

# Zh feladatok

## 1. feladat

1A: Készítsen olyan SQL lekérdezést a Northwind adatbázisban, mely termék-kategóriánként (név) és évenként adja meg a 90 dollárnál drágább egységárú termékek rendelési tételeken szereplő darabszámait összegét!

Segítség: A rendelési tételek az [order details] táblában találhatóak. Egy tételen a rendelt termék darabszámát a tábla quantity nevű mezője mutatja. Az adatbázis sémáját lásd a jegyzet elején.

1B: Az előző lekérdezés eredménytábláját pivotálja az egyik alkalmasan választott mező szerint.

```
1  -- 1a
2  select c.CategoryName, YEAR(o.OrderDate) as OrderYear, SUM(od.Quantity) as Qty
3  from [Order Details] od
4  join Products p on p.ProductID = od.ProductID
5  join Categories c on c.CategoryID = p.CategoryID
6  join Orders o on o.OrderID = od.OrderID
7  where p.UnitPrice > 90
8  group by c.CategoryName, YEAR(o.OrderDate)
9  order by c.CategoryName, OrderYear
10
11 -- 1b
12 select * from (
13     select c.CategoryName, YEAR(o.OrderDate) as OrderYear, od.Quantity as Qty from
14     [Order Details] od
15     join Products p on p.ProductID = od.ProductID
16     join Categories c on c.CategoryID = p.CategoryID
17     join Orders o on o.OrderID = od.OrderID
18     where p.UnitPrice > 90
19 ) as source_table
20 pivot (
21     sum(Qty)
22     for OrderYear in ([1996],[1997],[1998])
23 ) as pivot_table
```

	CategoryName	OrderYear	Qty
1	Beverages	1996	140
2	Beverages	1997	223
3	Beverages	1998	260
4	Meat/Poultry	1996	123
5	Meat/Poultry	1997	397
6	Meat/Poultry	1998	321

	CategoryName	1996	1997	1998
1	Beverages	140	223	260
2	Meat/Poultry	123	397	321

## 2. feladat

Készítsen egy felhasználói függvényt, amely paraméterként kapja egy ügynök (Employee) azonosítóját, és meghatározza, hogy hány terület van hozzárendelve.

Kiegészítés: Ha nincs az Employees táblában a megadott azonosítószámmal ügynök, akkor a visszaadott érték NULL legyen.

```
1  select count(*) from EmployeeTerritories where EmployeeID = 5
2
3  drop function if exists dbo.countTerritories
4  go
5  create function dbo.countTerritories(@user_id int) returns int
6  as
7  begin
8      declare @result int
9
10     if exists(select 1 from EmployeeTerritories where EmployeeID = @user_id)
11     begin
12         select @result = count(*) from EmployeeTerritories where EmployeeID =
13         @user_id
14     end
15     else
16         set @result = NULL
17
18     return @result
19 end
20 go
21 select dbo.countTerritories(5) as CountResult
```

	CountResult
1	7

### 3. feladat

Írjon egy eljárást, mely a Northwind cég menedzserének munkáját segíti az alábbiak szerint.

- Az eljárásnak egy paramétere van: egy dolgozó azonosítója. Az eljárás a paraméterül kapott azonosítóra ellenőrzi az 1. feladatban megírt függvény felhasználásával, hogy nincs-e már az illető dolgozónak 5-nél több területe: ha van, hibaüzenettel kilép.
- Ha nincs, akkor ellenőrzi, hogy van-e olyan terület, melynek még nincs felelőse. Ha minden területnek van már felelőse, hibaüzenettel kilép.
- Ha van még szabad terület, akkor hozzárendeli a dolgozóhoz a betűrendben első szabad területet, és megemeli az alkalmazott fizetését 25%-kal. Ezután visszaírja a dolgozóhoz hozzárendelt terület nevét.
- Ha bármilyen hiba történik, a script kiírja a hibakódot és a rendszertől származó angol nyelvű hibaüzenetet.

A tesztelést végző scriptben szerepeljen az eredeti helyzet helyreállítása és az eredmény ellenőrzése! A tesztelést a script minden ágára végezze el.

Segítség: A dolgozók az Employees táblában találhatók, a területek a Territories táblában, a dolgozók és a területek összerendelése az EmployeeTerritories táblában. A területek betűrendben való rendezéséhez a Territories tábla TerritoryDescription mezőjét használjuk. Az adatbázis sémáját lásd a jegyzet elején.

```
1 go
2 drop procedure if exists assignTerritoryToEmployee
3 go
4 create procedure assignTerritoryToEmployee @user_id int
5 as
6 begin
7     begin try --hibakezelés
8         declare @territory_count int;
9         declare @new_territory_id nvarchar(20);
10        declare @new_territory_desc nvarchar(50);
11
12        ---mehívjuk az előző feladat függvényét
13        set @territory_count = dbo.countTerritories(@user_id)
14
15        --ha nincs ilyen dolgozó hibát dobunk
16        if @territory_count is null
17        begin
18            raiserror('Nincs ilyen dolgozó!', 16, 1)
19            return
20        end
21        else if @territory_count >= 5
22        begin
23            raiserror('A dolgozóhoz nem lehet több területet hozzárendelni!', 16, 1)
24        end
25
26        -- ellenőrizzük hogy van-e olyan terület ami még senkihez nem tartozik
27        select top 1 @new_territory_id=TerritoryID,
28        @new_territory_desc=TerritoryDescription from Territories T
29        where not exists (select 1 from EmployeeTerritories ET where ET.TerritoryID =
30        T.TerritoryID)
31        order by T.TerritoryDescription
32
33        -- ha nincs ilyen akkor hibát dobunk
34        if @new_territory_id is null
35        begin
36            raiserror('Nincs szabad terület', 16, 1)
37            return
38        end
39
40        -- ha van üres rendeljük hozzá
41        insert into EmployeeTerritories (EmployeeID, TerritoryID)
42        values (@user_id, @new_territory_id);
```

```

42      -- fizetés emelés
43      update Employees set Salary = Salary * 1.25 where EmployeeID = @user_id
44
45      -- kiírjuk amit hozzárendeltünk
46      print 'Hozzárendelt terület: ' + @new_territory_desc;
47
48  end try
49  begin catch
50      -- hiba kiírása
51      print 'Hiba történt: ' + error_message()
52  end catch
53 end
54
55 -- tesztelés
56 select * into #BackupEmployeeTerritories from EmployeeTerritories --biztonsági
mentések tept táblákba
57 select EmployeeID, Salary into #BackupEmployees from Employees;
58
59 -- tesztelés: már több mint 5 területe van
60 delete from EmployeeTerritories where EmployeeID = 1; --kiürítjük az 1-es dolgozót
61 insert into EmployeeTerritories (EmployeeID, TerritoryID) select top 6 1, TerritoryID from
Territories; --hozzáadunk 6 területet
62 exec assignTerritoryToEmployee @user_id = 1; -- futtatjuk az eljárást -> hiba több
mint 5
63      Hiba történt: A dolgozóhoz nem lehet több területet hozzárendelni!
64
65 -- tesztelés: nincs szabad hely
66 delete from EmployeeTerritories where EmployeeID = 1; --ürítjük az 1 est
67 -- minden szabad területet lefoglalok
68 insert into EmployeeTerritories (EmployeeID, TerritoryID) select 1, TerritoryID from
Territories; --hozzáadjuk a 1-es dolgozóhoz mindet
69 exec assignTerritoryToEmployee @user_id = 2; -- futtatjuk az eljárást 2es dolgozóra ->
hiba: nincs több szabad hely (mert az 1-eshez adtunk mindent)
70      Hiba történt: A dolgozóhoz nem lehet több területet hozzárendelni!
71
72 -- tesztelés: kevesebb mint 5 területe van és hely is van
73 delete from EmployeeTerritories
74 insert into EmployeeTerritories (EmployeeID, TerritoryID)
75     select 1, TerritoryID from Territories
76     order by TerritoryDescription offset 0 rows fetch next 2 rows only;
77
78 -- megnézzük a dolgozó fizuját
79 select Salary from Employees where EmployeeID = 1;
80 exec assignTerritoryToEmployee @user_id = 1; -- futtatjuk az eljárást -> területet
kapott és fizetés emelést
81
82 -- megnézzük a dolgozó fizuját emelkedett-e
83 select Salary from Employees where EmployeeID = 1;
84 -- megnézzük kapott-e területet
85 select * from EmployeeTerritories where EmployeeID = 1;

```

	Salary
1	125,00

	EmployeeID	TerritoryID
1	1	01730
2	1	30346
3	1	44122
4	1	78759
5	1	98004

```

86  -- visszaállítjuk az eredeti állapotot
87  delete from EmployeeTerritories; -- urítjuk sz összerendelést
88
89  insert into EmployeeTerritories (EmployeeID, TerritoryID)
90      select EmployeeID, TerritoryID from #BackupEmployeeTerritories; --visszaállítjuk az
    eredeti összerendelést
91
92  -- visszaállítjuk a fizetéseket
93  update Employees set Salary = b.Salary from Employees e join #BackupEmployees b
    on b.EmployeeID = e.EmployeeID;
94
95  -- temp táblák törlése
96  drop table #BackupEmployees
97  drop table #BackupEmployeeTerritories

```

## 4. feladat

Amikor egy alkalmazotthoz hozzárendelnek egy területet (akár az előző eljárás futtatásával, akár az EmployeeTerritories táblába történő adatbeszúrás esetén), az alkalmazott fizetése automatikusan emelkedjen 25%-kal. Ellenkező esetben, ha elvesznek tőle egy területet, akkor automatikusan csökkenjen a fizetése 20%-kal.

```

1  go
2  drop trigger if exists update_salary_trigger
3
4  go
5  create trigger update_salary_trigger
6  on EmployeeTerritories after insert, delete as
7  begin
8      -- ha insert történik (fizu emelés)
9      if exists(select * from inserted)
10     begin
11         update Employees set Salary = Salary * 1.25 where EmployeeID in (select
            EmployeeID from inserted)
12     end
13     -- ha delete történik (fizu csökkentés)
14     if exists(select * from deleted)
15     begin
16         update Employees set Salary = Salary * 0.80 where EmployeeID in (select
            EmployeeID from deleted)
17     end
18 end
19
20 -- trigger tesztelése: létrehozunk egy uj területet amit hozzáadunk az 1-eshez -> fizu
    növekedés
21 insert into EmployeeTerritories (EmployeeID, TerritoryID)
22     select 1, TerritoryID from Territories where TerritoryDescription = 'Denver'

```

```

23
24 SELECT Salary FROM Employees WHERE EmployeeID = 1; -- fizu ellenorzes
25
26 -- majd eltavolitjuk -> fizu csokkenes
27 delete from EmployeeTerritories where EmployeeID = 1 and TerritoryID = 80202
28
29 SELECT Salary FROM Employees WHERE EmployeeID = 1; -- fizu ellenorzes

```

	Salary
1	125,00

	Salary
1	100,00

## 5. feladat

Módosítsa a 3. feladatban elkészült eljárást úgy, hogy az eljárás tranzakciót futtasson.

```

1  -- a 3-as feladatban létrehozott tárolt eljárás tranzakciós változata
2  go
3  drop procedure if exists assignTerritoryToEmployee
4  go
5  create procedure assignTerritoryToEmployee @user_id int
6  as
7  begin
8      BEGIN TRANSACTION; -- Tranzakció start
9
10     begin try --hibakezelés
11         declare @territory_count int;
12         declare @new_territory_id nvarchar(20);
13         declare @new_territory_desc nvarchar(50);
14
15         ---mehívjuk az előző feladat függvényét
16         set @territory_count = dbo.countTerritories(@user_id)
17
18         --ha nincs ilyen dolgozó hibát dobunk
19         if @territory_count is null
20         begin
21             raiserror('Nincs ilyen dolgozó!', 16, 1)
22             return
23         end
24         else if @territory_count >= 5
25         begin
26             raiserror('A dolgozóhoz nem lehet több területet hozzárendelni!', 16, 1)
27         end
28
29         -- ellenőrizzük hogy van-e olyan terület ami még senkihez nem tartozik
30         select top 1 @new_territory_id=TerritoryID,
31         @new_territory_desc=TerritoryDescription from Territories T
32         where not exists (select 1 from EmployeeTerritories ET where ET.TerritoryID =
33         T.TerritoryID)
34         order by T.TerritoryDescription
35
36         -- ha nincs ilyen akkor hibát dobunk

```



```

35     if @new_territory_id is null
36     begin
37         raiserror('Nincs szabad terület',16,1)
38         return
39     end
40
41     -- ha van üres rendeljük hozzá
42     insert into EmployeeTerritories (EmployeeID, TerritoryID)
43     values (@user_id, @new_territory_id);
44
45     -- fizetés emelés
46     update Employees set Salary = Salary * 1.25 where EmployeeID = @user_id
47
48     -- kiírjuk amit hozzárendeltünk
49     print 'Hozzárendelt terület: ' + @new_territory_desc;
50
51     COMMIT -- tranzakció end
52
53     end try
54     begin catch
55         -- hiba kiírása
56         ROLLBACK -- tranzakció visszavonás hiba esetén
57
58         print 'Hiba történt: ' + error_message()
59     end catch
60 end
61
62 exec assignTerritoryToEmployee @user_id = 1;
63

```

## 6. feladat

Készítsen olyan szkriptet, amely az angol ügynökök rekordjain iterál, és kiírja minden ügynök nevét, és hogy milyen területek tartoznak hozzájuk. A feladatot kurzor segítségével oldja meg

```

5. Buchanan Stev
- Edison
- Fairport
- Mellville
- Morristown
- New York
- Providence
6. King Michael:
- Bellevue
- Phoenix

```

```

1 declare @EmployeeID int, @FullName nvarchar(100), @counter int = 1;

```

```

2
3 declare employee_cursor cursor for
4 select EmployeeID, FirstName + ' ' + LastName
5 from Employees
6 where Country = 'UK'
7 order by LastName, FirstName
8
9 open employee_cursor
10 fetch next from employee_cursor into @EmployeeID, @FullName
11
12 while @@FETCH_STATUS = 0
13 begin
14     print cast(@counter as varchar) + '.' + @FullName + '.'
15
16     -- Most lekérdezzük a területeket ehhez az ügynökhöz
17     declare @Territory nvarchar(50)
18     declare territory_cursor cursor for
19
20     select T.TerritoryDescription from Territories T join EmployeeTerritories ET on
21     T.TerritoryID = ET.TerritoryID
22     where ET.EmployeeID = @EmployeeID
23
24     open territory_cursor
25     fetch next from territory_cursor into @Territory
26
27     while @@FETCH_STATUS = 0
28     begin
29         print ' - ' + @Territory
30         fetch next from territory_cursor into @Territory
31     end
32
33     close territory_cursor
34     deallocate territory_cursor
35
36     set @counter = @counter + 1
37     fetch next from employee_cursor into @EmployeeID, @FullName
38 end
39 close employee_cursor
40 deallocate employee_cursor

```

```

1. Steven Buchanan:
- Providence
- Morristown
- Edison
- New York
- New York
- Mellville
- Fairport
2. Anne Dodsworth:
- Hollis
- Portsmouth
- Southfield
- Troy
- Bloomfield Hills
- Roseville
- Minneapolis

```