Seminar 3

Stored Procedures, Global Variables, Dynamic Execution, the OUTPUT clause. Cursors in SQL SERVER

I Stored Procedures

- contains a group of SQL statements

```
CREATE PROCEDURE (sau PROC) usp_Name
AS
   BEGIN
      ....
      ....
   END
GO -- batch separator

EXEC usp_Name ____
usp_Name ...
EXECUTE
```

aici declar variabile
mainte de GO, după GO nu o să
mai fie valabile

ex.
```
CREATE PROC usp_HotelsName
AS
SELECT hname
FROM Hotel
GO

EXEC usp_HotelsName

ALTER proc usp_HotelsNames (@no_stars DECIMAL(0,2)   <- param
AS
    SELECT hname
    FROM Hotel
    WHERE STARS = @no_stars
GO
EXEC usp_HotelNames 5.0

-- Counting hotels
CREATE PROC usp_H (@no_stars DECIMAL(10,2) = 3.0, @NoHotels
                                          INT OUTPUT )
AS
    SELECT @NoHotels = count(*)
    FROM Hotel
    WHERE stars = NoStars
```

```
GO
DECLARE @ NoH INT=0
SET @ NoH =0
```

II Global Variables

we can't create our global var.

eg. @@ ERROR

@@ IDENTITY

@@ ROWCOUNT

@@ SERVERNAME

@@ SPID (process id of the current session)

@@ VERSION

III Dynamic Execution

EXEC ('select * from Hotel')

eg.
```
declare @id_sql varchar(1000='27')
declare @select_sql varchar (max)
set @select_sql = 'select * from Hotel where hid = 27'+@id_sql
exec (@select_sql)
```

IV Output Clause

- access inserted, deleted
- UPDATE → inserted + deleted
- INSERT → inserted
- DELETE → deleted

eg.
```
declare @LogTable Table (hid INT, hname VARCHAR(200), no_rooms_
old INT, no_rooms_new INT, username VARCHAR(100), dateofupdate DATETIME)

UPDATE Hotel
SET no_rooms =100
OUTPUT inserted.hid, inserted.hname, deleted.no_rooms, inserted.no_rooms
```

INTO @LogTable
WHERE id - 27
select * from @LogTable

(for delete and insert we write them instead of update)

V Cursors

- inefficient

- DECLARE CURSOR
  - specify the result set, using a select stmt (it won't get executed)
- OPEN
  - the associated select stmt will get executed

- FETCH
  - FIRST
  - LAST
  - NEXT
  - PRIOR
  - ABSOLUTE n (n: int)
    - n > 0 — n^{th} row starting from the 1^{st} row
    - n < 0 — n^{th} row before the last row
    - n = 0 — no rows
  - RELATIVE n, n - INT
    - n > 0 — the n^{th} row after the current row
    - n < 0 — n^{th} row before the current row
    - n = 0 — cursor is set on the current row
- UPDATE / INSERT
- CLOSE - some resources are freed eg. the result set.
- DEALLOCATE