

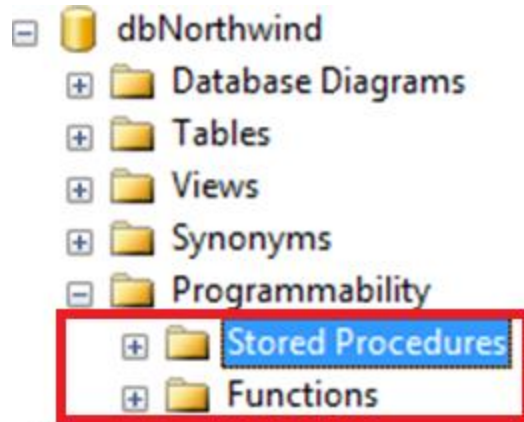


UNIVERSITETI AAB
FAKULTETI SHKENCA KOMPJUTERIKE
Inxhinieria Softuerike

Krijimi dhe përdorimi Funkcioneve dhe Procedurave

Hyrje

- DBMS posedojnë operacione të optimizuara që janë:
 - Procedurat dhe
 - Funkcionet



SQL statements

- Procedurat dhe funksionet janë elemente të database
- Procedurat dhe funksionet janë të ngjashme me querit
- Mund të thiren në ndonjë
 - Host gjuhë,
 - Si SQL komandë ose
 - Në ndërtimin e SQL komandave tjera
- Procedurat / funksionet mund të përmbajë sintaksa te SQL:
select, insert, update, delete

Sintaksa e përgjithshme

- Forma e përgjithshme e procedurës së memoruar
 - CREATE PROCEDURE <proc_name> (<parameters>)
 - <local deklaration>
 - <procedure body>;
- Forma e përgjithshme e funksionit
 - CREATE FUNCTION <func_name> (<parameters>) RETURNS <type>
 - <local deklaration>
 - <function body>;

Variablat

- Deklarimi variablave:

```
DECLARE @limit money
```

```
DECLARE @min_range int, @hi_range int
```

- Inicjalizon vleren e variables:

```
SET @min_range = 0, @hi_range = 100
```

```
SET @ID = ISNULL(@ID,0) + 1
```

- Inicjalizon vleren e variables ne SQL :

```
SELECT @ID = MAX(p.Id)
```

```
FROM dbo.tblPerdoruesi p
```

CASE ... WHEN

CASE input_expression

WHEN when_expression THEN result_expression

[WHEN when_expression THEN result_expression...n]

[ELSE else_result_expression]

END

Shembull:

SELECT CASE payterms

WHEN 'Net 30' THEN 'Payable 30 days after invoice'

WHEN 'Net 60' THEN 'Payable 60 days after invoice'

WHEN 'On invoice' THEN 'Payable upon receipt of invoice'

ELSE 'None'

END as Payment_Terms

FROM sales ORDER BY payterms

WHILE

WHILE Boolean_expression
SQL_statement | block_of_statements
[BREAK] SQL_statement | block_of_statements
[CONTINUE]

Example:

```
WHILE (SELECT AVG(royalty) FROM roysched) < 25
BEGIN
    UPDATE roysched
    SET royalty = royalty * 1.05
    IF (SELECT MAX(royalty)FROM roysched) > 27 BREAK
    ELSE CONTINUE
END
SELECT MAX(royalty) AS "MAX royalty"
FROM roysched
```

Procedura

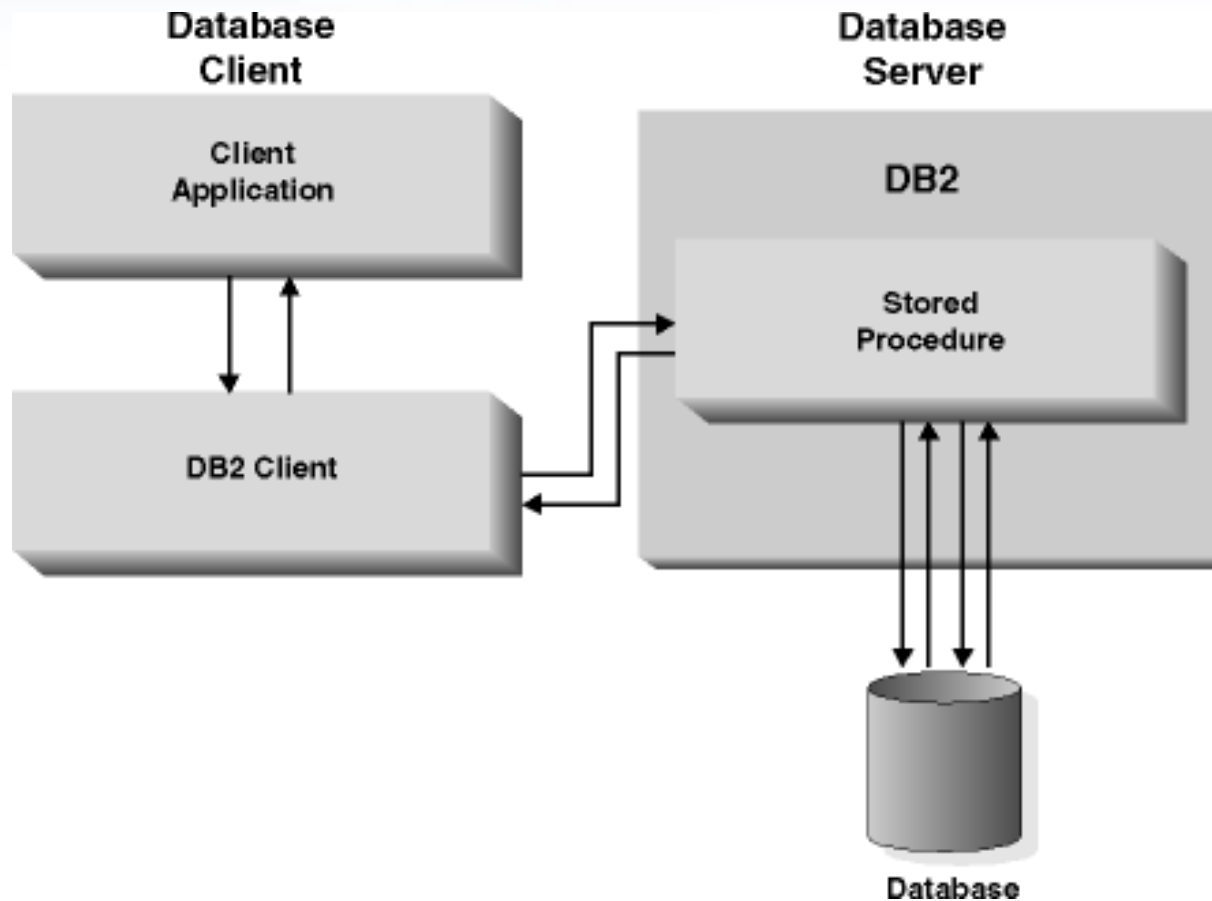
- Janë koleksione të kompletuara të SQL deklaratave që përpilohet në një plan të ekzekutimit të vetme.
- Procedura është e ruajtur në zonën cache memorie,
 - procedura ekzekutohet e para në mënyrë që ajo të mund të përdoret në mënyrë të përsëritur.
 - SQL Server nuk duhet të përpilohen atë çdo herë, ajo është e gatshme për ekzekutim.
- Ajo mund të pranojë parametrat hyrëse, kthehen vlerat output si parametra, ose kthen vlerat me sukses apo mesazhin e statusit të dështimit

Procedura

- Zvogëlojnë sasinë e kodit i cili nëpërmjet rrjetës dërgohet prej kompjuterit të shfrytëzuesit te serverit
- Kodin e pyetësorit e vendosim në server, e jo në kompjuterët e shfrytëuesve.
- DBMS komerciale lejojnë të definojnë procedura dhe funksione në shemën e database-it
- EXEC emri_i_procedurës_së_ruajtur

DBMS me Procedura

- Avantazhet e procedurës :
 - Redukton rrjetin ne trafik



Shembull

Te krijohet procedura e cila l paraqet te dhenat nga tabela e autori dhe lname eshte me radhitje nga DESC :

```
CREATE PROCEDURE DBO.Show_Authors AS  
SELECT au_fname, au_lname, address, city, state, zip  
FROM authors  
WHERE city = 'Oakland'  
ORDER BY au_lname DESC
```

Shembull (Me parametra)

Te krijohet procedura e cila paraqet shitjen prej datës në datë:

```
Create procedure [dbo].[Sales by Year]
```

```
    @Beginning_Date DateTime, @Ending_Date DateTime
```

```
AS
```

```
    SELECT Orders.ShippedDate,
```

```
    Orders.OrderID, [Order Subtotals].Subtotal,
```

```
    DATENAME(yy,ShippedDate) AS Year
```

```
FROM Orders INNER JOIN "Order Subtotals" ON Orders.OrderID =  
    "Order Subtotals".OrderID
```

```
WHERE Orders.ShippedDate Between @Beginning_Date And  
    @Ending_Date
```

Vazhdim...

Ekzekutimi procedures, metoda I:

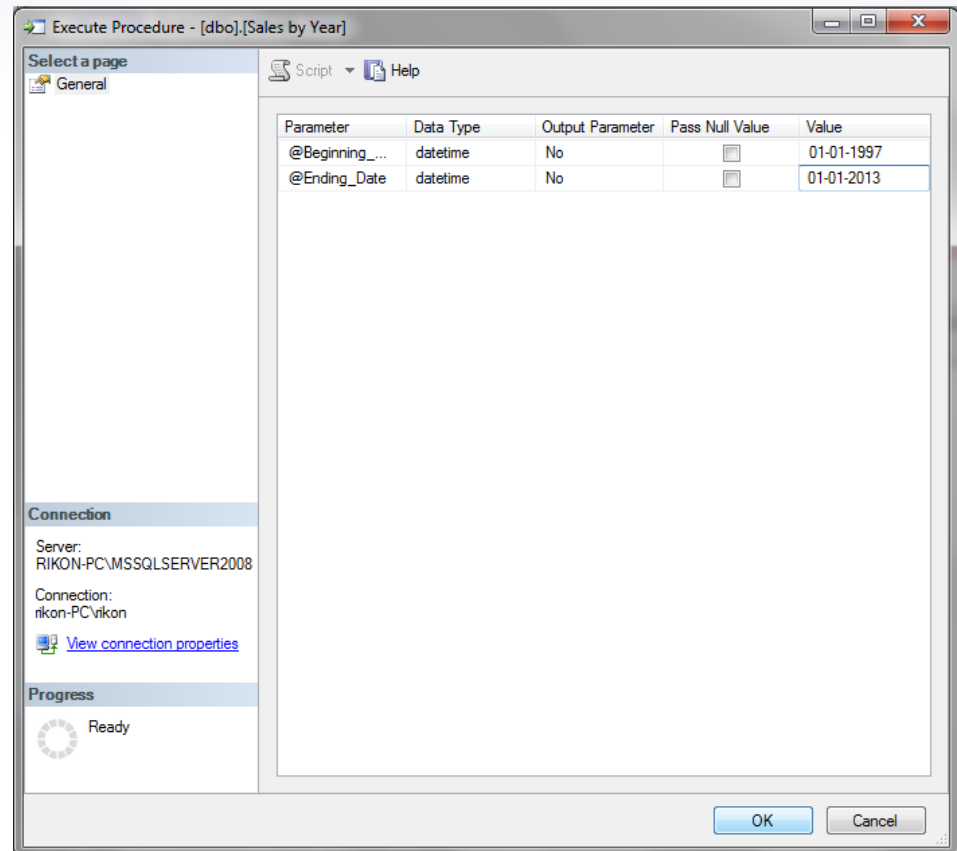
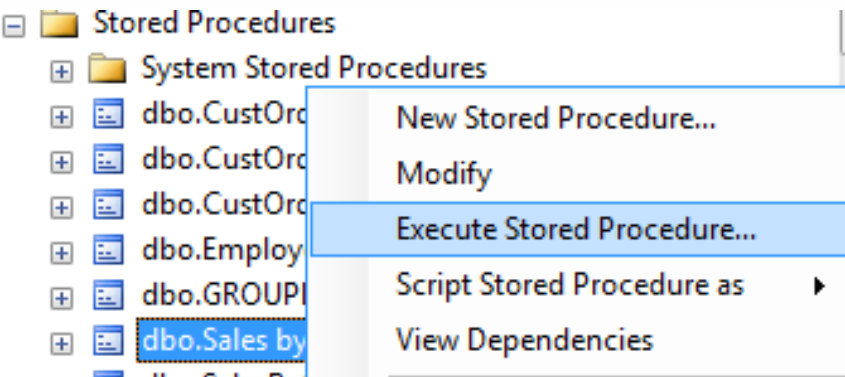
```
Execute [dbo].[Sales by Year] @Beginning_Date='01-01-1997', @Ending_Date= '01-01-2013'
```

Results Messages

	ShippedDate	OrderID	Subtotal	Year
393	1997-12-26 00:00:00.000	10775	228.00	1997
394	1997-12-26 00:00:00.000	10787	2622.76	1997
395	1997-12-26 00:00:00.000	10790	722.50	1997
396	1997-12-31 00:00:00.000	10789	3687.00	1997
397	1997-12-31 00:00:00.000	10792	399.85	1997
398	1997-12-31 00:00:00.000	10801	3026.85	1997
399	1998-01-01 00:00:00.000	10791	1829.76	1998
400	1998-01-02 00:00:00.000	10771	344.00	1998
...

Vazhdim...

Ekzekutimi procedures, metoda II:



Shembull (Pa parametera)

```
CREATE PROC Departments_Members
```

```
AS
```

```
    SELECT Dep_Name, COUNT(Emp_ID) NumberOfMember  
    FROM Departments D, Employees E  
    WHERE D.Dep_ID = E.Dep_ID  
    GROUP BY Dep_Name
```

Ekzekutimi procedures

```
Execute Departments_Members
```

Shembull (Perdorimi RETURN)

```
CREATE PROC GROUPLIDER_MEMBERS
    @Emp_Code varchar(10) = null
AS
    IF @Emp_Code is null
    BEGIN
        PRINT 'Please enter Employee Code!'
        RETURN
    END
    SELECT * FROM Employees
    WHERE EMP_EMP_ID = (SELECT EMP_ID FROM Employees
                        WHERE Emp_Code = @Emp_Code)
    ORDER BY Emp_Name
```


Shembull për insertim

Shembull të një procedure per insertim në SQL Server

```
CREATE PROCEDURE AddStudent ( @stud_id int,  
    @stud_name char(15),  
    @stud_surname char(15), @stud_gen char(1))  
AS  
BEGIN  
SET NOCOUNT ON;  
    insert into student(s_id, s_name, s_surname, s_gender)  
    values(@stud_id, @stud_name, @stud_surname, @stud_gen)  
END
```

Shembull për update

Shembull të një procedure per ndryshim në SQL Server

```
CREATE PROCEDURE UpdateStudent ( @stud_id int,  
    @stud_name char(15),  
    @stud_surname char(15), @stud_gen char(1))
```

```
AS
```

```
BEGIN
```

```
SET NOCOUNT ON;
```

```
UPDATE dbo. student
```

```
    SET  s_name = @stud_name,  
        s_surname= @stud_name,  
        s_gender= @stud_gen
```

```
WHERE stud_id = @stud_id
```

```
END
```

Shembull për fshirje

Shembull të një procedure per fshirje në SQL Server

```
CREATE PROCEDURE DeletStudent ( @stud_id int,  
    @stud_name char(15),  
    @stud_surname char(15), @stud_gen char(1))  
  
AS  
  
BEGIN  
  
SET NOCOUNT ON;  
  
  
DELETE FROM student  
WHERE stud_id = @stud_id  
  
END
```

Fshirja dhe modifikimi procedura-ve

Procedurat mund të fshihet ngjashëm sikurse objektet tjera:

```
drop PROCEDURE <pr_name>
```

p.sh. drop PROCEDURE UpdateStudent

Procedurat mund të modifikohet:

```
alter PROCEDURE < pr_name _name> <definimi i mëtutjeshëm>
```

p.sh. Alter PROC Departments_Members

AS

```
SELECT Dep_Name, COUNT(Emp_ID) NumberOfMember  
FROM Departments D, Employees E  
WHERE D.Dep_ID = E.Dep_ID  
GROUP BY Dep_Name
```

Kur ndryshohet procedura :

Exec sp_recompile[prName]-rikompajlimi I procedures ne menyrr qe te rritet performanca e saj

Funksionet

- Një funksion:
 - Është objekt në SQL që kthen një vlerë
 - Mund të ruhen në bazën e të dhënave, si një objekt i skemës për ekzekutim të përsëritura
 - Quhet si pjesë e një shprehje ose është përdorur për të siguruar një vlerë për parametër të caktuar
- Mund të manipulohet me vlerat e të dhënave

Vazhdim...

- Gjuha SQL, gjithashtu përmban një numër të konsiderueshëm të funksioneve.
- Ato janë të shfrytëzueshme sidomos kur kemi të bëjmë llogaritje të ndryshme me shenime.
- Mund të përdoren ku aktivitetet janë shumë komplekse, shumë të vështira në SQL
- Mund të rrisin efektshmërinë kur përdoren në klauzulen WHERE për të dhënat që filtrohen në kërkesë

Sintaksa per krijimin e funksioneve

Për krijimin e funksioneve të tilla, nevojitet urdhëri CREATE FUNCTION, i cili e ka këtë sintaksë:

```
CREATE FUNCTION [emri_i_db].emri_i_funksionit  
{  
  {@emri_i_parametrit tipi_i_shenimeve [=vlera_e_përcaktuar]}  
  [...n] ]  
}  
RETURNS tipin_e_shenimeve  
[AS]  
{BEGIN trupi_i_funksionit END}
```

Thirrja e funksioneve ne SQL

Funksionet:

- Ruhen në bazën e të dhënave
- Pranojë parametrat me të dhënat e llojeve të vlefshme ne SQL

Funksioni deklarohet ne SQL:

- Parametrat duhet të specifikohet me simbole dhe pozicione përkatëse
- Duhet vetëm emri funksionit dhe ekzekutimi privilegjeve

Vazhdim...

```
select * from dbo.fnLlojiPunesLista()
```

Results

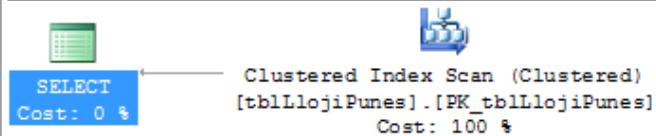
Messages

	IdLlojiPunes	Pershkrimi
1	1	Programere
2	2	I pa pune
3	3	Puntore
4	4	Menaxhere

Messages

Execution plan

Query 1: Query cost (relative to the batch): 100%
select * from dbo.fnLlojiPunesLista()



Clustered Index Scan (Clustered)

Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Estimated I/O Cost	0.003125
Estimated CPU Cost	0.0001614
Estimated Number of Executions	1
Estimated Operator Cost	0.0032864 (100%)
Estimated Subtree Cost	0.0032864
Estimated Number of Rows	4
Estimated Row Size	40 B
Ordered	False
Node ID	0

Object

[dbTest].[dbo].[tblLlojiPunes].[PK_tblLlojiPunes]

Output List

[dbTest].[dbo].[tblLlojiPunes].IdLlojiPunes, [dbTest].[dbo].[tblLlojiPunes].Pershkrimi

Llojet e funksioneve

Ekzistojnë tri lloje të ndryshme të funksioneve:

- Funksionet të cilat si rezultat japin nje tabele me shenime,
- Funksionet agregate të cilat si rezultat japin një vlerë (shembull, shumën e të gjitha vlerave në një kolonë të caktuar), dhe
- Funksionet skalare të cilat si vlera hyrëse mund të marrin një ose më shumë vlera, ndërsa si rezultat kthejnë në vlerë

Funkzione tabelare

```
CREATE FUNCTION AveragePricebyItems (@price money =  
0.0) RETURNS table
```

```
AS
```

```
RETURN ( SELECT Ite_Description, Ite_Price  
FROM Items  
WHERE Ite_Price > @price)
```

Use:

```
select * from AveragePricebyItems (15.00)
```

Funksione skalare

- Shembull, mund të krijojmë një funksion që shërben për dyfishimin e vlerës fillestare

```
CREATE FUNCTION Dyfishi  
  ( @hyrja int)  
  RETURNS int  
  AS  
  BEGIN  
  RETURN 2 * @hyrja  
  END
```

Use:

```
select Dyfishi(1)
```

Funkzione skalare

```
CREATE FUNCTION Revenue_Day (@Date datetime) Returns money
AS
BEGIN
    DECLARE @total money
    SELECT @total = sum( Quantity*UnitPrice )
    FROM  dbo.Orders as d inner join dbo.[Order Details] as od on
    d.OrderID=od.OrderID
    WHERE year(getdate()) = year(@Date)
           and month(OrderDate) = month(@Date) and day(OrderDate)= day(@Date)
    RETURN @total
END
```

Use:

```
select Revenue_Day (GETDATE())
```

Fshirja e funksioneve

Fshirja e një funksion të ruajtur:

Mund të fshihet një funksion i ruajtur duke përdorur sintaksë e mëposhtme:

```
DROP FUNCTION function_name
```

shembull:

```
DROP FUNCTION get_sal;
```

Të gjitha privilegjet që janë dhënë në një funksion largohen kur funksioni është fshire.

Sintaksa CREATE dhe zëvendësimi i sajë është e barabartë me një funksion të fshire dhe ri-krijimi tij.

Procedura vc Funkzioni

Procedura	Functionet (fn. tabelare)
Ekzekutohet si një komand ne SQL	Kërkohet si pjesë e një shprehje
Nuk përmbanë deklaratën RETURN në klauzulen header	Duhet të përmbajë një return në klauzolë e header –it
Mund të kthehen vlera (nëse ka) për parametrat output	Duhet të kthehen një vlerë të vetme?
Mund të përmbajë deklaratën RETURN dhe të mos kthehen vlerë	Duhet te përmbajnë të paktën një deklaratë RETURN

?