

Seminar 5

Functii definite de utilizator. Trigger

Funcții definite de utilizator

- ▶ Sunt funcții definite de utilizatori care pot fi folosite în interogări SQL
- ▶ În Microsoft SQL Server sunt trei tipuri de funcții definite de către utilizatori:
 - Funcții scalare
 - Funcții Inline Table-Valued
 - Funcții Multi-Statement Table-Valued



Funcții definite de utilizator – Funcții scalare

- ▶ Funcțiile scalare returnează o singură valoare
- ▶ Sintaxa pentru crearea unei funcții scalare:

```
CREATE FUNCTION scalar_function_name (  
    @param1 datatype1, @param2 datatype2)  
RETURNS datatype AS  
BEGIN  
    -- Instrucțiuni SQL  
RETURN value  
END
```



Funcții definite de utilizator – Funcții scalare

► Sintaxa pentru modificarea unei funcții scalare:

```
ALTER FUNCTION scalar_function_name (@param1
datatype1, @param2 datatype2)
RETURNS datatype AS
BEGIN
-- Instrucțiuni SQL
RETURN value
END
```

► Sintaxa pentru ștergerea unei funcții scalare:

```
DROP FUNCTION scalar_function_name
```



Funcții definite de utilizator - Funcții scalare

Exemplu

- ▶ Funcție care returnează numărul de publicații ale unui autor principal

```
CREATE FUNCTION getNrPublicatii (@Autor varchar(50))
RETURNS INT AS
BEGIN
    DECLARE @Nrpub INT
    SET @Nrpub = 0
    SELECT @Nrpub = COUNT(*) FROM Publicatie
    WHERE autorp = @Autor
    RETURN @Nrpub
END
```

- ▶ Apel

```
PRINT dbo.getNrPublicatii('Michael Brodie')
```



Funcții definite de utilizator – Funcții inline table-valued

- ▶ Funcțiile definite de utilizator de tip inline table-valued returnează un tabel în locul unei singure valori
- ▶ Pot fi folosite oriunde poate fi folosit un tabel, de obicei în clauza FROM a unei interogări
- ▶ O funcție definită de utilizator de tip inline table-valued conține **un singur SQL statement**



Funcții definite de utilizator – Funcții inline table-valued

Exemplu

- ▶ **Funcție care returnează toate publicațiile unui autor principal**

```
CREATE FUNCTION getPublicatii (@Autor  
    varchar(50))  
    RETURNS TABLE  
    AS  
    RETURN  
    SELECT titlu, autorp FROM Publicatie  
    WHERE autorp = @Autor
```

- ▶ **Apel**

```
SELECT * FROM dbo.getPublicatii('Michael  
Brodie')
```



Funcții definite de utilizator – Funcții multi-statement table-valued

- ▶ O funcție definită de utilizator de tipul multi-statement table-valued returnează un tabel și conține **mai multe instrucțiuni SQL**, spre deosebire de o funcție inline table-valued care conține o singură instrucțiune SQL



Funcții definite de utilizator – Funcții multi-statement table-valued

Exemplu

- ▶ Funcție care returnează numele autorilor care au publicatii de categoria cu id-ul transmis ca si parametru

```
CREATE FUNCTION getAuthorsByCat ( @idcat int )
    RETURNS @AuthorsByCat table (autorp VARCHAR(50))
    AS
    BEGIN
        INSERT INTO @AuthorsByCat  SELECT autorp FROM Publicatie
        WHERE id_c = @idcat
        IF @@ROWCOUNT = 0
        BEGIN
            INSERT INTO @AuthorsByCat  VALUES ('Nu a fost gasit nici
            un autor')
        END
        RETURN
    END
GO
```

- ▶ Apel

```
SELECT * FROM dbo. getAuthorsByCat(111)
```



Functii definite de utilizator - Exercitii

Fie baza de date:

- ▶ Categorie (id_cat, nume, descriere)
 - ▶ Publicatie (id_publ, titlu, abstract, autorp, id_cat)
 - ▶ Biblioteca (id_biblio, nume, site)
 - ▶ Indexare (id_publ, id_biblio)
-
- ▶ Ex.1: scrieti o functie definita de utilizator care returneaza (a) toate publicatiile din baza de date si (b) o alta functie care returneaza autorul cu cele mai multe publicatii.
 - ▶ Ex.2: Scrieti o functie definita de utilizator care calculeaza numarul publicatiilor de categoria jurnal a unui autor principal primit ca si parametru.
 - ▶ Ex.3: Scrieti o functie care returneaza numarul total de publicatii de categorie 'jurnal' pentru fiecare autor principal.
 - ▶ Ex 4: Scrieti o functie care primeste ca parametru un intreg reprezentand id-ul unei biblioteci electronice si care returneaza valoarea 1 daca exista in baza de date o biblioteca cu id-ul respectiv, iar in caz contrar, returneaza valoarea 0.



Trigger

- ▶ Tip special de procedura stocata care se execută automat când o instrucțiune DML sau DDL este executată
- ▶ Nu se pot executa în mod direct
- ▶ Instrucțiuni DML: INSERT, UPDATE, DELETE
- ▶ Instrucțiuni DDL: CREATE DATABASE, ALTER TABLE, DROP LOGIN, UPDATE STATISTICS, DROP TRIGGER
- ▶ Fiecare trigger aparține unui singur tabel



Trigger

► Sintaxa:

```
CREATE TRIGGER trigger_name
ON { table | view }
[ WITH <dml_trigger_option> [ ,...n ] ]
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
[ WITH APPEND ]
[ NOT FOR REPLICATION ]
AS { sql_statement [ ; ] [ ,...n ] | EXTERNAL NAME
<method specifier [ ; ] > }
```



Trigger

- ▶ Momentul execuției:
 - ▶ FOR
 - ▶ AFTER (se pot defini mai multe trigger-e de acest tip)
 - ▶ INSTEAD OF
- ▶ Dacă se definesc mai multe trigger-e pentru aceeași acțiune, ele se execută în ordine aleatorie
- ▶ Când se execută un trigger, sunt disponibile două tabele speciale, numite *inserted* și *deleted*



Trigger - Exemplu

- ▶ Trigger care afiseaza titlul si autorul principal la inserarea unei publicatii

```
CREATE TRIGGER [dbo].[On_Publication_Insert]
ON [dbo].[Publicatie]
FOR INSERT
AS
BEGIN
SET NOCOUNT ON;
INSERT INTO LogInsPub (titlu, autorp)
SELECT titlu, autor_p FROM inserted
END
```



Trigger - Exemplu

- ▶ Trigger care afiseaza titlul si autorul principal la stergerea unei publicatii

```
CREATE TRIGGER [dbo].[On_Publication_Delete]
ON [dbo].[Publicatie]
FOR DELETE
AS
BEGIN
SET NOCOUNT ON;
INSERT INTO LogDelPub (titlu, autorp)
SELECT titlu, autor_p FROM deleted
END
```



Trigger - Exercitii

- ▶ Ex.1: scrieti un trigger pentru tabela Publicatii pentru operatia de adaugare; afisati pe ecran numele tabelii si un mesaj care sa indice faptul ca s-a adaugat o inregistrare
- ▶ Ex.2: scrieti un trigger pentru tabela Biblioteca pentru operatia de stergere; triggerul va afisa un mesaj care sa indice faptul ca s-a sters o inregistrare si numele bibliotecii din inregistrarea stearsa

