Programming Applications with Databases

Exercise Set 3

1. Consider the following tables: *Products(ID, ProductName)*, *Prices(ProductID REF Products(ID), Currency REF Rates(Currency), Price)*, *Rates(Currency, PricePLN)*. Note that a product price may not be declared in all known currencies, but it always is declared in PLN as a reference. Prepare a batch script to update the *Prices* table based on *Rates* table. In case there is a row in *Prices* which references a currency that no longer exists in *Rates*, the row should be removed.

This task should be implemented using cursors.
[3p]

2. Create tables *Employees(ID, SalaryGros)* and *SalaryHistory(ID, EmployeeID, Year, Month, SalaryNet, SalaryGros)*. Implement a procedure that for a month number (given as a parameter) computes the salary to be paid out, based on:

- the current salary (*Employees.SalaryGros*),
- all previous salaries this year to determine tax deductions,
- the following tax thresholds need to be considerd: 18% for persons whose income does not exceed 85528 PLN/year and 32% for persons whose income is above this threshold (but applied to the excess of the income over this threshold).

Please take into account that an employee may be employed in any month during the year. Create also a table for logging the situations in which the salary could not be determined, e.g. because of a missing salary in one of the previous months (note that this doesn't contradict with the situation where employment starts not from January).

This task should be implemented using cursors.

Remark: possibly consider a slight modification to the schema of SalaryHistory table.
[3p]

3. Consider the following tables: *Cache(ID, UrlAddress, LastAccess)*, *History(ID, UrlAddress, LastAccess)* and *Parameters(Name, Value)*. The meaning of fields of *Cache* and *History* tables is as follows:

- *ID* — row identifier,
- *UrlAddress* — the address of a website,
- *LastAccess* — the time of the last visit with with an accuracy of a second.

The *Parameters* table contains only one row with *Name* = *max_cache* and *Value* set to the maximum size of the *Cache* table. At the time of inserting of a new row to *Cache* table the following conditions should be met:

- If the cache already contains the address of the website being inserted, the last access time should be only modified,
- Otherwise, the size of the cache should be compared to the value in the *max_cache* parameter from the *Parameters* table. There are two cases:
  - If it's lower, the new row should be inserted.
  - Otherwise, determine a website for which the last access is the oldest (if there are more than one, pick any of them). The corresponding row should be transferred to the *History* table, but in case it refers to a website already present there, the last access time should be only modified.

This task should be implemented using triggers.
[4p]

Paweł Rajba