Task list No. 4 for project laboratories 12, 13, 14, 15

Task 47. Define schema of the cats database in a relational-object form. Propose and define examples of methods for each object type. Reference bindings should be defined using reference types. Fill relations from such a scheme with data from the reality of cats and then perform sample SQL queries, using references (as joins), subqueries, grouping and methods defined within types. One own example of task and query resolving this task should be provided for each mechanism (reference, subquery, grouping). Additionally, under this new database scheme, two selected non-trivial tasks from list 2 and two from list 3 are to be carried out.

Task 48. Define the "overlay", in the form of object views, on the relational database of cats. Object types corresponding to relations are to contain examples of methods (they can be methods from Task 47). Model all reference associations using OID identifier and the MAKE_REF function. Perform for this database all SQL queries and PL/SQL blocks implemented as part of the task. 47.

Task 49. In connection with Poland's accession to the European Union, detailed records of hunted and consumed mice were necessary. Therefore, it was necessary to record both the cat that the mouse hunted (along with the date of hunting) and the cat that the mouse ate (together with the date of "payout"). In addition, the weight of the mouse has become important (this weight must meet the EU standard (please set this standard). Worst of all, however, the data had to be completed backwards, starting from January 1, 2004. Unfortunately, as it sometimes happens, there was a "slight" delay in the implementation of the recording program hunted and eaten mice. By a strange coincidence, this record was only possible the day before the date of returning the current list of tasks.

Write a block (s) that will carry out these records, so:

a) modify the database schema defining a new relation named Mice having the attributes: mouse_number (primary key), hunter (foreign key), eater (foreign key), mouse_weight, catch_date, release_date (always last Wednesday of the month),
b) fill the Mice relation by artificially generated data, starting from January 1, 2004, to the day before the date of delivery of this list. The number of mice entered, caught in a one month, is to be consistent with the number of mice that cats received as part of the "payout" that month (together with extra mice). When completing the data, it should be assumed that each cat is able to catch in a month the number of mice equal to the number of mice consumed, on average, per month by each cat (use any surpluses associated with rounding). The catch dates for the mice are to be spread evenly throughout the month. The issue date of the each mouse is to be the last Wednesday of each month.

The solution should use internal dynamic SQL (creating a new relation) and bulk binding (filling the relation with generated data). From the current date starting should be saved real data about hunted mice. It is therefore necessary to prepare a procedure that will allow for write in relation Mice of mice caught during the day by a particular cat (data on mice caught during the day are available in an individual relation, which is owned by each cat) and a procedure realizing monthly payment to cats in order of the their hierarchy. Bulk binding should be used for both procedures.