November 8, 2018
Arnab Chakrabarti, Rihan Hai

Implementation of Databases (WS 18/19)

Exercise 3

Due until November 20, 2018, 10am.
Please submit your solution *in a single PDF file* before the deadline to the L^{2}P system!
Please submit solutions in groups of three students.

Exercise 3.1 (Tableaux Containment and Minimization) (5 pts)

Given are the following tableaux:

T_1

a1	a2	
b3	a2	(R)
a1	b4	(R)
5	b3	(R)
b4	5	(R)

T_2

a1	a2	
b4	a2	(R)
b1	a2	(R)
a1	b3	(R)
b2	b4	(R)
b2	b1	(R)
b3	b2	(R)

T_3

a1	a2	
b3	a2	(R)
a1	b2	(R)
b4	b1	(R)
b1	b2	(R)
b2	b3	(R)
b1	b3	(R)

1. Find out if $T_i \subseteq T_j$ i.e., $T_i \equiv T_j$ for $i \neq j, i, j \in \{1, 2, 3\}$. (3 pt.)

2. Write down the minimal tableau for $T_i, i \in \{1, 2, 3\}$. (2 pt.)

Exercise 3.2 (Quant Graphs) (6 pts)

Given is the Chinook database schema (see Assignment 1).

1. Specify the following queries in TRC and draw the corresponding quant graphs, subsequently. Determine for each graph, if it contains a cycle. Explain in your own words what this means for the query.

   (a) EmployeeId and LastName of employees who have supported at least two different customers living in the same city.

   (b) Track name and composer of all tracks, which are from the genre "Rock" and where the artist is also the composer.

Exercise 3.3 (Join Implementations) (4 pts)

Let relations R and S have the following properties:

- R has 10,000 tuples and has 10 tuples per page
- S has 2000 tuples and also has 10 tuples per page
- Both the relations are stored as simple heap files and neither relation has any indexes built on it.

The total number of buffers available is 52.

Now considering the join $R \bowtie_{R.a=S.b} S$ assuming that attribute b on relation S is the primary key for S, answer the following questions: (The cost of writing out the result should be uniformly ignored.)

1. Calculate the I/O requirements of a **page-oriented simple nested loop join**.
2. Calculate the I/O requirements of a **block nested loop join**.

Exercise 3.4 (Cost Estimation) (15 pts)

1. Consider the following relations and the number of distinct values for the different attributes $V(R,a)$. Each of the relations has a size of 1000 tuples. Determine an optimal order for joining all three relations. For each intermediate step determine the sizes of the results in terms of tuples. The cost of each plan is the sum of all intermediate result sizes. Select the best plan based on your results. (4 pt.)

R(a,b)	S(b,c)	T(c,d)
V(R,a) = 100		
V(R,b) = 200	V(S,b)=100	
	V(S,c)=500	V(T,c)=20
		V(T,d)=50

2. Consider the following relational schema and SQL query. The schema captures information about employees, departments, and finances (organized on a per department basis).

$$Suppliers(\underline{sid}, sname, city)$$
$$Supply(\underline{sid, pid})$$
$$Parts(\underline{pid}, pname, price)$$

```
SELECT P.pname
FROM   Suppliers S, Parts P, Supply Y
WHERE  S.sid=Y.sid AND Y.pid=P.pid AND
 S.city='Aachen' AND P.price <= 200 AND
 Y.pid < 800
```

(a) Identify a relational algebra tree (or a relational algebra expression if you prefer) that reflects the order of operations a decent query optimizer would choose (applying the heuristics selection before join, and projections are done as many and as early as possible). (3 pt.)

(b) What indexes for single attributes might be of help in processing this query? Should it be clustered or unclustered indexes, using hash tables or B+ trees? (2 pt.)

(c) Suppose that the following additional information is available: The system's statistics indicate that there are suppliers from 100 cities, and part prices range from 1 to 1000 Euros. All attribute values are uniformly distributed and each id has been created in numerical order without any records deleted so far. There are a total of 400 suppliers, 2.000 parts, and 200.000 entries in the Supply relation in the database.

For each of the query's base relations (Supply, Suppliers, Parts) estimate the number of tuples that would be initially selected from that relation if all of the non-join predicates on that relation were applied to it before any join processing begins. (5 pt.)

(d) Based on the results for (c), which pair of relations should be joined first? (1 pt.)