

MySQL stored procedures

Δημήτρης Σουραβλιάς

Δομή παρουσίασης

- Εισαγωγή
- Δημιουργία μιας απλής stored procedure
- Μεταβλητές
- Παράμετροι
- Ροή ελέγχου
- Cursors

Εισαγωγή

Μια stored procedure:

- είναι ένα τμήμα προγράμματος που αποθηκεύεται στον κατάλογο της Βάσης Δεδομένων
- μπορεί να κληθεί από ένα πρόγραμμα ή μια άλλη stored procedure
- μοιάζει με διαδικασία μιας τυπικής γλώσσας προγραμματισμού

3

Εισαγωγή (2)

Πλεονεκτήματα:

- υπολογιστικό κέρδος, μια stored procedure εκτελείται πιο γρήγορα από ένα σύνολο MySQL εντολών
- μείωση επικοινωνιακού φορτίου μεταξύ εξυπηρετή Βάσεων Δεδομένων και εφαρμογής
- μια stored procedure είναι επαναχρησιμοποιήσιμη
- μια stored procedure είναι ασφαλής

4

Εισαγωγή (3)

Μειονεκτήματα

- αυξάνουν το φόρτο του εξυπηρέτη Βάσεων Δεδομένων
- δεν μπορούν να χρησιμοποιηθούν για πολύπλοκα προγράμματα
- το debugging είναι δύσκολο
- η δημιουργία και η διατήρηση stored procedures είναι μια εξειδικευμένη ικανότητα

5

Στα παραδείγματα που ακολουθούν θεωρούμε ότι έχουμε κατασκευάσει στη Βάση έναν πίνακα MOVIES (mid, title, year)

- mid, το id της ταινίας (πρωτεύον κλειδί)
- title, ο τίτλος της ταινίας
- year, το έτος δημιουργίας της ταινίας

6

Δημιουργία μιας απλής stored procedure

```
DELIMITER //  
  
CREATE PROCEDURE GetAllMovies()  
  
    BEGIN  
  
        SELECT * FROM MOVIES;  
  
    END //  
  
DELIMITER ;
```

7

Δημιουργία μιας απλής stored procedure (2)

Ο default delimiter της MySQL είναι το ελληνικό ερωτηματικό (;)

Πριν δημιουργήσουμε μια νέα stored procedure τον αλλάζουμε σε κάποιο άλλο σύμβολο

Π.χ. //, \$\$, | κ. λπ. με την εντολή DELIMITER

Μόλις ολοκληρωθεί η δημιουργία μιας stored procedure, επαναφέρουμε τον default delimiter με χρήση της εντολής DELIMITER ;

8

Δημιουργία μιας απλής stored procedure (3)

Κλήση μιας stored procedure:

```
CALL stored_procedure_name();
```

Π.χ. **CALL** GetAllMovies();

Διαγραφή μιας stored procedure:

```
DROP PROCEDURE stored_procedure_name;
```

Π.χ. **DROP PROCEDURE** GetAllMovies;

9

Δομή παρουσίασης

- Εισαγωγή
- Δημιουργία μιας απλής stored procedure
- Μεταβλητές
- Παράμετροι
- Ροή ελέγχου
- Cursors

10

Μεταβλητές

- **Τοπικές μεταβλητές** που μετά την εκτέλεση της stored procedure αποδεσμεύουν τη μνήμη που καταλαμβάνουν

Δήλωση

```
DECLARE variable_name datatype(size) DEFAULT  
default_value;
```

Π.χ. **DECLARE** total_movies **INT DEFAULT** 0;

Ανάθεση τιμής

```
SET variable_name = value;
```

Π.χ. **SET** total_movies = 100;

11

Μεταβλητές (2)

- **Μεταβλητές χρήστη**, που είναι το ανάλογο των global μεταβλητών σε μια τυπική γλώσσα προγραμματισμού

Δήλωση και αρχικοποίηση μεταβλητής χρήστη στη γραμμή εντολών:

```
SET @variable_name = variable_value;
```

Π.χ. **SET** @a = 'ABC';

Για να δούμε στη γραμμή εντολών την τιμή του a:

```
SELECT @a;
```

12

Παράμετροι

Οι παράμετροι σε μια stored procedure μπορεί να είναι τύπου:

IN: η τιμή της παραμέτρου περνάει στη stored procedure, αλλά η τιμή που πέρασε δε μεταβάλλεται

OUT: η τιμή της παραμέτρου επιστέφεται στο πρόγραμμα που την κάλεσε

INOUT: η τιμή της παραμέτρου περνάει στη stored procedure και η τροποποιημένη τιμή της επιστρέφεται στο πρόγραμμα που την κάλεσε

13

Παράμετροι (2)

```
DELIMITER //
```

```
CREATE PROCEDURE GetMoviesByYear (IN inputYear INT)
```

```
BEGIN
```

```
    SELECT mid, title
```

```
    FROM MOVIES
```

```
    WHERE year = inputYear;
```

```
END //
```

```
DELIMITER ;
```

14

Δομή παρουσίασης

- Εισαγωγή
- Δημιουργία μιας απλής stored procedure
- Μεταβλητές
- Παράμετροι
- Ροή ελέγχου
- Cursors

15

Ροή ελέγχου

Η εντολή IF:

```
IF <condition_1> THEN ...  
ELSEIF <condition_2> THEN  
... ..  
ELSEIF <condition_3> THEN ...  
ELSE ...  
END IF;
```

Η εντολή WHILE:

```
WHILE <condition> DO  
    <loop_body>  
END WHILE;
```

16

Ροή ελέγχου(2)

```
DELIMITER $$

CREATE PROCEDURE WhileProc (IN myYear INT, IN offset INT)

BEGIN

    DECLARE lastYear INT;
    SET lastYear = myYear + offset;

    WHILE myYear <= lastYear DO

        SELECT * FROM MOVIES WHERE year = myYear;
        SET myYear = myYear + 1;

    END WHILE;

END $$
DELIMITER ;
```

17

Ροή ελέγχου(3)

Άλλες εντολές για επανάληψη:

- **LOOP, LEAVE** και **ITERATE**
- **REPEAT ... UNTIL <condition> END REPEAT;**

18

Cursors

Χρησιμοποιούνται για να διατρέχουμε ν-άδες κάποιας σχέσης

Δήλωση

```
DECLARE cursor_name CURSOR FOR SELECT_Statement;
```

Άνοιγμα

```
OPEN cursor_name;
```

Διάβασμα ν-άδας

```
FETCH cursor_name INTO variable_list;
```

19

Cursors (2)

Η εντολή **FETCH** διαβάζει την τρέχουσα εγγραφή του πίνακα και αναθέτει τις τιμές των γνωρισμάτων στις μεταβλητές της `variable_list`. Στη συνέχεια μετακινεί τον `cursor` στην επόμενη γραμμή.

Κλείσιμο

```
CLOSE cursor_name;
```

+ **ΠΑΡΑΔΕΙΓΜΑ** από tutorial

20