

## Zadania do realizacji

1. Zadeklaruj dwie zmienne o nazwach `v_tekst`, `v_liczba` i wartościach odpowiednio „Baza danych to uporządkowany zbiór danych” oraz 2,7182. Wyświetl wartości tych zmiennych na ekranie.
2. Napisz program obliczający iloczyn dwóch liczb całkowitych. Liczby powinny być podane z konsoli przez użytkownika.
3. Napisać program obliczający pole powierzchni całkowitej i objętość kuli. W programie posłuż się zdefiniowaną przez siebie stałą `c_pi=3,14159265`. Wczytaj promień kuli od użytkownika i wyświetl wyniki na ekranie.
4. Utwórz tabelę `pracownik(id_pracownik, imie, nazwisko)` z przykładowymi rekordami:  
1 Jan Kowalski, 2 Jerzy Nowak, 3 Anna Gałka, 4 Hanna Miałka.  
Zadeklaruj zmienną `v_ilosc`, oblicz ilość rekordów w tabeli `pracownik` i otrzymany wynik zapisz do zmiennej `v_ilosc`. Używając zmiennej `v_ilosc` wypisz na ekranie komunikat w formacie: „W tabeli `pracownik` jest <ilość> rekordów”.
5. Używając typu zakotwiczonego zadeklaruj zmienne `v_imie` i `v_nazwisko` o tych samych typach co typy kolumn `imie` i `nazwisko` z tabeli `pracownik(id_pracownik, imie, nazwisko)`. Następnie wczytaj do nich imię i nazwisko pracownika o identyfikatorze równym 3. Wyświetl na ekranie wczytane imię i nazwisko. Przyjmijmy, że pracownik o identyfikatorze równym 3 zawsze istnieje i jest dokładnie jeden taki pracownik.
6. Do zmiennej `v_rekord` wczytaj rekord z danymi pracownika o identyfikatorze równym 2. (Zakładamy, że taki pracownik istnieje.) Wyświetl wczytane dane na ekranie w formacie: „Pracownik o identyfikatorze równym <numer> to <imie> <nazwisko>”. Następnie w zmiennej `v_rekord` zmień wartość pola `id_pracownik` na 10 i do tabeli `pracownik` dodaj nowy rekord o wartości zmiennej `v_rekord`. Wyświetl na ekranie zawartość tabeli `pracownik`.
7. Została utworzona tabela: `CREATE TABLE info(nazwa VARCHAR(50) NOT NULL, ilosc INT);` Dodaj do niej trzy rekordy, dla których kolumna `nazwa` przyjmuje odpowiednio wartości: `tabele`, `widoki`, `ograniczenia` (ang. `constraints`). Następnie zadeklaruj trzy zmienne i przypisz do nich odpowiednio wartości obliczeń ile zalogowany użytkownik ma tabel, widoków i ograniczeń w bazie danych. Zaktualizuj rekordy w tabeli `info` poprzez uaktualnienie wartości kolumny `ilosc` wpisując obliczoną wcześniej ilość tabel, widoków i ograniczeń.

Wskazówka: Polecenie `SELECT * FROM DICTIONARY`; wyświetli listę nazw widoków słowników systemowych zalogowanego użytkownika. Odszukaj na niej nazwy widoków potrzebnych do rozwiązania zadania (np.: `user_tables`).

8. Utwórz tabelę `osoba(id_osoba, imie, nazwisko, roczne_zarobki)` oraz dodaj do niej rekordy:

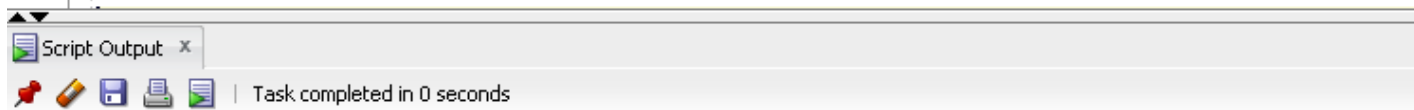
1 Jan Kowalski 8765,12; 2 Anna Nowak 6543,11; 3 Ewa Nowak 5555,55.

Zdefiniuj w oparciu o typ `NUMBER` własny typ pochodny o nazwie `KASA` i zadeklaruj zmienną tego typu. Wczytaj do niej roczne zarobki Jana Kowalskiego i wypisz je na ekranie.

## Rozwiązania

1. Zadeklaruj dwie zmienne o nazwach `v_tekst`, `v_liczba` i wartościach odpowiednio „Baza danych to uporządkowany zbiór danych” oraz 2,7182. Wyświetl wartości tych zmiennych na ekranie.

```
1  SET SERVEROUTPUT ON
2  /
3  DECLARE
4      v_tekst VARCHAR2(50) := 'Baza danych to uporządkowany zbiór danych';
5      v_liczba DECIMAL(5,4) := 2.7182;
6  BEGIN
7      DBMS_OUTPUT.PUT_LINE('v_tekst='||v_tekst);
8      --DBMS_OUTPUT.PUT_LINE('v_liczba='||to_char(v_liczba));
9      DBMS_OUTPUT.PUT_LINE('v_liczba='||v_liczba);
10 END;
```



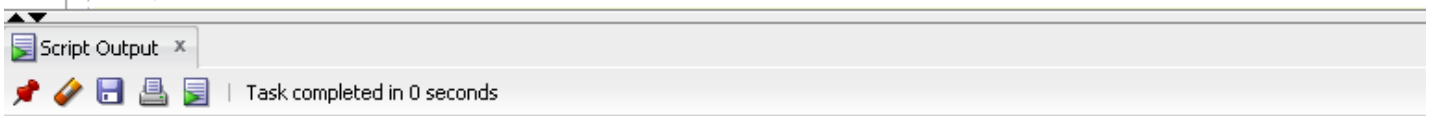
anonymous block completed

v\_tekst=Baza danych to uporządkowany zbiór danych

v\_liczba=2,7182

2. Napisz program obliczający iloczyn dwóch liczb całkowitych. Liczby powinny być podane z konsoli przez użytkownika.

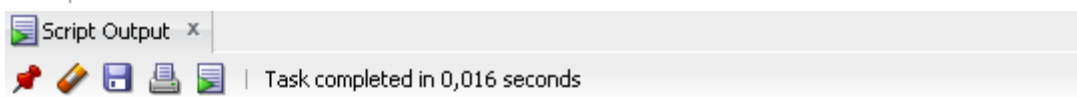
```
1 SET SERVEROUTPUT ON
2 /
3 DECLARE
4     v_liczba1 NUMBER(6) := &Podaj_pierwsza_liczbe;
5     v_liczba2 NUMBER(6) := &Podaj_druga_liczbe;
6     v_iloczyn NUMBER(12);
7 BEGIN
8     v_iloczyn:=v_liczba1*v_liczba2;
9     DBMS_OUTPUT.PUT_LINE(v_liczba1|| ' * '||v_liczba2||' = ' ||v_iloczyn);
10 END;
```



```
END;
new: DECLARE
    v_liczba1 NUMBER(6) := 2;
    v_liczba2 NUMBER(6) := 3;
    v_iloczyn NUMBER(12);
BEGIN
    v_iloczyn:=v_liczba1*v_liczba2;
    DBMS_OUTPUT.PUT_LINE(v_liczba1|| ' * '||v_liczba2||' = ' ||v_iloczyn);
END;
anonymous block completed
2 * 3 = 6
```

3. Napisać program obliczający pole powierzchni całkowitej i objętość kuli. W programie posłużyć się zdefiniowaną przez siebie stałą  $c\_pi=3,14159265$ . Wczytać promień kuli od użytkownika i wyświetlić wyniki na ekranie.

```
1  SET SERVEROUTPUT ON
2  /
3  DECLARE
4      c_pi CONSTANT NUMBER(16,8):=3.14159265;
5      v_r  NUMBER(16,8):=&Podaj_promien_kuli;
6      v_Pp NUMBER(16,8);
7      v_V  NUMBER(16,8);
8  BEGIN
9      v_Pp:=4*c_pi*v_r*v_r;
10     v_V:=4/3*c_pi*v_r*v_r*v_r;
11     DBMS_OUTPUT.PUT_LINE('Pole cakowite kuli = '||v_Pp);
12     DBMS_OUTPUT.PUT_LINE('Objętość kuli = '||v_V);
13 END;
```



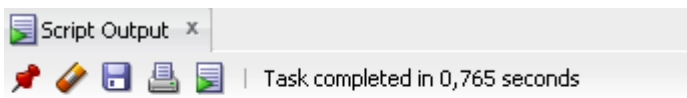
```
new: DECLARE
      c_pi CONSTANT NUMBER(16,8):=3.14159265;
      v_r  NUMBER(16,8):=1;
      v_Pp NUMBER(16,8);
      v_V  NUMBER(16,8);
BEGIN
      v_Pp:=4*c_pi*v_r*v_r;
      v_V:=4/3*c_pi*v_r*v_r*v_r;
      DBMS_OUTPUT.PUT_LINE('Pole cakowite kuli = '||v_Pp);
      DBMS_OUTPUT.PUT_LINE('Objętość kuli = '||v_V);
END;
anonymous block completed
Pole cakowite kuli = 12,5663706
Objętość kuli = 4,1887902
```

4. Utwórz tabelę pracownik(id\_pracownik, imie, nazwisko) z przykładowymi rekordami:

1 Jan Kowalski, 2 Jerzy Nowak, 3 Anna Gałka, 4 Hanna Miałka.

Zadeklaruj zmienną v\_ilosc, oblicz ilość rekordów w tabeli pracownik i otrzymany wynik zapisz do zmiennej v\_ilosc. Używając zmiennej v\_ilosc wypisz na ekranie komunikat w formacie: „W tabeli pracownik jest <ilość> rekordów”.

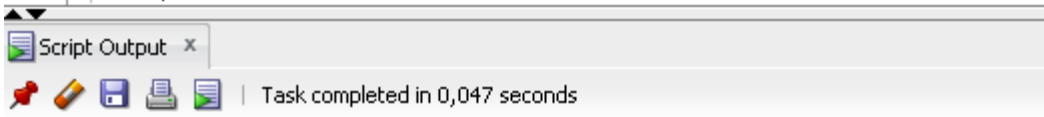
```
1 CREATE TABLE pracownik (  
2     id_pracownik NUMBER(11),  
3     imie VARCHAR2(15),  
4     nazwisko VARCHAR2(20)  
5 );  
6 /  
7 INSERT INTO pracownik VALUES (1, 'Jan', 'Kowalski');  
8 INSERT INTO pracownik VALUES (2, 'Jerzy', 'Nowak');  
9 INSERT INTO pracownik VALUES (3, 'Anna', 'Galka');  
10 INSERT INTO pracownik VALUES (4, 'Hanna', 'Mialka');  
11 /  
12 SET SERVEROUTPUT ON  
13 DECLARE  
14     v_ilosc NUMBER(11);  
15 BEGIN  
16     SELECT count(*) INTO v_ilosc FROM pracownik;  
17     DBMS_OUTPUT.PUT_LINE('W tabeli pracownik jest '||v_ilosc||' rekordów');  
18 END;
```



```
table PRACOWNIK created.  
1 rows inserted.  
1 rows inserted.  
1 rows inserted.  
1 rows inserted.  
anonymous block completed  
W tabeli pracownik jest 4 rekordów
```

5. Używając typu zakotwiczonego zadeklaruj zmienne v\_imie i v\_nazwisko o tych samych typach co typy kolumn imie i nazwisko z tabeli pracownik(id\_pracownik, imie, nazwisko). Następnie wczytaj do nich imię i nazwisko pracownika o identyfikatorze równym 3. Wyświetl na ekranie wczytane imię i nazwisko. Przyjmijmy, że pracownik o identyfikatorze równym 3 zawsze istnieje i jest dokładnie jeden taki pracownik.

```
1 SET SERVEROUTPUT ON
2 /
3 DECLARE
4     v_imie pracownik.imie%TYPE;
5     v_nazwisko pracownik.nazwisko%TYPE;
6 BEGIN
7     SELECT imie, nazwisko INTO v_imie, v_nazwisko
8     FROM pracownik WHERE id_pracownik=3;
9     DBMS_OUTPUT.PUT_LINE(v_imie||' '||v_nazwisko);
10 END;
```



anonymous block completed  
Anna Galka

6. Do zmiennej v\_rekord wczytaj rekord z danymi pracownika o identyfikatorze równym 2. (Zakładamy, że taki pracownik istnieje.) Wyświetl wczytane dane na ekranie w formacie: „Pracownik o identyfikatorze równym <numer> to <imie> <nazwisko>”. Następnie w zmiennej v\_rekord zmień wartość pola id\_pracownik na 10 i do tabeli pracownik dodaj nowy rekord o wartości zmiennej v\_rekord. Wyświetl na ekranie zawartość tabeli pracownik.

```
1 SET SERVEROUTPUT ON
2 DECLARE
3     v_rekord pracownik%ROWTYPE;
4 BEGIN
5     SELECT * INTO v_rekord
6     FROM pracownik WHERE id_pracownik=2;
7     DBMS_OUTPUT.PUT_LINE('Pracownik o identyfikatorze równym ' ||
8     v_rekord.id_pracownik || ' to ' || v_rekord.imie || ' ' || v_rekord.nazwisko);
9     v_rekord.id_pracownik:=10;
10    INSERT INTO pracownik VALUES v_rekord;
11 END;
12 /
13 SELECT * FROM pracownik;
```



anonymous block completed

Pracownik o identyfikatorze równym 2 to Jerzy Nowak

ID_PRACOWNIK	IMIE	NAZWISKO
1	Jan	Kowalski
2	Jerzy	Nowak
3	Anna	Galka
4	Hanna	Mialka
10	Jerzy	Nowak



7. Została utworzona tabela: `CREATE TABLE info(nazwa VARCHAR(50) NOT NULL, ilosc INT);` Dodaj do niej trzy rekordy, dla których kolumna nazwa przyjmuje odpowiednio wartości: tabele, widoki, ograniczenia (ang. constraints). Następnie zadeklaruj trzy zmienne i przypisz do nich odpowiednio wartości obliczeń ile zalogowany użytkownik ma tabel, widoków i ograniczeń w bazie danych. Zaktualizuj rekordy w tabeli info poprzez uaktualnienie wartości kolumny ilosc wpisując obliczoną wcześniej ilość tabel, widoków i ograniczeń.
- Wskazówka: Polecenie `SELECT * FROM DICTIONARY;` wyświetli listę nazw widoków słowników systemowych zalogowanego użytkownika. Odszukaj na niej nazwy widoków potrzebnych do rozwiązania zadania (np.: `user_tables`).

```

1 CREATE TABLE info(nazwa VARCHAR(50) NOT NULL, ilosc INT);
2 /
3 INSERT INTO info(nazwa) VALUES ('tabele');
4 INSERT INTO info(nazwa) VALUES ('widoki');
5 INSERT INTO info(nazwa) VALUES ('ograniczenia');
6 /
7 SET SERVEROUTPUT ON
8 DECLARE
9     v_ilosc_tabel INT;
10    v_ilosc_widokow INT;
11    v_ilosc_ograniczen INT;
12 BEGIN
13     SELECT count(*) INTO v_ilosc_tabel FROM user_tables;
14     SELECT count(*) INTO v_ilosc_widokow FROM user_views;
15     SELECT count(*) INTO v_ilosc_ograniczen FROM user_constraints;
16     UPDATE info SET ilosc=v_ilosc_tabel WHERE nazwa='tabele';
17     UPDATE info SET ilosc=v_ilosc_widokow WHERE nazwa='widoki';
18     UPDATE info SET ilosc=v_ilosc_ograniczen WHERE nazwa='ograniczenia';
19 END;
20 /
21 SELECT * FROM info;

```

Script Output x

Task completed in 0,953 seconds

table INFO created.

1 rows inserted.

1 rows inserted.

1 rows inserted.

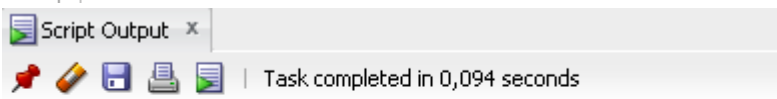
anonymous block completed

NAZWA	ILOSC
tabele	10
widoki	2
ograniczenia	31

8. Utwórz tabelę osoba(id\_osoba, imie, nazwisko, roczne\_zarobki) oraz dodaj do niej rekordy:  
1 Jan Kowalski 8765,12; 2 Anna Nowak 6543,11; 3 Ewa Nowak 5555,55.

Zdefiniuj w oparciu o typ NUMBER własny typ pochodny o nazwie KASA i zadeklaruj zmienną tego typu. Wczytaj do niej roczne zarobki Jana Kowalskiego i wypisz je na ekranie.

```
1 CREATE TABLE osoba (  
2     id_osoba NUMBER(11),  
3     imie VARCHAR2(15),  
4     nazwisko VARCHAR2(20),  
5     roczne_zarobki NUMBER(10,2)  
6 );  
7 /  
8 INSERT INTO osoba VALUES (1, 'Jan', 'Kowalski', 8765.12);  
9 INSERT INTO osoba VALUES (2, 'Anna', 'Nowak', 6543.11);  
10 INSERT INTO osoba VALUES (3, 'Ewa', 'Nowak', 5555.55);  
11 /  
12 SET SERVEROUTPUT ON  
13 DECLARE  
14     SUBTYPE KASA IS NUMBER(10,2);  
15     v_zarobki KASA;  
16 BEGIN  
17     SELECT roczne_zarobki INTO v_zarobki  
18     FROM osoba WHERE imie='Jan' AND nazwisko='Kowalski';  
19     DBMS_OUTPUT.PUT_LINE('Jan Kowalski rocznie zarabia '||v_zarobki||' zl');  
20 END;
```



```
table OSOBA created.  
1 rows inserted.  
1 rows inserted.  
1 rows inserted.  
anonymous block completed  
Jan Kowalski rocznie zarabia 8765,12 zl
```