

Zestaw 10

PL/SQL – pakiety, procedury i funkcje

Przy wykonaniu poniższych ćwiczeń korzystamy z tabel utworzonych za pomocą skryptu *SUMMIT.SQL*. Zadania praktyczne wykonać w konsoli *SQL Developer*, a następnie sporządzić opis w pliku tekstowym (zawierającym wszystkie polecenia PL/SQL) i przestać na skrzynkę Moodle.

Warto przeczytać:

A. Gramacki „Język PL/SQL”, rozdziały: 5-7.

Utworzyć pakiet o nazwie *PRACOWNICY* oraz zaimplementować podane niżej procedury i funkcje.

Należy uwzględnić również następujące uwagi:

- Wszystkie procedury i funkcje muszą należeć do pewnego pakietu – nie należy ich definiować niezależnie.
- Parametry wejściowe procedur powinny być zdefiniowane jako zakotwiczone (ang. *anchor*). Ta sama uwaga dotyczy również wszystkich innych zmiennych odnoszących się do poszczególnych kolumn w tabelach.
- We wszystkich procedurach i funkcjach należy dokładnie przemyśleć oraz zaimplementować obsługę ew. błędów, tak aby programy nie „wysypywały” się w sposób niekontrolowany.
- Zwrócić uwagę na konwencje w nazewnictwie zmiennych (powinna być jednolita we wszystkich programach). Proponuje się używać przedrostków *in_*, *inout_* oraz *out_* w nazwach parametrów wejściowych procedur i funkcji (w zależności od trybu zmiennej) oraz przedrostków *c_* w nazwach kursorów. Pozostałe zmienne mogą mieć przedrostek *uv_* (od *user variable*) aby łatwiej je było odróżnić od np. identyfikatorów nazw kolumn.
- Przyjąć zasadę, że wszystkie słowa kluczowe języka SQL oraz PL/SQL piszemy dużymi literami a pozostałe identyfikatory definiowane przez użytkownika małymi. Bardzo wydatnie zwiększa to czytelność programów!
- Należy również pamiętać o konsekwentnym i jednolitym stosowaniu wcięć, których brak bardzo utrudnia czytanie programów.

Procedury i funkcje, które należy zaimplementować:

1. *ADD_EMP* – zadaniem procedury ma być dodawanie nowego pracownika do tabeli *EMP*. Numer *ID* powinien być pobierany automatycznie ze zdefiniowanej w tym celu sekwencji.
2. *CHANGE_EMP* – procedura modyfikuje dane wskazanego pracownika.
3. *DELETE_EMP* – procedura kasuje dane wskazanego pracownika.
4. *CHANGE_SALARY* – zadaniem tej procedury jest zmiana zarobków wskazanego pracownika. Podajemy procentową zmianę zarobków. Przykładowo, gdy podamy 50 oznacza to wzrost zarobków o 50%, a kiedy podamy -10 oznacza to obniżkę zarobków o 10%.
5. *TOP_N_EMP* – wyświetla listę *N* pracowników (*N* podawane jako parametr wejściowy procedury), którzy zarabiają najwięcej. Dane o tych pracownikach (imię, nazwisko, zarobki) powinni zostać dodatkowo zapisani do tabeli o nazwie *TOP_N_EMP* (czy tę tabelę można stworzyć bezpośrednio w kodzie procedury?).
6. *CHANGE_DEPT* – procedura zmienia przypisanie pracownika do wydziału (tabela *DEPT*).
7. *STAT_EMP* – zadaniem funkcji jest zwrócenie wartości (w zależności od podanego parametru) zarobków maksymalnych, minimalnych, średniej lub też sumy zarobków wszystkich pracowników. Funkcja powinna przyjmować tylko jeden z czterech parametrów: *MAX*, *MIN*, *AVG*, *SUM*. Podanie innego parametru powinno wygenerować stosowne ostrzeżenie.