PROCEDURI STOCATE

O procedură stocată este o mulţime ordonată de instrucţiuni SQL stocată permanent pe server şi compilată la utilizare. Procedurile stocate reprezintă o modalitate de a crea rutine şi proceduri care să fie rulate pe server de către procesele serverului.
Aceste proceduri pot fi lansate în execuţie de o aplicaţie apelantă sau de declanşatoare sau de regulile de integritate. Deoarece SQL-server permite administrarea bazelor de date din sistem este logic ca serverul este locul cel mai bun pentru rularea procedurii stocate. Procedurile stocate pot să returneze valori şi să modifice, să execute operaţii de comparare cu valorile precomparate folosite de sistem. Procedurile stocate pot să primească şi să returneze valori care nu provin neaparat dintr-o tabelă ci sunt calculate prin execuţia procedurii. Procedurile stocate rulează pe servere care sunt calculatoare puternice.
Avantaje:

1. Procedurile stocate rulează pe servere care sunt calculatoare puternice;
2. Deoarece ele sunt stocate pe server nu mai necesită transfer de informaţii din baza de date, ele fiind pe acelaşi calculator;
3. Procedura beneficiază de un acces direct, rapid şi imediat la baza de date ce permite o prelucrare rapidă a informaţiilor;
4. Permite partajarea logicii aplicaţiei între aplicaţii diferite;
5. Simplifică integrările parametrizate facilitând rularea repetată aceleiaşi interogări cu seturi diferite;
6. Oferă facilităţi pentru dezvoltarea C/S prin modularizarea aplicaţei separând modulele pentru client şi server, ce permite scurtarea timpului de execuţie şi uşurarea gestiunii aplicaţei;
7. Componentele de pe server pot fi dezvoltate separat de componentele client fapt ce face posibilă refolosirea lor între aplicaţile client;
8. Ele pot furniza mecanisme de securitate: gestiunea vederilor şi a procedurilor stocate pot fi folosite ca instrumente de asigurare a securităţii . Pot fi create

proceduri stocate pentru operatiile de adăugare/ modificare/ ştergere/ citire controlând programat fiecare din aceste tipuri de acces la informaţii;

9. Impune pe partea de server reguli orientate spre date, permit impunerea de reguli şi relaţii logice care sporesc controlul informaţiilor în sistem;
10. Micşorează transferul de instrucţiuni T-SQL.

La utilizarea procedurilor stocate există 4 etape principale:

a) Etapa de creare în care se crează procedura cu ajutorul comenzii: **CREATE PROCEDURE**.
b) Executia de către utilizator-se execută cu ajutorul comenzii: EXECUTE.
c) Etapa de COMPILARE în care serverul compilează şi optimizează procedura.
d) Etapa de executare de catre server în care serverul rulează planul de execuţie compilat al procedurii în timpul unei comenzi EXEC.

Crearea unei proceduri stocate se face cu ajutorul comenzii:

CREATE PROCEDURE [posesor] nume_procedura:nr
@nume_parametru tip_data [=default][output]
@nume_parametru tip_data [=default][output]
…….
[FOR REPLICATION][WITH Recompile]
ENCRIPTION]
AS ……..
INSTRUCŢIUNI SQL
RETURN

*Nume_procedur*ă reprezinta numele noii proceduri stocate. Numele procedurii trebuie să fie unic în cadrul unei baze de date. Procedurile temporare globale sau locale sunt create prin adăugarea ca prefix la *nume_procedură* a unui caracter # (*#nume_procedur*ă) pentru proceduri temporare locale şi două caractere ## (*##nume_procedur*ă) pentru proceduri temporare globale. Numele complet al procedurii incluzând şi # sau ## nu poate depasi 128 caractere.

Observatie: după ce în baza de date sunt înscrise informaţii multe cu care aplicaţia lucrează trebuie încărcată din nou procedura stocată.

nr este un întreg optional folosit pentru a grupa procedurile cu acelaşi nume astfel ele pot şi eliminate împreună printr-o singură instrucţiune DROP PROCEDURE. De exemplu, într-o aplicaţie procedurile folosite se numesc orderproc;1, orderproc;2, etc.

Prin folosirea instructiunii

DROP PROCEDURE orderproc

se elimină întregul grup de proceduri.

Numele şi alte informaţii sunt înregistrate în tabelul ***sysobjects***, iar codul sursă este stocat în ***syscomments***.

Execuţia unei proceduri se face prin execuţia unei instrucţiuni:

EXECUTE nume_procedura

O procedură nu poate fi creată decat în baza de date curentă. Pentru că baza de date să fie curentă în care dorim să creem o procedură trebuie ca instrucţiunea de creare să fie precedată **de USE-nume_baz**ă.

Procedurile pot conţine orice comenzi mai puţin comanda de creare. La crearea unei proceduri stocate SQL se compilează şi se verifică operaţiile pe care le efectuează. Dacă sunt erori procedura este respinsă. Ea poate fi acceptată chiar dacă apelează o altă procedura stocată care nu a fost implementată, dar se va primi un mesaj de avertizare. Dacă nu se mai crează o procedură stocată apelată utilizatorul va primi un mesaj de eroare.

Exemplul 1:

```
CREATE PROCEDURE determină_salariaţi
    as
    SELECT * FROM salariaţi

EXEC determină_salariaţi.
```

Dacă apelul procedurii este primul dintr-o secvenţă de comenzi nu este nevoie să mai specificăm EXECUTE din partea de instrucţiuni. E ste suficient numai numele procedurii. Numele procedurii nu poate să conţină spaţii şi poate fi accesat prin

BAZA_DEDATE.<OBJECT>

Transmiterea de parametrii

Programatorul poate să transmită valori care pot să intervină în execuţia procedurii care respectă următoarele principii:

- o procedură poate avea o mulţime nevidă de parametrii;
- parametrii pot fi utilizaţi ca locaţii nominale;
- desemnarea parametrilor se realizează prin prefixarea numelor cu simbolul @;
- parametrii sunt locali pentru procedura în care sunt definiţi.

Parametrii sunt folosiţi pentru a transmite informaţii unei proceduri la apelul ei care se introduc după numele ei separaţi prin virgule şi urmaţi de tipul informaţiilor aşteptate(acelaşi cu tipurile de date ale SQLS).

Valoarea fiecărui parametru trebuie să fie înlocuită de către utilizator atunci când procedura este executată. O procedură stocată poate avea maxim 2100 parametrii

In exemplul 2 dăm o procedură de inserare.

```
CREATE PROCEDURE inserare
        @n char(15),
        @p char(15),
        @v  int ,
as
   INSERT INTO stud VALUES(@n,@p,@v)
Inserare('Albu','Ion',20)
```

Dacă numele proceduri este urmat de un punct şi virgulă şi un număr întreg acesta specifică versiunea de creare a procedurii, implicit se înţelege 1.

Comanda SET NO EXEC ON determină eventualele erori la prima execuţie a unei proceduri. Aceasta permite evitarea eventualelor erori. Stergerea unei proceduri se face prin

DROP PROCEDURE nume-procedură.

Exemple de proceduri.

Crearea procedurii de afişare a mediilor studenţilor unei grupe.

```
IF EXISTS (SELECT name
            FROM sysobjects
           WHERE name = 'medii' AND type = 'P')
 DROP PROCEDURE medii

CREATE PROCEDURE medii
  @Gr  INT,
AS
  SELECT  S.NrLeg, Nume, Prenume, Avg(C.Nota) Media
  FROM Student S, Catalog C
  WHERE S.NrLeg=C.NrLeg  AND Gupa=@Gr
  GROUP BY S.NrLeg, Nume, Prenume

EXEC  medii 221
```

Procedura calclează media unui student al cărui nume e dat ca parametru

```
IF EXISTS (SELECT name FROM sysobjects
      WHERE name =' medie_stud ' AND type = 'P')
  DROP PROCEDURE medie_stud

CREATE PROCEDURE medie_stud
   @nume varchar(40),
   @Medie float OUTPUT
AS
SELECT  S.NrLeg, Nume, Prenume, Avg(C.Nota) Medie
 FROM Student S, Catalog C
 WHERE S.NrLeg=C.NrLeg and Nume=@nume
 GROUP BY S.NrLeg, Nume, Prenume

EXEC  medie_stud ' Ionescu'
```

Procedura calclează numărul de examene la care s-a prezentat un student al cărui nume e dat ca parametru.

```
CREATE PROCEDURE sit_stud_
   @nume varchar(40)='L%'
 AS
  SELECT  S.NrLeg, Nume, Prenume, COUNT(C.Nota) Nr_examene
  FROM Student S, Catalog C
  WHERE S.NrLeg=C.NrLeg and Nume like @nume
  GROUP BY S.NrLeg, Nume, Prenume
  EXEC sit_stud_
```

EXEC sit_stud 'G%'

Execuţia de către utilizator

Prima dată când se execută o procedură nouă ea este citită din tabelul syscomments şi se rezolvă referirile ei la alte obiecte.In timpul acestui proces, procesorul de comenzi construieşte arborele de secvenţă sau interogare care este apoi transferat optimizatorului de interogare şi compilare.

Compilarea

După ce s-a reuşit crearea arborelui de interogare optimizatorul SQL Server compilează întreg pachetul şi îl optimizează şi verifică drepturile de acces. În timpul fazei de optimizare,optimizatorul transformă arborele de interogare într-o structură optimală ţinând cont de planul de accesare a datelor vizate de procedură.

În timpul acestei etape se ţine cont de:

- utilizarea unirilor pentru legarea tabelelor;
- cantitatea de date;
- condiţiile de grupare;
- gradul de distribuire al datelor;
- utilizarea comparaţiilor în clauzele WHERE şi HAWING.

Planul de execuţie al procedurii va fi plasat în memoria cache rezervată procedurii după ce optimizatorul l-a terminat de construit şi este compus din:

- paşi de realizare a procedurii stocate,
- paşii ce descriu impunerea restricţiilor;
- paşii de declanşare lansaţi de procedura stocată.

Executarea de către server

Faza de execuţie este cea în care se prelucrează secvenţial **planul de execuţie** , fiecare pas este trimis la procesorul corespunzător de gestiune (DDL, DML, gestionarul de tranzacţie, gestionarul de prelucrare a procedurilor automate, OLE ca managerul de proceduri stocate, managerul TSQL etc), care sunt apelaţi repetat în procesul de

prelucrare. Planurile de execuţie nu sunt niciodată stocate pe disc. Într-o procedură stocată partea stocată permanent este codul sursă.

La repornirea serverului sunt eliminate toate planurile de execuţie curente şi SQL-server reface automat planul de execuţie al unei proceduri când:

- mediul de execuţie al proceduri diferă semnificativ de mediul sau de creare
- când se modifică schema-ver din tabelul sysobjects pentru cel puţin un obiect. Coloanele SCHEMA-VER şi BASE-SCHEMA-VER sunt actualizate de fiecare data când se schimbă informaţiile din baza de date.
- sau modificat statisticile pentru obiectele referite de procedură şi în memoria chache nu este disponibilă o copie.

Putem forţa recompilările planului de execuţie cu opţiunea WITH RECOMPILE.

Afişarea şi editarea procedurilor

Procedura de sistem SP-HELPTEXT de afişare a procedurii se face prin intermediul administratorului de sistem SQL. Efectuarea unei dublu clic pe procedura dorită din fereastra de administrare a serverului permite ca procedura să fie operată şi poate fi modificată în fereastra STORED PROCEDURE PROPERTIE

Exemplu de procedură cu parametri returnabili

```
IF EXISTS (SELECT name FROM sysobjects
      WHERE name = 'medie_stud' AND type = 'P')
  DROP PROCEDURE medie_stud


CREATE PROCEDURE medie_stud
  @@numest varchar(40) = '%',
  @@medie float OUTPUT
AS
 SELECT Nume
  FROM Student
 WHERE Nume LIKE @@numest
 SELECT @@medie=Avg(C.Nota)
  FROM Student S, Catalog C
 WHERE S.NrLeg=C.NrLeg and  Nume like @@numest
```

```
DECLARE @@med float
EXECUTE medie_stud '%', @@med OUTPUT

SELECT @@med

PRINT @@med   --pentru afişare mod text

DECLARE @@med float
EXECUTE medie_stud 'Ionescu', @@med OUTPUT
SELECT @@med

DECLARE @@med float
EXECUTE medie_stud 'I%', @@med OUTPUT
IF @@med < 8
 BEGIN
   PRINT ' '
   PRINT 'Media mai mică ca 8'
 END
ELSE
 BEGIN
  PRINT ' '
  PRINT 'Media mai mare ca 8'
  SELECT 'Aceasta este ' + RTRIM(CAST(@@med AS varchar(20)))
END
```

Exercitii

Scrieţi procedura care calculează media unei grupe date pentru studenţii integralişti.

Scrieţi procedura de determinare a studenţilor bursieri.