

**T.C.
MİLLÎ EĞİTİM BAKANLIĞI**

BİLİŞİM TEKNOLOJİLERİ

**AĞ VERİ TABANINDA TABLO İŞLEMLERİ
481BB0100**

Ankara, 2012

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

İÇİNDEKİLER

AÇIKLAMALAR	ii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	3
1. TABLO TANIMLAMA.....	3
1.1. Tablo Tanımlama ve Veri Tipleri	3
1.1.1. Kesin Sayısal Veri Tipleri	7
1.1.2. Yaklaşık Sayısal Veri Tipleri.....	8
1.1.3. Parasal Veri Tipleri.....	8
1.1.4. Tarih ve Zaman Veri Tipleri.....	8
1.1.5. Karakter Veri Tipleri	9
1.1.6. Unicode Karakter Veri Tipleri.....	9
1.1.7. İkili (Binary) Veri Tipleri	10
1.2. Anahtar Sahaları.....	10
1.2.1. Birincil Anahtar (primary key)	10
1.2.2. Yabancı Anahtar (foreign key)	11
1.3. Tablolar Arası İlişkiler	11
1.4. İlişki Diyagramı Oluşturma	14
UYGULAMA FAALİYETİ	18
ÖLÇME VE DEĞERLENDİRME	22
ÖĞRENME FAALİYETİ-2	23
2. VERİ TABANI İŞLEMLERİ.....	23
2.1. Veri Tabanı Yedeğini Alma.....	23
2.2. Yedeği Alınmış Veri Tabanını Geri Yükleme	26
2.3. Otomatik Yedek Alma Görevi Oluşturma	33
2.4. Veri Tabanı Sunucu İşlemleri	40
2.4.1. Harici Veri Tabanını Sunucuya Dâhil Etme	40
2.4.2. Veri Tabanının Suncuyla İlişğini Kesme.....	44
2.5. Veri Tabanının Script'ini Oluşturma	44
UYGULAMA FAALİYETİ	49
ÖLÇME VE DEĞERLENDİRME	52
MODÜL DEĞERLENDİRME	53
CEVAP ANAHTARLARI.....	54
KAYNAKÇA	55

AÇIKLAMALAR

KOD	481BB0100
ALAN	Bilişim Teknolojileri
DAL/MESLEK	Veritabanı Programcılığı
MODÜLÜN ADI	Ağ Veritabanında Tablo İşlemleri
MODÜLÜN TANIMI	Ağ veritabanında tablo işlemleri ile ilgili bilgilerin verildiği öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	“Ağ Veritabanı Kurulumu” modülünü başarmış olmak
YETERLİK	Tablo işlemlerini yapmak
MODÜLÜN AMACI	Genel Amaç Bu modül ile gerekli ortam sağlandığında; sunucu üzerinde tablo tanımlama, diyagram yönetme, veri aktarma, var olan veri tabanından yeni veri tabanı oluşturma işlemlerini yapabileceksiniz. Amaçlar 1. Tabloları tanımlayabilecek ve diyagramları yönetebileceksiniz. 2. Veri Tabanı aktarma işlemleri yapabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Ortam: Bilgisayar Laboratuvarı Donanım: Bilgisayar, lisanslı ağ veritabanı yazılımları
ÖLÇME VE DEĞERLENDİRME	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığımız bilgi ve becerileri ölçerek sizi değerlendirecektir.

GİRİŞ

Sevgili Öğrenci,

Bilişimin temel konusu olan veri tabanı; dağınık şekilde duran ve işe yaramaz halde olan bilgilerin, işlenebilir, içinde arama yapılabilir, sıralanabilir hale gelmesini sağlayan bir yardımcı programdır. Aynı zamanda bilgilerin kullanıcıya, hızlı bir şekilde ulaştırılmasını da sağlar.

Bu modül sayesinde günlük hayatta fark etmediğiniz detaylar dikkatinizi çekecek, disiplinli olarak çevrenizdeki bilgi parçalarını planlayıp organize edeceksiniz. Detaylara özen gösterip zamanınızı etkin olarak kullanacaksınız.

Bu modülde “tablolar” ile ilgili konuları göreceksiniz. *Tablo, bir veri tabanı dosyasının temel ögesidir.*

ÖĞRENME FAALİYETİ-1

AMAÇ

Bu öğrenme faaliyetinde, ağ veri tabanı yazılımında tablolar oluşturabilecek ve diyagramları yönetebileceksiniz.

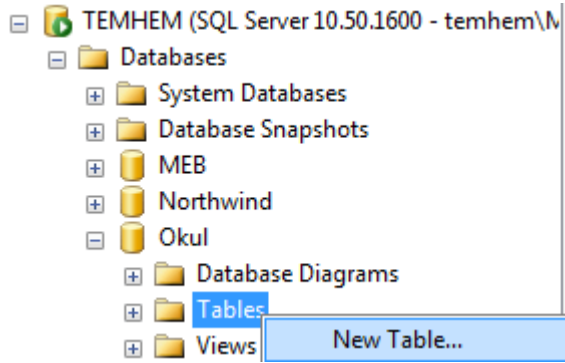
ARAŞTIRMA

- Veritabanında, verilerin saklanma biçimlerini araştırınız.
- Bir veritabanı için tabloların önemini araştırınız.

1. TABLO TANIMLAMA

1.1. Tablo Tanımlama ve Veri Tipleri

Tabloyu oluşturacağımız veritabanı düğümünü açıyoruz. Açılan düğümde **Tables** düğümü üzerinde sağ tuşa tıklayarak açılan menüden, **New Table** seçeneğini seçiyoruz.



Resim 1.1: Object Explorer penceresi yeni tablo ekleme

Karşımıza Resim 1.2'deki gibi bir pencere gelecektir.

Column Name	Data Type	Allow Nulls
		<input type="checkbox"/>

Resim 1.2: Kayıt özellikleri

Column Name: Tablo içinde yer alacak sütunların belirlendiği bölümdür.

Data Type: Sütunun hangi tipde veri tutacağını belirlediği bölümdür.

Allow Nulls: Sütuna değer girilmesi gerekliliğinin zorlandığı bölümdür. Yani bir sütuna değer girilmesi zorunluluğu yoksa Allow Nulls kutucuğu işaretlenmelidir. Eğer kutucuk işaretlenmedi ve kayıt eklenirken o alana değer girilmemişse sorgumuz hata verecek ve kayıt eklenmeyecektir.

Column Name	Data Type	Allow Nulls
ogrenciID	int	<input type="checkbox"/>
ogrenciNo	int	<input type="checkbox"/>
ogrenciAd	nvarchar(50)	<input type="checkbox"/>
ogrenciSoyad	nvarchar(50)	<input type="checkbox"/>
ogrenciAlan	int	<input type="checkbox"/>
		<input type="checkbox"/>

Resim 1.3: Oluşturulacak tablonun son hali

Resim 1.3’de görüldüğü gibi sütun isimlerini, sütunların tutacağı verinin tipini ve sütuna veri girişinin zorunlu olduğu durumları belirledik.

Şimdi burada bir ayrıntıya değinelim. Önceki bilgilerimizden hatırlayacağımız üzere tablodaki kayıtların birbirine karışmasını engellemek için benzersiz değer tutan bir sütun belirliyorduk. Genellikle bu sütuna değer girmiyor, otomatik olarak kendinin artırım yapması için bir tip tanımlıyorduk. SQL Server da bu işlem için sütunun veri tipini int tanımlıyoruz. Daha sonra sütun özellikleri penceresinde **Identity Specification** özelliğini Resim 1.4’teki gibi ayarlıyoruz.

Column Properties	
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes

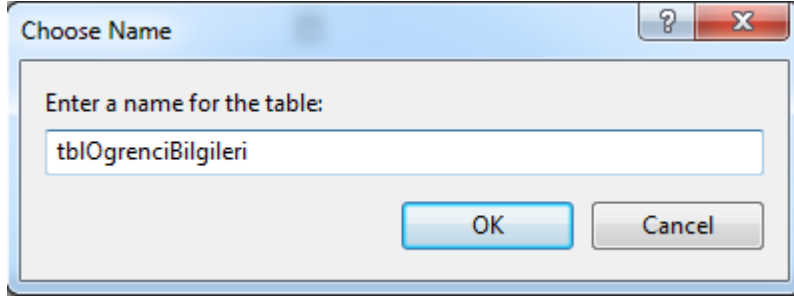
Resim 1.4: Kolon özellikleri penceresi

(Is Identity): Arttırımın otomatik olarak yapılıp yapılmayacağını belirliyoruz. Yes seçilmiş ise otomatik artırım yapılacaktır.

Identity Increment: Kaçar kaçar artırım yapılacağını belirliyoruz.

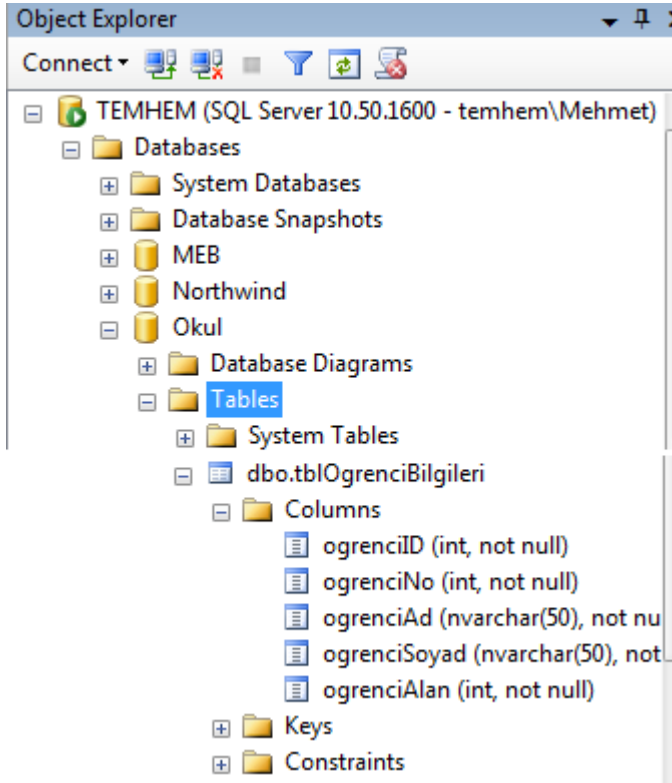
Identity Seed: Arttırıma ilk hangi değerden başlanacağını belirliyoruz.

Kaydet butonuna basıyoruz. Karşımıza tabloya isim vermemizi sağlayan **Choose Name** penceresi çıkacaktır. Burda tablomuza içinde tuttuğu verileri hatırlatacak şekilde bir isim vermemiz gerekmektedir.



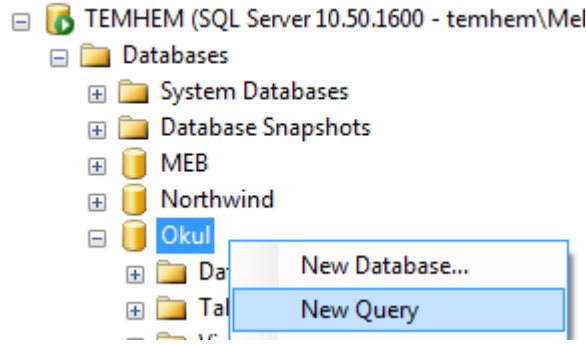
Resim 1.5: Tablo isimlendirme penceresi

Tablomuzaya bir isim verdikten sonra OK butonuna basarak tablomuzay oluşturuyoruz. Resim 1.6'de görüldüğü gibi tablomuz Tables düğümü altında yer almaktadır.



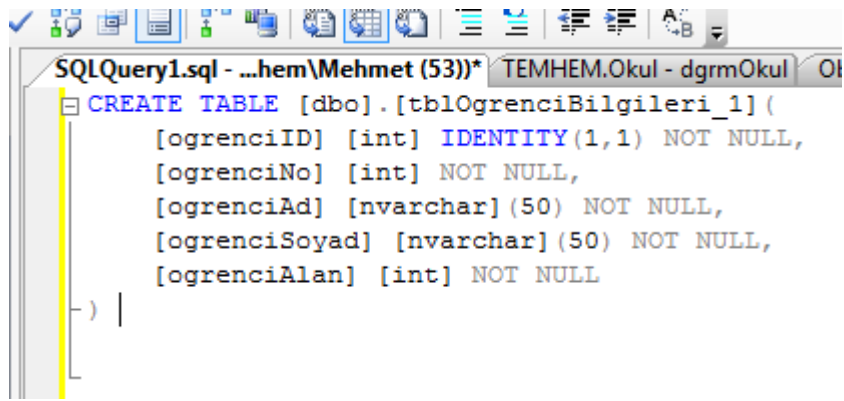
Resim 1.6: Object Explorer penceresi tablo listesi

Şimdi de SQL sorgusuyla nasıl tablo oluşturabileceğimize bakalım. Öncelikle tabloyu oluşturacağımız veri tabanı üzerinde sağ tuşa tıklıyoruz, açılan menüden New Query seçeneğini seçiyoruz.



Resim 1.7: Object Explorer penceresi yeni sorgu sayfası açma

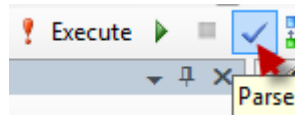
Açılan SQL Query penceresine tablo oluşturma sorgu cümlesini yazıyoruz.



Resim 1.8: Sorgu penceresi

Not: Identity anahtar kelimesi, o sütunun otomatik sayı gibi davranmasını sağlar.

Yazmış olduğumuz sorguda hata olup olmadığını denemek için parse butonuna tıklıyoruz. Parse butonu, sadece sorgumuzu derleyecek; sorguyu çalıştırıp herhangi bir sonuç üretmeyecektir.



Resim 1.9: Sorgu derleme ve çalıştırma butonları

Yazdığımız sorguda bir hata yoksa Execute butonuna tıklayarak sorgunun çalıştırılmasını sağlıyoruz. Tables düğümü altına baktığımızda yeni tablomuzun oluşturulduğunu göreceksiniz.

Soru: Veli bilgilerini tutan bir tablo oluřturunuz.

Column Name	Data Type	Allow Nulls
veliID	int	<input type="checkbox"/>
veliAd	nvarchar(50)	<input type="checkbox"/>
veliSoyad	nvarchar(50)	<input type="checkbox"/>
veliTelefon	nvarchar(12)	<input checked="" type="checkbox"/>
veliAdres	nvarchar(250)	<input checked="" type="checkbox"/>

Column Name
veliID
veliAd
veliSoyad
veliTelefon
veliAdres

Resim 1.10: tblVeliBilgileri tablosu ayrıntıları

1.1.1. Kesin Sayısal Veri Tipleri

Veri Tipi	Boyut	Minimum Deęer	Maximum Deęer
Bit	1 bit	0	1
tinyint	1 byte	0	255
smallint	2 byte	-32768	32767
İnt	4 byte	-2, 147, 483, 648	2, 147, 483, 647
Bigint	8 byte	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
decimal(p,s)	5-17 byte		
numeric(p,s)	5-17 byte		

bit daha çok evet-hayır, doğru-yanlıř gibi durumlarda kullanılan bir veri tipidir.

tinyint, 0-255 arasındaki tamsayıları tutmak için kullanılan veri tipidir.

smallint, **int**, **bigint** tabloda belirtilen aralıklarda negatif ve pozitif sayı tutmak için kullanılan veri tipleridir.

decimal(p,s) ise ondalık sayıları tutmak için kullanılan veri tipidir. İlk parametre (p) verinin kaç karakterden oluřtuęunun belirtildięi, ikinci parametre(s) ise toplam uzunluęun kaç karakterinin ondalık kısım olduęunun belirtildięi bölümlerdir.

Örneęin: decimal (7,2) řeklinde bir tanımlamada girilecek olan veri, 56347,12 řeklinde bir sayıdan oluřmalıdır.

İlk Parametre(p)	Boyut
1 – 9	5 byte
10-19	9 byte
20-28	13 byte
29-38	17 byte

numeric(p,s), kullanımı decimal ile aynıdır.

1.1.2. Yaklaşık Sayısal Veri Tipleri

Veri Tipi	Boyut	Minimum Değer	Maximum Değer
Real	4 byte		
Float	n değerine bağlıdır.		

n Değeri	Hassasiyet	Boyut
1-24	7 karakter	4 byte
25-53	15 karakter	8 byte

float veri tipi, ondalık sayıları tutmak için kullanılır. Burada tutulan veri kesin değeri ifade etmediğinden yaklaşık sayısal veri tipi olarak geçmektedir.

Öreniğin sütunun tutacağı verinin tipini float olarak belirlediyssek ve bu sütuna;

2,18181818181818181818181818181818 gibi bir değer girersek bu sütunda saklanacak olan veri;

2,1818181818181819 şeklinde olacaktır.

real veri tipi de float(24) şeklinde davranan bir veri tipidir.

Eğerki yukardaki sayıyı tutacak sütunun veri tipini real olarak belirleseydik saklanan veri;

2,18181819 şeklinde olacaktır.

1.1.3. Parasal Veri Tipleri

Veri Tipi	Boyut	Minimum Değer	Maximum Değer
money	8 byte	-922 337 203 685 47,5808	922 337 203 685 477,5807
smallmoney	4 byte	-214 748,3648	214 748,3647

Parasal verileri tutmaya yarar.

1.1.4. Tarih ve Zaman Veri Tipleri

Veri Tipi	Boyut	Minimum Değer	Maximum Değer
Date	3 byte	01.01.0001	31.12.9999
time(n)	3-5 byte	00:00:00.0000000	23:59:50.9999999
datetime	8 byte	01.01.1753	31.12.9999
smalldatetime	4 byte	01.01.1990	06.06.2079
datetime2	6-8 byte	01.01.0001 00:00:00.0000000	31.12.9999 23:59:59.9999999
datetimeoffset	8-10 byte	01.01.0001 00:00:00.0000000[+ - }hh:mm]	31.12.9999 23:59:59.9999999[+ - }hh:mm]

date, **datetime** ve **smalldatetime** tabloda belirtilen aralıklarda tarih bilgisini tutan veri tipleridir.

time(n) saat bilgisini tutan veri tipidir. “n” değerine göre saniyeden sonraki ondalık kısım belirlenir. “n” değeri 0-2 arasında ise 3 byte, 3-4 arasında ise 4 byte, 5-7 arasında ise 5 byte yer kaplar. “time” olarak kullanıldığında “n” değeri varsayılan olarak 7’dir.

datetime2 veri tipi, date ve time veri tiplerinin birleşmesinden oluşan bir veri tipidir. Datetime2 şeklinde kullandığımızda varsayılan olarak saniyeden sonraki ondalık kısım, 7 basamağa kadar değer tutabilir. Datetime(n) şeklinde de kullanmak mümkündür. “n” değeri 0-2 arasında ise 6 byte, 3-4 arasında ise 7 byte ve 5-7 arasında ise 8 byte yer kaplar.

datetimeoffset aslında baktığımızda datetime2 veri tipi ile aynı aralıkta veri tutar. Farkı ise dünya saatleri arasındaki farkı da tutmamıza olanak tanınmasıdır. Projemizde farklı ülkelerin tarih ve saat bilgilerini tutacaksa bu veri tipini kullanmak uygundur. Datetimeoffset(n) şeklinde kullanmakta mümkündür. “n” değeri 0-2 arasında ise 8 byte, 3-4 arasında ise 9 byte ve 5-7 arasında ise 10 byte yer kaplar.

Örneğin veri tipi, datetimeoffset olarak tanımlanmış bir kolonda tutulabilen veri aşağıdaki gibidir:

(06.03.2012 12:50:23.2723232 +02:00) şeklindedir.

1.1.5. Karakter Veri Tipleri

Veri Tipi	Boyut	Maximum Değer
char(n)	Değişken	8000 karakter
varchar(n)	Değişken	8000 karakter
varchar(max)	Değişken	1.073.741.824 karakter
Text	2 gigabyte	1.073.741.824 karakter

Karakter veri tipleri, tabloda belirtilen maximum değer ölçüsünde karakter tutabilen veri tipleridir. ANSI standartlarına uyan herhangi bir karakter 1 byte yer kaplar.

“char” veri tipiyle varchar veri tipi arasındaki fark; char belirtilen n değeri kadar yer kaplarken varchar içinde tuttuğu karakter kadar yer kaplar.

Örneğin; char(10) şeklinde tanımlanmış bir alana kaç karakterlik veri girersek girelim her bir alan için 10 karakterlik bir yer ayrılacaktır. varchar(10) şeklinde tanımlanmış bir alana kaç karakterlik veri girdiysek o kadar yer ayrılacaktır.

1.1.6. Unicode Karakter Veri Tipleri

Veri Tipi	Boyut	Maximum Değer
nchar(n)	Değişken	4000 karakter
nvarchar(n)	Değişken	4000 karakter
nvarchar(max)	Değişken	536.870.912 karakter
Ntext	2 gigabyte	536.870.912 karakter

Karakter veri tipleriyle kullanımları aynıdır. Farkı ise Unicode karakterleri tutabilmeleridir. Unicode karakterleri hafızada tutabilmek için 1 byte yeterli değildir. Fazladan 1 byte daha kullanılır ve 2 byte yer kaplar.

Unicode karakterler, farklı dillere özgü karakterlerdir. Örneğin Türkçede kullandığımız ç,ş,ğ,ö gibi karakterler ANSI standartlarında yer almamaktadır. Dolayısıyla bu karakterleri barındıran kolonlar oluşturacaksa veri kaybı olmaması için Unicode karakter veri tiplerinden birini kullanmalıyız.

Not: Eğer veritabanımızdaki kayıtlarda birden fazla dil kullanılacaksa veya veritabanının dil seçeneğinden (collation) farklı bir dil ile kayıt yapılacaksa unicode karakter veri tipleri kullanılmalıdır.

1.1.7. İkili (Binary) Veri Tipleri

SQL Server’de dosyaları saklamak için kullanılan veri tipleridir.

binary(n): “n” değeri 1 ile 8000 arasında olabilir. “n” byte kadar yer kaplar. Boyutu sabittir.

varbinary(n): “n” değeri 1 ile 8000 arasında olabilir. “binary(n)”den farkı, kaydedilen dosya boyutuna göre boyutunun değişebilmesidir.

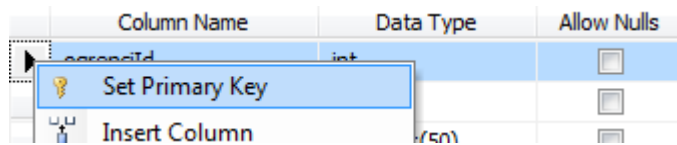
varbinary(max): Maksimum 2 GB’ye kadar değişken boyutta binary veri tutmak için kullanılır.

image: varbinary(max) veri tipiyle kullanımı aynıdır. Sadece resim dosyaları değil, diğer tür dosyalarıda hafızada tutabilirler. Gelecekte bu veri tipinden vazgeçilmesi söz konusu olduğundan varbinary(max) tercih edilmesi daha doğrudur.

1.2. Anahtar Sahaları

1.2.1. Birincil Anahtar (primary key)

Bir tabloda, kayıtların birbiriyle karışmasını engellemek için benzersiz değer içeren alanlara verilen bir özelliktir. Birincil anahtar olarak belirtilmiş alanda veri tekrarı olmamalıdır. Bir alana birincil anahtar özelliği atamak için kolon ismi yanında, sağ tuşa tıklayıp açılan menüden **Set Primary Key** seçeneği seçilmelidir.



Resim 1.11: Birincil anahtar özelliği atama

Birden fazla kolona birincil anahtar özelliği atayabiliriz. Bunun için kolon ismi yanında shift tuşuyla birlikte sağ tuşa tıklamalıyız.

Eğerki tablolar arasında ilişki diyagramı oluşturulacaksa ilişkilendirilecek kolonların en az birine, bu özelliğin atanmış olması gerekmektedir.

1.2.2. Yabancı Anahtar (foreign key)

Bir ilişki yapısında birincil anahtarla eşleşen sütun veya sütun gruplarına verilen isimdir. Bir alanın direk yabancı anahtar (foreign key) olarak atanma şekli yoktur.

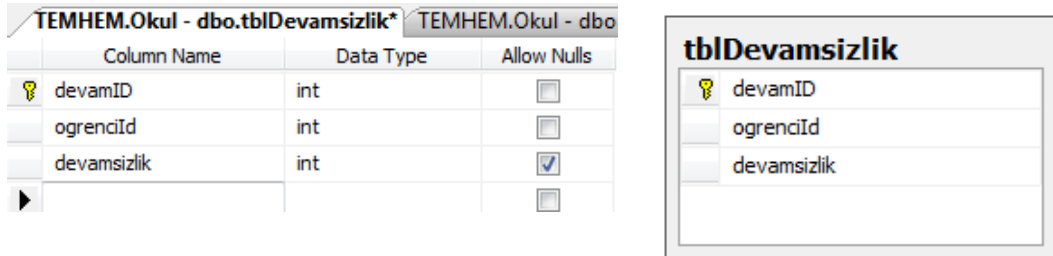
1.3. Tablolar Arası İlişkiler

Hatırlayacağınız gibi bir veri tabanı oluşturulurken çok kolonlu tablolar oluşturmak yerine, verileri gruplandırarak birden fazla tablo şeklinde oluşturulmasının öneminden bahsetmiştik. Bu şekilde oluşturulan bir yapıda, bir birini tamamlayan tablolar arasında bir bağ oluşturulması gerekmektedir. Biz bu bağa ilişki diyoruz.

Tablolar arasında üç tür ilişki vardır.

Bire Bir İlişki: A tablosundaki bir kaydın, B tablosunda sadece bir kayda karşılık geldiği ilişki türüdür.

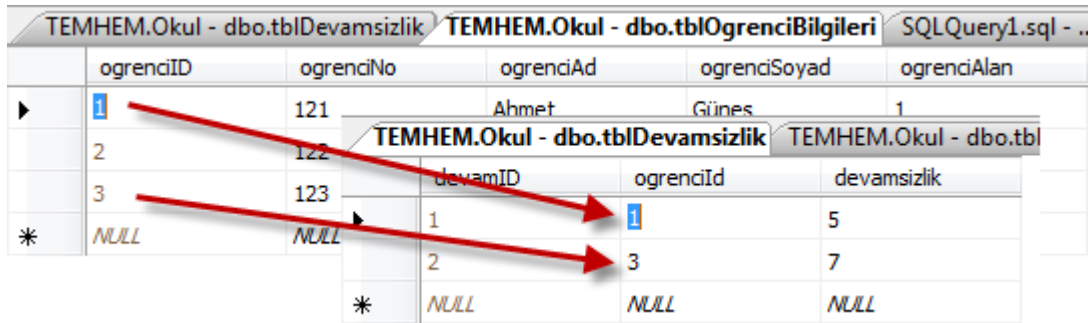
Örnek: Öğrenci devamsızlıklarını tutan bir tablo oluşturalım.



Column Name	Data Type	Allow Nulls
devamID	int	<input type="checkbox"/>
ogrenciId	int	<input type="checkbox"/>
devamsizlik	int	<input checked="" type="checkbox"/>

Resim 1.12: tblDevamsizlik tablosu ayrıntıları

Dikkat ederseniz “ogrenciId” adında bir kolon oluşturduk. Bu kolonla tblOgrenciBilgileri tablosundaki ogrenciID kolonu arasında bir ilişki oluşturacağız.



ogrenciID	ogrenciNo	ogrenciAd	ogrenciSoyad	ogrenciAlan
1	121	Ahmet	Günes	1
2	122			
3	123			
*	NULL	NULL	NULL	NULL

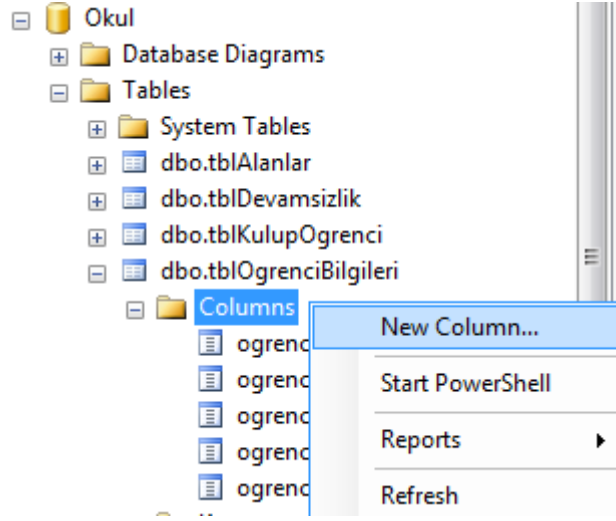
devamID	ogrenciId	devamsizlik
1	1	5
2	3	7
*	NULL	NULL

Resim 1.13: tblDevamsizlik ve tblOgrenciBilgileri tablosu arasındaki ilişki

tblOgrenciBilgileri tablosundaki ogrenciID kolonu tblDevamsizlik tablosundaki ogrenciId kolonunda bir kayda karşılık geldiğinden burada bire bir ilişki söz konusudur.

Bire Çok İlişki: A tablosundaki bir kaydın, B tablosunda birden fazla kayda karşılık geldiği ilişki türüdür.

Örnek: tblOgrenciBilgileri tablomuzda, tblVeliBilgileri tablosunu bağlamak için bir kolon oluşturalım. Bu işlem için tblOgrenciBilgileri düğümü altındaki Columns düğümüne sağ tuşa tıklayalım ve açılan menüden New Column seçeneğini seçelim.



Resim 1.14: Object Explorer penceresi tabloya yeni kolon ekleme

Column Name	Data Type	Allow Nulls
ogrenciID	int	<input type="checkbox"/>
ogrenciNo	int	<input type="checkbox"/>
ogrenciAd	nvarchar(50)	<input type="checkbox"/>
ogrenciSoyad	nvarchar(50)	<input type="checkbox"/>
ogrenciAlan	int	<input type="checkbox"/>
veliId	int	<input type="checkbox"/>

Resim 1.15: tblOgrenciBilgileri tablosunun son hali

veliId adında, int veri tipinde ve her öğrencinin bir velisi bulunmak zorunda olduğu için bu alana veri girişini zorlamak için Allow Nulls kutucuğunu işaretlemeyen yeni sütunumuzu oluşturuyoruz. Kaydet butonuna basarak değişikliği kaydediyoruz.

Not: Tabloya daha önceden veri girişi yapılmışsa Allow Nulls kutucuğunu işaretlemediğimizden dolayı kayıt esnasında hata verecektir. Çünkü daha önceden kayıt eklenmiş ve yeni bir sütun oluşturuluyorsa default değer olarak daha önceki kayıtlar için bu sütuna Null değeri atanmaktadır.

Bu hatayı engellemek için Column Properties(sütun özellikleri) penceresinde General sekmesi altında yer alan Default Value or Binding değerine 0 yazıyoruz. Dolayısıyla default değer olarak null değil 0 kullanılacağından hata ortadan kalkacaktır.

TEMHEM.Okul - dbo.tblOgrenciBilgileri			
id	ogrenciSoyad	ogrenciAlan	veliId
	Güneş	1	1
	Yıldız	1	2
	Güneş	2	1
	NULL	NULL	NULL

TEMHEM.Okul - dbo.tblVeliBilgileri			
veliID	veliAd	veliSoyad	
1	Ali	Güneş	
2	Salih	Yıldız	
*	NULL	NULL	NULL

Resim 1.16: tblOgrenciBilgileri ile tblVeliBilgileri tabloları arasındaki ilişki

tblVeliBilgileri tablosundaki veliID kolonu, tblOgrenciBilgileri tablosundaki veliId kolonunda birden fazla kayda karşılık geldiğinden burada birden çoğa ilişki söz konusudur.

Bir okulda bir öğrencinin bir tane veli kaydı varken bir velinin birden fazla öğrencisi olabilir.

Çoğa Çok İlişki: A tablosundaki bir kaydın, B tablosunda birden çok kayda karşılık geldiği aynı zamanda B tablosundaki bir kaydın, A tablosunda birden çok kayda karşılık geldiği ilişki türüdür. Bu durumlarda genellikle ilişkiyi kuran üçüncü bir tablo kullanılır.

Örnek: Okulda bulunan sosyal kulüp bilgilerini tutan bir tablo oluşturalım.

TEMHEM.Okul - dbo.tblKulupOgrenci		
Column Name	Data Type	Allow Nulls
kulupID	int	<input type="checkbox"/>
kulupAd	nvarchar(50)	<input type="checkbox"/>
		<input type="checkbox"/>

tblSosyalKulupler	
kulupID	
kulupAd	

Resim 1.17: tblSosyalKulupler tablosu ayrıntıları

Senaryomuz şöyle olsun: Okulda bir öğrenci istediği kadar kulübe üye olabilsin. Dolayısıyla bir öğrencinin daha önceden kaç kulübe üye olabileceğini bilmediğimizden tblOgrenciBilgileri tablomuzda kulüp için açılacak sütun sayısı belirsizdir. İşte burada iki tablo arasındaki ilişkiyi tutmak için üçüncü bir tablo oluşturuyoruz.

Column Name	Data Type	Allow Nulls
koID	int	<input type="checkbox"/>
kulupId	int	<input type="checkbox"/>
ogrenciId	int	<input type="checkbox"/>

Resim 1.18: tblKulupOgrenci tablosu ayrıntıları

Öğrencileri kulüplere dahil ediyoruz.

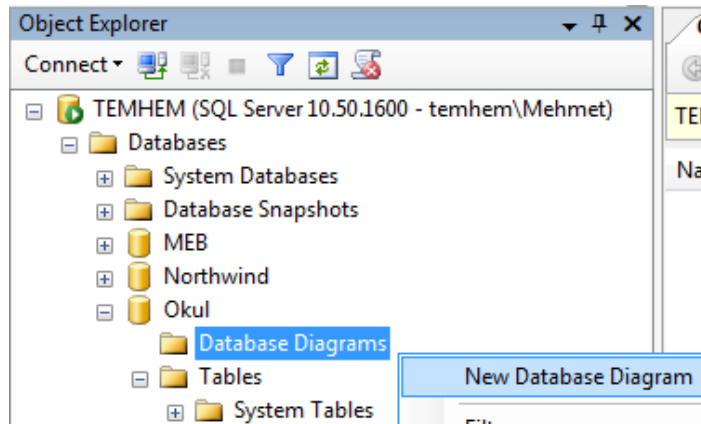
	koID	kulupId	ogrenciId
	1	1	1
	2	2	1
	3	1	2
	4	3	3
	5	3	2
*	NULL	NULL	NULL

Resim 1.19: tblKulupOgrenci tablosunun veri girilmiş hali

kulupId değeri 1 olan kulübe ogrenciId değeri 1 ve 2 olan öğrenciler üyeyken ogrenciId değeri 2 olan öğrenci kulupId değeri 1 ve 3 olan kulüplere üyedir. Yani bir öğrenci birden fazla kulübe üye olabiliyorken bir kulübe de birden fazla öğrenci üye olabilmektedir. Dolayısıyla burada çoktan çoğa bir ilişki söz konusudur.

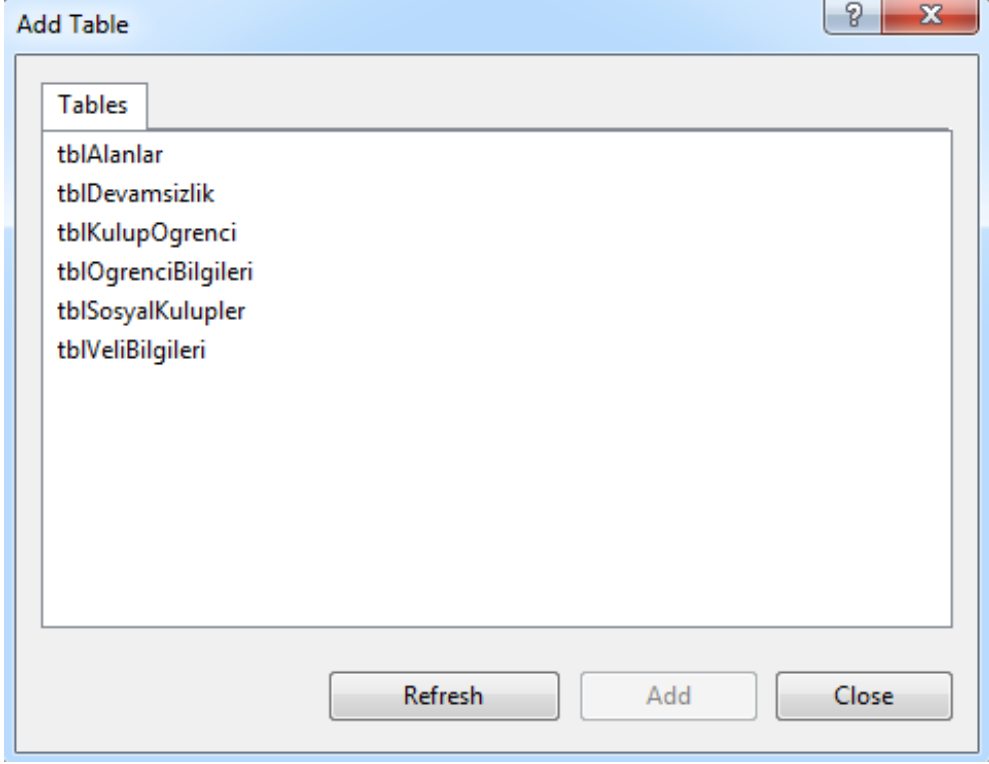
1.4. İlişki Diyagramı Oluşturma

Tablolar arasında bir ilişki diyagramı oluşturmak için Database Diagrams düğümü üzerinde sağ tuşa tıklıyoruz ve açılan menüden New Database Diagram seçeneğini seçiyoruz.



Resim 1.20: Object Explorer penceresi yeni diyagram ekleme

Karşımıza diyagrama eklemek için veritabanımızda bulunan tabloların listesi olan Add Table penceresi gelecektir.



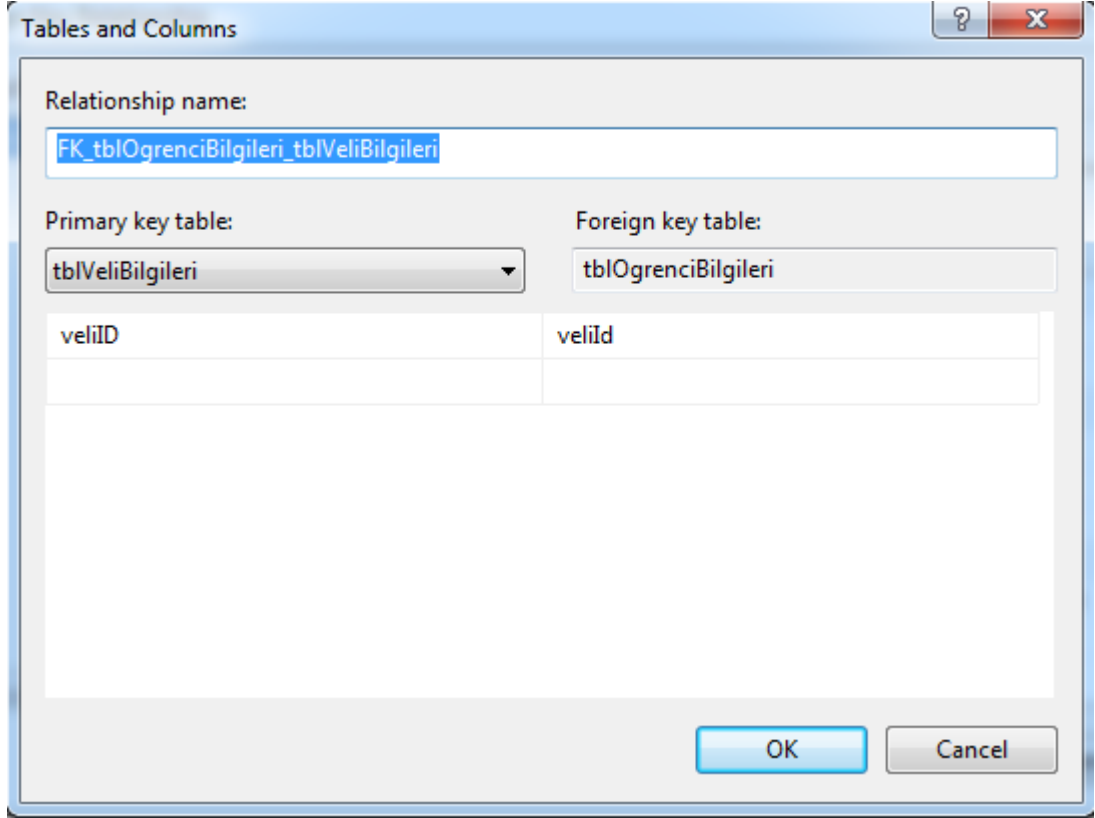
Resim 1.21: Veritabanında tanımlı tabloların listesi

Buradan hangi tabloları diyagrama ekleyeceksek seçip Add butonuna basıyoruz. Biz tüm tabloları seçip Add butonuna basıyoruz. Daha sonra pencereyi kapatmak için Close butonunu kullanıyoruz.

Tabloların başlık bilgisi üzerinden tutup sürükleyerek diyagram üzerinde hareket ettirebiliriz.

İlişki oluşturacağımız kolonlardan birinin üstünde sol tuşa tıklayıp bırakmadan diğer kolon üzerinde faremizi sürükleyip bırakıyoruz.

Karşımıza ilişki durumunu gösteren bir pencere çıkacaktır.



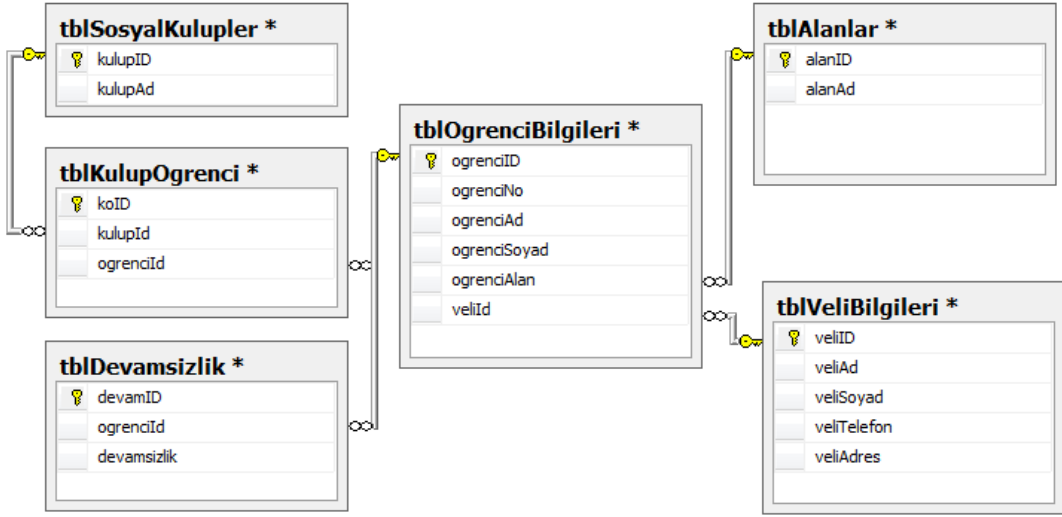
Resim 1.22: Tablolar arasında ilişki belirleme penceresi

Relationship name: İlişkiye verilen addır. İsteğe göre değiştirilebilir.

Primary key table: Birincil anahtarın hangi tabloda ve hangi kolon olduğu belirtilir.

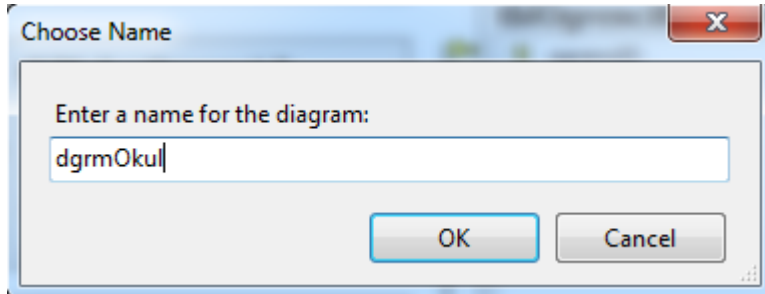
Foreign key table: Yabancı anahtarın hangi tabloda ve hangi kolon olduğu belirtilir.

OK butonuna basarak ilişkiyi oluşturuyoruz. Bu şekilde tüm ilişkileri oluşturuyoruz.



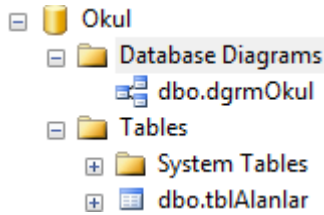
Resim 1.23: İlişki diyagramı

Tüm ilişkileri oluşturduktan sonra kaydet butonuna basıyoruz. Diyagrama bir isim atayıp OK butonuna basıyoruz.



Resim 1.24: Diyagramı isimlendirme penceresi

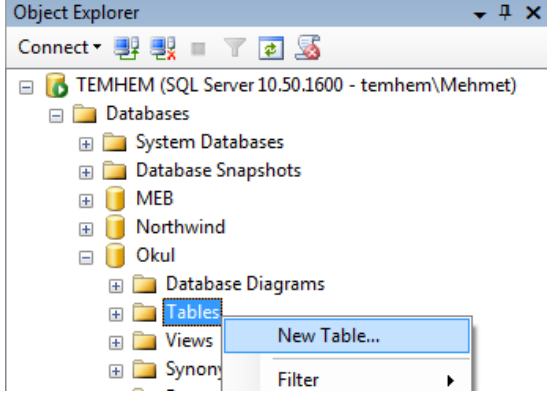
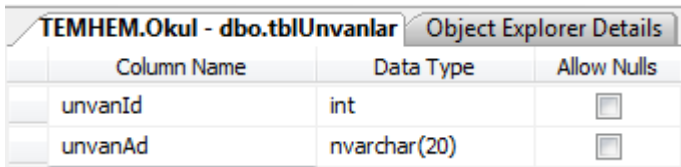
Oluşturduğumuz diyagram artık Database Diagrams düğümü altında listeleniyor olacaktır.

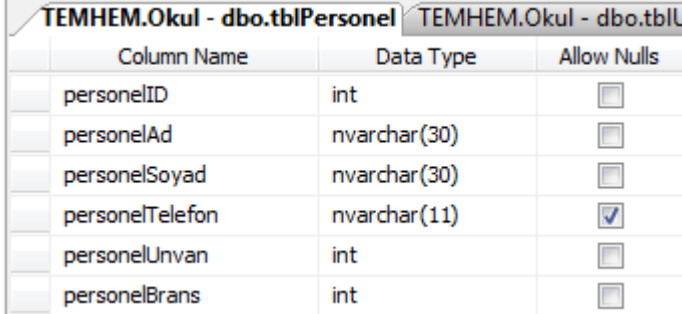


Resim 1.25: Object Explorer penceresi diyagram listesi

UYGULAMA FAALİYETİ

Daha önce oluşturduğumuz Okul veritabanı üzerinde okuldaki öğretmenlerin bilgilerini tutan tblPersonel ve unvanların listesini tutan tblUnvanlar adında tabloları oluşturalım ve bu iki tablo arasındaki ilişkiyi oluşturunuz.

İşlem Basamakları	Açıklama
<ul style="list-style-type: none"> Okul veritabanı düğümünü açalım, Tables düğümü üzerinde sağ tuşa tıklayalım ve açılan menüden New Table seçeneğini seçelim. 	
<ul style="list-style-type: none"> Öncelikle unvan listesini tutacak olan tablomuzu oluşturalım. 	
<ul style="list-style-type: none"> Unvan tablosu ile personel tablosu arasında ilişki kuracağımızdan unvan tablosunda birincil anahtar olacak bir alan tanımlamalıyız. 	<ul style="list-style-type: none"> Adı unvanID ve veri türünde int olsun. Her kayıt eklediğimizde bu alandaki değerın otomatik artması için Identity özelliğini yes yapalım.
<ul style="list-style-type: none"> Unvan isimlerini tutacak bir alan tanımlayalım. 	<ul style="list-style-type: none"> Adı unvanAd ve veri türü nvarchar(20) olsun.
<ul style="list-style-type: none"> Tablomuzu tblUnvanlar adında kaydedelim. 	
<ul style="list-style-type: none"> Personel bilgilerini tutacak tablomuzu oluşturmaya başlayalım. 	<ul style="list-style-type: none"> Birinci adımda yaptıklarımızı tekrarlıyoruz.
<ul style="list-style-type: none"> Aynı ad ve soyada sahip personelin karışmaması için benzersiz veri tutacak bir alan tanımlayalım. 	<ul style="list-style-type: none"> Adı personelID ve veri türünde int olsun. Her kayıt eklediğimizde bu alandaki değerın otomatik artması için Identity özelliğini yes yapalım.

<p>➤ Personelin ad, soyad, telefon numarası, unvan ve branş bilgilerini tutacak alanları tanımlayalım.</p>	<p>➤ Personel adı için alan adımız personelAd ve veri türü nvarchar(30) olsun.</p> <p>➤ Personel soyadı için alan adımız personelSoyad ve veri türü nvarchar(30) olsun.</p> <p>➤ Personel telefon numarası için alan adımız personelTelefon ve veri türü üzerinde sayısal işlem yapmayacağımız için nvarchar(11) olsun.</p> <p>➤ Personel unvanı için alan adımız personelUnvan ve veri türünde tblUnvanlar tablosunda unvanID alanıyla ilişkilendireceğimizden aynı tür olması için int olsun.</p> <p>➤ Personel branşı için alan adımız personelBrans ve veri türünde daha önceden oluşturduğumuz tblAlanlar tablosunda alanID alanıyla ilişkilendireceğimizden aynı tür olması için int olsun.</p>																					
<p>➤ Tablomuzu tblPersonel adında kaydedelim.</p>	 <table border="1"> <thead> <tr> <th>Column Name</th> <th>Data Type</th> <th>Allow Nulls</th> </tr> </thead> <tbody> <tr> <td>personelID</td> <td>int</td> <td><input type="checkbox"/></td> </tr> <tr> <td>personelAd</td> <td>nvarchar(30)</td> <td><input type="checkbox"/></td> </tr> <tr> <td>personelSoyad</td> <td>nvarchar(30)</td> <td><input type="checkbox"/></td> </tr> <tr> <td>personelTelefon</td> <td>nvarchar(11)</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>personelUnvan</td> <td>int</td> <td><input type="checkbox"/></td> </tr> <tr> <td>personelBrans</td> <td>int</td> <td><input type="checkbox"/></td> </tr> </tbody> </table>	Column Name	Data Type	Allow Nulls	personelID	int	<input type="checkbox"/>	personelAd	nvarchar(30)	<input type="checkbox"/>	personelSoyad	nvarchar(30)	<input type="checkbox"/>	personelTelefon	nvarchar(11)	<input checked="" type="checkbox"/>	personelUnvan	int	<input type="checkbox"/>	personelBrans	int	<input type="checkbox"/>
Column Name	Data Type	Allow Nulls																				
personelID	int	<input type="checkbox"/>																				
personelAd	nvarchar(30)	<input type="checkbox"/>																				
personelSoyad	nvarchar(30)	<input type="checkbox"/>																				
personelTelefon	nvarchar(11)	<input checked="" type="checkbox"/>																				
personelUnvan	int	<input type="checkbox"/>																				
personelBrans	int	<input type="checkbox"/>																				
<p>➤ Şimdi oluşturduğumuz bu iki tablo arasındaki ilişikiyi oluşturalım.</p>																						
<p>➤ Daha önce oluşturduğumuz dgrmOkul diyagramını açalım.</p>																						
<p>➤ Bu diyagrama yeni oluşturduğumuz tblUnvanlar ve tblPersonel tablolarını ekleyelim.</p>	<p>➤ Diyagramın boş bir alanında sağ tuşa tıklayıp açılan menüden Add Table seçeneğini seçelim. Açılan pencereden ekleyeceğimiz tabloları seçerek Add butonuna basalım.</p>																					
<p>➤ tblUnvanlar tablosundaki unvanID alanında faremizin sol tuşuna basılı tutup bırakmadan tblPersonel tablosundaki personelUnvan alanına sürükleyelim.</p>																						

<p>➤ tblAlanlar tablosundaki alanID alanında faremizin sol tuşuna basılı tutup bırakmadan tblPersonel tablosundaki personelBrans alanına sürükleyelim.</p>	
<p>➤ Diyagramımızın son halini kaydedelim.</p>	

KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri		Evet	Hayır
1	Yeni bir tablo oluşturmaya başlayabildiniz mi?		
2	Tablo içinde tutulabilecek verileri belirleyebildiniz mi?		
3	Tutucak veriler için veri türlerini belirleyebildiniz mi?		
4	Oluşturduğunuz tabloları kaydedebildiniz mi?		
5	Oluşturduğumuz tabloları diyagrama ekleyebildiniz mi?		
6	Tablolar arasındaki ilişkiyi oluşturabildiniz mi?		

DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Sütuna veri girişini zorlamak için sütunun hangi özelliği kullanılır?
A) Column Name B) Data Type C) Allow Nulls D) Enabled
2. Bir tablo oluşturmak için kullanılan Sql komutu hangisidir?
A) CREATE TABLE B) ALTER TABLE
C) USER TABLE D) NEW TABLE
3. Aşağıdakilerden hangisi yaklaşık sayısal veri tiplerindedir?
A) bit B) int C) decimal D) float
4. 0-255 arasında bir değer tutabilen kesin sayısal veri tipi aşağıdakilerden hangisidir?
A) bit B) tinyint C) smallint D) int
5. Dünya ülkeleri arasındaki saat farkını da tutabilen tarih ve zaman veri tipi hangisidir?
A) datetime2 B) time C) date D) datetimeoffset
6. A tablosundaki bir kaydın B tablosundan birden fazla kayda karşılık geldiği ilişki türü hangisidir?
A) Bire Bir B) Bire Çok C) Çoğa Çok D) Hiçbiri

Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

7. Bir sütunun otomatik sayı gibi davranması için özelliği true yapılmalıdır.
8. bir ilişki yapısında birincil anahtarla eşleşen sütun veya sütun gruplarına verilen addır.
9. Yazdığımız sql cümlesinde hata olup olmadığını denemek için butonuna basmalıyız.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Bu öğrenme faaliyetinde, veritabanı aktarma işlemlerini yapabileceksiniz.

ARAŞTIRMA

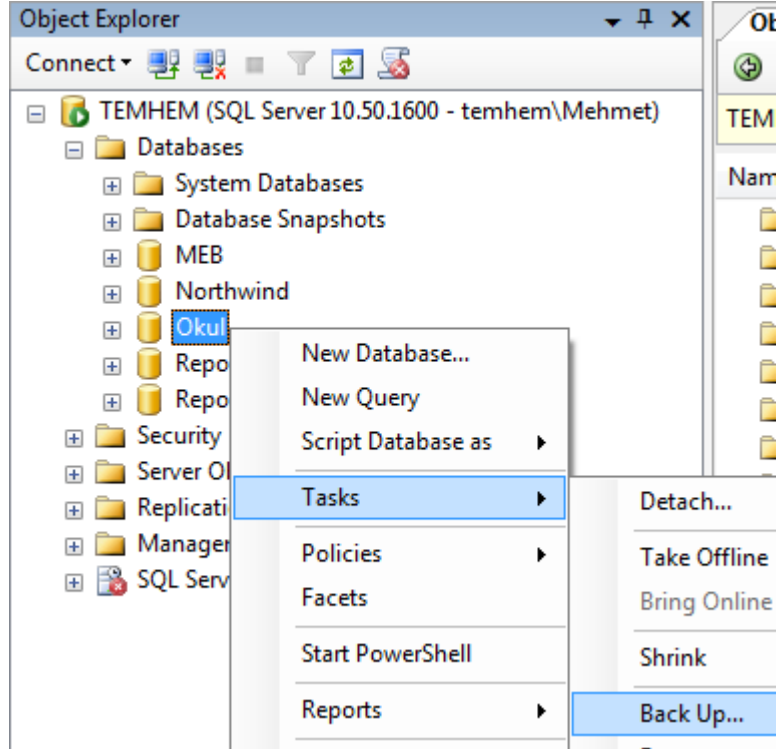
- Verilerin neden yedeklenmesi gerektiğini araştırınız.

2. VERİ TABANI İŞLEMLERİ

2.1. Veri Tabanı Yedeğini Alma

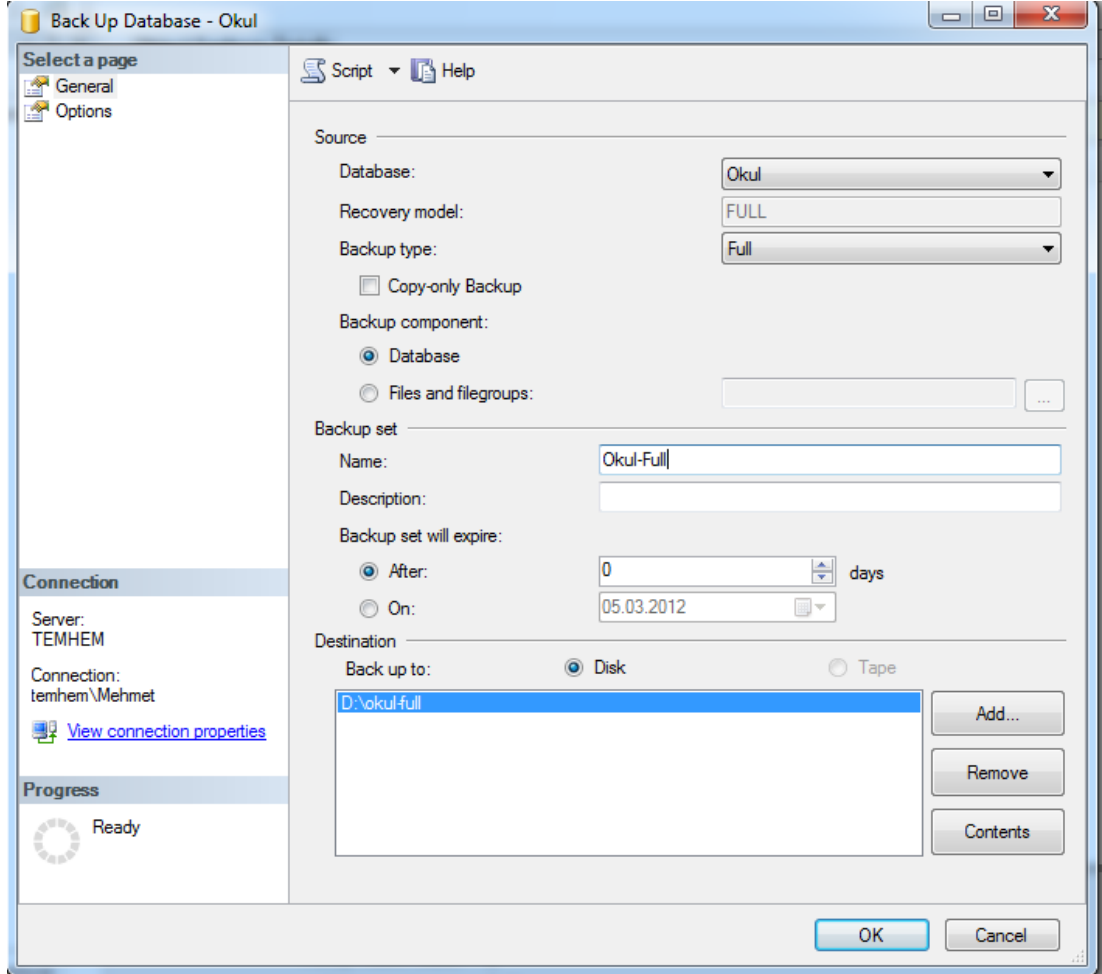
Bilgiler hiç beklenmedik bir zamanda silinebilir, bozulabilir veya doğal felaketler sonucunda ortadan tamamıyla kaybolabilir. Sistem yöneticisi olarak veritabanlarını belirli zaman aralıklarında yedeklemek ve yedeklerin güvenilir bir yerde saklamak çok önemlidir.

Veritabanının yedeğini almak için öncelikle hangi veritabanının yedeğini alacaksak üzerinde sağ tuşa tıklıyoruz. Açılan menüden **Tasks**→**Back Up..** seçeneğini seçiyoruz.



Resim 2.1: Object explorer penceresi yedek alma işlemi

Karşımıza Back Up Database penceresi gelecektir.



Resim 2.2: Yedek alma penceresi

Source bölümünde:

Database kısmında yedeği alınacak veritabanı seçilir. Hangi veritabanı üzerinde işleme başladysak burada seçili olarak gelir.

Backup type kısmında, ne tür bir yedek alacağımızı belirleriz. Full, Differential ve Transaction Log olmak üzere üç seçenek vardır. Full seçeneğinde, veri tabanının tüm bileşenleriyle birlikte yedeğinin alınması sağlanır. Differential seçilirse alınan en son full yedekten sonra yapılan değişikliklerin yedeği alınır. Transaction Log seçeneğinde ise veritabanının log dosyasının yedeği alınır. Differential ve Transaction Log seçenekleri için daha önceden en az bir kere full yedek alınmış olması gerekmektedir.

Backup set bölümünde:

Name kısmına yedek için bir isim tanımlanır. Backup type seçeneğine bakılarak otomatik bir isim atanır. Değişiklik yapabilirsiniz.

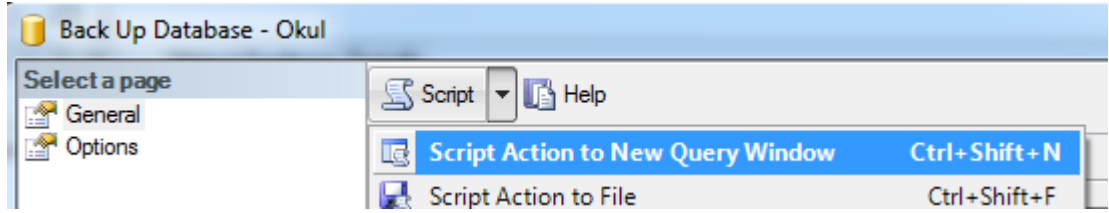
Description kısmına, yedekle ilgili bir açıklama yazabilirsiniz.

Backup set will expire kısmında, yedeğin ne zamana kadar korunacağı belirlenir. Korunduğu süre boyunca üzerine herhangi bir veri yazılmaz. After seçeneğinde girilen gün sayısı boyunca korunur. 0-99999 arasında bir değer girilebilir. Eğerki sıfır girilmiş ise bir süresinin olmadığı anlamına gelir ve üzerine veri yazılmaz. On seçeneğindeyse bir tarih seçilerek yedeğin o tarihe kadar korunması sağlanır.

Destination bölümünde:

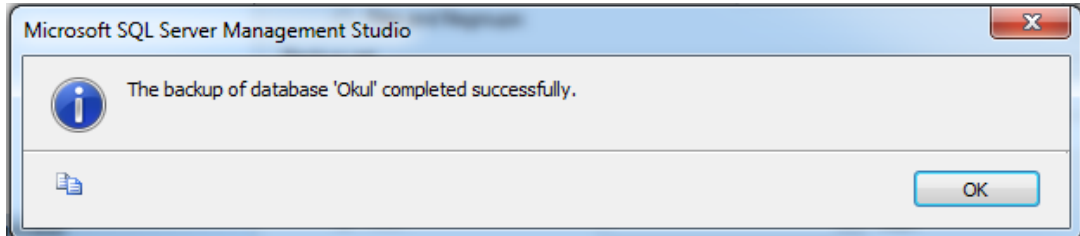
Yedeğin disk üzerinde nerede saklanacağını belirlediği bölümdür. Yedeğin disk üzerinde yeri değiştirilmek istenirse Add butonu kullanılarak yeni yer tayin edilebilir.

Yedek alma işleminden sonra SQL sorugusu oluşturulmasını istiyorsak Script menüsünden, Script Action to New Query Window seçeneği seçilmelidir.



Resim 2.3: Yedek alma penceresi script menüsü

Tüm bu işlemleri yaptıktan sonra, Ok butonuna basarak yedeğin oluşturulmasını sağlıyoruz.



Resim 2.4: İşlemin bittiğini gösteren mesaj penceresi

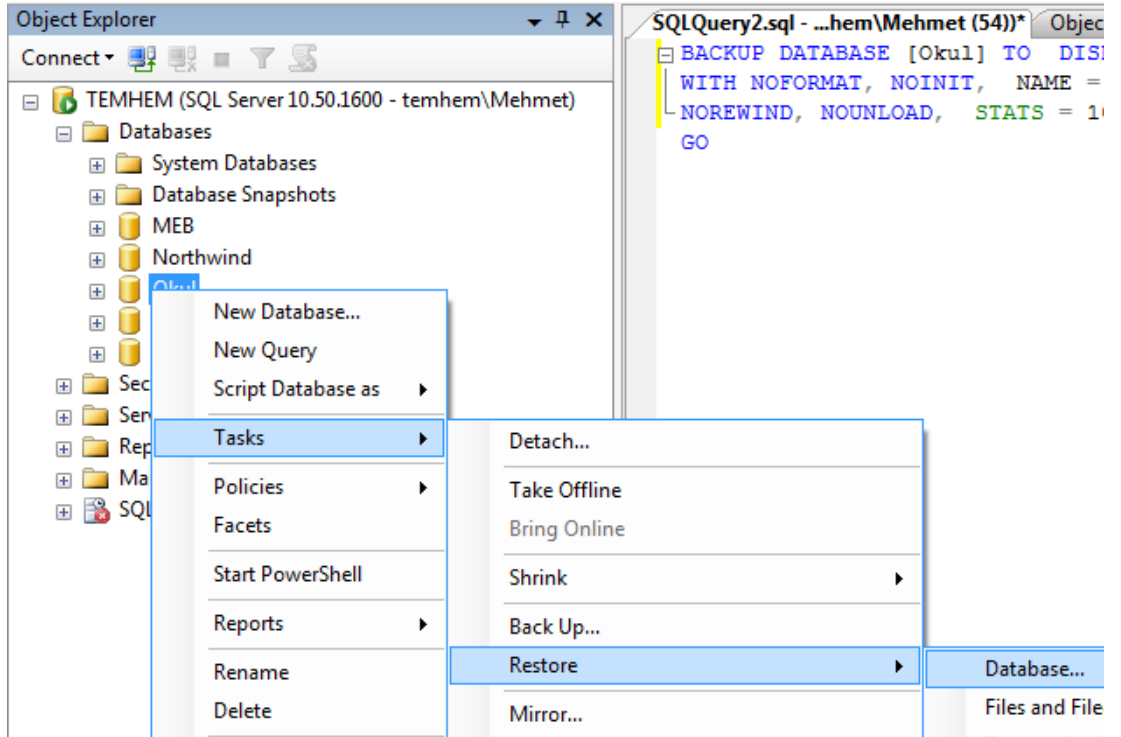
Eğer SQL sorugusu oluşturulmasını istediyseniz yedek alma işlemi bittikten sonra yeni bir sorgu penceresinde, SQL cümlesinin oluşturulduğunu göreceksiniz.

```
SQLQuery2.sql - ...hem\Mehmet (54))* Object Explorer Details
BACKUP DATABASE [Okul] TO DISK = N'D:\okul-full'
WITH NOFORMAT, NOINIT, NAME = N'Okul-Full', SKIP,
NOREWIND, NOUNLOAD, STATS = 10
GO
```

Resim 2.5: Yedek alma sorgu komutları

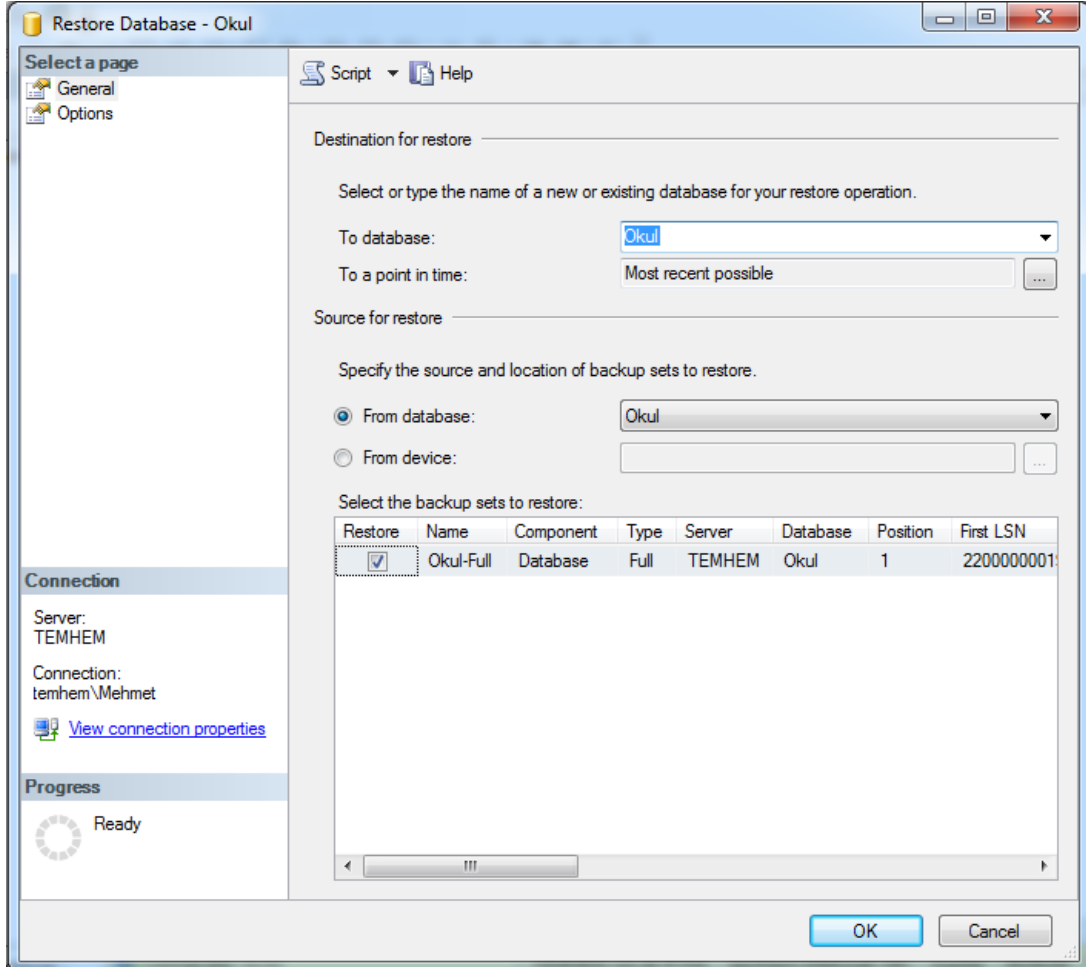
2.2. Yedeği Alınmış Veri Tabanını Geri Yükleme

Mevcut bir veritabanını restore etmek istiyorsak üzerinde sağ tuşa tıklayıp açılan menüden Tasks→Restore→Database seçeneğini seçeriz. Eğerki veritabanı mevcut değilse aynı isimde bir veritabanı oluşturup aynı yolu izleriz.



Resim 2.6: Object explorer penceresi restore işlemi

Karşımıza Restore Database penceresi çıkacaktır.



Resim 2.7: Restore işlemi penceresi

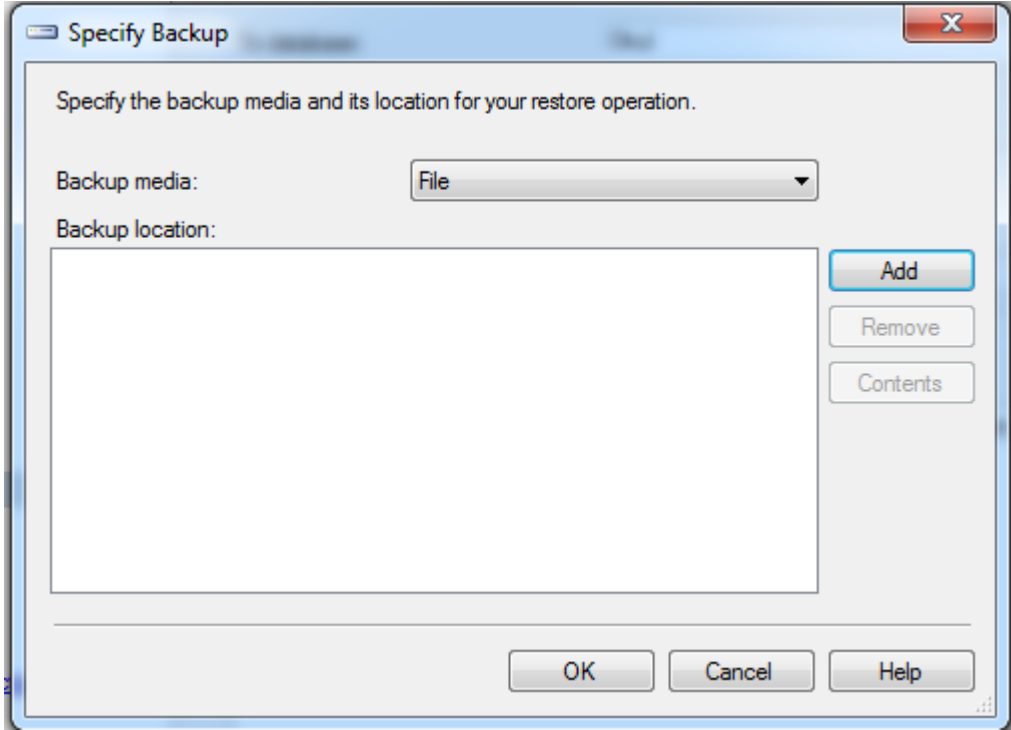
Destination for restore bölümünde:

To database kısmında restore edilecek veritabanı belirlenir. Hangi veritabanı üzerinde işleme başlandıysa o veritabanı seçili olarak gelir.

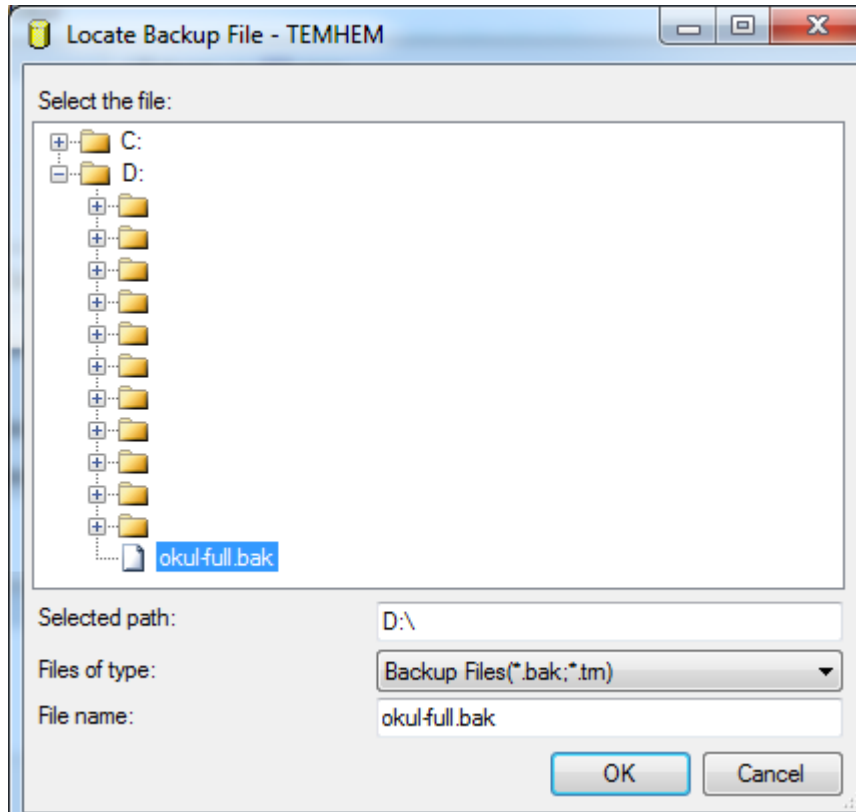
To a point time kısmında restore işlemi için bir tarih ve saat belirleyerek işlemin belirlendiğimiz zamanda gerçekleşmesini sağlayabiliriz.

Source for restore bölümünde:

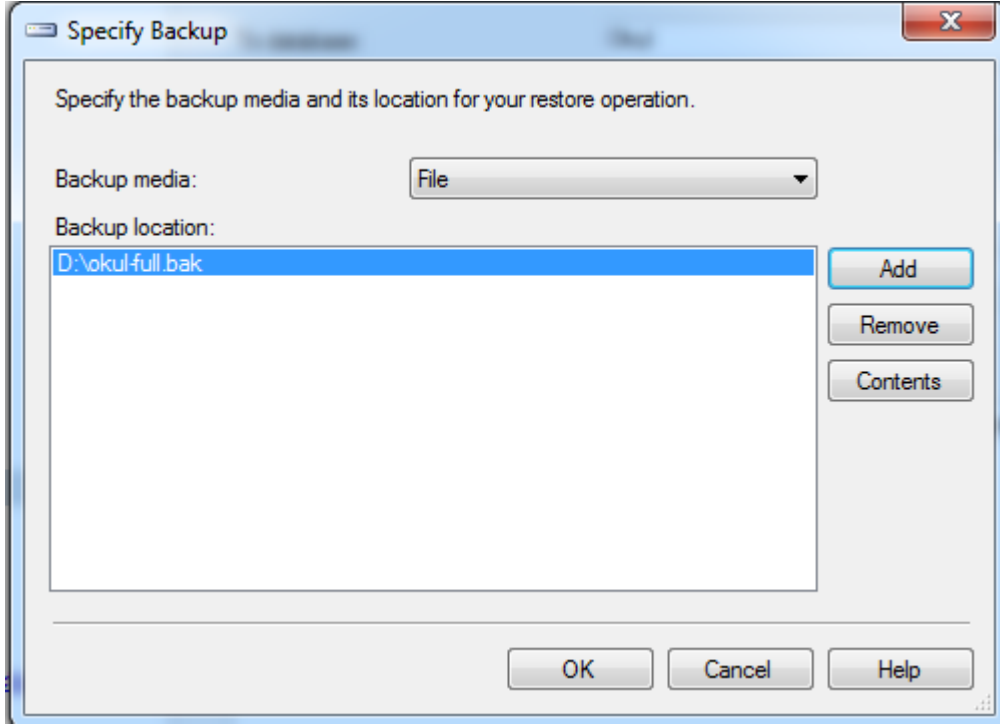
Daha önceden aldığımız yedeği yüklemek için From device seçeneğini seçiyoruz. “...” butonuna basıyoruz. Specify Backup penceresi karşımıza gelecektir. Add butonuna tıklayarak yedeğin disk üzerindeki yerini gösteriyoruz.



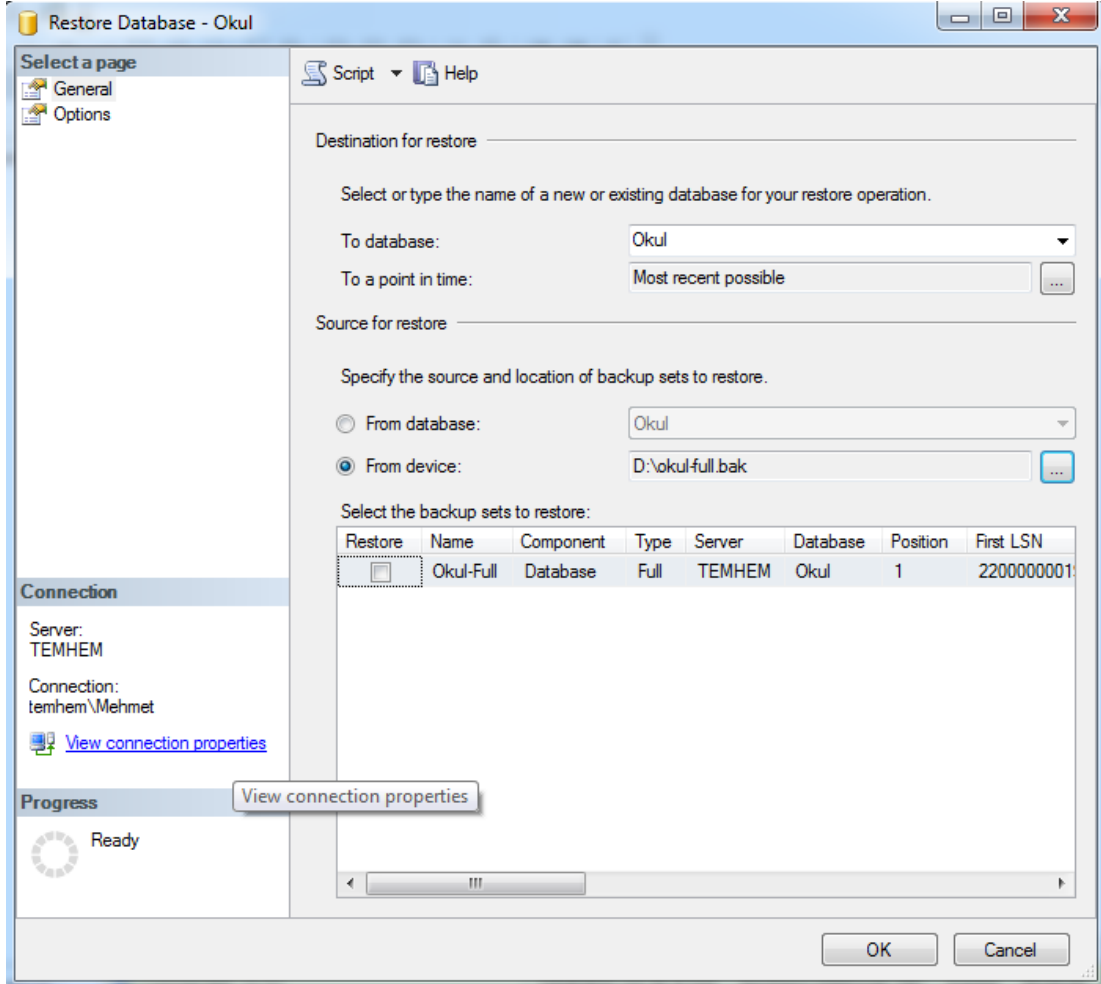
Resim 2.8: Disk üzerindeki yedek dosyalarını ekleme penceresi



Resim 2.9: Windows gezgin penceresi

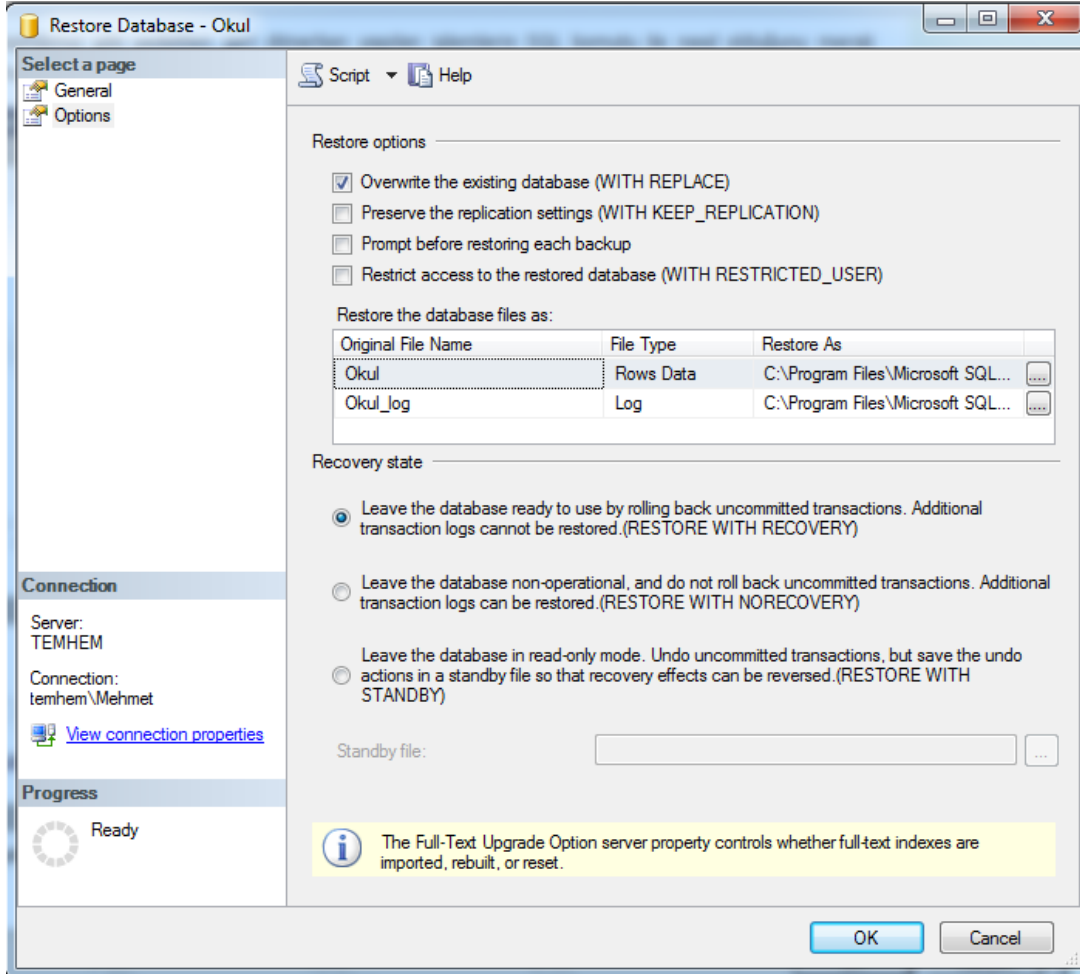


Resim 2.10: Disk üzerindeki yedek dosyalarını ekleme penceresi
OK butonuna basarak işlemimize devam ediyoruz.



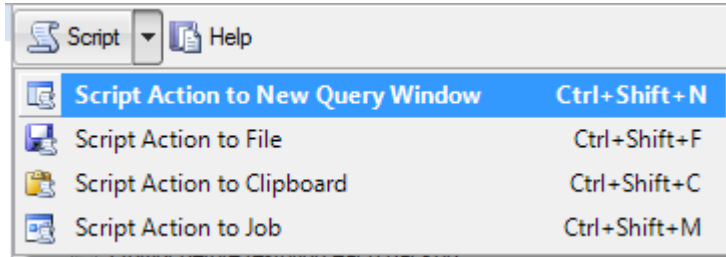
Resim 2.11: Restore işlemi penceresi

Yedeğin mevcut olan veritabanı üzerine yazılması için Options sekmesine geçiyoruz. Restore options seçeneklerinden, Overwrite the existing database seçeneğini işaretliyoruz.



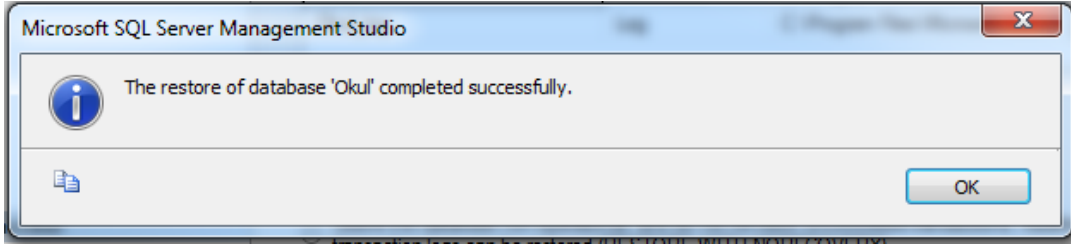
Resim 2.12: Restore işlemi penceresi options sekmesi

Eğerki restore işleminden sonra SQL sorgu cümlesinin oluşmasını istiyor isek Script menüsünden Script Action to New Query Window seçeneğini seçmeliyiz.



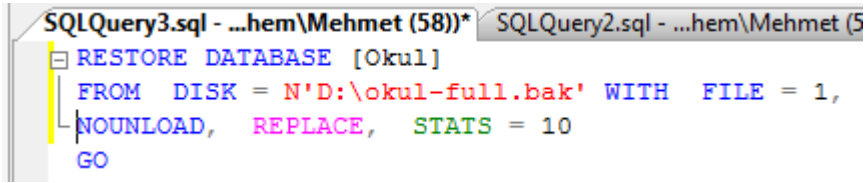
Resim 2.13: Restore işlemi penceresi script menüsü

Tüm bu işlemleri yaptıktan sonra Ok butonuna basarak restore işlemi başlatıyoruz.



Resim 2.14: İşlemin tamamlandığını belirten mesaj penceresi

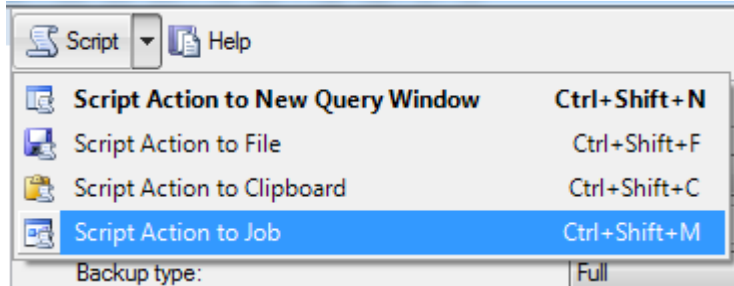
Eğer SQL soruğu oluşturulmasını istediyssek restore işlemi bittikten sonra yeni bir sorgu penceresinde SQL cümlesinin oluşturulduğunu göreceksiniz.



Resim 2.15: Restore işlemi sorgu komutları

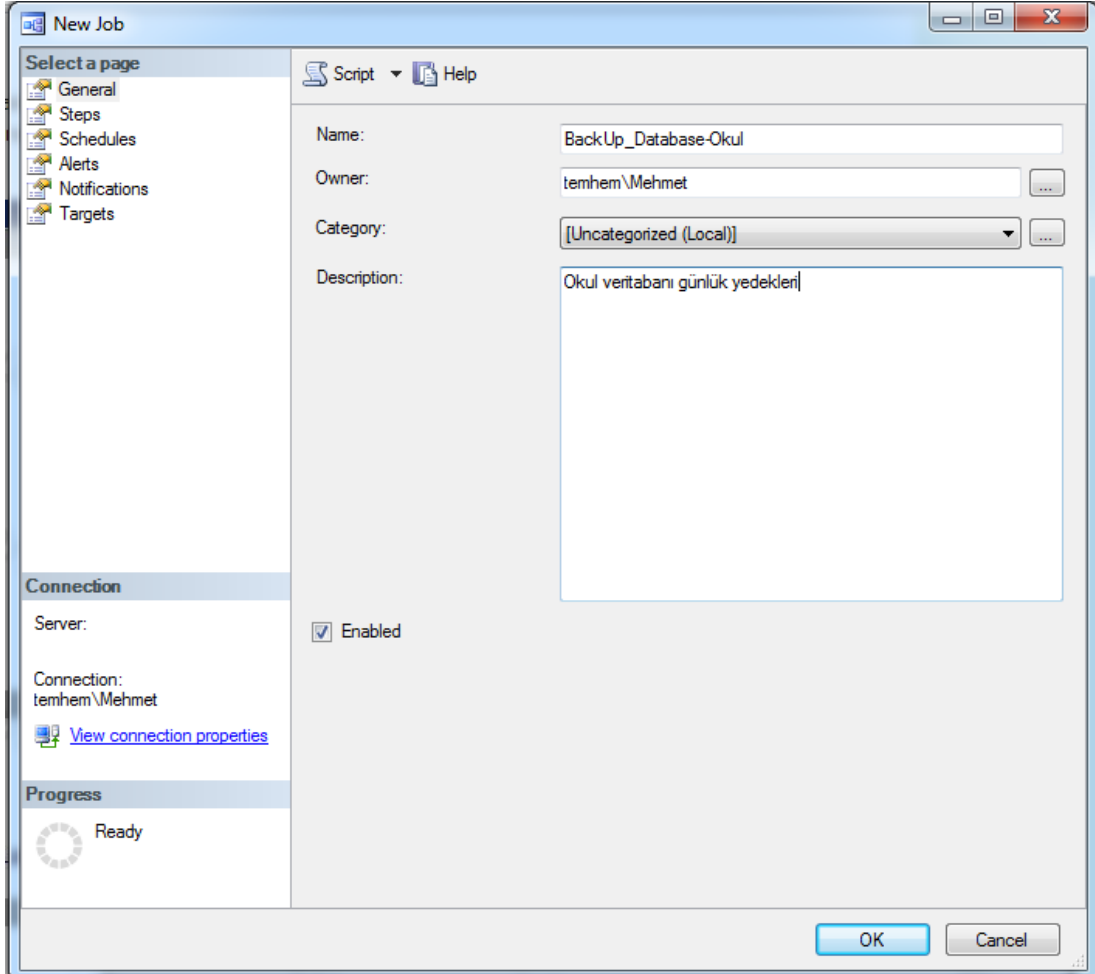
2.3. Otomatik Yedek Alma Görevi Oluşturma

Bir veritabanı yedeğinin belirlenen tarih veya zaman aralıklarında alınmasını istiyorsak bunun için bir görev oluşturmamız gerekmektedir. Yedek alma işlemi sırasında Script menüsü altında bulunan Script Action to Job seçeneğini seçiyoruz.



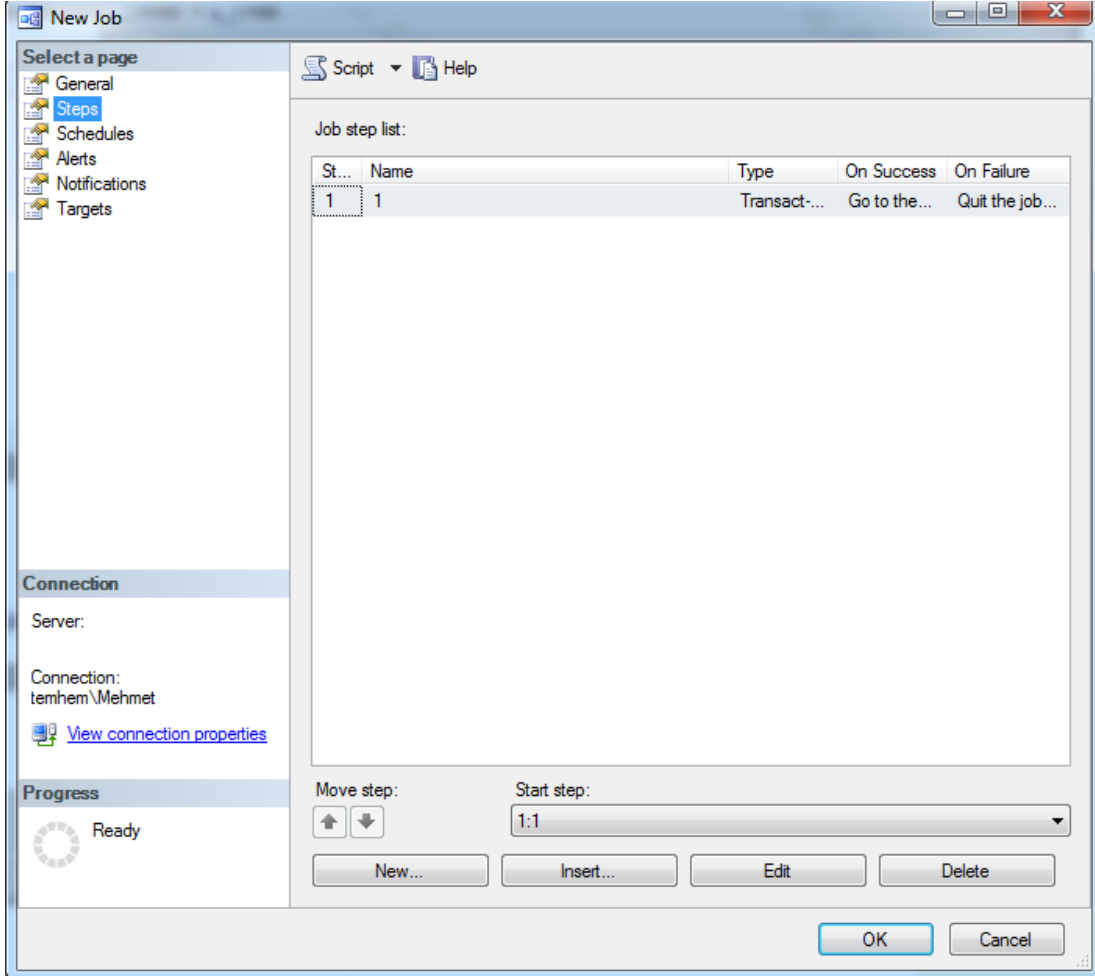
Resim 2.16: Restore işlemi script menüsü

Karşımıza New Job penceresi gelecektir.



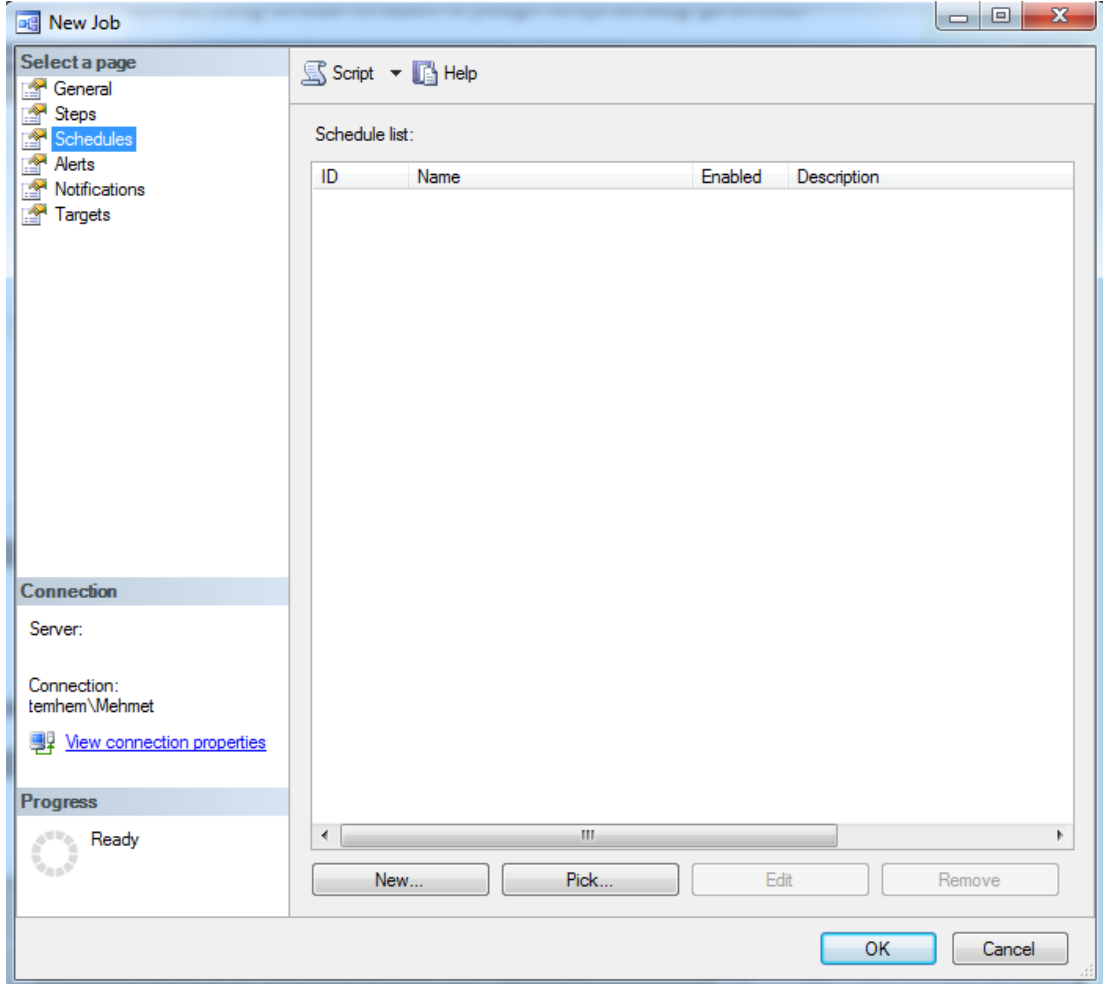
Resim 2.17: Yeni görev oluşturma penceresi general sekmesi

- General sekmesinde görevle ilgili temel özellikler belirlenir.
- Name kısmında, göreve bir isim tanımlıyoruz.
- Owner kısmında, hangi kullanıcının yetkilendirileceği belirlenir.
- Description kısmında, görevle ilgili bir açıklama yazılabilir.
- Buradaki düzenlemeleri yaptıysak Steps sekmesine geçebiliriz.



Resim 2.18: Yeni görev oluřturma penceresi steps sekmesi

Steps sekmesinde, yapılacak iřlemler liste halinde sıralanır.



Resim 2.19: Yeni görev oluřturma penceresi schedules sekmesi

Schedules sekmesi, görev için en önemli bölümdür. Burada görevin gerçekleştirilme ve tekrar edilme ayarları yapılır. Burada New butonuna basıyoruz. Karřımıza New Job Schedule penceresi çıkacaktır.

Resim 2.20: Yeni görev zamanlama penceresi

Name kısmında, zaman aralıkları için bir isim tanımlarız. Schedule type kısmında, görevin hangi durumlarda çalışacağını belirleriz.

Resim 2.21: Zamanlama tipi seçenekleri

Start automatically when SQL Server Agent starts: SQL Server Agent çalıştığında otomatik olarak görevin çalıştırılması isteniyorsa bu seçenek seçilmelidir.

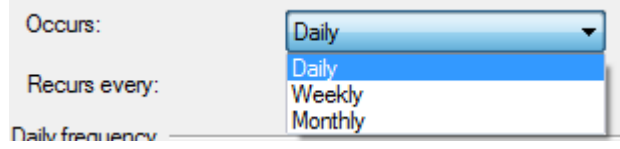
Starts whenever the CPUs become idle: CPU ne zaman boş kalırsa yani CPU'nun yoğun olmadığı dönemlerde görevin çalıştırılmasını istiyorsak bu seçenek seçilmelidir.

Recurring: Belirlediğimiz aralıklarda tekrar eden görevler oluşturmak istiyorsak bu seçeneği seçmeliyiz.

One time: Görevin belirlenen tarih ve saatte bir kez çalıştırılması isteniyorsa bu seçenek seçilmelidir.

Biz Recurring seçeneğini seçerek işleme devam ediyoruz.

Frequency bölümünde, görevin hangi aralıklarda tekrarlanacağı belirlenir.

The image shows a software interface for setting a recurring task. It features a section titled "Frequency" with two main controls: "Occurs:" and "Recurs every:". The "Occurs:" dropdown menu is open, showing three options: "Daily", "Weekly", and "Monthly". The "Recurs every:" field is currently empty.

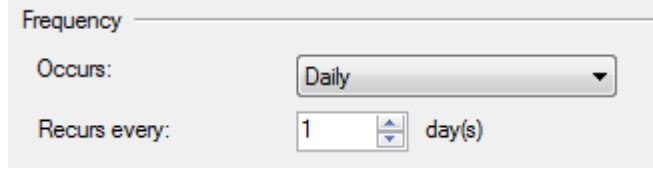
Resim 2.22: Frekans aralığı seçenekleri

Occurs kısmında görevin aralık süresi belirlenir.

Daily: Günlük

Weekly: Haftalık

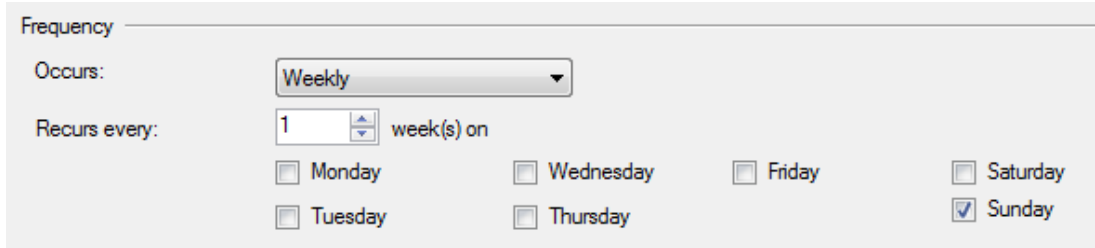
Monthly: Aylık

The image shows the "Frequency" dialog box with "Occurs:" set to "Daily" and "Recurs every:" set to "1" with the unit "day(s)".

Resim 2.23: Frekans aralığı günlük

Tekrarlama süresi Daily seçilirse:

Recurs every kısmında, kaç günde bir yapılacağı belirtilir. Örneğin buraya 3 değerini girersek 3 günde bir tekrarlanacağı anlamına gelir.

The image shows the "Frequency" dialog box with "Occurs:" set to "Weekly" and "Recurs every:" set to "1" with the unit "week(s) on". Below this, there are seven checkboxes for the days of the week: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. The "Sunday" checkbox is checked.

Resim 2.24: Frekans aralığı haftalık

Tekrarlama süresi Weekly seçilirse:

Recurs every kısmında, kaç haftada bir yapılacağı belirtilir. Ek olarak, haftanın hangi günlerinde görevin çalıştırılması isteniyorsa o günler seçilebilir.

Resim 2.25: Frekans aralığı aylık

Tekrarlama süresi Monthly seçilirse:

Day seçeneğinde ilk kutucuğa ayın hangi günü, ikinci kutucukta ise kaç ayda bir görevin çalıştırılacağı belirtilir.

Örneğin; birinci kutucuğa 3, ikinci kutucuğa 2 değerini girdiğimizi farz edelim. Bu şekilde görevimiz 2 ayda bir, ayın 3. günü yürütülecektir anlamına gelir.

The seçeneği seçilirse ilk seçenek kaçınıcı hafta, ikinci seçenek haftanın hangi günü, üçüncü kutucuk ise kaç hafta aralıklarla görevin yürütüleceği belirlenir.4

Örneğin ilk kutucukta second, ikinci kutucukta weekend day, üçüncü kutucuğada 1 değerini girdiğimizi farz edelim. Bu şekilde görevimiz ayda bir kere, ayın 2. haftasının hafta sonları yürütülecektir anlamına gelir.

Resim 2.26: Günlük frekans seçenekleri

Daily frequency bölümünde, görevin hangi saatler arasında yürütüleceği belirlenir.

Occurs once at seçeneğinde, sadece bir kez belirtilen saatte görevin yürütülmesi sağlanır.

Occurs every seçeneğindeyse gün içerisinde kaç saate bir, görevin yürütüleceği belirtilir.

Resim 2.27: Görevin başlangıç-bitiş süre seçenekleri

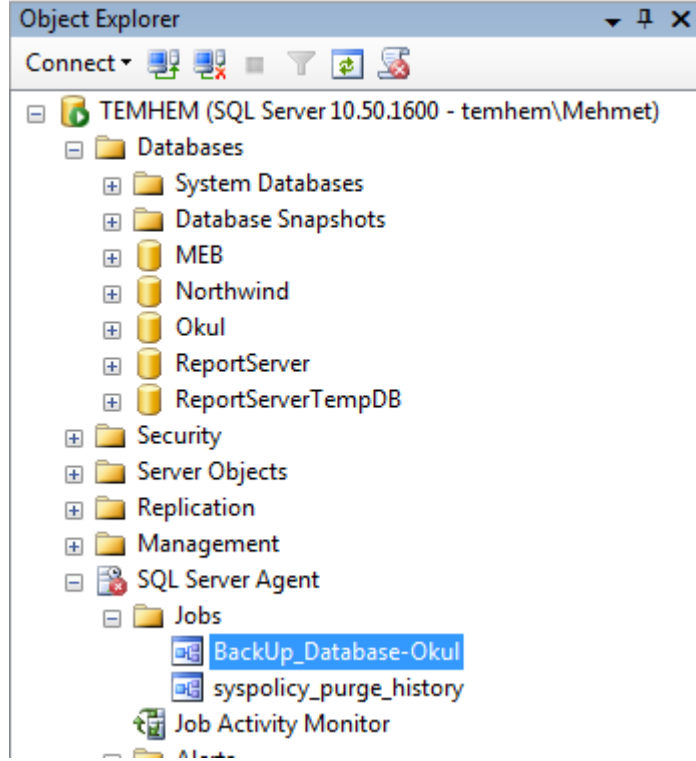
Duration bölümünde ise görevin hangi tarih aralığında yürütüleceği belirlenir.

Start date: Görevin başlama tarihidir.

End date: Görevin bitiş tarihidir.

No end date: Görevin belli bir tarihte sonlanmasını istemiyorsak bu seçenek seçilmelidir.

İsteğimize göre değerleri belirledikten sonra Ok butonuna basarak pencereyi kapatıyoruz. Artık görevimizi oluşturmaya hazırız. New Job penceresinde de OK butonuna basarak görevimizi oluşturuyoruz. Oluşturduğumuz bu görev SQL Server Agent başlığı altındaki Jobs düğümü altında listeleniyor olacaktır.



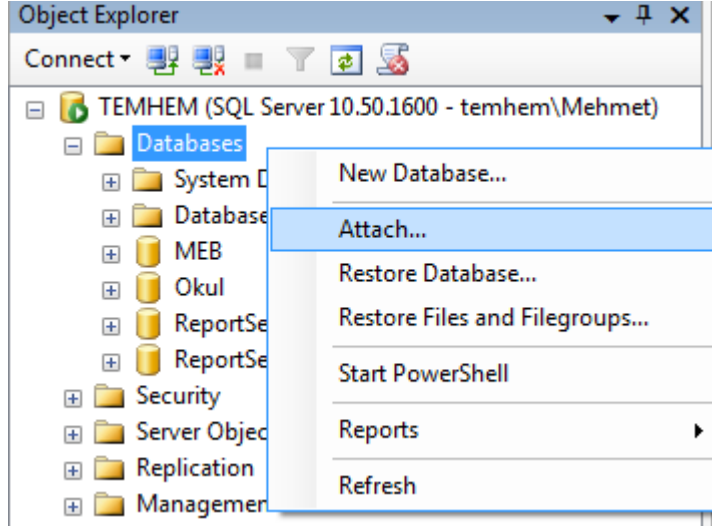
Resim 2.28: Object explorer penceresi görev listesi

2.4. Veri Tabanı Sunucu İşlemleri

2.4.1. Harici Veri Tabanını Sunucuya Dâhil Etme

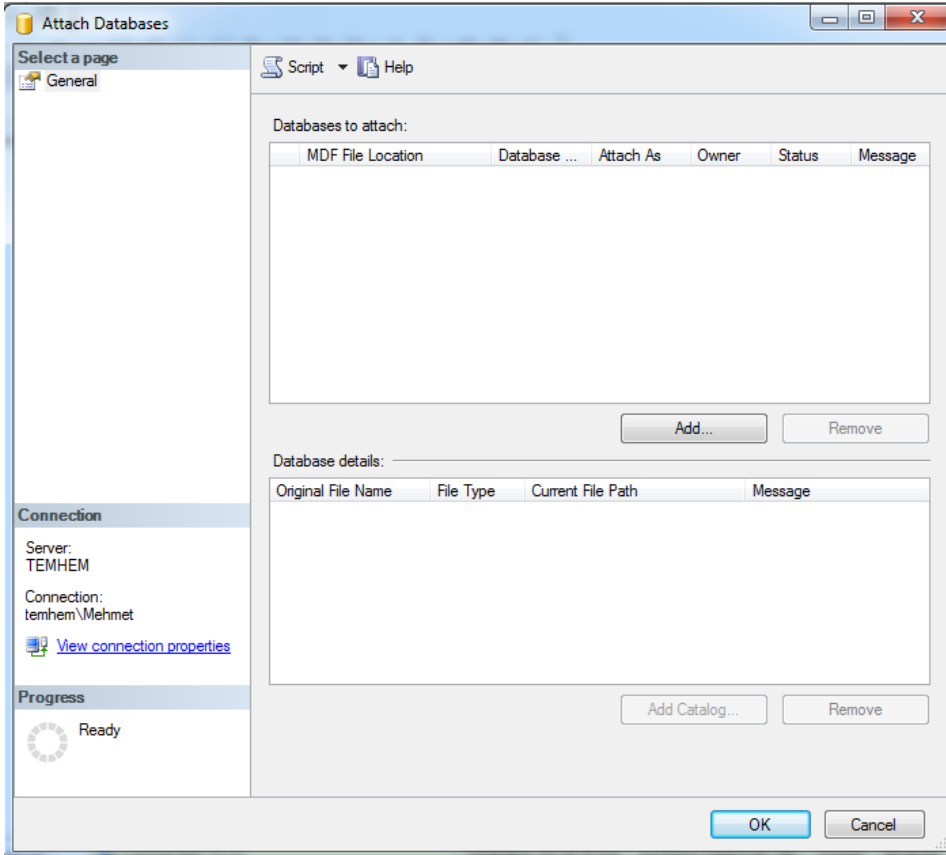
Bir nedenden ötürü ağ veritabanı yazılımımızı kaldırdık ve yeniden kurduk diyelim veya bir başkasının hazırlamış olduğu bir veritabanı dosyası üzerinde işlem yapmak istiyoruz. İşte bu noktada veritabanlarını sunucumuza dâhil etmemiz gerekiyor.

Object Explorer penceresinde, bağlı bulunduğumuz sunucu altında bulunan Databases düğümüne sağ tuşa tıklarız. Açılan menüden Attach seçeneğini seçiyoruz.

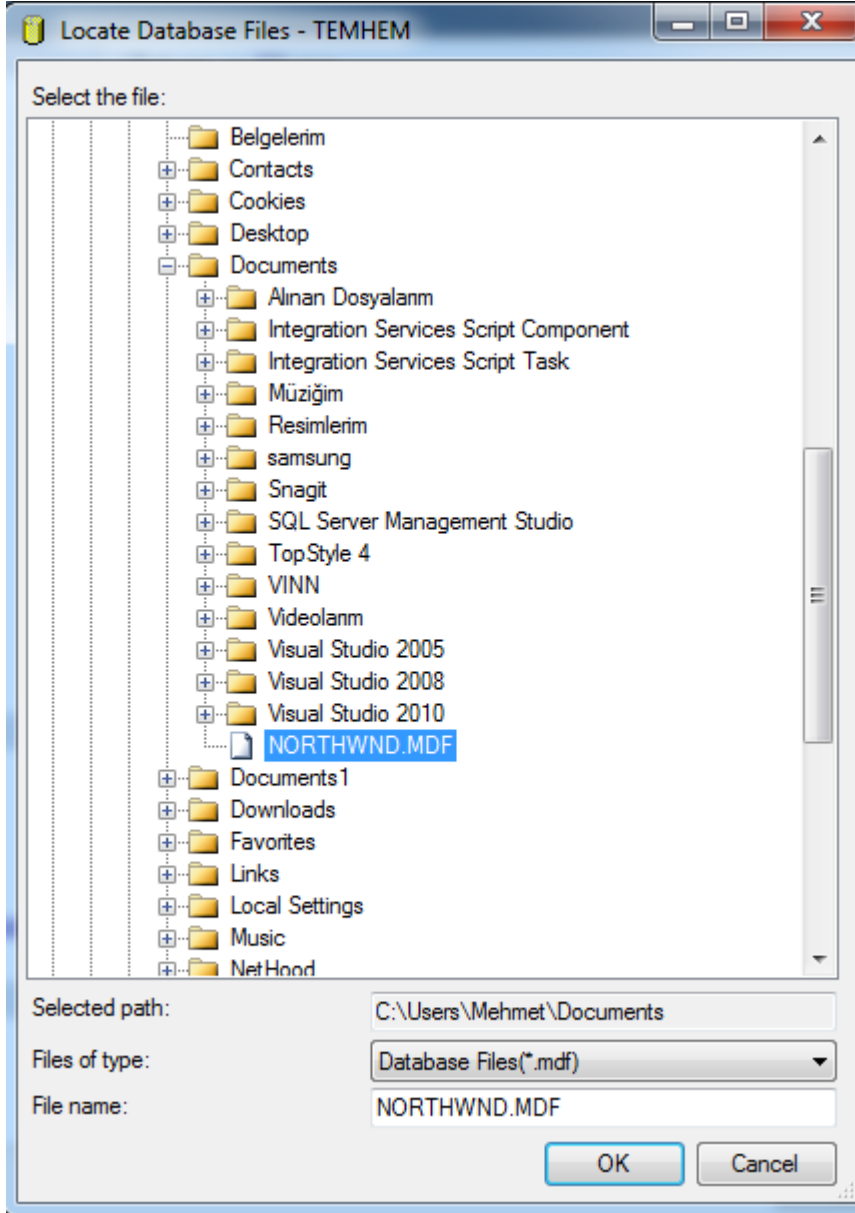


Resim 2.29: Object explorer attach database menüsü

Karşımıza Attach Databases penceresi gelecektir. Bu pencere bulunan Add butonuna tıklıyoruz.

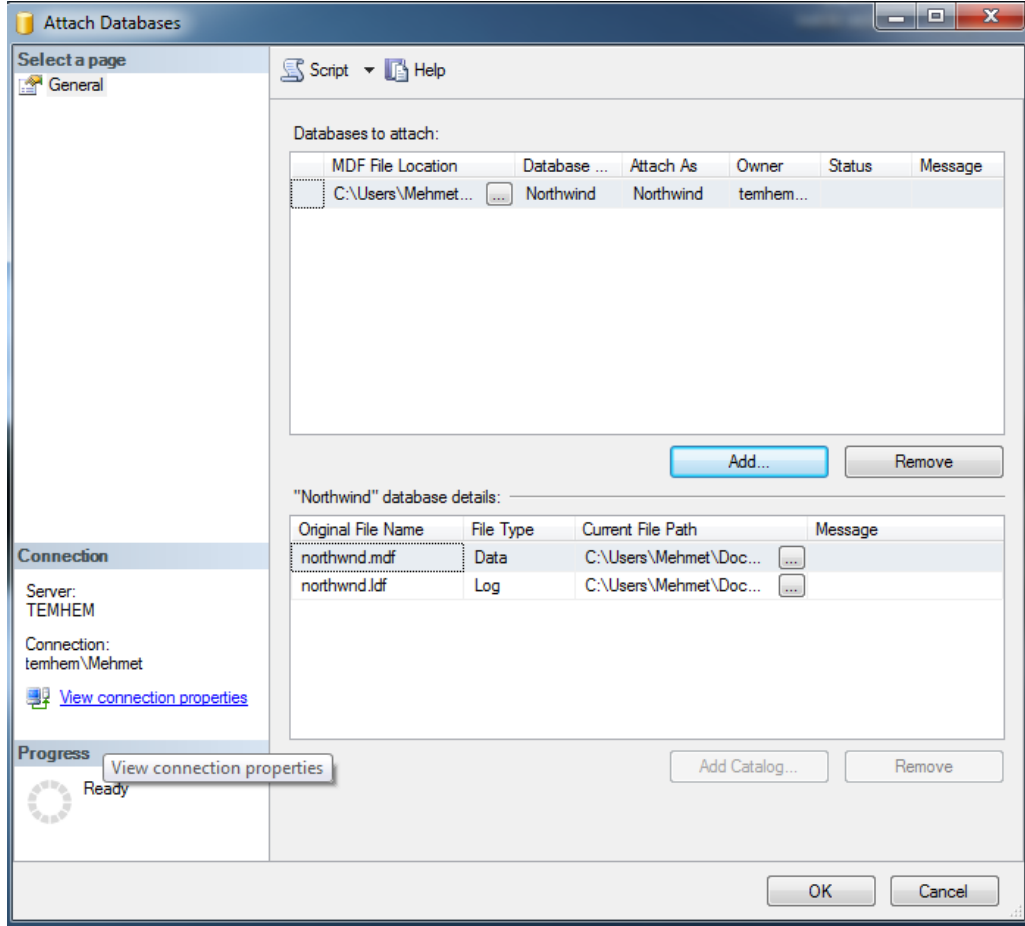


Resim 2.30: Attach databases penceresi



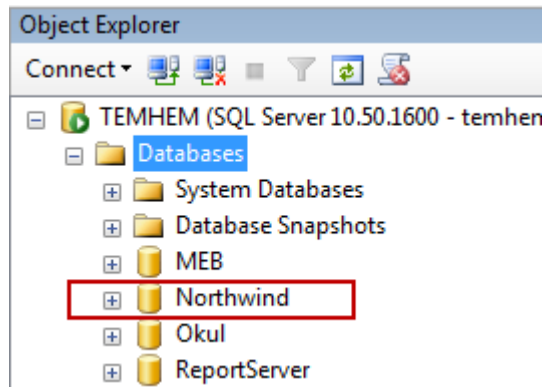
Resim 2.31: Windows explorer penceresi

Locate Database Files penceresinde, veritabanı dosyamızın yerini bulup OK butonuna basıyoruz.



Resim 2.32: Attach databases penceresi

En son Attach Database penceresinde de OK butonuna basarak veritabanını sunucumuza dâhil etmiş oluyoruz. Object Explorer penceresinde Refresh butonuna basarak listenin güncellenmesini sağlıyoruz. Dâhil ettiğimiz veritabanı Databases düğümü altında listeleniyor olacaktır.

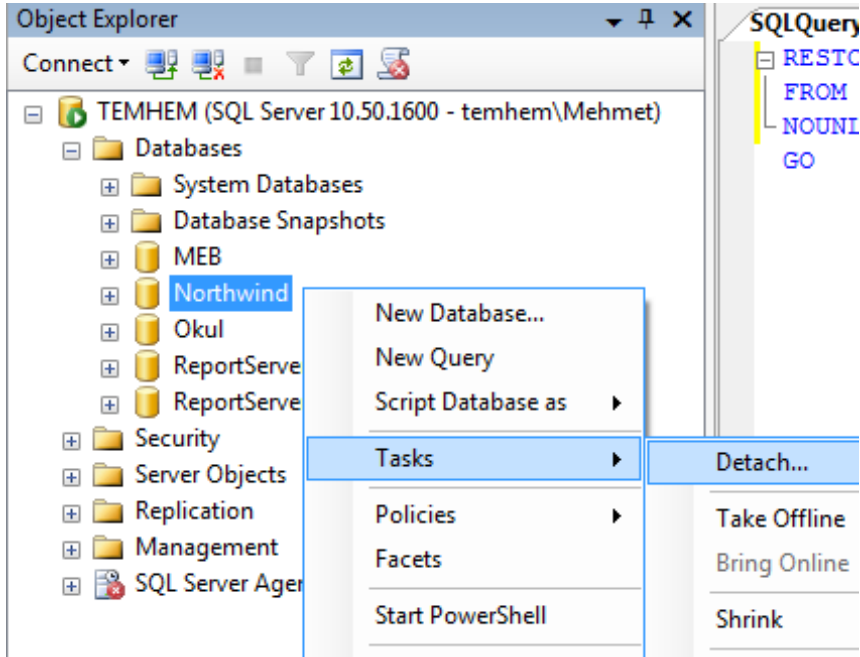


Resim 2.33: Object explorer penceresi veritabanı listesi

2.4.2. Veri Tabanının Sunucuya İlişğini Kesme

Veritabanı dosyamızın disk üzerinde yerini değiştirmemiz gerekirse öncelikle, veritabanının sunucusuyla ilişkisinin kesilmesi gerekmektedir. Dosyaları istediğimiz yere taşıdıktan sonra bir önceki konuda öğrendiğimiz şekilde sunucuya tekrar dâhil etmeliyiz.

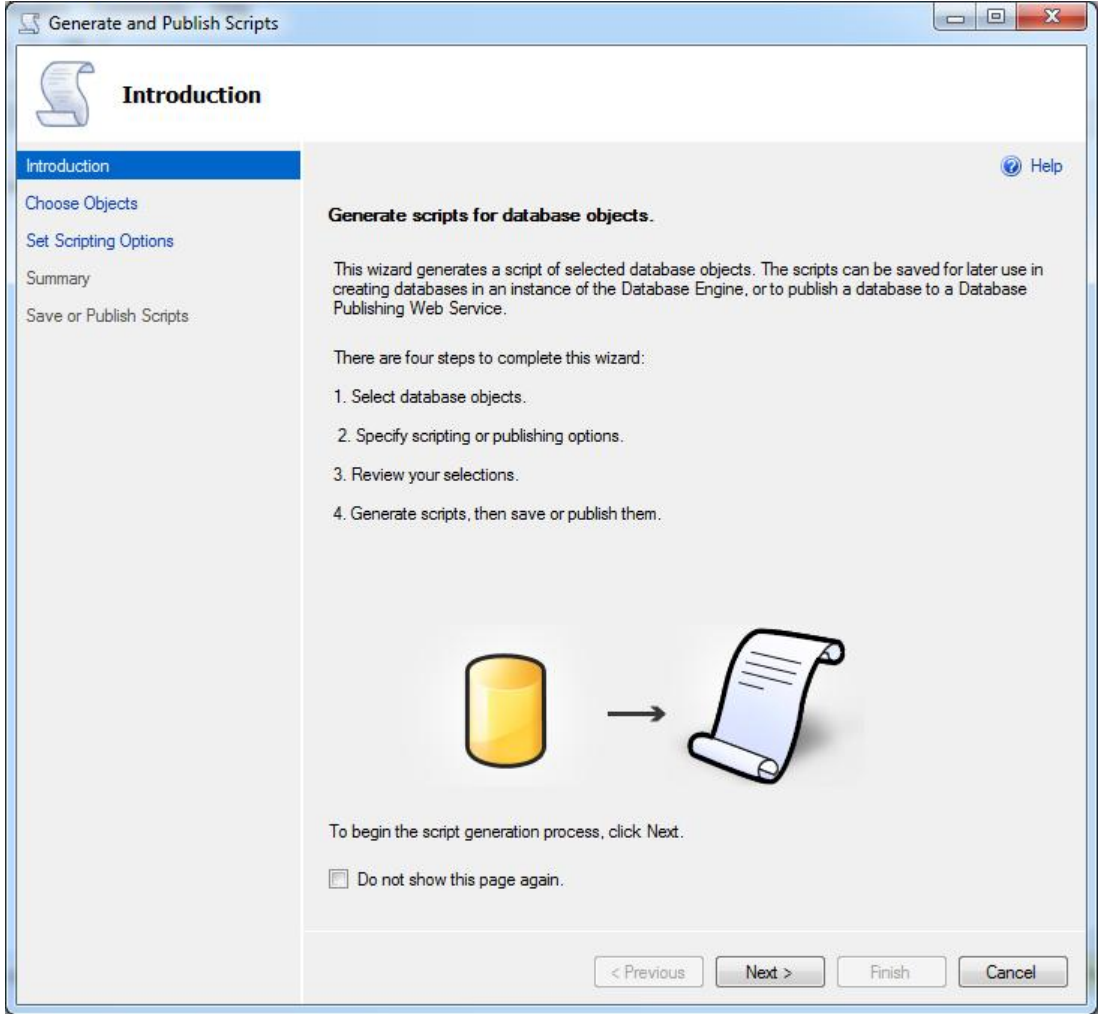
Veritabanının sunucusuyla ilişkisini kesmek için veritabanı üzerinde sağ tuşa tıklıyoruz. Açılan menüden Tasks→Detach seçeneğini seçiyoruz. Karşımıza Detach Database penceresi gelecektir. Ok butonuna basarak veritabanının sunucusuyla ilişkisini kesiyoruz.



Resim 2.34: Object explorer penceresi detach menüsü

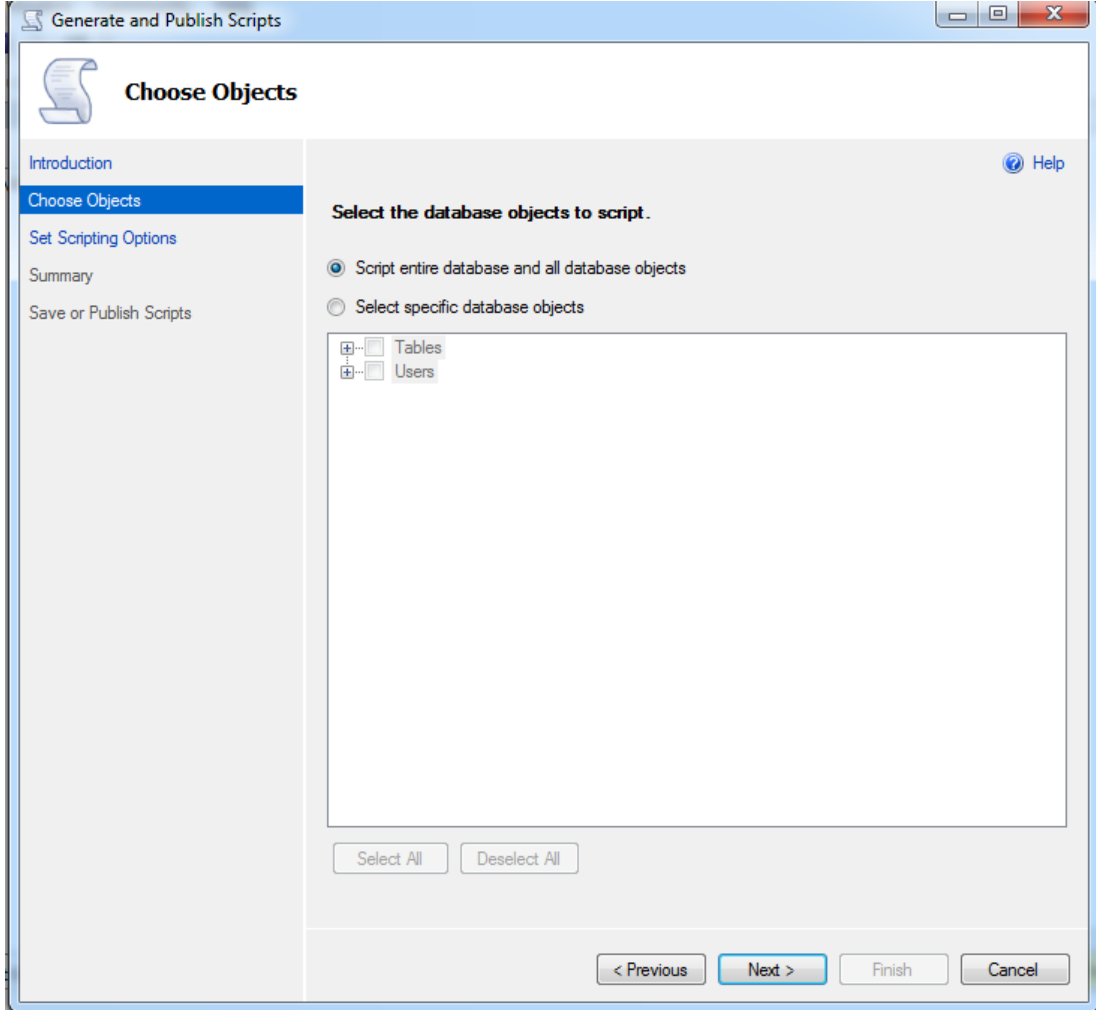
2.5. Veri Tabanının Script'ini Oluşturma

Bir veritabanı yapısını başka sunuculara taşımamız gerekebilir ya da bir üst sürüm olan sunucudan, bir alt sürüm sunucuya taşımak gerekebilir. Fakat geriye doğru restore yapamadığımızdan veritabanı dosyalarımızı t-sql komutlarına çevirerek bunu gerçekleştirebiliriz. Bu işlem için t-sql komutlarına çevrilecek olan veritabanı üzerinde sağ tuşa tıklıyoruz. Açılan menüden, Tasks→Generate Scripts seçeneğini seçiyoruz. Karşımıza Generate and Publish Script penceresi gelecektir.



Resim 2.35: Veritabanı script oluşturma penceresi introduction sekmesi

Introduction sekmesinde bize script oluşturmada yardımcı olacak araç ve hangi aşamalardan geçeceğimiz hakkında bilgilendirme yapılmaktadır. **Do not show this page again** kutucuğu işaretlenirse bir dahaki script oluşturmada bu pencere görüntülenmeyecektir. Next butonuna basarak bir sonraki adıma geçiyoruz.



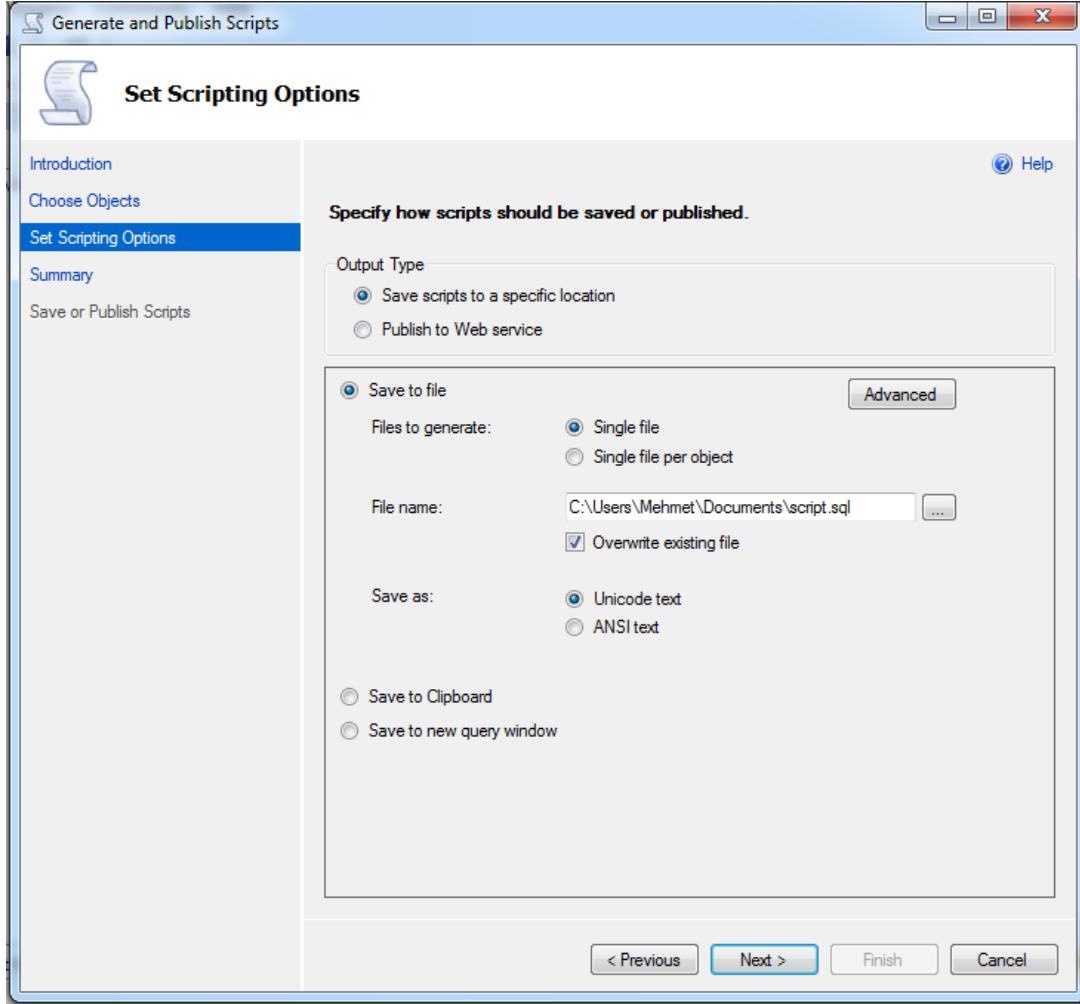
Resim 2.36: Veritabanı script oluşturma penceresi choose objects sekmesi

Choose Objects sekmesinde, veritabanı üzerindeki hangi nesnelerin t-sql komutlarına çevrilmesini istediğimizi belirliyoruz.

Script entire database and all database objects seçeneğini seçersek tüm nesnelerin t-sql komutlarına çevrilmesini sağlarız.

Select specfic database objects seçeneğini seçerek sadece istediğimiz nesnelerin t-sql komutlarına çevrilmesini sağlayabiliriz.

Seçeneklerden birini seçip Next butonuna basarak bir sonraki adıma geçiyoruz.



Resim 2.37: Veritabanı script oluşturma penceresi set scripting options sekmesi

Set Scripting Options sekmesinde script dosyamızın kayıt özellikleri yer alıyor.

Output Type bölümünde script dosyamızın disk üzerinde bir yere kayıt edeceğimizi mi yoksa web servisi olarak mı yayınlayacağımızı belirliyoruz.

Save to file seçeneği seçilirse:

Files to generate kısmında,Single file seçeneği seçilirse tüm t-sql komutlarını tek bir dosyada hazırlar.

Single file per object seçeneği seçilirse her nesne için ayrı bir script dosyası oluşturulur.

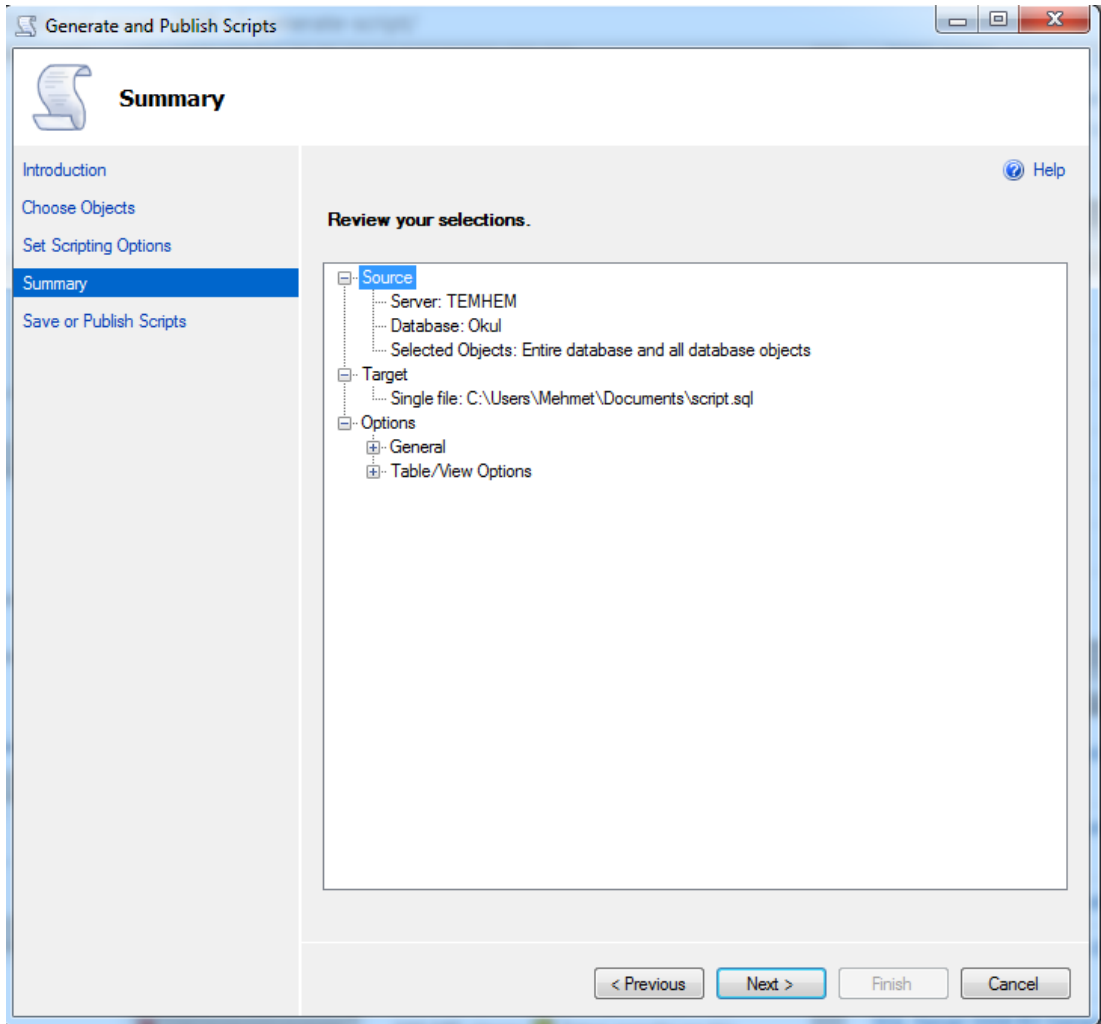
File name: Script dosyamızın disk üzerindeki konumu ve adını belirlediğimiz bölümdür. Overwrite existing file kutucuğu işaretlenirse file name bölümünde belirttiğimiz

konumda, aynı dosya adına sahip başka bir script dosyası varsa üzerine yazılma izni vermiş oluyoruz.

Save to Clipboard seçeneği seçilirse t-sql komutları kopyalama yapılmış gibi bellekte saklanır. Komutları istediğimiz bir dosya üzerine yapılandırabiliriz.

Save to new query window seçeneği seçilirse t-sql komutları yeni bir sorgu penceresine yerleştirilirler.

İhtiyacımıza göre bir seçeneği seçip Next butonuna basarak bir sonraki adıma geçiyoruz.

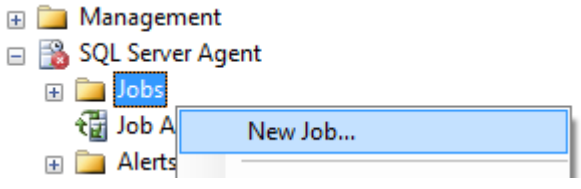
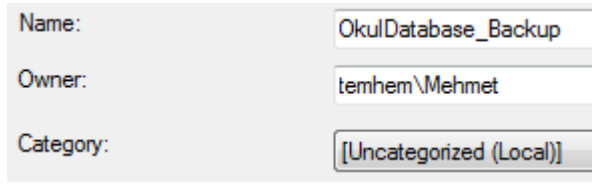
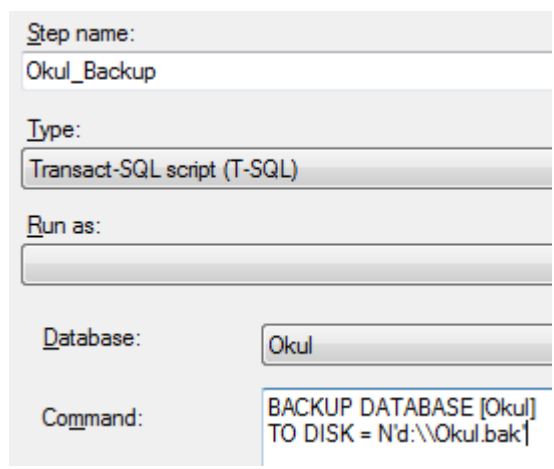


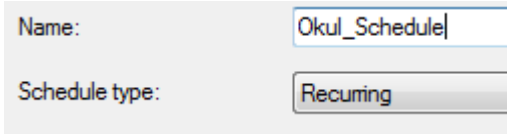
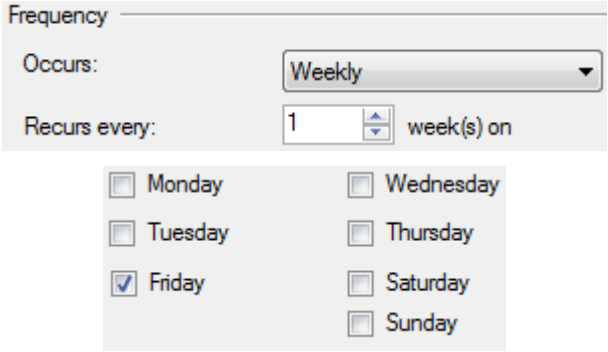
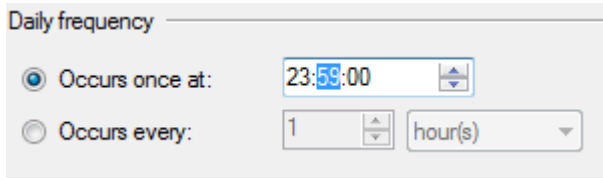
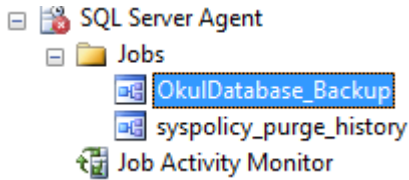
Resim 2.38: Veritabanı script oluşturma penceresi summary sekmesi

Summary sekmesinde şimdiye kadar yaptığımız ayarların özeti listelenir. Kontrol işleminden sonra Next butonuna basarak script dosyasının oluşturulmasını başlatıyoruz. Herhangi bir hata ile karşılaşılmadıysa script dosyamız belirlediğimiz şekilde ve konumda oluşturulacaktır.

UYGULAMA FAALİYETİ

Okul veritabanımızın yedeğini haftanın Cuma günleri saat 23.59 da alan her hafta tekrarlayacak şekilde bir görev oluşturunuz.

İşlem Basamakları	Açıklama
<p>➤ Object Explorer pencesinde bulunan SQL Server Agent düğümünü açıyoruz.</p>	
<p>➤ Jobs düğümü üzerinde sağ tuşa tıklıyor, açılan menüden New Job seçeneğini seçiyoruz.</p>	
<p>➤ Karşımıza gelen Yeni Görev penceresinin General sekmesinde bulunan Name bölümüne görev adını belirtiyoruz. Daha sonra Steps sekmesine geçiyoruz.</p>	
<p>➤ Steps sekmesinde yapılacak işler listelenir. New butonuna basarak yeni bir iş tanımlamamız gerekiyor.</p> <p>➤ Step name bölümüne bir isim tanımladıktan sonra Command bölümüne yapılacak işi belirtiyoruz. Ok butonuna basarak iş tanımlamamızı bitiriyoruz. Schedules adımına geçiyoruz.</p>	
<p>➤ Schedules bölümünde yapılacak işin zaman aralıklarını tanımlıyorduk. New butonuna basarak görevin çalışacağı zaman aralıklarını oluşturmaya başlıyoruz.</p>	

<ul style="list-style-type: none"> ➤ Name bölümüne bir isim tanımlıyoruz. ➤ Görevin belirttiğimiz zaman aralığında çalışması için Schedule type bölümünden Recuring seçeneğini seçiyoruz. 	
<ul style="list-style-type: none"> ➤ Görevimiz haftada bir çalışacağı için Frekans bölümündeki Occurs seçeneklerinden Weekly seçeneğini seçiyoruz. ➤ Bir kere tekrar edeceği için Recurs every bölümüne bir giriyoruz. ➤ Haftanın Cuma günleri çalışacağı için gün seçeneklerinden Friday'yi işaretliyoruz. 	
<ul style="list-style-type: none"> ➤ Görevimizin saat 23.59 da çalışması için Occurs once at bölümünde bu değeri belirtiyoruz. 	
<ul style="list-style-type: none"> ➤ Bütün ayarlamaları yaptığımıza göre Ok butonuna basarak zamanlama işlemini bitiriyoruz. 	
<ul style="list-style-type: none"> ➤ Görevimizi bitirmek için tüm adımları tamamladık. Yeni görev penceresindeki Ok butonuna basarak görevi tanımlama işlemini bitiriyoruz. 	
<ul style="list-style-type: none"> ➤ Object Explorer penceresinde yenile butonuna basalım. Yeni oluşturduğumuz görevin Jobs düğümü altında listelendiğini görüyoruz. 	

KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri		Evet	Hayır
1	Yeni bir görev oluşturmaya başlayabildiniz mi?		
2	Göreve bir isim atayabildiniz mi?		
3	Görevin yapacağı işi tanımlayabildiniz mi?		
4	Görevin çalışacağı zaman aralıklarını tanımlayabildiniz mi?		

DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Veritabanının tüm bileşenleriyle birlikte yedeğinin almak için Backup type bölümünde hangi seçenek seçilmelidir?
A) Differential B) Transaction Log C) Full D) Select All
2. Yedeği alınmış bir veri tabanını geri yükleme işlemine ne ad verilir?
A) Restore B) Backup C) Recovery D) Generate
3. Bir görev oluşturmak için aşağıdaki yollardan hangisi izlenmelidir?
A) Script>Script Action to New Query Window
B) Script>Script Action to File
C) Script>Script Action to Clipboard
D) Script>Script Action to Job
4. Bir görevin belirttiğimiz aralıklarda tekrar etmesi için Schedule type seçeneklerinden hangisini seçmeliyiz?
A) Start automatically when SQL Server Agent starts
B) Starts whenever the CPUs become idle
C) Recurring
D) One time
5. Veritabanının sunucusuyla ilişkisini kesme işlemine ne ad verilir?
A) Attach B) Detach C) Backup D) Restore

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise “Modül Değerlendirme”ye geçiniz.

MODÜL DEĞERLENDİRME

Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

1., sütunun hangi tipde veri tutacağıının belirlendiği bölümdür.
2.olarak belirtilmiş alanda veri tekrarı olmamalıdır.
3.veri tipi date ve time veri tiplerinin birleşmesinden oluşan bir veri tipidir.

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

4. Daha çok evet-hayır, doğru-yanlış gibi verileri tutmak için hangi veri türü kullanılmalıdır?
A) Evet/Hayır B) Boolean C) Bit D) Int
5. decimal(5,2) şeklinde tanımlanmış bir sütunun tutacağı değer nasıl olmalıdır?
A) 5,2 B) 34262,23 C) 34,26223 D) 342,62
6. time(2) şeklinde tanımlanmış veri tipi bellekte ne kadar yer kaplar?
A) 2 byte B) 3 byte C) 4 byte D) 5 byte
7. Aşağıdakilerden hangisi unicode karakter veri tiplerindedir?
A) nchar B) float C) char D) binary
8. A tablosundaki bir kaydın B tablosunda tek bir kayda karşılık gelen ilişki türü hangisidir?
A) Bire Bir B) Bire Çok C) Çoğa Çok D) Hiçbiri
9. Veritabanı yedeği almak için hangi yol izlenmelidir?
A) Tasks>Restore B) Tasks>Back Up
C) Tasks>Script D) Tasks>Database
10. Yedek alma işleminden sonra Sql sorgusu oluşturulmasını istiyorsak hangi yok izlenmelidir?
A) Script>Script Action to Job
B) Script>Script Action to Clipboard
C) Script>Script Action to File
D) Script>Script Action to New Query Window

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ 1'İN CEVAP ANAHTARI

1	C
2	A
3	D
4	B
5	D
6	B
7	Identity Specification
8	Yabancı Anahtar
9	Parse

ÖĞRENME FAALİYETİ 2'NİN CEVAP ANAHTARI

1	C
2	A
3	D
4	C
5	B

MODÜL DEĞERLENDİRME CEVAP ANAHTARI

1	Data Type
2	Birincil anahtar
3	datetime2
4	C
5	D
6	B
7	C
8	A
9	A
10	D

KAYNAKÇA

- <http://www.microsoft.com/sqlserver/2008/tr/tr/overview.aspx>