

T.C.
MİLLÎ EĞİTİM BAKANLIĞI



MEGEP

(MESLEKÎ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

BİLİŞİM TEKNOLOJİLERİ

VERİTABANI YÖNETİMİ

ANKARA 2008

Millî 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 İnternet üzerinden ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR	iii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	3
1. VERİ YÖNETİMİ.....	3
1.1. Veri Alma ve Dışarı Veri Yollama (Import ve Export)	3
1.1.1. Veri Alma (Import)	3
1.1.2. Dışarı Veri Yollama (Export).....	12
1.2. “bcp” Komut İstemi Programı.....	16
1.3. Bulk Insert Komutu	18
1.4. Veri Transfer Hizmetleri	19
1.4.1. SQL Server 2000 Veritabanını SQL Server 2005’e Aktarmak.....	19
UYGULAMA FAALİYETİ.....	21
ÖLÇME VE DEĞERLENDİRME.....	26
ÖĞRENME FAALİYETİ-2	27
2. İLERİ SORGU TEKNİKLERİ	27
2.1. Inner Join ve Outer Join	27
2.1.1. Inner Join.....	27
2.1.2. Outer Join	28
2.2. Alt Sorgular	29
2.2.1. IN ve NOT IN	29
2.2.2. İç İçe SELECT	29
2.2.3. Türetilmiş Tablolar.....	31
2.2.4. EXISTS ve NOT EXISTS.....	33
UYGULAMA FAALİYETİ.....	35
ÖLÇME VE DEĞERLENDİRME.....	39
ÖĞRENME FAALİYETİ-3	40
3. İMLEÇ	40
3.1. Kullanım Amaçları.....	40
3.2. T-SQL İmleçler	40
3.2.1. T-SQL İmleç Kullanımı	41
3.2.2. @@FETCH_STATUS ve @@ROWCOUNT	41
3.2.3. İmleç Seçeneklerini Ayarlamak	44
3.3. İmleçlerle Satırlar Arasında Dolaşmak	44
3.4. İmleçle Kayıtları Düzenlemek	45
3.5. Ortak Zamanlı Çalışma	46
UYGULAMA FAALİYETİ.....	47
ÖLÇME VE DEĞERLENDİRME.....	49
ÖĞRENME FAALİYETİ-4.....	50
4. TETİKLEYİCİ	50
4.1. Tetikleyici Ne Zaman Kullanılır?	50
4.2. Tetikleyiciyi Ateşleyen Olaylar	51
4.3. Tetikleyici Türleri	51

4.3.1. After Tetikleyicileri.....	51
4.3.2. Instead Of Tetikleyicileri	51
4.4. Tetikleyici Oluřturmak.....	52
4.4.1. INSERT Tetikleyicisi.....	52
4.4.2. DELETE Tetikleyicisi.....	54
4.4.3. UPDATE Tetikleyicisi	56
4.4.4. INSTEAD OF Tetikleyicisi.....	57
4.5. Tetikleyici Yönetimi	59
4.5.1. Tetikleyiciyi Düzenlemek (ALTER).....	59
4.5.2. Tetikleyiciyi Silmek (DROP).....	60
4.5.3. Tetikleyiciyi Aktif ya da Pasif Yapma.....	61
UYGULAMA FAALİYETİ.....	62
ÖLÇME VE DEĞERLENDİRME.....	64
ÖĞRENME FAALİYETİ-5	65
5. GÖRÜNÜM	65
5.1. Hangi Amaçlar İçin Kullanılır?.....	65
5.2. View Oluřturmak	65
5.3. Kısıtlamalar ve İzinler	70
5.4. Birden Fazla Tablo Kullanımı.....	71
5.5. View’de Değişiklik Yapmak veya View’i Silmek.....	72
5.5.1. Değişiklik Yapmak.....	72
5.5.2. Silmek.....	72
UYGULAMA FAALİYETİ.....	73
ÖLÇME VE DEĞERLENDİRME.....	76
ÖĞRENME FAALİYETİ-6.....	77
6. İNDEKS	77
6.1. İlişkisel Veritabanında İndeksler.....	77
6.2. Küme ve Küme Olmayan İndeksler	78
6.3. İndeks Oluřturmak	78
6.3.1. Unique İndeks	79
6.3.2. Karma (Composite) veya Birleşik İndeks	79
6.4. İndeks’te Değişiklik Yapmak.....	80
6.4.1. İndeks Silmek.....	81
6.5. “sp_helpindex” Komutu.....	81
UYGULAMA FAALİYETİ.....	83
ÖLÇME VE DEĞERLENDİRME.....	84
MODÜL DEĞERLENDİRME	85
CEVAP ANAHTARLARI.....	86
KAYNAKLAR.....	88

AÇIKLAMALAR

KOD	481BB0046
ALAN	Bilişim Teknolojileri
DAL/MESLEK	Veritabanı Programcılığı
MODÜLÜN ADI	Veritabanı Yönetimi
MODÜLÜN TANIMI	SQL Server’da veri bütünlüğü veya kısıtlamaları ile ilgili öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	Veri Bütünlüğü modülünü bitirmiş olmak
YETERLİK	
MODÜLÜN AMACI	Genel Amaç Gerekli ortam sağlandığında, verileri sorgulayabilecek, tetikleyici ve görünüm tabanlı çalışmayı yapabileceksiniz. Amaçlar 1. Veriye ulaşabileceksiniz. 2. Veriye ileri sorgu teknikleriyle ulaşabileceksiniz. 3. İmleci kullanabileceksiniz. 4. Tetikleyicileri kullanabileceksiniz. 5. Görünüm tabanlı çalışabileceksiniz. 6. İndeks oluşturabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Ortam Atölye, laboratuvar, bilgi teknolojileri ortamı (İnternet) vb. kendi kendinize veya grupla çalışabileceğiniz tüm ortamlar. Donanım Ağ veritabanını çalıştırabilecek yeterlikte bilgisayar, yedekleme için gerekli donanım (CD yazıcı, flash bellek), raporlama için yazıcı, kâğıt ve kalem.
ÖLÇME VE DEĞERLENDİRME	Modülün içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz, modül sonunda ise bilgi ve beceriyi belirlemek amacıyla, öğretmeniniz tarafından belirlenecek ölçme aracıyla değerlendirileceksiniz.

GİRİŞ

Sevgili Öğrenci,

Okul yaşantınızda öğreneceğiniz her konu, yaptığımız uygulama ve tamamladığınız her modül bilgi dağarcığınızı geliştirecek ve ileride atılacağınız iş yaşantınızda size başarı olarak geri dönecektir. Eğitim sürecinde daha özverili çalışır ve çalışma disiplinini 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 veritabanı programıyla yapabileceğiniz gibi ağ ortamında güvenlik ve hızlı erişim açısından en iyi sonucu veren SQL Server veritabanıyla da yapabilirsiniz. Bu program, milyonlarca kaydın olduğu tablolar üzerinde işlem yaparken tüm kullanıcılara hitap edebilmektedir. İstenilen sorgu sonuçlarını da en hızlı şekilde elde edebilmenizi sağlamaktadır.

Bu modülle, SQL Server’da veri yönetimini, ileri sorgu tekniklerini, imleç kullanmayı, tetikleyici, görünüm ve indeks oluşturup kullanabilmeyi öğreneceksiniz.

ÖĞRENME FAALİYETİ-1

AMAÇ

SQL Server’da veri yönetimini öğreneceksiniz.

ARAŞTIRMA

- Programlama dilleri ya da programlama dili destekli paket programlarda veri yönetiminin nasıl yapıldığını araştırınız.

1. VERİ YÖNETİMİ

1.1. Veri Alma ve Dışarı Veri Yollama (Import ve Export)

Veri yönetimi, değişik kaynaklardan elde edilen bilgilerin toplanabilmesi, güncelleştirilebilmesi, ayrıştırılabilmesi, bir araya getirilebilmesi ve sonuç olarak değerlendirilebilmesidir.

SQL Server’da veri alıp verme işlemleri SQL Server Import and Export Wizard ile yapılmaktadır. Bu veri alışverişi Access, Excel, virgülle bölünmüş text dosya vb. kaynaklarla olabilir.

1.1.1. Veri Alma (Import)

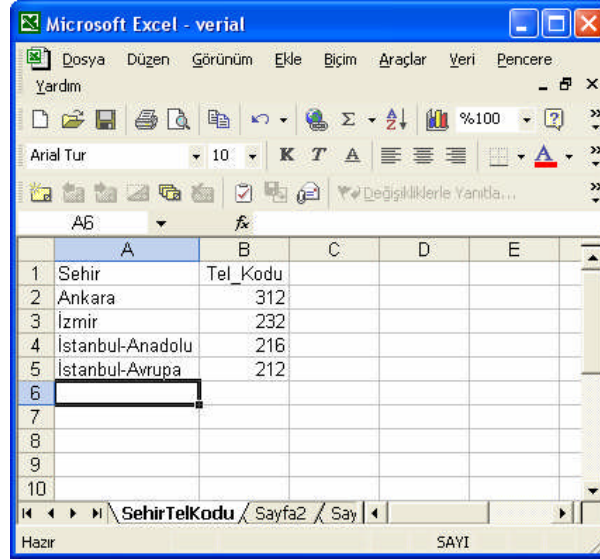
Hazırladığınız veritabanına dışarıdan veri almak için SQL Server Import and Export Wizard kullanılmaktadır. Veri alınacak kaynak bir text dosya, Access veritabanı dosyası veya Excel çalışma sayfası olabilir.

Veritabanına dışarıdan veri almak için veritabanı üzerinde fareyle sağ tıklayarak açılan menüden Task komutuyla Import Data komutunu kullanılmaktadır.

Örnek olması açısından bir Excel çalışma sayfasından nasıl veri alınacağını uygulayarak bakalım.

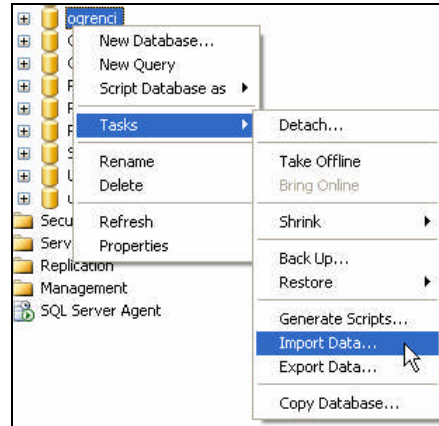
Örnek

- Excel programını açarak A ve B sütununa Resim 1.1'deki gibi bilgileri giriniz. Bu bilgiler şehir adı ve şehrin telefon kodu şeklinde olacaktır.



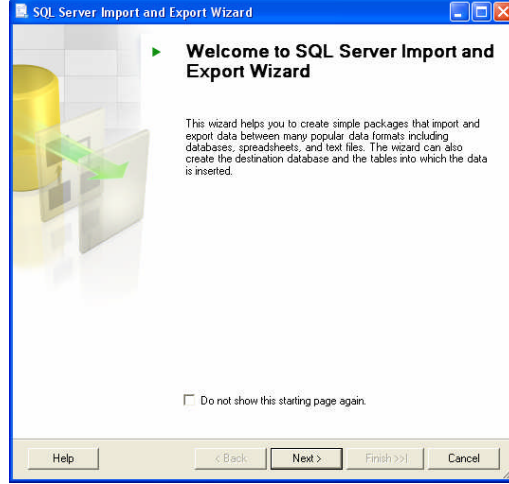
Resim 1.1: Veri alınacak Excel çalışma sayfası

- Çalışma sayfasının adını “SehirTelKodu” olarak değiştiriniz.
- Excel dosyasına “verial” ismini vererek kaydediniz.
- Çalıştığımız veritabanı üzerinde fareyle sağ tıklayarak açılan menüden Task komutuyla Import Data komutunu veriniz.



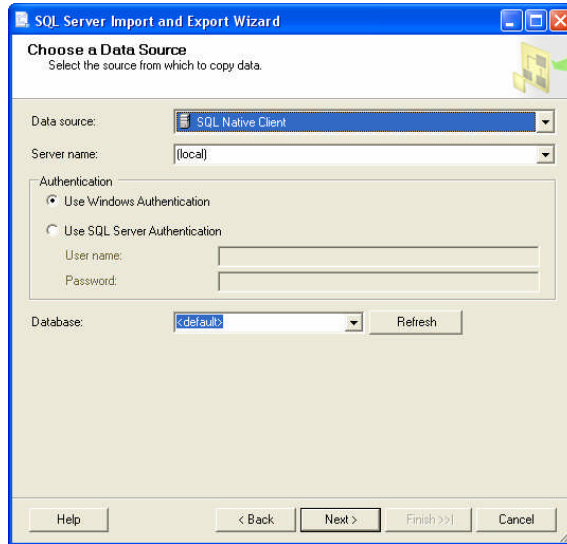
Resim 1.2: Import Data komutu

- Import Data komutunu tıkladığınızda ekrana SQL Server Import and Export Wizard gelecektir.



Resim 1.3: SQL Server Import and Export Wizard ekranı

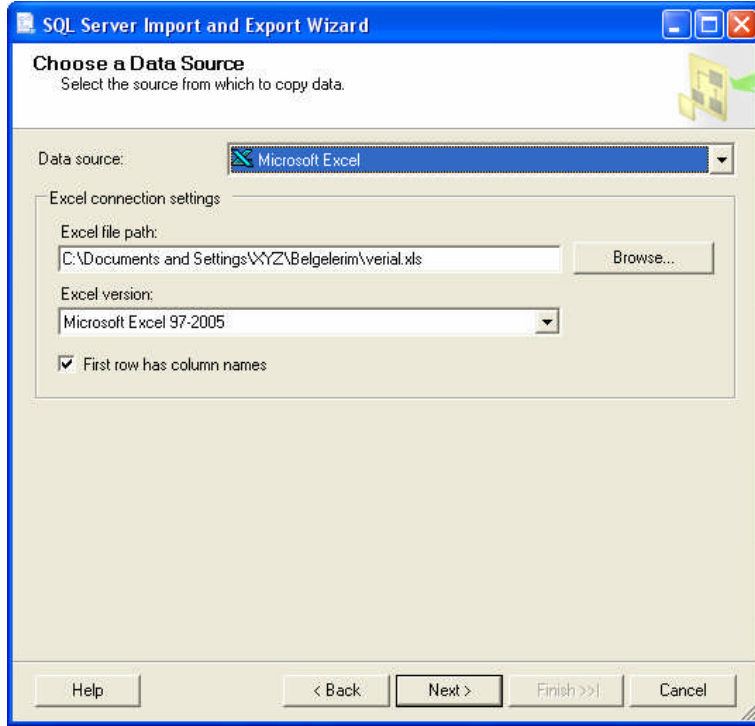
- Bu ilk pencere sihirbaz ile ne gibi işlemler yapabileceğinizi anlatmaktadır. Ekrana gelen bu ilk sihirbaz penceresini bir daha görmek istemiyorsanız “Do not show this starting page again” onay kutusunu işaretleyiniz.
- Sihirbazın diğer adımına geçmek için Next düğmesini tıklatınız. Next düğmesini tıkladığınızda veri kaynağının seçileceği “Choose a Data Source” penceresi ekrana gelir.



Resim 1.4: Veri kaynağının seçimi penceresi

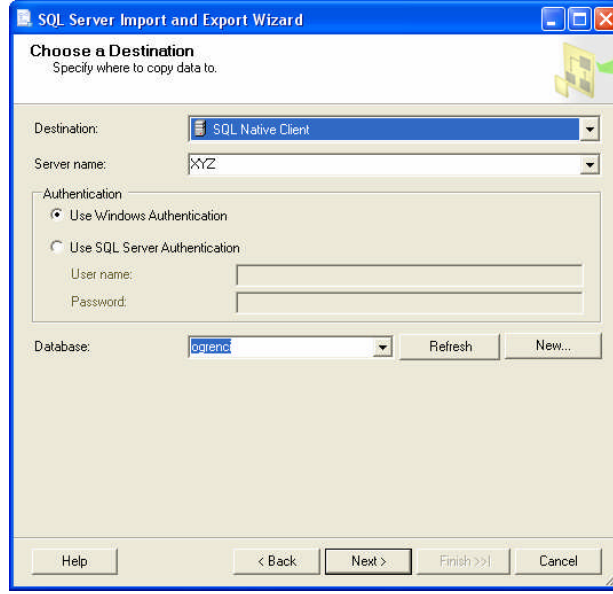
Bilgilerin alınacağı ya da kopyalanacağı veri kaynağının seçimi bu pencerede yapılmaktadır. Data Source açılır liste kutusundan bu kaynağı seçebilirsiniz.

- Excel'den veri alınacağı için Data Source açılır liste kutusundan Microsoft Excel'i seçiniz.
- "Excel File Path" kutusuna dosyanızın yolunu yazınız ya da "Browse" düğmesini tıklayarak oluşturduğunuz Excel dosyasının yolunu belirtiniz.



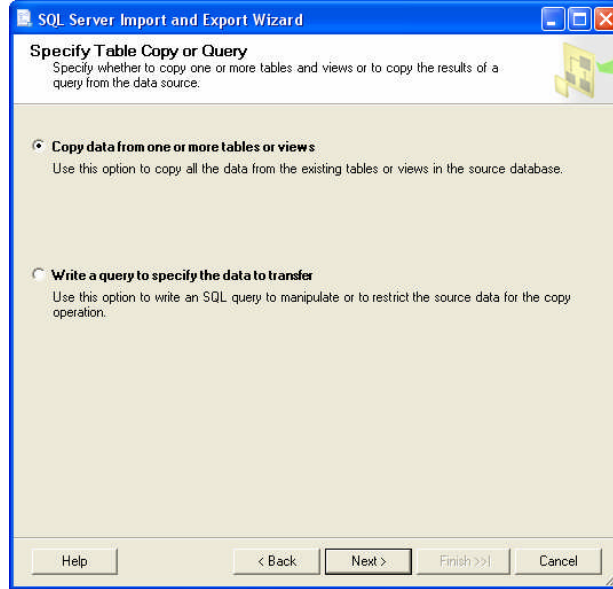
Resim 1.5: Excel dosyasının veri kaynağı olarak belirtilmesi

- Excel dosyasının yolu belirtildikten sonra çalışma sayfasındaki başlıkların veritabanındaki tabloda sütun adı olarak belirlenmesi için "First row has column name" onay kutusunu işaretleyiniz.
- Bir sonraki adıma geçmek için Next düğmesini tıklayınız. Ekranı bilgilerin kopyalanacağı, hedefin seçiminin yapıldığı "Choose a Destination" penceresi gelecektir.



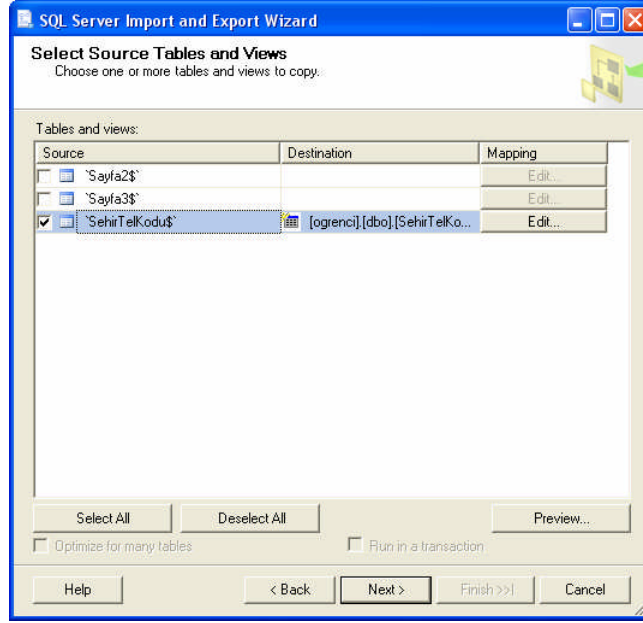
Resim 1.6: SQL Server hedef veritabanının seçilmesi

- Database açılır liste kutusundan verinin alınacağı hedef veritabanını seçebilirsiniz. Eğer bilgileri yeni bir veritabanına alacaksanız New komutunu tıklayarak yeni bir veritabanı oluşturabilirsiniz.
- Veritabanınızı seçtikten sonra Next düğmesini tıklayınız. Ekranı tablonun nasıl oluşacağını sağlayacak pencere gelecektir.



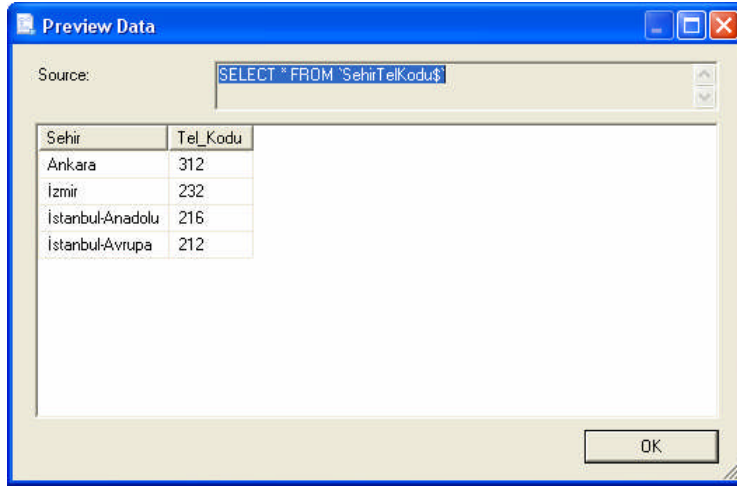
Resim 1.7: Tablonun oluşum ekranı

- “Copy data from one or more tables or views” seçimini yaparak Next düğmesini tıklayınız.
- Ekranı kaynak tabloların ve görünümünün seçileceği “Select Source Tables and Views” penceresi gelecektir.



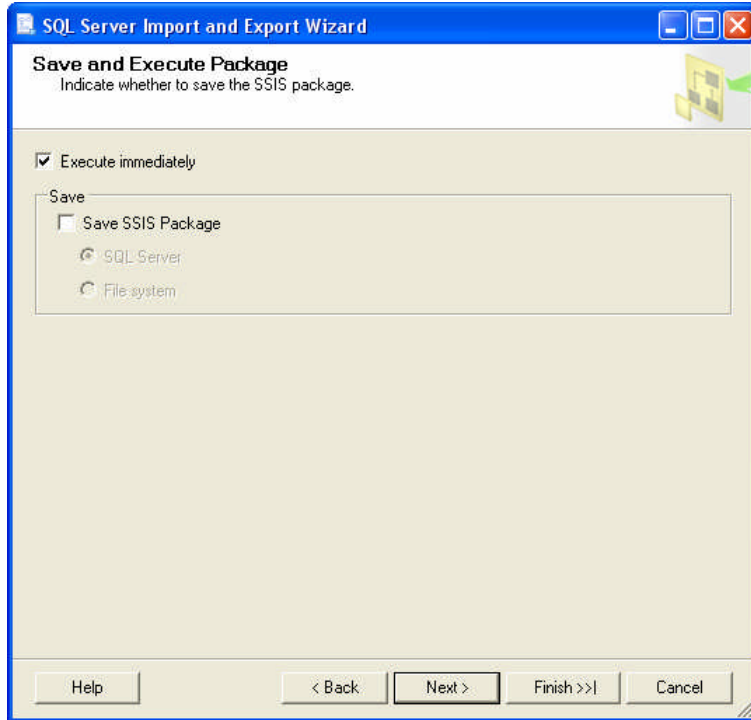
Resim 1.8: Kaynak tablolar ve görünümünün seçimi

- Excel’de hazırlanmış çalışma sayfasının adı olan “SehirTelKodu”nu seçiniz. Seçimi yaptığınızda Edit düğmesi etkin hâle gelecektir.
- Eğer, tabloya aktarılacak olan verilerin veri türlerini ve boyutlarını değiştirmek isterseniz Edit düğmesi tıklandığında açılan “Column Mappings” penceresinde düzeltme yapabilirsiniz.
- Çalışma sayfasında var olan tüm sayfaları seçmek isterseniz “Select All” düğmesini, seçilmiş olan sayfaların seçimini kaldırmak isterseniz “Deselect All” düğmesini tıklayabilirsiniz.
- Seçilen çalışma sayfasının verilerini Preview düğmesini tıklayarak görebilirsiniz.



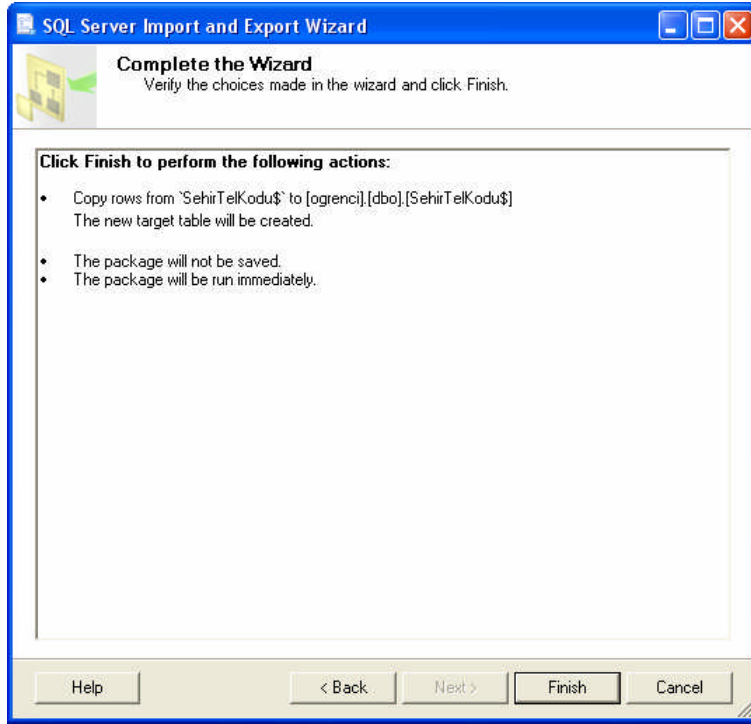
Resim 1.9: Çalışma sayfasının verilerinin önizlemesi

- Bilgileri gördükten sonra OK düğmesini ve bir sonraki adım için Next düğmesini tıklayınız.
- Ekranı kayıt işleminin yapılacağı ve paketin çalıştırılacağı “Save and Execute Package” penceresi gelecektir.



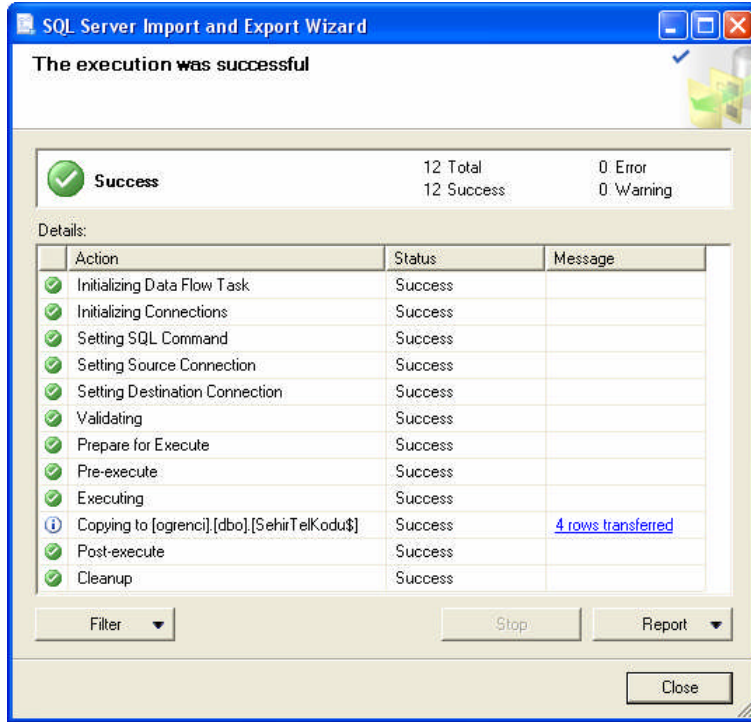
Resim 1.10: “Save and Execute Package” penceresi

- “Execute Immediately” seçeneği, paketi hemen çalıştırmak için kullanılır. Paketi oluşturup veri taşıma işlemi yapacaksanız “Save SSIS Package” seçeneğini seçiniz. Saklama ortamının ve hassas verilerin özellikleri bu seçenekle ayarlanmaktadır.
- Seçiminizi yaptıktan sonra Next düğmesini tıklatınız. Ekranı sihirbazın tamamlandığını gösteren “Complete Wizard” penceresi gelecektir.



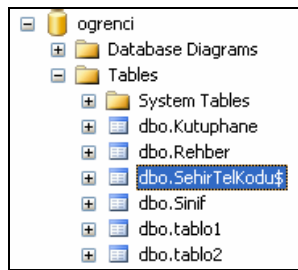
Resim 1.11: “Complete Wizard” penceresi

- Yapılan işlemlerin özetinin gösterildiği bu pencerede sihirbazı tamamlamak için Finish düğmesini tıklatınız. Çalıştırılma sonucunda yapılan işlemlerin başarılı bir şekilde gerçekleşip gerçekleşmediğini göreceksiniz.



Resim 1.12: Sonuç penceresi

- İşlemler başarılı olduysa Status sütununda Success ifadesini görebilirsiniz.
- Close düğmesine basarak sihirbazı kapatınız.
- Veritabanınızdaki tabloları seçtiğinizde Excel'den aldığınız tablonun eklendiğini görebilirsiniz. Göremiyorsanız tabloları Refresh yapmayı unutmayınız.



Resim 1.13: Import edilen verinin tablolardaki görüntüsü

- Tablo içindeki verileri de görmek isterseniz tabloyu Open Table komutuyla açınız.

Table - dbo.SehirTelKodu\$		Summary
	Sehir	Tel_Kodu
▶	Ankara	312
	İzmir	232
	İstanbul-Anadolu	216
	İstanbul-Avrupa	212
*	NULL	NULL

Resim 1.13: Tablo içeriği

1.1.2. Dışarı Veri Yollama (Export)

Veritabanınıza dışarıdan veri alınabileceği gibi dışarı veri de yollanabilir. Bunun için de Export Data komutu kullanılmalıdır.

Dışarı veri yollama işlemini veri alma işlemi gibi örnekle anlatalım.

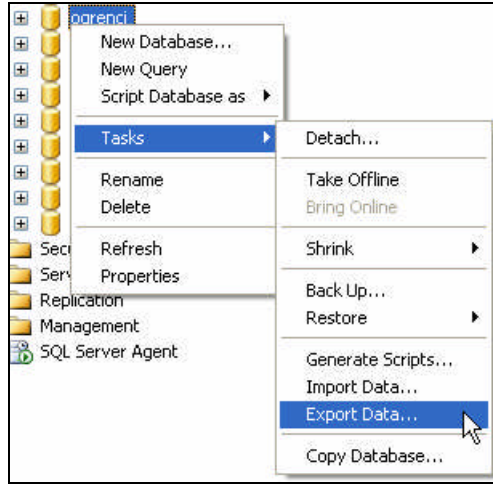
Örnek

- Daha önceki örneklerde yapılan “Sınıf” tablosunu Export edelim.

Table - dbo.Sınıf						Summary
	Ogr_No	Ad	Soyad	Cinsiyet	Yas	
▶	101	Hasan	KAYA	E	17	
	103	Veli	CAN	E	16	
	104	Ayşe	KOCAER	K	17	
	106	Fatma	YILMAZ	K	18	
	109	Defne	YAĞMUR	K	16	
*	NULL	NULL	NULL	NULL	NULL	

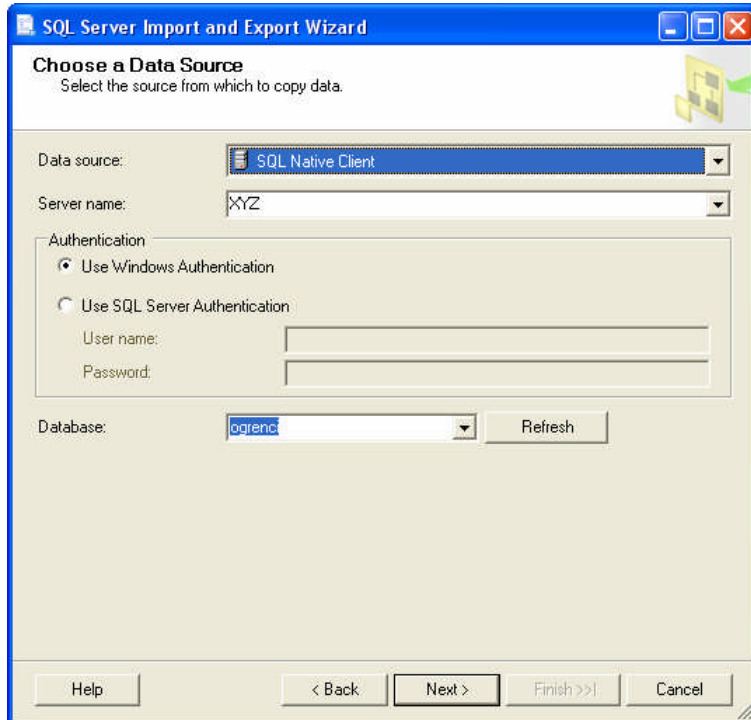
Resim 1.14: Sınıf tablosu

- Bunun için önce Excel’de verilerin alınacağı bir Excel dosyası oluşturunuz. Adını da “veri_export” olarak tanımlayınız.
- Veri yollamak için Tasks komutu üzerinden Export Data komutunu veriniz.



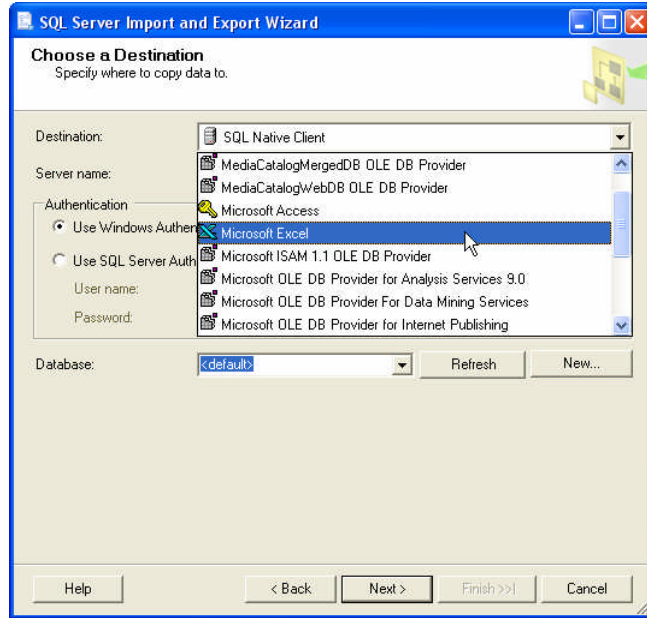
Resim 1.15: Export Data komutu

- Veri alma ve yollama sihirbazının ilk adımını geçerek veri kaynağını belirtiniz. Veri kaynağınız, tabloyu göndermek istediğiniz veritabanının adı olacaktır.



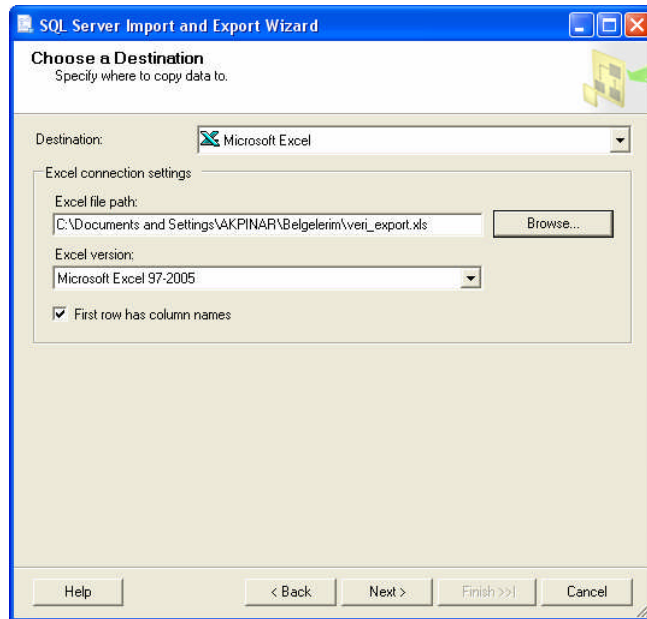
Resim 1.16: Veri kaynağının seçimi

- Veri kaynağınızı seçtikten sonra Next düğmesini, tıklayınız. Ekrana hedef seçiminin yapılacağı pencere gelecektir.



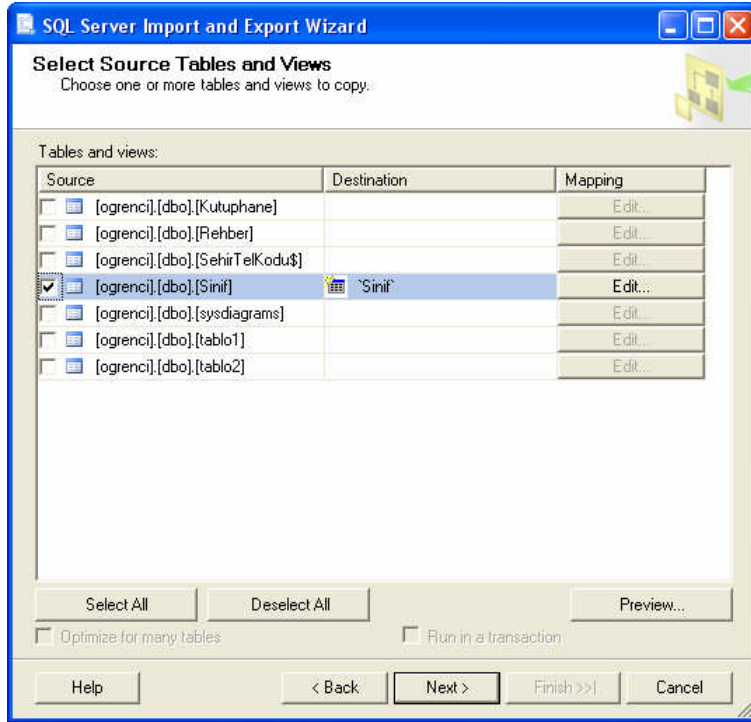
Resim 1.17: Hedefin seçimi

- Destination kutusundan veriyi yollayacağınız hedefi (Microsoft Excel) seçiniz. Excel'i seçtiğinizde yollanacak verinin aktarılacağı Excel dosyasının yolunu belirtmeniz istenir. Browse düğmesine basarak daha önceden oluşturduğunuz "veri_export" dosyasını belirtiniz.



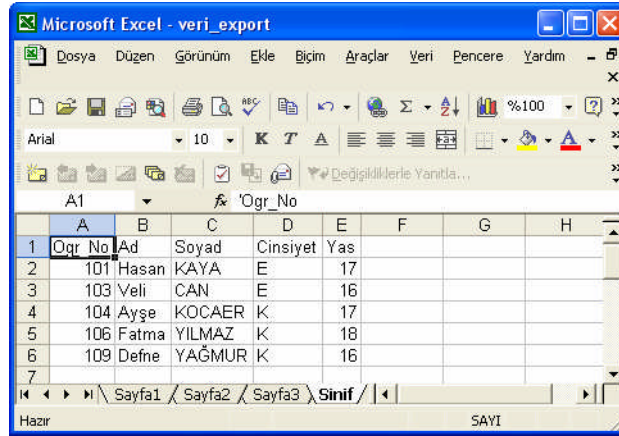
Resim 1.18: Hedefin seçimi

- Next düğmesini tıklayarak diğer adıma geçiniz. Ekrana tablonun nasıl oluşacağını sağlayacak Resim 1.7'deki pencere gelecektir. İlk seçeneği seçip Next düğmesini tıklayınız.
- Kaynak tabloların ve görünülerin seçileceği “Select Tables and Views” penceresi ekrana gelecektir. Yollayacağınız tablo ya da tabloları seçerek Next düğmesini tıklayınız.



Resim 1.19: Tabloların ve görünülerin seçimi penceresi

- Ekrana kayıt işlemi ve paketin çalıştırılacağı Resim 1.10'daki “Save and Execute Package” penceresi gelecektir. Seçiminizi yaparak Next düğmesini tıklayınız.
- Sihirbazın son adımında da Finish düğmesine basarak paketin oluşturulmasını sağlayınız. Paketin oluşturulması esnasında hata olup olmadığını kontrol ediniz (Resim 1.12).
- Close düğmesine basarak verilerin yollandığı Excel dosyasını açınız.
- Excel dosyasında verilerin bir çalışma sayfası olarak gönderildiğini göreceksiniz.



Resim 1.20: Export işleminin sonucu

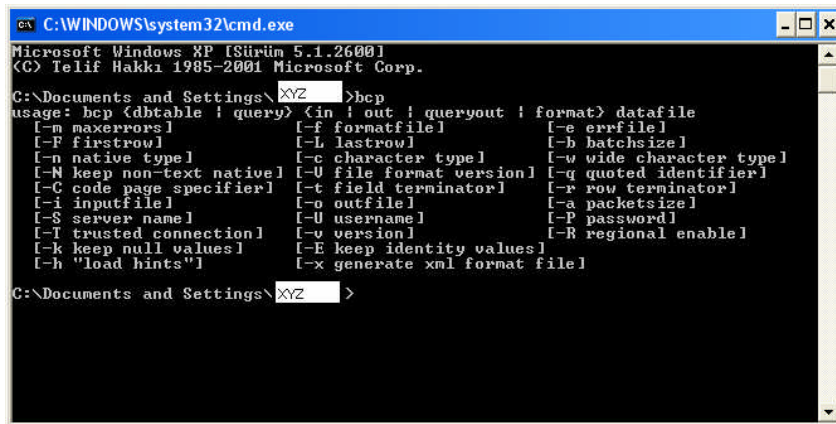
1.2. “bcp” Komut İstemi Programı

“bcp”, bir komut istemi programıdır. Verilerinizi dosyadan alma veya dosyaya yollama işlemleri için kullanılır.

“bcp” ile bir dosyaya veri aktarırken veritabanınızın tablo ya da tablolar üzerinde SELECT komutunu kullanma hakkı olmalıdır. Bir dosyadan veri alırken de yine tablo ya da tablolar üzerinde veritabanınızın mutlaka SELECT ve INSERT komutlarını kullanma hakkı olmalıdır.

Dışarıdan alınan veriler ancak veritabanında var olan bir tabloya aktarılabilir. Veritabanından bir dosyaya veri aktarılıyorsa belirtilen dosya olmasa da dosya oluşturulur ve aktarma gerçekleştirilir.

“bcp” komutunu çalıştırabilmek için Başlat/Çalıştır komutunu seçerek Cmd yazınız. Komut istemine bcp yazıp Enter tuşuna bastığınızda bcp ile kullanılan parametreleri görebilirsiniz.

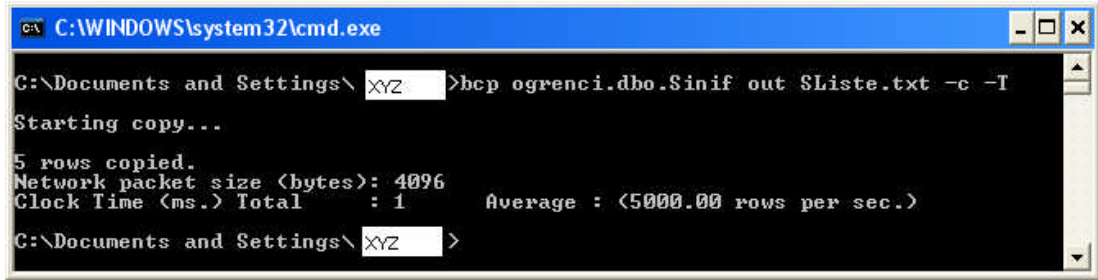


Resim 1.21: Bcp komutu parametreleri

Örnek

“Sinif” tablosundaki verileri bcp komutu ile bir dosyaya aktaralım.

- Komut istemine bcp komutunu Resim 1.22’deki gibi yazınız ve Enter tuşuna basınız.



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\XYZ >bcp ogrenci.dbo.Sinif out SListe.txt -c -T
Starting copy...
5 rows copied.
Network packet size (bytes): 4096
Clock Time (ms.) Total      : 1      Average : (5000.00 rows per sec.)
C:\Documents and Settings\XYZ >
```

Resim 1.22: Bcp komutunun out kullanımı

- “SListe.txt” dosyası ilgili klasörün altında oluşturulacaktır.
- Dosya içeriğini görmek için dosyayı Not Defteri programı ile açabilirsiniz.



Dosya	Düzen	Biçim	Görünüm	Yardım	
101	Hasan	KAYA	E	17	
103	Veli	CAN	E	16	
104	Ayşe	KOCAER	K	17	
106	Fatma	YILMAZ	K	18	
109	Defne	YALMUR	K	16	

Resim 1.23: Dosya içeriği

- “-c” parametresi karakter tipinde veri aktarımının yapılacağını, “-T” ise güvenli bağlantının kullanılacağını belirtir.

Bir dosyadan veritabanınıza veri almak için veri aktaracağınız tablonun veritabanında var olmasına dikkat etmeniz gerekir.

- Veri aldığımız “SListe.txt” dosya içeriğini bir tabloya aktarmak için de bcp komutunu Resim 1.24’teki gibi yazınız ve Enter tuşuna basınız. Bu işlem için yeni bir tablonuz olduğu varsayılacaktır.

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\XYZ >bcp ogrenci.dbo.bcp_yeni_tablo in SListe.txt
c -I
Starting copy...
5 rows copied.
Network packet size (bytes): 4096
Clock Time (ms.) Total : 31 Average : (161.29 rows per sec.)
C:\Documents and Settings\XYZ >
```

Resim 1.24: Bcp komutunun in kullanımı

- Burada sadece out ifadesi yerine in ifadesinin geldiğine dikkat ediniz.
- Veritabanındaki tablonuzu da açarak verilerin tabloya aktarılıp aktarılmadığını görebilirsiniz.

Table - dbo.bcp_yeni_tablo		Summary			
	nu	ad	soyad	cins	yas
▶	101	Hasan	KAYA	E	17
	103	Veli	CAN	E	16
	104	Ayşe	KOCAER	K	17
	106	Fatma	YILMAZ	K	18
	109	Defne	YAĞMUR	K	16
*	NULL	NULL	NULL	NULL	NULL

Resim 1.25: Verilerin tabloya aktarılmış hâli

1.3. Bulk Insert Komutu

Bcp komutuyla yapılan dışarıdan veri alma işlemi Bulk Insert komutuyla bir Query yazarak da yapılabilir. Bu işlem için de yeni bir tablo oluşturduğunuz varsayılacaktır.

- Bulk Insert komutu için Resim 1.26’da verilen kodları yeni bir Query’e yazarak F5 ile çalıştırınız.

```
XYZ.ogrenci - SQLQuery1.sql* Summary
BULK INSERT bulk_tablo
FROM 'C:\Documents and Settings\XYZ\SListe.txt'
WITH (DATAFILETYPE='CHAR')
```

Resim 1.26: Bulk Insert komutunun kullanımı

- “SListe.txt” dosyası içerisinde var olan verilerin yeni tablonuz “bulk_tablo”ya aktarıldığını göreceksiniz.

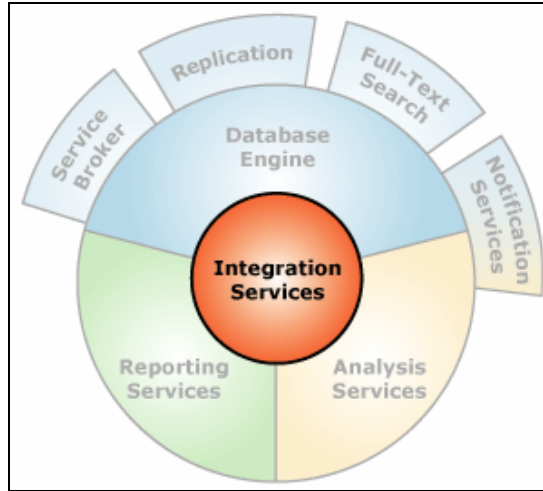
Table - dbo.bulk_tablo		Summary			
	num	ad	soy	cinsiyet	yas
▶	101	Hasan	KAYA	E	17
	103	Veli	CAN	E	16
	104	Ayşe	KOCAER	K	17
	106	Fatma	YILMAZ	K	18
	109	Defne	YAĞMUR	K	16
*	NULL	NULL	NULL	NULL	NULL

Resim 1.27: Bulk Insert ile verilerin tabloya aktarılmış hâli

1.4. Veri Transfer Hizmetleri

Veri transfer işlemi, veritabanı yönetim sistemlerinin önemli bir parçasıdır. Veri transfer hizmetleri, kişilerin yeni OLEDB veri kaynakları ve hedefler, görevler ve dönüşümler oluşturmasına izin verir. Bu hizmetle kullanıcı, veri transfer paketleri oluşturabilir ve çalıştırabilir.

Bu hizmet SQL Server 2005 ile yeniden oluşturularak SSIS (SQL Server Integration Services) adını almıştır. SSIS, veri bütünleştirme çözümleri geliştirmede kullanılır. Veri aktarma ve dışarıdaki veri kaynaklarından veri alma, temelli çözümler için kullanılabilen servistir.



Resim 1.28: Integration services

1.4.1. SQL Server 2000 Veritabanını SQL Server 2005'e Aktarmak

SQL Server 2000'deki kurulmuş olan bir sistemi SQL Server 2005'e aktarma işlemi iki şekilde gerçekleşir.

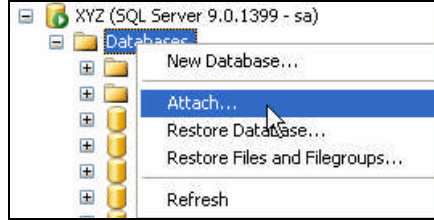
➤ Yan Yana Aktarım

Side by side Upgrade de denilen bu sistemde, eski sürümün yanına yeni bir sürüm kurularak eski sistemin devamının sağlanması ve yeni sistem üzerinden çalışabilir hale getirilmesidir.

➤ Yerinde Aktarım


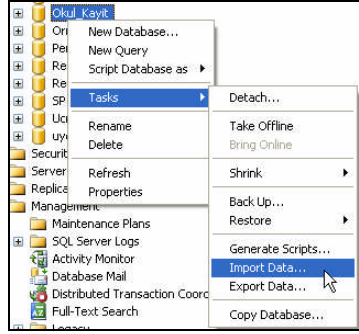
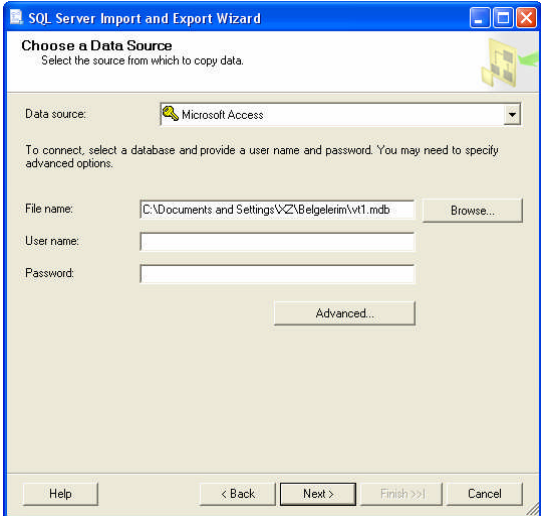
In-Place Upgrade de denilen bu sistemde, eski sistemin çalıştığı yere yeni sistemin kurulması demektir. Böylece, eski sistem tamamıyla kaldırıldığı için yeni sistemle çalışılabilir hale gelmektedir.

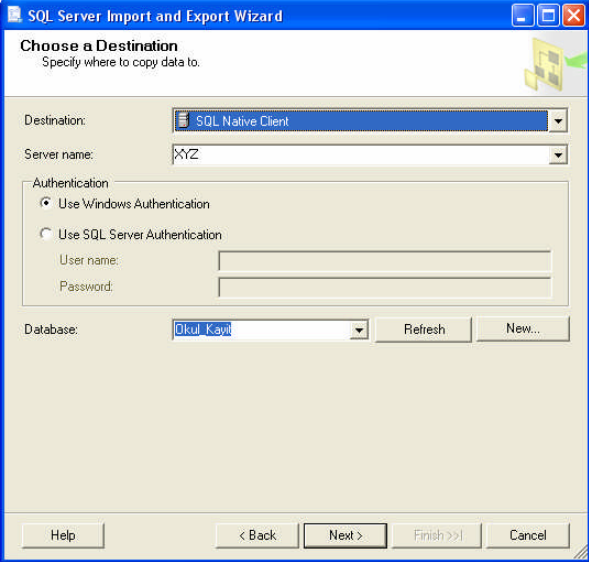
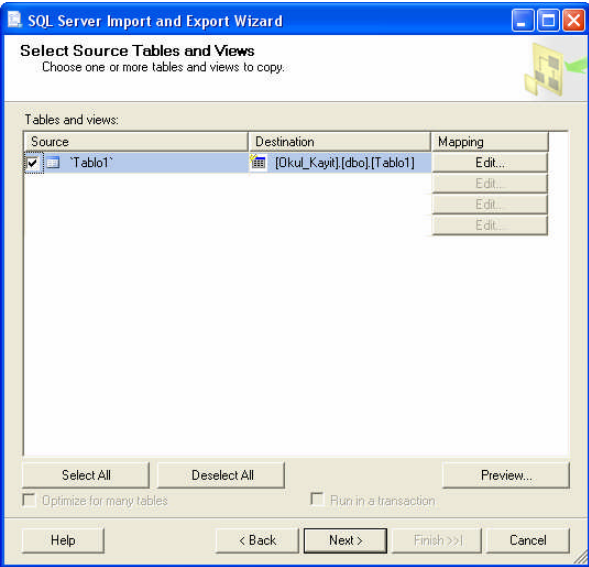
Eski sürümde oluşturulan dosyalarınızı, SQL Server 2005'in veri dosyalarının bulunduğu klasöre kopyalayarak Attach edebilirsiniz.



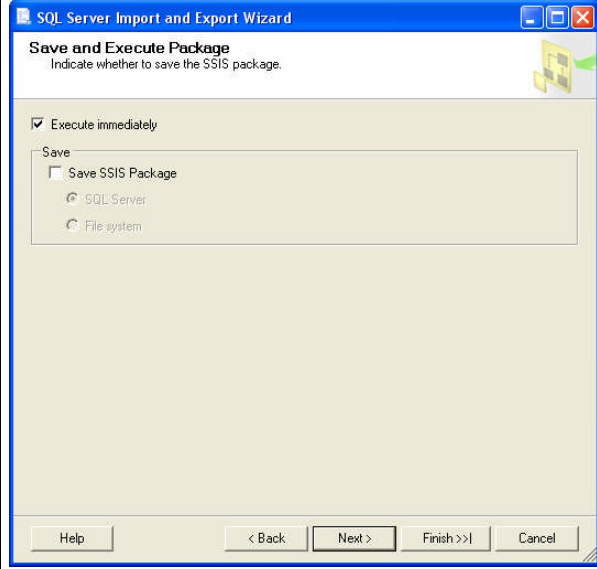
Resim 1.29: Attach işlemi

UYGULAMA FAALİYETİ 1

İşlem Basamakları	Öneriler
<p>➤ Microsoft Access programını açarak bir veritabanı ve bir tablo oluşturunuz.</p>	 <p>Resim 1.30: M.Access'ta oluşturulacak tablo</p>
<p>➤ Tabloyu kaydediniz.</p>	<p>➤ Tablo1 adında kaydedebilirsiniz.</p>
<p>➤ SQL Server Management Studio'da daha önceden oluşturduğunuz bir veritabanına Access tablosundaki verilerin alınması için veritabanı üzerinde sağ tıklayınız ve Import Data komutunu seçiniz.</p>	 <p>Resim 1.31: Import Data komutunun verilmesi</p>
<p>➤ Açılan sihirbaz ekranında Data Source olarak Microsoft Access'ı seçiniz ve File Name kutusuna dosya adını yazınız ve Browse düğmesine basarak dosyanın yolunu belirtiniz.</p>	 <p>Resim 1.32: Veri kaynağının ve dosyanın belirtilmesi</p>

<p>➤ Next düğmesini tıklayınız.</p>	
<p>➤ Verilerin kopyalanacağı hedefi (destination) seçiniz.</p>	<p>➤ SQL Native Client</p>
<p>➤ Verilerin kopyalanacağı veritabanını seçiniz ve Next düğmesini tıklayınız.</p>	 <p>Resim 1.33: Hedefin ve veritabanının seçilmesi</p>
<p>➤ Tablonun nasıl oluşacağını seçiniz ve Next düğmesini tıklayınız.</p>	<p>➤ “Copy data from one or more tables and views”</p>
<p>➤ Microsoft Access ile oluşturduğunuz tabloyu seçiniz ve Next düğmesini tıklayınız.</p>	 <p>Resim 1.34: Tablonun seçilmesi</p>

- Kayıt işlemi ve paketin çalıştırılacağı tercihlerinizi onaylayıp Next düğmesini tıklayınız.

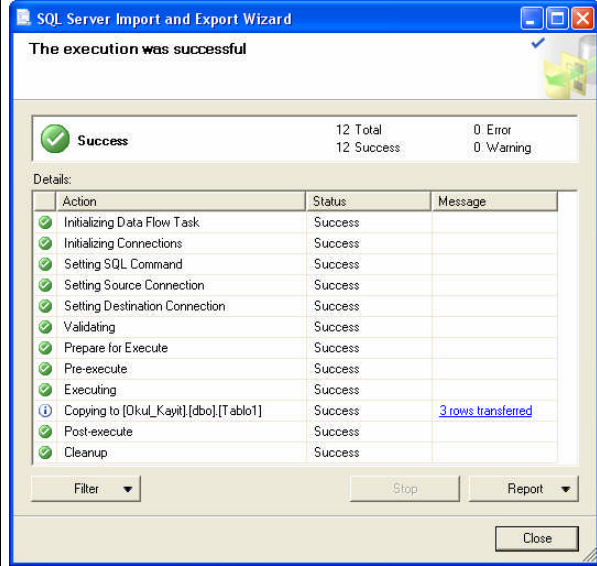


Resim 1.35: Hemen çalıştır seçimi

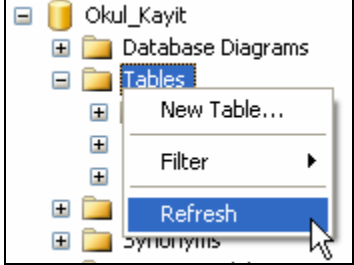
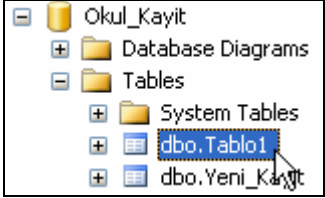
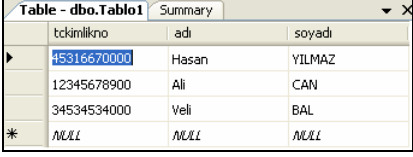
- Finish düğmesini tıklayarak sihirbazı tamamlayınız.





- Çalıştırılma sonucunda yapılan işlemlerin başarılı bir şekilde gerçekleşip gerçekleşmediğini kontrol ediniz ve Close düğmesini tıklayınız.



Resim 1.36: İşlemlerin doğruluğunun kontrolü

<p>➤ Veritabanınızdaki tablolara bakınız ve eklediğiniz tablo görünmüyorsa Refresh ediniz.</p>	 <p>Resim 1.37: Tabloların Refresh edilmesi</p>
<p>➤ Microsoft Access'tan Import ettiğiniz tablonun Tables dizini altında olup olmadığını kontrol ediniz.</p>	 <p>Resim 1.38: Access'ten Import edilen tablo</p>
<p>➤ Import edilen tabloyu Open Table ile açarak içeriğine bakınız.</p>	 <p>Resim 1.39: Tablo içeriği</p>
<p>➤ Farklı bir veritabanınızı SQL Server'dan Microsoft Access'a aktarınız.</p>	<p>➤ SQL Server Import and Export Wizard</p>

UYGULAMA FAALİYETİ 2

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"> ➤ Başlat/Çalıştır' ı tıklayınız. 	<ul style="list-style-type: none"> ➤
<ul style="list-style-type: none"> ➤ Cmd komutunu yazıp Enter'a basınız. 	 <p>Resim 1.40: Komut isteminin çalıştırılması</p>
<ul style="list-style-type: none"> ➤ Bir önceki uygulama faaliyetiyle veritabanınıza aktardığınız tablo içeriğini bcp komutuyla bir text dosyaya aktarınız. 	<ul style="list-style-type: none"> ➤ Bcp Okul_Kayit.dbo.Tablo1 out Kimlik.txt -c -T
<ul style="list-style-type: none"> ➤ Dosyanın konumunu bularak not defteri ile açınız. 	 <p>Resim 1.41: Bcp komutuyla oluşturulan text dosya içeriği</p>
<ul style="list-style-type: none"> ➤ Oluşturduğunuz text dosyayı var olan başka bir veritabanınıza aktarınız. 	<ul style="list-style-type: none"> ➤ Management Studio'daki veritabanınızda veri alacağınız tablonun var olması gerektiğini unutmayınız. ➤ Bcp Personel.dbo.Tablo1 in Kimlik.txt -c -T
<ul style="list-style-type: none"> ➤ Dışarıdan veri aldığınız tablonuzu açarak bilgilerinizi kontrol ediniz. 	<ul style="list-style-type: none"> ➤ Open Table

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. Dışarıdan veri alma işlemine, dışarı veri yollama işlemine denir.
2. Veri alıp verme işlemleri aşağıdaki programlar için gerçekleşmez?
A) Microsoft Access
B) Text File
C) Microsoft Excel
D) Microsoft Reader
3. Management Studio’da veri alıp verme işlemi için SQL Server Wizard kullanılır.
4. Excel çalışma sayfasından alınan veri içeriğini sihirbazı kullanırken görmek için Preview düğmesine tıklanır (D/Y).
5. Alınan verileri paket olarak saklamak için IIS kullanılır (D/Y).
6. Veri kaynağına, veri hedefine denir.
7. Komut istemiyle veri almak ve veri yollamak için komutu kullanılır.
8. Bcp ile veri almak için “out”, veri yollamak için “in” parametresi kullanılır (D/Y).
9. Bir Query ile veri aktarmak için komutu kullanılır.
10. SQL Server 2000’de var olan verileri SQL Server 2005’e aktarmak için servisi kullanılır (D/Y).

DEĞERLENDİRME

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

ÖĞRENME FAALİYETİ-2

AMAÇ

SQL Server’da ileri sorgu tekniklerini kullanmayı öğreneceksiniz.

ARAŞTIRMA

- Daha önceden öğrendiğiniz veritabanı paket programında tabloların nasıl birleştirildiğini araştırınız?

2. İLERİ SORGU TEKNİKLERİ

2.1. Inner Join ve Outer Join

2.1.1. Inner Join

Tablo birleştirmek için kullanılan yöntemdir. Her iki tabloda da eşleşen ortak alanların olması gerekir.

Yazılışı

```
.....  
FROM Tablo1 INNER JOIN Tablo2  
ON Tablo1.Alan_adi1 karşılaştırma operatörü Tablo2.Alan_adi2
```

- Tablo1 ve Tablo2, kayıtların birleştirileceği tablo adlarını ifade eder.
- Alan1 ve Alan2, birleştirilecek alan adlarını belirtir.
- Karşılaştırma operatörü genelde “=” işaretidir. Ama diğer karşılaştırma operatörleri de kullanılabilir.

Birleştirme işlemi sadece iki tablo için kısıtlı değildir. En fazla 256 adet tabloyu JOIN işlemine tabi tutabilirsiniz.

Örnek

```
SELECT PERSONEL_AD,YAKIN_AD  
FROM PERSONEL INNER JOIN PERSONELYAKIN  
ON PERSONEL.PERSONEL_ID=PERSONELYAKIN.PERSONEL_ID
```

Bu sorgu, PERSONEL ve PERSONELYAKIN tablolarındaki alanlardan PERSONEL_ID alanı eşit olmak şartıyla PERSONEL_AD ve YAKIN_Ad alanlarını göstermektedir.

Eğer, SELECT ifadesiyle “*” kullanılırsa tablolardaki tüm alanlar gösterilir.

2.1.2. Outer Join

Inner Join, iki veya daha fazla tablodaki ilişkili bulunan satırları listelerken Outer Join, tabloda ilişkili olmayan satırları da gösterecektir.

Bu demektir ki tablodaki tüm satırlar gösterilecektir. Ancak, ilişkisi olmayan satırlarda değer NULL olarak gösterilir.

Yazılışı

.....

```
FROM Tablo1 Join-Tipi OUTER JOIN Tablo2  
ON Tablo1.Alanadı1 = Tablo2.Alanadı2
```

Kullanışlarına göre Join-Tipi farklılık göstermektedir. 3 tip Join-Tipi vardır.

- **LEFT OUTER JOIN:** Bu kelimenin solundaki tablo, belirleyici tablodur ve bu kullanım, bir ilişkisi bulunsun veya bulunmasın birinci tablodaki bütün satırları listeleyecektir. İkinci tabloda ise ilişkili satırlar listelenecektir. Eğer bir ilişki bulunmuyorsa sağdaki tablo verileri NULL olarak gösterilir.

```
SELECT PERSONEL.*, PERSONELYAKIN.*  
FROM PERSONEL LEFT OUTER JOIN PERSONELYAKIN  
ON PERSONEL.PERSONEL_ID = PERSONELYAKIN.PERSONEL_ID
```

- **RIGHT OUTER JOIN:** Bu kelimenin sağındaki tablo belirleyici tablodur ve bu kullanım bir ilişkiye ait olsun veya olmasın ikinci tablodaki bütün satırları gösterecektir. Birinci tabloda ise ilişkili satırlar listelenecektir. Eğer bir ilişki bulunmuyorsa soldaki tablo verileri NULL olarak listelenir.

```
SELECT PERSONEL.PERSONEL_AD, PERSONELYAKIN.YAKIN_AD  
FROM PERSONELYAKIN RIGHT OUTER JOIN PERSONEL  
ON PERSONEL.PERSONEL_ID=PERSONELYAKIN.PERSONEL_ID
```

- **FULL OUTER JOIN:** Left ve Right Outer Join’de karşılıklı olarak eşit satırı olmayan satırlar da listelenir. FULL OUTER JOIN, LEFT ve RIGHT OUTER JOIN’in birleşim kümesidir.

```
SELECT PERSONEL.PERSONEL_AD, PERSONELYAKIN.YAKIN_AD  
FROM PERSONEL FULL OUTER JOIN PERSONELYAKIN  
ON PERSONEL.PERSONEL_ID=PERSONELYAKIN.PERSONEL_ID
```

2.2. Alt Sorgular

SQL Server 2005'te sorgu içinde sorgu da oluşturulabilir. Sorgu içinde sorgu, içteki sorgunun dışta olan sorguya değer üretmesidir. Bu, bir değer veya birden fazla değer olabilir.

2.2.1. IN ve NOT IN

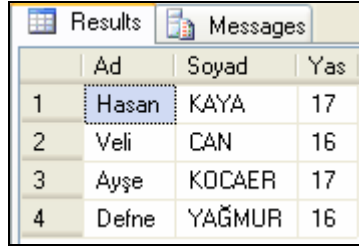
Tablodaki alan içeriklerine ulaşmak için IN deyimini kullanılır.

Örnek

Bir sınıftaki 16 ve 17 yaşındaki öğrencileri listelemek istediğinizi düşünün. IN deyimiyile sorgunun yazımı şu şekildedir:

```
SELECT Ad,Soyad,Yas
FROM Sinif
WHERE Yas IN(16,17)
```

Sorguyu çalıştırıp sonuçlarını Results ekranından görebilirsiniz.



	Ad	Soyad	Yas
1	Hasan	KAYA	17
2	Veli	CAN	16
3	Ayşe	KOCAER	17
4	Defne	YAĞMUR	16

Resim 2.1: IN deyimini kullanılarak yazılan sorgunun sonuç ekranı

Örnek

Sınıftaki öğrencilerden 16 yaşında olmayan öğrencileri listelemek istediğinizde,

```
SELECT Ad,Soyad,Yas
FROM Sinif
WHERE Yas NOT IN(16)
```

şeklinde bir sorgu yazmalısınız.

2.2.2. İç İçe SELECT

Hazırladığınız sorguyu karmaşıklıktan kurtarabilmeniz veya sorgunun işlevselliğini test edebilmeniz açısından parçalara ayırmanız gerekebilir. Bunun için iç içe SELECT kullanmanız gerekir. İç içe yazılan her SELECT sorgusu parantez içinde yazılmalıdır.

- Yazdığınız SELECT ifadesi geriye sadece bir değer döndürüyorsa buna tekil alt sorgu denir. T-SQL’de kullanılan matematiksel fonksiyonlar (MIN, MAX, SUM vb.) veya WHERE ifadesiyle yazılan bir şart ile geriye tek bir değer döndürebilirsiniz.

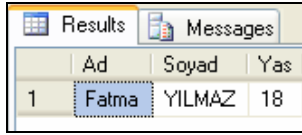
Örnek

Yaşı 18 olan öğrenciyi bulmak istediğinizde,

```
SELECT Ad,Soyad,Yas
FROM Sinif
WHERE Yas= ( SELECT Yas
              FROM Sinif
              WHERE Yas=18
            )
```

şeklinde iç içe SELECT yazabilirsiniz.

Sorguyu çalıştırıp sonuçlarını Results ekranından görebilirsiniz.



	Ad	Soyad	Yas
1	Fatma	YILMAZ	18

Resim 2.2: İç içe SELECT kullanımının sonuç ekranı

- WHERE’den sonra sorguları bağlamak için kullanılan “=” operatörü yerine diğer karşılaştırma operatörleri de kullanılabilir. Eğer, sorgunuzda geriye birden fazla değer döndürülüyorsa SQL Server hata verir.
- Hazırlanan alt sorgu asıl SELECT ifadesinin içinde de kullanılabilir.

Örnek

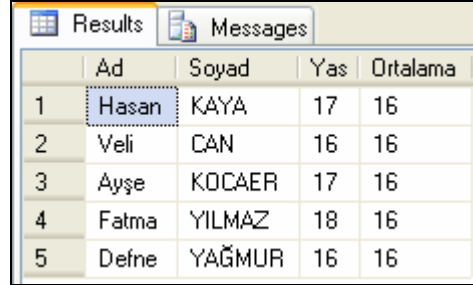
Öğrencilerin ad, soyad ve ortalama yaşlarını veren bir sorgu yazılacağını varsayınız.

Sorgunun nasıl yazıldığına dikkat ederek Resim 2.3’teki gibi sorgu Query Editor’e yazınız.

```
SELECT Ad,Soyad,Yas,Ortalama=(
                                SELECT AVG(Yas)
                                FROM Sinif
                                )
FROM Sinif
```

Resim 2.3: Sorgunun Query Editor’e yazılışı

Sorguyu çalıştırıp sonuçlarını Results penceresinde görebilirsiniz.



	Ad	Soyad	Yas	Ortalama
1	Hasan	KAYA	17	16
2	Veli	CAN	16	16
3	Ayşe	KOCAER	17	16
4	Fatma	YILMAZ	18	16
5	Defne	YAĞMUR	16	16

Resim 2.4: Sonuç penceresi

- Yazdığınız sorgu birden fazla değer döndürüyorsa sorguları IN deyimiyile birbirine bağlamanız gerekir.

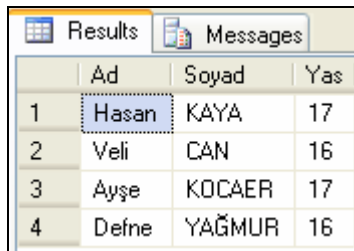
Örnek

Yaşı 18'den küçük olan öğrencileri listeleyeceğinizi varsayınız.

```
SELECT Ad, Soyad, Yas
FROM Sinif
WHERE Yas IN (
                SELECT Yas FROM Sinif WHERE Yas<18
            )
```

Resim 2.5: Çoklu değer sorgusu

Yine sonucu Results penceresinden görebilirsiniz.



	Ad	Soyad	Yas
1	Hasan	KAYA	17
2	Veli	CAN	16
3	Ayşe	KOCAER	17
4	Defne	YAĞMUR	16

Resim 2.6: Çoklu değer sorgusu sonuç penceresi

2.2.3. Türetilmiş Tablolar

Türetilmiş tablo, alt sorgularla çalışmanın özel bir hâlidir. İç içe sorgu yaparken elde edilen değerler hep bir sütun hâlinde gösterilmiştir. İç içe sorgular, bir sorgu sonucunun tabloyu gibi tekrar sorgulanmasını da sağlar.

Türetilmiş tablo, FROM ifadesinden sonraki SELECT ifadesinin parantez içine yazılıp bir takma isim verilmesiyle oluşturulur.

Yazılışı

SELECT ifade
FROM (SELECT ifade) AS takmaad

Örnek

Yaşı 16'dan büyük olan öğrencilerin cinsiyetleriyle beraber listeleneceğini varsayınız.

Sorguyu Resim 2.7'deki gibi yazınız.

```
SELECT Sayi.*  
FROM(  
    SELECT Yas,Cinsiyet  
    FROM Sinif  
)  
Sayi( Yasi , Cinsi )  
WHERE Sayi.Yasi > 16
```

Resim 2.7: Türetilmiş tablo örnek sorgusu

Sorguda, “Sayi” için türetilmiş tablo denir. Results penceresindeki sütun adlarının başlıkları da Sayi (Yasi,Cinsi) satırıyla verilmiş olur.

Sorguyu çalıştırdığınızda Results penceresinin görünümü Resim 2.8'deki gibi olur.



	Yasi	Cinsi
1	17	E
2	17	K
3	18	K

Resim 2.8. Results penceresi

Örnek

Yaşları aynı olan öğrencilerin sayılarını listeleyeceğinizi varsayınız. Önce tablodaki tüm kayıtları listeleyip öğrenci yaşlarını görünüz.

```

SELECT * FROM Sinif

SELECT MAX(grup.adet)
FROM ( SELECT Yas, COUNT (*) as adet
      FROM Sinif
      GROUP BY Yas
    ) grup

```

Resim 2.9: Başka bir türetilmiş tablo örnek sorgusu

Sorguyu çalıştırdığınızda Results penceresinin görünümü Resim 2.10'daki gibi olur.

Ogr_No	Ad	Soyad	Cinsiyet	Yas
101	Hasan	KAYA	E	17
103	Veli	CAN	E	16
104	Ayşe	KOCAER	K	17
106	Fatma	YILMAZ	K	17
109	Defne	YAĞMUR	K	16

(No column name)
3

Resim 2.10: Results penceresi

2.2.4. EXISTS ve NOT EXISTS

EXISTS kullanıldığında, içerideki sorguda, bir veya daha fazla kayıt dönerse dışarıdaki sorgu çalıştırılır. Kendisinden sonra gelen sorgu cümlesi ile eşleşen en az bir kayıt mevcut ise "True" mantıksal değerinin üretilmesini sağlar. Kendisiyle birlikte AND, OR ve NOT mantıksal operatörleri kullanılabilir. Hiç kayıt dönmezse, dışarıdaki sorgu çalıştırılmaz. "False" mantıksal değeri üretilir. NOT EXISTS ise içerideki sorgunun sonucunda sıfır kayıt dönüyorsa dışarıdaki sorgunun çalıştırılması için kullanılır.

Örnek

PERSONEL ve PERSONELYAKIN tablosu kullanılarak PERSONEL tablosundaki personellerden en az bir yakını olanların listesini bulacağınızı varsayınız.

```

SELECT PERSONEL_ID,PERSONEL_AD,PERSONEL_SOYAD
FROM PERSONEL
WHERE EXISTS (
    SELECT *
    FROM PERSONELYAKIN
    WHERE PERSONELYAKIN.YAKIN_ID=PERSONEL.PERSONEL_ID
)

```

	PERSONEL_ID	PERSONEL_AD	PERSONEL_SOYAD
1	1	ALI	CAN
2	2	VELİ	YAMAN
3	3	AHMET	PINAR

Resim 2.11: EXISTS kullanımı

Örnek

Hiç yakını olmayan personeli listeleyeceğinizi varsayınız.

```

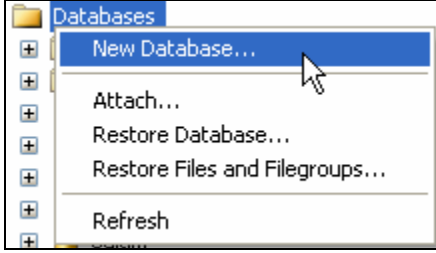
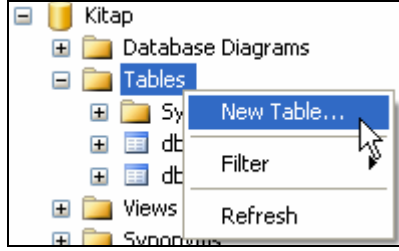
SELECT PERSONEL_ID,PERSONEL_AD,PERSONEL_SOYAD
FROM PERSONEL
WHERE NOT EXISTS (
    SELECT *
    FROM PERSONELYAKIN
    WHERE PERSONELYAKIN.YAKIN_ID=PERSONEL.PERSONEL_ID
)

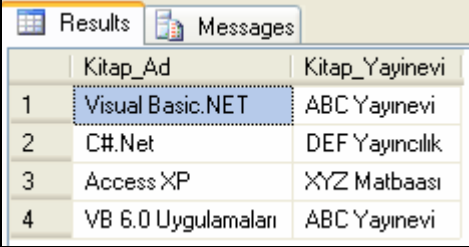
```

	PERSONEL_ID	PERSONEL_AD	PERSONEL_SOYAD
1	4	KEMAL	TURK

Resim 2.12: NOT EXISTS kullanımı

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler																																													
<p>➤ “Kitap” adında yeni bir veritabanı oluşturunuz.</p>	 <p>Resim 2.13: Yeni veritabanının oluşturulması</p>																																													
<p>➤ “tablo_Kitap” adında yeni bir tabloyu “Kitap” veritabanında oluşturunuz.</p>	 <p>Resim 2.14: Yeni tablonun oluşturulması</p>																																													
<p>➤ “tablo_Kitap” tablosunun alan adlarını “Kitap_Id”, “Kitap_Ad”, “Kitap_Yazar”, “Kitap_Sayfa” olacak şekilde belirleyiniz ve verilerini giriniz.</p>	<table border="1"> <thead> <tr> <th colspan="5">Table - dbo.tablo_Kitap</th> </tr> <tr> <th></th> <th>Kitap_Id</th> <th>Kitap_Ad</th> <th>Kitap_Yazar</th> <th>Kitap_Sayfa</th> </tr> </thead> <tbody> <tr> <td></td> <td>123</td> <td>Visual Basic.NET</td> <td>Ali CAN</td> <td>550</td> </tr> <tr> <td></td> <td>128</td> <td>C#.Net</td> <td>Veli BAL</td> <td>700</td> </tr> <tr> <td></td> <td>150</td> <td>Access XP</td> <td>Osman GÜR</td> <td>450</td> </tr> <tr> <td></td> <td>200</td> <td>VB 6.0 Uygulam...</td> <td>Sena PINAR</td> <td>330</td> </tr> <tr> <td></td> <td>250</td> <td>C++ Uygulamaları</td> <td>Yasemin ZEKİ</td> <td>270</td> </tr> <tr> <td></td> <td>251</td> <td>Delphi 7.0</td> <td>Yağmur AK</td> <td>310</td> </tr> <tr> <td>▶*</td> <td>NULL</td> <td>NULL</td> <td>NULL</td> <td>NULL</td> </tr> </tbody> </table> <p>Resim 2.15: Tablonun alan adlarının oluşturulması ve verilerinin girilmesi</p>	Table - dbo.tablo_Kitap						Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_Sayfa		123	Visual Basic.NET	Ali CAN	550		128	C#.Net	Veli BAL	700		150	Access XP	Osman GÜR	450		200	VB 6.0 Uygulam...	Sena PINAR	330		250	C++ Uygulamaları	Yasemin ZEKİ	270		251	Delphi 7.0	Yağmur AK	310	▶*	NULL	NULL	NULL	NULL
Table - dbo.tablo_Kitap																																														
	Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_Sayfa																																										
	123	Visual Basic.NET	Ali CAN	550																																										
	128	C#.Net	Veli BAL	700																																										
	150	Access XP	Osman GÜR	450																																										
	200	VB 6.0 Uygulam...	Sena PINAR	330																																										
	250	C++ Uygulamaları	Yasemin ZEKİ	270																																										
	251	Delphi 7.0	Yağmur AK	310																																										
▶*	NULL	NULL	NULL	NULL																																										
<p>➤ “tablo_KitapBilgi” adında yeni bir tabloyu “Kitap” veritabanında oluşturunuz.</p>	<p>➤ “New Table...” komutu</p>																																													

<p>➤ “tablo_KitapBilgi” tablosunun alan adlarını “Kitap_Id”, “Kitap_Yayınevi”, “Kitap_ISBNNo”, “Kitap_BasımYılı” olacak şekilde belirleyiniz ve verilerini giriniz.</p>	<table border="1" data-bbox="649 333 1262 540"> <thead> <tr> <th colspan="4">Table - dbo.tablo_KitapBilgi</th> </tr> <tr> <th>Kitap_Id</th> <th>Kitap_Yayınevi</th> <th>Kitap_ISBNNo</th> <th>Kitap_BasımYılı</th> </tr> </thead> <tbody> <tr> <td>123</td> <td>ABC Yayınevi</td> <td>1234567890</td> <td>2006</td> </tr> <tr> <td>128</td> <td>DEF Yayıncılık</td> <td>5678901234</td> <td>2007</td> </tr> <tr> <td>150</td> <td>XYZ Matbaası</td> <td>3456789012</td> <td>2007</td> </tr> <tr> <td>200</td> <td>ABC Yayınevi</td> <td>1234567123</td> <td>2006</td> </tr> <tr> <td>➤*</td> <td>NULL</td> <td>NULL</td> <td>NULL</td> </tr> </tbody> </table> <p style="text-align: center;">Resim 2.16: Tablonun alan adlarının oluşturulması ve verilerinin girilmesi</p>	Table - dbo.tablo_KitapBilgi				Kitap_Id	Kitap_Yayınevi	Kitap_ISBNNo	Kitap_BasımYılı	123	ABC Yayınevi	1234567890	2006	128	DEF Yayıncılık	5678901234	2007	150	XYZ Matbaası	3456789012	2007	200	ABC Yayınevi	1234567123	2006	➤*	NULL	NULL	NULL																												
Table - dbo.tablo_KitapBilgi																																																									
Kitap_Id	Kitap_Yayınevi	Kitap_ISBNNo	Kitap_BasımYılı																																																						
123	ABC Yayınevi	1234567890	2006																																																						
128	DEF Yayıncılık	5678901234	2007																																																						
150	XYZ Matbaası	3456789012	2007																																																						
200	ABC Yayınevi	1234567123	2006																																																						
➤*	NULL	NULL	NULL																																																						
<p>➤ “Kitap_Id” alanları eşleşecek şekilde “Kitap_Ad” ve “Kitap_Yayınevi” alanları için INNER JOIN yöntemini uygulayacak sorguyu yazınız.</p>	<pre data-bbox="649 692 1262 779">SELECT Kitap_Ad,Kitap_Yayınevi FROM tablo_Kitap INNER JOIN tablo_KitapBilgi ON tablo_Kitap.Kitap_Id=tablo_KitapBilgi.Kitap_Id</pre> <p style="text-align: center;">Resim 2.17: INNER JOIN yöntemi</p>																																																								
<p>➤ Sonucu Results penceresinde görünüz.</p>	 <p style="text-align: center;">Resim 2.18: Sonucun görüntülenmesi</p>																																																								
<p>➤ Birinci tablodaki kayıtlara göre ikinci tablodaki ilişkili kayıtları gösterecek OUTER JOIN yöntemini uygulayacak sorguyu yazınız.</p>	<pre data-bbox="649 1278 1262 1365">SELECT tablo_Kitap.*, tablo_KitapBilgi.* FROM tablo_Kitap LEFT OUTER JOIN tablo_KitapBilgi ON tablo_Kitap.Kitap_Id = tablo_KitapBilgi.Kitap_Id</pre> <p style="text-align: center;">Resim 2.19: LEFT OUTER JOIN yöntemi</p>																																																								
<p>➤ Sonucu Results penceresinde görünüz.</p>	<table border="1" data-bbox="649 1491 1262 1657"> <thead> <tr> <th></th> <th>Kitap_Id</th> <th>Kitap_Ad</th> <th>Kitap_Yazar</th> <th>Kitap_Sayfa</th> <th>Kitap_Id</th> <th>Kitap_Yayınevi</th> <th>Kitap_ISBNNo</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>123</td> <td>Visual Basic.NET</td> <td>Ali CAN</td> <td>550</td> <td>123</td> <td>ABC Yayınevi</td> <td>1234567890</td> </tr> <tr> <td>2</td> <td>128</td> <td>C#.Net</td> <td>Veli BAL</td> <td>700</td> <td>128</td> <td>DEF Yayıncılık</td> <td>5678901234</td> </tr> <tr> <td>3</td> <td>150</td> <td>Access XP</td> <td>Osman GÜR</td> <td>450</td> <td>150</td> <td>XYZ Matbaası</td> <td>3456789012</td> </tr> <tr> <td>4</td> <td>200</td> <td>VB 6.0 Uygula...</td> <td>Sena PINAR</td> <td>330</td> <td>200</td> <td>ABC Yayınevi</td> <td>1234567123</td> </tr> <tr> <td>5</td> <td>250</td> <td>C++ Uygulamaları</td> <td>Yasemin Z...</td> <td>270</td> <td>NULL</td> <td>NULL</td> <td>NULL</td> </tr> <tr> <td>6</td> <td>251</td> <td>Delphi 7.0</td> <td>Yağmur AK</td> <td>310</td> <td>NULL</td> <td>NULL</td> <td>NULL</td> </tr> </tbody> </table> <p style="text-align: center;">Resim 2.19: Sonucun görüntülenmesi</p>		Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_Sayfa	Kitap_Id	Kitap_Yayınevi	Kitap_ISBNNo	1	123	Visual Basic.NET	Ali CAN	550	123	ABC Yayınevi	1234567890	2	128	C#.Net	Veli BAL	700	128	DEF Yayıncılık	5678901234	3	150	Access XP	Osman GÜR	450	150	XYZ Matbaası	3456789012	4	200	VB 6.0 Uygula...	Sena PINAR	330	200	ABC Yayınevi	1234567123	5	250	C++ Uygulamaları	Yasemin Z...	270	NULL	NULL	NULL	6	251	Delphi 7.0	Yağmur AK	310	NULL	NULL	NULL
	Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_Sayfa	Kitap_Id	Kitap_Yayınevi	Kitap_ISBNNo																																																		
1	123	Visual Basic.NET	Ali CAN	550	123	ABC Yayınevi	1234567890																																																		
2	128	C#.Net	Veli BAL	700	128	DEF Yayıncılık	5678901234																																																		
3	150	Access XP	Osman GÜR	450	150	XYZ Matbaası	3456789012																																																		
4	200	VB 6.0 Uygula...	Sena PINAR	330	200	ABC Yayınevi	1234567123																																																		
5	250	C++ Uygulamaları	Yasemin Z...	270	NULL	NULL	NULL																																																		
6	251	Delphi 7.0	Yağmur AK	310	NULL	NULL	NULL																																																		

<p>➤ “tablo_Kitap” tablosunda mevcut kitaplardan sayfa sayısı 450 ve 550 olan kitapları listeleyen sorguyu yazınız.</p>	<pre>SELECT * FROM tablo_Kitap WHERE Kitap_Sayfa IN (450,550)</pre> <p>Resim 2.20: IN deyimi kullanımı</p>																																								
<p>➤ “tablo_Kitap” tablosunda mevcut kitaplardan sayfa sayısı 310 ve 330 olmayan kitapları listeleyen sorguyu yazınız.</p>	<pre>SELECT * FROM tablo_Kitap WHERE Kitap_Sayfa NOT IN (310,330)</pre> <p>Resim 2.21: NOT IN deyimi kullanımı</p>																																								
<p>➤ Her iki sorgu sonucunu Results penceresinde görünüz.</p>	<table border="1"> <thead> <tr> <th></th> <th>Kitap_Id</th> <th>Kitap_Ad</th> <th>Kitap_Yazar</th> <th>Kitap_Sayfa</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>123</td> <td>Visual Basic.NET</td> <td>Ali CAN</td> <td>550</td> </tr> <tr> <td>2</td> <td>150</td> <td>Access XP</td> <td>Osman GÜR</td> <td>450</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th></th> <th>Kitap_Id</th> <th>Kitap_Ad</th> <th>Kitap_Yazar</th> <th>Kitap_Sayfa</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>123</td> <td>Visual Basic.NET</td> <td>Ali CAN</td> <td>550</td> </tr> <tr> <td>2</td> <td>128</td> <td>C#.Net</td> <td>Veli BAL</td> <td>700</td> </tr> <tr> <td>3</td> <td>150</td> <td>Access XP</td> <td>Osman G...</td> <td>450</td> </tr> <tr> <td>4</td> <td>250</td> <td>C++ Uygulamaları</td> <td>Yasemin ...</td> <td>270</td> </tr> </tbody> </table> <p>Resim 2.22: IN ve NOT IN deyimlerinin sorgu sonucu</p>		Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_Sayfa	1	123	Visual Basic.NET	Ali CAN	550	2	150	Access XP	Osman GÜR	450		Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_Sayfa	1	123	Visual Basic.NET	Ali CAN	550	2	128	C#.Net	Veli BAL	700	3	150	Access XP	Osman G...	450	4	250	C++ Uygulamaları	Yasemin ...	270
	Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_Sayfa																																					
1	123	Visual Basic.NET	Ali CAN	550																																					
2	150	Access XP	Osman GÜR	450																																					
	Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_Sayfa																																					
1	123	Visual Basic.NET	Ali CAN	550																																					
2	128	C#.Net	Veli BAL	700																																					
3	150	Access XP	Osman G...	450																																					
4	250	C++ Uygulamaları	Yasemin ...	270																																					
<p>➤ “tablo_Kitap” tablosundan “Kitap_Id”, “Kitap_Ad”, “Kitap_Yazar” alanlarını seçerek sayfa sayısı 450’den büyük olan kitapları gösteren iç içe sorguyu yazınız.</p>	<pre>SELECT Kitap_Id,Kitap_Ad,Kitap_Yazar FROM tablo_Kitap WHERE Kitap_Sayfa IN(SELECT Kitap_Sayfa FROM tablo_Kitap WHERE (Kitap_Sayfa>450))</pre> <p>Resim 2.23: İç içe sorgu</p>																																								
<p>➤ Sorgunun sonucunu Results penceresinde görünüz.</p>	<table border="1"> <thead> <tr> <th></th> <th>Kitap_Id</th> <th>Kitap_Ad</th> <th>Kitap_Yazar</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>123</td> <td>Visual Basic.NET</td> <td>Ali CAN</td> </tr> <tr> <td>2</td> <td>128</td> <td>C#.Net</td> <td>Veli BAL</td> </tr> </tbody> </table> <p>Resim 2.24: İç içe sorgu sonucu</p>		Kitap_Id	Kitap_Ad	Kitap_Yazar	1	123	Visual Basic.NET	Ali CAN	2	128	C#.Net	Veli BAL																												
	Kitap_Id	Kitap_Ad	Kitap_Yazar																																						
1	123	Visual Basic.NET	Ali CAN																																						
2	128	C#.Net	Veli BAL																																						

<p>➤ “tablo_Kitap” tablosunda Kitap Id’si 130’dan büyük olan kitapları listeleyen türetilmiş tablo sorgusunu yazınız.</p>	<pre>SELECT Sayfa.* FROM (SELECT Kitap_Id,Kitap_Sayfa FROM tablo_Kitap) Sayfa(Kitap_Id,Kitap_Sayfa) WHERE Sayfa.Kitap_Id>130</pre> <p>Resim 2.25: Türetilmiş tablo sorgusu</p>															
<p>➤ Sorgunun sonucunu Results penceresinde görünüz.</p>	<table border="1"> <thead> <tr> <th></th> <th>Kitap_Id</th> <th>Kitap_Sayfa</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>150</td> <td>450</td> </tr> <tr> <td>2</td> <td>200</td> <td>330</td> </tr> <tr> <td>3</td> <td>250</td> <td>270</td> </tr> <tr> <td>4</td> <td>251</td> <td>310</td> </tr> </tbody> </table> <p>Resim 2.26: Sorgu sonucu</p>		Kitap_Id	Kitap_Sayfa	1	150	450	2	200	330	3	250	270	4	251	310
	Kitap_Id	Kitap_Sayfa														
1	150	450														
2	200	330														
3	250	270														
4	251	310														
<p>➤ Kitap bilgisi olmayan kitapları listeleyen sorguyu yazınız.</p>	<pre>SELECT Kitap_Id,Kitap_Ad,Kitap_Yazar FROM tablo_Kitap WHERE NOT EXISTS (SELECT * FROM tablo_KitapBilgi WHERE tablo_KitapBilgi.Kitap_Id=tablo_Kitap.Kitap_Id)</pre> <p>Resim 2.27: NOT EXISTS kullanımı</p>															
<p>➤ Sorgunun sonucunu Results penceresinde görünüz.</p>	<table border="1"> <thead> <tr> <th></th> <th>Kitap_Id</th> <th>Kitap_Ad</th> <th>Kitap_Yazar</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>250</td> <td>C++ Uygulamaları</td> <td>Yasemin.ZEKİ</td> </tr> <tr> <td>2</td> <td>251</td> <td>Delphi 7.0</td> <td>Yağmur AK</td> </tr> </tbody> </table> <p>Resim 2.28. Sorgu sonucu</p>		Kitap_Id	Kitap_Ad	Kitap_Yazar	1	250	C++ Uygulamaları	Yasemin.ZEKİ	2	251	Delphi 7.0	Yağmur AK			
	Kitap_Id	Kitap_Ad	Kitap_Yazar													
1	250	C++ Uygulamaları	Yasemin.ZEKİ													
2	251	Delphi 7.0	Yağmur AK													

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. Inner Join ile tabloları birleştirirken eşleşen ortak alan olup olmadığına bakılmaz (D/Y).
2. Tablo birleştirme işleminde en fazla adet tablo birleştirilebilir.
3. Bir tabloda ilişkili olmayan satırlar ile listelenebilir. Ancak, değerler olarak gösterilir.
4. FULL OUTER JOIN,'in ve 'in birleşim kümesidir.
5. Aşağıdakilerden hangisi yazılan SELECT ile geriye bir tane değer döndüren sorgudur?
A) Çoğul alt sorgu
B) Tekli alt sorgu
C) Tekil alt sorgu
D) Tekil sorgu
6. Aşağıdaki deyimlerden hangisi SELECT ile geriye dönen değer birden fazla ise sorguları birbirine bağlar?
A) SELECT B) EXISTS C) IN D) NOT IN
7. Sorgu içinde sorgu, içteki sorgunun dıştaki sorgu için değer üretmesidir (D/Y).
8. Türetilmiş tablonun bir takma ismi olur (D/Y).
9. İçerideki sorguda, bir veya daha fazla kayıt dönerse dışarıdaki sorgu çalıştırılmaz (D/Y).
10. EXISTS deyimi mantıksal değerler olan TRUE veya FALSE değerlerinden birini üretir (D/Y).

DEĞERLENDİRME

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

ÖĞRENME FAALİYETİ-3

AMAÇ

SQL Server’da imleç kullanmayı öğreneceksiniz.

ARAŞTIRMA

- Kelime işlemci programında imlecin nasıl davrandığını inceleyiniz.

3. İMLEÇ

SQL Server’daki imleç (cursor), bir metin editöründeki imleçlerle aynı görevi yapar. Metin editöründe çalışma anında, imleç nerede bulunuyorsa oradaki verileri temel alan işlemler yapılabilir. SQL Server’daysa imlecin bulunduğu yerdeki verilere satır satır erişilebilir. Veritabanı ortamında bir seçme sorgusu sonucu erişilen kayıtlara bir döngü çerçevesinde tek tek erişim sağlayabilmek için imleç programlamak gerekir. İmleçler, veritabanlarında saklanmaz.

İmleçler performans açısından çok tercih edilmemektedir. Bu yüzden gerekmedikçe imleç kullanılmamalıdır. CASE yapısı ya da geçici tablolar kullanarak imleç kullanmadan aynı sonuçlar elde edilebilmektedir. İmleçler, satır satır hareket etmenize olanak sağlayarak kimi zaman işinizi çok kolaylaştırırsalar da çoğu zaman performans kaybına sebep olurlar. Bu yüzden, gereksiz yere imleç kullanmaya çalışmayınız.

3.1. Kullanım Amaçları

İmleçler;

- Bir sonuç setinde o anda bulunulan pozisyondaki satır veya takip eden satırlardaki verilere ulaşmak,
- Yapılan bir sorgu sonucu geri dönen değerlerden her satırı ayrı ayrı değerlendirip duruma göre verileri değiştirmek,
- Tetikleyici (trigger) veya Stored Procedure’lerin bir sonuç setine satır satır ulaşmasını sağlamak,
- Diğer kullanıcılar tarafından yapılan değişikliklerin görünebilirlik seviyesini ortak zamanlı çalışmada ayarlamak için kullanılırlar.

3.2. T-SQL İmleçler

T-SQL imleçler sonucu tarafından çalışır ve sadece bir sonuç seti (resultset) üzerinde tanımlanırlar. Bu şekilde tanımlanan bir imleçte, kayıt seçmek için yazılan SELECT sorgusunda COMPUTE, COMPUTE BY, INTO deyimleri bulunmaz.

3.2.1. T-SQL İmleç Kullanımı

Beş adımdan oluşur:

- İmleç değişken tanımlama: Bir SELECT ifadesinde, bir değişken tanımlanarak seçme işlemi sonucunda elde edilecek değeri tutacak imleç tanımlanmış olur.
- İmleci açma: SELECT işleminden sonra imleç ilk satırı işaret edecek şekilde ayarlanır.
- Erişim ile ilgili işlemler: FETCH deyimi kullanılarak, bir döngüyle satırlara erişilir. Değişiklik ve okuma yapılabilir.
- İmleci kapatma: CLOSE deyimiyle imleç kapatılır. Tekrar açılabilir.
- İmleci bellekten silme: DEALLOCATE deyimi imleci bellekten siler.

3.2.2. @@FETCH_STATUS ve @@ROWCOUNT

Yapılan işlemler sonucunda imlecin son satıra gelip gelmediğini anlamak için @@FETCH_STATUS ve @@ROWCOUNT fonksiyonları kullanılır.

FETCH_STATUS fonksiyonu, en son çalıştırılan FETCH komutunun sonucu hakkında bilgi verir.

FETCH_STATUS fonksiyonu üç farklı değer alabilir.

Değer	Açıklama
0	Bir önceki FETCH komutu başarıyla gerçekleştirildi.
-1	Bir önceki FETCH komutunda hata ile karşılaşıldı.
2	Son kayıta ulaşıldı. (end of resultset)

Tablo 3.1: @@FETCH_STATUS fonksiyonu değer tablosu

@@ROWCOUNT fonksiyonu, bir önceki FETCH komutu başarıyla gerçekleştirildikten sonra sonuç setinde toplam kaç kaydın bulunduğunu tutan fonksiyondur. FETCH komutu hiç kullanılmamışsa, imlecin işaretlediği sonuç setinde toplam kaç kaydın yer aldığını gösterir.

Örnek

Kitaplar tablosunun istenilen içeriğini yazdıracak bir imleç tanımlayıp kullanacağınızı varsayınız.

Table - dbo.Kitaplar					
	Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_sayfa	Kitap_Fiyat
▶	123	VB.NET	Ali CAN	550	45
	128	C#.Net	Veli BAL	700	55
	150	Access XP	Osman GÜR	400	25
	200	VB 6.0 Uygulamaları	Sena KOÇ	330	12
	250	C++ Uygulamaları	Yeliz AKCA	270	12
	251	Delphi 7.0	Defne ULUKAN	330	15
*	NULL	NULL	NULL	NULL	NULL

Resim 3.1: Kitaplar tablosu

- İlk adımda imleç tanımlanır.

```
DECLARE im_imlec CURSOR FOR
SELECT Kitap_Id, Kitap_Ad
FROM Kitaplar
```

Resim 3.2: İmlecin tanımlanması ve SELECT sorgusu

- İkinci adımda imleç açılır.

```
OPEN im_imlec
```

Resim 3.3: İmlecin açılması

- Üçüncü adımda imleç üzerinde dolaşmak için FETCH komutundan faydalanılır. Bu sorgu tarafından döndürülen iki satıra karşılık değişkenler tanımlanır. FETCH ile ilk satırın üstüne gelinir ve ilk kayıta yer alan iki değer yazdırılır.

```
DECLARE @KitapKod int, @Ad VARCHAR(50)
FETCH im_imlec INTO @KitapKod,@Ad
Print CAST(@KitapKod AS VARCHAR(3))
Print @Ad
```

Resim 3.4: FETCH komutunun kullanılması

- İmleç kapatılır.

```
CLOSE im_imlec
```

Resim 3.5: İmlecin kapatılması

- Gerekirse hafızadan silinir.

```
DEALLOCATE im_imlec
```

Resim 3.6: İmlecin hafızadan silinmesi

- Eğer, birden fazla kayıt yazdırılacaksa döngü kullanılmalıdır.

```
WHILE (@@FETCH_STATUS=0)
BEGIN
    Print CAST(@KitapKod AS VARCHAR(3))
    Print @Ad
    FETCH im_imlec INTO @KitapKod,@Ad
END
```

Resim 3.7: Birden fazla kayıt için döngü kullanılması

```
DECLARE im_imlec CURSOR FOR
SELECT Kitap_Id, Kitap_Ad
FROM Kitaplar

OPEN im_imlec

DECLARE @KitapKod int, @Ad VARCHAR(50)

FETCH im_imlec INTO @KitapKod,@Ad

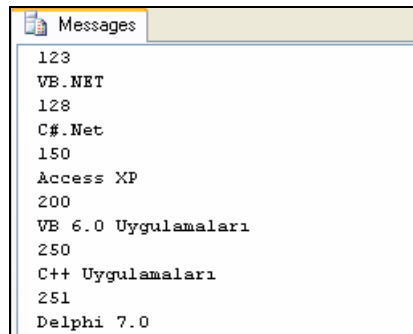
WHILE (@@FETCH_STATUS=0)
BEGIN
    Print CAST(@KitapKod AS VARCHAR(3))
    Print @Ad
    FETCH im_imlec INTO @KitapKod,@Ad
END

CLOSE im_imlec

DEALLOCATE im_imlec
```

Resim 3.8: İmlec oluşturma ve kapatma işleminin tamamı

CAST fonksiyonuyla int tipindeki @KitapKod değişkeni VARCHAR tipine dönüştürülmektedir. Sorguyu çalıştırdıktan sonra elde edilen sonuçları Messages penceresinden görebilirsiniz.



Resim 3.9: İmlecın sonucu

3.2.3. İmleç Seçeneklerini Ayarlamak

Bir imlecin tipi ve davranışı ayarlanabilir.

- CursorType: Forward-Only, static, keyset-driven, dynamic değerlerinden birini alabilir. Genel olarak istemci tipi imleçler, forward-only'dir. ODBC, OLE DB veya DB-Library gibi API'ler tarafından bu değer otomatik olarak ayarlanabilir. Kullanıcının istediği şekle de getirilebilir.
- CursorBehaviors: SCROLL ve INTENSIVE tiplerinden birini alabilir. ODBC ve DB-Library bu özellik için değer ataması yapmaz.

3.3. İmleçlerle Satırlar Arasında Dolaşmak

Bir imleç satırlar üzerinde dört temel işlevi yerine getirir: İlk kayıta, son kayıta, önceki veya sonraki kayıta gidebilir ve SCROLL CURSOR olarak adlandırılırlar.

İfade	İşlevi
FETCH FIRST	İlk satıra konumlanır.
FETCH LAST	Son satıra konumlanır.
FETCH NEXT	Bir sonraki satıra konumlanır.
FETCH PRIOR	Bir önceki satıra konumlanır.
FETCH RELATIVE n	Bulunulan satırdan n kayıt ileriye konumlanır.
FETCH ABSOLUTE	Baştan n. kayıta konumlanır.

Tablo 3.2: Hareket ifadeleri

Örnek

“Kitaplar” tablosu üzerinde imleci çeşitli şekillerde hareket ettireceğinizi varsayınız.

```
SELECT Kitap_Id, Kitap_Ad FROM Kitaplar
ORDER BY Kitap_Id

DECLARE Hareket SCROLL CURSOR FOR
SELECT Kitap_Id,Kitap_Ad FROM Kitaplar
ORDER BY Kitap_Id

OPEN Hareket

FETCH LAST FROM Hareket

FETCH PRIOR FROM Hareket

FETCH ABSOLUTE 2 FROM Hareket

FETCH RELATIVE 3 FROM Hareket

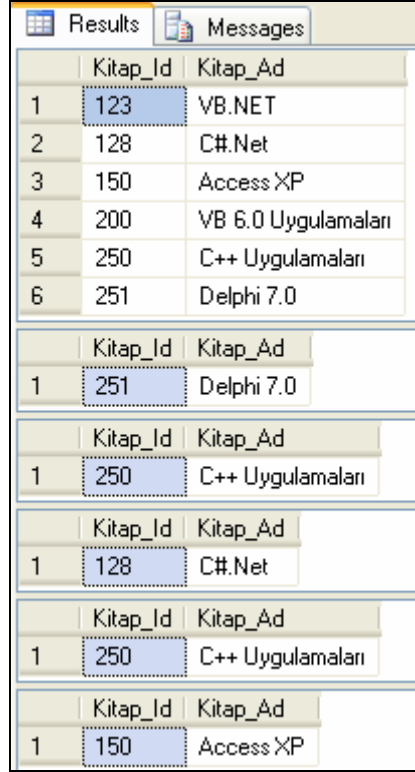
FETCH RELATIVE -2 FROM Hareket

CLOSE Hareket

DEALLOCATE Hareket
```

Resim 3.10: FETCH komutu ile satırlar arasında dolaşım örneği

Bu Query'i çalıştırdığımızda elde edilen sonuçları Results penceresinden görebilirsiniz.



	Kitap_Id	Kitap_Ad
1	123	VB.NET
2	128	C#.Net
3	150	Access XP
4	200	VB 6.0 Uygulamaları
5	250	C++ Uygulamaları
6	251	Delphi 7.0

Resim 3.11: Sonuç penceresi

3.4. İmleçle Kayıtları Düzenlemek

Bir imleçle, üzerinde bulunulan satırı silebilir veya düzenlemeler yapabilirsiniz. Bu işlemler aslında UPDATE ve DELETE komutlarıyla da yapılabilir ancak bir UPDATE veya DELETE ifadesiyle ifade edemeyeceğiniz bir işlem karşınıza gelirse bu tür bir yöntemi tercih edebilirsiniz.

UPDATE ve DELETE ifadelerinin WHERE kısmı için **WHERE CURRENT OF** ifadesini kullanabilirsiniz.

```
WHERE CURRENT OF @fiyatlar
```

Resim 3.12: WHERE CURRENT OF ifadesi

3.5. Ortak Zamanlı Çalışma

İmleçlerle bir veriye erişirken aynı anda başka başka kullanıcıların da bu veriye erişimlerini düzenlemek için kilitleme mekanizması devreye konulabilir.

Bir imlecin verilere ortak erişiminin düzenlenmesi için iki yöntemden biri kullanılabilir. Oturum seviyesinde kilitleme seçeneğini tayin ederek imleç veriye eriştiği anda diğer erişim isteklerinin hangi haklara sahip olacağı otomatik olarak tayin edilebilir. İmleci tanımlayan FROM cümlesinden sonra kilitleme seçeneği belirtilerek sadece imlecin işi bitinceye kadar olan süre zarfında, aynı kaynaklara erişmek isteyenler için bir kilitleme seçeneği belirtilebilir.

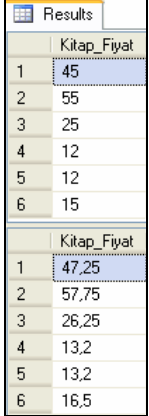
FROM'dan sonra Tablo 3.3'teki ifadelerden biri gelebilir.

İfade	İşlevi
READ_ONLY	Pozisyona bağlı güncellemelere izin verilmez ve imleç açıldığında herhangi bir kilit konmaz.
OPTIMISTIC WITH VALUES	Hiçbir satırı kilitlemez. Satır bazında değişiklikleri SQL Server takip eder.
OPTIMISTIC WITH ROW VERSIONING	Timestamp türü sütunu olan bir tablo üzerindeki imleçler için tanımlanabilir. Bu tür bir imleç, sadece timestamp sütun değerine bakarak kendisinden sonra değişip değişmediğini anlamasını sağladığı için server açısından daha az yük bindiricidir.
SCROLL LOCKS	SELECT cümlesinde verilen kilitleme ipucuna göre kilitleme yapar.

Tablo 3.3: Ortak zamanlılık seçenekleri

UYGULAMA FAALİYETİ

<p>➤ “Kitaplar” tablosunu kullanarak fiyatı 20 liranın altında olan kitaplara % 10, üstünde olan kitaplara % 5’lik bir fiyat artışı uygulayacak bir imleç programlayacağınızı varsayınız.</p>	<table border="1" data-bbox="679 478 1265 741"> <thead> <tr> <th colspan="6">Table - dbo.Kitaplar</th> </tr> <tr> <th></th> <th>Kitap_Id</th> <th>Kitap_Ad</th> <th>Kitap_Yazar</th> <th>Kitap_sayfa</th> <th>Kitap_Fiyat</th> </tr> </thead> <tbody> <tr> <td>▶</td> <td>123</td> <td>VB.NET</td> <td>Ali CAN</td> <td>550</td> <td>45</td> </tr> <tr> <td></td> <td>128</td> <td>C#.Net</td> <td>Veli BAL</td> <td>700</td> <td>55</td> </tr> <tr> <td></td> <td>150</td> <td>Access XP</td> <td>Osman GÜR</td> <td>400</td> <td>25</td> </tr> <tr> <td></td> <td>200</td> <td>VB 6.0 Uygulamaları</td> <td>Sena KOÇ</td> <td>330</td> <td>12</td> </tr> <tr> <td></td> <td>250</td> <td>C++ Uygulamaları</td> <td>Yeliz AKCA</td> <td>270</td> <td>12</td> </tr> <tr> <td></td> <td>251</td> <td>Delphi 7.0</td> <td>Defne ULUKAN</td> <td>330</td> <td>15</td> </tr> <tr> <td>*</td> <td>NULL</td> <td>NULL</td> <td>NULL</td> <td>NULL</td> <td>NULL</td> </tr> </tbody> </table> <p style="text-align: center;">Resim 3.13: Kitaplar tablosu</p>	Table - dbo.Kitaplar							Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_sayfa	Kitap_Fiyat	▶	123	VB.NET	Ali CAN	550	45		128	C#.Net	Veli BAL	700	55		150	Access XP	Osman GÜR	400	25		200	VB 6.0 Uygulamaları	Sena KOÇ	330	12		250	C++ Uygulamaları	Yeliz AKCA	270	12		251	Delphi 7.0	Defne ULUKAN	330	15	*	NULL	NULL	NULL	NULL	NULL
Table - dbo.Kitaplar																																																							
	Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_sayfa	Kitap_Fiyat																																																		
▶	123	VB.NET	Ali CAN	550	45																																																		
	128	C#.Net	Veli BAL	700	55																																																		
	150	Access XP	Osman GÜR	400	25																																																		
	200	VB 6.0 Uygulamaları	Sena KOÇ	330	12																																																		
	250	C++ Uygulamaları	Yeliz AKCA	270	12																																																		
	251	Delphi 7.0	Defne ULUKAN	330	15																																																		
*	NULL	NULL	NULL	NULL	NULL																																																		
<p>➤ “@fiyat” adında bir değişken tanımlayınız.</p>	<p>➤ DECLARE @fiyat int</p>																																																						
<p>➤ “@fiyatlar” adında bir imleç tanımlayarak “Kitaplar” tablosundaki “Kitap_Fiyat” alanını seçiniz.</p>	<p>➤ DECLARE @fiyatlar CURSOR SET @fiyatlar = CURSOR FOR SELECT Kitap_Fiyat FROM Kitaplar</p>																																																						
<p>➤ İmleci açınız.</p>	<p>➤ OPEN @fiyatlar</p>																																																						
<p>➤ İlk kayıta konumlanarak “Kitap_Fiyat” alanındaki değeri imleç yardımıyla @fiyat değişkenine aktarınız.</p>	<p>➤ FETCH NEXT FROM @fiyatlar INTO @fiyat</p>																																																						
<p>➤ Tablodaki asıl kitap fiyat verilerini görmek için bir SELECT sorgusu yazınız.</p>	<p>➤ SELECT Kitap_Fiyat FROM Kitaplar</p>																																																						
<p>➤ İmlecin tabloda son kayıta gelip gelmediğini kontrol edecek ve diğer kayıtlardaki verileri okuyacak bir döngü hazırlayınız.</p>	<p>➤ WHILE (@@FETCH_STATUS = 0) BEGIN FETCH NEXT FROM @fiyatlar INTO @fiyat END</p>																																																						
<p>➤ Kitap fiyatının 20 liradan az olması durumunda % 10’luk bir artışı ve imlecin bulunduğu kayıtlarla çalışmasını sağlayacak (WHERE CURRENT OF) kodu döngü içerisine yazınız.</p>	<p>➤ IF @fiyat < 20 UPDATE Kitaplar SET Kitap_Fiyat = (@fiyat + (@fiyat * .1)) WHERE CURRENT OF @fiyatlar</p>																																																						
<p>➤ Kitap fiyatı 20 liranın üzerinde ise % 5’lik artışı ve imlecin bulunduğu kayıtlarla çalışmasını sağlayacak kodu yazınız.</p>	<p>➤ ELSE UPDATE Kitaplar SET Kitap_Fiyat = (@fiyat + (@fiyat * .05)) WHERE CURRENT OF @fiyatlar</p>																																																						

➤ İmlecin çalışmasından sonra tabloda bulunan kitap fiyatlarının son hâlini görmek için yine bir SELECT seçme sorgusu yazınız.	➤ SELECT Kitap_Fiyat FROM Kitaplar
➤ İmleci kapatınız.	➤ CLOSE @fiyatlar
➤ İmleci hafızadan siliniz.	➤ DEALLOCATE @fiyatlar
➤ Yazdığınız imleci çalıştırınız.	➤ F5
➤ Sonuçları Results penceresinden görünüz.	

Resim 3.14: Sonuçların gösterimi

```

DECLARE @fiyat int
DECLARE @fiyatlar CURSOR
SET @fiyatlar = CURSOR FOR
    SELECT Kitap_Fiyat FROM Kitaplar
OPEN @fiyatlar
FETCH NEXT FROM @fiyatlar INTO @fiyat
SELECT Kitap_Fiyat FROM Kitaplar
WHILE ( @@FETCH_STATUS = 0)
BEGIN
    IF @fiyat < 20
        UPDATE Kitaplar SET Kitap_Fiyat = (@fiyat + (@fiyat * .1))
        WHERE CURRENT OF @fiyatlar
    ELSE
        UPDATE Kitaplar SET Kitap_Fiyat = (@fiyat + (@fiyat * .05))
        WHERE CURRENT OF @fiyatlar
        FETCH NEXT FROM @fiyatlar INTO @fiyat
END
SELECT Kitap_Fiyat FROM Kitaplar
CLOSE @fiyatlar
DEALLOCATE @fiyatlar

```

Resim 3.15: @fiyatlar imleci

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. İmleç, bir metin editörüyle aynı işlemi yapar. Yani, bulunduğu satırdaki verileri baz alır (D/Y).
2. T-SQL imleçler, istemci tarafından çalıştırılır (D/Y).
3. İmleç, komutu ile açılır, komutu ile kapatılır.
4. FETCH komutunun durumu hakkında bilgi veren komut'tür.
5. Aşağıdaki komutlardan hangisi imleci bellekten silmek için kullanılır?
A) DELETE
B) DEL
C) DROP
D) DEALLOCATE
6. Resultset'te toplam kaç kaydın bulunduğunu gösteren komut aşağıdakilerden hangisidir?
A) RESULT B) ROWCOUNT C) @@ROWCOUNT D) COUNT
7. CAST fonksiyonu bir tipteki değişkeni başka bir tipe çevirmek için kullanılır (D/Y).
8. İmleç ile önceki kayda gitmek için, son kayda gitmek için kullanılır.
9. İmlecin bulunduğu kayıt ile çalışmasını sağlamak için WHERE CURRENT OF ifadesi kullanılır (D/Y).
10. Pozisyona bağlı güncellemelere izin vermeyecek ve imleç açıldığında herhangi bir kilit koymayacak ifade'dir.

DEĞERLENDİRME

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

ÖĞRENME FAALİYETİ-4

AMAÇ

SQL Server’da tetikleyici oluşturmayı ve kullanmayı öğreneceksiniz.

ARAŞTIRMA

- Programlama dillerinde belirtilen şartı yerine getiren yapıları araştırınız.

4. TETİKLEYİCİ

Tetikleyici (trigger), SQL Server üzerinde herhangi bir işlem gerçekleştiğinde başka bir işlemin tetiklenmesi için kullanılır. Tetiklenecek işlem tamamen bizim belirlediğimiz çerçevede başka bir kayıt eklemek, rapor almak veya genelde kontrol mekanizmaları oluşturmak için kullanılabilir. Tetikleyici, bir çeşit özel Stored Procedure’dür. Bir kaydın eklenmesi (INSERT), silinmesi (DELETE) veya değiştirilmesi (UPDATE) durumlarında otomatik olarak devreye girecektir.

Tetikleyiciler, Query ekranında T-SQL kodu ile veya Management Studio ile oluşturulup yönetilebilirler. Tetikleyiciler, görünümler (View) ve tablolar üzerinde tanımlanabilir.

SQL Server 2005, DDL (Data Defination Language) tetikleyicileri adında yeni bir kavramı destekler. DDL tetikleyiciler, olay gerçekleşikten sonra değil de CREATE, ALTER ve DROP gibi DDL ifadeleri için tanımlanırlar.

4.1. Tetikleyici Ne Zaman Kullanılır?

Çeşitli amaçlara uygun olarak tetikleyiciler kullanılabilir.

Bu amaçlar;

- Değişiklikleri takip etmek,
- Birincil anahtar üretmek,
- Karmaşık iş kurallarını gerçekleştirmek,
- E-posta atmak gibi olayları otomatik olarak yapmak,
- Standart hata mesajlarının dışında bir hata mesajı elde etmek,
- Veritabanı erişimlerini takip edebilmek,
- Nesnede meydana gelebilecek değişiklikleri takip ve engellemektir.

4.2. Tetikleyiciyi Ateşleyen Olaylar

Bir işlemin gerçekleşmesini veya gerçekleşmeye başlamasını tetikleyici ateşleyebilir.

Tetikleyiciyi ateşleyen bir işlem ile karşılaşıldığında tetikleyici ile işlem bir blokta ele alınır. Bu blok, RAM'de bulunan geçici hafıza bloğudur. Tetikleyiciyi çağıran işlemi onaylamak anlamında hiçbir işlem yapmayabilir veya işlem başarılı olmadığında işlemi geçersiz kılabilir.

Tetikleyici çalıştığı zaman Inserted ve Deleted adı verilen sahte tabloları kullanır. Bu tablolar tetikleyicinin ateşlendiği tabloyla eşdeğer alanlara sahiptir. Bunlar da mantıksal olarak RAM'de bulunur. Asıl tabloya bir kayıt eklendiğinde ve tetikleyici ateşlendiğinde bu kayıt Inserted tablosuna da eklenir. Tablodan bir kayıt silindiğinde silinen kayıt Deleted sahte tablosuna da eklenir. Update işlemi ise önce silme (Delete) ve ardından bir kayıt ekleme (Insert) olarak ele alınır. Bir kayıt güncellendiğinde asıl kayıt Deleted sahte tablosuna, değişen kayıt da Inserted sahte tablosuna yazılır.

Inserted ve Deleted sahte tabloları RAM'deki hafıza bölgesinde elde edilir. Bundan dolayı, TRUNCATE TABLE gibi yazılmayan tablolara yansımaya değerlikler tetikleyici tarafından yakalanamaz.

Tetikleyiciler dışarıdan parametre almazlar. Ancak, sahte tablolar sayesinde son işlemten etkilenmekte olan kayıtlar tespit edilebilir.

4.3. Tetikleyici Türleri

SQL Server'da iki farklı tür tetikleyici vardır. Bunlar **After** ve **Instead Of** tetikleyicileridir.

4.3.1. After Tetikleyicileri

After tetikleyicileri, kendiyile ilişkili işlem gerçekleştikten hemen sonra ateşlenir. Veritabanındaki temel işlemler için (ekleme, silme ve güncelleme) **After** tetikleyicileri tanımlanabilir. Örneğin, yeni bir personel kaydı silindiğinde farklı tablolarda o personele ait diğer bilgilerin silinmesi **After Delete** tetikleyicisi ile sağlanabilir. Birden fazla tetikleyici, bir iş için tanımlanabildiği gibi, bir tetikleyici de birden fazla iş için tanımlanabilir. **After** tetikleyicileri, sadece tablolar için tanımlanabilir.

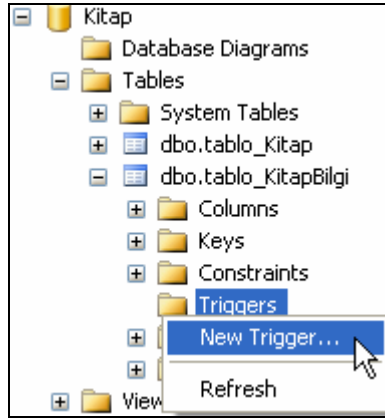
4.3.2. Instead Of Tetikleyicileri

Instead of tetikleyicileri, belirlenen işlem gerçekleşirken devreye girer ve kendi içinde tanımlanan komutları icra etmeye başlar. Yani, belirlenen işlemin yerine geçer. **Instead of** tetikleyicileri işlemlerin arasına girebildiğinden kontrol amaçlı kullanılabilirler. Örneğin, ekleme işlemi için tanımlanan bir tetikleyici, ekleme işleminden hemen önce araya girerek uygun koşulların sağlanıp sağlanmadığını kontrol edebilir. Tıpkı **After** tetikleyicileri gibi temel veritabanı işlemleri için **Instead Of** tetikleyicileri tanımlanabilir. Fakat, **After**

tetikleyicileri sadece tablolar için tanımlanabilirken **Instead Of** tetikleyicileri hem tablolar için hem de görünüm (views) için tanımlanabilirler.

4.4. Tetikleyici Oluşturmak

Tetikleyiciyi oluşturmak için CREATE TRIGGER ifadesi veya Managemet Studio'da tablo altında bulunan Triggers klasörü üzerinde sağ tıklayarak açılan menüden New Trigger komutu kullanılır.



Resim 4.1: New Trigger komutu

CREATE TRIGGER'i kullanabilmek için sysadmin, db_owner veya db_ddladmin rolüne sahip olmak gerekir.

Bir tetikleyici (trigger) başka bir tabloya erişecekse bu tablo için de tetikleyici oluşturan kullanıcının erişim izni veya güncelleme izni olması gerekir.

Yazılışı

```
CREATE TRIGGER tetikleyici_adi  
ON tablo_adi  
FOR veya AFTER veya INSTEAD OF (INSERT veya UPDATE veya DELETE)  
AS Sql ifadeleri
```

4.4.1. INSERT Tetikleyicisi

Bir tablo üzerinde yeni kayıtlar girildikten sonra devreye giren tetikleyicidir. Insert tetikleyici devreye girdikten sonra Inserted tablosunda yeni eklenen kayıtların bir kopyası tutulur. Inserted tablosu, asıl tablonun yapısal bir kopyası olup tetikleyici sonlanana kadar saklanır. Eğer, bir tabloya birden fazla kayıt girildiyse tetikleyici her kayıt için ayrı ayrı devreye girmez. Tek seferde işlemi gerçekleştirir. Çünkü SQL Server, satır bazında tetikleyici desteği sağlamamaktadır.

Örnek

“tablo_KitapBilgi” tablosunda kitapların basım yılını otomatik olarak yazacak bir tetikleyici yazacağınızı varsayınız.

Table - dbo.tablo_KitapBilgi				
	Kitap_Id	Kitap_Yayinevi	Kitap_ISBNNo	Kitap_BasimYili
	123	ABC Yayınevi	1234567890	2006
	128	DEF Yayıncılık	5678901234	2007
	150	XYZ Matbaası	3456789012	2007
	200	ABC Yayınevi	1234567123	2006
*	NULL	NULL	NULL	NULL

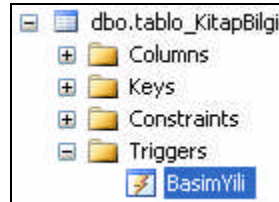
Resim 4.2: “tablo_KitapBilgi” tablosu

Bunun için, bir tetikleyici oluşturmanız gerekmektedir.

```
CREATE TRIGGER BasimYili
ON tablo_KitapBilgi
AFTER INSERT
AS
UPDATE tablo_KitapBilgi Set Kitap_BasimYili=2007
WHERE tablo_KitapBilgi.Kitap_Id=(SELECT Kitap_Id FROM INSERTED)
```

Resim 4.3: Tetikleyicinin oluşturulması

“tablo_KitapBilgi” tablosuna yeni bir kayıt girildiğinde ve kitap basım yılı boş geçildiğinde kitap basım yılı olarak 2007 değeri otomatik olarak yazılacaktır. Tetikleyiciyi yazdıktan sonra Execute ediniz. Artık tetikleyici, Tables klasörü altında oluşturulacak ve tetiklenmeye hazır hâle gelecektir.



Resim 4.4: Oluşturulan tetikleyicinin konumu

“tablo_KitapBilgi” tablosunu açarak yeni bir kayıt giriniz ve basım yılı alanına değeri girmeden Enter tuşuna basınız.

Table - dbo.tablo_KitapBilgi				
	Kitap_Id	Kitap_Yayinevi	Kitap_ISBNNo	Kitap_BasimYili
	123	ABC Yayinevi	1234567890	2006
	128	DEF Yayıncılık	5678901234	2007
	150	XYZ Matbaası	3456789012	2007
	200	ABC Yayinevi	1234567123	2006
▶ !	205	DEF Yayıncılık	9876543210	NULL
*	NULL	NULL	NULL	NULL

Resim 4.5: Yeni kaydı girilmesi


Basım yılı alanını boş geçip Enter tuşuna bastığımızda tetikleyici devreye girecektir. Eklenen kaydın son durumunu görmek için tabloyu Refresh etmeniz gerekir. Bunun için,  Execute SQL simgesini tıklayarak ya da kapatıp açarak tabloyu yenileyiniz. Basım yılı alanına otomatik olarak 2007 değerinin yazıldığını göreceksiniz.

Table - dbo.tablo_KitapBilgi				
	Kitap_Id	Kitap_Yayinevi	Kitap_ISBNNo	Kitap_BasimYili
▶	123	ABC Yayinevi	1234567890	2006
	128	DEF Yayıncılık	5678901234	2007
	150	XYZ Matbaası	3456789012	2007
	200	ABC Yayinevi	1234567123	2006
	205	DEF Yayıncılık	9876543210	2007
*	NULL	NULL	NULL	NULL

Resim 4.6: Tablonun yenilendikten sonraki görünümü

4.4.2. DELETE Tetikleyicisi

Tablodan bir kayıt silindiğinde otomatik olarak yapılması istenen işlemler için DELETE tetikleyicisi kullanılır. DELETE tetikleyicisi çalıştıktan sonra silinen kayıt Deleted sahte tablosuna kaydedilir. Deleted tablosunun Inserted tablosundan farkı, asıl tablodan silinen kayıt artık Deleted tablosunda yer almaktadır.

Örnek

“tablo_Kitap” tablosunda “Kitap_Id”si 150’den büyük olan kitapların silinmesini engelleyecek bir DELETE tetikleyicisi yazdığınızı varsayınız.

Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_Sayfa
123	Visual Basic.NET	Ali CAN	550
128	C#.Net	Veli BAL	700
150	Access XP	Osman GÜR	450
200	VB 6.0 Uygulam...	Sena PINAR	330
250	C++ Uygulamaları	Yasemin ZEKİ	270
251	Delphi 7.0	Yağmur AK	310
NULL	NULL	NULL	NULL

Resim 4.7: “tablo_Kitap” tablosu

“tablo_Kitap” tablosu üzerinde bir tetikleyici oluşturulmaktadır. Tablodaki toplam kayıt sayısının 0’dan büyük olması durumu göz önüne alınmaktadır. Bu halde, “Kitap_Id”si 150’ye eşit ve büyük id numaralarının silinmesi engellenecektir.

```

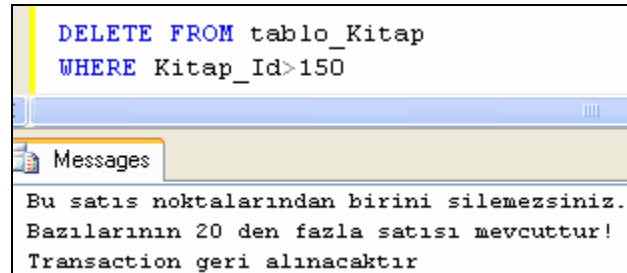
CREATE TRIGGER Sil
ON tablo_Kitap
FOR DELETE AS
IF (SELECT COUNT(*) FROM deleted
WHERE deleted.Kitap_Id >= 150) > 0
BEGIN
PRINT 'Bu kitap id lerinden birini silemezsiniz.'
PRINT 'Silme işlemi'
PRINT 'DELETE tetikleyici tarafından engellenmiştir.'
PRINT 'Transaction geri alınacaktır'
ROLLBACK TRANSACTION
END

```

Resim 4.7: DELETE tetikleyicisinin kullanımı

Tetikleyiciyi Execute ettiğinizde “tablo_Kitap” altında “Sil” adında bir tetikleyici oluşturulacaktır.

Yeni bir sorgu sayfası açıp tablodan kitap id’si 150’den büyük olanları silmek istediğinizde uyarı mesajı ekrana gelecektir.



Resim 4.8: Silme sorgusu ve sonuç ekranı

4.4.3. UPDATE Tetikleyicisi

Tablo üzerindeki kayıt ya da kayıtlarda güncelleme olduğunda devreye girecek olan tetikleyicidir. INSERT ve DELETE tetikleyicilerden biraz farklıdır. Farkı ise UPDATE tetikleyici devreye girdiğinde Inserted sahte tablosu asıl tablodaki kayıtlardan, düzenlenmiş kayıtların kopyasını, Deleted sahte tablosu ise kayıtların düzenleme işleminden önceki hâllerini tutar.

Örnek

“tablo_Kitap” tablosunda bulunan kitap fiyatlarından kaçının fiyatının değiştiğini bulacak bir UPDATE tetikleyici yazdığınızı varsayınız.

Bunun için “tablo_Kitap” tablosuna kitap fiyatlarının tutulacağı “Kitap_Fiyat” adında ve int tipinde bir sütun oluşturarak sütun içerisine kitap fiyat bilgilerini giriniz.

Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_Sayfa	Kitap_Fiyat
123	Visual Basic.NET	Ali CAN	550	45
128	C#.Net	Veli BAL	700	35
150	Access XP	Osman GÜR	450	25
200	VB 6.0 Uygulam...	Sena PINAR	330	12
250	C++ Uygulamaları	Yasemin ZEKİ	270	12
251	Delphi 7.0	Yağmur AK	310	20
NULL	NULL	NULL	NULL	NULL

Resim 4.9: Kitap fiyatlarının tabloya eklenmesi

Buna göre UPDATE tetikleyicisini Resim 4.10’a göre oluşturunuz.

```
CREATE TRIGGER indirim
ON tablo_Kitap
FOR UPDATE
AS
raiserror ('%d kayıt üzerinde değişiklik yapılmıştır',0,1,@@rowcount)
RETURN
```

Resim 4.10: UPDATE tetikleyicisi örneği

Bu tetikleyicide kullanılan RAISERROR fonksiyonu, SQL Server’da hata meydana geldiğinde hata mesajı vermek için kullanılır. Burada ise sadece mesaj verme işlemi için kullanılmıştır. @@rowcount ise SQL Serverde en son yapılan işlemde kaç kaydın etkilendiğini verir.

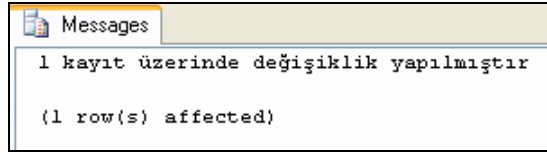
Bu tetikleyicinin otomatik olarak devreye girebilmesi için bu tetikleyiciyi ateşleyen bir SQL kodunun olması gerekir. Örnek olması açısından “tablo_Kitap” tablosunda “Kitap_Id”

numarası 250'den büyük kayıtların fiyatlarında 5 YTL'lik bir indirim yapılması istensin. Tetikleyiciyi ateşleyecek SQL kodlarını Resim 4.11'de gösterildiği gibi yazınız ve Execute ediniz.

```
UPDATE tablo_Kitap
SET Kitap_Fiyat=Kitap_Fiyat-5
WHERE Kitap_Id>250
```

Resim 4.11: Tetikleyiciyi ateşleyecek SQL kodları

Execute işleminden sonra Messages penceresinde sorgu sonucu kaç kaydın bu işlemten etkilendiği görüntülenecektir.



Resim 4.12: İşlemin sonucu

Tabloyu “Open Table” komutuyla açtığımızda etkilenen satırın son satır olduğunu göreceksiniz.

Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_Sayfa	Kitap_Fiyat
123	Visual Basic.NET	Ali CAN	550	45
128	C#.Net	Veli BAL	700	35
150	Access XP	Osman GÜR	450	25
200	VB 6.0 Uygulam...	Sena PINAR	330	12
250	C++ Uygulamaları	Yasemin ZEKİ	270	12
251	Delphi 7.0	Yağmur AK	310	15
NULL	NULL	NULL	NULL	NULL

Resim 4.13: Tablonun son hâli

4.4.4. INSTEAD OF Tetikleyicisi

INSTEAD OF tetikleyicisi, belirlenen işlem gerçekleşirken devreye girer ve kendi içinde tanımlanan komutları icra etmeye başlar. Yani, belirlenen işlemin yerine geçer.

Bu tetikleyiciden önceki tetikleyiciler, veriler uygun değilse ROLLBACK ile işlemleri geri alırlar. INSTEAD OF, işlem gerçekleşirken verilerin uygunluğunu denetleyecektir.

Örnek

- Uygulaması yapılacak olan bu tabloda, kayıtlı olan müşteri silinmeye çalışıldığı anda silme işleminin gerçekleştiği tarih ve saat otomatik olarak “SilmeGirisimi” adlı alana yazılacaktır.
- “Musteriler” adında bir tablo oluşturularak tabloda kullanılacak alanları T-SQL kodu ile tanımlayınız.

```
CREATE TABLE Musteriler
(
    Musteri_No int IDENTITY (1, 1) NOT NULL ,
    Ad_Soyad VARCHAR(30),
    SilmeGirisimi DATETIME
)
```

Resim 4.14: “Musteriler” tablosu

- EXECUTE ederek tablonun oluşumunu tamamlayınız. Oluşturulan bu tabloda müşteri numaraları 1’den başlayarak otomatik olarak artacaktır.
- Oluşturulan tabloya kayıt girişini de yine T-SQL kodları ile gerçekleştiriniz. Sorguyu çalıştırarak kayıtların tabloya eklenmesini sağlayınız.

```
INSERT INTO Musteriler (Ad_Soyad) VALUES ('Kemal ORTA')
INSERT INTO Musteriler (Ad_Soyad) VALUES ('Orkun TAN')
INSERT INTO Musteriler (Ad_Soyad) VALUES ('Naciye BÜYÜK')
INSERT INTO Musteriler (Ad_Soyad) VALUES ('Naci BÜYÜK')
INSERT INTO Musteriler (Ad_Soyad) VALUES ('Metin KÜÇÜK')
```

Resim 4.15: Musteriler tablosuna kayıt eklenmesi

- “Musteri_Silme” adında bir tetikleyici oluşturularak bu tetikleyicinin tablodan kayıt silinmesinin başlaması durumunda sistem tarih ve saatini “SilmeGirisimi” sütununa yazmasını ve sorguyu EXECUTE ederek tetikleyicinin oluşmasını sağlayınız.

```
CREATE TRIGGER Musteri_Silme
ON Musteriler
INSTEAD OF DELETE
AS
UPDATE Musteriler SET SilmeGirisimi = GETDATE()
WHERE Musteri_No IN (SELECT Musteri_No FROM DELETED)
```

Resim 4.16: INSTEAD OF tetikleyicisinin oluşturulması

- “Musteriler” tablosunda üçüncü ve beşinci kayıtları silecek yeni bir sorgu oluşturunuz ve çalıştırınız.

```

SELECT * FROM Musteriler
DELETE Musteriler WHERE Musteri_No=3

SELECT * FROM Musteriler
DELETE Musteriler WHERE Musteri_No=5

SELECT * FROM Musteriler

```

Resim 4.17: Kayıt silme sorgusu

- 3 ve 5 numaralı kayıt silinmek istendiğinde tetikleyici devreye girecek ve “SilmeGirisimi” sütununa sistem tarih ve saatini yazacaktır.
- Uygulamanın sonuç penceresine bakarak yapılan işlemleri görebilirsiniz.

Musteri_No	Ad_Soyad	SilmeGirisimi
1	Kemal ORTA	NULL
2	Orkun TAN	NULL
3	Naciye BÜYÜK	2007-05-27 21:12:59.780
4	Naci BÜYÜK	NULL
5	Metin KÜÇÜK	NULL

Musteri_No	Ad_Soyad	SilmeGirisimi
1	Kemal ORTA	NULL
2	Orkun TAN	NULL
3	Naciye BÜYÜK	2007-05-27 21:13:26.750
4	Naci BÜYÜK	NULL
5	Metin KÜÇÜK	NULL

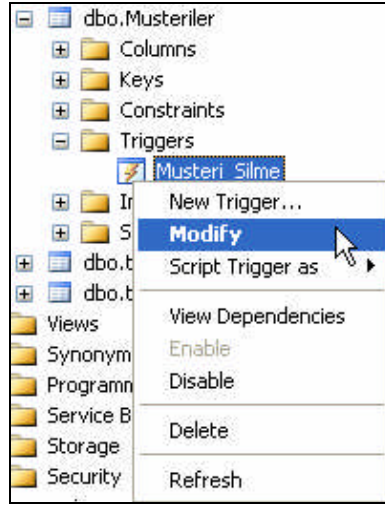
Musteri_No	Ad_Soyad	SilmeGirisimi
1	Kemal ORTA	NULL
2	Orkun TAN	NULL
3	Naciye BÜ...	2007-05-27...
4	Naci BÜYÜK	NULL
5	Metin KÜÇ...	2007-05-27...

Resim 4.18: Sorguyla ateşlenen tetikleyicinin ürettiği sonuç

4.5. Tetikleyici Yönetimi

4.5.1. Tetikleyiciyi Düzenlemek (ALTER)

CREATE ile oluşturulmuş bir tetikleyiciyi daha sonra üzerinde değişiklikler yapmak üzere tekrar açmak isteyebilirsiniz. Bunun için, tablonuzda oluşturulmuş bir tetikleyici üzerindeyken sağ tıklayarak açılan menüden Modify komutunu vermeniz gerekir.



Resim 4.19: Modify komutu

Modify komutunu verdiğinizde daha önceden oluşturduğunuz tetikleyici açılacaktır.

```
set ANSI_NULLS ON
set QUOTED_IDENTIFIER ON
go

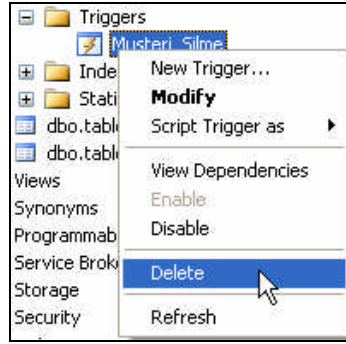
ALTER TRIGGER [Musteri_Silme]
ON [dbo].[Musteriler]
INSTEAD OF DELETE
AS
UPDATE Musteriler SET SilmeGirisimi = GETDATE()
WHERE Musteri_No IN (SELECT Musteri_No FROM DELETED)
```

Resim 4.20: Tetikleyicinin düzenleme için açılmış durumu

Burada dikkat edilmesi gereken nokta, tetikleyici oluşturulurken CREATE komutu ile oluşturulmuştu. Modify (düzenle) komutunu verdiğinizde ise tetikleyici, ALTER komutu ile açılacaktır. Daha önceden de bildiğiniz gibi ALTER, var olan tablo, tetikleyici gibi nesnelere üzerinde değişiklik yapmak için kullanılan komuttur. Daha önceden oluşturulmuş tetikleyiciyi düzenlerken yeni bir sorgu sayfası açıp Modify komutunu kullanmadan da ALTER komutu ile tetikleyiciyi yeniden yazarak düzenleyebilirsiniz. Ancak bu önerilmez.

4.5.2. Tetikleyiciyi Silmek (DROP)

Oluşturulan bir tetikleyiciyi DROP komutuyla ya da tetikleyici üzerinde sağ tıklayarak açılan menüden Delete komutunu vererek silebilirsiniz.



Resim 4.21: Tetikleyici silmek için Delete komutu

T-SQL kodu olarak,

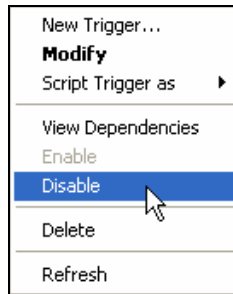
```
DROP TRIGGER Musteri_Silme
```

şeklinde yazabilirsiniz.

4.5.3. Tetikleyiciyi Aktif ya da Pasif Yapma

Oluşturulan bir tetikleyici, her zaman aktif durumdadır. Tetikleyiciyle ilgili bir durum oluştuğunda otomatik olarak devreye girer. İstenildiğinde oluşturulan bu tetikleyiciler pasif hâle getirilebilir.

Tetikleyici üzerinde sağ tıklayarak açılan menüde Disable komutunu verdiğinizde tetikleyici pasif hâle gelir.



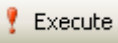
Resim 4.22: Disable komutu

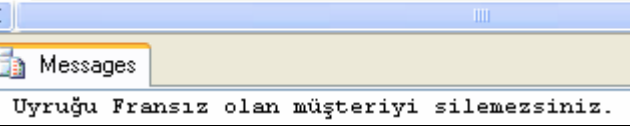
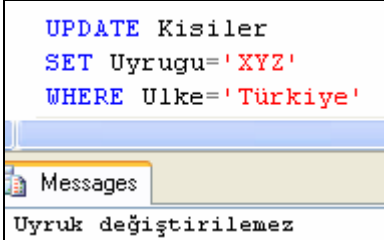
Tetikleyiciyi pasif hâle getirmenin bir diğer yolu da kod yazmaktır.

```
ALTER TABLE tablo_adi  
DISABLE TRIGGER Musteri_Silme
```

Tetikleyiciyi tekrar aktif hâle getirebilmek için menüden Enable komutunu veya ENABLE TRIGGER komutunu kullanmanız gerekir.

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler																																										
<ul style="list-style-type: none"> ➤ “Musteri_Bilgi” adında bir veritabanı oluşturunuz. 	<ul style="list-style-type: none"> ➤ New Database 																																										
<ul style="list-style-type: none"> ➤ Yeni bir tablo oluşturarak adını “Kisiler” olarak belirleyiniz. 	<ul style="list-style-type: none"> ➤ New Table 																																										
<ul style="list-style-type: none"> ➤ Tablo sütun adlarını “Ad_Soyad”, “Sehir”, “Ulke”, “Y_Dil”, ve “Uyrugu” olarak belirleyiniz. 	<ul style="list-style-type: none"> ➤ “Ad_Soyad” VARCHAR(30), “Sehir” VARCHAR(15), “Ulke” VARCHAR(15), “Y_Dil” VARCHAR(15), “Uyrugu” VARCHAR(10), “Onay” 																																										
<ul style="list-style-type: none"> ➤ Tabloya beş kayıt giriniz. 	<ul style="list-style-type: none"> ➤ Open Table komutu <table border="1"> <thead> <tr> <th colspan="6">Table - dbo.Kisiler</th> </tr> <tr> <th>Ad_Soyad</th> <th>Sehir</th> <th>Ulke</th> <th>Y_Dil</th> <th>Uyrugu</th> <th></th> </tr> </thead> <tbody> <tr> <td>Ahmet ERCAN</td> <td>Ankara</td> <td>Türkiye</td> <td>İngilizce</td> <td>Türk</td> <td></td> </tr> <tr> <td>Uğur BAŞÇI</td> <td>İstanbul</td> <td>Türkiye</td> <td>İngilizce</td> <td>Türk</td> <td></td> </tr> <tr> <td>John STROMUA</td> <td>Paris</td> <td>Fransa</td> <td>Fransızca</td> <td>Fransız</td> <td></td> </tr> <tr> <td>Ally KAMPBELL</td> <td>New York</td> <td>U.S.A.</td> <td>İngilizce</td> <td>Amerikan</td> <td></td> </tr> <tr> <td>Grater HANSEN</td> <td>Helsinki</td> <td>Almanya</td> <td>Almanca</td> <td>Alman</td> <td></td> </tr> </tbody> </table> <p>Resim 4.23: Kayıtların tabloya girilmesi</p>	Table - dbo.Kisiler						Ad_Soyad	Sehir	Ulke	Y_Dil	Uyrugu		Ahmet ERCAN	Ankara	Türkiye	İngilizce	Türk		Uğur BAŞÇI	İstanbul	Türkiye	İngilizce	Türk		John STROMUA	Paris	Fransa	Fransızca	Fransız		Ally KAMPBELL	New York	U.S.A.	İngilizce	Amerikan		Grater HANSEN	Helsinki	Almanya	Almanca	Alman	
Table - dbo.Kisiler																																											
Ad_Soyad	Sehir	Ulke	Y_Dil	Uyrugu																																							
Ahmet ERCAN	Ankara	Türkiye	İngilizce	Türk																																							
Uğur BAŞÇI	İstanbul	Türkiye	İngilizce	Türk																																							
John STROMUA	Paris	Fransa	Fransızca	Fransız																																							
Ally KAMPBELL	New York	U.S.A.	İngilizce	Amerikan																																							
Grater HANSEN	Helsinki	Almanya	Almanca	Alman																																							
<ul style="list-style-type: none"> ➤ “Uyrugu” sütununa hiçbir şey yazılmazsa kişinin uyrugunu “Türk.” olarak belirleyen bir tetikleyici yazınız. 	<pre>CREATE TRIGGER Musteri_Bilgi ON Kisiler AFTER INSERT AS UPDATE Kisiler Set Kisiler.Uyrugu='Türk' WHERE Kisiler.Ulke='Türkiye'</pre> <p>Resim 4.24: Tetikleyicinin oluşturulması</p>																																										
<ul style="list-style-type: none"> ➤ Tetikleyiciyi oluşturduktan sonra tabloya yeni bir kayıt girerek tabloyu EXCEUTE ediniz. 	<ul style="list-style-type: none"> ➤  																																										
<ul style="list-style-type: none"> ➤ Uyrugu “Fransız” olan müşteriye silmeyi engelleyecek tetikleyiciyi ve örnek sorgusunu yazınız. 	<pre>CREATE TRIGGER Musteri_Sil ON Kisiler AFTER DELETE AS IF NOT EXISTS (SELECT * FROM Kisiler K Join deleted D ON D.Uyrugu=K.Uyrugu WHERE K.Uyrugu ='Fransız') BEGIN RAISERROR('Uyrugu Fransız olan müşteriye silemezsiniz.',10,1) ROLLBACK END</pre> <p>Resim 4.25: DELETE tetikleyicisi</p>																																										

	<pre>DELETE FROM Kisiler WHERE Uyrugu='Fransız'</pre>  <p>Resim 4.26: Silme sorgusu</p>
<p>➤ Hiçbir uyruk değişimine izin vermeyen tetikleyiciyi ve örnek sorgusunu yazınız.</p>	<pre>CREATE TRIGGER Uyruk_Degismez ON Kisiler AFTER UPDATE AS IF UPDATE (Uyrugu) BEGIN IF EXISTS (SELECT * FROM Kisiler K JOIN Deleted D ON D.Ulke=K.Ulke WHERE D.Uyrugu<>K.Uyrugu) BEGIN RAISERROR('Uyruk değiştirilemez', 10, 1) ROLLBACK END END</pre> <p>Resim 4.27: UPDATE tetikleyicisi</p> <pre>UPDATE Kisiler SET Uyrugu='XYZ' WHERE Ulke='Türkiye'</pre>  <p>Resim 4.28: Güncelleme sorgusu</p>

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. Aşağıdaki durumlardan hangisinde tetikleyici otomatik olarak devreye girmez?
A) CREATE B) INSERT C) UPDATE D) DELETE
2. Tabloya bir kayıt eklendiğinde ve trigger tetiklendiğinde bu kayıt sahte tablosuna da eklenir.
3. SQL Server’da kaç tetikleyici türü vardır?
A) 1 B) 2 C) 3 D) 4
4. Bir tetikleyiciyi oluşturabilmek için aşağıdaki rollerden hangilerine sahip olmak gerekir?
A) public B) db_securityadmin C) db_accessadmin D) sysadmin
5. Tabloya birden fazla kayıt girildiyse tetikleyici her kayıt için ayrı ayrı devreye girer (D/Y).
6. Asıl tablodan silinen kayıt, Deleted tablosunda yer alır (D/Y).
7. UPDATE tetikleyicisi ateşlendiğinde, Inserted tablosunda yeni kayıtların kopyası, Deleted tablosunda eski kayıtların kopyası kalır (D/Y).
8. SQL Server’da en son yapılan işlemde kaç kaydın etkilendiğini verir.
9. After tetikleyicileri veriler uygun değilse ile işlemleri geri alır.
10. Tetikleyiciyi düzenlemek için, silmek için komutu kullanılır.

DEĞERLENDİRME

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

ÖĞRENME FAALİYETİ-5

AMAÇ

SQL Server’da görünüm oluşturmaya öğreneceksiniz.

ARAŞTIRMA

- Verilere erişmek için kullanılan en güvenli metotları genel olarak araştırınız.

5. GÖRÜNÜM

Görünüm (view), sorguları basitleştirmek, erişim izinlerini düzenlemek, farklı sunuculardaki benzer verileri karşılaştırmak ve bazen sorgu süresini kısaltmak için kullanılan, aslında var olmayan, SELECT ifadesi ile tanımlanmış sanal tablolardır.

Tabloları, olduğundan farklı göstermek için filtreler kullanılır. Bu gibi işlemler için view kullanılır. View’ler kaydedilmiş sorgulardan oluşur.

Bir view’in verileri aldığı asıl tabloya temel tablo denir. View, temel tabloda bulunan sütun ya da sütunları veya satırları kapsayabileceği gibi birden fazla temel tablo üzerinde tanımlanmış olabilir.

5.1. Hangi Amaçlar İçin Kullanılır?

View’ler;

- Kullanıcıların, önemli bir tabloda istenilen sütun veya satırları görmek,
- Kullanıcıların, uygun birim dönüşümleri yapılmış değerlerini (mil, km gibi) görmek,
- Tablolarda bulunan verileri başka bir tablo formatında sunmak,
- Karmaşık sorguları basitleştirmek amacıyla kullanılır.

5.2. View Oluşturmak

Tablolarla aynı özelliklere sahiptir. En fazla 1024 sütun oluşturulabilir. View, CREATE VIEW ifadesi kullanılarak oluşturulur.

Genel Kullanımı

```
CREATE VIEW view_adi  
AS  
SELECT sütun_adları  
FROM temel_tablo
```

View, iki türlü oluşturulabilir:

- T-SQL kod yazarak
- Object Explorer’da aktif veritabanı görünümünden New View komutunu vererek

Örnek

Personel veritabanındaki “Person_Bilgi” tablosunda, cinsiyeti “Kadın” olan personeli ad, soyad ve bölümüyle gösterecek bir view hazırlayacağınızı varsayınız.

Table - dbo.Person_Bilgi								
	Sicil_No	Ad	Soyad	Cinsiyet	Bolum	Unvan	Meslek	Brut_Ucret
▶	1234	Semiha	İPEK	Kadın	Yayın Kurulu	Yayın Kur.Üyesi	Eğitimci	1250
	1235	Nazlı	DENİZ	Kadın	Yayın Kurulu	Yayın Kur.Üyesi	Ekonomist	1250
	1236	Tamer	DEMİRAY	Erkek	Yayın Kurulu	Yayın Kur.Üyesi	Yazar	1250
	1237	Mehmet	AYDIN	Erkek	Üretim	Yönetici	Makine Mühendisi	1500
	1238	Nevzat	GÜVEN	Erkek	Üretim/Satınalma	Şef	İşletmeci	1000
	1239	Hüsamettin	ÇELİK	Erkek	Üretim/Satınalma	İşçi	İşletmeci	1000
	1240	Hülya	BİLEK	Kadın	Üretim/Dizgi	İşçi	Dizgi Operatörü	1000
	1241	Ahmet	AKKAYA	Erkek	Üretim/Dizgi	İşçi	Tashihiçi	900
	1242	Hüseyin	ATAKENT	Erkek	Üretim/Basım	İşçi	Grafiker	900
	1243	Hasan	KUŞ	Erkek	Üretim/Basım	İşçi	Grafiker	900
	1244	Toprak	TÜRKER	Erkek	Pazarlama	Uzman	Pazarlamacı	1000
	NULL	NULL	NULL	NULL	Reklam	Uzman	Reklamcı	1000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Resim 5.1: “Person_Bilgi” tablosu

- Yeni bir Query açarak, view’i oluşturmak için Resim 5.2’deki kodları yazınız.

```
CREATE VIEW V_Cinsiyet  
AS  
SELECT Ad, Soyad, Bolum  
FROM Person_Bilgi  
WHERE Cinsiyet='Kadın'
```

Resim 5.2: View’in oluşturulması

- CREATE View ile görünüme bir isim verilir.

- SELECT ile tablodan kullanılacak alanlar belirlenir.
- WHERE ile istenirse bir şart konulabilir.
- Yeni bir Query daha açarak oluşturulan view'in çalıştırılması gerekir.

```
SELECT * FROM V_Cinsiyet
```

Resim 5.3: Oluşturulan view'in çalıştırılması

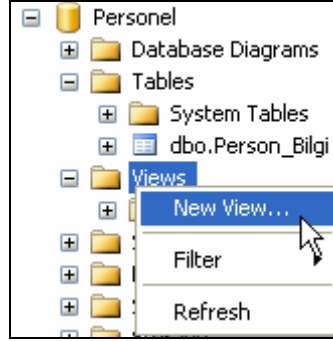
- "V_Cinsiyet" view'ini çalıştırdığımızda verilen şarta göre sonuçları Results penceresinde görebilirsiniz.

	Ad	Soyad	Bolum
1	Semiha	İPEK	Yayın Kurulu
2	Nazlı	DENİZ	Yayın Kurulu
3	Hülya	BİLEK	Üretim/Dizgi

Resim 5.4: View işleminin sonucu

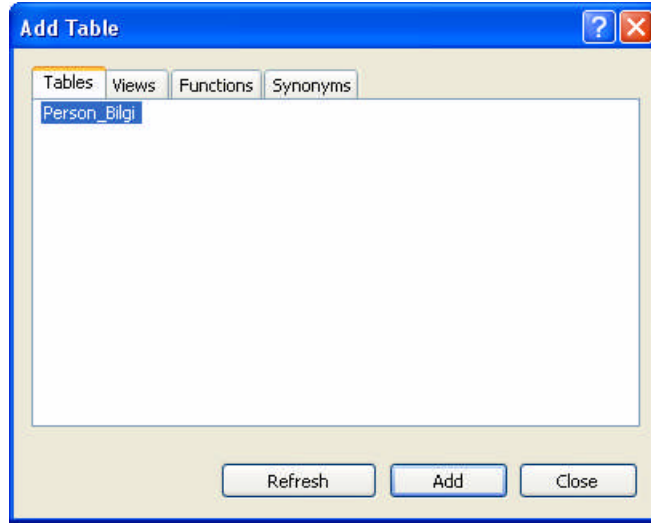
Aynı işlemi Management Studio'yu kullanarak da yapabilirsiniz.

- Veritabanı içindeki Views klasörü üzerinde fareyle sağ tuşu tıklayınız.



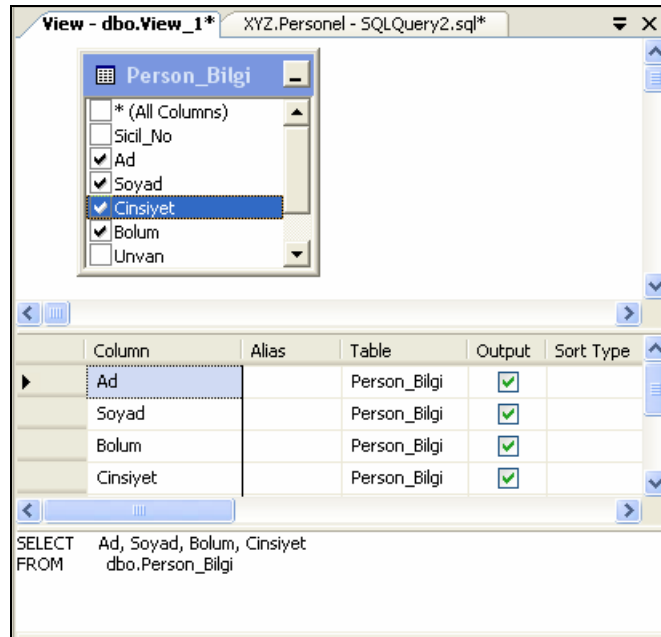
Resim 5.5: "New View" komutu

- Açılan "Add Table" penceresinde kullanacağımız "Person_Bilgi" tablosunu seçerek Add komut düğmesini tıklayınız. Close düğmesiyle de pencereyi kapatınız.



Resim 5.6: Add Table iletişim penceresi

- Eklenen tablo içerisinde kullanılacak alanları işaretleyiniz.



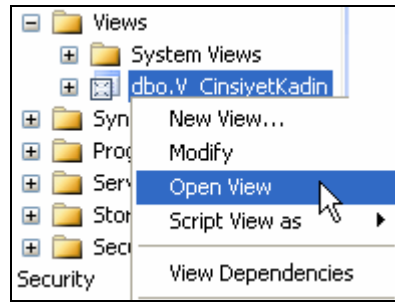
Resim 5.7: Alanların seçimi

- Alanları seçtiğinizde SELECT ifadesi otomatik olarak yazılacaktır.
- Cinsiyeti kadın olan personel view ile gösterileceğinden “Cinsiyet” alanında bir filtreleme yapılması gerekmektedir.



Resim 5.8: “Cinsiyet” alanı için filtrelemenin yapılması

- Oluşturulan view’i vereceğiniz bir isimle (V_CinsiyetKadin) kaydediniz.
- Son olarak view’i çalıştırmak kaldı. “Open View” komutuyla oluşturulan view’i çalıştırınız.



Resim 5.9: Oluşturulan view’in çalıştırılması

- View çalıştıktan sonra verilen ölçütlere göre sonucu görebilirsiniz.

View - dbo.V_CinsiyetKadin				
	Ad	Soyad	Bolum	Cinsiyet
▶	Semiha	İPEK	Yayın Kurulu	Kadın
	Nazlı	DENİZ	Yayın Kurulu	Kadın
	Hülya	BİLEK	Üretim/Dizgi	Kadın
*	NULL	NULL	NULL	NULL

Resim 5.10: View işleminin sonucu

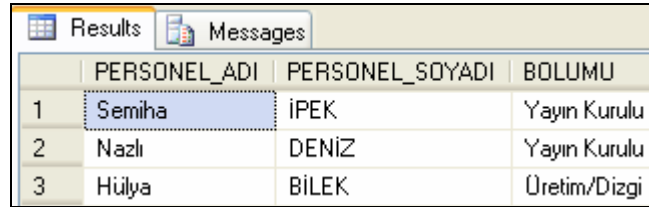
View’lerde sütun adları belirtilmediği sürece temel tabloda bulunan sütun adlarıyla aynı olur. View’le sütun adlarını ihtiyaca göre de değiştirebilirsiniz. View’de sütunlar için veri tipi belirtilmez çünkü view, temel tablodan baz alındığı için sütun veri tipleri temel tablodaki veri tipleriyle aynı olur.

Aynı örneği sütun adlarını değiştirerek yazarsak view, Resim 5.11'deki gibi olur.

```
CREATE VIEW V_Cinsiyet
(PERSONEL_ADI, PERSONEL_SOYADI, BOLUMU)
AS
SELECT Ad, Soyad, Bolun
FROM Person_Bilgi
WHERE Cinsiyet='Kadın'
```

Resim 5.11: View'le sütun adlarının belirlenmesi

Resim 5.3'teki sorguyu çalıştırdığımızda view'in sonucu Resim 5.12'deki gibi olacaktır.



	PERSONEL_ADI	PERSONEL_SOYADI	BOLUMU
1	Semiha	IPEK	Yayın Kurulu
2	Nazlı	DENİZ	Yayın Kurulu
3	Hülya	BİLEK	Üretim/Dizgi

Resim 5.12: Sorgu sonucu

5.3. Kısıtlamalar ve İzinler

View'ler geçici tabloları temel tablo olarak kullanamaz. Bunun dışında view'de kullanılan SELECT ifadesi ORDER BY, COMPUTE veya COMPUTE BY yan cümleciklerini alamaz.

View'in tanımlandığı veritabanına erişim izni olmayan kullanıcılar view'e erişemez. Ancak, veritabanında izni olan kullanıcılar miras yöntemi sayesinde view'e erişebilir. Dolayısıyla, view'in bulunduğu veritabanına erişmek demek, view'e erişmek demektir. Bu durumda view'i oluşturduktan sonra bir de kullanıcılar için erişim izni tanımlamak gerekir.

Kullanıcılara izin verilirken temel tabloya asla erişim izni vermemek ve sadece view'e erişim izni tanımak gerekir.

Bir view'in hangi tablolara bağlı olduğunu ve bu tabloların sahiplerinin kim olduğunu sp_depends sistem stored procedure'ü ile görmek mümkündür.

```
sp_depends V_Cinsiyet
```

Resim 5.13: "sp_depends" ile sahiplik sorgulama

Sonuç olarak sahipler Results penceresinde gösterilir.

	name	type	updated	selected	column
1	dbo.Person_Bilgi	user table	no	yes	Ad
2	dbo.Person_Bilgi	user table	no	yes	Soyad
3	dbo.Person_Bilgi	user table	no	yes	Cinsiyet
4	dbo.Person_Bilgi	user table	no	yes	Bolum

Resim 5.14: Sahiplik sonuçları

Genellikle SQL’de, nesnelerin sahibi dbo’dur. Bu nedenle, view oluşturulurken de sahibini dbo olarak belirlemek sonradan oluşacak karmaşıklıkların önüne geçecektir.

5.4. Birden Fazla Tablo Kullanımı

Bir join ifadesiyle birden fazla tablonun verilerinin görüntülenmesi sağlanabilir.

Genel olarak bu işlem için şu yapı kullanılır:

```
CREATE VIEW view_adi
AS
SELECT sütun_adları
FROM tablo_1 JOIN tablo_2
ON birleştirme_şartı
```

Örnek

“FİRMA” veritabanında, her bir personelin kaç yakınının olduğuyla ilgili bir view oluşturacağımızı varsayınız. Personel bilgileri “PERSONEL” tablosunda, personel yakınlarının bilgileri “PERSONEL_YAKIN” tablosunda tutulmaktadır.

```
CREATE VIEW V_PERSONELYAKIN
AS
(
SELECT PERSONEL.PERSONEL_ID, PERSONEL.AD, PERSONEL.SOYAD,
PERSONELYAKIN.YAKIN_ID, PERSONELYAKIN.YAKIN_AD, PERSONELYAKIN.YAKIN_SOYAD
FROM PERSONEL PERSONEL LEFT OUTER JOIN PERSONELYAKIN PERSONELYAKIN
ON PERSONEL.PERSONEL_ID=PERSONELYAKIN.PERSONEL_ID
)
```

Resim 5.15: Birden fazla tablo ile view kullanımı örneği

View oluşturulduktan sonra SELECT ifadesiyle sorguyu yazıp çalıştırdığınızda sonucu Results penceresinde görebilirsiniz.

	PERSONEL_ID	AD	SOYAD	YAKIN_ID	YAKIN_AD	YAKIN_SOYAD
1	11	Hasan	MERT	1	Havva	MERT
2	11	Hasan	MERT	2	Ayşe	MERT
3	11	Hasan	MERT	3	Ali	MERT
4	12	Mehmet	KUL	1	Merve	KUL
5	12	Mehmet	KUL	2	Rıza	KUL
6	13	Duran	AKIN	1	Hediye	AKIN
7	13	Duran	AKIN	2	Can	AKIN
8	13	Duran	AKIN	3	Canan	AKIN
9	13	Duran	AKIN	4	Cansu	AKIN
10	14	Çetin	ARSLAN	1	Zekiye	ARSLAN
11	14	Çetin	ARSLAN	2	Zeki	ARSLAN

Resim 5.16: Birden fazla tablo ile view'in sonucu

5.5. View'de Değişiklik Yapmak veya View'i Silmek

5.5.1. Değişiklik Yapmak

View üzerinde değişiklik yapmak için ALTER deyimi kullanılır.

```
ALTER VIEW view_adi
WITH seçenekler
AS
SELECT ifadesi
```

View üzerinde değişiklik yapabilmenin diğer bir yolu da Object Explorer penceresinde veritabanı altında oluşturduğunuz view üzerinde fareyle sağ tıklayarak açılan menüden Modify komutunu vermektir.

5.5.2. Silmek

View'leri silmek için DROP deyimi kullanılır.

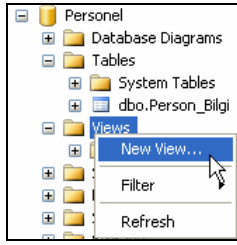
```
DROP view_adi
```

Diğer bir yöntem ise, Object Explorer penceresinde veritabanı altında oluşturduğunuz view üzerinde fareyle sağ tıklayarak açılan menüden Delete komutunu vermektir.

UYGULAMA FAALİYETİ

Table - dbo.Person_Bilgi								
	Sicil_No	Ad	Soyad	Cinsiyet	Bolum	Unvan	Meslek	Brut_Ucret
▶	1234	Semiha	İPEK	Kadın	Yayın Kurulu	Yayın Kur.Üyesi	Eğitimci	1250
	1235	Nazlı	DENİZ	Kadın	Yayın Kurulu	Yayın Kur.Üyesi	Ekonomist	1250
	1236	Tamer	DEMİRAY	Erkek	Yayın Kurulu	Yayın Kur.Üyesi	Yazar	1250
	1237	Mehmet	AYDIN	Erkek	Üretim	Yönetici	Makine Mühendisi	1500
	1238	Nevzat	GÜVEN	Erkek	Üretim/Satınalma	Şef	İşletmeci	1000
	1239	Hüsamettin	ÇELİK	Erkek	Üretim/Satınalma	İşçi	İşletmeci	1000
	1240	Hülya	BİLEK	Kadın	Üretim/Dizgi	İşçi	Dizgi Operatörü	1000
	1241	Ahmet	AKKAYA	Erkek	Üretim/Dizgi	İşçi	Tashihiçi	900
	1242	Hüseyin	ATAKENT	Erkek	Üretim/Basım	İşçi	Grafiker	900
	1243	Hasan	KUŞ	Erkek	Üretim/Basım	İşçi	Grafiker	900
	1244	Toprak	TÜRKER	Erkek	Pazarlama	Uzman	Pazarlamacı	1000
	NULL	NULL	NULL	NULL	Reklam	Uzman	Reklamcı	1000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Resim 5.17: “Person_Bilgi” tablosu

İşlem Basamakları	Öneriler
➤ Yeni bir query oluşturunuz.	➤ New Query
➤ Ünvanlara göre hazırlanacak bir view oluşturunuz.	➤ CREATE VIEW V_Unvan
➤ “Person_Bilgi” tablosundan ilgili alanları SELECT ile seçiniz.	➤ SELECT Sicil_No,Ad,Soyad FROM Person_Bilgi
➤ “Person_Bilgi” tablosunda ünvanı “İşçi” olan personel için bir filtreleme uygulayınız.	➤ WHERE Unvan='İşçi'
➤ Oluşturduğunuz view’i kaydediniz ve çalıştırınız.	➤ Save – Execute
➤ Yeni bir query daha açarak oluşturduğunuz view’i çalıştıracak SELECT’i yazınız.	➤ SELECT * FROM V_Unvan
➤ Management Studio ile kullandığınız veritabanında yeni bir view oluşturunuz.	

Resim 5.18: New View komutu

<p>➤ Add Table penceresinden kullanacağınız tabloyu seçiniz.</p>	<p>➤ Resim 5.6</p>
<p>➤ Kullanılacak alanları belirleyiniz.</p>	<div data-bbox="857 406 1069 644" data-label="Image"> </div> <p style="text-align: center;">Resim 5.19: Alanların seçimi</p>
<p>➤ Brüt ücreti 900 YTL'den büyük olan personel için filtreleme yapınız.</p>	<div data-bbox="792 764 1136 1006" data-label="Image"> </div> <p style="text-align: center;">Resim 5.20: "Brut_Ucret" alanı için filtreleme</p>
<p>➤ View'i kaydediniz.</p>	<div data-bbox="736 1131 1193 1317" data-label="Image"> </div> <p style="text-align: center;">Resim 5.21: View'in kaydedilmesi</p>
<p>➤ View'i açınız.</p>	<div data-bbox="836 1442 1092 1659" data-label="Image"> </div> <p style="text-align: center;">Resim 5.22: Open View komutu</p>

➤ Sonucu gözlemleyiniz.

View - dbo.V_BrutUcret				
	Sicil_No	Ad	Soyad	Brut_Ucret
▶	1234	Semiha	İPEK	1250
	1235	Nazlı	DENİZ	1250
	1236	Tamer	DEMİRAY	1250
	1237	Mehmet	AYDIN	1500
	1238	Nevzat	GÜVEN	1000
	1239	Hüsamettin	ÇELİK	1000
	1240	Hülya	BİLEK	1000
	1244	Toprak	TÜRKER	1000
	NULL	NULL	NULL	1000

Resim 5.23: View sonucu

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. Görünüm (view), erişim izinlerini düzenler ve sorguları basitleştirir (D/Y).
2. Bir view'in verileri aldığı asıl tabloya temel tablo denir (D/Y).
3. View ile en fazla adet sütun oluşturulabilir.
4. Bir view oluşturmak için kullanılan ifade'dir.
5. View'le bir filtreleme uygulamak için, istenen şart, ifadesinden sonra yazılır.
6. Management Studio'da bir view için kullanılacak alanlar seçildiğinde T-SQL kodları otomatik olarak yazılır (D/Y).
7. Aşağıdakilerden hangisi view'deki sütunlarla ilgili bir özellik değildir?
A) Sütun adları, temel tablodaki sütun adlarıyla aynıdır.
B) Sütun adları istenirse değiştirilebilir.
C) Sütun adları değiştirilemez.
D) Sütun veri tipleri için temel tablo baz alınır.
8. Aşağıdakilerden hangisi view oluştururken kullanılan SELECT ifadesiyle beraber kullanılabilir?
A) ORDER BY B) GROUP BY C) COMPUTE D) COMPUTE BY
9. View'in sahiplik sorgusu ile yapılabilir.
10. View'de değişiklik yapmak için deyimi, view'i silmek için deyimi kullanılır.

DEĞERLENDİRME

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

ÖĞRENME FAALİYETİ-6

AMAÇ

SQL Server’da indeks oluşturmaya öğreneceksiniz.

ARAŞTIRMA

- Programlama dillerinde indeksin kullanımını ve önemini genel hatlarıyla araştırınız.

6. İNDEKS

İndeks, tablolardan veri çekmek için gerekli sorgular çalıştırılırken gereken süreyi azaltmak amacıyla kullanılır. Kayıt sayısının milyonlarla ifade edildiği durumlarda çok önemli bir yere sahiptir.

İndeksi bir örnekle anlatmak gerekirse bir ilaç deposunu hayalinizde canlandırınız. İlaçların raflardaki dizilişlerini ele alınız. Bir ilaç aradığınızda, ilaçlar bir kurala göre dizilmemişse her bir ilacı tek tek kontrol etmeniz gerekir. İlaçları raflara alfabetik olarak dizerseniz, her bir ilacı tek tek gözden geçirmek zorunda kalmazsınız. Aynı şekilde, ilaç numaralarına göre sıralanmış bir liste olursa bu ölçütlere göre de aradığınız ilacı kolayca bulabilirsiniz.

6.1. İlişkisel Veritabanında İndeksler

Veritabanlarında indeks oluşturarak verileri, veritabanında kayıtlı oldukları sıradan değil de başka bir sırada gösterebiliriz.

Temelde indekslerin ilişkisel veritabanında şu üç işlevi vardır:

- Tekil indeksler, veri ilişkilerini ve veri bütünlüğünü sağlayan birincil anahtar alanlar oluşturmada kullanılır.
- İndeks olan alanın değerine göre bir kaydın kayıtlar arasındaki sırasını gösterir.
- Sorguların neticelenme sürelerini kısaltır.

6.2. Küme ve Küme Olmayan İndeksler

Küme indekslere clustered index, küme olmayan indekslere de nonclustered index de denir.

Clustered indekste tabloda yer alan kayıtlar, fiziksel olarak indeks tanımlı sütuna göre dizilirler. Bir clustered indeks tarandığında varılan son nokta verinin kendisidir. Bu yüzden, clustered indeksle yapılan aramalar çok hızlı sonuç verirler. Bir tabloda en fazla bir tane clustered indeks tanımlanabilir. Clustered indeks, özellikle aralık sorguları için yüksek performans sağlar.

SQL Server'da tanımlanan bir clustered indeks sütununa karşılık, kaydedilen değerlerin mutlaka daha önce kaydedilmemiş bir değer olması, yani tekil olması gerekir. Bu durum, bir tablonun Primary Key Constraint veya Unique Key Constraint tanımlı sütunu üstünde tanımlı indeksler için sorun çıkarmaz.

SQL Server, verilere erişmek için bir clustered indeksi iki şekilde kullanabilir:

- Clustered indeks araması (seek): Sorgu sonucunda dönecek kayıtlar üstünde bir kısıtlama varsa (WHERE cümlecisi gibi), clustered indeks araması kullanılır.
- Clustered indeks taraması (scan): Sorgu sonucunda dönecek kayıtlar üstünde bir kısıtlama yoksa clustered indeks taraması kullanılır.

Tablo üstünde clustered indeksin diğer indekslerden önce oluşturulmasında fayda vardır. Çünkü diğer indekslerin tamamında bu indeksin oluşturulduğu tekil değer kullanılır.

Nonclustered indeks, ya yığın (heap) üzerinden veya clustered indeksin üzerinden verilere erişebilir. Tek başına, doğrudan verilere erişemez.

Nonclustered indeksler veri erişiminde hızı artırır ancak clustered indeks araması veya clustered indeks taraması ile birlikte kullanılabilir olduğundan performansları clustered indekslerden daha düşüktür.

6.3. İndeks Oluşturmak

İndeks oluşturmak için kullanılacak en basit ifade şu şekildedir:

```
CREATE indeks_tipi INDEX indeks_adi  
ON tablo_adi(sütun_adi)
```

Genel kullanımda;

- İndeks_tipi: Unique clustered veya sadece clustered, nonclustered şeklinde indeksin tipini belirtir. Tip belirtilmediğinde nonclustered'dir.
- İndeks_adi: İndekse verilen isim.

- Tablo_adı: İndeksin tanımlandığı tablo ya da view adı.
- Sütun_adı: Tablo ya da view'de indekslenmesi istenen sütun veya sütunların ismi.

Örnek

“Person_Bilgi” tablosunda “Sicil_No” sütunu üstünde Clustered Index tanımlansın.

```
CREATE CLUSTERED INDEX Ind_Sicil  
ON Person_Bilgi (Sicil_No)
```

SQL Server’da sıralı indeks oluşturabilir. Sıralı indeks için ASC veya DESC deyimleri sütun adlarıyla beraber kullanılır. Varsayılan sıralama değeri ASC’dir.

Örnek

```
CREATE INDEX Ind_Brut_Ucret  
ON Person_Bilgi (Brut_Ucret DESC)
```

6.3.1. Unique İndeks

Verilerin tekrarlanmaması için kullanılır. Verilerin tekrarlanmaması için UNIQUE deyimini kullanmak yeterlidir. Veri alma ve tekrarı önlemede aynı indeks kullanılabilir.

UNIQUE indeks, hem clustered hem de nonclustered tipe olabilir. Kısıtlayıcı (Primary Key Constraint veya Unique Constraint) tanımı yapıldığında Unique indeks otomatik olarak oluşur. Bu oluşum yapılırken indeksin clustered olup olmayacağı belirtilmelidir. Kullanıcı bir seçenek belirtmezse Primary Key Constraint SQL Server’ca Unique indeks olarak Unique Constraint’de nonclustered unique indeks olarak belirlenir.

Örnek

```
ALTER TABLE Person_Bilgi  
ADD CONSTRAINT PK_Sicil PRIMARY KEY(Sicil_No)  
CLUSTERED
```

6.3.2. Karma (Composite) veya Birleşik İndeks

Birbiriyle ilişkili tablolar arasında bir tablonun iki sütunu birincil anahtar olarak belirlenebilir. Bu duruma karmaşık veya birleşik indeks denir.

Örnek

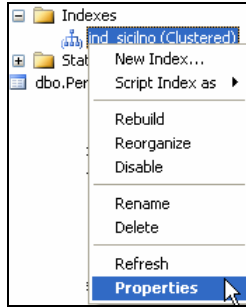
```
ALTER TABLE Personel  
ADD CONSTRAINT PK_Personel PRIMARY KEY (TCKimlikNo,Sicil_No)  
CLUSTERED
```

6.4. İndeks'te Değişiklik Yapmak

İndekslerde çeşitli amaçlar doğrultusunda değişiklikler yapılabilir.

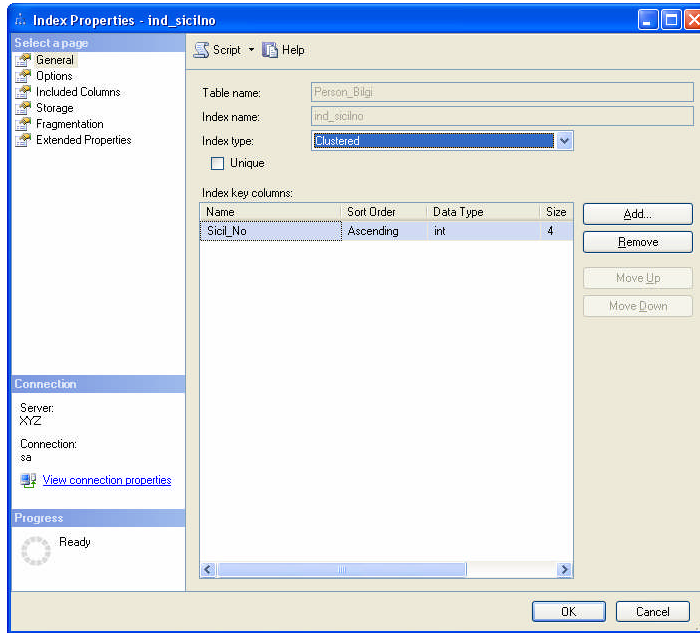
Bir indeks, yeniden derlenebilir (rebuild), yeniden düzenlenebilir (reorganize), kullanıma kapatılabilir (disable) ve seçenekleri değiştirilebilir (set).

Bir indekste değişiklik iki yolla yapılabilir: Management Studio ve DDL ifadelerinden ALTER INDEX komutu indekste değişiklikler yapılmasını sağlar. İndekste değişiklik yapmak için tabloda oluşturulmuş indeks üzerinde fareyle sağ tıklayınız ve Properties komutunu seçiniz.



Resim 6.1: İndeks düzenlemek

Properties komutunu verdiğinizde indeksle ilgili bilgiler ekrana gelecektir.



Resim 6.2: İndeks özellikleri penceresi

Bu pencere üzerinden indeksin tipini deęiřtirebilir, unique özellięi verebilir, yeni bir indeks daha ekleyebilirsiniz.

ALTER INDEX'in kullanımı ise řoyledir:

```
ALTER INDEX indeks_adi veya ALL
ON (tablo_adi.indeks_adi)
(
  REBUILD veya
  REORGANIZE veya
  DISABLE veya
  SET (indeks_seęenekleri)
)
```

6.4.1. İndeks Silmek

Oluřturulan indeks, iřlevini yitirse silinmesi gerekebilir. Bunun için, DROP INDEX komutu veya indeks özellikleri penceresi kullanılabilir.

Genel Kullanımı

```
DROP INDEX indeks_adi
ON tablo_adi
```

SQL Server 2000'de bu yazım farklıdır.

```
DROP INDEX tablo_adi.indeks_adi
```

Bir başka yol olarak da Resim 6.2'deki indeks özellikleri penceresinde bulunan Remove düęmesiyle de indeksi silebilirsiniz.

6.5. “sp_helpindex” Komutu

Bir tabloda hangi alanlarda ne tür indekslerin tanımlı olduęunu görmek mümkündür. Bu indeksleri görmek için “sp_helpindex” komutu kullanılır.

Genel Kullanımı

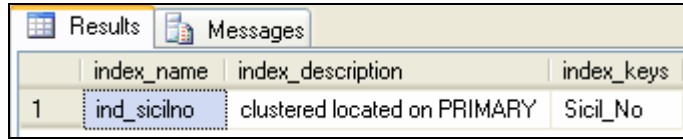
```
sp_helpindex tablo_adi
```

Örnek

“Person_Bilgi” tablosunda bulunan indeksleri görebilmek için ;

```
sp_helpindex Person_Bilgi
```

yazmak gerekir. Bu yazım, yeni bir query sayfasında olmalıdır. Query sayfasına yazılan komutu çalıştırdığınızda sonuç Resim 6.3'teki gibi olacaktır.

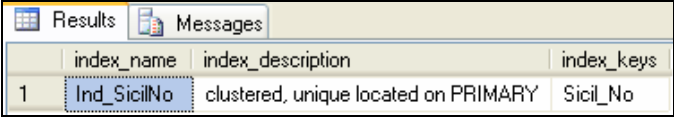
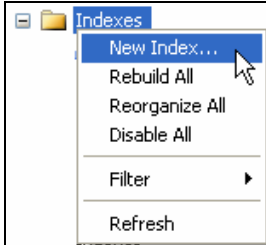


The image shows a screenshot of a SQL Server Enterprise Manager interface. At the top, there are two tabs: 'Results' and 'Messages'. Below the tabs is a table with three columns: 'index_name', 'index_description', and 'index_keys'. The table contains one row of data. The first column has the value '1', the second column has 'ind_sicilno', and the third column has 'clustered located on PRIMARY Sicil_No'. The 'ind_sicilno' cell is highlighted with a blue border.

	index_name	index_description	index_keys
1	ind_sicilno	clustered located on PRIMARY	Sicil_No

Resim 6.3: Tablodaki indeks bilgileri

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler						
<p>➤ “Person_Bilgi” tablosunda “Sicil_No” sütununa göre clustered bir indeks oluşturunuz.</p>	<pre>CREATE CLUSTERED INDEX Ind_SicilNo ON Person_Bilgi (Sicil_No)</pre> <p>Resim 6.4: İndeksin oluşturulması</p>						
<p>➤ Oluşturduğunuz indeksi siliniz.</p>	<pre>DROP INDEX Ind_SicilNo ON Person_Bilgi</pre> <p>Resim 6.5: İndeksin silinmesi</p>						
<p>➤ “Sicil_No” sütununa benzer verilerin girilmemesi için indeksi yeniden oluşturunuz.</p>	<pre>CREATE UNIQUE CLUSTERED INDEX Ind_SicilNo ON Person_Bilgi (Sicil_No)</pre> <p>Resim 6.6: Unique indeksin oluşturulması</p>						
<p>➤ Oluşturduğunuz indeks hakkında bilgi alınız.</p>	<pre>sp_helpindex Person_Bilgi</pre> <p>Resim 6.7: İndeks hakkında bilgi alınması</p>						
<p>➤ Sonuçları Results penceresinde görünüz.</p>	 <table border="1"> <thead> <tr> <th>index_name</th> <th>index_description</th> <th>index_keys</th> </tr> </thead> <tbody> <tr> <td>1 Ind_SicilNo</td> <td>clustered, unique located on PRIMARY</td> <td>Sicil_No</td> </tr> </tbody> </table> <p>Resim 6.8: Results penceresi</p>	index_name	index_description	index_keys	1 Ind_SicilNo	clustered, unique located on PRIMARY	Sicil_No
index_name	index_description	index_keys					
1 Ind_SicilNo	clustered, unique located on PRIMARY	Sicil_No					
<p>➤ Oluşturduğunuz indeksi silerek aynı işlemleri Management Studio ile yapınız.</p>	 <p>Resim 6.9: Management Studio ile indeksin oluşturulması</p>						

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. İndeks oluşturmak veriye erişimi hızlandırır (D/Y).
2. Veri ilişkilerini ve bütünlüğünü sağlayan birincil anahtar alanlar oluşturmadakullanılır.
3. Küme indekse, küme olmayan indekse indeks denir.
4. İndeks tarandığında ulaşılan sonuç verinin kendisidir (D/Y).
5. Aşağıdakilerden hangisi veriye direkt ulaşamaz?
A) Clustered B) Nonclustered C) Heap D) Yığın
6. Sorgu sonrası elde edilen kayıtlarda bir kısıtlama yoksa küme indeks araması kullanılır (D/Y).
7. Aşağıdakilerden hangisi bir indekste değişiklik yapmaz?
A) REBUILD
B) RECOGNIZE
C) DISABLE
D) SET
8. Tabloda belirtilen sütun için verilerin tekrarlanmaması gerekiyorsa Unique indeks tanımlanmaz (D/Y).
9. İndeksi silmek için komutu kullanılır.
10. İndeks hakkında bilgi almak için komutu kullanılır.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz 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 yeterliği, öğretmeniniz işlem basamaklarına göre 0 ile 4 puan arasında olacak şekilde değerlendirecektir.

DEĞERLENDİRME KRİTERLERİ	Puan
➤ Dışarıdan veri alabilme	
➤ Verilerin kopyalanacağı veritabanını seçebilme	
➤ Tablonun nasıl oluşacağını seçebilme	
➤ Kayıt işlemi ve paket tercihlerini yapabilme	
➤ Veritabanını bir metin dosyasına aktarabilme	
➤ Veritabanı ve tablolar oluşturabilme	
➤ Tablolar arası ilişki kurabilme	
➤ İlişkilerde yöntem uygulayabilme	
➤ İç içe select kullanabilme	
➤ Türetilmiş tablo oluşturabilme	
➤ İmleç tanımlayabilme	
➤ İmlecin konumunu test edebilme	
➤ İmlecin bulunduğu kayıtla çalışmasını sağlayabilme	
➤ İmleci kapatabilme	
➤ İmleci hafızadan silebilme	
➤ Tetikleyici oluşturabilme	
➤ Tetikleyiciyi kullanabilme	
➤ Görünüm oluşturabilme	
➤ Görünümü açabilme ve kapatabilme	
➤ Filtreleme uygulayabilme	
➤ İndeks oluşturabilme	
➤ Küme indeks oluşturabilme	
➤ Unique indeks oluşturabilme	
➤ İndeksi silebilme	
➤ İndeks hakkında bilgi alabilme	
Toplam (100 puan)	

DEĞERLENDİRME

Yaptığınız değerlendirme sonucunda eksikleriniz varsa öğrenme faaliyetlerini tekrarlayınız.

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

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1.	Import, Export
2.	D Şıkkı
3.	Import and Export
4.	Doğru
5.	Yanlış
6.	Data Source, Destination
7.	Bcp
8.	Yanlış
9.	Bulk Insert
10.	Doğru

ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1.	Yanlış
2.	256
3.	OUTER JOIN-NULL
4.	LEFT OUTER JOIN- RIGHT OUTER JOIN
5.	C Şıkkı
6.	C Şıkkı
7.	Doğru
8.	Doğru
9.	Yanlış
10.	Doğru

ÖĞRENME FAALİYETİ-3'ÜN CEVAP ANAHTARI

1.	Doğru
2.	Yanlış
3.	OPEN-CLOSE
4.	FETCH_STATUS
5.	DEALLOCATE
6.	@@ROWCOUNT
7.	Doğru
8.	FETCH PRIOR – FETCH LAST
9.	Doğru
10.	READ_ONLY

ÖĞRENME FAALİYETİ-4'ÜN CEVAP ANAHTARI

1.	A Şıkkı
2.	Inserted
3.	B Şıkkı
4.	D Şıkkı
5.	Yanlış
6.	Doğru
7.	Doğru
8.	@@rowcount
9.	ROLLBACK
10.	ALTER-DROP

ÖĞRENME FAALİYETİ-5'İN CEVAP ANAHTARI

1.	Doğru
2.	Doğru
3.	1024
4.	CREATE VIEW
5.	WHERE
6.	Doğru
7.	C Şıkkı
8.	B Şıkkı
9.	sp_depends
10.	ALTER-DROP

ÖĞRENME FAALİYETİ-6'NİN CEVAP ANAHTARI

1.	Doğru
2.	Tekil indeks
3.	Clustered- nonclustered
4.	Doğru
5.	B Şıkkı
6.	Yanlış
7.	B Şıkkı
8.	Yanlış
9.	DROP INDEX
10.	sp_helpindex

KAYNAKLAR

- GÖZÜDELİ Yaşar, “**Yazılımcılar İçin SQL Server 2005 ve Veritabanı Programlama**”, Seçkin Yayıncılık, Ankara, 2006.