

Zadania do realizacji

1. Zadeklaruj zmienną powiązaną o nazwie *v_wynik*.
Utwórz etykietę dla bloku zewnętrznego o nazwie *etykieta_zewnetrzna* oraz zadeklaruj zmienną o nazwie *v_liczba*. Następnie dla bloku wewnętrznego zadeklaruj zmienną o identycznej nazwie *v_liczba*. (Mamy dwie zmienne o nazwie *v_liczba* – w bloku zewnętrznym i wewnętrznym.)
Wykonaj mnożenie obu zmiennych *v_liczba* w bloku wewnętrznym z przypisaniem wyniku do zmiennej powiązanej *v_wynik*. Zakończ działanie bloku anonimowego. Wyświetl na ekranie wartość zmiennej *v_wynik*. Wartości obu zmiennych *v_liczba* pobierz od użytkownika.
2. Dana jest tabela z przykładowymi rekordami:

```
CREATE TABLE buty(id_buty NUMBER(11),  
typ VARCHAR2(20),marka VARCHAR2(20),rozmiar NUMBER(2));
```

```
INSERT INTO buty VALUES(1,'sportowe','nike',44);  
INSERT INTO buty VALUES(2,'sportowe','reebok',45);  
INSERT INTO buty VALUES(3,'pantofle','fly-high',46);
```


Wczytaj od użytkownika rozmiar buta, sprawdź czy taki rozmiar buta jest dostępny i wypisz na ekranie odpowiednią informację "Posiadamy buty w twoim rozmiarze, zapraszamy." lub "Nie posiadamy butów w twoim rozmiarze, przykro nam."
3. Pobierz od użytkownika numer miesiąca i wypisz na ekranie jego nazwę lub komunikat "nie ma takiego miesiąca". Użyj instrukcji CASE z wyrażeniem.
4. Dana jest tabela i przykładowe rekordy:

```
CREATE TABLE student (id_student NUMBER(11), imie VARCHAR2(15),  
nazwisko VARCHAR2(20), srednia NUMBER(4,2), stypendium NUMBER(8,2));
```

```
INSERT INTO student VALUES (1,'Jan','Kowalski',5,2000);  
INSERT INTO student VALUES (2,'Anna','Zdolna',4, 1000);  
INSERT INTO student VALUES (3,'Agata','Muza',3.5, 100);  
INSERT INTO student VALUES (4,'Anna','Kula',3, 1);  
INSERT INTO student VALUES (5,'Kacper','Adamek',2, 0);
```


Wczytaj identyfikator studenta. (Zakładamy, że zawsze będzie istniał wczytany identyfikator.)
Odczytaj wysokość stypendium wskazanego studenta i wypisz na ekranie komunikat:
a) jeśli stypendium jest najwyższe: "Najwyższe stypendium",
b) jeśli stypendium jest najniższe: "Najniższe stypendium",
c) jeśli nie jest to ani najwyższe ani najniższe stypendium to wypisz jeden z pasujących poniższych komunikatów:

c1) "Stypendium powyżej średniej",

c2) "Stypendium poniżej średniej",

c3) "Stypendium równe średniej".

Użyj instrukcji IF.

5. Wykonaj ponownie powyższe zadanie używając instrukcji CASE z wyszukiwaniem, nie używaj instrukcji IF.

6. Utwórz tabelę osoba(id_osoba,imie,nazwisko,pensja) zawierającą rekordy

1 Jan Kowalski 1900, 2 Anna Makurat 2100, 3 Jerzy Łoś 2300, 4 Ewa1 Kot1 2601,

4 Ewa2 Kot2 2602.

Wczytaj od użytkownika identyfikator osoby. Jeśli osoba o wczytanym identyfikatorze nie istnieje to wyświetl na ekranie komunikat "Osoba o id_osoba = <id_osoba> nie istnieje!", jeśli istnieje więcej niż jedna taka osoba, to wtedy wyświetl komunikat "Identyfikator id_osoba = <id_osoba> nie jest unikatowy!", jeśli istnieje dokładnie jedna taka osoba to wyświetl na ekranie wielkość pensji tej osoby oraz

a) w przypadku gdy ona posiada pensję mniejszą niż 2000 zł, to podwyższ jej pensję o 10%, odczytaj pensję po podwyżce i wyświetl komunikat "Pensja po podwyższeniu wynosi <pensja> zł",

b) w przypadku gdy ona posiada pensję wyższą niż 3000 zł, to obniż jej pensję o 2%, odczytaj pensję po obniżce i wyświetl komunikat "Pensja po obniżce wynosi <pensja> zł",

c) w przeciwnym przypadku wyświetl komunikat "Pensja pomiędzy 2000 zł a 3000 zł".

7. Napisz program, który będzie wyświetlał, w zależności od wyboru użytkownika, bieżącą datę systemową (wybór 'D') lub bieżący czas systemowy (wybór 'C').

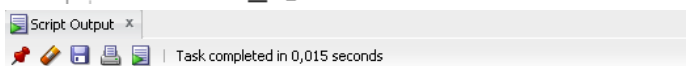
Rozwiązania

1. Zadeklaruj zmienną powiązaną o nazwie *v_wynik*.

Utwórz etykietę dla bloku zewnętrznego o nazwie *etykieta_zewnetrzna* oraz zadeklaruj zmienną o nazwie *v_liczba*. Następnie dla bloku wewnętrznego zadeklaruj zmienną o identycznej nazwie *v_liczba*. (Mamy dwie zmienne o nazwie *v_liczba* – w bloku zewnętrznym i wewnętrznym.)

Wykonaj mnożenie obu zmiennych *v_liczba* w bloku wewnętrznym z przypisaniem wyniku do zmiennej powiązanej *v_wynik*. Zakończ działanie bloku anonimowego. Wyświetl na ekranie wartość zmiennej *v_wynik*. Wartości obu zmiennych *v_liczba* pobierz od użytkownika.

```
1 VARIABLE v_wynik NUMBER;
2 /
3 SET SERVEROUTPUT ON
4 <<etykieta_zewnetrzna>>
5 DECLARE
6     v_liczba NUMBER:=&Podaj_liczbe_1;
7 BEGIN
8     DECLARE
9         v_liczba NUMBER:=&Podaj_liczbe_2;
10    BEGIN
11        :v_wynik:=etykieta_zewnetrzna.v_liczba*v_liczba;
12    END;
13 END;
14 /
15 PRINT :v_wynik;
```



```
new:<<etykieta_zewnetrzna>>
DECLARE
  v_liczba NUMBER:=2;
BEGIN
  DECLARE
    v_liczba NUMBER:=3;
  BEGIN
    :v_wynik:=etykieta_zewnetrzna.v_liczba*v_liczba;
  END;
END;
anonymous block completed
V_WYNIK
-
6
```

2. Dana jest tabela z przykładowymi rekordami: CREATE TABLE buty(id_buty NUMBER(11), typ VARCHAR2(20), marka VARCHAR2(20), rozmiar NUMBER(2));
INSERT INTO buty VALUES(1,'sportowe','nike',44);
INSERT INTO buty VALUES(2,'sportowe','reebok',45);
INSERT INTO buty VALUES(3,'pantofle','fly-high',46);
Wczytaj od użytkownika rozmiar buta, sprawdź czy taki rozmiar buta jest dostępny i wypisz na ekranie odpowiednią informację "Posiadamy buty w twoim rozmiarze, zapraszamy." lub "Nie posiadamy butów w twoim rozmiarze, przykro nam."

```
1 CREATE TABLE buty(  
2   id_buty NUMBER(11),  
3   typ VARCHAR2(20),  
4   marka VARCHAR2(20),  
5   rozmiar NUMBER(2)  
6 );  
7 /  
8 INSERT INTO buty VALUES(1,'sportowe','nike',44);  
9 INSERT INTO buty VALUES(2,'sportowe','reebok',45);  
10 INSERT INTO buty VALUES(3,'pantofle','fly-high',46);  
11 /  
12 SET SERVEROUTPUT ON  
13 DECLARE  
14   v_ilosc INT;  
15   v_rozmiar buty.rozmiar%TYPE:=&Podaj_szukany_rozmiar;  
16 BEGIN  
17   SELECT count(rozmiar) INTO v_ilosc FROM buty WHERE rozmiar=v_rozmiar;  
18 IF (v_ilosc>0) THEN  
19   DBMS_OUTPUT.PUT_LINE('Posiadamy buty w twoim rozmiarze, zapraszamy.');20 ELSE  
21   DBMS_OUTPUT.PUT_LINE('Nie posiadamy butów w twoim rozmiarze, przykro nam.');22 END IF;  
23 END;
```

Script Output x

Task completed in 0,11 seconds

```
new: DECLARE  
  v_ilosc INT;  
  v_rozmiar buty.rozmiar%TYPE:=44;  
BEGIN  
  SELECT count(rozmiar) INTO v_ilosc FROM buty WHERE rozmiar=v_rozmiar;  
  IF (v_ilosc>0) THEN  
    DBMS_OUTPUT.PUT_LINE('Posiadamy buty w twoim rozmiarze, zapraszamy.');  ELSE  
    DBMS_OUTPUT.PUT_LINE('Nie posiadamy butów w twoim rozmiarze, przykro nam.');  END IF;  
END;  
anonymous block completed  
Posiadamy buty w twoim rozmiarze, zapraszamy.
```

3. Pobierz od użytkownika numer miesiąca i wypisz na ekranie jego nazwę lub komunikat "nie ma takiego miesiąca". Użyj instrukcji CASE z wyrażeniem.

```
1 SET SERVEROUTPUT ON
2 DECLARE
3     v_miesiac INT := &Podaj_numer_miesiaca;
4     v_nazwa VARCHAR2(30);
5 BEGIN
6     CASE v_miesiac
7         WHEN 1 THEN v_nazwa:='styczeń';
8         WHEN 2 THEN v_nazwa:='luty';
9         WHEN 3 THEN v_nazwa:='marzec';
10        WHEN 4 THEN v_nazwa:='kwiecień';
11        WHEN 5 THEN v_nazwa:='maj';
12        WHEN 6 THEN v_nazwa:='czerwiec';
13        WHEN 7 THEN v_nazwa:='lipiec';
14        WHEN 8 THEN v_nazwa:='sierpień';
15        WHEN 9 THEN v_nazwa:='wrzesień';
16        WHEN 10 THEN v_nazwa:='październik';
17        WHEN 11 THEN v_nazwa:='listopad';
18        WHEN 12 THEN v_nazwa:='grudzień';
19        ELSE v_nazwa:='nie ma takiego miesiąca';
20    END CASE;
21    DBMS_OUTPUT.PUT_LINE(v_nazwa);
22 END;
```

4. Dana jest tabela i przykładowe rekordy:

```
CREATE TABLE student (id_student NUMBER(11), imie VARCHAR2(15),  
nazwisko VARCHAR2(20), srednia NUMBER(4,2), stypendium NUMBER(8,2));
```

```
INSERT INTO student VALUES (1,'Jan','Kowalski',5,2000);
```

```
INSERT INTO student VALUES (2,'Anna','Zdolna',4, 1000);
```

```
INSERT INTO student VALUES (3,'Agata','Muza',3.5, 100);
```

```
INSERT INTO student VALUES (4,'Anna','Kula',3, 1);
```

```
INSERT INTO student VALUES (5,'Kacper','Adamek',2, 0);
```

Wczytaj identyfikator studenta. (Zakładamy, że zawsze będzie istniał wczytany identyfikator.)

Odczytaj wysokość stypendium wskazanego studenta i wypisz na ekranie komunikat:

a) jeśli stypendium jest najwyższe: "Najwyższe stypendium",

b) jeśli stypendium jest najniższe: "Najniższe stypendium",

c) jeśli nie jest to ani najwyższe ani najniższe stypendium to wypisz jeden z pasujących poniższych komunikatów:

c1) "Stypendium powyżej średniej",

c2) "Stypendium poniżej średniej",

c3) "Stypendium równe średniej".

Użyj instrukcji IF.

```
1 CREATE TABLE student (  
2     id_student NUMBER(11),  
3     imie VARCHAR2(15),  
4     nazwisko VARCHAR2(20),  
5     srednia NUMBER(4,2),  
6     stypendium NUMBER(8,2)  
7 );  
8 /  
9 INSERT INTO student VALUES (1, 'Jan', 'Kowalski', 5, 2000);  
10 INSERT INTO student VALUES (2, 'Anna', 'Zdolna', 4, 1000);  
11 INSERT INTO student VALUES (3, 'Agata', 'Muza', 3.5, 100);  
12 INSERT INTO student VALUES (4, 'Anna', 'Kula', 3, 1);  
13 INSERT INTO student VALUES (5, 'Kacper', 'Adamek', 2, 0);  
14 /
```

```
15 SET SERVEROUTPUT ON;
16 DECLARE
17     v_srednia_studenta student.srednia%TYPE;
18     v_id_student student.id_student%TYPE := &Podaj_id_studenta;
19     v_max NUMBER;
20     v_min NUMBER;
21     v_avg NUMBER;
22 BEGIN
23     SELECT srednia INTO v_srednia_studenta FROM student WHERE id_student=v_id_student;
24     SELECT max(srednia), min(srednia), avg(srednia) INTO v_max, v_min, v_avg FROM student;
25     IF v_srednia_studenta = v_max THEN
26         DBMS_OUTPUT.PUT_LINE('Najwyższe stypendium');
27     ELSIF v_srednia_studenta = v_min THEN
28         DBMS_OUTPUT.PUT_LINE('Najniższe stypendium');
29     ELSIF v_srednia_studenta > v_avg THEN
30         DBMS_OUTPUT.PUT_LINE('Stypendium powyżej średniej');
31     ELSIF v_srednia_studenta < v_avg THEN
32         DBMS_OUTPUT.PUT_LINE('Stypendium poniżej średniej');
33     ELSE
34         DBMS_OUTPUT.PUT_LINE('Stypendium równe średniej');
35     END IF;
36 END;
```

5. Wykonaj ponownie powyższe zadanie używając instrukcji CASE z wyszukiwaniem, nie używaj instrukcji IF.

```
1 SET SERVEROUTPUT ON;
2 DECLARE
3     v_srednia_studenta student.srednia%TYPE;
4     v_id_student student.id_student%TYPE := &Podaj_id_studenta;
5     v_max NUMBER;
6     v_min NUMBER;
7     v_avg NUMBER;
8 BEGIN
9     SELECT srednia INTO v_srednia_studenta FROM student WHERE id_student=v_id_student;
10    SELECT max(srednia), min(srednia), avg(srednia) INTO v_max, v_min, v_avg FROM student;
11 CASE
12     WHEN v_srednia_studenta = v_max THEN DBMS_OUTPUT.PUT_LINE('Najwyższe stypendium');
13     WHEN v_srednia_studenta = v_min THEN DBMS_OUTPUT.PUT_LINE('Najniższe stypendium');
14     WHEN v_srednia_studenta > v_avg THEN DBMS_OUTPUT.PUT_LINE('Stypendium powyżej średniej');
15     WHEN v_srednia_studenta < v_avg THEN DBMS_OUTPUT.PUT_LINE('Stypendium poniżej średniej');
16     ELSE DBMS_OUTPUT.PUT_LINE('Stypendium równe średniej');
17 END CASE;
18 END;
```


6. Utwórz tabelę osoba(id_osoba, imie, nazwisko, pensja) zawierającą rekordy
1 Jan Kowalski 1900, 2 Anna Makurat 2100, 3 Jerzy Łoś 2300, 4 Ewa1 Kot1 2601,
4 Ewa2 Kot2 2602.

Wczytaj od użytkownika identyfikator osoby. Jeśli osoba o wczytanym identyfikatorze nie istnieje to wyświetl na ekranie komunikat "Osoba o id_osoba = <id_osoba> nie istnieje!", jeśli istnieje więcej niż jedna taka osoba, to wtedy wyświetl komunikat "Identyfikator id_osoba = <id_osoba> nie jest unikatowy!", jeśli istnieje dokładnie jedna taka osoba to wyświetl na ekranie wielkość pensji tej osoby oraz

a) w przypadku gdy ona posiada pensję mniejszą niż 2000 zł, to podwyższ jej pensję o 10%, odczytaj pensję po podwyżce i wyświetl komunikat "Pensja po podwyższeniu wynosi <pensja> zł",

b) w przypadku gdy ona posiada pensję wyższą niż 3000 zł, to obniż jej pensję o 2%, odczytaj pensję po obniżce i wyświetl komunikat "Pensja po obniżce wynosi <pensja> zł",

c) w przeciwnym przypadku wyświetl komunikat "Pensja pomiędzy 2000 zł a 3000 zł".

```
1 DROP TABLE osoba;
2 /
3 CREATE TABLE osoba(
4     id_osoba NUMBER(11),
5     imie VARCHAR2(15),
6     nazwisko VARCHAR2(20),
7     pensja NUMBER(8,2)
8 );
9 /
10 INSERT INTO osoba(id_osoba, imie, nazwisko, pensja) VALUES (1, 'Jan', 'Kowalski', 1900);
11 INSERT INTO osoba(id_osoba, imie, nazwisko, pensja) VALUES (2, 'Anna', 'Makurat', 2500);
12 INSERT INTO osoba(id_osoba, imie, nazwisko, pensja) VALUES (3, 'Jerzy', 'Łoś', 3100);
13 INSERT INTO osoba(id_osoba, imie, nazwisko, pensja) VALUES (4, 'Ewa1', 'Kot1', 3601);
14 INSERT INTO osoba(id_osoba, imie, nazwisko, pensja) VALUES (4, 'Ewa2', 'Kot2', 3602);
```

```
16 SET SERVEROUTPUT ON
17 DECLARE
18     v_ile_osob INT;
19     v_id osoba.id_osoba%TYPE := &Podaj_id_osoby;
20     v_pensja osoba.pensja%TYPE;
21 BEGIN
22     SELECT count(*) INTO v_ile_osob from osoba WHERE id_osoba=v_id;
23     IF v_ile_osob=0 THEN
24         DBMS_OUTPUT.PUT_LINE('Osoba o id_osoba = '||v_id||' nie istnieje!');
25     ELSIF v_ile_osob>1 THEN
26         DBMS_OUTPUT.PUT_LINE('Identyfikator id_osoba = '||v_id||' nie jest unikatowy!');
27     ELSE
28     BEGIN
29         SELECT pensja INTO v_pensja FROM osoba WHERE id_osoba=v_id;
30         IF v_pensja<2000 THEN
31             DBMS_OUTPUT.PUT_LINE('Pensja wynosi '||v_pensja||' zł');
32             UPDATE osoba SET pensja=v_pensja*1.1 WHERE id_osoba=v_id;
33             SELECT pensja INTO v_pensja FROM osoba WHERE id_osoba=v_id;
34             DBMS_OUTPUT.PUT_LINE('Pensja po podwyższeniu wynosi '||v_pensja||' zł');
35         ELSIF v_pensja>3000 THEN
36             DBMS_OUTPUT.PUT_LINE('Pensja wynosi '||v_pensja||' zł');
37             UPDATE osoba SET pensja=v_pensja*0.98 WHERE id_osoba=v_id;
38             SELECT pensja INTO v_pensja FROM osoba WHERE id_osoba=v_id;
39             DBMS_OUTPUT.PUT_LINE('Pensja po obniżeniu wynosi '||v_pensja||' zł');
40         ELSE
41             DBMS_OUTPUT.PUT_LINE('Pensja pomiędzy 2000 zł a 3000 zł');
42         END IF;
43     END;
44     END IF;
45 END;
```

7. Napisz program, który będzie wyświetlał, w zależności od wyboru użytkownika, bieżącą datę systemową (wybór 'D') lub bieżący czas systemowy (wybór 'C').

```
1  SET SERVEROUTPUT ON
2  /
3  DECLARE
4      v_wybor CHAR(1) := '&Twoj_wybor';
5  BEGIN
6      IF UPPER(v_wybor)='D' THEN
7          DBMS_OUTPUT.PUT_LINE('Data systemowa: '||to_char(sysdate,'yyyy-mm-dd'));
8      ELSIF UPPER(v_wybor)='C' THEN
9          DBMS_OUTPUT.PUT_LINE('Czas systemowy: '||to_char(sysdate,'hh24:mi:ss'));
10     ELSE
11         DBMS_OUTPUT.PUT_LINE('Nieznany wybór');
12     END IF;
13 END;
```