

| | |
|---|---|
| Instrukcja laboratoryjna 6 | Bazy Danych |
| | Temat: Procedury i funkcje |
| | Przygotowali: mgr inż. Barbara Łukawska, mgr inż. Maciej Lasota |

1) Procedury

Tworzenie procedury:

```
CREATE [OR REPLACE] PROCEDURE nazw_procedury
    [(argument [IN | OUT | IN OUT] typ_danych, ... , ... )]
IS
    [sekcja_deklaracyjna]
BEGIN
    [instrukcje];
    [obsługa_wyjątków]
END;
```

gdzie:

- ❖ **OR REPLACE** - modyfikuje/zastępuje istniejącą procedurę,
- ❖ **IN | OUT | IN OUT** - określa sposób przekazywania parametrów do odczytu, do zapisu, do odczytu i do zapisu (domyślnie).

Polecenie kompilacji procedury: `"/`

Wyświetlanie błędów procedury: `SHOW ERRORS [PROCEDURE nazwa_procedury]`

Wywoływanie procedury:

```
nazwa_procedury[(parametr1,parametr2,...)];
EXEC[UTE] nazwa_procedury[(parametr1,parametr2,...)];
```

Usuwanie procedury:

```
DROP PROCEDURE nazwa_procedury;
```

2) Funkcje

Tworzenie funkcji:

```
CREATE [OR REPLACE] FUNCTION nazwa_funkcji
    [(argument [IN | OUT | IN OUT] typ_danych, ... , ... )]
RETURN typ_danych IS
    [sekcja_deklaracyjna]
BEGIN
    [instrukcje];
    [obsługa_wyjatków]
END;
```

gdzie:

- ❖ **OR REPLACE** - modyfikuje/zastępuje istniejącą funkcję,
- ❖ **IN | OUT | IN OUT** - określa sposób przekazywania parametrów do odczytu, do zapisu, do odczytu i do zapisu (domyślnie).

Polecenie kompilacji funkcji: „/”

Wyświetlanie błędów funkcji: **SHOW ERRORS [FUNCTION nazwa_funkcji]**

Wywoływanie funkcji:

```
nazwa_funkcji[(parametr1,parametr2,...)];
SELECT nazwa_funkcji[(parametr1,parametr2,...)] FROM DUAL;
```

Usuwanie funkcji:

```
DROP FUNCTION nazwa_funkcji;
```

3) Typy danych, instrukcje, wyświetlanie danych, obsługa wyjątków.

Typy danych:

Wszystkie typy poprawne dla SQL i Oracle oraz dodatkowo:

boolean – typ logiczny o wartościach true, false i null,

binary_integer – typ całkowity,

tabela.kolumna%type – typ danych taki, jak typ podanej kolumny,

tabela.%rowtype – zmienna złożona odpowiadająca całemu wierszowi.

Deklaracja stałej:

```
stała CONSTANT typ [:= wartość];
```

Deklaracja zmiennej:

```
zmienna typ [:= wartość];
```

Instrukcje:

instrukcja warunkowa:

```
IF warunek THEN
    polecenia;
{ELSIF warunek_i THEN
    polecenia;}
[ELSE
    polecenia;]
END IF;
```

instrukcja pętli:

```
WHILE warunek LOOP
    polecenia;
END LOOP;
```

Wyświetlanie danych:

z poziomu SQL*Plus:

```
SET SERVEROUTPUT ON
```

w podprogramie:

```
DBMS_OUTPUT.ENABLE; -- włączenie wyświetlania
```

```
DBMS_OUTPUT.PUT_LINE(tekst); -- wyświetlenie tekstu ze zmianą linii
```

```
DBMS_OUTPUT.PUT(tekst); -- wyświetlenie tekstu
```

gdzie tekst – napis, zmienna typu char lub varchar2;

przy wyświetlaniu liczb lub dat należy zamienić je na tekst za pomocą polecenia

```
TO_CHAR()
```

Obsługa wyjątków:**predefiniowane wyjątki systemowe:**

NO_DATA_FOUND – w poleceniu SELECT nie znaleziono odpowiadających wierszy

TOO_MANY_ROWS – polecenie SELECT zwróciło więcej niż jeden wiersz

ZERO_DIVIDE – dzielenie przez 0

tworzenie wyjątków użytkownika:

w sekcji deklaracji: *nazwa* EXCEPTION;

w miejscu wystąpienia: RAISE *nazwa*;

obsługa wyjątków – w bloku obsługi wyjątków:

WHEN *nazwa* THEN *polecenia*;

{WHEN *nazwa* THEN *polecenia*;}

[WHEN OTHERS THEN *polecenia*;]