Laboratorium 1

Bazy Danych 2

*mgr. inż. Aleksander Wojtowicz*

**Typy danych w PL/SQL**

Poniżej zostały przedstawione najważniejsze typy danych w PL/SQL. Przechowują one:

**1. Typy numeryczne:**

**NUMBER**: Liczby zmiennoprzecinkowe lub całkowite.

**PLS\_INTEGER** i **BINARY\_INTEGER**: Liczby całkowitych.

**BINARY\_FLOAT** i **BINARY\_DOUBLE**: Liczby zmiennoprzecinkowe pojedynczej lub podwójnej precyzji.

**2. Typy znakowe:**

**VARCHAR2(n):** Ciąg znaków o zmiennej długości.

**CHAR(n):** Ciąg znaków o stałej długości.

**3. Typy logiczne:**

**BOOLEAN:** Wartości logiczne.

**4. Typy daty i czasu:**

**DATE:** Data i czas.

**TIMESTAMP:** Data i czas z precyzją do nanosekund.

**5. Typy LOB (Large Object):**

**BLOB, CLOB, NCLOB:** Duże obiekty binarnych i znakowych.

**6. Typy referencyjne:**

**REF CURSOR:** Kursor, który pozwala na manipulację wynikami zapytania SQL.

**7. Typy złożone:**

**RECORD:** Struktura danych składająca się z zestawu pól, które mogą mieć różne typy.

**TABLE:** Kolekcja elementów tego samego typu.

**VARRAY:** Kolekcja o zmiennej wielkości elementów tego samego typu.

**Deklaracja zmiennych w PL/SQL**

DECLARE

-- Deklaracja zmiennej typu NUMBER

moja\_liczba NUMBER := 10;

-- Deklaracja zmiennej typu VARCHAR2

moj\_tekst VARCHAR2(100) := 'Witaj w PL/SQL!';

-- Deklaracja zmiennej typu BOOLEAN

moj\_stan BOOLEAN := TRUE;

-- Deklaracja zmiennej typu DATE

moja\_data DATE := SYSDATE;

BEGIN

-- reszta programu

END;

*Ten kod inicjalizuje cztery zmienne: liczbę, tekst, wartość logiczną i datę, które mogą być później wykorzystane w programie.*

**Wyświetlenie zmiennych**

**DECLARE**

moja\_liczba NUMBER := 10;

moj\_tekst VARCHAR2(100) := 'Witaj w PL/SQL!';

moja\_data DATE := SYSDATE;

**BEGIN**

-- Wypisanie wartości zmiennych

DBMS\_OUTPUT.PUT\_LINE('Moja liczba to: ' || moja\_liczba);

DBMS\_OUTPUT.PUT\_LINE('Mój tekst to: ' || moj\_tekst);

DBMS\_OUTPUT.PUT\_LINE('Moja data to: ' || TO\_CHAR(moja\_data, 'DD-MM-YYYY'));

**END**;

*Ten kod wyświetla wartości zainicjalizowanych zmiennych. Wykorzystuje funkcję* ***DBMS\_OUTPUT.PUT\_LINE*** *do wypisania wartości zmiennych. W zależności od wartości zmiennej moj\_stan, wypisze ‘Mój stan to: PRAWDA’ lub ‘Mój stan to: FAŁSZ’.*

**UWAGA:** Jeśli żadne zmienne nie zostaną wyświetlone, należy wprowadzić komendę **set serveroutput on** jako osobne polecenie. Następnie należy ponownie wpisać powyższy kod.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

**Instrukcje warunkowe IF**

Jedną z pierwszych instrukcji sterujących jest instrukcja IF.

**IF THEN:** Wykonuje instrukcje, jeśli warunek jest prawdziwy.

**IF THEN ELSE:** Wykonuje jeden zestaw instrukcji, jeśli warunek jest prawdziwy, a inny zestaw, jeśli warunek jest fałszywy.

**IF THEN ELSIF:** Pozwala na sprawdzenie wielu warunków i wykonanie różnych instrukcji dla każdego z nich.

Przykład kodu z instrukcją **IF THEN**:

**DECLARE**

moja\_liczba NUMBER := 10;

**BEGIN**

**IF** moja\_liczba > 5 **THEN**

DBMS\_OUTPUT.PUT\_LINE('Moja liczba jest większa od 5');

**END** **IF**;

**END**;



Przykład kodu z instrukcją **IF THEN ELSE**:

**DECLARE**

moja\_liczba NUMBER := 5;

**BEGIN**

**IF** moja\_liczba > 5 **THEN**

DBMS\_OUTPUT.PUT\_LINE('Moja liczba jest większa od 5');

**ELSE**

DBMS\_OUTPUT.PUT\_LINE('Moja liczba jest mniejsza lub równa 5');

**END** **IF**;

**END**;



Przykład funkcji **IF THEN ELSIF**:

**DECLARE**

moja\_liczba NUMBER := 5;

**BEGIN**

**IF** moja\_liczba > 10 **THEN**

DBMS\_OUTPUT.PUT\_LINE('Moja liczba jest większa od 10');

**ELSIF** moja\_liczba = 10 **THEN**

DBMS\_OUTPUT.PUT\_LINE('Moja liczba jest równa 10');

**ELSE**

DBMS\_OUTPUT.PUT\_LINE('Moja liczba jest mniejsza od 10');

**END IF**;

**END**;



**Zadania:**

1. **Deklaracja i inicjalizacja zmiennych:** Napisz blok PL/SQL, który deklaruje i inicjalizuje następujące zmienne:
   1. moj\_wiek typu NUMBER ze swoim wiekiem.
   2. moje\_imie typu VARCHAR2 ze swoim imieniem.
   3. jestem\_studentem typu BOOLEAN.
2. **Wypisywanie zmiennych:** Dodaj do powyższego bloku kodu instrukcje, które wypisują wartości zmiennych moj\_wiek, moje\_imie i jestem\_studentem.
3. **Instrukcje warunkowe IF THEN:** Napisz blok PL/SQL, który sprawdza, czy osoba jest pełnoletnia. Jeżeli nie, wypisz odpowiednią wiadomość.
4. **Instrukcje warunkowe IF THEN ELSE**: Zmodyfikuj powyższy blok kodu, dodając instrukcję ELSE, aby sprawdzać, czy osoba jest pełnoletnia, czy nie. Wypisz wiadomość w obu przypadkach.
5. **Instrukcje warunkowe IF THEN ELSIF:** Napisz kod, który sprawdza wiek osoby oraz czy jest nieletnią, osobą dorosłą czy emerytem. Sprawdź również, czy osoba jest studentem, a następnie wyświetl wiadomość zawierającą te informacje.

*Przykład:*

Dla osoby o imieniu Adam, która ma 21 lat i jest studentem, program powinien wygenerować wiadomość:

*Adam ma 21 lat, jest pełnoletni i jest studentem.*