Query Optimization ’16
Exercise Session 2

Bernhard Radke

November 14
Important

- Please **literally** prepend **[qo16]** to the subject of emails regarding query optimization exercises!
Homework

- Find all professors whose lectures attended at least two students
- No Group By in TinyDB
Find all professors whose lectures attended at least two students

No Group By in TinyDB

```
select p.name
  from Professoren p, Vorlesungen v,
       Hoeren h1, Hoeren h2
where p.persnr=v.gelesenvon
  and v.vorlnr=h1.vorlnr
  and v.vorlnr=h2.vorlnr
  and h1.matrnr<>h2.matrnr;
```
Given the selectivity $f_R$ of a selection $\sigma(R)$

$$|\sigma(R)| = f_R \cdot |R|$$
Selectivities

- Given the selectivity $f_R$ of a selection $\sigma(R)$

  $$|\sigma(R)| = f_R \cdot |R|$$

- Given the selectivity $f_{1,2}$ of a join $R_1 \bowtie R_2$

  $$|R_1 \bowtie R_2| = f_{1,2} \cdot |R_1| \cdot |R_2|$$
Join Ordering

Basic cost function

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

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 alternative
Physical Optimization

Choose the actual implementation of an operator
  - choosing indexes or table scan
    - index vs table scan: 10% selectivity threshold
    - clustered index
    - non-clustered index
  - choosing types of joins
    - nested loops join
    - blockwise nested loops join
    - index nested loop join
    - merge join
    - hash join
Physical Optimization

- 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 '02.09.%'
  and E.Room not like '02.09.%';
Physical Optimization

- 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 '02.09.%'
    and E.Room not like '02.09.%';
```

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

Search space is defined by:

- Query graph type
Search space

Search space is defined by:

- Query graph type (chain, star, tree, clique, cycle, grid)
- Join tree class
Search space

Search space is defined by:

- Query graph type (chain, star, tree, clique, cycle, grid)
- Join tree class (left-deep, zig-zag, bushy)
- Cost function class
Search space

Search space is defined by:

▶ Query graph type (chain, star, tree, clique, cycle, grid)
▶ Join tree class (left-deep, zig-zag, bushy)
▶ Cost function class (symmetry, ASI)
Search space

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

- What kind of query graph is it?
Search space

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

▸ What kind of query graph is it?
▸ Let’s allow cross-products ⇒ no restrictions on the order in which relations are joined
select *
from R1, R2, R3, R4
where R1.a = R2.b
    and R2.c = R3.d
    and R3.e = R4.f

▶ What kind of query graph is it?
▶ Let’s allow cross-products ⇒ no restrictions on the order in which relations are joined
▶ Count left-deep trees
▶ Count zig-zag trees
▶ Count bushy trees
Consider the TPC-H benchmark (http://www.tpc.org/tpch/) and the query:

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'.

Do canonical translation and logical optimization.
Homework: Task 2 (10 points)

Given $|R1|$, $|R2|$, and sizes of domains $|R1.x|$ and $|R2.y|$ and the information if $R1.x$ and/or $R2.y$ are keys of $R1$ and $R2$

- How can we estimate the selectivity of $\sigma_{R1.x=c}$, where $c$ is a constant?
- How can we estimate the selectivity of $\Join_{R1.x=R2.y}$?

NB: we can not assume that we know the size of $\Join_{R1.x=R2.y}$ (the other way round, we estimate the join size using the selectivity estimation. But how to estimate the selectivity?)
Homework: Task 3 (10 points)

Given are two relations R and S, with sizes 1,000 and 100,000 pages respectively.

Each page has 50 tuples.

The relations are stored on a disk, the average access time for the disk is 10 ms and the transfer speed is 10,000 pages/sec.

Question 1: How long does it take to perform the Nested Loops Join of R and S?

Question 2: How long does it take to perform the Block Nested Loops Join with a block size of 100 pages?

Assume that CPU costs are negligible and ignore I/O costs for the join output.
Offer

- Master Students?

Send your CV to Andrey Gubichev (gubichev@google.com)
Offer

- Master Students?
- Internship @ Google?

Send your CV to Andrey Gubichev (gubichev@google.com)
Offer

- Master Students?
- Internship @ Google?
- Send your CV to Andrey Gubichev (gubichev@google.com)
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