Exercise Sheet 5: Query Optimization (until Friday 06.06.2014)

Exercises will be discussed on **Friday** on week after the respective lecture was given. The handouts are optional and do not have to be handed in. They only serve as optional preparation for the oral exams at the end of the semester.

Exercise 1

Assume the relations given in Appendix A.

a. Draw a **canonical operator tree** for following queries and denote the number of rows of the **intermediate result sets** of each operation in the tree:
   1. **SELECT** title **FROM** courses
   2. **SELECT** firstname, lastname **FROM** students **WHERE** sex='f'
   3. **SELECT** s.firstname, s.lastname, r.result, c.title **FROM** students s, results r, courses **WHERE** s.matNr=r.matNr AND r.crsNr=c.crsNr AND c.crsNr=100

b. Rewrite query 3 from exercise a) using **Algebraic Transformation Rules** in such a way that the intermediate results are reduced significantly. Please mark which rule you used and draw the resulting operator tree annotated with the new rows per intermediate result count.

Exercise 2

Assume the relations given in Appendix A. You want to retrieve all female students (mat no) that got a 1.0 in the course "Secret Identities 2".

a. Create an **SQL query** for performing that task.

b. Draw an operator tree that you have optimized with best effort.

c. Suppose you only have statistical information about the relations in appendix A. Your information includes the **number of rows** in each relation and for each attribute the **minimum value**, **the maximum value** and the **number of distinct values**.
   Annotate each node in your optimized operator tree with estimated number of rows (float numbers are permitted). Your estimations must only be based on the **statistical information** you have. Do you think it is a good estimation?

   Hint: To apply the rules as given in the lecture, I recommend **not** to introduce joins but remain it as a cross product directly followed by the respective selection.

d. Suppose every relation contains several thousands of entries. What secondary indexes would be beneficial to significantly reduce the **number of block accesses** to answer the query?

Appendix A

Students

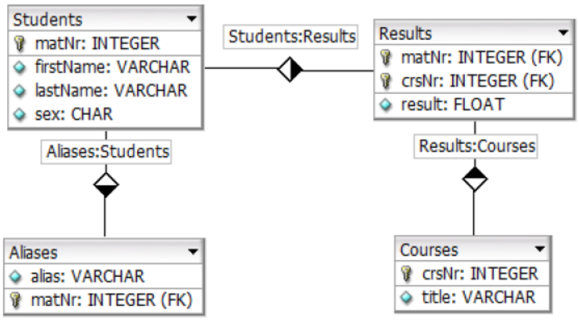
matNr	firstName	lastName	sex
1000	Clark Joseph	Kent	m
1001	Louise	Lane	f
1002	Lex	Luthor	m
1003	Charles	Xavier	m
1004	Erik	Magnus	m
1005	Jeanne	Gray	f
1006	Ororo	Munroe	f
1007	Tony Edward	Stark	m
1008	Matt	Murdock	m
1009	Raven	Wagner	f
1010	Robert Bruce	Banner	m

Results

matNr	crsNr	result
1009	100	3.7
1002	102	5.0
1000	101	4.0
1000	100	1.3
1004	102	1.3
1003	101	1.7
1007	103	3.0
1006	100	1.7
1009	102	1.3
1003	103	1.0
1009	101	1.0
1008	101	1.7

Aliases

alias	matNr
Mystique	1009
Daredevil	1008
Kal-El	1000
Professor X	1003
Hulk	1010
Windrider	1006
Superman	1000
Phoenix	1005
Ironman	1007
Magneto	1004
Mockingbird	1002
Storm	1006
Golden Avenger	1007
Queen of Wakanda	1006



Courses

crsNr	title
100	Introduction to Superheroism
101	Secret Identities 2
102	How to take over the world
103	Codes of Justice