

Laboratorium 4

Bazy Danych 2

mgr. inż. Aleksander Wojtowicz

Funkcje, jak i **procedury** są podprogramami, które umożliwiają grupowanie instrukcji do wykonania określonego zadania.

Zwracanie wartości:

- **Funkcja** zawsze zwraca wartość. W definicji funkcji musi być określony typ zwracanej wartości.
- **Procedura** nie zwraca wartości. Może jednak zwrócić wiele wyników za pomocą parametrów wyjściowych.

Wywołanie:

- **Funkcję** można wywołać jako część wyrażenia.
Na przykład, może być użyta w instrukcji SELECT, przypisaniu do zmiennej, itp.
- **Procedurę** wywołuje się jako osobną instrukcję w bloku PL/SQL.

Zastosowanie:

- **Funkcje** są zazwyczaj używane, gdy potrzebujemy obliczyć i zwrócić wartość.
- **Procedury** są używane do wykonania określonych działań, takich jak modyfikacje w bazie danych.

```
-- Przykładowa funkcja
CREATE OR REPLACE FUNCTION kwadrat_liczby (n NUMBER) RETURN
NUMBER IS
BEGIN
    RETURN n*n;
END;

--Przykładowe wywołanie funkcji
DECLARE
    num NUMBER;
    result NUMBER;
BEGIN
    num := 4;
    result := kwadrat_liczby(num);
    DBMS_OUTPUT.PUT_LINE('Kwadrat liczby ' || num || ' wynosi '
|| result);
END;
```

Kwadrat liczby 4 wynosi 16

W powyższym przykładzie, zmienna `num` jest ustawiona na 4, a następnie **funkcja** `kwadrat_liczby` jest wywoływana z `num` jako argumentem. Wynik jest przechowywany w zmiennej `result` i wyświetlany za pomocą `DBMS_OUTPUT.PUT_LINE`.

```
-- Przykładowa funkcja
CREATE OR REPLACE FUNCTION kwadrat_liczby (n NUMBER) RETURN
NUMBER IS
BEGIN
    RETURN n*n;
END;

-- Przykładowa procedura
CREATE OR REPLACE PROCEDURE wypis_kwadratu(n NUMBER) IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Kwadrat liczby ' || TO_CHAR(n) || '
wynosi ' || TO_CHAR(kwadrat_liczby(n)));
END;

-- Wywołanie procedury
DECLARE
    num NUMBER;
BEGIN
    num := 4;
    wypis_kwadratu(num);
END;

Kwadrat liczby 4 wynosi 16
```

W tym kodzie, najpierw tworzymy **funkcję** `kwadrat_liczby(n)`, która zwraca kwadrat liczby `n`. Następnie tworzymy **procedurę** `wypis_kwadratu(n)`, która wywołuje **funkcję** `kwadrat_liczby(n)` i wyświetla wynik. Na końcu wywołujemy **procedurę** `wypis_kwadratu(n)` z argumentem `num`, który jest ustawiony na 4. Jeśli wszystko jest poprawne, powinno to wyświetlić "Kwadrat liczby 4 wynosi 16".

PL/SQL oferuje wiele wbudowanych funkcji do manipulacji danymi. Oto kilka przykładów:

- `UPPER(string)`: Konwertuje wszystkie litery w ciągu na wielkie.
- `LOWER(string)`: Konwertuje wszystkie litery w ciągu na małe.
- `SUBSTR(string, start, length)`: Zwraca podciąg ciągu zaczynający się od pozycji `start` o długości `length`.
- `LENGTH(string)`: Zwraca długość ciągu.

Modyfikowanie funkcji lub procedury w PL/SQL jest proste. Wystarczy użyć klauzuli `CREATE OR REPLACE` podczas tworzenia funkcji lub procedury.

```
CREATE OR REPLACE FUNCTION kwadrat(n NUMBER) RETURN NUMBER IS
BEGIN
    RETURN n*n;
END;
```

Usuwanie funkcji lub procedury używa instrukcji `DROP`. Na przykład:

```
--Przykład usunięcia funkcji
DROP FUNCTION kwadrat;
```

Podsumowanie

- **Funkcje i procedury:** Funkcje i procedury w PL/SQL są podprogramami, które umożliwiają grupowanie instrukcji do wykonania określonego zadania. Główna różnica między nimi polega na tym, że funkcje zwracają wartość, a procedury nie.
- **Tworzenie i wywoływanie funkcji:** Funkcje tworzymy za pomocą składni `CREATE FUNCTION`, a wywołujemy je, używając ich nazwy w wyrażeniu.
- **Modyfikowanie i usuwanie funkcji i procedur:** Modyfikowanie funkcji lub procedury jest możliwe za pomocą klauzuli `CREATE OR REPLACE`. Usuwanie funkcji lub procedury odbywa się za pomocą instrukcji `DROP`.

Zadania:

1. Napisz funkcję, która przyjmuje ciąg znaków i zwraca jego długość.
2. Napisz funkcję, która przyjmuje dwie liczby i zwraca ich sumę.
3. Napisz funkcję, która przyjmuje datę urodzenia i zwraca wiek osoby.
4. Napisz funkcję, która przyjmuje ciąg znaków i zwraca ten sam ciąg, ale z odwróconą kolejnością znaków.
5. Napisz funkcję, która przyjmuje ciąg znaków i zwraca TRUE, jeśli ciąg jest palindromem, a FALSE w przeciwnym razie.
Palindrom to słowo, które czytane od przodu i od tyłu brzmi tak samo.
6. Napisz funkcję, która oblicza pierwiastek kwadratowy liczby. Następnie zmodyfikuj tę funkcję, aby obliczała pierwiastek sześcienny.
7. Napisz procedurę, która wyświetla wiadomość powitalną. Następnie zmodyfikuj tę procedurę, aby wyświetlała wiadomość pożegnalną.
8. Napisz funkcję, która dzieli dwie liczby. Dodaj obsługę błędów, aby uniknąć dzielenia przez zero.
9. Napisz procedurę, która aktualizuje rekordy w tabeli. Dodaj obsługę błędów, aby obsłużyć sytuację, gdy aktualizowany rekord nie istnieje.
10. Usuń jedną z napisanych funkcji lub procedur.