

HİTİT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

Veri Tabanı Yönetim
Sistemleri

İÇİNDEKİLER

1. Veri Tabanı Temel Kavramları <i>Doç. Dr. ABDULKADİR ÖZDEMİR</i>	4
2. Veri Tabanı ve Normalizasyon <i>Doç. Dr. ABDULKADİR ÖZDEMİR</i>	24
3. Veri Tabanı Araçlarının Kurulumunu Yapmak <i>Dr. Öğr. Üyesi SERDAR AYDIN</i>	43
4. Tabloları Oluşturmak ve Özelliklerini Belirlemek <i>Dr. Öğr. Üyesi SERDAR AYDIN</i>	67
5. TSQL ile Veri Tabanı ve Tabloları Oluşturmak, Özelliklerini Belirlemek <i>Dr. Öğr. Üyesi SERDAR AYDIN</i>	91
6. Sorgu Oluşturmak ve Çeşitlerini Kullanmak-1 <i>Dr. Öğr. Üyesi SNAN KUL</i>	112
7. Sorgu Oluşturmak ve Çeşitlerini Kullanmak-2 <i>Dr. Öğr. Üyesi SNAN KUL</i>	136
8. Sorgu Oluşturmak ve Çeşitlerini Kullanmak - 3 <i>Dr. Öğr. Üyesi SNAN KUL</i>	157
9. Sorgu Oluşturmak ve Çeşitlerini Kullanmak - 4 <i>Ar. Gör. YAKUP BAYOĞLU</i>	180
10. İlişkili Tablolar ile Sorgu Hazırlamak <i>Ar. Gör. YAKUP BAYOĞLU</i>	203
11. Görüntü (VIEW), Store Prosedür ve Fonksiyonlar <i>Ar. Gör. YAKUP BAYOĞLU</i>	223
12. Veri Tabanı Yönetimi Yapmak <i>Dr. Öğr. Üyesi AHMET KAMALLI KABAKU</i>	248
13. Veri Tabanı Güvenliğini Sağlamak <i>Dr. Öğr. Üyesi AHMET KAMALLI KABAKU</i>	271
14. Veri Tabanında Yedekleme ve Geri Yükleme <i>Dr. Öğr. Üyesi AHMET KAMALLI KABAKU</i>	291

SORGU OLUŐTURMAK VE ÇEŐİTLERİNİ KULLANMAK -4



- Tabloya Satır Ekleme
 - INSERT Deyimi
- NULL Deęer Ekleme
- Bir Dięer Tablo Kullanarak Yeni Satır Ekleme
 - SELECT ... INTO ... FROM Deyimi

İÇİNDEKİLER



- Bu üniteyi çalıştıktan sonra;
 - Tabloya satır eklemeyi,
 - NULL deęer eklemeyi,
 - Ekleme işlemindeki kısıtları ve dikkat edilmesi gereken durumları öğrenebileceksiniz.

HEDEFLER

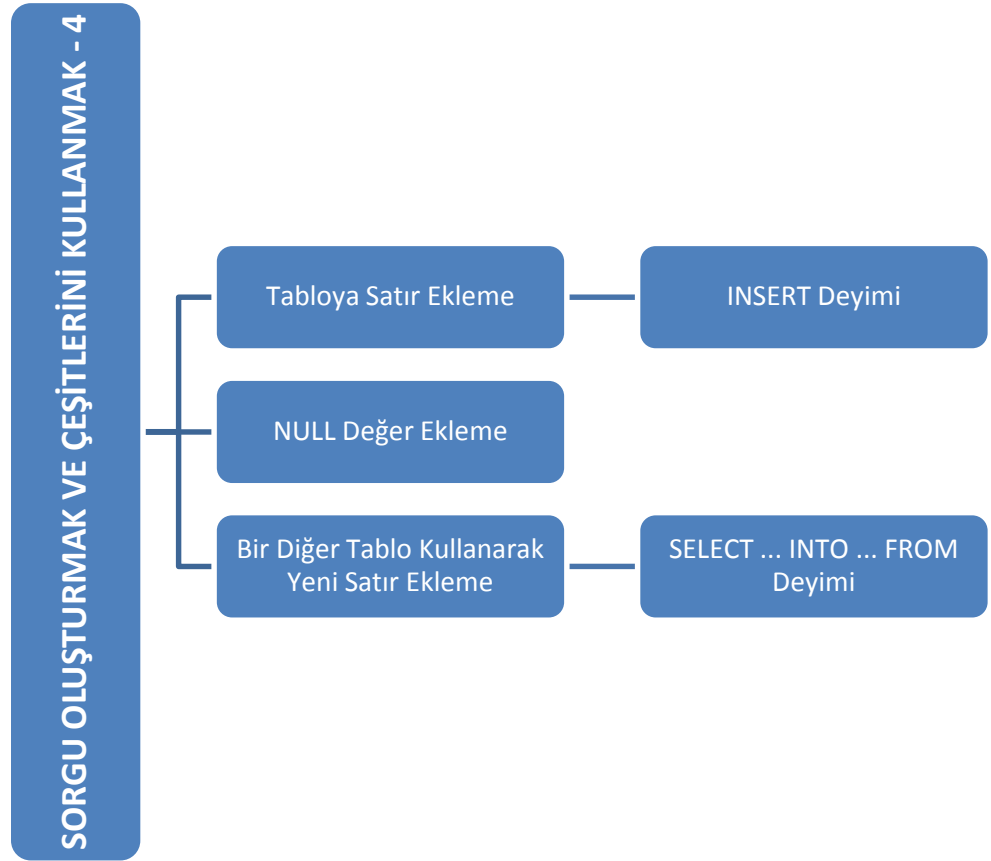


Atatürk Üniversitesi
Açıköğretim Fakültesi

VERİ TABANI YÖNETİM SİSTEMLERİ

Arş. Gör. Yakup
BAYOĞLU

ÜNİTE 9



GİRİŞ

Önceki birkaç ünite boyunca veri tabanından çeşitli sorgulamalar yapmayı ve veriler üzerinde analiz yapmayı öğrendik. Tablo(lar) üzerinde bu tür sorgulama ve analizlerin yapılabilmesi için tablolarda veri olmalıdır, aksi halde sorgular boş dönecektir. Bu ünite de tablolara nasıl veri ekleneceğini işleyeceğiz.

Tablolara veri eklemek için INSERT INTO' komutu kullanılır. Genel kullanımı "INSERT INTO Tablo Adı (Sütun1, Sütun2, Sütun3, ...) VALUES (Değer1, Değer2, Değer3, ...)" şeklindedir. Tabloya eklenecek verileri kendimiz yazabileceğimiz gibi ekleme yapılacak olan tablo veya veri tabanındaki diğer tablolardaki, hatta sunucu üzerindeki başka bir veri tabanına ait olan tablolardaki verileri kullanarak da ekleme yapılabilir. Tabloya aynı anda tek bir satır veya birden çok satır eklenebilir fakat aynı anda sadece tek bir tabloya ekleme yapılabilir. Tabloya veri eklerken veri tiplerine ve veri sırasına dikkat edilmelidir. Tabloya veri eklenirken tüm sütunlara veri girilmesi zorunlu değildir. Fakat NULL değer kabul edilmeyen sütunlara mutlaka veri girilmelidir. Tablodaki birincil anahtar sütunlara veri girilmez, bu sütunların değeri ekleme esnasında SQL Server tarafından otomatik olarak oluşturulur. Aynı şekilde tablodaki hesaplanmış sütunların da içeriği SQL Server tarafından oluşturulacağı için bu sütunlara da veri girilmez. Yabancı anahtar sütunlar için belirtilen değerlerin hedef tabloda bir karşılığı bulunmalıdır. Bir tabloya eklenen en son Id değerine ihtiyaç duyulduğunda @@IDENTITY sistem değişkeni veya IDENT_CURRENT() Fonksiyonu kullanılabilir. Ünite boyunca veri eklemenin çeşitli yöntemlerini ve veri eklerken dikkat edilmesi gereken hususları ve bazı özel durumları inceleyeceğiz.

TABLOYA SATIR EKLEME

INSERT Deyimi

Tabloya veri (satır) eklemek için 'INSERT INTO' komutu kullanılır. (Gözüdeli, 2012). Genel kullanımı aşağıdaki gibidir:

```
INSERT INTO Tablo_Adı
(
    Sütun1,
    Sütun2,
    Sütun3,
    [...]
)
VALUES
(
    Değer1,
    Değer2,
    Değer3,
    [...]
)
```



Tabloya veri eklemek için INSERT komutu kullanılır.

'INSERT INTO' deyiminden sonra tablo adı ve parantez içerisinde veri girilecek olan sütunlar yer alır. 'VALUES' deyiminden sonra yine parantez içerisinde ilgili sütunlara girilecek olan değerler yer alır.

Örnek veri tabanındaki Birey tablosunun ilk 10 kaydına bakalım:

```
SELECT TOP 10
*
FROM
    Birey
ORDER BY
    ID
```

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti	
1	2	12312312312	Hidayet	Çölkusu	Ayşe	Yusuf	Erkek
2	3	35462349764	Taha	Bayram	Fatma	Ali	Erkek
3	4	79854631320	Ayşe	Yıldız	Hayriye	Cihan	Kız
4	5	54678431348	Nazlı	Yasar	Neslihan	Erver	Kız
5	6	65874110244	Erkan	Toprak	Hatice	Cihad	Erkek
6	7	78984654164	AHMET	Akan	Özlem	Samet	Erkek
7	8	98971354212	Mustafa	Toroman	Irem	Yusuf	Erkek
8	9	38312472124	Selmani	Hatipoğlu	Meltem	Musa	Erkek
9	10	78974564762	Samet	Öztürk	Yasemin	Murat	Erkek
10	11	74547984626	Nazife	Kıbar	Aynur	Burak	Kız

Şekil 9.1. Birey Tablosundaki İlk On Kayıt

Şimdi Birey tablosuna yeni bir satır ekleyelim:

```
INSERT INTO Birey
(
    TcNo,
    Adi,
    Soyadi,
    AnneAdi,
    BabaAdi,
    Cinsiyeti
)
VALUES
(
    '12332112332',
    'Ali',
    'Demir',
    'Özge',
    'Emre',
    'Erkek'
)
```



'INSERT INTO' deyiminden sonra belirtilen sütun isimleri ile 'VALUES' deyiminden sonra yazılan değerler aynı sırada olmalıdır.

Yukarıdaki örnekte Birey tablosuna 'Ali Demir' isimli, anne adı 'Özge', baba adı 'Emre', T.C. kimlik numarası '12332112332' olan erkek bir birey eklemiştir.

Results		Messages					
ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti	
1	52	12332112332	Ali	Demir	Özge	Emre	Erkek

Şekil 9.2. Birey Tablosuna En Son Eklenen Kayıt

Not: 'INSERT INTO' deyiminden sonra belirtilen sütun isimleri ile 'VALUES' deyiminden sonra yazılan değerler *aynı sırada* olmalıdır.

Şimdi, az önce Birey tablosuna eklediğimiz 'Ali Demir' isimli kişi için Kamu Yönetimi bölümüne öğrenci kaydı ekleyelim. Bunun için Ogrenci_Bolum tablosuna ekleme yapmamız gerekecek:

```
INSERT INTO Ogrenci_Bolum
(
    Birey_ID,
    Birim_ID,
    OgrenciNo,
    KayitTarihi,
    OgrenimYili
)
VALUES
(
    52, -- Ali Demir'in Birey_ID'si
    5, -- Kamu Yönetimi'nin Birim_ID'si
    '130303555', -- Öğrenci No
    '2013-09-08', -- Kayıt Tarihi
    1
)
```

Yukarıda Ali Demir (52) için Kamu Yönetimi (5) bölümüne öğrenci numarası '130303555' ve kayıt tarihi '2013-09-08' olan 1. sınıf öğrencisi kaydı eklemiş olduk. 'INSERT INTO ... VALUES' kalıbında 'VALUES' yerine 'SELECT' deyimini de kullanılabilir. Bir önceki örneğimizi 'SELECT' deyimini ile yeniden yazalım:

```
INSERT INTO Ogrenci_Bolum
(
    Birey_ID,
    Birim_ID,
    OgrenciNo,
    KayitTarihi,
    OgrenimYili
)
SELECT
    52, -- Ali Demir'in Birey_ID'si
    5, -- Kamu Yönetimi'nin Birim_ID'si
    '130303555', -- Öğrenci No
    '2013-09-08', -- Kayıt Tarihi
    1
```



Veri girişi için değerler yazılırken karakter veya tarih veri tipindeki sütunların değerleri tek tırnak içerisinde yazılmalıdır.

Not: veri girişi için değerler yazılırken karakter veya tarih veri tipindeki sütunların değerleri tek tırnak içerisinde yazılmalıdır.

Veri girişi yapacağımız tablonun (birincil anahtar olan sütunlar hariç) tüm sütunlarına birden veri girişi yapılacaksa eğer tablo adından sonra sütunlar belirtilmeden de ekleme işlemi yapılabilir. Bir önceki örneğimizde birincil anahtar olan ID sütunu hariç tüm sütunlara veri girişi yapmıştık, aynı işlemi aşağıdaki kod ile de gerçekleştirebiliriz:

```
INSERT INTO Ogrenci_Bolum
SELECT
    52, -- Ali Demir'in Birey_ID'si
    5,  -- Kamu Yönetimi'nin Birim_ID'si
    '130303555', -- Öğrenci No
    '2013-09-08', -- Kayıt Tarihi
    1
```

Tabloya aynı anda birden fazla satır eklememiz de mümkündür; bunun için 'UNION [ALL]' deyimi kullanılır. Birey tablosuna Yusuf ve Yeliz isimli iki kardeşin kayıtlarını aynı anda ekleyelim:

```
INSERT INTO Birey
SELECT
    '47852147852',
    'Yusuf',
    'Yerli',
    'Emine',
    'Mehmet',
    'Erkek'
UNION [ALL]
SELECT
    '12365478963',
    'Yeliz',
    'Yerli',
    'Emine',
    'Mehmet',
    'Kız'
```

'UNION' deyimi ile (SQL Server'in müsaade ettiği limiti aşmamak kaydı ile) dilediğimiz kadar select ifadesini birleştirip tek bir insert deyimi ile aynı anda ekleme yapabiliriz.



Aynı anda sadece bir tabloya ekleme işlemi yapılabilir.



Bireysel Etkinlik

- 'UNION' ve 'UNION ALL' deyimlerinin farkını araştırınız.
- Bu farkın, tabloya satır ekleme işlemi nasıl etkileyeceğini düşününüz.

Not: Aynı anda sadece *bir* tabloya ekleme işlemi yapılabilir.

Tabloya yeni bir satır eklerken tüm sütunlar için veri eklemek zorunda değiliz. Örneğin bir öğrenci otomasyon sisteminde öğrenciler için, ilk olarak ön kayıt alındığını ve kesin kayıt yapıldıktan sonra kayıt tarihinin oluşturulduğunu düşünelim. Bu durumda ön kayıt alınırken Ogresnci_Bolum tablosuna ekleme işlemi aşağıdaki gibi olacaktır:

```
INSERT INTO Ogresnci_Bolum
(
    Birey_ID,
    Birim_ID,
    OgresnciNo,
    OgresnimYili
)
SELECT
    52, -- Ali Demir'in Birey_ID'si
    5, -- Kamu Yönetimi'nin Birim_ID'si
    '130303555', -- Öğrenci No
    1
```

Dikkat ederseniz yukarıdaki ekleme işleminde sütunları belirtirken KayıtTarihi sütununu belirtmedik ve dolayısıyla değerler kısmında da kayıt tarihini yazmadık, KayıtTarihi sütunu bu ekleme işleminden sonra NULL (boş) kalacaktır.

ID	Birey_ID	Birim_ID	OgresnciNo	Kayıt Tarihi	OgresnimYili
1	52	5	130303555	NULL	1

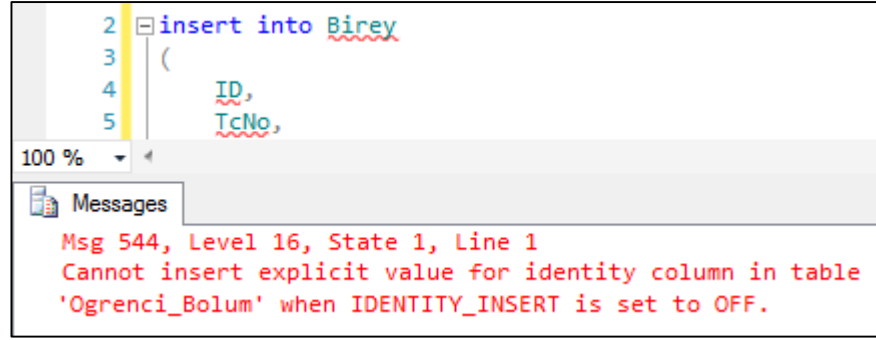
Şekil 9.3. Kayıt Tarihi Olmadan Veri Ekleme



Değeri SQL Server tarafından oluşturulan birincil anahtar durumundaki sütuna veri girişi yapılmaz (*).

Not: Değeri SQL Server tarafından oluşturulan birincil anahtar durumundaki sütuna veri girişi yapılmaz (*), bu sütun(lar)ın alacağı (sıradaki benzersiz) değer SQL Server tarafından otomatik olarak oluşturulur ve sütuna atanır. (Petkovic, 2006).

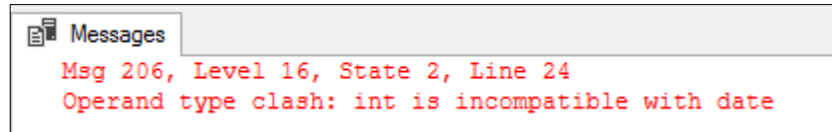
İlk örneğimizde Birey tablosuna yeni bir satır eklerken veri girişi yapılacak sütunların içerisinde birincil anahtar olan ID sütununu belirtmedik fakat ekleme işleminden sonra gördük ki ID sütununa da değer (52) SQL Server tarafından eklendi. Değeri SQL Server tarafından otomatik olarak üretilen birincil anahtar sütununa veri eklemeye çalışılması SQL Server'ın hata döndürmesine sebep olur. Bu hatanın bir örneği aşağıda gösterilmiştir.



Şekil 9.4. Değeri SQL Server Tarafından Otomatik Olarak Üretilen Birincil Anahtar Sütununa Veri Eklenmeyeceğine Dair Hata Uyarısı

Veri ekleme işleminde eklenen verilerin veri tipininin tablonun sütun veri tipleri ile uyumlu olması gerekir. Örneğin Ogrenci_Bolum tablosuna ekleme yaparken tarih veri tipindeki KayitTarihi sütununa tamsayı bir değer eklersek şekil 9.5.teki hatayı alırız:

```
INSERT INTO Ogrenci_Bolum
(
    Birey_ID,
    Birim_ID,
    OgrenciNo,
    KayitTarihi,
    OgrenimYili
)
VALUES
(
    52, -- Ali Demir'in Birey_ID'si
    5,  -- Kamu Yönetimi'nin Birim_ID'si
    '130303555', -- Öğrenci No
    2013,      -- Kayıt Tarihi
    1
)
```



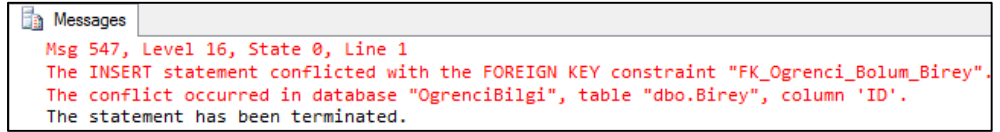
Şekil 9.5. Sütun Veri Tipinden Farklı Bir Veri Tipi ile Kayıt Eklenildiğinde SQL Server'ın Verdiği Hata Mesajı

Dikkate alınması gereken bir diğer husus da yabancı anahtar sütunlarına girilen değerlerin referans tabloda bir karşılığının olması gerektiğidir. Ogrenci_Bolum tablosundaki Birey_ID sütunu Birey Tablosundaki ID sütununa, Birim_ID sütunu da Birim tablosundaki ID sütununa yabancı anahtar olduğu için Birey_ID ve Birim_ID sütunlarına eklenen verilerin ilgili tablolarda karşılığı olması gerekmektedir. Bir önceki örneğimizde Birey_ID sütunu için 52 değerini değil de 525 değerini girmiş olsa idik Birey tablosunda 525 ID değerine sahip bir kayıt olmadığı için SQL Server hata verecekti:

```

INSERT INTO Ogrenci_Bolum
(
    Birey_ID,
    Birim_ID,
    OgrenciNo,
    OgrenimYili
)
SELECT
    525, -- Ali Demir'in Birey_ID'si
    5,   -- Kamu Yönetimi'nin Birim_ID'si
    '130303555', -- Öğrenci No
    1

```



Şekil 9.6. Yabancı Anahtar Bir Sütuna Referans Tabloda Karşılığı Olmayan Bir Değer Girildiğinde SQL Server'in Verdiği Hata



Hesaplatılmış
sütunlara veri
eklenemez.

Satır ekleme işleminde dikkat edilmesi gereken bir diğer husus hesaplatılmış sütunlara veri eklenemeyecidir. Üniversite akademik takvim bilgilerini tutmak üzere AkademikYil isimli bir tablo oluşturalım:

```

CREATE TABLE AkademikYil
(
    ID                INT IDENTITY,
    BaslangicYili    INT NOT NULL,
    BitisYili        AS BaslangicYili + 1,
    AkademikYilAdi  AS CAST(BaslangicYili AS
    VARCHAR(4))
                    + ' - '
                    + CAST(BaslangicYili + 1 AS
    VARCHAR(4))
                    + ' Akademik Yılı',
    BaslangicTarihi  DATE NULL,
    BitisTarihi      DATE NULL
)

```

Yukarıdaki AkademikYil tablosunda, BitisYili ve AkademikYilAdi kolonları BaslangicYili sütunundan türetilmektedir (hesaplanmaktadır). BaslangicYili sütununa 2013 değeri girelim ve diğer sütunların alacağı değerleri görelim:

```

INSERT INTO AkademikYil
(
    BaslangicYili,
    BaslangicTarihi,
    BitisTarihi
)
SELECT
    2013,
    '2013-09-23',
    '2014-09-21'

```

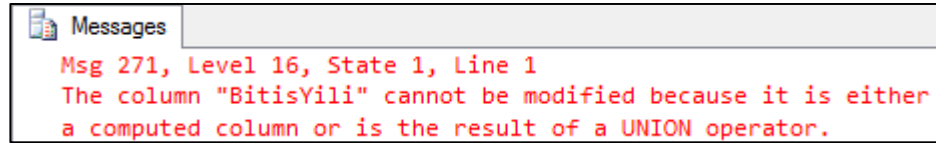
```
SELECT * FROM AkademikYil
```

ID	BaslangicYili	BitisYili	AkademikYilAdi	BaslangicTarihi	BitisTarihi
1	2013	2014	2013 - 2014 Akademik Yili	2013-09-23	2014-09-21

Şekil 9.7. Hesaplatılmış Sütun Örneği

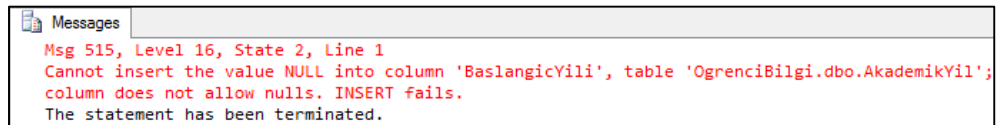
BaslangicYili sütununa 2013 değeri girildiğinde Şekil 9.6.da görüldüğü gibi BitisYili sütununda 2014, AkademikYilAdi sütununda da '2013 - 2014 Akademik Yılı' değerleri oluştu (hesaplatıldı).

BitisYili ve AkademikYilAdi sütunları hesaplatılmış sütun oldukları için bu sütunlara veri girilmek istendiğinde SQL Server aşağıdaki hatayı döndürür.



Şekil 9.8. Hesaplatılmış Bir Sütuna Veri Eklemeye Çalışıldığında SQL Server'in Verdiği Hata

Tabloya yeni bir satır eklenirken tablo tasarımında NULL değer girilmesine izin verilmeyen sütunlar için mutlaka bir değer girilmelidir. Yukarıda AkademikYil tablosu oluşturulurken BaslangicYili sütunu bu şekilde tanımlanmış bir sütun olduğu için AkademikYil tablosuna satır ekleneceği zaman BaslangicYili sütununa mutlaka bir değer girilmelidir. Aksi takdirde SQL server aşağıdaki hata uyarısını verir:



Şekil 9.9. NULL Değer Girilmesine İzin Verilmeyen Bir Sütuna Veri Girilmeden Satır Eklemeye Çalışıldığında SQL Server'in Verdiği Hata Uyarısı

Tablo tasarımında varsayılan (default) değer belirtilmiş bir sütun var ise ve yeni satır ekleme işleminde bu sütun için bir değer girilmemiş ise sütunda NULL değer değil, varsayılan değer oluşur. Örnek olarak Ders tablosunda bool tipinde ZorunluMu isimli bir sütunun olduğunu ve varsayılan değerinin 'true' olduğunu varsayalım. Bu durumda Ders tablosuna yeni satır eklerken ZorunluMu sütunu için bir değer girilmişse eğer girilen değer, girilmemişse varsayılan değer (true) ZorunluMu sütununda oluşur. Bu şekilde yeni ders eklerken zorunlu mu, seçmeli mi olduğu bilinmeyen dersler zorunlu kabul edilerek eklenmiş olur.

NULL Değer Ekleme

Tabloya NULL değer ekleme iki şekilde gerçekleştirilebilir.

Birinci durum: Daha önce de belirtildiği üzere veri eklenirken 'INSERT INTO' deyiminden sonra belirtilmeyen sütunlara herhangi bir değer eklenmez, yani sütun içeriği NULL olur.

İkinci durum: Veri girilecek sütunlar belirtilirken NULL olması istenen sütun(lar) da dâhil edilir ve 'SELECT' kısmında bu sütunlar için NULL değeri yer alır. Daha önce Öğrenci_Bolum tablosuna KayıtTarihi sütunu NULL olacak şekilde veri eklemiştik ki o örnek birinci durum için uygun bir örnektir. Şimdi aynı ekleme işlemini ikinci durumda bahsettiğimiz şekilde yeniden yazalım:

```
INSERT INTO Öğrenci_Bolum
(
    Biyey_ID,
    Birim_ID,
    ÖğrenciNo,
    KayıtTarihi,
    ÖğrenimYili
)
SELECT
    52, -- Ali Demir'in Biyey_IDsi
    5,  -- Kamu Yönetimi'nin Birim_ID'si
    '130303555', -- Öğrenci No
    NULL,      -- Kayıt Tarihi
    1
```



Tabloya satır eklerken
NULL değer yazılabilir.

Bir Diğer Tablo Kullanarak Yeni Satır Ekleme

Şimdiye kadar tabloya eklediğimiz verileri hep değerleri kendimiz yazmak sureti ile gerçekleştirdik. Eklenacak verilerin başka bir tablo veya tablolardan getirilmesi de mümkündür. Hatta aynı sunucu üzerinde yer alan farklı bir veri tabanındaki tablolardan da getirilebilir.

Kullanıcı adında yeni bir tablo oluşturalım ve içerisine kullanıcı adları T.C. Kimlik numaraları, şifreleri de isim.soyisim formatında olacak şekilde her birey için veri ekleyelim:

```
CREATE TABLE Kullanici
(
    ID          INT IDENTITY,
    BiyeyId     INT,
    KullaniciAdi VARCHAR(50),
    Sifre       VARCHAR(50)
)

INSERT INTO Kullanici
SELECT
    ID,
    TcNo,
    Adi + '.' + Soyadi
FROM
    Biyey

SELECT * FROM Kullanici
```

	ID	BireyId	KullaniciAdi	Sifre
1	1	2	12312312312	Hidayet.Çölkusu
2	2	3	35462349764	Taha.Bayram
3	3	4	79854631320	Ayse.Yildiz
4	4	5	54678431348	Nazli.Yasar
5	5	6	65874110244	Erkan.Toprak

Şekil 9.10. Kullanici Tablosunun İlk Beş Satırı

SELECT ... INTO ... FROM Deyimi

Kullanici tablosunda birtakım denemeler yapmak istersek ve verileri bozmamak için de denemeleri üzerinde yapmak için tablonun bir kopyasını almak istersek yazacağımız kod aşağıdaki gibi olacaktır.

```
CREATE TABLE KullaniciYedek
(
    ID          INT,
    BireyId     INT,
    KullaniciAdi VARCHAR(50),
    Sifre       VARCHAR(50)
)

INSERT INTO KullaniciYedek
SELECT
*
FROM
    Kullanici
```



'SELECT ... INTO ... FROM' deyimi ile aynı anda tablo oluşturup içine satır(lar) eklenilebilir.

Yukarıda önce KullaniciYedek isimli bir tablo oluşturduk, sonra bu tabloya Kullanici tablosundaki tüm satırları ekledik. Sıfırdan bir tablo oluşturup içerisine veri ekleneceği zaman ne yapmamız gerektiğini yukarıdaki örnekte gördük. Fakat bu işlemi 'SELECT ... INTO ... FROM' komutu ile daha pratik bir şekilde gerçekleştirebiliriz. Bu komutun genel kullanımını aşağıdaki gibidir:

```
SELECT
    sütun1,
    sütun2
[... ]
INTO
    YeniTabloAdı
FROM
    ReferansTabloAdı
```

SELECT ifadesinden sonra gelen sütunları içeren bir tablo oluşturulur ve select sorgusunun sonucu bu tabloya eklenir. Bir önceki örneğimizi yeniden yazalım:

```
SELECT
*
INTO
```

```
KullaniciYedek
FROM
Kullanici
```

(*) Birincil anahtar sütununa veri ekleme

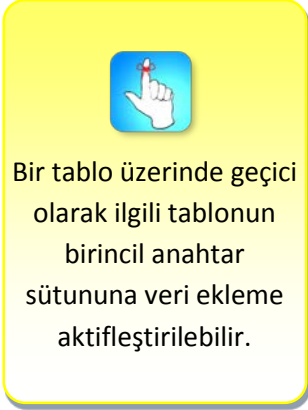
Normâlde değeri SQL Server tarafından otomatik olarak üretilen birincil anahtar sütununa veri eklenmez. İstisnai bir durum olarak böyle bir birincil anahtar sütununa veri ekleme ihtiyacı duyarsak geçici olarak ilgili tablonun birincil anahtar sütununa veri eklemeyi aktifleştirebiliriz. Mesela bu ünitenin ilk örneğinde Birey tablosuna eklenen Ali Demir'in kaydının (ID numarası 52 olan kayıt) tablodan silindiğini ama Birey tablosunun ID sütununa yabancı anahtar barındıran Ogrenci_Bolum, Kullanici, Ogrenci_Ders vs. gibi diğer tablolardaki Birey_ID değeri 52 olan satırların hala mevcut olduğunu varsayalım. 'Ali Demir' kaydını yeniden eklenirse bir önceki seferde aldığı ID değerinden farklı bir ID değeri alacaktır ve bu durumda Ali Demir'e ait diğer tablolardaki kayıtlar, yeni eklenen Ali Demir kaydı ile ilişkili olmayacaktır; çünkü diğer tablolarda yer alan 'Ali Demir'e ait bilgi içeren satırlarda Birey_ID sütununun içeriği hâlâ 52 dir ama 'Ali Demir'in Birey tablosundaki ID değeri artık 52 değildir. Dolayısı ile silindikten sonra Birey tablosuna aynı bireyi yine eski birey_ID ile yeniden eklemeye ihtiyacımız olabilir, bu durumda yazacağımız kod aşağıdaki gibi olacaktır:

```
SET IDENTITY_INSERT Birey ON
```

```
INSERT INTO Birey
(
    ID,
    TcNo,
    Adi,
    Soyadi,
    AnneAdi,
    BabaAdi,
    Cinsiyeti
)
SELECT
(
    52,
    '12332112332',
    'Ali',
    'Demir',
    'Özge',
    'Emre',
    'Erkek'
)
```

```
SET IDENTITY_INSERT Birey OFF
```

'SET IDENTITY_INSERT Birey ON' komutu ile Birey tablosu için geçici olarak birincil anahtar sütununa veri eklemeye izin verilmiş oldu ve devamında 'Ali Demir' kaydı eski ID değeri (52) ile yeniden eklenmiş oldu. Son satırda yer alan



'SET IDENTITY_INSERT Birey OFF' komutu ile Birey tablosuna geçici olarak verilmiş olan birincil anahtar sütununa veri ekleme izni kaldırılmış oldu.

Tablonun birincil anahtar sütununa veri ekleme özelliği aktif hâle getirildiği zaman, işlem bittikten hemen sonra bu özelliğin geri kapatılması unutulmamalıdır. Yukarıdaki kod, eklenmek istenen birincil anahtar değerinin tabloda yer alıp almadığından bağımsız olarak çalışır; yani Birey tablosunda hâlihazırda Birey_ID değeri 52 olan bir satır var mıdır diye kontrol edilmeden çalışır. Dolayısı ile bu şekilde birincil anahtar sütununa veri eklerken çok dikkatli olmak gerekir. Zira Birey tablosunda aynı Id değerinin birden fazla kere bulunması diğer tablolarla ilişki kurarken sorun teşkil edecektir.

@@IDENTITY değişkeni

@@IDENTITY bir sistem değişkenidir ve veri tabanına en son eklenen birincil anahtar değerini içinde barındırır.

Bazen bir tabloya bir satır veri ekledikten sonra, son eklenen bu satırdaki kayıt için kullanılan Id değerinin bilinmesine ihtiyaç olabilir. Örneğin bir Öğrenci Bilgi Sistemine yeni bir öğrenci kaydı eklediğimizi düşünelim. Öğrencinin kaydı sisteme eklenirken farklı tablolara kayıtlar eklenecektir. Öncelikle kişinin özlük bilgilerinin yer aldığı Birey tablosuna kayıt eklenecektir. Akabinde de Öğrenci_Bolum tablosuna bu yeni öğrenci için kayıt eklenecektir. Öğrenci_bolum tablosuna kayıt eklenirken öğrencinin Birey tablosundaki Birey_Id değeri bilinmelidir. İşte bu noktada @@Identity sistem değişkeni kullanılabilir. Bu ünitenin başlarında Ali Demir isimli bir öğrenciyi önce Birey tablosuna eklemiş ve akabinde de Ali Demir için Öğrenci_Bolum tablosuna veri eklemiştik. Bu işlemi aşağıdaki gibi yapabiliriz:

```
INSERT INTO Birey
(
    TcNo,
    Adi,
    Soyadi,
    AnneAdi,
    BabaAdi,
    Cinsiyeti
)
VALUES
(
    '12332112332',
    'Ali',
    'Demir',
    'Özge',
    'Emre',
    'Erkek'
)
```

```
INSERT INTO Öğrenci_Bolum
(
    Birey_ID,
```




```

        Birim_ID,
        OgrenciNo,
        KayitTarihi,
        OgrenimYili
    )
VALUES
(
    @@IDENTITY, -- Veri tabanına en son eklenen Id
    deęeri
    5, -- Kamu Yönetimi'nin Birim_ID'si
    '130303555', -- Öğrenci No
    '2013-09-08', -- Kayıt Tarihi
    1
)

```

Yukarıda Ali Demirin kaydı Birey tablosuna eklendi. Bu işlemin sonunda @@IDENTITY değişkeninde 52 değeri mevcuttur çünkü veri tabanında en son olarak Birey tablosuna veri eklenmiştir ve Birey tablosuna eklenen son ID değeri de 52 dir. Ogrenci_Bolum tablosuna veri eklerken de Birey_Id sütununa eklenecek değer olarak @@IDENTITY değişkeni kullanılmıştır.

IDENT_CURRENT() Fonksiyonu

IDENT_CURRENT() fonksiyonu parametre olarak tablo adı alır ve bu tabloya en son eklenen Id değerini döner.



Belirli bir tabloya eklenen en son ID değeri IDENT_CURRENT() fonksiyonu ile öğrenilebilir.

@@IDENTITY değişkeninin sağladığı kolaylığı yukarıda gördük. Fakat @@IDENTITY değişkeni her zaman istediğimiz sonucu vermeyebilir. Birey tablosunda meydana gelen değişikliklerin (ekleme, silme ve güncelleme) tetikleyiciler vasıtasıyla Birey_Log tablosunda tutulduğunu varsayalım. Bu durumda Birey tablosuna bir kayıt eklendiğinde otomatik olarak Birey_Log tablosuna da bir kayıt eklenecektir. Dolayısı ile Birey tablosuna Ali Demir'in kaydı eklendikten hemen sonra Birey_Log tablosuna da bir kayıt eklenecektir. Bu durumda @@IDENTITY değişkeninin içerisinde 52 yani Birey tablosuna en son eklenen Id değeri değil, Birey_Log tablosuna eklenen en son ID değeri yer alacaktır. Fakat bize Birey_log tablosuna eklenen en son Id değeri değil, Birey tablosuna eklenen en son Id değeri lazımdır. İşte bu noktada IDENT_CURRENT() fonksiyonu kullanılabilir. IDENT_CURRENT('Birey') bize Birey tablosuna en son eklenen Id değerini dönecektir. Bu durumda bir önceki örneğimizi aşağıdaki gibi yeniden yazabiliriz:

```

INSERT INTO Birey
(
    TcNo,
    Adi,
    Soyadi,
    AnneAdi,
    BabaAdi,
    Cinsiyeti
)
VALUES
(

```

```
        '12332112332' ,
        'Ali' ,
        'Demir' ,
        'Özge' ,
        'Emre' ,
        'Erkek'
    )

INSERT INTO Ogrenci_Bolum
(
    Birey_ID,
    Birim_ID,
    OgrenciNo,
    KayitTarihi,
    OgrenimYili
)
VALUES
(
    IDENT_CURRENT('Birey'), -- Birey tablosuna en son
eklenen ID değeri
    5, -- Kamu Yönetimi'nin Birim_ID'si
    '130303555' , -- Öğrenci No
    '2013-09-08' , -- Kayıt Tarihi
    1
)
```



Bireysel Etkinlik

- Birey tablosuna:
 - 'INSERT INTO ... VALUES' kalıbını kullanarak yeni kayıt ekleyiniz.
 - 'INSERT INTO ... SELECT' kalıbını kullanarak yeni kayıt ekleyiniz.
 - Bazı sütunların içeriği NULL kalacak şekilde ekleme yapınız.
 - UNION deyimini kullanarak aynı anda birden fazla kayıt ekleyiniz.
 - Aynı değerleri içeren select ifadelerini 'UNION' deyimini ile birleştirerek ekleme yapınız.
 - Aynı değerleri içeren select ifadelerini bu sefer 'UNION ALL' deyimini ile birleştirerek ekleme yapınız ve 'UNION' ile 'UNION ALL' deyimlerinin farkını gözlemleyiniz.
- Bir önceki adımda birey tablosuna eklediğiniz kişiler için Öğrenci_Bolum tablosuna kayıtlar ekleyiniz.
- Birey tablosunun bir kopyasını oluşturunuz ve Birey tablosundaki tüm verileri oluşturduğunuz kopya tabloya ekleyiniz.
- Bir önceki adımda yaptığınız işlemleri 'SELECT ... INTO ... FROM' kalıbı kullanarak tekrarlayınız.
- Birincil anahtar sütununa ekleme yapmaya çalışınız ve SQL Server'in vereceği hatayı görünüz.
- Birincil anahtar sütununa ekleme yapmaya izin veriniz ve bir önceki adımdaki işlemi yapınız işiniz bitince az önce verdiğiniz birincil anahtar sütununa ekleme iznini kaldırınız.
- Az önce oluşturduğunuz yedek Birey tablosuna yeni bir (hesaplatılmış) sütun ekleyiniz ve bu sütun için veri eklemeye çalışınız ve SQL Server'in vereceği hatayı gözlemleyiniz.



Özet

- Tabloya veri (sıra) eklemek için 'INSERT INTO' komutu kullanılır.
- 'INSERT INTO' komutunun genel kullanımı "INSERT INTO Tablo_Adı (Sütun1, Sütun2, Sütun3, ...) VALUES (Değer1, Değer2, Değer3, ...)" şeklindedir.
- 'INSERT INTO' deyiminden sonra belirtilen sütun isimleri ile 'VALUES' deyiminden sonra yazılan değerler aynı sırada olmalıdır.
- 'INSERT INTO ... VALUES' kalıbında 'VALUES' yerine 'SELECT' deyimini de kullanılabilir.
- Veri girişi için değerler yazılırken karakter veya tarih veri tipindeki sütunların değerleri tek tırnak içerisinde yazılmalıdır.
- Veri girişi yaparken girilen verilerin tipi, tablodaki sütun tipleri ile aynı olmalıdır.
- Veri girişi yapacağımız tablonun (birincil anahtar olan sütunlar hariç) tüm sütunlarına birden veri girişi yapılacaksa eğer tablo adından sonra sütunlar belirtilmeden de ekleme işlemi yapılabilir.
- Tabloya yeni bir sıra eklerken tüm sütunlar için veri eklemek zorunlu değildir.
- Tabloya aynı anda tek bir sıra veya birden çok sıra eklenebilir fakat aynı anda sadece tek bir tabloya ekleme yapılabilir.
- Tabloya aynı anda birden çok sıra eklemek için 'UNION [ALL]' deyimini kullanılır.
- Tabloya eklenecek verileri kendimiz yazabileceğimiz gibi ekleme yapılacak olan tablo veya veri tabanındaki diğer tablolardaki, hatta sunucu üzerindeki başka bir veri tabanına ait olan tablolardaki verileri kullanarak da ekleme yapılabilir.
- Veri eklenirken 'INSERT INTO' deyiminden sonra belirtilmeyen sütunlara herhangi bir değer eklenmez, sütun içeriği NULL olur.
- Veri eklenirken ilgili sütun değeri olarak NULL yazılarak da tabloya NULL değer ekleme işlemi gerçekleştirilebilir.
- Tablo tasarımında NULL değer girilmesine izin verilmeyen sütunlar için mutlaka bir değer girilmelidir.
- Tablodaki birincil anahtar durumundaki sütun(lar)a veri girişi yapılamaz (bu kuralın istisnası "Birincil anahtar sütununa veri ekleme" başlığı altında belirtildi); bu sütun(lar)ın alacağı değer SQL Server tarafından otomatik olarak oluşturulur ve sütuna atanır.
- Birincil anahtar sütununa veri eklemeye çalışılması SQL Serverin hata döndürmesine sebep olur.
- Hesaplatılmış sütunlara veri eklenemez. Bu sütunların içeriği SQL Server tarafından otomatik olarak oluşturulur.



Özet (devamı)

- Yabancı anahtar sütunlarına girilen değerlerin referans tabloda bir karşılığının olması gerekir.
- Tablo tasarımında varsayılan (default) değer belirtilmiş bir sütun var ise ve yeni satır ekleme işleminde bu sütun için bir değer girilmemiş ise sütunda NULL değer değil, varsayılan değer oluşur.
- "SELECT ... INTO ... FROM" deyimi ile aynı anda bir tablo oluşturulup içerine veri eklenebilir. Bu kullanımda SELECT ifadesinden sonra gelen sütunları içeren bir tablo oluşturulur ve select sorgusunun sonucu bu tabloya eklenir.
- İstisnai bir durum olarak birincil anahtar sütununa veri ekleme ihtiyacı duyulduğunda geçici olarak tablonun birincil anahtar sütununa veri eklemeye izin verilebilir. Bunun için SET IDENTITY_INSERT '
- TabloAdı' ON komutu kullanılır. Ekleme işlemi bittikten sonra SET IDENTITY_INSERT 'TabloAdı' OFF komutu ile izin kaldırılması unutulmamalıdır.
- @@IDENTITY bir sistem değişkenidir ve veri tabanına en son eklenen birincil anahtar değerini içinde barındırır.
- IDENT_CURRENT() fonksiyonu parametre olarak tablo adı alır ve bu tabloya en son eklenen id değerini döner.

DEĞERLENDİRME SORULARI

1. Tabloya veri eklemek için hangi komut kullanılır?
 - a) SELECT
 - b) INSERT
 - c) UPDATE
 - d) DELETE
 - e) TRUNCATE
2. Bir sorgu içerisinde aynı anda kaç farklı tabloya veri eklenebilir?
 - a) 1
 - b) 2
 - c) 256
 - d) 1024
 - e) Sınırsız
3. Tabloya değerleri kendimiz yazarak veri ekleme işleminde birden fazla satırın aynı anda eklenebilmesi için hangi komut kullanılır?
 - a) INNER JOIN
 - b) LEFT JOIN
 - c) RIGHT JOIN
 - d) CROSS JOIN
 - e) UNION

```
... .. Tablo1
...
Sütun1,
Sütun2,
Sütun3
...
Tablo2
```

4. Bir tabloya, başka bir tablodaki verileri kullanarak ekleme işlemi yapan yukarıdaki kod içerisinde boş bırakılan yerlere sırası ile seçeneklerden hangisi gelmelidir?
 - a) SELECT – INTO – INSERT – FROM
 - b) SELECT – INSERT – INTO – FROM
 - c) INSERT – INTO – FROM – SELECT
 - d) INSERT – INTO – SELECT – FROM
 - e) INSERT – SELECT – INTO – FROM

```
INSERT INTO Ogrenci_Bolum
(
    Birey_ID,
    Birim_ID,
    OgrenciNo,
    OgrenimYili
)
VALUES (52, 5, '130303555', 1)
```

5. Yukarıdaki kod ile Ogrenci_Bolum tablosuna bir satır eklendikten sonra Ogrenci_Bolum tablosunun birincil anahtar sütunu olan ID sütununun içeriği ne olur?
- NULL değer yazılır.
 - Rasgele bir değer yazılır.
 - Herhangi bir değer yazılmaz.
 - Sıradaki benzersiz değer yazılır.
 - ID sütununa en son verilen değer yazılır.

```
INSERT INTO Ogrenci_Bolum
(
    Birey_ID,
    Birim_ID,
    OgrenciNo,
    OgrenimYili
)
SELECT 52, 5, '130303553', 1
UNION
SELECT 53, 6, '130303553', 1
UNION
SELECT 53, 7, '130303553', 1
```

6. Yukarıdaki kod ile Ogrenci_Bolum tablosuna kaç satır eklenir?
- 0
 - 1
 - 2
 - 3
 - 4

I. Hesaplatılmış sütuna veri eklenemez.

II. Varsayılan değere sahip sütuna veri eklenemez.

III. NULL veri kabul etmeyen sütun için değer yazılması zorunludur.

7. Tabloya satır ekleme ile ilgili olarak aşağıdakilerin hangisi ya da hangileri yanlıştır?
- Yalnız I
 - Yalnız II
 - Yalnız III
 - I ve II
 - II ve III

```
...  
    Sütun1,  
    Sütun2  
...  
    Tablo2  
...  
    Tablo1
```

8. Başka bir tablodaki verileri kullanarak ekleme işlemini, eklenecek olan tabloyu o anda oluşturarak yapan yukarıdaki kod içerisinde boş bırakılan yerlere sırası ile aşağıdakilerden hangisi gelmelidir?
- SELECT – INTO – FROM
 - INSERT – INTO – FROM
 - INSERT – SELECT – FROM
 - SELECT – INSERT – FROM
 - INSERT – INTO – SELECT
9. **'SET IDENTITY_INSERT Birey ON'** komutu Birey tablosu için ne yapar?
- Tabloya veri eklenmesini engeller.
 - Tabloya veri eklenmesine izin verir.
 - Tablo için birincil anahtar sütununa veri eklenmesine izin verir.
 - Tablo için yabancı anahtar sütununa veri eklenmesine izin verir.
 - Tablo için hesaplatılmış sütununa veri eklenmesine izin verir.
- I. 'INSERT INTO ... VALUES' kalıbında 'VALUES' yerine 'SELECT' deyimini kullanılabilir.
- II. Yabancı anahtar sütunlarına girilen değerlerin referans tabloda bir karşılığının olması gerekir.
- III. Tabloya yeni bir satır eklerken tüm sütunlar için veri eklemek zorunludur.
10. Tabloya veri ekleme ile ilgili olarak yukarıdakilerden hangisi ya da hangileri doğrudur?
- Yalnız I
 - I ve II
 - I ve III
 - II ve III
 - I, II ve III

Cevap Anahtarı

1.b, 2.a, 3.e, 4.d, 5.d, 6.d, 7.b, 8.a, 9.c, 10.b

YARARLANILAN KAYNAKLAR

Ekeleme Örnekleri (Transact-SQL) 03.06.2019 tarihinde

[http://technet.microsoft.com/tr-](http://technet.microsoft.com/tr-tr/library/dd776381%28v=sql.105%29.aspx)

[tr/library/dd776381%28v=sql.105%29.aspx](http://technet.microsoft.com/tr-tr/library/dd776381%28v=sql.105%29.aspx) adresinden erişildi.

Gözüdeli, Y. (2012). Yazılımcılar için SQL Server 2008 R2 & Veri tabanı

Programlama, 5. Baskı, Ankara: Seçkin Yayıncılık

INSERT (Transact-SQL) 02.06.2019 tarihinde [http://technet.microsoft.com/en-](http://technet.microsoft.com/en-us/library/ms174335%28v=sql.105%29.aspx)

[us/library/ms174335%28v=sql.105%29.aspx](http://technet.microsoft.com/en-us/library/ms174335%28v=sql.105%29.aspx) adresinden erişildi.

INSERT (Transact-SQL) 02.06.2019 tarihinde [http://technet.microsoft.com/tr-](http://technet.microsoft.com/tr-tr/library/ms174335%28v=sql.105%29.aspx)

[tr/library/ms174335%28v=sql.105%29.aspx](http://technet.microsoft.com/tr-tr/library/ms174335%28v=sql.105%29.aspx) adresinden erişildi.

INSERT Examples (Transact-SQL) 03.06.2019 tarihinde

[http://technet.microsoft.com/en-](http://technet.microsoft.com/en-us/library/dd776381%28v=sql.105%29.aspx)

[us/library/dd776381%28v=sql.105%29.aspx](http://technet.microsoft.com/en-us/library/dd776381%28v=sql.105%29.aspx) adresinden erişildi.

Özhan, C. (2013). İleri Seviye T-SQL Programlama, 1. Baskı, İstanbul: KODLAB

Yayınevi

Petkovic, D. (2006). Microsoft SQL Server, 1. Baskı, İstanbul: Alfa Yayınları