

T.C.
MİLLÎ EĞİTİM BAKANLIĞI



MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

BİLİŞİM TEKNOLOJİLERİ

AĞ VERİ TABANI PLANLAMA

ANKARA 2008

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR	ii
GİRİŞ	1
ÖĞRENME FAALİYETİ - 1	3
1. VERİ TABANI	3
1.1. Veri Tabanı Oluşturmak.....	3
1.1.1. Management Studio ile Veri tabanı Oluşturmak	3
1.1.2. Transact SQL ile Veri Tabanı Oluşturmak	5
1.2. Veri Tabanı Seçeneklerini Ayarlamak	6
1.3. Veri Tabanı Silmek	8
1.4. İlişkisel Veri Tabanı Nesneleri	8
1.5. Tablolar	8
1.5.1. Tablo Oluşturmak	8
1.5.2. Tabloya Sütun Ekleme.....	10
1.5.3. Sütun Özelliklerini Değiştirmek	12
1.5.4. Tablodan Sütun Silmek.....	14
1.5.5. Tablo Silmek.....	14
1.5.6. Nesnelerin T-SQL İfadelerini Almak	15
1.6. Veri Türleri	20
1.6.1. Karakter Dizilimi	21
1.6.2. Unicode Karakter Dizilimi	22
1.6.3. Tarih ve Saat Veri Türleri.....	23
1.6.4. Tahmini Sayısal Veri Türü	23
1.6.5. Kesin Sayısal Veri Türü.....	23
1.6.6. Tam Sayı Veri Türü	24
1.6.7. Parasal Veri Türü.....	24
1.6.8. Binary (İkili) Veri Türü	24
1.6.9. Özel Veri Türleri.....	25
1.6.10. Kullanıcı Tanımlı Veri Türleri.....	25
UYGULAMA FAALİYETİ	27
ÖLÇME VE DEĞERLENDİRME	28
ÖĞRENME FAALİYETİ - 2	28
2. PLANLAMA	28
2.1. Veri Tabanı Normalizasyonu	28
2.2. Kayıtların İlişkilendirilmesi	31
2.2.1. Bire-Bir İlişki.....	31
2.2.2. Bir-Çok İlişki	31
2.2.3. Çok-Çok İlişki	32
2.3. Dosya Grupları (File Groups)	32
2.3.1. Dosya Gruplarında (File Groups) Dikkat Edilmesi Gereken Hususlar	32
2.3.2. Dosya Gruplarının (File Groups) Faydaları.....	33
UYGULAMA FAALİYETİ	34
ÖLÇME VE DEĞERLENDİRME	38
MODÜL DEĞERLENDİRME	39
CEVAP ANAHTARLARI	40
KAYNAKÇA	41

AÇIKLAMALAR

KOD	481BB0043
ALAN	Bilişim Teknolojileri
DAL/MESLEK	Veri Tabanı Programcılığı
MODÜLÜN ADI	Ağ Veri Tabanı Planlama
MODÜLÜN TANIMI	Bir veri tabanı oluşturma, veri tabanı bileşenlerini kullanma ve veri modeli yapabilmeye ilgili bilgilerin kazandırıldığı öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	Ağ Veri Tabanı- 1 modülünü bitirmiş olmak
YETERLİK	
MODÜLÜN AMACI	Genel Amaç Gerekli ortam sağlandığında, veri tabanı oluşturup veri tabanı nesnelere kullanabileceksiniz. Amaçlar 1. Veri tabanı kavramları ve nesnelere kullanabileceksiniz. 2. Veri tabanı planlaması üzerinde çalışabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Ortam: Atölye, laboratuvar, bilgi teknolojileri ortamı (internet) vb., kendi kendinize veya grupta çalışabileceğiniz tüm ortamlar. Donanım: Ağ veri tabanını çalıştırabilecek yeterlikte bilgisayar, yedekleme için gerekli donanım (CD yazıcı, flash bellek), raporlama için yazıcı, kâğıt ve kalem
ÖLÇME VE DEĞERLENDİRME	➤ Her faaliyet sonrasında o faaliyetle ilgili değerlendirme soruları ile kendi kendinizi değerlendireceksiniz. ➤ Modül sonunda uygulanacak ölçme araçları ile modül uygulamalarında kazandığınız bilgi ve beceriler ölçülerek değerlendirilecektir.

GİRİŞ

Sevgili Öğrenci,

Okul yaşantınızda öğreneceğiniz her konu, yaptığınız uygulamalar ve tamamladığınız her modül bilgi dağarcığınızı geliştirecek ve ilerde atılacağınız iş yaşantınızda size başarı getirecektir. Eğitim sürecinde daha öz verili çalışır ve çalışma disiplini kazanırsanız; başarılı olmamanız için hiçbir neden yoktur.

Son yıllarda yapılan birçok proje çok sayıda bilgisayar tarafından kullanılabilir şekilde tasarlanmaktadır. Bu yüzden, ağ ortamında birden fazla kullanıcı aynı proje üzerinde çalışabilmektedir. Bu işlemleri çok sık kullandığınız veri tabanı programıyla da yapabilmenize rağmen ağ ortamında güvenlik ve hızlı erişim açısından en iyi sonucu veren SQL Server veri tabanı da yapabilirsiniz. Bu programla, milyonlarca kaydın olduğu tablolar üzerinde işlem yaparken tüm kullanıcılara hitap edebilmektedir. İstenilen sorgu sonuçlarını da en hızlı şekilde elde edebilmenizi sağlar.

Bu modülle, veri tabanı nesnelere, veri tabanının oluşturulmasını ve planlamasını öğreneceksiniz.

ÖĞRENME FAALİYETİ-1

AMAÇ

Veri tabanı oluşturmayı, tablo nesnesini kullanmayı ve veri türlerini öğreneceksiniz.

ARAŞTIRMA

- Daha önce öğrendiğiniz modüllerde bir veri tabanının nasıl oluşturulduğunu araştırınız.
- Bir tablo oluşturmanın faydalarını araştırınız.

1. VERİ TABANI

Veri tabanı düzenli bilgiler topluluğudur.

Kelimenin anlamı; bilgisayar ortamında saklanan düzenli verilerle sınırlı olmamakla birlikte, daha çok bu anlamda kullanılmaktadır.

Bilgisayar terminolojisinde, sistematik erişim imkânı olan, yönetilebilir, güncellenebilir, taşınabilir, birbirleri arasında tanımlı ilişkiler bulunabilen bilgiler kümesidir.

Bir başka tanımı da, bir bilgisayarda sistematik şekilde saklanmış, programlarca istenebilecek veri yığınıdır.

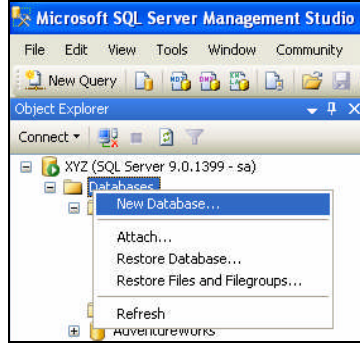
1.1. Veri Tabanı Oluşturmak

Veri tabanı iki yöntemle oluşturulur:

- Management Studio kullanılarak
- T-SQL ifadesi olan *CREATE DATABASE* deyimi kullanılarak

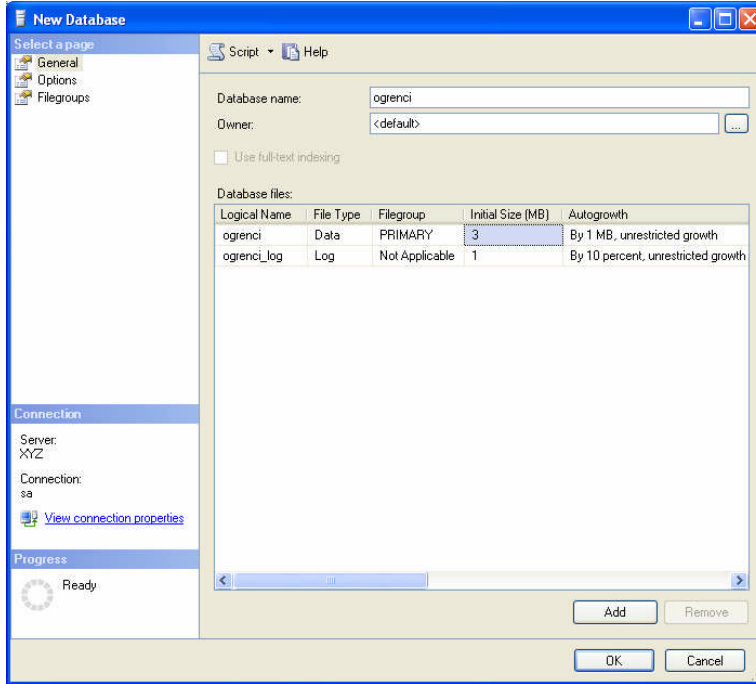
1.1.1. Management Studio ile Veri tabanı Oluşturmak

Management Studio ile veri tabanı oluşturmak için SQL Server Management Studio'yu açınız. Object Explorer'daki Databases üzerinde fareyle sağ tıklayarak New Database komutunu seçiniz.



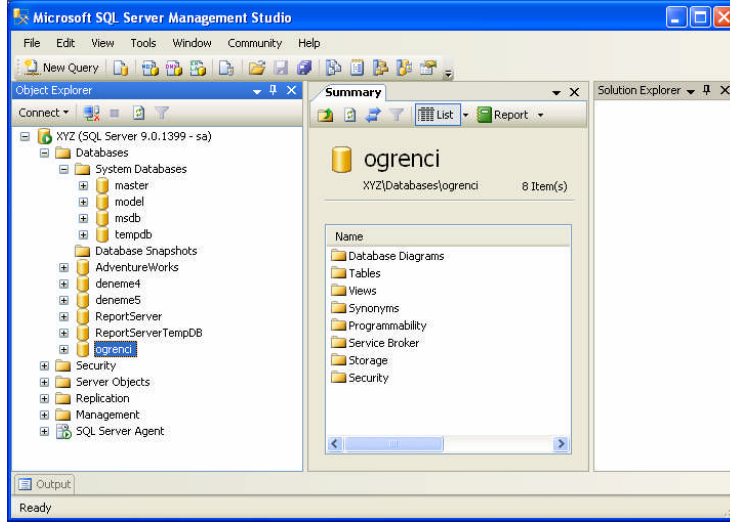
Resim 1.1: New Database komutunun seçimi

Database name alanına veri tabanı için uygun bir isim girerek OK düğmesine tıklatınız.



Resim 1.2: Veri tabanına isim verme

Veri tabanınız oluşmuş olacaktır.



Resim 1.3: Oluşturulan veri tabanının Object Explorer'daki görüntüsü

1.1.2. Transact SQL ile Veri Tabanı Oluşturmak

Bir veri tabanı Management Studio ile oluşturulabildiği gibi diğer yöntem olan T-SQL ile de oluşturulabilir. Ayrıca, Management Studio ile yapılan tüm işlemler T-SQL ile de yapılabilir.

T-SQL ile bir veri tabanının oluşturulması en basit hâliyle şu şekildedir.

```
CREATE DATABASE veri_tabanı_ismi
```

Ancak, bu şekilde bir veri tabanı oluşturmak, model veri tabanının kopyalanması ve yeni kopyanın adının belirtilen veri tabanı ismiyle değiştirilmesidir. Bir veri tabanı dosyasını çeşitli özellikleriyle oluşturacak olursak genel kullanımı şöyle olacaktır.

```
CREATE DATABASE veri_tabanı_ismi
ON
PRIMARY ( NAME=dosya_takma_ismi
, FILENAME=fiziki_dosya_ismi
, SIZE=dosya_boyutu
, MAXSIZE=maksimum_dosya_boyutu
, FILEGROWTH=dosya_artım_miktarı )
```

```
LOG ON
( NAME=dosya_takma_ismi
, FILENAME=fiziki_dosya_ismi
, SIZE=dosya_boyutu
, MAXSIZE=maksimum_dosya_boyutu
, FILEGROWTH=dosya_artım_miktarı )
```

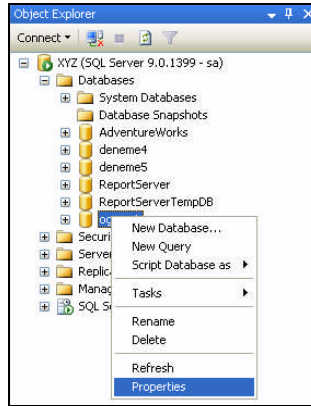
Birincil (Primary) dosya, veri tabanı için gerekli olan tüm nesnelere saklayan dosyadır. Bir veri tabanı oluşturulurken birincil dosya belirtilmemişse ilk oluşturulan dosya birincil dosya olarak belirlenir. Veri tabanında mutlaka birincil dosya bulunmalıdır.

Oluşturulan veri tabanında kullanılan ifadelerin anlamları şöyledir.

- **Veri tabanı_ismi:** Veri tabanına verilecek isimdir.
- **Dosya takma_ismi:** Veri tabanındaki dosyalar için belirlenen takma isimdir. İlgili dosyaya erişimde pratiklik sağlar. İşletim sisteminde bilinen isimdir.
- **Fiziki dosya_ismi:** İşletim sisteminde saklanacak dosyanın adını ve yolunu belirtir. Bu isimde işletim sisteminde bilinen bir isimdir.
- **Dosya boyutu:** Veri tabanı tanımlandığındaki boyutunu belirtir. Başlangıçta 1MB'tır. Management Studio'da bu 3MB olarak tanımlanır. Oluşturulacak log dosyası da bunun %10'u kadardır.
- **Maksimum dosya boyutu:** Dosyanın maksimum boyutunu belirtir. Belirtilmezse disk dolana kadar dosya artım miktarı kadar artmaya devam eder.
- **Dosya artım miktarı:** Dosyanın belirtilen boyutu dolduğunda dosya boyutu otomatik olarak artar. Artım miktarı bu parametreyle belirtilir.

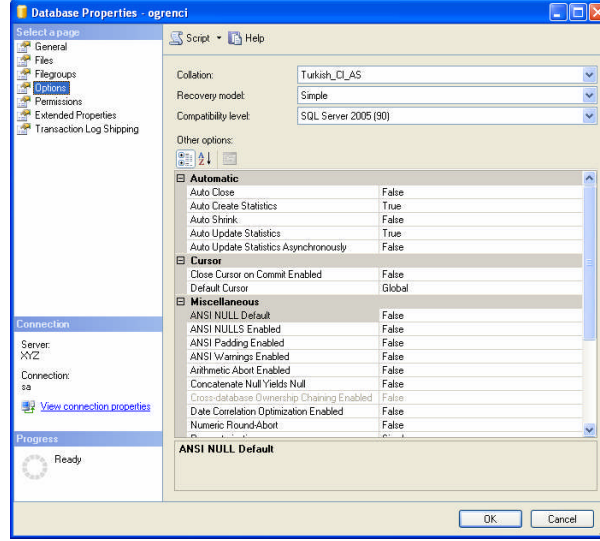
1.2. Veri Tabanı Seçeneklerini Ayarlamak

Veri tabanı özelliklerinin düzenlenmesinin gerektiği durumlarda bazı ayarlamalar yapmak gerekir. Bu ayarlar, erişim ve nesnelere çalışmasıyla ilgili ayarlardır. Ayarlamalar için Management Studio'da iken veri tabanı üzerinde sağ tıklanır ve Properties komutu seçilir.



Resim 1.4: Veri tabanı özellikleri için Properties komutunun seçilmesi

Properties komutu seçilince ekrana Database Properties iletişim penceresi gelir.



Resim 1.5: Database Properties iletişim penceresi

Options sekmesinden gerekli olan ayarlamalar yapılabilir. Ayrıca, T-SQL deyimi olan *ALTER DATABASE* deyimi de aynı işlevi görmektedir.

Veri tabanında ayarlanabilecek bazı özellikler şunlardır:

- **Auto Close:** Server kullanıcıları bağlantılarını kestiklerinde veri tabanı sunucusu otomatik olarak dosyaları serbest bırakır. Bu dosyalar işletim sisteminden değiştirilebilir hâle gelir.
- **Auto Create Statistics ve Auto Update Statistics:** SQL Server'da bazı komutlar için sorgulamalar yapılırken sorgu sonuçları istatistiksel olarak kaydedilir ve güncellenebilir.
- **Auto Shrink:** Veri tabanının otomatik olarak küçültülmesini ayarlar.
- **ANSI Null Default:** Bu seçenek False olursa tablodaki alan boş olamaz demektir. True bunun tam tersidir.
- **Database Read Only:** Bu seçenek veri tabanının sadece okunabilir olmasını sağlar. Değiştirilme yapılamaz.
- **Recovery Model:** Veri tabanı üzerinde yapılan işlemlerin ayrıntılı olup olmaması bu seçimle ayarlanır.
- **Restrict Access:** Restricted seçimi, Server'a bağlı kullanıcılar ve db_owner (sahip) kullanıcı dışındaki diğer kullanıcıların veri tabanı dosyasına erişimini engeller. Seçeneğin Single olması, tek bir kişinin veri tabanını kullanmasını sağlar. Multiple ise birçok kullanıcının bu veri tabanını kullanmasını sağlamaktadır.
- **Collation:** Dil ayarlarının yapıldığı seçenektir.

1.3. Veri Tabanı Silmek

Veri tabanını silmek için Management Studio programı veya *DROP DATABASE* deyimi kullanılır. Ancak, veri tabanının yedek yükleme, replication veya kullanıcı erişimi gibi başka bir işlem yapmıyor olması gerekir. db_owner veya sysadmin rollerinden birisi veri tabanını silebilir.

1.4. İlişkisel Veri Tabanı Nesneleri

İlişkisel veri tabanı, tabloların birbirleriyle ilişkili olması demektir. Böylece, ilişkisel veri tabanları sayesinde çok fazla veriyi az bir alanda ve daha fazla kontrol edilebilir olarak saklamamızı sağlar. Bir veri tabanı, veri, tablo ve diğer nesnelere bir yığındır. Veri tabanı nesnelere, veriyi tasarlamada ve veri güvenlik mekanizmasını tanımlamada yardımcı olur.

İlişkisel veri tabanı nesnelere şunlardır:

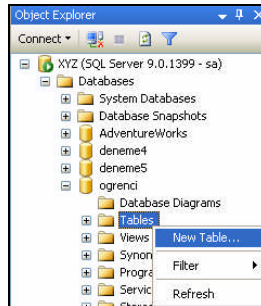
- Tablolar
- Diyagramlar
- Görünümler
- Saklı yordamlar
- Kullanıcılar
- Roller
- Kurallar
- Varsayılan Değerler
- Kullanıcı Tanımlı Veri Türleri
- Kullanıcı Tanımlı Fonksiyonlar

1.5. Tablolar

SQL Server’da her işlem tablolarda yapıldığından, tabloların çeşitli özelliklerini ve tabloları yönetmeyi bu bölümde öğreneceksiniz. Tablolarda işlemler Management Studio’yla yapılabildiği gibi T-SQL komutlarıyla da yapılabilir.

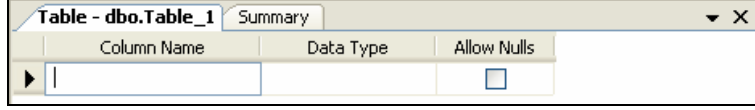
1.5.1. Tablo Oluşturmak

Management Studio’yla tablo oluşturmak için Object Explorer penceresinde daha önceden oluşturmuş olduğunuz veri tabanının Tables seçeneği üzerinde fareyle sağ tıklatarak açılan menüden New Table seçeneğini tıklattınız. Böylece bir tablo oluşturmak için gerekli olan ilk adımı gerçekleştirmiş olacaksınız.



Resim 1.6: New Table komutu


New Table komutunu verdiğinizde tabloların alanlarını, veri tipini ve alana bilgi girilip girilmeyeceği zorunluluğunun belirtildiği ekran görüntüsü karşınıza gelecektir.



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

Resim 1.7: Tablo oluşturma ekranı

Column Name, tablodaki alanların adını; Data Type, alanların alacağı verilerin türlerini; Allow Nulls, alanın bilgi içerip içermeyeceğinin zorunluluğunu göstermektedir.

Tabloda bulunmasını istediğiniz alanları belirledikten sonra oluşturduğunuz tabloyu kaydetmeniz gerekmektedir. Bunun için araç çubuğunda bulunan  (Save) düğmesine tıklamalısınız. Save düğmesine tıkladığınızda ekrana Choose Name diyalog kutusu gelecektir.



Choose Name

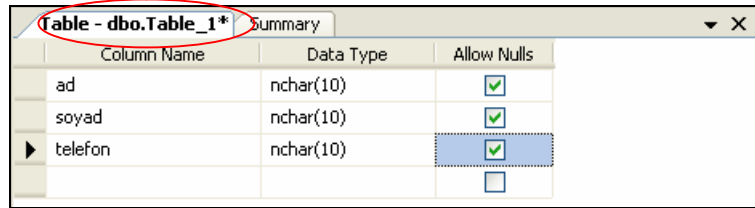
Enter a name for the table:

Table_1

OK Cancel

Resim 1.8: Choose Name diyalog kutusu

Tablo kaydedilmeden önce varsayılan adı Table_1'dir. İsterseniz bu adı değiştirmeden de kullanabilirsiniz.



Column Name	Data Type	Allow Nulls
ad	nchar(10)	<input checked="" type="checkbox"/>
soyad	nchar(10)	<input checked="" type="checkbox"/>
telefon	nchar(10)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

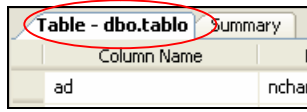
Resim 1.9: Tablonun varsayılan adının gösterimi

İsterseniz tablonun adını uygulamanızın içeriğine göre de adlandırabilirsiniz. Müşterilerle ilgili bir veri tabanında oluşturduğunuz tabloya "TabloSiparis" adını verebilirsiniz.



Resim 1.10: Tablo adının isteğe göre düzenlenmesi

Tablo adını yazdıktan sonra OK düğmesine tıklatınız. Artık tablonun adı yazdığınız ad olarak değiştirilecektir.



Resim 1.11: Tablo adının kaydedildikten sonraki görüntüsü

Böylece Management Studio kullanılarak bir tablo oluşturulmuş olur.

T-SQL ile de tablo oluşturulabilir. Tablo Oluşturma için genel ifade şu şekildedir:

```
CREATE TABLE tablo_adi  
(kolon_adi1 veri_tipi[NOT NULL],[  
kolon_adi2 veri_tipi[NOT NULL],.....])
```

şeklindedir.

[NOT NULL] ifadesi yazılması zorunlu olmayan bir ifadedir. Kolon adları istenilen sayıda olabilir.

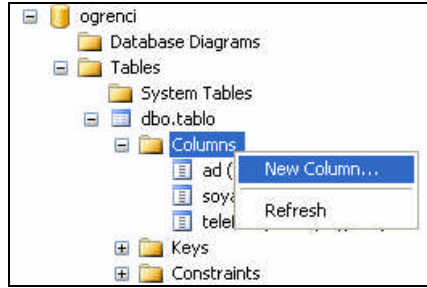
Örnek

```
CREATE TABLE tablo2  
(ad nchar NOT NULL,  
soyad nchar NOT NULL)
```

şeklinde bir T-SQL yazımıyla da tablo oluşturulabilir.

1.5.2. Tabloya Sütun Ekleme

Tabloya sütun eklemek için, oluşturduğunuz tablonun alt seçeneği olan Column seçeneği üzerinde fareyle sağ tıkladığınızda açılan menüden New Column komutunu tıklamanız gerekmektedir.



Resim 1.12: New Column komutu

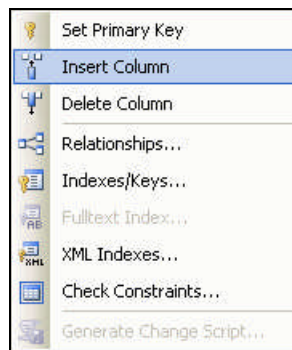
New Column komutunu verdiğinizde daha önceden oluşturduğunuz sütunların en altına bir sütun daha eklenecektir.

Column Name	Data Type	Allow Nulls
ad	nchar(10)	<input checked="" type="checkbox"/>
soyad	nchar(10)	<input checked="" type="checkbox"/>
telefon	nchar(10)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Resim 1.13: Yeni bir sütunun eklenmesi

Eklenen sütuna bir ad verdikten ve veri tipini seçtikten sonra tablonuza bir sütun daha eklemiş olursunuz.

Bir başka yöntemle, tablo ekranında farenin sağ tuşuna bastığınızda gelen menüden Insert Column komutunu vermeniz yeterli olacaktır.



Resim 1.14: Sağ tuşla gelen menüden sütun eklenmesi

T-SQL komutuyla da tabloya sütun ekleyebilirsiniz. Tablo üstünde değişiklik yapabilmek için genel olarak *ALTER TABLE* deyimi kullanılır.

Tabloya sütun eklemek için kullanılacak genel yapı şu şekilde olmalıdır.

```
ALTER TABLE tablo_adi  
ADD sutun_adi sutun_ozelligi
```

Sütun özelliği parametresinde veri tipi, sütunun uzunluğu gibi tanımlamalar yer alır.

Örnek

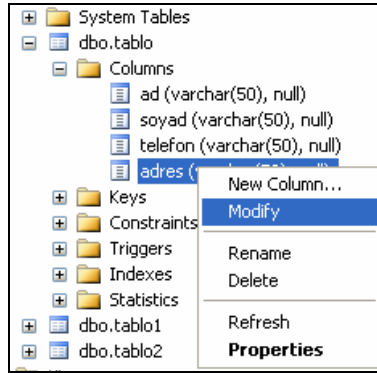
```
ALTER TABLE tablo1  
ADD adres VARCHAR(50)
```

şeklinde bir yazımla tablonuza bir sütun eklemiş olursunuz.

Bir tabloya, kayıt girildikten sonra sütun eklenecekse, bu sütun NOT NULL ile tanımlanırken dikkat etmek gerekir. Hesaplanmış sütun veya default değer olmadan bu işlem gerçekleşmez.

1.5.3. Sütun Özelliklerini Değiştirmek

Bir tabloda istenilen sütunun adını, türünü ve genişliğini değiştirmek mümkündür. Bunun için oluşturulmuş tablonun üzerinde sağ klik yaparak açılan menüden Modify komutunu tıklamanız gerekmektedir.



Resim 1.15: Modify komutunun verilmesi

Modify komutunun verilmesiyle değiştirmek istediğiniz sütun diğer mevcut sütunlarla birlikte ekrana gelecektir.

Column Name	Data Type	Allow Nulls
ad	varchar(50)	<input checked="" type="checkbox"/>
soyad	varchar(50)	<input checked="" type="checkbox"/>
telefon	varchar(50)	<input checked="" type="checkbox"/>
adres	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Resim 1.16: Değiştirilmek istenen sütun

Sütun adı Resim 1.17'deki gibi değiştirilebilir.

Column Name	Data Type	Allow Nulls
ad	varchar(50)	<input checked="" type="checkbox"/>
soyad	varchar(50)	<input checked="" type="checkbox"/>
telefon	varchar(50)	<input checked="" type="checkbox"/>
adresler	varchar(100)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Resim 1.17: Sütun adının değiştirilmesi

Sütunun genişliğini veya veri tipini değiştirmek için de Column Properties (Sütun özellikleri) penceresinden gerekli değişiklikleri yapabilirsiniz.

Column Properties	
<div style="border: 1px solid gray; padding: 2px;"> A Z ↓ </div>	
(General)	
(Name)	adresler
Allow Nulls	Yes
Data Type	varchar
Default Value or Binding	
Length	100
Table Designer	
Length	

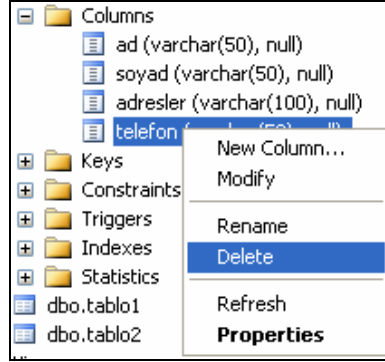
Resim 1.18: Sütun özellikleri penceresi

Tüm bu işlemleri yaptıktan sonra projenizi kaydetmeyi unutmayınız.

Ayrıca, Object Explorer penceresinde de yapılan değişiklikleri görmek için sütun (Columns) üzerinde iken sağ klik yapıp açılan menüden Refresh komutunu vermelisiniz.

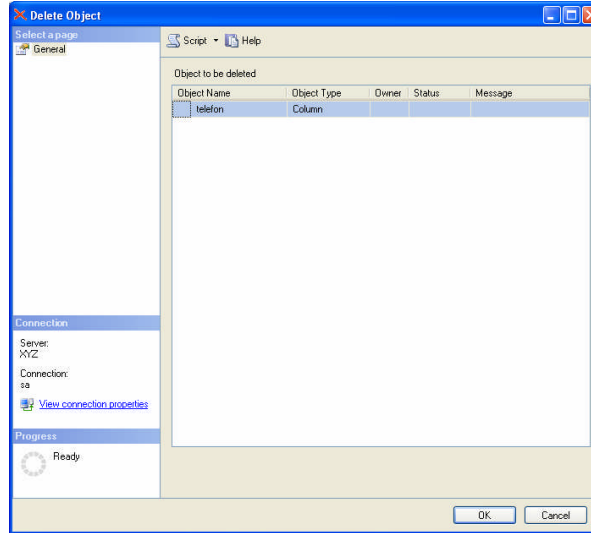
1.5.4. Tablodan Sütun Silmek

Tabloda bir sütunu silmek için o sütun üzerinde sağ klik yapıp açılan menüden Delete komutunu vermeniz yeterli olacaktır.



Resim 1.19: Sütunun silinmesi

Delete komutunu verdiğinizde ekrana Delete Object penceresi gelecektir.

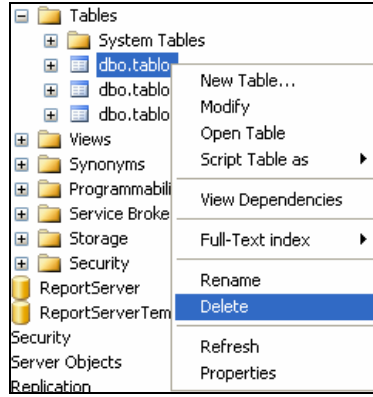


Resim 1.20: Delete Object penceresi

Silmek istediğiniz sütun adı bu pencerede gösterilecek ve OK düğmesine tıklamanız hâlinde sütun silinmiş olacaktır.

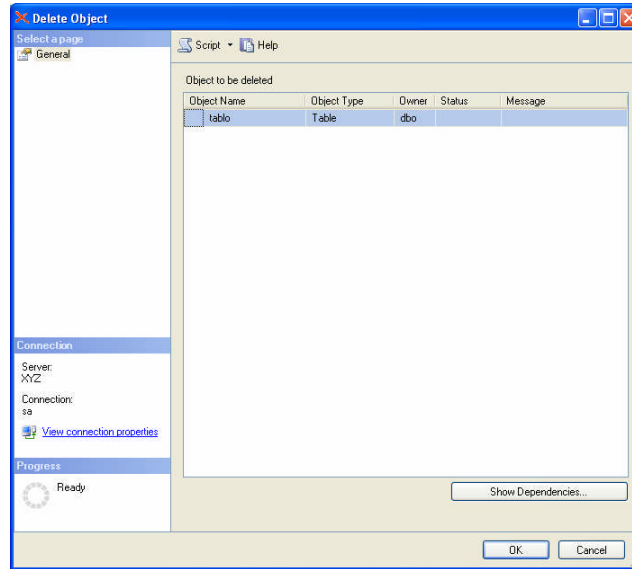
1.5.5. Tablo Silmek

Oluşturulan bir tabloyu silmek için Object Explorer penceresinde silmek istediğiniz tablonun üzerinde sağ klik yapıp açılan menüden Delete komutunu vermeniz yeterli olacaktır.



Resim 1.21: Delete komutunun verilmesi

Delete Object penceresinde silmek istediğiniz tablo gösterilecektir. OK düğmesine basarak tabloyu silebilirsiniz.

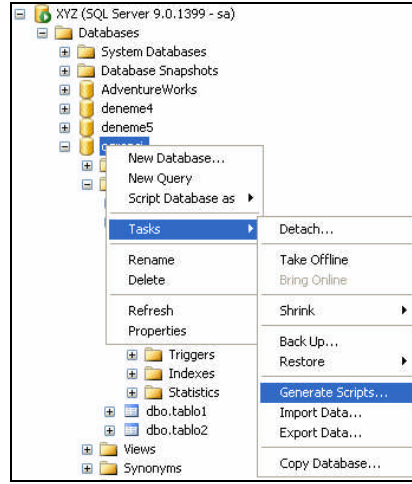


Resim 1.22: Delete Object ekranı

1.5.6. Nesnelerin T-SQL İfadelerini Almak

Oluşturulan projeler başka bilgisayarlarda kullanılmak istendiğinde bu projelerin veri tabanını oluşturulan nesnelerinin T-SQL ifadelerini bir dosya hâlinde saklamak mümkündür. Böylece bu dosya başka bir ortamda çalıştırılarak aynı şema elde edilebilir.

Management Studio ile veri tabanının T-SQL script'ini oluşturmak için veri tabanı üzerinde sağ klik yapıldıktan sonra açılan menüden Tasks\Generate Scripts komutu tıklanır.



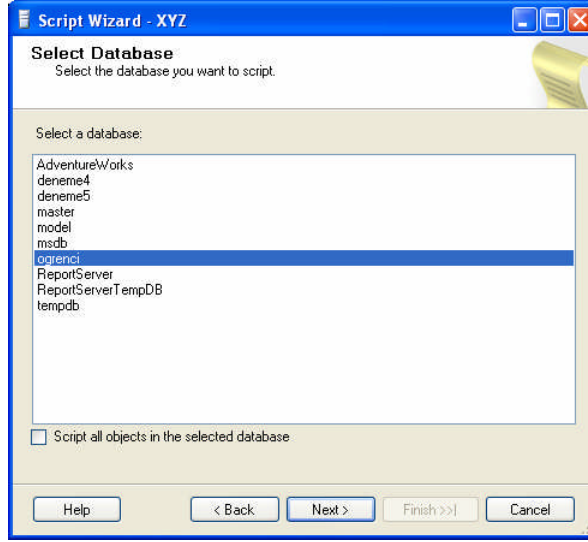
Resim 1.23: Generate Scripts komutunun verilmesi

Bu komutla Script sihirbazı ekrana gelecektir.



Resim 1.24: Script Wizard ekranı

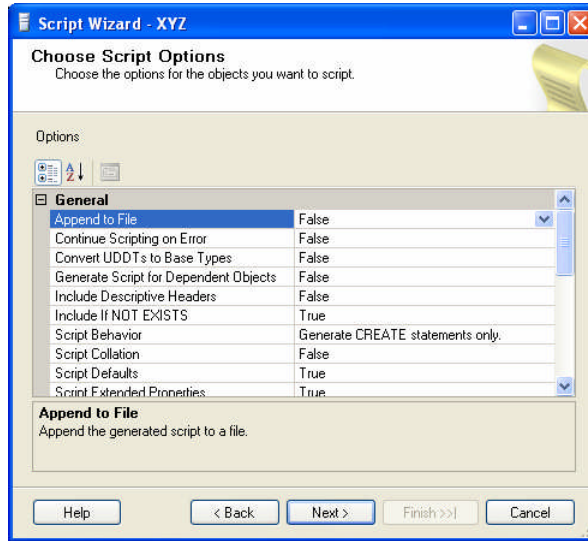
Açılış ekranı geldikten sonra Next düğmesine tıklanır. T-SQL ifadesi alınacak veri tabanının seçilmesi için Resim 1.25 ekrana gelir.



Resim 1.25: Veri tabanı seçim ekranı

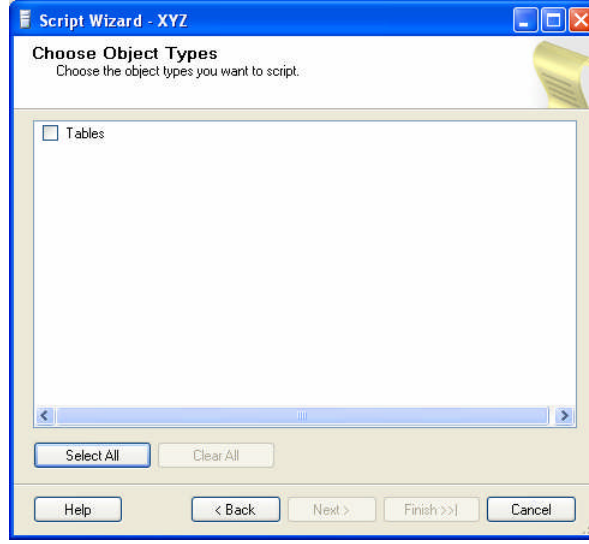
Veri tabanı seçilip Next düğmesine tıklanmalıdır.

Nesnelerin nasıl script edileceğinin belirlendiği pencere ekrana gelecektir.



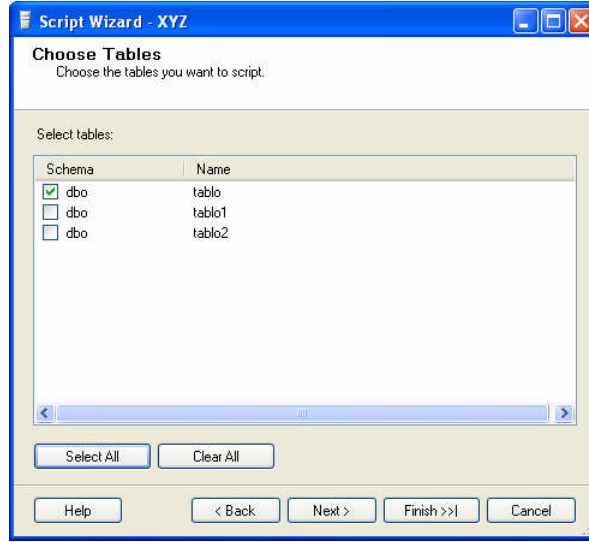
Resim 1.26: Script Options ekranı

Bu pencerede script özelliklerinin ayarlamaları yapılabilir. Next düğmesi ile ekrana nesne seçim ekranı gelecektir.



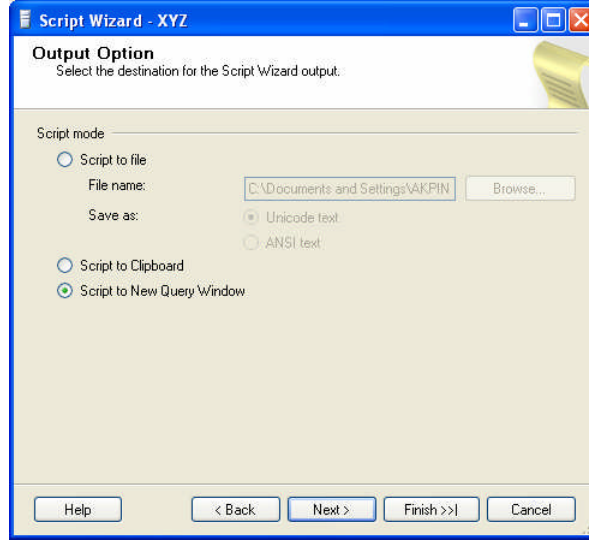
Resim 1.27: Nesne seçim ekranı

Bu projede sadece tablolar olduğu için Tables seçimi vardır. Seçimi işaretleyip Next düğmesine tıklandığında tablo seçim ekranı karşınıza gelecektir.



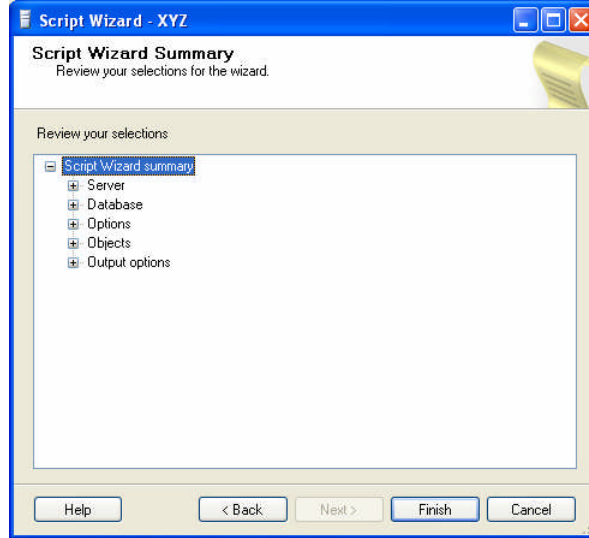
Resim 1.28: Tablo seçimi

İfadesi alınacak tablo seçimi yapıлып Next düğmesine tıklanmalıdır. Çıkış seçeneği ekranı gelecektir.



Resim 1.29: Çıkış seçeneği ekranı

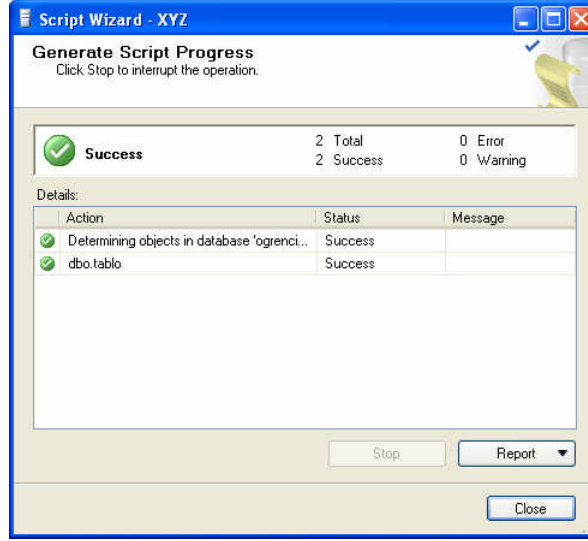
Elde edilecek verilerin nereye alınacağı bu ekranda sorulur. Yeni bir sorgu ekranı seçimini yapabilirsiniz. İlgili seçimi yapıp Next düğmesine tıklayınız. Ekranı özet (Summary) ekran penceresi gelecektir.



Resim 1.30: Özet (Summary) ekranı

Hiçbir düzenleme yapmadan bu ekranda Finish düğmesine basınız.

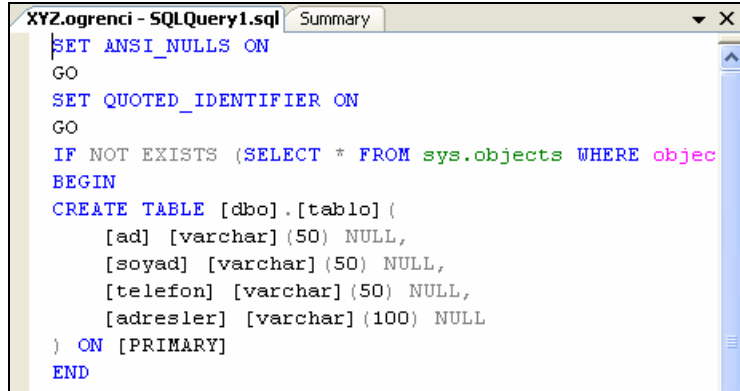
Script alma işlemini gösteren bir ekran karşınıza gelir.



Resim 1.31: Script alma işlemi sonucunun gösterilmesi

Close düğmesine tıklayarak işlemi bitiriniz.

Böylece T-SQL ifadeleri yeni bir sorgu dosyasına alınmış olacaktır.



Resim 1.32: T-SQL ifadelerinin bir dosyaya alınmış hâli

1.6. Veri Türleri

Veri türleri, numara, resim, tarih gibi bir sütunda veya bir değişkende depolanabilen türdeki bilgiyi belirtir. SQL Server 28 "sistem tanımlı" veri türünü tanır. Bu veri türleri haricinde, belirli ihtiyaçları karşılayabilmeniz için kullanıcı tanımlı veri türlerini T-SQL ve .NET ile oluşturabilirsiniz.

- Sistem tanımlı veri türleri kategorileri şunlardır.
 - Karakter dizilimi (Character strings)
 - Unicode karakter dizilimi (Unicode character strings)
 - Tarih ve saat (Date and time)
 - Tahmini sayısal (Approximate numeric)
 - Kesin sayısal (Exact numeric)
 - Tam sayılar (Integer numbers)
 - Parasal (Monetary)
 - İkili (Binary)
 - Özel (Special)

Bazı durumlarda, bir veri türünü T-SQL kodda ilişkilendirmek için farklı tanımlayıcılar/tanıtıcılar kullanabilirsiniz. Örneğin; Char veri türü Karakter olarak, varchar ise karakter çeşitliliği olarak ele alınabilir. Bu eş anlamlı sözcüklerin bazıları ANSI SQL-92 standart gereksinimlerine dayanmaktadır.

1.6.1. Karakter Dizilimi

Karakter veri türleri karakter dizilimlerini içerir. Dört farklı karakter türü, uzunlukları ve depolama özellikleri açısından çeşitlilik gösterir.

- char
- varchar
- varchar(max)
- text

char veri türü, sabit boyuttaki dizilimleri depolamak için kullanılır. Daha önce belirtildiği gibi, bu veri türünün maksimum uzunluğu/boyutu 8000 karakterdir ki bu, eski versiyonlardaki 255 karakter limiti bağlamında kayda değer bir artıştır.

Örnek

```
TCKimlikNo char(11)
```

varchar veri türü, 8000 karakter uzunluğa kadar olan çeşitli boyutlarda dizilimleri depolar. Uzunluğu nominal boyuttan az olan karakter değeri bir sütuna veya bir değişkene atandığında SQL Server peşinden boşluklar eklemek yerine onu olduğu gibi kaydeder. Varchar veri türleri dizilimin uzunluğunu kaydetmek için 2 ilave byte kaplar.

Örnek

```
ad varchar(50)
```

varchar (max), varchar veri türü gibi görünür ve işler ancak dâhili yapı ve işlevselliği daha çok metin veri türü gibidir. Aynı zamanda 2 GB'a kadar olan geniş dizilimleri depolamak için tasarlanmıştır. Dizilim, kaydın geri kalanıyla veri sayfasında değil özel veri

sayfalarında depolanır. Bunun en büyük avantajı varchar ile çalışan çoğu dizilim operasyon fonksiyonunun aynı zamanda varchar (max) ile de çalışmasıdır; bununla birlikte ilave işlevsellik için birkaç yeni fonksiyon bulunmaktadır.

Örnek

Yeni_deger varchar(max))

text veri türü, büyük miktarlardaki veriyi depolamak için kullanılır. Tek bir alan 2GB (231 - 1 byte) bilgiye kadar depolayabilir. Sadece 16 byte tabloda depolanır. Bu nedenle ilave işlem, metin sütunlarının kullanımıyla ilişkilendirilir. Metin değerleri işlemi için bazı özel fonksiyonlar bulunmaktadır.

Örnek

durum text

1.6.2. Unicode Karakter Dizilimi

Evrensel Kod standardının kullanılması ve onu destekleyen yazılımların bulunması son zamanlardaki küresel yazılım teknolojilerinin en popüler konusudur.

Evrensel Kod'un, istemci-sunucu ya da çok katmanlı uygulamalar ve web sitelerinde kullanımı önemli bir düzeyde tasarruf sağlamaktadır. Evrensel Kod tek bir yazılımın ya da bir web sitesinin yeniden tasarlanmasına ihtiyaç duyulmadan her türlü dil ve ülke için uyumlu hâle getirilmesini sağlar. Aynı zamanda verinin çeşitli işletim sistemleri içerisinde aktarımını da sorunsuz hâle getirmektedir.

Microsoft SQL Server 2005 non-ASCII (ASCII karakteri olmayan) karakter setleri kullanan Unicode veriyi depolamak için dört karakter veri türüne sahiptir. Bunlar char, varchar, varchar (max) ve metin veri türlerine eşdeğerdir ve aşağıdaki gibi adlandırılır:

- nchar
- nvarchar
- nvarchar(max)
- ntext

Bu yeni veri türleri ile eski veri türleri arasındaki fark; yeni veri türlerinin karakter başına 2 byte kaplayan Unicode karakterlerini tutabilmeleridir. Bu nedenle, depolayabildikleri maksimum dizilim uzunluğu, eski veri türlerindeki yarısıdır (4000).

Örnek

TCKimlikNo nchar(11)

Örnek

ad nvarchar(50)

Örnek

Yeni_deger nvarchar(max))

Örnek

durum ntext

1.6.3. Tarih ve Saat Veri Türleri

SQL Server tarih ve saati depolamak için iki T-SQL veri türünü destekler.

- Datetime
- Smalldatetime

Bu iki veri türü arasındaki fark kapladıkları alandır. datetime 8 byte, smalldatetime sadece 4 byte kullanır.

Aralarındaki diğer farklar ise depolanan tarihin doğruluğu/kesinliği ve kullanılacak tarihlerin genişliğidir. Smalldatetime'in doğruluğu bir dakikadır ve 1 Ocak 1900'den 6 Haziran 2079'a kadar olan zamanı kapsar ki bu da yeterinden fazladır. Datetime'in doğruluğu 3.33 milisaniyedir ve 1 Ocak 1753'ten 31 Aralık 9999'a kadar olan süreyi kapsar.

1.6.4. Tahmini Sayısal Veri Türü

Ondalık sayılar genellikle tek veya çift kesinlik (single and double precision) olarak da bilinen real ve float veri türlerinde depolanır. Fazla yer kaplamamaları ve geniş aralıkta sayıları tutabilmeleri avantajdır. Tek sorun şudur ki; kati/kesin değildir. Yaklaşık olarak orijinal ondalık sayıya eş değer yaklaşık/tahmini ama kesin olmayan sayının çift değişkenli temsilini depolar.

Kesinlik/doğruluk (Precision) numaradaki önemli/karakteristik basamak/hanelerin sayısı ve Scale ondalık noktanın sağa doğru uzanan basamak/hanelerin sayısıdır.

Örneğin; 123456.789 sayısının 9 doğruluğu ve 3 ondalığı vardır. Gerçek sayıların basamakları 7'ye kadardır ve float sayıların kesinliği 15 basamağa kadardır. Bu nedenle, Dünya ve Ay arasındaki mesafeyi ölçerken birkaç metrenin önemsenmeyeceği bilim ve mühendislik için idealdirler. Fakat şirket bütçesinin son kuruşuna kadar kesin olması gereken finansal endüstri için yeterli değildir.

1.6.5. Kesin Sayısal Veri Türü

Ondalık veya sayısal veri türleri sayıları depolarken yaklaşıklık/tahmini kullanmazlar. Ne yazık ki gerçek ve float veri türlerine göre daha fazla yer/alan gerektirirler. Ondalık bir sütun veya değişken tanımlandığında/belirtildiğinde basamağını ve doğruluğunu (scale and precision) belirtmelisiniz. SQL Server maximum 38 doğruluktaki sayıları depolayabilir. Basamak (scale) doğruluktan daha az ya da onunla eşit olabilir.

1.6.6. Tam Sayı Veri Türü

SQL Server 1-, 2-, 4-, ve 8-byte tam sayıları destekler. Bit veri türü mantıksal doğru ve yanlış değerleri temsil eden 1 ve 0 depolamak için kullanılır. Aşağıdaki tablo tam sayı veri türlerini, depolama boyutlarını ve değerlerin aralığını listeler.

Veri Türü	Boyutu	Minimum	Maximum
int	4 bytes	-2, 147, 483, 648 (-2G)	2, 147, 483, 647(26?1)
smallint	2 bytes	-32768 (-32Ks)	32767 (32K-1)
tinyint	1 bytes	0	255(28-1)
bigint	8 bytes	-9,223,372,036,854,775,808 (-263)	9,223,372,036,854,775,807 (263-1)
bit	1 bit	0	1

int veri türleriyle ilgili en güzel şey küçük alanda oldukça fazla sayı depolayabilmesidir. Bu nedenle sık sık anahtar değerler (key values) için kullanılır. Ana anahtarın veri türü int ise tablo dört milyara kadar kayıt depolayabilir ki miktar herhangi bir amaç için oldukça yeterlidir. Bu nedenle tablonuzun boyutundan emin değilseniz ana anahtar olarak int kullanınız.

1.6.7. Parasal Veri Türü

Tam sayıya ondalık ifadesi eklenmiş veri türleridir. Duyarlılık ve ölçekleme değiştirilmez. SQL Server, verileri bir tam sayı gibi ele alır. Parasal sabitlerden önce \$ veya diğer 18 para sembolünden biri gelebilir (SQL Server Books OnLine'da listelenmişlerdir).

Veri Türü	Boyutu	Sınırı
smallmoney	4 bytes	-214 bin ile +214 bin arasında değerler için
money	8 bytes	-922 milyar ile +922 milyar arasında değerler için kullanılır.

1.6.8. Binary (İkili) Veri Türü

Binary veri tipleri bitlerin diziliminin yerleştirilmesi için kullanılır. SQL Server dört temel çift değişkenli veri tipini, karakter veri türü ile benzer nitelikleri desteklemektedir.

- binary
- varbinary
- varbinary(max)
- image

binary and varbinary veri türleri 8000 byte bilgi ve resim, varbinary (max) 2 GB veri depolayabilir.

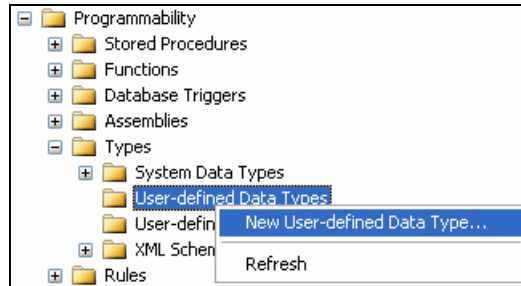
1.6.9. Özel Veri Türleri

Belirtilen veri türleri dışında bir gruba dâhil olmayan veri türleri vardır.

Veri Türü	Açıklama
Cursor	Tabloların yapısında yer almayan bir tür olan Cursor, kayıtlara bir sonuç kümesi (resultset) yerine, erişim için kullanılan bir değişken türüdür.
Table	table veri türü sonraki süreçlerde recordset yerleştirmek için kullanılır. Bazı açılardan bu veri türü geçici tablo (temporary table) ile benzerdir. Bu veri türünü sütun tanımlamak için kullanamazsınız. Sadece fonksiyonun değerini geri döndürmek için local değişken olarak kullanılabilir.
Timestamp	Satır sürüm numarası olarak tanımlı bir türdür. Ortak zamanlı çalışma için, düzenleyici unsur olarak kullanılır.
Uniqueidentifier	Her çalışmada benzeri olmayan bir 16 byte'lık 16'lı tabanda sayı üreten sayı türüdür. Çok geniş bir ölçümde tekil değerler elde etmek için kullanılır.

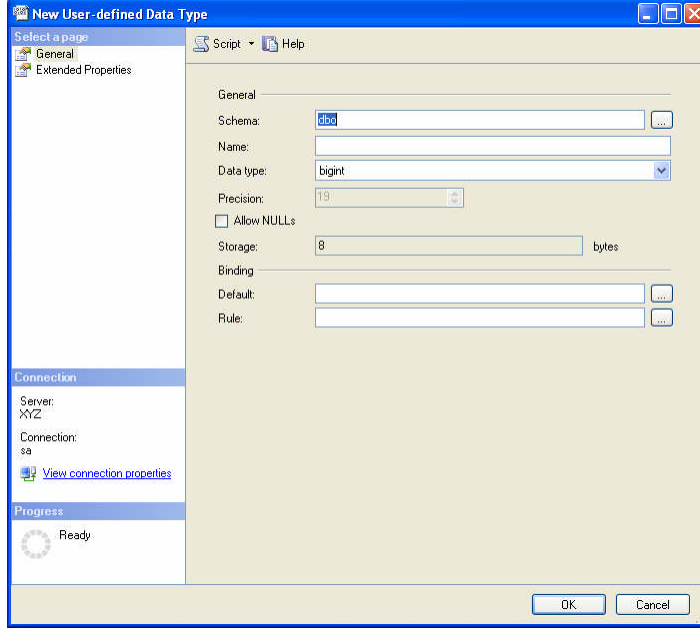
1.6.10. Kullanıcı Tanımlı Veri Türleri

Kullanıcı tanımlı veri türü tanımlamanın iki yöntemi vardır. Management Studio'da veri tabanı açılıp User Defined Types altında sağ tıklayıp açılan menüden New User Defined Type'ı seçerek veya *CREATE TYPE* ifadesi ile kullanıcı tanımlı veri türü oluşturulabilir.



Resim 1.33: Kullanıcı tanımlı veri türü oluşturma

Komut verildiğinde ekrana New User defined Data Type penceresi ekrana gelir.



Resim 1.34: Yeni bir kullanıcı tanımlı veri türü oluşturma penceresi

Gerekli düzenlemeleri yapıp OK düğmesine tıkladığınızda kullanıcı tanımlı veri türü oluşturmuş olacaksınız.

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
➤ Management Studio'yu kullanarak bir veri tabanı oluşturunuz.	➤ Object Explorer'daki Databases üzerinde fareyle sağ tıklayarak New Database komutunu verebilir ve veri tabanı ismini de Uygulama1 olarak verebilirsiniz.
➤ Veri tabanınızda bir tablo oluşturunuz.	➤ Tablo sütun adları olarak TCKimlikNo, Ad, Soyad olarak belirleyiniz. Sütun veri türleri için uygun veri türlerini seçiniz.
➤ Tabloyu kaydediniz.	➤ Araç çubuğundaki Save düğmesini tıklayarak Tablo1 olarak kaydedebilirsiniz.
➤ Tablonuzda bir sütun daha ekleyiniz.	➤ Tables'ın alt seçeneği Columns üzerinde fareyle sağ tıklayarak açılan menüden New Column'u seçebilirsiniz.
➤ Yeni sütun adını giriniz.	➤ VergiNo olarak girebilir ve veri türü olarak da uygun bir tür seçebilirsiniz.
➤ VergiNo sütununun veri türünü varchar veri türü olarak değiştiriniz.	➤ Columns seçiminde sütun adı üzerinde sağ tıklayarak Modify komutunu verebilirsiniz.
➤ Veri tabanınızda ikinci bir tablo oluşturunuz ve sütun adlarını yazınız.	➤ Öneri 2-3-4'teki adımları izleyebilirsiniz.
➤ Tablo1'deki TCKimlikNo sütununu siliniz.	➤ Columns seçiminde gösterilen sütun adının üzerinde sağ tıklayarak Delete komutunu verebilirsiniz.
➤ Tablo1'i siliniz.	➤ Tables seçiminde gösterilen tablo adının üzerinde sağ tıklayarak Delete komutunu verebilirsiniz.
➤ Veri tabanının T-SQL ifadesini alınız.	➤ Object Explorer'da veri tabanı adı üzerinde sağ tıklayarak açılan menüden Tasks ve Generate Scripts komutunu verebilirsiniz. Sihirbaz yardımıyla scripti alabilirsiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru/yanlış seçenekli sorularda uygun harfleri yuvarlak içine alınız. Seçenekli sorularda ise uygun şıkkı işaretleyiniz. Boşluk doldurmalı sorularda boşluklara uygun cevapları yazınız.

1. Düzenli bilgi topluluğuna denir.
2. T-SQL ile veri tabanı oluşturmak için komutu kullanılır.
3. Veri tabanında mutlaka birincil dosya vardır (D/Y).
4. Veri tabanının otomatik olarak küçültülmesini sağlayan özellik aşağıdakilerden hangisidir?
A) Auto Create
B) Auto Shrink
C) Restrict Access
D) Collation
5. Aşağıdakilerden hangisi ilişkisel veri tabanı nesnesi değildir?
A) Saklı yordamlar
B) Görünüm
C) Diyagram
D) Form
6. Management Studio'da yeni bir tablo oluşturmak için Object Explorer penceresinde \..... komutu seçilir.
7. Management Studio'da tabloya yeni bir sütun eklemek için ADD komutu kullanılır. (D/Y)
8. Management Studio'da sütun özelliklerini değiştirmek için komutu kullanılır.
9. Nesnelerin T-SQL ifadelerini almak için Management Studio'da veri tabanı üzerine sağ tıklanıp açılan menüden \..... komutu seçilir..
10. 2 GB'akadar olan geniş dizilimleri depolamak için hangi veri türü kullanılır?
A) char
B) text
C) varchar(max)
D) varchar

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları geri dönerek tekrar inceleyiniz. Tüm sorulara doğru cevap verdiğinizde diğer modüle geçiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Veri tabanı planlaması üzerinde çalışabilecek ve mantıksal veri modeli oluşturabileceksiniz.

ARAŞTIRMA

- Daha önce öğrenmiş olduğunuz veri tabanı ile programlama modüllerine de dayanarak basit bir planlama nasıl yapılmalıdır?

2. PLANLAMA

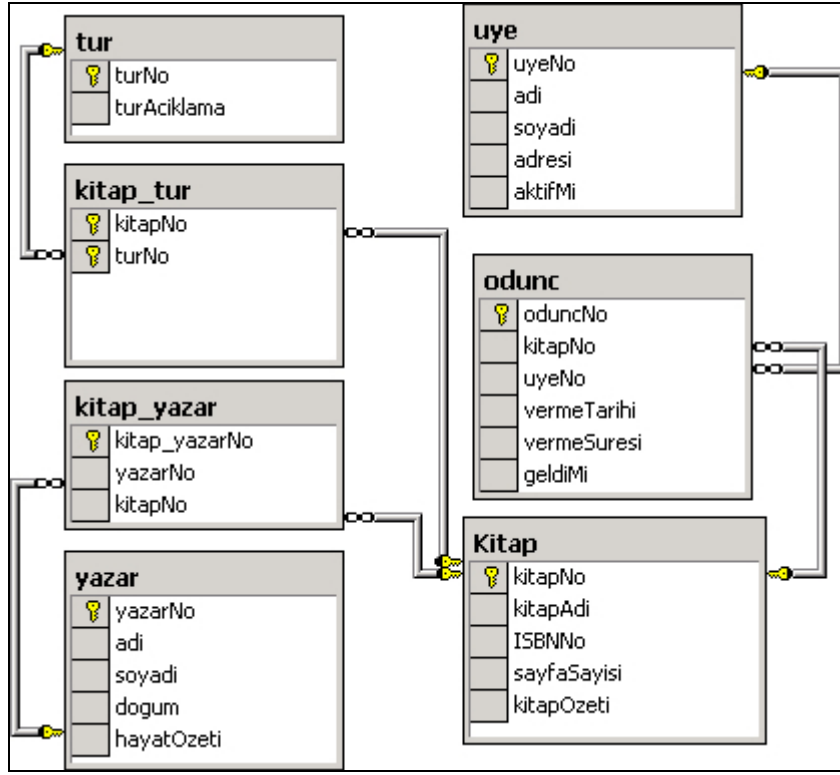
2.1. Veri Tabanı Normalizasyonu

Bir tablo bir tek varlığa ait bilgileri tutmak için tasarlanır. Örneğin bir kitap tablosunda sadece kitaplara ait bilgiler vardır. 20 adet kitap bilgisi varsa, bu tabloya 20 adet satır eklenecek demektir. İlişkisel veri tabanı yaklaşımını ilişkisel yapan asıl unsur verilerin tablolara parçalanarak saklanmasıdır.

Tabloların kaç adet olacağı ve birbiriyle nasıl ilişkilendirileceğine karar verirken var olan kuralları kullanarak mantıksal bir plan elde etmelisiniz.

Eğer tablolar üzerinde verileri normalize etmezseniz birçok sıkıntı oluşabilir. Bazı bilgiler birden fazla tekrarlanabilir. Bir bilgiyi güncellemek, veri eklemek ve silmek için birden fazla yerde bulma, değiştirme ve silme yapmanız gerekebilir. Bu nedenle normalizasyon kurallarını kullanmalısınız.

Genel kabul gören beş normalizasyon kuralı vardır. Bu kurallar, ilişkisel veri tabanının tanımı ile birlikte ortaya konmuştur. Veri tabanı normalizasyonu Resim 2.1'e bağlı olarak anlatılacaktır. Açıklamalar yapılırken bu resme dikkat etmelisiniz.



Resim 2.1: Örnek veri tabanı normalizasyonu

➤ 1. Normalizasyon Kuralı

Bir satırdaki bir alan yalnızca bir tek bilgi içerebilir.

Örneğin kitap tablosunda, birden fazla yazarı olan kitap için yazar1, yazar2, yazar3 diye alanlar açarsanız, bu kurala uymamış olursunuz. Böyle bir durumda, ayrıca yazarlar tablosu da oluşturularak kuralı çiğnememeye dikkat etmelisiniz.

➤ 2. Normalizasyon Kuralı

Bir tablo için, anahtar olmayan her alan, birincil anahtar olarak tanımlı tüm alanlara bağlı olmak zorundadır.

Mesela, Ödünç tablosuna KitapAdi diye bir alan ekleydik, bu sadece ödünç verilen kitap ile ilgili bir bilgi olacaktı ve oduncNo'na bağlı bir nitelik olmayacaktı. Bunu çözmek için, kitap adlarını ayrı bir tabloda tutarak sorun çözülür.

Ya da anahtar alanın birden fazla alandan oluştuğu tablolarda, anahtar alanlardan sadece birine bağlı veriler, tabloda yer almamalı, ayrı bir tabloya taşınmalıdır.

Bunun tersi de geçerlidir. Yani iki ya da daha fazla tablonun birincil anahtarı aynı olamaz. Şayet böyle ise, bu iki tablo tek tabloya indirilmelidir.

➤ 3. Normalizasyon Kuralı

Bir tablo için, anahtarı olmayan bir alan, anahtarı olmayan başka hiç bir alana bağlı olamaz.

Örneğin, kitaplarımız için cilt tipi adında bir alan ekleyip burada da karton kapak için K, deri cilt için D, spiral cilt için S yazsaydık, bu kodlama, kitap tablosunun birincil anahtarı olan kitapNo alanına bağlı bir kodlama olmayacaktı. Çünkü bu kodlama bir başka anahtarı olmayan alana bağlıdır. Bunun sonucunda da veri tabanımızda, karşılığı olmayan bir kodlama yer almış olacaktır. Cilt tipi bilgisini kodlu olarak tutan alan aslında cilt tipi açıklaması olan başka bir alana bağlıdır. Bu ilişki başka bir tabloda tutulmalıdır.

Bu durumda, ciltŞekli adında bir tablo açmamız gerekir. Bu tablonun alanları da ciltTipKodu ve ciltSekli olmalıdır. Ancak bundan sonra, kitaplar tablosunda ciltTipi adında bir sütun açıp buraya da D,S,K gibi kodları yazabilirsiniz..

➤ 4. Normalizasyon Kuralı

Birincil anahtar alanlar ile anahtarı olmayan alanlar arasında, birden fazla bağımsız bire-çok ilişkisine izin verilmez.

Örneğin, tablomuzda yer alan bir kitap hem hikaye kitabı hem de kişisel gelişim kitabı olabilir.

4.normalizasyon kuralını sağlamak için, her bağımsız bire-çok ilişki için ayrı bir tablo oluşturmak gerekir.

Bu örnekte, türler diye bir tablo açmamız ve daha sonra da kitapTurleri diye bir başka tablo daha açmamız gerekiyor. “Kişisel Gelişim Hikâyeleri” adlı kitap için, öncelikle kitap numarası, Hikâye bölümünün kodunun yer aldığı bir satır; ardından da yine kitap numarası, ardından da kişisel gelişim türünün kodunun aldığı yeni bir satır daha eklemek gerekir.

➤ 5. Normalizasyon Kuralı

Tekrarlamaları ortadan kaldırmak için her bir tabloyu mümkün olduğunca küçük parçalara bölmek gerekir. Aslında ilk 4 kural sonuçta bu işe yarar ancak, bu kurallar kapsamında olmayan tekrarlamalar da 5.normalizasyon kuralı ile giderilir.

Örneğin, kitaplarımız için bir edinme şekli bilgisi girilecek sütun eklemek isteyelim: Bu bölüme girilebilecek bilgiler bellidir: Bağış veya satın alma gibi.

Bu bilgileri başka bir tabloda tutabiliriz. Böylece, kullanıcıların bu alana gelişigüzel bilgiler girmesini engellemiş olursunuz. Bu da sorgulama esnasında verileriniz arasında bir tutarlılık sağlar. Bu işlem sonucunda, tutarsızlıklara neden olabilecek ve sık tekrarlayan

veriler başka bir tabloya taşınmış olur. Bu tablo için, veri tabanı programlamada “look-up table” terimi kullanılır.

Ancak, veri tabanı normalizasyon kuralları, bir ilişkisel veri tabanının tasarlanma aşamalarını değil de ilişkisel veri tabanında yer alacak kayıtların ilişkisel veri tabanı ile uyumlu olup olmadığını denetlemeye yöneliktir.

Özetle ilişkisel bir veri tabanı tasarımı şu dört ögeyi barındırmalıdır.

- Veri tekrarı yapılmamalıdır.
- Boş yer mümkün olduğunca az olmalıdır.
- Veri bütünlüğü sağlanmalıdır.
- Veriler, aralarında bir ilişki tanımlamaya müsait olmalıdır.

2.2. Kayıtların İlişkilendirilmesi

İlişkiler tablolardan verileri birbirine bağlayarak tabloların kullanılabilirliğini artırır. Böylece ilişkisel veri tabanları oluşturulur. Veriler konu veya göreve göre çeşitli tablolarda depolanır, ancak veriler ilişkilidir ve belirttiğiniz şekilde bir araya getirilebilir.

Tablolar arasında ilişki kurmak suretiyle, veri tabanı dosyası düz veri tabanı olmaktan kurtarılır. Tablolar arası ilişki, daha hızlı sorgu icrası yapılmasını sağlar; veri-işlem işleri daha verimli olur. Bu sayede verileri güncelleştirirken, bütün tabloları elden geçirmek ve hatta yeniden girmek zorunda kalınmaz. Sadece güncelleştirilmesi gereken tabloya ek veya bu tabloda değişiklik yapılır.

İlişkilendirme temelde bir tablodaki varlığa ait bilgilerin anahtar olan sütundan hareketle başka bir tablodan takip edilmesini sağlar. Tablolar arasında üç tip ilişki vardır.

2.2.1. Bire-Bir İlişki

İlişkilendirdiğiniz iki tablodan birincisi ile ikincisi arasında sadece bir kayıt eşleşebilir. Bu tür ilişkilere bire-bir ilişki denir. İki tablo arasında bire bir ilişki varsa bu iki tablo birbirinin devamı demektir ve genellikle bir tek tablo olarak birleştirilir. Bu tür ilişkilendirme tablolarda sık kullanılmaz.

2.2.2. Bir-Çok İlişki

En çok kullanılan ilişki şeklidir. Bu ilişkide A tablosundaki bir kayıt B tablosundaki birden çok eşleşen kayda sahiptir. Bunun tersi doğru değildir. Yani B tablosundaki bir kayıt A tablosundaki yalnızca bir kayıt ile eşleşir.

Örneğin, **Müşteriler** ve **Siparişler** tablolarının birçok ilişkisi vardır: Her müşteri birçok sipariş verir ancak her sipariş bir müşteriye aittir.

2.2.3. Çok-Çok İlişki

Bir çok-çok ilişkide, A tablosundaki bir satırın B tablosunda çok sayıda eşi vardır ve tam tersi de geçerlidir. Bu tür ilişkileri, birincil anahtarları A ve B tablolarının yabancı anahtarlarından oluşan bağlantı tablosu diye adlandırılan üçüncü bir tablo tanımlayarak oluşturabilirsiniz.

2.3. Dosya Grupları (File Groups)

SQL Server, dosyaları daha kolay yönetebilmek için dosya gruplarını kullanır. Bir veri tabanı ilk oluşturulduğunda otomatik olarak Default adında bir dosya grubu oluşturur. Grup oluşturulması geniş veri tabanlarında yedekleme işlemlerinin daha kısa zamanda ve daha az disk kapasitesi kullanılarak yapılmasını sağlamaktadır. Bu dosya grubu, sistem için gerekli bileşenleri de içerdiğinden Primary File Group olarak anılabilir.

Veri dosyalarının farklı disklerde oluşturulması ile performans artışı sağlanabilir. Dosya, tablo ve indekslerin farklı disklerde tutulması ile SQL Server aynı zaman diliminde farklı indekslerden okuma yapabilir.

İkinci veri dosyası da (Secondary Data Files) bu gruba eklenebileceği gibi yeni kullanıcı açılarak da ikinci veri dosyası oluşturulabilir. İkinci veri dosyasının, bölümlendirilmiş bir diskte oluşturulması performans artışı sağlayacaktır.

Transaction log (işlem günlüğü) dosyaları herhangi bir dosya grubu (file group) dâhilinde değildir. Log alanı, veri alanından ayrı olarak yönetilir.

2.3.1. Dosya Gruplarında (File Groups) Dikkat Edilmesi Gereken Hususlar

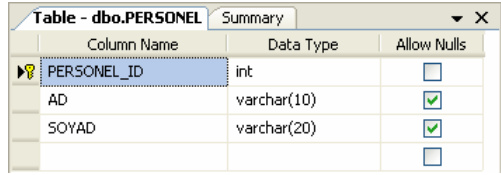
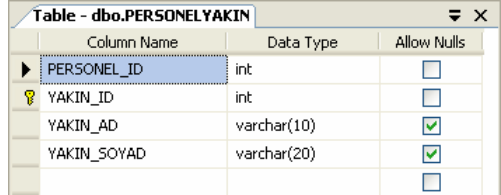
- Birinci veri dosyası (primary data file) birinci dosya grubu (primary file group) üzerinde yerleşmiş olmalıdır.
- Bütün sistem dosyaları birincil dosya grubu (primary file group) üzerinde yerleşmelidir.
- Herhangi bir dosya, birden fazla dosya grubuna (file group) aynı anda üye olmamalıdır.
- Dizinler (indexes), tablolar (tables), text, ntext ve image verileri dosya grupları (file groups) üzerinde tutulmaz.
- Dosya gruplarında (file group) boyutlandırma çok kritik bir durumdur. Geçerli dosya grubu (file group) bütün tabloları ve sistem dosyalarını tutacağından dolayı, hepsini tutabilecek kapasitede olmalıdır. Yeterli alana sahip olmayan bir dosya grubu (file group), veri tabanına yeni bilgi eklenememesine sebep olur. Otomatik büyüme seçeneği işaretlenmemişse, belirli bir süre sonra, dosya grubu (file group) dolar. Bu durum kullanımı etkilediği için, otomatik büyüme seçeneğinin işaretli olması tavsiye edilir.

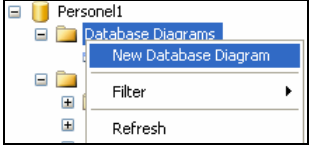
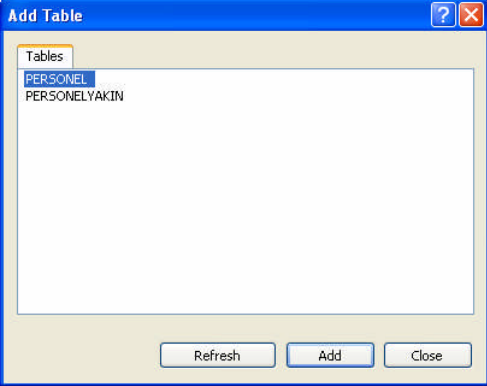
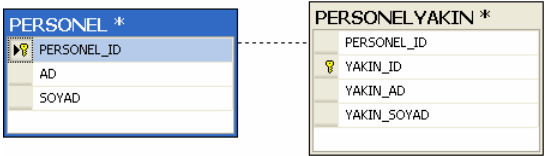
2.3.2. Dosya Gruplarının (File Groups) Faydaları

Dosya grupları (file group) kullanmak ileri seviyede veri tabanı tasarlama yöntemidir. Küçük ölçekli veri tabanları için değil de, geniş çaplı bir veri tabanı üzerinde uygulandığında, herhangi bir yedekleme ve geri yükleme (restore) işlemlerinde performansa katkısı oldukça fazladır. Bunun en küçük örneği, veri tabanının tamamını yedeklemek veya geri yüklemek yerine, dosya grupları (file groups) ve veri dosyaları (data files) yedeklenir. Bu sayede performans sağlanır.

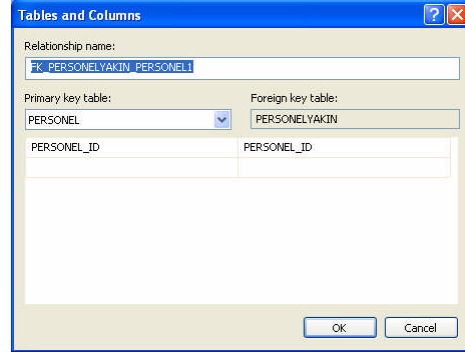
Dosya gruplarını (file group) yönetmek için, veri tabanını, içerdiği veri türlerini, içerdiği veriyi, kullanılan sorguları çok iyi kavramak gerekmektedir.

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"> ➤ Management Studio'yu kullanarak bir veri tabanı oluşturunuz. 	<ul style="list-style-type: none"> ➤ Object Explorer'daki Databases üzerinde fareyle sağ tıklayarak New Database komutunu verebilir ve veri tabanı ismini de Personel1 olarak verebilirsiniz.
<ul style="list-style-type: none"> ➤ Veri tabanınızda bir tablo oluşturunuz. 	<ul style="list-style-type: none"> ➤ Tablo sütun adları olarak PERSONEL_ID, AD, SOYAD olarak belirleyiniz. Sütun veri türleri için uygun veri türlerini seçiniz.
<ul style="list-style-type: none"> ➤ Sütun veri türlerini belirleyiniz. 	<ul style="list-style-type: none"> ➤ PERSONEL_ID için int, AD için varchar(10), SOYAD için varchar(20) veri türlerini kullanınız.  <p style="text-align: center;">Resim 2.2: PERSONEL tablosu</p>
<ul style="list-style-type: none"> ➤ Tabloyu kaydediniz. 	<ul style="list-style-type: none"> ➤ Araç çubuğundaki Save düğmesini tıklayarak PERSONEL olarak kaydedebilirsiniz.
<ul style="list-style-type: none"> ➤ Veri tabanınızda bir tablo daha oluşturunuz. 	<ul style="list-style-type: none"> ➤ Tablo sütun adları olarak PERSONEL_ID, YAKIN_ID, YAKINAD, YAKINSOYAD olarak belirleyiniz. Sütun veri türleri için uygun veri türlerini seçiniz.
<ul style="list-style-type: none"> ➤ Sütun veri türlerini belirleyiniz. 	<ul style="list-style-type: none"> ➤ PERSONEL_ID için int, YAKIN_ID için int, YAKINAD için varchar(10), YAKINSOYAD için varchar(20) veri türlerini kullanınız.  <p style="text-align: center;">Resim 2.3: PERSONELYAKIN tablosu</p>
<ul style="list-style-type: none"> ➤ Tabloların birincil anahtarlarını belirleyiniz. 	<ul style="list-style-type: none"> ➤ PERSONEL tablosu için PERSONEL_ID, PERSONELYAKIN tablosu için YAKIN_ID

	<p>sütunlarını birincil anahtar olarak belirleyebilirsiniz.</p>
<p>➤ Tablolar arasında bir ilişki kurmak için bir diyagram oluşturunuz.</p>	<p>➤ Database Diagrams seçiminde iken sağ tıklayarak açılan menüden New Database Diagrams komutunu veriniz.</p>  <p>Resim 2.4: New Database Diagrams komutu</p>
<p>➤ Tabloları diyagrama ekleyiniz.</p>	<p>➤ Tabloları seçip Add komut düğmesine tıklayabilirsiniz.</p>  <p>Resim 2.5: Add Table penceresi</p> <p>➤ Close komut düğmesiyle Add Table penceresini kapatabilirsiniz.</p>
<p>➤ Tablolar arasındaki ilişki için PERSONEL tablosundaki PERSONEL_ID ile PERSONELYAKIN tablosundaki PERSONEL_ID sütunlarını ilişkilendiriniz.</p>	<p>➤ PERSONEL_ID üzerinde fareyle bir kere tıklayarak bırakmadan diğer tablodaki PERSONEL_ID sütunu üzerine bırakabilirsiniz.</p>  <p>Resim 2.6: Tabloların ilişkilendirilmesi</p>

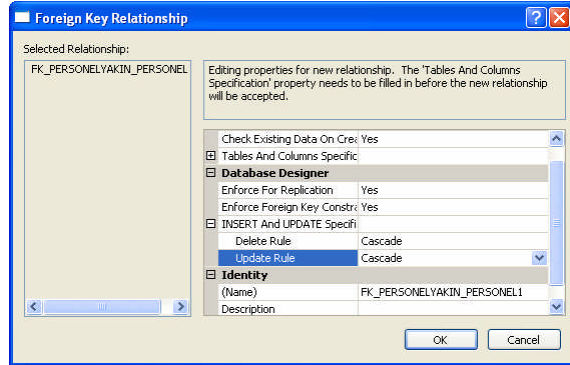
- İlişki isimlendirmesini yapınız.



Resim 2.7: İlişki isminin belirlenmesi

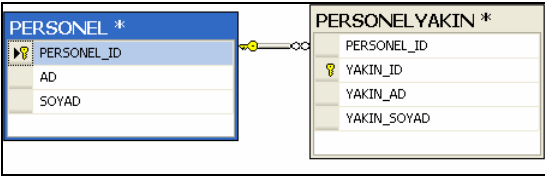
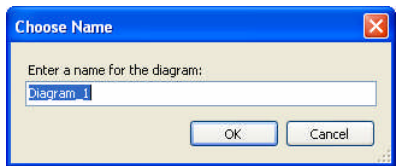
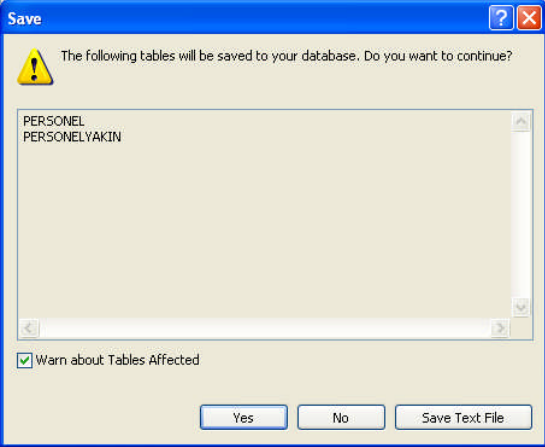
- İlişki ismi otomatik olarak yazılacağından OK düğmesine tıklayarak isimlendirmeyi bitirebilirsiniz.

- Yabancı anahtar (foreign key) belirlenmesini yapınız.



Resim 2.8: Yabancı anahtarın belirlenmesi

- Yabancı anahtarla N tane tabloyla bağlantı yapılabilir.
- Ana kaydın silinmesi hâlinde detay kayıtların da silinmesi için Delete Rule seçeneğini Cascade yapabilirsiniz.
- Ana kaydın bağlantı alanlarının değişmesi hâlinde detay kayıtların da bağlantı alanlarının otomatik değişmesi için Update Rule seçeneğini Cascade yapabilirsiniz.
- OK düğmesine tıklayarak işlemi bitirebilirsiniz.

<p>➤ İlişkilendirmeyi görünüz.</p>	 <p>Resim 2.9: İlişkinin görülmesi</p>
<p>➤ Oluşturduğunuz diyagramı kaydediniz.</p>	<p>➤ Diyagramı kaydetmek için Save düğmesine tıklayınız.</p>  <p>Resim 2.10: Diyagramın kaydedilmesi</p>
<p>➤ Veri tabanınız için kaydedilecek tabloların gösterimini onaylayınız.</p>	 <p>Resim 2.11: Tabloların kaydedilmesi</p>

Not: İki tablo arasında kurulan ilişkileri görüntüleyebilmemiz için View nesnesi yazabilirsiniz. View nesnesi hakkındaki bilgilere ilerleyen konularda değinilecektir.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru/yanlış seçenekli sorularda uygun harfleri yuvarlak içine alınız. Seçenekli sorularda ise uygun şıkkı işaretleyiniz. Boşluk doldurmalı sorularda boşluklara uygun cevapları yazınız.

1. SQL Server'a ait servisleri, ağ ve istemci erişim ayarlarını düzenlemeye yarayan programa SQL Server yönetim programı (SQL Server Configuration Manager) denir. (D/Y)
2. Solution Explorer (proje yönetim penceresi) sorguların yazıldığı penceredir. (D/Y)
3. Bir sorguyu çalıştırmak için komutu kullanılır.
4. Sorgunun doğruluğunu denetlemek için komutu kullanılır.
5. Surface Configuration aracı, SQL Server'ın özelliklerini, servislerini veya bağlantı özelliklerini aktifleştirip pasifleştirmek için kullanılır. (D/Y)
6. Birden fazla SQL Server tek bir bilgisayara kurulamaz. (D/Y)
7. Her bir SQL Server kurulumuna.....(.....) denir.
8. Kullanıcı tanımlarının ve temel bilgilerin yer aldığı veri tabanı aşağıdakilerden hangisidir?
 - A) Master veri tabanı
 - B) Model veri tabanı
 - C) Msdb veri tabanı
 - D) Resource veri tabanı
9. SQL Server, veri tabanını diske sayfalar (page) halinde kaydeder. Ar arda kayıtlı sekiz sayfaya denir.
10. SQL Query Analyzer'da yeni bir kullanıcı oluşturmak için CREATE USER deyimini kullanılır. (D/Y)

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konulara geri dönerek tekrar inceleyiniz. Tüm sorulara doğru cevap verdiyseniz diğer öğrenme faaliyetine geçiniz.

MODÜL DEĞERLENDİRME

PERFORMANS TESTİ (YETERLİK ÖLÇME)

Modül ile kazandığınız yeterlik, aşağıdaki işlem basamaklarına göre değerlendirilecektir.

Değerlendirme Ölçütleri	Evet	Hayır
1. Management Studio'yu açabildiniz mi?		
2. Management Studio'da veri tabanı oluşturabildiniz mi?		
3. Management Studio'da tablo oluşturabildiniz mi?		
4. Tabloyu kaydedebildiniz mi?		
5. Tabloya sütun ekleyebildiniz mi?		
6. Sütun veri türlerini belirleyebildiniz mi?		
7. Yeni bir sütun adını girebildiniz mi?		
8. Sütunun veri türünü değiştirebildiniz mi?		
9. Yeni bir tablo ekleyebildiniz mi?		
10. Sütunu silebildiniz mi?		
11. Tabloyu silebildiniz mi?		
12. Veri tabanının T-SQL ifadesini alabildiniz mi?		
13. Birincil anahtarları belirleyebildiniz mi ?		
14. Tablolar arası ilişki kurabildiniz mi?		
15. Diyagram oluşturabildiniz mi?		
16. Tabloları diyagrama ekleyebildiniz mi?		
17. Yabancı anahtarları belirleyebildiniz mi?		
18. İlişkilendirmeyi görüntüleyebildiniz mi?		
19. Diyagramı kaydedebildiniz mi?		
20. Tabloların gösterimini onaylayabildiniz mi?		

DEĞERLENDİRME

Yaptığınız değerlendirme sonucunda eksikleriniz varsa öğrenme faaliyetlerini tekrarlayınız.

Modülü tamamladınız, tebrik ederiz. Öğretmeniniz size çeşitli ölçme araçları uygulayacaktır, öğretmeninizle iletişime geçiniz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1 CEVAP ANAHTARI

1	Veri tabanı
2	CREATE DATABASE
3	Doğru
4	B şıkkı
5	D şıkkı
6	Tables\New Table
7	Yanlış
8	Modify
9	Tasks\Generate Scripts
10	C şıkkı

ÖĞRENME FAALİYETİ-2 CEVAP ANAHTARI

1	Doğru
2	Yanlış
3	Execute
4	Parse
5	Doğru
6	Yanlış
7	Instance(örnek)
8	A şıkkı
9	Extend
10	Doğru

KAYNAKÇA

- GÖZÜDELİ Yaşar, **Yazılımcılar İçin SQL Server 2005 ve Veri Tabanı Programlama**, Seçkin Yayıncılık, Ankara, 2006.
- GÜRKAN Osman, **Microsoft Access XP**, Nirvana Yayınları, Ankara, 2005.
- YURTSEVER İsmail, **Microsoft SQL Server 2005 Stored Procedure Programming in T-SQL & .NET, Third Edition by Dejan Junderi** (çeviri).
- www.microsoft.com/sql
- www.sqlnedir.com
- www.verivizyon.com
- www.yazilimuzmani.com