
Προηγμένα Θέματα Βάσεων Δεδομένων

Διδάσκων: Άγγελος Μιχάλας

Περιεχόμενα

- **Stored Functions**

Εισαγωγή

- *Μία stored function είναι ένα ειδικό αποθηκευμένο πρόγραμμα το οποίο επιστρέφει μια τιμή.*
- Χρησιμοποιούνται για να ενθυλακώσουν κανόνες ή λογική που είναι κοινή μεταξύ δηλώσεων **SQL statements** ή **αποθηκευμένων προγραμμάτων**.

Σύνταξη

```
CREATE FUNCTION function_name (param1, param2, ...)
  RETURNS data type
  [NOT] DETERMINISTIC
  statements
```

- Όλες οι παράμετροι είναι IN.
- Καθορισμός του τύπου δεδομένων επιστροφής στη δήλωση RETURNS.
 - Π.χ. Μπορεί να είναι κάθε έγκυρος τύπος δεδομένων.

Σύνταξη

- Εάν μία stored function επιστρέφει το ίδιο αποτέλεσμα για την ίδια είσοδο, τότε θεωρείται deterministic, διαφορετικά θεωρείται μη deterministic.
- Καθορισμός τουλάχιστον μίας δήλωσης RETURN.

Functions ή Procedures

1. Μία procedure δεν επιστρέφει τιμή και καλείται με μια εντολή CALL.
 - Οι παράμετροι της procedure μπορούν να οριστούν ως input-only, output-only, ή και τα δύο.
 - Όλες οι παράμετροι μιας function θεωρούνται **IN παράμετροι**.
2. Μία function καλείται σε μια έκφραση & επιστρέφει μια τιμή.
 - Πρέπει να υπάρχει ένας «RETURNS» όρος στον ορισμό της function για να υποδεικνύει τον τύπο της επιστρεφόμενης τιμής.
 - Πρέπει να υπάρχει τουλάχιστον μία δήλωση RETURN
3. Δεν μπορεί να γίνει κλήση μιας function με την εντολή CALL και δεν μπορεί να καλεστεί μία procedure σε ένα expression.

Functions ή Procedures

4. Οι stored procedures και οι functions δε μοιράζονται τον ίδιο χώρο ονομάτων.
5. Μία Function μπορεί να χρησιμοποιηθεί σε SQL Queries ενώ η procedure δεν μπορεί.
 - Μία function μπορεί να καλεστεί σε μια δήλωση SQL όπως `select func_name from table_name`.
6. Οι αποθηκευμένες ρουτίνες (stored procedures και stored functions) σχετίζονται με ένα συγκεκριμένο σχήμα όπως οι πίνακες και οι όψεις.
 - Όταν διαγράφεται ένα σχήμα ΒΔ, όλες οι αποθηκευμένες ρουτίνες διαγράφονται.

Παράδειγμα

```
CREATE FUNCTION CustomerLevel (creditLimit double) RETURNS
  VARCHAR(10) DETERMINISTIC
BEGIN
  DECLARE lvl varchar(10);
  IF creditLimit > 50000 THEN
    SET lvl = 'PLATINUM';
  ELSEIF (creditLimit <= 50000 AND creditLimit >= 10000) THEN
    SET lvl = 'GOLD';
  ELSEIF creditLimit < 10000 THEN
    SET lvl = 'SILVER';
  END IF;
  RETURN (lvl);
END
```


Παράδειγμα

use customerproduct;

SELECT customerName, CustomerLevel(creditLimit)

FROM customers

ORDER BY customerName;

Παράδειγμα

```
CREATE PROCEDURE GetCustomerLevel2(IN p_customerNumber
    INT(11), OUT p_customerLevel varchar(10))
BEGIN
    DECLARE creditlim DOUBLE;
    SELECT creditlimit INTO creditlim
    FROM customers
    WHERE customerNumber = p_customerNumber;

    SELECT CUSTOMERLEVEL(creditlim) INTO p_customerLevel;
END

use customerproduct;
CALL GetCustomerLevel2 (103,@customerLevel);
SELECT @customerLevel;
```