

SELECT INTO deyimi

SELECT INTO deyimi daha çok tabloların kopyalarını yapmakta veya bir kümenin kopyasını disk üzerinde oluşturmak için kullanılır.

Yapı

```
SELECT kolon_ad(ları) INTO [bir_baska_veritabanı.dbo.]yeni_tablo_adı  
FROM kaynak
```

Yedekleme amacıyla kopya oluşturma

Örneğin Kişiler tablosunda bir takım değişiklikler yapılacaksa önceden Kişiler tablosunun yedeğini almakta yarar vardır.

```
SELECT * INTO Kişiler_yedek  
FROM Kişiler
```

Bu örnekte Kişiler tablosunun Kişiler_yedek tablosuna bir kopyası yazılmaktadır. Bu tablonun aynı server üzerinde olan bir başka veritabanına kopyası da yapılabilir.

Örnek:

```
SELECT * INTO Northwind.dbo.Kişiler  
FROM Kişiler
```

Bu SQL deyimi Kişiler tablosunun bir kopyasını Northwind veri tabanına adı Kişiler olan bir tabloya kopyalayacaktır.

Her iki durumda da INTO dan sonra gelen tablo adı, kopyanın yapılacağı veritabanında bulmamalıdır.

Tüm bir veri tabanını kopyalamak yerine, INTO deyimi ile bir SELECT sonucu elde edilen bir sonuç kümesini bir tablo haline getirebiliriz.

Örnek:

```
SELECT    K.AD, K.SOYAD, A.fkSIRANO, D.DERSKOD, D.DERSAD, A.YIL,  
A.Donem, A.BasNot  
INTO TRANSCRIPT_9  
FROM      KUTUK A INNER JOIN  
          DERS D ON A.fkDersSIRANO = D.pkDersSIRANO INNER  
JOIN      Kişiler K ON A.fkSIRANO = K.SIRANO  
WHERE     (A.fkSIRANO = 9)
```

Bu SQL deyimi 9 nolu öğrencinin aldığı dersleri, derslerin alındığı dönemleri ve notları TRANSCRIPT_9 adlı bir tabloya kopyalamaktadır.

CREATE VIEW deyimi

Yapı

```
CREATE VIEW View_adı  
AS  
SELECT deyimi
```

Önce **view** deyiminin ne olduğunu öğrenelim.

Bir view, bir SELECT deyimi ile elde edilen bir sonuç kümesine dayalı oluşturulan bir sanal tablodur. Bunu anlamak için aşağıdaki örneği inceleyelim.

Örnek:

```
CREATE VIEW TRANSCRIPT  
AS  
SELECT A.fkSIRANO, D.DERSKOD, D.DERSAD, A.YIL, A.Donem,  
A.BasNot  
FROM KUTUK A INNER JOIN  
DERS D ON A.fkDersSIRANO = D.pkDersSIRANO
```

Bu SQL deyimi bize SQL Server içinde her zaman kullandığımız veritabanında **TRANSCRIPT** adında bir view oluşturur. Bu view bir sanal tablo olarak düşünülebilir. Bu sanal tablonun kolonları ve kolonların alındığı özgün alanlar düşünülerek aşağıdaki gibidir:

fkSIRANO	int
DERSKOD	varchar(6)
DERSAD	varchar(30)
YIL	smallint
Donem	smallint
BasNot	varchar(2)

Artık bu sanal tabloyu SELECT deyimlerinde kullanabiliriz.

Örnek:

```
SELECT    K.SIRANO, K.AD, K.SOYAD, T.DERSKOD, T.DERSAD, T.YIL,
T.Donem, T.BasNot
FROM      TRANSCRIPT T INNER JOIN
          Kişiler K ON T.fkSIRANO = K.SIRANO
```

Bir başka view örneği verelim.

Örnek:

```
CREATE VIEW TEKRAR_EDEN_ILCE
AS
SELECT    COUNT(ILCEAD) AS SAY, ILCEAD
FROM      ILCE
GROUP BY ILCEAD
HAVING    (COUNT(ILCEAD) > 1)
```

Şimdi bu view bir SELECT deyiminde kullanalım

Örnek:

```
SELECT    I.ILAD, C.ILCEAD, V.SAY
FROM      ILCE C INNER JOIN
          TEKRAR_EDEN_ILCE V ON C.ILCEAD = V.ILCEAD INNER JOIN
          IL I ON C.ILKOD = I.ILKOD
```

Not: Veri tabanı view da oluşan datayı saklamaz. Dolayısıyla bir viewun geçtiği SQL deyimi her çalıştırıldığında ilgili view sanal kümeyi yeniden oluşturur.