# Query Optimization: Exercise

## Session 2

Bernhard Radke

October 29, 2018

# Homework

## Exercise 1

- Find all students that attended the lectures together with '*Schopenhauer*', excluding *Schopenhauer* himself.
  - SQL

```
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
```

  - tuple calculus

$$\{s_1 | s_1 \in \text{Studenten} \wedge \exists h_1 \in \text{hoeren}(s_1.MatrNr = h_1.MatrNr \wedge \exists h_2 \in \text{hoeren}(h_1.VorlNr = h_2.VorlNr \wedge h_1.MatrNr \neq h_2.MatrNr \wedge \exists s_2 \in \text{Studenten}(h_2.MatrNr = s_2.MatrNr \wedge h_2.Name = \text{'Schopenhauer'}))\}$$

- Find all students that attended the lectures together with '*Schopenhauer*', excluding *Schopenhauer* himself.
  - domain calculus

$$\{[n_1] | \exists m_1, s_1([m_1, n_1, s_1] \in \text{Studenten}$$
$$\wedge \exists v([m_1, v] \in \text{hoeren}$$
$$\wedge \exists m_2([m_2, v] \in \text{hoeren} \wedge m_2 \neq m_1$$
$$\wedge \exists n_2, s_2([m_2, n_2, s_2] \in \text{Studenten} \wedge n_2 = \text{'Schopenhauer'})$$
$$))))\}$$

- Find all professor whose lectures attended at least two students
- No group by in TinyDB

# Textbook Optimization

- Selectivity $f_R$ of a selection $\sigma(R)$

$$f_R = \frac{|\sigma(R)|}{|R|}$$

- Selectivity $f_{1,2}$ of a join $R_1 \bowtie R_2$

$$f_{1,2} = \frac{|R_1 \bowtie R_2|}{|R_1 \times R_2|} = \frac{|R_1 \bowtie R_2|}{|R_1| \cdot |R_2|}$$

- Basic cost function

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

- Find the cheapest execution plan

# Physical Optimization

Choose the actual implementation of an operator

- choosing index or table scan
  - index vs. table scan: 10% selectivity threshold
  - clustered vs. non-clustered index
- choosing types of joins
  - nested loops join
  - blockwise nested loops join
  - index nested loop join
  - merge join
  - hash join

- Courses(ID,Title,Room,Time)
- Exercises(ID,CID,TID,Room)
- Tutors(ID,Name)

```
select C.Name, T.Name, E.Room
from Courses C, Tutors T, Exercises E
where C.ID = E.CID and T.ID = E.TID
      and C.Room like '02.11.%'
      and E.Room not like '02.11.%'
```

- non-clustered index on Courses.Room
- a) clustered indexes on Exercises.TID, Tutors.ID
- b) only clustered index on Tutors.ID

# Homework

- Prove an equivalence
- Derive formulae to estimate selectivities
- Join costs: nested loops vs. blockwise nested loops

How to get the new exercise task:

- add the repository you forked from as additional remote:

  ```
  git remote add tasks ssh://git@gitlab.db.in.tum.de:2222/qo18/tasks
  ```

- pull the new task from this remote:

  ```
  git pull tasks master
  ```

- 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