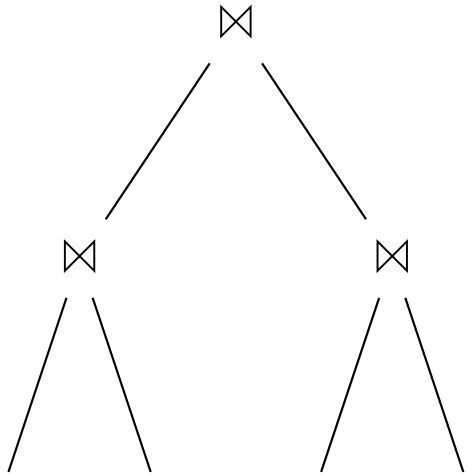
- When is a cross product beneficial?
- When is a bushy tree beneficial?

Join Ordering

Join Tree







Query Graph

```
select *
from R1, R2, R3, R4
where R1.a=R2.b
  and R2.c=R3.d
  and R3.e=R4.f
```

```
select *
from R1, R2, R3, R4
where R1.a=R2.b
  and R2.c=R3.d
  and R3.e=R4.f
  and R4.g=R1.h
```

```
select *
from R1, R2, R3, R4
where R1.a=R2.b
  and R1.c=R3.d
  and R1.e=R4.f
```

```
select *
from R1, R2, R3, R4
where R1.a=R2.b
  and R1.c=R3.d
  and R1.e=R4.f
  and R2.g=R3.h
  and R2.i=R4.j
  and R3.k=R4.l
```

Cardinality, Selectivity and Cost Function

- $|\sigma(R)| = f_R \cdot |R|$
- $|R_1 \bowtie R_2| = f_{1,2} \cdot |R_1||R_2|$

- $|\sigma(R)| = f_R \cdot |R|$
- $|R_1 \bowtie R_2| = f_{1,2} \cdot |R_1||R_2|$
- $$|T| = \begin{cases} |R_i| & \text{if } T \text{ is a leaf } R_i \\ (\prod_{R_i \in T_1, R_j \in T_2} f_{i,j}) |T_1||T_2| & \text{if } T = T_1 \bowtie T_2 \end{cases}$$

- $C_{\text{out}}(T) = \begin{cases} 0 & \text{if } T \text{ is a leaf } R_i \\ |T| + C_{\text{out}}(T_1) + C_{\text{out}}(T_2) & \text{if } T = T_1 \bowtie T_2 \end{cases}$

- $C_{\text{out}}(T) = \begin{cases} 0 & \text{if } T \text{ is a leaf } R_i \\ |T| + C_{\text{out}}(T_1) + C_{\text{out}}(T_2) & \text{if } T = T_1 \bowtie T_2 \end{cases}$
- $C_{NL}(T_1 \bowtie T_2) = |T_1||T_2|$
- $C_{HJ}(T_1 \bowtie T_2) = 1.2|T_1|$
- $C_{SMJ}(T_1 \bowtie T_2) = |T_1|log(|T_1|) + |T_2|log(|T_2|)$

First Greedy Heuristics

Construct a linear join tree

Construct a linear join tree

- GreedyJoinOrdering-1: order relations by cardinality

Construct a linear join tree

- GreedyJoinOrdering-1: order relations by cardinality
- GreedyJoinOrdering-2: order relations by selectivity

Construct a linear join tree

- GreedyJoinOrdering-1: order relations by cardinality
- GreedyJoinOrdering-2: order relations by selectivity
- GreedyJoinOrdering-3: order by selectivity, try each relation as start relation

Homework

- Give an example query graph for which GOO does not give the optimal join tree
- Perform IKKBZ heuristic on this query and compare C_{out}
- Implement a Query Graph for TinyDB

- Slides: db.in.tum.de/teaching/ws1819/queryopt
- Exercise task: gitlab
- Questions, Comments, etc:
  mattermost @ mattermost.db.in.tum.de/qo18
- Exercise due: 9 AM next monday