

HİTİT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

Veri Tabanı Yönetim
Sistemleri

TSQL İLE VERİ TABANI VE TABLOLARI OLUŞTURMAK, ÖZELLİKLERİNİ BELİRLEMEK

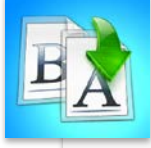


Atatürk Üniversitesi
Açıköğretim Fakültesi

VERİ TABANI YÖNETİM SİSTEMLERİ

Dr. Öğr. Üyesi
Serdar AYDIN

İÇİNDEKİLER



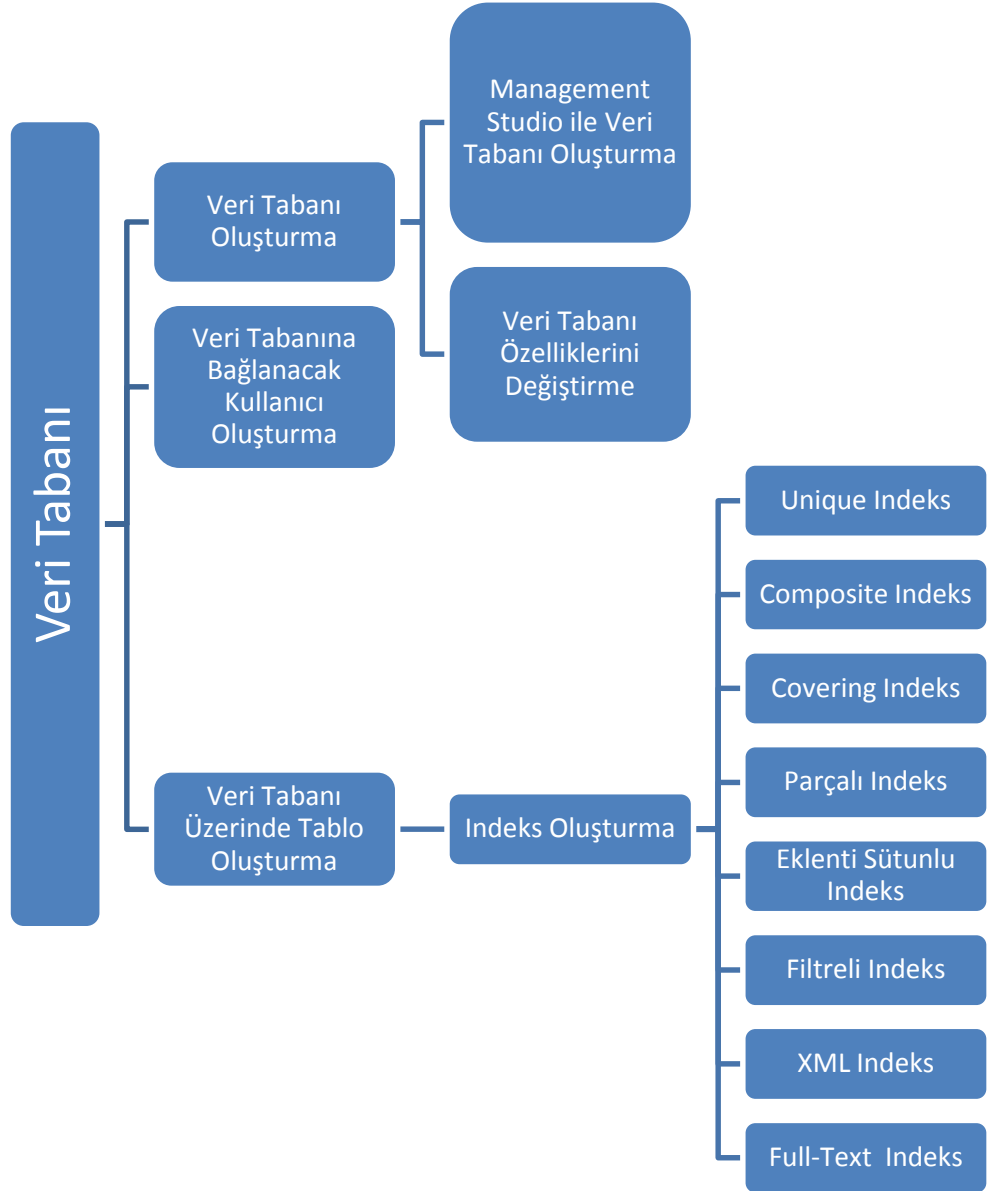
- SQL Server 2008 R2 Ortamında Tablo Oluşturmak
- TSQL ile Veri tabanı oluşturmak
- TSQL ile Tablo oluşturmak
- TSQL ile Tablo düzenlemek
- TSQL ile Tablo silmek

HEDEFLER



- Bu üniteyi çalıştıktan sonra;
- SQL Server 2008 R2 üzerinde TSQL ile tablo oluşturabilecek,
- Tablolara çeşitli veri türlerinde alanlar tanımlayabilecek,
- Arama sihirbazı kullanarak veri türü tanımlamanın nasıl gerçekleştirildiğini kavrayabileceksiniz.

ÜNİTE 5



GİRİŞ

Bu bölümde T-SQL ile Veri tabanı oluşturma, tablo oluşturma, tablo düzenleme silme gibi işlemlerin nasıl yapılabileceğine bakacağız. TSQL Microsoft'un veri tabanı sorgulama dilidir. Transact-SQL, SQL Server ve istemci (client) arasında iletişimi sağlayan SQL sorgulama dilinin gelişmiş bir versiyonudur. SQL yapılandırılmış sorgulama dili (Structured Query Language) anlamına gelen Transact Structured Query Language kelimesinin kısaltmasıdır. T-SQL kullanarak veri tabanına kayıt eklenebilir, silinebilir, güncellenebilir; kullanıcı, eklenebilir, silinebilir, güncellenebilir ya da sorgulama ve raporlama yapılabilir. TSQL bir programlama dili değil sorgu dilidir.

T-SQL ile aşağıdaki işlemler yapılabilir:

- Veri Tabanı oluşturmak,
- Tablo oluşturmak,
- Veri tabanı ve tablolar üzerinde değişiklikler yapmak,
- Kayıt ekleme, silme, güncelleme,
- Verileri Filtrelemek.

T-SQL ile döngü veya mantıksal işlemler yapmak için bir derleyiciye gerek yoktur. Herhangi bir programlama dili öğrenmeden de T-SQL ile tüm amaçlarınıza hitap edecek projeler gerçekleştirebilirsiniz. T-SQL ifadelerini çalıştırabilmek için bir Management Studio ile SQL Server'a erişmeniz gerekir.

TSQL ile Management Studio'ya ihtiyaç duymaksızın herhangi bir SQL istemcisi ile SQL Server üzerinde bütün işlemleri yapabilirsiniz.

SQL deyimleri veri tabanları üzerinde çeşitli işlemleri yerine getirir. Veri tabanından sorgulama yapmak için SELECT, ekleme yapmak için INSERT güncelleme yapmak için UPDATE, silme yapmak için DELETE, yeni tablo oluşturmak için CREATE TABLE gibi komutlara sahiptir. Bu komutlar işlevlerine göre şu şekilde ayrılır.



TSQL ile Management Studio'ya ihtiyaç duymaksızın herhangi bir SQL istemcisi ile SQL Server üzerinde bütün işlemleri yapabilirsiniz.

T-SQL ile Veri Tabanı Oluşturmak

Veri tabanını Management Studio ile oluşturabildiğimiz gibi T-SQL ile de oluşturabiliriz.

Bunun için en basit şekilde,

```
CREATE DATABASE veri_tabanı_ismi
```

komutunu kullanabiliriz. Bu komutla veri tabanı oluşturduğumuzda model veri tabanı kopyalanacak ve bu kopyanın ismi verdiğiniz isimle değiştirilerek veri tabanı oluşturulacaktır. Veri tabanımızı özelliklerini kendimiz belirterek oluşturmak istediğimizde ise;

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

T-SQL komutunu kullanmamız gerekir. Bu komutu inceleyelim:

Primary (Birincil) dosya veri tabanı için gerekli olan tüm nesnelere saklandığı dosyadır. Veri tabanı için birincil dosya mutlaka olmalıdır.

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 sistemince bilinen isimdir.
- **Fiziki_dosya_ismi:** İşletim sisteminde saklanacak dosyanın adını ve yolunu belirtir. Bu isimde işletim sistemince 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_miktari:** Dosyanın başlangıçta belirtilen boyutu dolduğunda sistem tarafından boyutu otomatik olarak artırılabilir. Bu aşamada veri tabanının boyunun ne kadar artırılacağı bu parametre ile belirtilir. % oran verilebileceği gibi sabit bir artım boyutu da girilebilir.

Örneğimizde çok fazla veri değişikliği olacağı için veri dosyası ile log dosyası arasındaki oran geleneklere göre üst sınır olan %25'te tutulmuştur.

Log dosyası farklı bir sürücüde olacak şekilde tanımlanmıştır. Bu özellikle performans artırıcı olsun diye yapılmış bir ayarlamadır. Transaction log dosyası ve tempdb'ye ait dosyalar için ayrı birer disk ayarlamak I/O işlemini artırdığı için veri değişikliği ekleme ve silme yapan veri tabanında performansı artıracaktır.



Management Studio'nun aksine TSQL ile veri tabanı oluştururken tüm parametreleri bilmeniz ve eksiksiz girmeniz gerekmektedir.

Veri Tabanına Erişecek Uygulama Kullanıcısını Ayarlamak

Tasarladığımız bu veri tabanına elbette üst katmandan erişilerek C#, VB.NET, Java, Delphi gibi dillerle programlar yazılacak. Peki bu diller içerisinde veri tabanımıza nasıl erişilecek? Elbette ki bir uygulama rolü veya kullanıcı üstünden.

Bu aşamaya kadar veri tabanını oluşturdu iseniz hemen arkasından, veri tabanına erişecek login ve kullanıcıları oluşturmanız gerekir. Aşağıdaki ifade ile *webuser* adında bir veri tabanı uygulama erişimi için login ve veri tabanı kullanıcı oluşturabilirsiniz:

```
USE master GO
CREATE LOGIN webuser WITH PASSWORD = 'sifre'
GO
USE ogrencibilgi GO
CREATE USER webuser FOR LOGIN webuser
```

Aynı ayarlamaları, SQL Management Studio kullanarak SQL Server'a bağlandıktan sonra Nesne gezgini (Object Explorer) üstünden de önce Security düğümüne gidip yeni bir login ardından da sırayla Databases, ogrencibilgi veri tabanı, Security düğümünü tıklayıp yeni veri tabanı kullanıcı oluşturabilirsiniz. (Ünite 4)

Bütün nesnelerin şemasını (sahibini) tek tek değiştirmek gerekiyordu. Bu probleme, “işten ayrılan eleman sorunu” denmektedir. SQL Server, herhangi bir kullanıcı ile herhangi bir şemanın ilişkilendirilebilmesine olanak sağlar. Böylece şemalar gerçek anlamda nesnelere gruplama işlevini yerine getirir.

Şemaları Yönetmek

Bir veri tabanı içerisinde şema oluşturmak için şu genel ifade kullanılabilir:

```
CREATE SCHEMA schema ismi; (varsa şema içerisine eklenecek öğeler)
```



Şema yapısı veri tabanı içerisindeki nesnelere erişim izni ve sahiplik atamaları gibi işlevleri kolaylaştırmak için gruplamaya yarar.

Örnek



- ogrencibilgi veri tabanı içerisinde katalog adında bir şema oluşturalım ve içerisine bir tablo ekleyelim:
- CREATE SCHEMA ogrencibilgisema
- CREATE TABLE [dbo].[Birey](
 - [ID] [int] IDENTITY(1,1) NOT NULL,
 - [TcNo] [varchar](11) NOT NULL,
 - [Adi] [varchar](50) NULL,
 - [Soyadi] [varchar](50) NULL,
 - [AnneAdi] [varchar](50) NULL,
 - [BabaAdi] [varchar](50) NULL,
 - [Cinsiyeti] [varchar](5) NULL,
- CONSTRAINT [PK_Ogren] PRIMARY KEY CLUSTERED
- [ID] ASC
- WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
- ON [PRIMARY]
- Daha sonra, bu şemayı bir kullanıcının default şeması olarak atamak istersek şu genel ifadeden yararlanabiliriz:
- ALTER USER kullanıcı ismi
- WITH DEFAULT_SCHEMA = sema ismi

Ardından bir önceki ünite de öğrendiğimiz şekilde webuser kullanıcıasına ogrencibilgisema adlı şema içerisindeki veri tabanı öğelerinden seçme yapma yetkisi verelim:

```
GRANT SELECT ON SCHEMA::ogrencibilgisema TO webuser
```

Örnek



- Webuser isimli kullanıcının default şemasını katalog olarak değiştirelim:
- ALTER USER webuser
- WITH DEFAULT_SCHEMA = ogrencibilgisema

Şema tasarımı yapılırken şu kurallar göz önünde bulundurulmalıdır:

- Bir şemanın birden fazla kullanıcı sahibi olabilir. (Windows rolü, veri tabanı kullanıcısı, veri tabanı uygulama rolü, Ahmet, yerine işe başlayan Ali...)
- Birden fazla kullanıcı aynı default şema tanımını ile ilişkilendirilmiş olabir

- Ortak kullanılan nesnelerin sahibi dbo olmak zorunda değildir. Bütün kullanıcıların erişebileceği başka bir genel şema oluşturulup bu amaç için kullanılabilir.
- Bir kullanıcıyı sildiğinizde bu kullanıcıya ait bütün nesnelere silme veya sahibini değiştirme zorunluluğunu ortadan kaldırır.

Veri Tabanını Yönetmek

Bir veri tabanı oluşturulduktan sonra bazı özelliklerinin zaman içerisinde yeniden ayarlanması gerekebilir. Veri tabanı seçeneklerinden erişim ve nesnelerin çalışma seçenekleri gibi ayarlar yapıldır. Bazı durumlarda veri tabanı dosyalarının boyutunu azaltmak veya artırmak ihtiyacı doğabilir. Aynı şekilde log dosyalarının boyutları ile ilgili ayarlamalar yapmak gerekebilir.



Veri tabanını düzenlemek için ALTER DATABASE komutu kullanılır.

Veri Tabanı Seçeneklerini Ayarlamak

Veri tabanı seçenekleri Management Studio'dan bir veri tabanı sağ tıklanıp özellikler menüsüne geçildikten sonra, options sekmesinden ayarlanabileceği gibi, ALTER DATABASE deyimini ile ayarlanabilir veya bir veri tabanı ilk oluşturulduğu anda burada verilen parametreler de tayin edilebilir.

Genel olarak ALTER DATABASE deyimini, veri tabanı seçeneklerini ayarlarken şu şekilde kullanılır:

```
ALTER DATABASE veritabani ismi  
SET secenek durum
```

Hâlihazırdaki veri tabanı seçenek grubu için, parametrelerden hangilerinin geçerli olduğu şu şekilde de öğrenilebilir:

```
SELECT DATABASEPROPERTYEX ('veritabani ismi','ozellik')
```

Tablo 5.1. Veri Tabanı Seçenekleri

Seçenek Grubu	Seçenek veya	Durum
Otomatik (automatic)	AUTO_SHRINK	Veri tabanının otomatik olarak küçültülmesini ayarlayan özelliktir. Bu özellik seçili iken veri tabanı dosyası ve log dosyası belli aralıklarla sistem tarafından otomatik olarak gereksiz alanları (%25'den fazlası boş olursa) dosyadan
Otomatik	AUTO_CLOSE	Bütün kullanıcılar veri tabanından bağlantıyı kestiğinde, veri tabanı sunucusu otomatik olarak dosyaları bırakır ve doğrudan işletim sistemi ortamında müdahale edilebilir hâle gelir. Canlı uygulamalarda bu seçenek kapalı olmalıdır.
Otomatik	AUTO_CREATE_STATISTICS ve AUTO_UPDATE_STATISTICS	SOL Server WHERE ve JOIN ifadeleri için sorgu süresi-maliyeti kestirirken kullanılmak üzere istatistikler tutar ve bu istatistikleri UPDATE eder. Bu seçeneklerin canlı bir sistem için açık olması performansı artırır.

Durum (state)	RESTRICTED USER	Bu özellik seçildiği anda sistemde login olan kullanıcılar ve db_owner sabit sistem rolüne haiz kullanıcılar dışında diğer kullanıcılar bu veri tabanı dosyasını kullanamaz. Hâlihazırda sisteme login olan kullanıcılar ise sistemden çıkmaları halinde db_owner rolüne sahip değil ise tekrardan veri tabanına erişemez. Yedek indirmek gibi tamir işlemleri esnasında, bu modda çalıştırmak bir zorunluluk olabilir. Veri tabanı dosyasını aynı anda bir tek kişinin kullanmasını sağlar.
Durum	SINGLE USER	Bu komut çalıştırıldığı anda veri tabanında birden fazla kişi bağlıysa bu özellik açıldıktan sonra kullanmaya devam ederler ancak bağlantı kesildiği andan itibaren bir tek kullanıcı kuralı dikkate alınır.
Durum	MULTI USER	Yedek indirmek gibi tamir işlemleri esnasında, Normal olarak bir veri tabanı bu erişim modundadır. Herkes tarafından hakları dâhilinde erişilebilir.
Durum	READ ONLY READ WRITE	Veritabanı dosyasının sadece okunabilir olmasını sağlayan seçenektir. Bu seçenek seçildiğinde kullanıcılar veri tabanından veri okuma (SELECT) komutlarını hakları doğrultusunda çalıştırabilir ancak DELETE, UPDATE, INSERT gibi DML komutları ile ALTER, CREATE ve DROP gibi DDL komutlarını çalıştıramaz. Tersi READ_WRITE modudur. Bu modda herkes yetkileri dahilinde okur ve yazar. Bir durum belirtilmeden kullanılırlar. Bu seçenekte veri tabanı üstünde yapılan işlemlerin ayrıntılı logları tutulur. Bir veri kurtarma işleminde, log dosyaları ile birlikte veri tabanı kayıtları da elde edildikten sonra veri tabanı kurtarma işlemi gerçekleştirilebilir. Standart olan kurtarma seçeneği bu seçenektir. Böylece yarıda kalmış transactionlar bile Full Recovery Model ile yaklaşık aynı seçenekleri kullanır. Farklı olarak toplu işlemler için daha az log tutar. Örneğin SELECT INTO vb. gibi komutlar için daha kısıtlı bir log tutulur.
Kurtarma (Recovery)	RECOVERY FULL	Bir tablonun bir sütunu için NULL olabilirlik konusunda
Kurtarma	BULK LOGGED	bir tanımlama yapılmazsa server nasıl davranacak? ON: belirtilmedi ise NULL kabul edebilir OFF: belirtilmedi ise NULL olamaz. NULL olabilecek sütunlar ayrıca belirtmeli
ANSI	ANSI_NULL _ DEFAULT	NULL değerler içeren sütunların bir eşitlik sorgulamasında, eşittir mi yoksa eşit değildir mi olarak kabul edileceğine karar verir.
ANSI	ANSI_NULLS	NULL kıyas sonuçlar için OFF: doğru, ON:yanlış

Bu şekilde durumları kontrol edilebilen daha bir çok parametre vardır. Diğer seçenekler hakkında, bu şekilde bilgi almak için, Books Online'da **DATABASEPROPERTYEX** ifadesini aratarak bilgi edinebilirsiniz.

Öğrenci bilgi veri tabanında, gereksiz dosya boyutlarının otomatik olarak küçültülmesini istiyoruz.



Örnek

- ALTER DATABASE öğrencibilgi SET AUTO_SHRINK ON
- Daha sonra ilgili değişikliğin yapıp yapılmadığını görüntülemek için
- SELECT DATABASEPROPERTYEX('dukkân', 'IsAutoShrink')

diyerek durumunu görebiliriz.

ogrencibilgi veri tabanında, bir süre için kimsenin değişiklik yapamamasını istiyoruz.



Örnek

- ALTER DATABASE ogrencibilgi
- SET READ_ONLY



Veri tabanını sadece okunabilir moda çalıştırabilirsiniz.

şeklinde kullanılır.

Veri Tabanı Seviyeli Collation Ayarı Yapmak

SQL Server, veri tabanı seviyeli collation ayarı yapmaya müsaade eder.

Özellikle İngilizce bir Windows sürümü üstünde çalışan ve default olarak İngilizce ayarlanmış bir SQL Server, uygulamalarınızda karakter problemine neden olabilir. Bu türden durumlarda, aşağıdaki şekilde collation ayarı yapabilirsiniz:

```
USE master
GO
ALTER DATABASE dukkan COLLATE Turkish_CI_AS
```

Bu türden bir ayarlamayı çalıştırabilmeniz için, ogrencibilgi veri tabanı içerisinde collation ayarlarına bağlı bulunan nesne olmaması gerekir.

Daha sonra ilgili değişikliği görmek için

```
sp_helpdb 'ogrencibilgi'
```

İfadesini çalıştırarak göz atabilirsiniz.

Yine SQL Server'in İngilizce sistem üstünde kurulu olduğu durumlarda, tarih formatı ile ilgili problemlerin önüne geçmek üzere (MDY yerine DMY), uygulama kullanıcınız için şu ifadeyi de çalıştırabilirsiniz:

```
sp_defaultlanguage 'webuser','Turkish'
```

Böylece, bu kullanıcı tarafından yapılan tarih değişimlerinde MDY formatı yerine DMY formatını kabul eder hâle gelecektir. Veritabanına erişen başka Türkçe dil ayarlı kullanıcılar bulunması gerekiyorsa aynı ayarları *webuser* yerine onların adlarını da vererek yapabilirsiniz.

Veri Tabanı Dosyalarının Boyutunu Değiştirmek

Veri tabanında yer alan veri dosyalarının veya Log dosyalarının boyutunu zaman içerisinde artırmak gerekebilir veya yüksek bir alan verildiği için küçültün



Seviyeli Collation ayarları ile tarih formatı para birimi gibi formatların SQL tarafından otomatik düzenlenmesini sağlar.

gerekebilir. Bir veri tabanı üstünde bu türden ayarlamaları yaparken, *sysadmin* rolüne sahip olmak veya veri tabanı sahiplik rolüne sahip olmak gerekir.

Veri Tabanı Dosyalarının Boyut Artımını Ayarlamak

Bir veri tabanın boyutu küçük geliyorsa var olan veri dosyasının veya log dosyasının boyutunu artırarak çözüm olabileceği gibi, başka bir veri dosyası veya log dosyası eklemek de çözüm olarak kullanılabilir. Özellikle disk dolmuşsa başvurulacak çözüm, yeni bir dosya tanımlamaktır.

Otomatik artırmaya ayarlama

Bu seçenek, veri tabanının tanımında yapılması gereken FILEGROWTH parametresi ile belirtilir. Daha önceden oluşturulmuş bir veri tabanı için bu seçeneği şu genel ifadeden yola çıkarak düzenleyebiliriz:

```
ALTER DATABASE veritabani ismi
MODIFY FILE (NAME=dosya takma ismi,
SIZE = dosya boyutu,
MAXSIZE = maksimum dosya boyutu,
FILEGROWTH = dosya artim miktarı)
```

Veri Tabanlarını Küçültmek (Shrinking)

Veri tabanlarını yönetirken daha önceden ayrılan yerin çok fazla olduğuna karar verdiğimizde, henüz veri tabanı tarafından kullanılmamakta olan alandan vazgeçebiliriz. Başka bir şekilde ifade etmek gerekirse veri tabanının veya dosyalarının boyutunu, veri tabanının asgari kullanımına kadar düşürebiliriz.

Örneğin veri tabanımızın boyutu 40MB olarak tanımlansın. İçerdiği veriler de 4MB ise bu veri tabanının boyutu 4 MB'a kadar düşürülebilir. Ancak daha fazla düşürülemez.

Veri tabanının veya dosyalarının boyutunu değiştirmek, master veri tabanındaki bilgilerin değişmesini gerektirir. Bu nedenle herhangi bir hata ihtimaline karşılık, veri tabanının boyutu ile ilgili değişikliklerden önce ve sonra master veri tabanının yedeğini almak, herhangi bir riske karşı tavsiye edilir.

Bir veri tabanının boyutunu azaltmak için iki yöntem vardır, otomatik olarak veri tabanının boyutunun küçültülmesi veya herhangi bir anda, elle küçültmek. Bu işi otomatik olarak SQL Server'e yaptırmak için, AUTO_SHRINK veri tabanı özelliği

```
ALTER DATABASE veritabani ismi
SET AUTO_SHRINK ON
```

ifadesi kullanılarak ayarlanabilir veya Management Studio'dan, veri tabanı özellikleri ekranından düzenlenebilir.

Herhangi bir anda, veri tabanının boyutu ile ilgili küçültmeye gidilecekse ya doğrudan veri tabanının boyutu küçültülebilir veya veri ya da log dosyalarında



Veri tabanlarını yönetirken daha önceden ayrılan yerin çok fazla olduğuna karar verdiğimizde, henüz veri tabanı tarafından kullanılmamakta olan alandan vazgeçebiliriz.

birinin boyutunu küçültmek gerekebilir. Bu iki işlem için iki ayrı ifade kullanılır:

DBCC SHRINKFILE, veri tabanına ait bir dosyanın boyutunu küçültür. Bu ifade de dosyaların takma adlarını kullanarak erişir. Bir dosyanın boyutunun uygun olan en küçük değere indirilmesi için, genel ifade şu şekildedir:

```
DBCC SHRINKFILE(dosya adi, TRUNCATEONLY)
```

Boşalan alanın tamamının işletim sistemine devredilmesi işlemi yerine getiren bu ifadede, TRUNCATEONLY deyimi isteğe bağlıdır. Bir dosyanın belli bir orana kadar boşaltılması ve boşalan alanın da veri tabanı bünyesinde tutulmaya devam edilmesi için şu genel yapı kullanılır:

```
DBCC SHRINKFILE(dosya adi, yuzde kaca inecek, NOTRUNCATE)
```

Bir dosya içerisinde yer alan verilerin, aynı dosya grubunda yer alan diğer dosyalara aktarılmak suretiyle içinin boşaltılması için, EMPTYFILE seçeneği ile kullanılabilir.

```
DBCC SHRINKFILE(dosya adi, EMPTYFILE)
```

DBCC SHRINKDATABASE: Bir veri tabanının boyutu küçültülür. Ancak bu işlem, bir job olarak ele alındığından, veri tabanının boyutunun küçüldüğünü görmek zaman alabilir. Genel kullanımı şu şekildedir:

```
DBCC SHRINKDATABASE (veritabani ismi,TRUNCATEONLY)
```

Bu ifadede, bir veri tabanı adı verildiğinde kullanılmayan son extend'den başlamak üzere fazladan atanmış olan alan işletim sistemine devredilir. Burada TRUNCATE ONLY isteğe bağlı bir ifadedir. Hiç kullanılmaya da bilir.

Veri tabanından boşaltılan alanın, işletim sistemine devredilmeden, veri tabanı kapsamında tutulmaya devam edilmesi için de şu genel ifade kullanılabilir:

```
DBCC SHRINKDATABASE(veritabani ismi, yuzde kaca inecek, NOTRUNCATE)
```

Burada bir hedef yüzde oranı verilebildiğine dikkat edin.

Veri Tabanlarını Silmek

Pek gerekmeseyse de bazen veri tabanlarını silmek de bir ihtiyaç olabilir. Bu durumda, veri tabanını Management Studio'dan veya DROP DATABASE deyimi ile silebiliriz. Ancak bir veri tabanını silebilmek için bazı şartların sağlanması gerekir:

- Veri tabanının, yedek yükleme, replication, kullanıcı erişimi gibi başka bir işlem tarafından kilitlenmemiş olması gerekir.
- Sizin veri tabanında db_owner veya sysadmin rollerinden birine sahip kullanıcı ile erişilebilir olmanız gerekir.



Veri tabanlarının kullanmadığı alanları sisteme bırakmasını sağlayarak etkin disk alanı kullanımı sağlanabilir.

Bir veri tabanı silindiği zaman, içerdiği bütün veriler kaybolur. Veri tabanı ile birlikte alınmış yedeklerini de silebilirsiniz veya alınmış yedeklerini bırakabilirsiniz. Bu size bir seçenek olarak sunulmuştur.

Bütün bunlardan sonra, veri tabanı silmek için genel ifade şu şekildedir:

```
DROP DATABASE veritabani ismi
```

Tabanı kapsamında tutulmaya devam edilmesi için de şu genel ifade kullanılabilir: şeklinde kullanılır.



- ogrencibilgi adlı veritabanını sileceksek:
- DROP DATABASE ogrencibilgi

Tabloları Oluşturmak

SQL Server'da iki şekilde tablo oluşturulabilir: Management Studio'da veri tabanı sağ tıklanarak açılan menüden yeni tablo seçeneği işaretlenebilir veya sorgu ekranında CREATE TABLE ifadesi kullanılarak oluşturulabilir. SQL Server₁ bir tablo için binlerce sütuna müsaade eder. Ayrıca SQL Server hafızayı page'ler

hâlinde yönettiğinden ve her page için 8060 byte kullanılabilir alan kaldığından, bir tablonun bir satır için sütun uzunlukları toplamının 8060byte'i geçmemesi gerekir. Bu uzunluğa ntext, text ve image alanlar dâhil değildir. Çünkü bu türden veriler BLOB olarak adlandırılır ve farklı yapılarla saklanır. MAX parametresi ile tanımlanan VARCHAR, CHAR, BINARY, VARBINARY tipler için SQL Server LOB veya tablo içerisinde saklama konusunda otomatik ayarlama yapabilir. SQL Server bir veri tabanında iki milyondan fazla tablo saklayabilir.

Bir tablo T-SQL ile şu format kullanılarak oluşturulur:

```
CREATE TABLE tabloAdi  
(kolon ad1 veri tipi[NOT NULL],  
kolon ad2 veri tipi [NOT NULL],  
..... )
```

Tablolar Üstünde Değişiklik Yapmak

Genel olarak tablo üstünde değişiklik yapmak için ALTER TABLE deyimini kullanılır.

Sütun eklemek

Bir tabloya sütun eklemek için şu genel yapıdan faydalanılır:



Tablolar oluşturulurken sütünlarda taşınacak verinin türüne göre veri tipinin seçilmesi önemlidir.



ALTER Table komutu ile tablo üzerinde düzenleme yapılabilir.

```
ALTER TABLE tablo ismi  
ADD COLUMN sutun ismi sutun özellikleri
```

sutun_ozellikleri parametresi ile veri tipi, gerek varsa, uzunluğu ve sütuna dair daha sonradan öğreneceğimiz tanımlamalar yer alabilir.



Örnek

- Birey Tablosuna Doğum Yılı adında bir sütun ekleyelim:
- ALTER TABLE [Birey]
- ADD DogumYili AS varchar(2) NULL

Sütun değiştirmek

Bir tablonun sütunun türünü veya genişliğini değiştirmek için ALTER COLUMN ifadesi kullanılabilir.

```
ALTER TABLE tablo ismi  
ALTER COLUMN sutun ismi sutun tanımları
```

Sütun silmek

Artık işe yaramadığı düşünülen bir sütun silinebilir. Genel ifadesi şu şekildedir:

```
ALTER TABLE tablo ismi  
DROP COLUMN sutun ismi sutun tanımları
```

Tablo silmek

Bir tabloya artık ihtiyaç kalmadığını düşünüyorsanız, silebilirsiniz.

Sadece içerdiği verileri etkin olarak silmek için DELETE veya TRUNCATE deyimini kullanabilirsiniz. Bir tablo silmek için şu genel ifade kullanılır:

```
DROP TABLE tablo ismi;
```

Geçici tablolarla çalışmak

Komplike sorguları adımlara ayırmak için, geçici bir süre kayıtları tutmak üzere ek tablolara ihtiyaç duyabiliriz. Bu tür durumlarda, T-SQL ile geçici tablolar oluşturup onları kullanabiliriz. Geçici tablolar, diğer bütün geçici nesnelere gibi tempdb'de saklanır ve kullanıcı çıkış yaptığı veya SQLServer kapatıldığında otomatik olarak silinir. Global ve yerel geçici tablolar tanımlamak mümkündür. Global geçici tablolar, birden fazla bağlantı için erişilebilir durumda iken yerel geçici tablolar sadece oturumu açan kişi tarafından erişilebilir.



Geçici tablolar ile ana veri tabanı dosyasında düzenleme yapmaksızın geçici bir tablo üzerinde işlem yapılabilir böylelikle büyük dosyayı düzenleme yerine küçük dosyalar oluşturulup üzerinde çalışılıp silinebilir.

Geçici tablolar oluşturmanın iki yolu vardır:

1 . Yöntem: Oturum boyunca geçerli geçici tablolar oluşturmak için kullanılır.

```
CREATE TABLE #tablo_adi(Alan1 turl[(boyut!)] [[NOT]
NULL],[...])
```

Bu şekilde oluşturulan tablolar oturum kapandığında veya SQL Server durdurulduğunda silinir.



Örnek

- Geçici bir kitaplar tablosu oluşturalım:
- CREATE TABLE #ogrencibilgi(
•ogrenciNo INT,
•ogrenciAdi VARCHAR(55))
•GO
•Daha sonra tabloya, normal bir tabloymuş gibi kayıt ekleyebiliriz:
- INSERT INTO #ogrencibilgi VALUES (1, 'Ali Demir') ya da tabloda yer alan kayıtları
- SELECT * FROM #ogrencibilgi

ile görebiliriz. Kayıtları güncelleyebiliriz ya da silebiliriz. Ancak bu tablo, oturumu kapattığımız anda veya SQLServer kapatıldığı anda silinecektir. Bunun dışında biz de istediğimiz zaman bu tabloyu bildiğimiz yöntemle silebiliriz.

```
DROP TABLE #ogrencibilgi
```

Global bir tablo tanımlamak için tablonun isminin iki adet ## ile başlaması gerekir.

2. Yöntem: Geçici tablo oluşturmanın bir diğer yolu, tempdb adlı veri tabanı dosyasına bir tablo açmaktır. Bu veritabanındaki tablolar, sadece SQLServer kapatıldığında silinir.

Genel yapısı şöyledir:

```
CREATE TABLE tempdb..tablo_adi( Alan1 turl[(boyut!)] [[NOT] NULL],[...])
```

Bu yöntemle oluşturulan geçici tablolar, SQL Server durdurulduğu anda kaybolur. Çünkü tempdb adlı veri tabanı SQL server açıldığı anda boşlatacaktır. Bazen, kullanıcı çıkış yaptığı halde geçici tablonun saklanması ihtiyacı olabilir. Bu tür durumlarda, geçici tablo SQL Server kapanmıca kadar hafızada kalıp SQL

server kapatıldığında silinecekse tempdb'de tablo açma yöntemi kullanılabilir.

Örnek

- tempdb..tablo_ismi şeklinde bir kitap adında geçici tablo oluşturulur:
- CREATE TABLE tempdb..ogrencibilgi (ogrenciNo INT, ogrenciAdi VARCHAR(55))

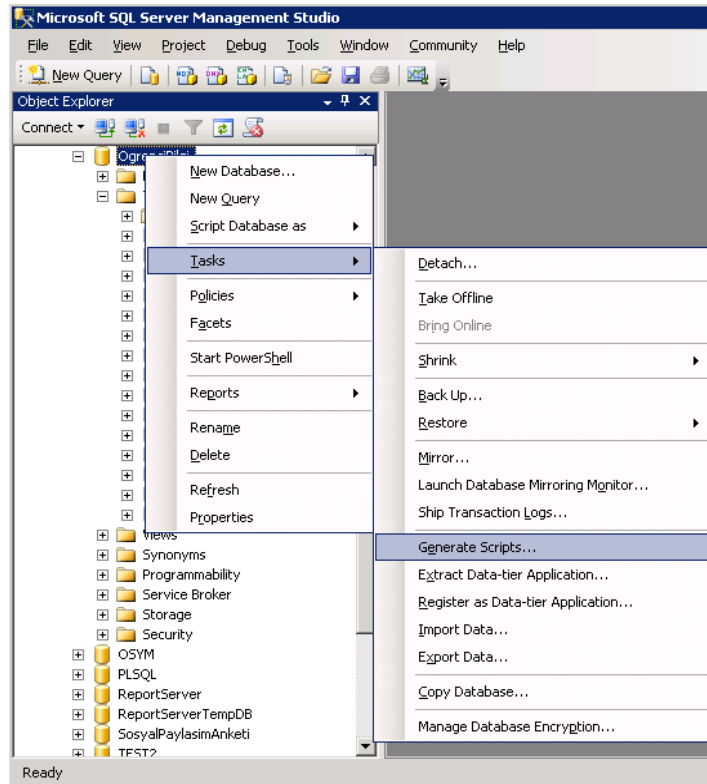
Nesnelerin T-SQL İfadelerini Almak

Bir proje tamamlandıktan sonra, taşınması gerekiyor olabilir. Bu tür durumlarda genellikle Integration Services (eski adı ile DTS) birbiri ile aynı ağ üzerindeki sunucular arasında veri alışverişi için kullanılabilir tekniklerdir. Ancak durum her zaman böyle olmayabilir ya da geliştirilen uygulama, bir paket hâline getirilip dağıtılıyor olabilir. Bu türden durumlarda, veri tabanını oluşturan nesnelerin Transact-SQL ifadelerini bir düz metin dosyasında saklayıp daha sonra bu metin bir şekilde çalıştırılarak aynı şemanın başka bir sunuya aktarılması sağlanabilir.

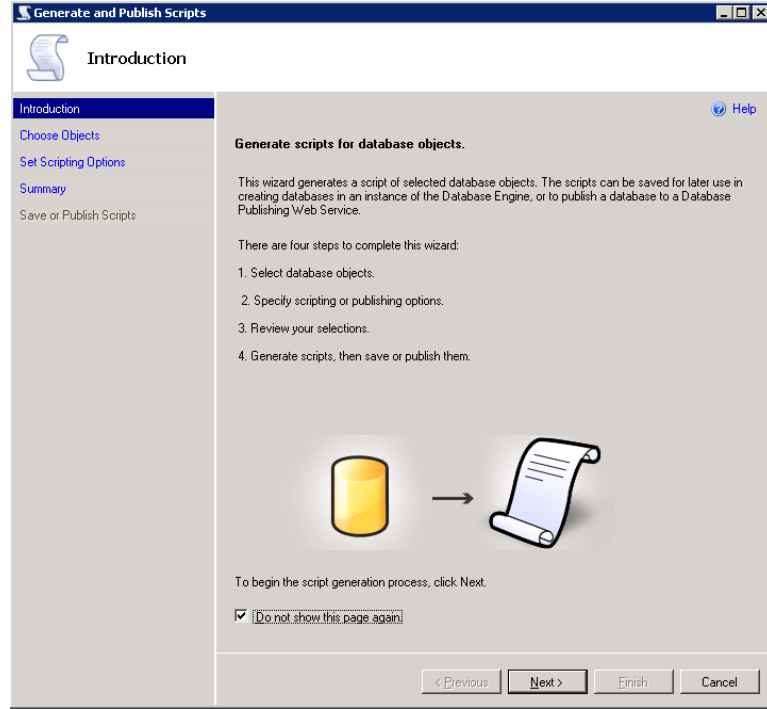
Management Studio ile bir veri tabanının Transact-SQL scriptini oluşturmak oldukça kolaydır. Veri tabanı üstüne sağ tıklandıktan sonra, Tasks \ Generate Scripts menüsünden "Script Wizard"ı başlatabilirsiniz.



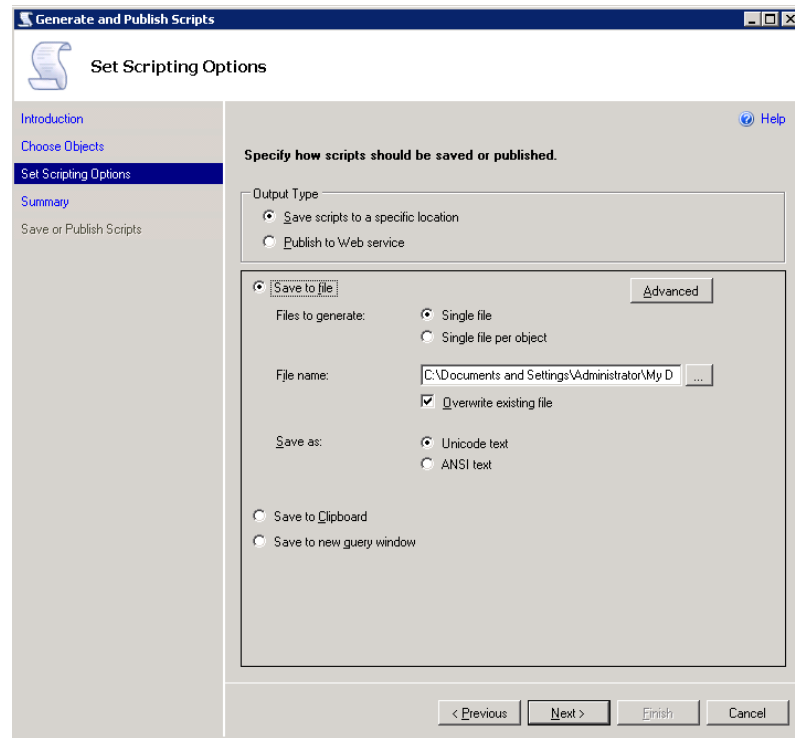
Management Studio ile gerçekleştirilen tüm işlemlerin TSQL karşılıkları alınabilir. Veri tabanı- tablo yapı ve verileri TSQL ifadesi olarak alınabilir.



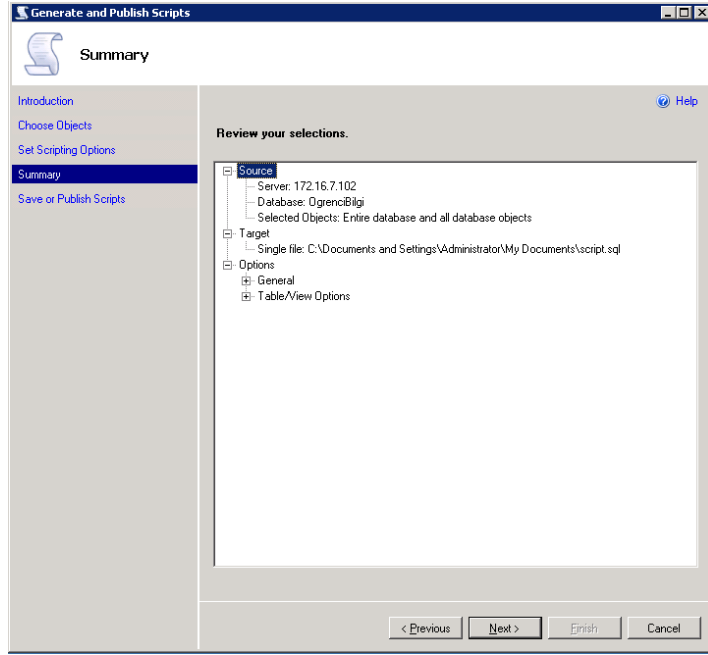
Şekil 5.1. Nesnelerin TSQL İfadelerinin Alınması Adım-1



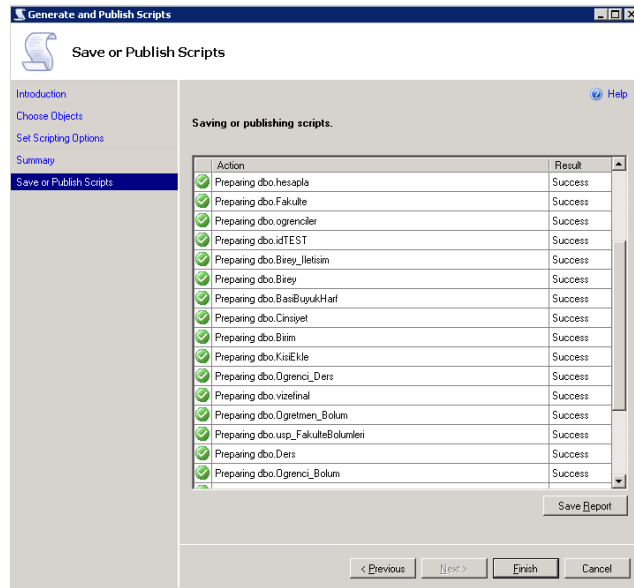
Şekil 5.2. Nesnelerin TSQL İfadelerinin Alınması Adım-2



Şekil 5.3. Nesnelerin TSQL ile İfadelerinin Alınması Adım-3



Şekil 5.4. Nesnelerin TSQL İfadelerinin Alınması Adım-4



Şekil 5.5. Nesnelerin TSQL İfadelerinin Alınması Adım-4



Özet

•TSQL

•Transact-SQL, SQL Server ve istemci (client) arasında iletişimi sağlayan SQL sorgulama dilinin gelişmiş bir versiyonudur. SQL Yapılandırılmış sorgulama dili (Structured Query Language) anlamına gelen Transact Structured Query Language kelimelerinin kısaltmasıdır. T-SQL kullanarak veri tabanına kayıt eklenebilir, silinebilir, güncellenebilir; kullanıcı, eklenebilir silinebilir, güncellenebilir ya da sorgulama ve raporlama yapılabilir. TSQL bir programlama dili değil sorgu dilidir.

•T-SQL ile aşağıdaki işlemler yapılabilir:

- Veri Tabanı oluşturmak
- Tablo oluşturmak
- Veri tabanı ve tablolar üzerinde değişiklikler yapmak
- Kayıt ekleme, silme, güncelleme
- Verileri Filtrelemek

•T-SQL ile döngü veya mantıksal işlemler yapmak için bir derleyiciye gerek yoktur. Herhangi bir programlama dili öğrenmeden de T-SQL ile tüm amaçlarınıza hitap edecek projeler gerçekleştirebilirsiniz.

•TSQL ile Veri Tabanı Oluşturmak

•"CREATE DATABASE veri_tabanı_ismi " komutu ile veri tabanı oluşturabilirsiniz.

•Veri Tabanına Erişecek Uygulama Kullanıcısını Ayarlamak

•"CREATE LOGIN webuser WITH PASSWORD = 'sifre' " komutu ile SQL SERVER üzerinde kullanıcı oluşturabilirsiniz.

•Veri Tabanı Özelliklerini Değiştirmek

•"ALTER DATABASE veritabanı_ismi SET secenek durum" komutu ile veri tabanı özellikleri değiştirilebilir.

•Veri Tabanını Silmek

•"DROP DATABASE ogrencibilgi" komutuyla veri tabanı silinebilir.

•Tablo Oluşturmak

•"CREATE TABLE tabloAdi (kolon adı1 veri tipi[NOT NULL], kolon adı2 veri tipi [NOT NULL],)" komutuyla veri tabanı üzerinde tablo oluşturulabilir.

•Var olan tabloya sütun eklemek

•"ALTER TABLE tablo_ismi ADD COLUMN sütun_ismi sütun özellikleri " komutuyla tabloya yeni sütun eklenebilir.

•Varolan sütunları değiştirmek

•"ALTER TABLE tablo_ismi ALTER COLUMN sütun_ismi sütun tanımları" komutuyla sütunlar değiştirilebilir.

•Varolan sütunları silmek

•"ALTER TABLE tablo_ismi DROP COLUMN sütun_ismi sütun tanımları" komutuyla sütunlar değiştirilebilir.

•Geçici tablolarla çalışmak

•Kısa süreli sorguları adımlara ayırmak için, geçici bir süre kayıtları tutmak üzere ek tablolara ihtiyaç duyabiliriz. Bu tür durumlarda, T-SQL ile geçici tablolar oluşturup onları kullanabiliriz.

•1.Yöntem: Oturum boyunca geçerli geçici tablolar oluşturmak için kullanılır.

•CREATE TABLE #tablo_adi(Alan1 turl[(boyut1)] [[NOT] NULL],[...])

•2.Yöntem: Geçici tablo oluşturmanın bir diğer yolu, tempdb adlı veri tabanı dosyasına bir tablo açmaktır. Bu veritabanındaki tablolar, sadece SQLServer kapatıldığında silinir.

•CREATE TABLE tempdb..tablo_adi(Alan1 turl[(boyut1)] [[NOT] NULL],[...])

DEĞERLENDİRME SORULARI

1. TSQL ile tablo oluşturma komutu aşağıdakilerden hangidir?
 - a) CREATE TABLE
 - b) CREATE DATABASE
 - c) ADD TABLE
 - d) ADD DATABASE
 - e) ALTER TABLE
2. TSQL ile tablo oluştururken aşağıdakilerden hangisi zorunludur?
 - a) Veri tabanı ismi
 - b) Tablo ismi
 - c) Veri tabanı kullanıcı adı
 - d) Tablo verisinin saklanacağı dosya
 - e) Tablo verisinin saklanacağı dosya boyutu
3. Tablodan bir kayıt silmek için hangi komut kullanılır?
 - a) ADD
 - b) CREATE
 - c) INSERT
 - d) DROP
 - e) DELETE
4. SQL Serverda oluşturulmuş bir tablonun bir sütununu değiştirmek için hangi komut kullanılır?
 - a) TABLE <tablo ismi> DROP COLUMN <kolon ismi>
 - b) ALTER TABLE <tablo ismi> DELETE COLUMN <kolon ismi>
 - c) TABLE <tablo ismi> DELETE COLUMN <kolon ismi>
 - d) ALTER TABLE <tablo ismi> ALTER COLUMN <kolon ismi>
 - e) ALTER TABLE <tablo ismi> DROP COLUMN <kolon ismi>
5. Aşağıdakilerden hangisi T-SQL ile yapılabilir?
 - a) Sunucu oluşturmak
 - b) Web sitesi tasarlamak
 - c) Veri tabanı oluşturmak
 - d) Windows formu oluşturmak
 - e) Program derlemek
6. Aşağıdakilerden hangisi T-SQL ile yapılamaz?
 - a) Kayıt silinebilir.
 - b) Kayıt eklenebilir.
 - c) Form oluşturulabilir.
 - d) Tablo üzerinde değişiklik yapılabilir.
 - e) Tablo oluşturulabilir.

7. Aşağıdakilerden hangisi veri tabanında çeşitli işlemleri yerine getiren komutlardan biri değildir?
- DELETE
 - INSERT
 - UPDATE
 - ERASE
 - CREATE
8. Grant komutu ne için kullanılır?
- Veri tabanı kullanıcılarını düzenler.
 - Veri tabanını düzenler.
 - Tabloya erişecek kullanıcı haklarını düzenler.
 - Veri tabanına erişecek kullanıcı şifresini düzenler.
 - Veri tabına erişecek kullanıcı haklarını düzenler.
9. Veri tabanına tablo silmek için hangi TSQL komutu kullanılır?
- DROP Database
 - DROP User
 - DROP Table
 - DROP Login
 - DROP tablo
10. Veri tabanı kullanıcı oluşturulurken hangi TSQL komutu kullanılır?
- Create Database
 - Create File
 - Alter Database
 - Create User
 - Create Login

Cevap Anahtarı

1.a, 2.b, 3.e, 4.d, 5.c, 6.c, 7.d, 8.e, 9.c

YARARLANILAN KAYNAKLAR

Sql Server Books Online, (2012). 20 Temmuz 2019 tarihinde

[https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms130214\(v=sql.105\)](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms130214(v=sql.105)) adresinden erişildi.