

2. Unit: Schema Generation, Hierarchical Queries, and Optimization

The following exercises should be solved by defining views and additional tables. The formular on the web is not appropriate to this exercise sheet.

Exercise 2.1 (Symmetric borders view; 10 P.)

- What do the properties *reflexive*, *irreflexive*, *symmetric* and *asymmetric* mean with respect to relations? Specify queries checking these properties for the *borders* relation.
- Define a view *sym_borders* that contains the symmetric closure of *borders*. This view should be used in subsequent exercises.
(Hint: http://en.wikipedia.org/wiki/Closure_%28mathematics%29)
- Discuss the benefits and downsides of symmetry/asymmetry with respect to tables and views.

Exercise 2.2 (Views for intermediate results; 15 P.)

Calculate the population density of the region containing the countries Algeria, Libya and all of their neighbors. Compare the result with the population density of the region excluding the desert areas. Therefore you need to examine if there are deserts in mondial that are both inside and outside the above region. Use views to structure your solution *reasonably*.

Exercise 2.3 (Complex Attribute Types; 5 P.)

Specify the nearest city for every mountain in Russia.

Hint: The city may be in a neighboring country.

Exercise 2.4 (Update; 10 P.)

Exchange the members of NATO and EU *without* introducing auxiliary constructs (tables, views, auxiliary entries, PL/SQL). Hint: compute the new value of the attribute *Organization* as a function from the original one. Note: do *not* use ORACLE's DECODE-function.

What has to be done to get the original state of the database again?

Exercise 2.5 (Dates; 15 P.)

The table *politics* contains an column with the independence date. State the following queries:

- All countries with an independence date between 1300 and 1600.
- The average independence date of the european countries.
- Specify the independence date for all countries in a table with two columns. The first column should contain the country code, the second the date in the form of the following example (without quotes): "18. Jan"
- Compute all pairs of countries whose independence dates differ less than half a year.

Note: Use the data formats specified in the ORACLE documentation.

Exercise 2.6 (Transitive closure, hierarchical queries; 15 P.)

- Compute all tributary rivers of the river *Zaire*, i.e., all rivers that flow either directly or indirectly into the Zaire. Use an auxiliary relation that collects rivers. Do *not* use ORACLE's CONNECT BY PRIOR. Compute the length of the whole river system.
- Now compute the river system by using CONNECT BY PRIOR (see "Hierarchical Queries" of ORACLE's "SQL Reference").
- Discuss the possibilities to retrieve the seas connected with the North Sea by using CONNECT BY PRIOR.

Exercise 2.7 (Tools for performance analysis; 15 P.)

ORACLE provides several tools for analyzing and optimizing queries. This exercise deals with two of them, the command or system variable `TIMING` that measures the runtime of a query execution. Furthermore `EXPLAIN PLAN` shows, how ORACLE executes, if it has to answer a query.

- At first read the documentation of ORACLE for `TIMING` and `EXPLAIN PLAN` and understand how to use these features. The former one can be found in “SQL*Plus User’s Guide and Reference”, and the latter one is explained in ORACLE’s “Performance Tuning Guide”.
- Some queries in exercise sheet 1 have a long response time. Choose an interesting one of them and generate a version that has the same result, but a different runtime behaviour. Measure the time with the `TIMING` feature. (Hint: For example, the same query can be written either with `NOT IN` expressions, or `NOT EXISTS`, however, the efficiency can be considerably different.) Inspect the execution plan of ORACLE by `EXPLAIN PLAN`. Format the result and explain the differences in the runtime behaviour.

Exercise 2.8 (Optimization via Index; 15 P.)

Choose one of the queries with long execution time from the previous exercise sheet (not the same as in the previous exercise). Examine this query with `EXPLAIN PLAN` and accelerate it by one or several indexes. Document the improvement by `TIMING` and `EXPLAIN PLAN`.

Deadline: 24.05.2006, 12:00