

یک.

```

create function status ()
returns varchar(10)
as
begin
    declare @num int;
    set @num = (select COUNT(id) from dbo.Student where grade < 10);
    if @num <= 1
        return 'good';
    if @num <= 3
        return 'normal';
    return 'bad';
end

```

	name	id	grade
	a	1	10
	b	2	5
	c	3	11
	d	4	12

```
SELECT dbo.status() AS ClassStatus;
```

100 %

Results Messages

	ClassStatus
1	good

دو.

```

create procedure add_num
@num int
as
begin
    declare @c int;
    set @c = (select count(id) from dbo.Student where grade <= 10);
    if @c < @num |
        update dbo.Student set grade = grade + 1 where grade >= 9 and grade < 10;
    else
        update dbo.Student set grade = grade + 0.5 where grade >= 9.5 and grade <= 10;
end

```

	name	id	grade
	a	1	10
▶	b	2	5
	c	3	11
	d	4	12
	e	5	9
	f	6	9.2
*	NULL	NULL	NULL

```
exec add_num @num = 2;
```

	name	id	grade
▶	a	1	10.5
	b	2	5
	c	3	11
	d	4	12
	e	5	9
	f	6	9.2
*	NULL	NULL	NULL

سه

```
create procedure swap (
    @a int, @b int,
    @a_o int output, @b_o int output)
as
begin
    set @a_o = @b;
    set @b_o = @a;
end
```

```
declare @a int;
declare @b int;

set @a = 123;
set @b = 24;

exec swap @a = @a, @b = @b, @a_o = @a, @b_o = @b;

print @a;
print @b;
```

100 %

Messages

24
123

چهار.

```
create function select_ali(@name varchar(10))
returns real
as
begin
return (select top 1 grade from dbo.Student where name=@name);
end
```

```
select dbo.select_ali('ali') as Ali;
```

10 %

Results Messages

	Ali
1	14

پنج.

```

select first, last,
rank() over (order by code) as CodeRank
from dbo.Person;

```

100 %

Results Messages

	first	last	CodeRank
1	Garrett	Vargas	1
2	Jillian	Carson	1
3	Michael	Blythe	1
4	Pamela	Ansman-Wolfe	1
5	Linda	Mitchell	1
6	Tsvi	Reiter	1
7	David	Campbell	7
8	Lynn	Tsoflias	7
9	Rachel	Valdez	7
10	Ranjit	Varkey Chudukatil	7
11	Shu	Ito	7
12	Tete	Mensa-Annan	7
13	José	Saraiva	7
14	Jae	Pak	7

شش.

```

create function four(@current date)
returns nvarchar
as
begin
declare @after date;
set @after = DATEADD(DAY, 4, @current);
return datename(day, @after);
end

```

```

DECLARE @LastChangeDate as date
SET @LastChangeDate = GetDate();

select dbo.four(@LastChangeDate) as Four;

```

100 %

Results Messages

	Four
1	2

هفت.

As the message suggests, you are not allowed **to use** the **UPDATE** statement inside a **function** unless the **UPDATE** statement is directed **to** a table variable local **to** the **function**

```
create function updoot()  
returns int  
as  
begin  
    update dbo.Student set grade = grade + 1;  
    declare @a int;  
    set @a = 12;  
    return @a;  
end
```

10 %

Messages

Msg 443, Level 16, State 15, Procedure updoot, Line 5 [Batch Start Line 0]
Invalid use of a side-effecting operator 'UPDATE' within a function.