

Βάσεις Δεδομένων: PL/SQL (Procedural Language for SQL)

Μάιος 22, 2020

Χαρακτηριστικά της PL/SQL

- Παρέχει δυνατότητα **ελέγχου σφαλμάτων**.
- Προσφέρει διάφορους **τύπους δεδομένων**.
- Προσφέρει διάφορους **δομές δεδομένων**.
- Δυνατότητα χρήσης **συναρτήσεις (functions)** και **διαδικασίες (procedures)**.
- Υποστηρίζει των αντικειμενοστραφή προγραμματισμό.

PL/SQL και τμήματα κώδικα (Blocks)

- Κάθε PL/SQL κώδικας οργανώνεται σε **blocks**.
- Τα τμήματα κώδικα (blocks) μπορεί να είναι:
 - Ανώνυμα τμήματα (**anonymous block**)
 - Συναρτήσεις (**function**)
 - Διαδικασίες (**procedure**)
 - Σκανδάλες (**trigger**)

Η δομή μιας εντολής PL/SQL

DECLARE

/*Δηλώσεις (προαιρετικά):

Μεταβλητές, τύποι και τοπικά υποπρογράμματα.*/

BEGIN

/* Εκτελέσιμο (υποχρεωτικό):

διαδικασίες και οι εντολές PL/SQL. */

EXCEPTION

/* Χειρισμός εξαιρέσεων (προαιρετικό):

εντολές χειρισμού λαθών*/

END;

Τμήμα δηλώσεων

DECLARE

name [CONSTANT] datatype [(constraint)] [:= value];

π.χ.:

v_empno Number(4) ;

pi constant Number(6,4) := 3.1421;

myvariable INTEGER DEFAULT 1;

xyz VARCHAR2(200) := 'Hello World';

does_empl_exist Boolean ; --true, false, null

next_checkup Date Not Null := '2-Dec-92';

yourSalary employee.salary%TYPE;

Τμήμα εντολών

BEGIN

**/* Εκτελέσιμο (υποχρεωτικό):
διαδικασίες και εντολές PL/SQL. */**

DBMS_OUTPUT.ENABLE;

DBMS_OUTPUT.PUT_LINE(myvariable);

DBMS_OUTPUT.PUT_LINE(xyz);

myvariable := 15;

DBMS_OUTPUT.PUT_LINE(myvariable);

END;

/

Εκτέλεση

- Για να εκτελέσουμε ένα πρόγραμμα PL/SQL:
 - Γράφουμε τις εντολές στην γραμμή εντολών της sqlplus
 - Εισάγουμε στο τέλος μια γραμμή που περιέχει το χαρακτήρα /

Μεταβλητές και τύποι

- Τύποι μεταβλητών:
 - Ακέραιοι: **PLS_INTEGER**, **NUMBER**(length)
 - Αλφαριθμητικά: **CHAR**(length), **VARCHAR2**(max length)
 - Πραγματικοί: **NUMBER**(precision, scale), **REAL**, **FLOAT**(binary precision)
 - Ίδιος με τον τύπο μιας στήλης ενός πίνακα.
 - Ίδιος με τον τύπο μιας **εγγραφής** με πολλά πεδία (**tuple**).
- Οι μεταβλητές δεν πρέπει να έχουν ίδιο όνομα με στήλες πινάκων.
- Η αρχική τιμή μεταβλητών είναι **NULL**.
- Στις μεταβλητές ανατίθεται τιμή με τον τελεστή “:=”.

'Hello World'

```
DECLARE
```

```
    message  varchar2(20) := 'Hello, World!';
```

```
BEGIN
```

```
    dbms_output.put_line(message);
```

```
END;
```

```
/
```

Σχόλια

```
DECLARE
    -- variable declaration
    message varchar2(20) := 'Hello, World!';
BEGIN
    /*
     * PL/SQL executable statement(s)
     */
    dbms_output.put_line(message);
END;
/
```

Χρήση προσδιοριστικού %TYPE

```
DECLARE
    -- Specific column from table.
    l_username all_users.username%TYPE;
BEGIN
    -- Specific column from table.
    SELECT username
    INTO    l_username
    FROM    all_users
    WHERE   username = 'SYS';
    DBMS_OUTPUT.put_line('l_username=' || l_username);
END;
```

Δομές ελέγχου

create or replace function **bmi** (height int , weight float) return float is

BEGIN

if height <= 0.3 or height > 3.0 then

dbms_output.put_line ('height must be in [0.3, 3.0] meters') ;

end if;

if weight <= 0 then

dbms_output.put_line('weight must be positive');

elsif weight > 800 then

dbms_output.put_line('Check the specified value is correct');

end if;

return weight/(height*height); --Μπορώ να χρησιμοποιήσω και pl/sql functions π.χ. power(height,2)

END;

Δομές επανάληψης (loops)

```
DECLARE
```

```
  i NUMBER := 1;
```

```
BEGIN
```

```
  LOOP
```

```
    INSERT INTO T1 VALUES(i,i);
```

```
    i := i+1;
```

```
    EXIT WHEN i>100;
```

```
  END LOOP;
```

```
END;
```

Παράδειγμα δημιουργίας SQL Function

- 1) Εμφανίστε τους μισθούς υπαλλήλων μαζί με ένα bonus, το οποίο να είναι 1000 για όσους απασχολούνται σε περισσότερα από 3 έργα και 100 για όλους τους υπόλοιπους.
- 2) Χρειάζομαι ένα Function που θα υπολογίζει το bonus των υπαλλήλων.
- 3) Άρα θα φτιάξουμε ένα function που θα παίρνει το ssn ενός υπαλλήλου, θα μετράει το πλήθος των έργων του και θα επιστρέφει το bonus του.
- 4) “SELECT fname, lname, salary+**bonus**(ssn) from employee;”
- 5) Q: Πώς θα βρίσκαμε τον αριθμό των τέκνων ενός υπαλλήλου;

Παράδειγμα: “Επιστροφή bonus υπαλλήλου”

```
create or replace function bonus(ssn int) return float is
    numprojects int;
    bonus float;
BEGIN
    bonus:=100;
    Select count(*) into numprojects from works_on where essn=ssn;
    if numprojects >= 3 then
        bonus:=1000;
    end if;
    return bonus;
END;
```