Lecția 8 – Proceduri stocate

Proceduri
Funcții
Tipuri de parametri
Variabile

Proceduri stocate

- **Procedurile** şi **funcţiile** reprezintă un *set de instrucţiuni stocate pe server* în baza de date. **Procedurile** şi **funcţiile** mai poartă numele de **rutine.**

- O **procedură** poate să aibă *mai multe valori returnate*, poate să întoarcă o tabelă în timp ce o **funcţie** returnează *o singură valoare (un scalar).*

- O **funcţie** trebuie să conţină neapărat o instrucţiune **RETURN** prin care se precizează valoarea pe care acea **funcţie** o returnează (întoarce).

Proceduri stocate

- O **funcţie** poate fi folosită în expresii **SQL**, în timp ce o **procedură** nu poate fi folosită în expresii **SQL**.

- **Funcţiile** şi **procedurile** pot să primească şi *parametri*. O **funcţie** poate primi doar *parametri de intrare*, în timp ce o **procedură** poate să primească *parametri de intrare*, *parametri de ieşire* şi *parametri de intrare-ieşire*.

- O **procedură** se apelează prin intermediul instrucţiunii **CALL** în timp ce o **funcţie** se apelează prin numele ei.

Tipuri de parametri

Tipuri de parametri:

- **de intrare;**
- **de ieșire;**
- **de intrare/ieșire;**

Procedurile pot primi parametrii din toate cele trei tipuri existente (*intrare*, *ieşire*, *intrare/ieşire*), în timp ce **funcţiile** primesc doar parametri de *intrare*.

Funcţiile returnează *o singură valoare*, deci ele nu au parametri de ieşire.

Variabile. Tipuri de variabile

- O **variabilă** reprezintă un identificator pentru o zonă de memorie care îşi poate schimba valoarea.

- În cadrul procedurilor stocate, în **MySQL**, avem 2 tipuri de variabile:
  - **variabile de sesiune** care sunt valabile pe întreaga sesiune a unui utlizator;
  - **variabile locale** care se definesc în interiorul unui bloc de instrucţiuni şi sunt valabile doar acolo.

Variabile

Variabile de tip sesiune

- Variabilele de tip sesiune se declară folosind simbolul @ înaintea numelui variabilei, deci forma este **@nume_variabilă**.

- Instrucţiunea prin care unei variabile i se atribuie (setează) o valoare este următoarea:
SET @var1 := valoare;

Variabile

Variabile de tip sesiune

- Unei variabile i se poate atribui o valoare în cadrul unei instrucţiuni **SELECT**:
  SELECT @var_1 := val_1;

- **SELECT val_2 INTO @var_2; -** reprezintă introducerea unei valori extrase în cadrul instrucţiunii **SELECT** într-o variabilă, deci, o asociere a unei valori.

- **SELECT val_3, val_4 INTO @var_3, @var_4; -** reprezintă o asociere de valori pentru mai multe variabile printr-o singură instrucţiune **SELECT**.

Variabile

Variabile locale

- **DECLARE var_1 INT(3);**
  DECLARE var_2 VARCHAR(50);

- Cu o instrucţiune **DECLARE** se pot declara doar variabile de acelaşi tip. Deci trebuie să folosim câte o instrucţiune **DECLARE** pentru fiecare tip de variabile (întregi, şir de caractere, etc.).

- La declararea unei variabile se poate specifica şi o valoare implicită pentru variabila respectivă. Pentru acest lucru vom folosi cuvântul cheie **DEFAULT**. Dacă nu se specifică o valoare implicită, atunci valoarea implicită este **NULL**.

Proceduri

CREATE PROCEDURE *nume_procedură*
(parametru$_1$, parametru$_2$, ...)
BEGIN
...
END;

Implicit, **MySQL** consideră că instrucțiunea curentă se încheie atunci când se întâlnește caracterul (“;”).
Redefinirea temporară a operatorului de delimitare (“;”) într-un altul se face folosind instrucțiunea **DELIMITER**.

```
delimiter //
...
//
delimiter ;
```

Tipuri de parametri

Proceduri - Tipuri de parametri

- *Parametrii de intrare* sunt parametrii care intră în **procedură**; se definesc prin intermediul cuvântului cheie **IN** (**IN *nume_parametru* tip_parametru**).
- *Parametrii de ieşire* reprezintă valorile rezultate în urma efectuării setului de instrucţiuni din cadrul procedurii; se definesc prin intermediul cuvântului cheie **OUT** (**OUT *nume_parametru* tip_parametru**).
- *Parametrii de intrare/ieşire* reprezintă valori primite de procedură, prelucrate şi apoi returnate; aceşti parametri se definesc prin intermediul cuvântului cheie **INOUT** (**INOUT *nume_parametru* tip_parametru**).

Tipuri de parametri

Proceduri - Tipuri de parametri

- În interiorul **procedurilor** se pot face doar operaţii din **L**imbajul de **M**anipulare al **D**atelor (**LMD**).

- Parametrii **nu** sunt obligatorii într-o **procedură**. Deci putem avea şi **proceduri** care să nu primească *nici un parametru*.

- Ordinea în care apar paramterii, atunci când sunt mai mulţi, nu contează. Important este faptul că pentru fiecare parametru trebuie specificat tipul lui.

Blocuri

Blocuri de instrucțiuni

- Tot ceea ce se află între instrucţiunile **BEGIN** și **END** reprezintă **corpul procedurii**.
- În corpul principal al procedurii, delimitat prin **BEGIN** şi **END**, putem avea alte blocuri delimitate prin **BEGIN** şi **END**, deci blocuri imbricate.
- Blocurile cuprinse între instrucţiunile **BEGIN** şi **END** pot fi denumite folosind etichete (**LABELS**).

Blocuri

Blocuri de instrucțiuni

```
bloc1: BEGIN
...
        bloc2: BEGIN
                ...
END bloc2;
...
END bloc1;
```

Folosirea etichetelor pentru blocurile de instrucţiuni este utilă atunci când avem blocuri imbricate în interiorul unei proceduri.

Proceduri

- Variabile locale se declară după instrucţiunea **BEGIN** cu ajutorul instrucţiunii **DECLARE**.
- O procedură poate fi ştearsă prin instrucţiunea: **DROP PROCEDURE *nume_procedură*;**
- O procedură poate fi apelată prin instrucţiunea: **CALL nume_procedură();**

Funcții

O **funcţie MySQL** se creează folosind următoarea sintaxă:

CREATE FUNCTION *nume_funcţie* (param$_1$, param$_2$, ...) RETURNS tip_valoare_returnată
BEGIN
...
RETURN ...;
END;

O **funcţie** primeşte doar parametri de intrare şi returnează o valoare al cărei tip este specificat în clauza **RETURNS**. Valoarea returnată trebuie declarată în interiorul funcţiei ca o variabilă locală.

Funcții

- În momentul în care se execută instrucţiunea **RETURN** se iese din **funcţie.**
- La parametrii unei **funcţii** nu se mai specifică tipul **IN** pentru că toţi parametrii unei funcţii sunt de *intrare*.
- O **funcţie** se apelează prin numele ei.