

## Proceduri

### Parametri de intrare

```
CREATE PROCEDURE ReturneazaAtractiiCuVarstaMin @varsta_min INT
AS
BEGIN
    SELECT nume, descriere, varsta_min FROM Atractii
    WHERE varsta_min = @varsta_min;
END;
```

Se apelează cu:

```
EXEC ReturneazaAtractiiCuVarstaMin 12;
-- sau
EXEC ReturneazaAtractiiCuVarstaMin @varsta_min=12;
```

### Modificare definiție

Parametrii de ieșire se specifică cu cuvântul-cheie OUTPUT.

```
ALTER PROCEDURE ReturneazaAtractiiCuVarstaMin
    @varsta_min INT,
    @nr_atractii INT OUTPUT
AS
BEGIN
    SELECT @nr_atractii=COUNT(*) FROM Atractii WHERE varsta_min=@varsta_min;
END;
```

Se apelează cu:

```
DECLARE @nr_atractii_out AS INT;
SET @nr_atractii_out = 0;
-- parametrul și variabila pot avea același nume
EXEC ReturneazaAtractiiCuVarstaMin 12, @nr_atractii=@nr_atractii_out OUTPUT;
PRINT @nr_atractii_out;
```

### Mesaje de eroare

Se prindează cu RAISERROR(msg, severity, state).

```
ALTER PROCEDURE ReturneazaAtractiiCuVarstaMin
    @varsta_min INT,
    @nr_atractii INT OUTPUT
AS
BEGIN
    SELECT @nr_atractii=COUNT(*) FROM Atractii WHERE varsta_min=@varsta_min;
    IF (@nr_atractii = 0)
        RAISERROR('Nu a fost returnata nici o atracție!', 16, 1);
END;
```

## Ștergere

```
DROP PROCEDURE ReturneazaAtractiiCuVarstaMin;
```

## Variabile globale

Pentru SQL Server variabilă globală înseamnă funcție sistem.

Ele încep cu @@.

Exemple:

- @@ERROR, numărul ultimii erori
- @@IDENTITY, valoarea câmpului IDENTITY al ultimei înregistrări inserate
- @@ROWCOUNT, numărul de linii afectate de ultima operație
- @@SERVERNAME, numele instanței
- @@SPID, id-ul procesului utilizatorului curent
- @@VERSION, versiunea SQL Server-ului

Putem folosi SET NOCOUNT ON/SET NOCOUNT OFF pentru a opri/porni afișarea numărului de linii afectate.

## Execuție dinamică

```
CREATE PROCEDURE ReturneazaDateDinTabel
    @nume_tabel VARCHAR(100)
AS
BEGIN
    -- No bueno! SQL injection posibil.
    EXEC ('SELECT * FROM ' + @nume_tabel);
END;

DECLARE @sql NVARCHAR(100);
SET @sql = N'SELECT nume, descriere FROM Sectiuni WHERE nume<>@nume;';
EXEC sp_executesql @sql, N'@nume AS VARCHAR(100)', @nume='sectiunea 1';
```

## Control al fluxului

```
BEGIN
    -- instrucțiuni
END
```

RETURN iese dintr-o interogare/procedură.

RETURN [status\_code] se poate folosi pentru a returna un status code.

```
DECLARE @status INT;
EXEC @status = procedura_care_returneaza_status;
SELECT 'Status' = @status;
```

## IF și ELSE

```
IF [condiție]
    -- instrucțiune sau bloc de instrucțiuni
ELSE
    -- instrucțiune sau bloc de instrucțiuni
```

## WHILE

```
WHILE [condiție]
    -- instrucțiune sau bloc de instrucțiuni
```

## GOTO

```
Label:
    -- instrucțiuni
GOTO Label;
```

## WAITFOR

```
-- execuția continuă la 22:00
WAITFOR TIME '22:00';
-- execuția continuă după 3 ore
WAITFOR DELAY '03:00:00';
```

## THROW, TRY, CATCH THROW error\_number, message, state

```
THROW 50002, 'Înregistrarea nu există', 1;
```

Catch captează doar erori cu severitate mai mare decât 10, și acelea după care nu se închide conexiunea la baza de date.

```
BEGIN TRY
    -- instrucțiuni
END TRY
BEGIN CATCH
    ERROR_NUMBER();    -- numărul erorii
    ERROR_SEVERITY();  -- severitate erorii
    ERROR_STATE();     -- state-ul erorii

    ERROR_PROCEDURE(); -- procedura în care s-a produs eroarea
    ERROR_LINE();      -- linia la care s-a produs eroarea
    ERROR_MESSAGE();   -- mesajul erorii
END CATCH
```