Witaj w pracy domowej **modułu 6. - ZMIENNE ZŁOŻONE**Na kolejnych stronach znajdziesz zadania do wykonania, które dla Ciebie przygotowałem.

PS. Nie zapomnij przywitać się na grupie i oznaczyć pracy domowej hasztagiem #MODUŁ6.

Jeśli masz problem z którymś z zadań i chcesz bym sprawdził Twoją pracę domową to:

* umieść pracę domową na dysku google(tak, bym nie musiał pobierać pliku)
* oznacz mnie w poście i napisz z czym masz konkretnie problem

Nie ma znaczenia czy Twój plik ma rozszerzenie .docx czy .sql. Zrób jak   
Ci wygodniej :)

# 

Uwaga. Wszystkie zadania należy wykonywać będąc zalogowanym jako użytkownik kurs\_plsql, chyba, że wskazano inaczej.

1. Używając kotwiczenia %ROWTYPE wczytaj całą tabelę departments do kolekcji:
   1. asocjacyjnej
   2. zagnieżdżonej
   3. varray

Użyj pętli FOR by wyświetlić nazwę departamentu i id\_managera na konsoli.

**-- a) tabela asocjacyjna**

**DECLARE**

**TYPE at\_departments IS TABLE OF departments%ROWTYPE INDEX BY PLS\_INTEGER ;**

**kolekcja\_departments at\_departments;**

**BEGIN**

**SELECT \***

**BULK COLLECT INTO kolekcja\_departments**

**FROM departments;**

**FOR i in kolekcja\_departments.FIRST..kolekcja\_departments.LAST**

**LOOP DBMS\_OUTPUT.PUT\_LINE(kolekcja\_departments(i).department\_name||'-'|| kolekcja\_departments(i).manager\_id);**

**END LOOP;**

**END;**

**-- b) tabela ZAGNIEŻDZONA**

**DECLARE**

**TYPE nt\_departments IS TABLE OF departments%ROWTYPE ;**

**kolekcja\_departments nt\_departments;**

**BEGIN**

**SELECT \***

**BULK COLLECT INTO kolekcja\_departments**

**FROM departments;**

**FOR i in kolekcja\_departments.FIRST..kolekcja\_departments.LAST**

**LOOP DBMS\_OUTPUT.PUT\_LINE(kolekcja\_departments(i).department\_name||'-'|| kolekcja\_departments(i).manager\_id);**

**END LOOP;**

**END;**

**-- c) tabela varray**

**DECLARE**

**TYPE vt\_departments IS VARRAY(100) OF departments%ROWTYPE ;**

**kolekcja\_departments vt\_departments;**

**BEGIN**

**SELECT \***

**BULK COLLECT INTO kolekcja\_departments**

**FROM departments;**

**FOR i in kolekcja\_departments.FIRST..kolekcja\_departments.LAST**

**LOOP DBMS\_OUTPUT.PUT\_LINE(kolekcja\_departments(i).department\_name||'-'|| kolekcja\_departments(i).manager\_id);**

**END LOOP;**

**END;**

1. Stwórz kolekcję asocjacyjną indeksowaną tekstem. Zapisz do niej listę wszystkich predefiniowanych wyjątków Oracle. Indeks kolekcji to kod ORA, wartość przechowywana w kolekcji to nazwa wyjątku, np. kolekcja(‘ora-01422’) powinna zwracać wartość “too\_many\_rows”, a kolekcja(‘ora-01001’) powinna zwracać wartość “invalid\_cursor”. Do wyświetlenia zawartości kolekcji w konsoli użyj pętli WHILE.

Uwaga. Listę wyjątków znajdziesz w Twojej poprzedniej pracy domowej (moduł 5 zadanie 5)

**DECLARE**

**TYPE at\_wyjatki\_ora IS TABLE OF VARCHAR2(100) INDEX BY VARCHAR(10);**

**kolekcja\_wyjatki\_ora at\_wyjatki\_ora:= at\_wyjatki\_ora('ORA-6530'=>'ACCESS\_INTO\_NULL',**

**'ORA-6592'=>'CASE\_NOT\_FOUND',**

**'ORA-6531'=>'COLLECTION\_IS\_NULL',**

**'ORA-6511'=>'CURSOR\_ALREADY\_OPEN',**

**'ORA-1'=>'DUP\_VAL\_ON\_INDEX',**

**'ORA-1001'=>'INVALID\_CURSOR',**

**'ORA-1722'=>'INVALID\_NUMBER',**

**'ORA-1017'=>'LOGIN\_DENIED',**

**'ORA+100'=>'NO DATA FOUND',**

**'ORA-6548'=>'NO\_DATA\_NEEDED',**

**'ORA-1012'=>'NOT\_LOGGED\_ON',**

**'ORA-6501'=>'PROGRAM\_ERROR',**

**'ORA-6504'=>'ROWTYPE\_MISMATCH',**

**'ORA-30625'=>'SELF\_IS\_NULL',**

**'ORA-6500'=>'STORAGE\_ERROR',**

**'ORA-6533'=>'SUBSCRIPT\_BEYOND\_COUNT',**

**'ORA-6532'=>'SUBSCRIPT\_OUTSIDE\_LIMIT',**

**'ORA-1410'=>'SYS\_INVALID\_ROWID',**

**'ORA-51'=>'TIMEOUT\_ON\_RESOURCE',**

**'ORA-1422'=>'TOO\_MANY\_ROWS',**

**'ORA-6502'=>'VALUE\_ERROR',**

**'ORA-1476'=>'ZERO\_DIVIDE');**

**i\_index varchar(10):=kolekcja\_wyjatki\_ora.FIRST;**

**BEGIN**

**WHILE (i\_index is not null)**

**LOOP**

**DBMS\_OUTPUT.PUT\_LINE('wyjatki '||i\_index||' '||kolekcja\_wyjatki\_ora(i\_index));**

**i\_index:=kolekcja\_wyjatki\_ora.NEXT(i\_index);**

**END LOOP;**

**END;**

1. Stwórz kolekcję nested table zawierającą liczby od 1 do 10. Następnie wykorzystując metodę DELETE usuń element 2, 5 i 9. Wyświetl następnie zawartość całej kolekcji wykorzystując metodę NEXT.

Uwaga. Metoda NEXT jest Ci potrzebna do tego, by w pętli iterować wyłącznie po elementach kolekcji, tzn. by z indeksu 1 przejeść do indeksu 3, a z indeksu 4 do indeksu 6, itd. Inaczej mówiąc Twoja pętla powinna wykonać 7, a nie 10 iteracji, bo tyle elementów zawiera kolekcja.

**DECLARE**

**TYPE typ\_nt IS TABLE OF NUMBER;**

**kolekcja\_nt typ\_nt := typ\_nt(1,2,3,4,5,6,7,8,9,10);**

**i\_indeks NUMBER;**

**BEGIN**

**kolekcja\_nt.DELETE(2);**

**kolekcja\_nt.DELETE(5);**

**kolekcja\_nt.DELETE(9);**

**i\_indeks:=kolekcja\_nt.FIRST;**

**WHILE (i\_indeks IS NOT NULL)**

**LOOP**

**DBMS\_OUTPUT.PUT\_LINE(kolekcja\_nt(i\_indeks));**

**i\_indeks:=kolekcja\_nt.NEXT(i\_indeks);**

**END LOOP;**

**END;**

**--Zadanie 3 Praca domowa Moduł 6 (za pomocą exception)**

**DECLARE**

**TYPE typ\_nt IS TABLE OF NUMBER;**

**kolekcja\_nt typ\_nt := typ\_nt(1,2,3,4,5,6,7,8,9,10);**

**BEGIN**

**kolekcja\_nt.DELETE(2);**

**kolekcja\_nt.DELETE(5);**

**kolekcja\_nt.DELETE(9);**

**FOR i IN kolekcja\_nt.FIRST..kolekcja\_nt.LAST**

**LOOP**

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE (kolekcja\_nt(i));**

**EXCEPTION**

**WHEN NO\_DATA\_FOUND THEN**

**DBMS\_OUTPUT.PUT\_LINE (i||' = NULL');**

**END;**

**END LOOP;**

**END;**

1. Udziel odpowiedzi na poniższe pytania pisząc: AT(kolekcja asocjacyjna), NT (kolekcja nested table) lub VT(kolekcja VARRAY).

Która kolekcja:

* ma maksymalną liczbę elementów? **VT**
* może być indeksowana tekstem? **AT**
* ma dodatkowe operatory do porównywania kolekcji? **NT**
* nie wymaga inicjalizacji? **AT**
* nie może być rzadka? **VT**
* nie może być użyta w SQL? **AT**