Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

January 30, 2020

Databases IIB: DBMS-Implementation — Exercise Sheet 12 —

Please read Part a) think about the answers, and mark questions which you want to discuss in class. You only have to submit Part c) and d). Please upload your solution into the StudIP file folder called "Hausaufgabe_12" in the StudIP entry of the lecture. The deadline is February 4 (the day before the next lecture).

The deadline for the implementation project (Homework 13 to 15) is February 22. If you really cannot make it within the deadline, you can try to propose some other date.

Repetition Questions

a) What would you answer to the following question in an oral exam?

- A query evaluation plan (or "access plan") is relatively similar to an expression (operator tree) of relational algebra. What are the main differences?
- Name some operations that appear in Oracle QEPs.
- How can one view the query evaluation plan the the Oracle optimizer has chosen for a query?
- How can one check whether Oracle really uses an index? What are the options if Oracle does not use an index that one has created to speed up a particular query?
- Why is it good to avoid the materialization of temporary results during the evaluation of a QEP, i.e. the storage of the result relation of a subtree of the QEP?
- For which operation is it unavoidable to materialize temporary results?
- Are there situations in which the storage of intermediate results might be advantageous? Discuss advantages and disadvantages of pipelined/lazy evaluation.
- Explain the interface of nodes in the QEP operator tree if one wants to use pipelined/lazy evaluation.
- Explain the parameters `sort_area_size` and `sort_area_retained_size` of the Oracle server. Why does it make sense to choose `sort_area_retained_size` smaller than `sort_area_size` if memory is scarce?
- Where does Oracle store temporary data for sorting if memory is not sufficient?

- Explain how the Mergesort algorithm works. How are sorted "runs" merged? How can it be implemented with four files?
- What is the complexity of Mergesort (and of sorting in general)?
- How can the Mergesort algorithm be improved if the input is already partially sorted?
- How can the Mergesort algorithm be improved if some memory is available (not enough to do the complete sort in memory)?
- Name four different join algorithms.
- Explain the nested loop join. What is the complexity of the nested loop join if both tables have n rows?
- How can the nested loop join be improved to make use of available memory?
- Explain the merge join. What is the complexity of the merge join if both tables have n rows, and the join attribute is a key on one side?
- Compare nested loop join and merge join. What can the nested loop join do that is not possible with the merge join?
- Explain the index join. What is its complexity? In which case is the index join clearly better than nested loop and merge join? What is an advantage of the merge join?
- Explain the hash join. What is its basic idea?
- Name some equivalences of relational algebra that can be used for algebraic query optimization.
- Why is it usually better to do a selection before a join? Explain an example where this is not better.
- Suppose an SQL query declares tuple variables over four relations `R1`, `R2`, `R3`, `R4` under `FROM`. Which structure of joins will a normal DBMS consider? What is a "bushy join"?

In-Class Exercises

b) We will continue to discuss the exam from 2015/16, which you find here:

[http://www.informatik.uni-halle.de/˜brass/dbi17/exam15.pdf]

Homework Exercises

c) Let a relation `R(A,B,C,D)` be given. The attribute types are:

- `A`, `B`, `C`: `NUMERIC(3)`,
- `D`: `VARCHAR(1000)`.

Attribute `A` is the key of the relation. Consider the following query:

```
SELECT A, D
FROM   R
WHERE  B = 5
AND    C BETWEEN 30 AND 100
```

There are the following indexes:

- `I1` on `R(A)`,
- `I2` on `R(B)`, and
- `I3` on `R(C)`.

First consider the evaluation of the query with a full table scan. Give the Oracle query evaluation plan (in the graphical tree notation or the textual indented notation) for this execution of the query. Furthermore, give an SQL query to the data dictionary that returns an estimate of the number of block accesses that the query will need. Of course, you can assume that the `ANALYZE TABLE` has just been done, so the size information in the data dictionary is available and current. If data should be missing in the Oracle data dictionary, explain what information would be needed.

d) Now do the same for access via `I2` (with condition `B = 5`): Give the Oracle QEP and an SQL query that returns an estimate of the number of blocks that will be accessed.

Homework 13 (Deadline: February 22)

Please submit a short text that explains the algorithm and the data/file structure that you use for the implementation project.

Homework 14 (Deadline: February 22)

Please submit the source code (in C++) for the program that reads the input file with lines such as

```
Ann|Smith|1|10
```

and produces your data file. Your program will be called in the form

```
./prog -c datafile input.txt
```

You may write only `datafile`.
The size of the input file will be around 100 MB. Your data file should not be larger than 300 MB, and the total memory that your program uses ("resident set size") should not be more than 1 GB.
You may not use any non-standard libraries and also no data structures from the standard library. If you want, you can produce an alternative version with any library you wish. We will measure the time for that, too, but it does not participate in the competition.
Note that the exercise numbers are not necessarily sequential and may span the entire range of non-negative integers.

Homework 15 (Deadline: February 22)

Please write a program (in C++) that accepts command line arguments for first and last name and yields the sum of the homework points. I.e. your program is called as

```
./prog -q datafile Ann Smith
```

and must print the total number of homework points (10 if the above line is the only entry for `Ann Smith`). For the competition, you must be prepared to quickly exchange the sum of homework points by some other computation based on exercise numbers and points. The purpose of this rule is that you are not allowed to precompute the sum of homework points. You must be able to access exercise number and points for all records with the given first and last name.
The sum of the runtimes of the creation of the data file and ten query executions will be the number that counts for the competition.