Laboratorium 2

Bazy Danych 2

*mgr. inż. Aleksander Wojtowicz*

**Instrukcje warunkowe CASE**

DECLARE  
 numer\_dnia NUMBER := 3;  
 dzien VARCHAR2(20);  
BEGIN  
 CASE numer\_dnia  
 WHEN 1 THEN dzien := 'Poniedziałek';  
 WHEN 2 THEN dzien := 'Wtorek';  
 WHEN 3 THEN dzien := 'Środa';  
 WHEN 4 THEN dzien := 'Czwartek';  
 WHEN 5 THEN dzien := 'Piątek';  
 ELSE dzien := 'Nieznany dzień';  
 END CASE;  
  
 DBMS\_OUTPUT.*PUT\_LINE*('Dzień tygodnia: ' || dzien);  
END;



*CASE w PL/SQL jest używana do wykonywania sekwencji instrukcji warunkowych. Działa to podobnie do instrukcji if-else w innych językach programowania.*

1. **CASE numer\_dnia**: Ta linia rozpoczyna instrukcję CASE i mówi, że będziemy sprawdzać wartość zmiennej numer\_dnia.
2. **WHEN 1 THEN dzien := 'Poniedziałek'** : Jeśli numer\_dnia jest równe 1, to dzien zostanie ustawione na 'Poniedziałek'.
3. **ELSE dzien := 'Nieznany dzień'** : Jeśli żaden z poprzednich warunków nie jest spełniony *(tzn. numer\_dnia nie jest żadnym z numerów od 1 do 5),* to dzien zostanie ustawione na 'Nieznany dzień'.
4. **END CASE**: Ta linia kończy instrukcję CASE.

**Instrukcje warunkowe LOOP**

DECLARE  
 suma NUMBER := 0;  
BEGIN  
 FOR i IN 1..5 LOOP  
 suma := suma + i;  
 END LOOP;  
  
 DBMS\_OUTPUT.PUT\_LINE('Suma liczb od 1 do 5: ' || suma);  
END;



**FOR i IN 1..5 LOOP ... END LOOP**: Pętla FOR, która wykonuje blok kodu dla każdej wartości **i** od 1 do 5.

**Instrukcje warunkowe WHILE LOOP:**

DECLARE  
 licznik NUMBER := 1;  
 maksimum NUMBER := 5;  
BEGIN  
 WHILE licznik <= maksimum LOOP  
 DBMS\_OUTPUT.*PUT\_LINE*('Aktualny licznik: ' || licznik);  
 licznik := licznik + 1;  
 END LOOP;  
END;

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

**WHILE licznik <= maksimum LOOP**: Pętla WHILE wykonuje się dopóki warunek licznik <= maksimum jest prawdziwy.

**licznik := licznik + 1;**: Zwiększa wartość licznik o 1 w każdym obiegu pętli.

**Instrukcje warunkowe EXIT WHEN**

DECLARE  
 suma NUMBER := 0;  
 maksymalna\_suma NUMBER := 100;  
 liczba NUMBER;  
BEGIN  
 FOR i IN 1..10 LOOP  
 liczba := DBMS\_RANDOM.VALUE(1, 20);  
 DBMS\_OUTPUT.PUT\_LINE('Losowa liczba: ' || TO\_CHAR(liczba, '999.9999'));  
  
 suma := suma + liczba;  
  
 EXIT WHEN suma > maksymalna\_suma;  
 END LOOP;  
  
 DBMS\_OUTPUT.PUT\_LINE('Łączna suma: ' || TO\_CHAR(suma, '999.9999'));  
END;

**EXIT WHEN suma > maksymalna\_suma;**:Ta linia powoduje wyjście z pętli, gdy suma przekroczy maksymalna\_suma.

**TO\_CHAR(liczba, '999.9999')** konwertuje liczbę na ciąg znaków. Format '999.9999' oznacza trzy cyfry przed i cztery cyfry po przecinku. Na przykład, jeśli liczba wynosi 12.3, to TO\_CHAR(liczba, '999.9999') zwróci ciąg znaków 12.3000

**Przykład tworzenia bazy i podstawowe operacje**

*-- Tworzenie tabeli*CREATE TABLE pracownicy (  
 id\_pracownika NUMBER PRIMARY KEY,  
 imie VARCHAR2(50),  
 nazwisko VARCHAR2(50),  
 pensja NUMBER  
);  
  
*-- Dodawanie danych*INSERT INTO pracownicy VALUES (1, 'Jan', 'Kowalski', 50000);  
INSERT INTO pracownicy VALUES (2, 'Anna', 'Nowak', 60000);  
INSERT INTO pracownicy VALUES (3, 'Piotr', 'Wiśniewski', 55000);  
  
*-- Aktualizacja danych*UPDATE pracownicy SET pensja = 58000 WHERE id\_pracownika = 1;  
  
*-- Zapytanie o dane*SELECT \* FROM pracownicy;  
  
*-- Usuwanie danych*DELETE FROM pracownicy WHERE id\_pracownika = 3;  
  
*-- Ponowne zapytanie o dane po usunięciu*SELECT \* FROM pracownicy;

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

**Prosty kursor do wyświetlenia wszystkich pracowników**

DECLARE  
 CURSOR pracownicy\_cursor IS  
 SELECT \* FROM pracownicy;  
BEGIN  
 FOR pracownik\_rec IN pracownicy\_cursor LOOP  
 DBMS\_OUTPUT.*PUT\_LINE*('ID: ' || pracownik\_rec.id\_pracownika || ', Imię: ' || pracownik\_rec.imie || ', Nazwisko: ' || pracownik\_rec.nazwisko || ', Pensja: ' || pracownik\_rec.pensja);  
 END LOOP;  
END;

**Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie**

**CURSOR pracownicy\_cursor IS SELECT \* FROM pracownicy**; deklaruje kursor o nazwie pracownicy\_cursor, który przechowuje wszystkie rekordy z tabeli pracownicy.

Kursor w PL/SQL to jakby “wskaźnik” do wyników zapytania SQL. Pozwala przeglądać wyniki zapytania wiersz po wierszu.

**Kursor z warunkiem do wyświetlenia pracowników z pensją powyżej określonej wartości**

DECLARE  
 CURSOR pracownicy\_cursor IS  
 SELECT \* FROM pracownicy WHERE pensja > 55000;  
BEGIN  
 FOR pracownik\_rec IN pracownicy\_cursor LOOP  
 DBMS\_OUTPUT.*PUT\_LINE*('ID: ' || pracownik\_rec.id\_pracownika || ', Imię: ' || pracownik\_rec.imie || ', Nazwisko: ' || pracownik\_rec.nazwisko || ', Pensja: ' || pracownik\_rec.pensja);  
 END LOOP;  
END;

**Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie**

**CURSOR pracownicy\_cursor IS SELECT \* FROM pracownicy WHERE pensja > 55000;:** Tworzy kursor o nazwie pracownicy\_cursor, który przechowuje wszystkie rekordy z tabeli pracownicy, gdzie pensja jest większa niż 55000.

**Kursor z parametrem do wyświetlenia pracownika o określonym ID**

DECLARE  
 CURSOR pracownik\_cursor (v\_employee\_id NUMBER) IS  
 SELECT \* FROM pracownicy WHERE id\_pracownika = v\_employee\_id;  
 *-- ID pracownika do wyszukania*  
v\_id\_pracownika pracownicy.id\_pracownika%TYPE := 2;  
  
  
BEGIN  
 FOR pracownik\_rec IN pracownik\_cursor(v\_id\_pracownika) LOOP  
 DBMS\_OUTPUT.*PUT\_LINE*('ID: ' || pracownik\_rec.id\_pracownika || ', Imię: ' || pracownik\_rec.imie || ', Nazwisko: ' || pracownik\_rec.nazwisko || ', Pensja: ' || pracownik\_rec.pensja);  
 END LOOP;  
END;



**CURSOR pracownik\_cursor (v\_employee\_id NUMBER) IS SELECT \* FROM pracownicy WHERE id\_pracownika = v\_employee\_id;**: Tworzy kursor parametryzowany o nazwie pracownik\_cursor, który przechowuje wszystkie rekordy z tabeli pracownicy, gdzie id\_pracownika jest równe 2.

**Zadania:**

* **Instrukcja CASE:** Napisz procedurę, która przyjmuje jako parametr liczbę i zwraca jej wartość w systemie rzymskim (do 10).
* **Pętla CASE:** Napisz procedurę, która przyjmuje jako parametr liczbę i wyświetla, czy jest ona parzysta czy nieparzysta.

**Instrukcja CASE:** procedurę, która przyjmuje jako parametr miesiąc (jako liczbę) i wyświetla, ile dni ma ten miesiąc.

* **Pętla WHILE:** Napisz procedurę, która wyświetla wszystkie liczby od podanej liczby do 1 w porządku malejącym.
* **Pętla LOOP i EXIT WHEN:** Napisz procedurę, która wyświetla wszystkie liczby od 1 do podanej liczby, ale przerywa pętlę, gdy iloczyn tych liczb przekroczy 12345.

*Dla poniższych zadań należy dodać dodatkowych pracowników do bazy Pracownicy*

* **Kursory:** Napisz procedurę, która używa kursora do zwiększenia pensji o 10% dla pracowników, którzy zarabiają mniej niż 60000.
* **Kursory:** Napisz procedurę, która przyjmuje nazwisko jako parametr i używa kursora do wyświetlenia wszystkich pracowników o tym nazwisku.
* **Kursory:** Napisz funkcję, która używa kursora do zliczenia liczby pracowników, którzy zarabiają więcej niż średnia pensja wśród wszystkich pracowników.