

T-SQL VE SORGU

Anahtar (Key)

USE, GO ve Print Komutu

Veri Tanımlama Dili(create, alter, drop)

Anahtar (Key)

- *Anahtar bir veya birden fazla alanın bir satır için niteleyici olarak girilmesi için zorlanan bir çeşit zorlayıcıdır.*
- *2 çeşit anahtar vardır:*
 - *Birincil Anahtar (Primary Key)*
 - *Yabancı Anahtar (Foreign Key)*

Birincil anahtar(Primary Key)

- *Bir kayıta ulaşmayı sağlayacak anahtar veridir.*
- *Örneği öğrenciler arasında iki Ahmet var. Arama yaparken istediğimiz Ahmet'i bulmak için her bir öğrenciye özel bir numara olmalıdır.*
- *Örneğin öğrenci numarası*
- *Birden fazla alanda birlikte birincil anahtar olabilir*

Yabancı anahtar(Foreign Key)

- *Bir tabloya girilebilecek kayıtları başka bir tablonun belli alanındaki verilerle
 - *sınırlandırmaya ve*
 - *ilişkilendirmeye yarar.**
- *Örneğin öğrencilerin not verilerinin girildikleri tablodaki her satıra öğrenci bilgileri tablosundaki öğrenci no ile eşleşen bir değer girilmesi gibi.*

IDENTITY(Kimlik)

- *SQL Server'da IDENTITY, kayıt ekleme gibi işlemlerde büyük öneme sahiptir.*
- *Her bir kayıt için otomatik olarak bir numara verir.*
- *Bir sütunu, IDENTITY sütun olarak belirlediğinizde, SQL Server her kayıt için o sütuna sayı atar.*

- *Atanan sayı, seed (başlangıç) ve increment (artış miktarı) kadar artar ya da azalır.*
- *Örneğin, 5'den başlar ve beşer beşer artırabilirsiniz. Bu durumda ilk kayıt için 5 değeri verilir ve 5, 10, 15 şeklinde artarak devam eder.*
- *IDENTITY sütunlar sayısal olmak zorundadır. Veri tipi olarak da INT (integer) ve BIGINT kullanımı yaygındır. Ancak bunlardan daha küçük sayısal veri tipleri de kullanılabilir.*

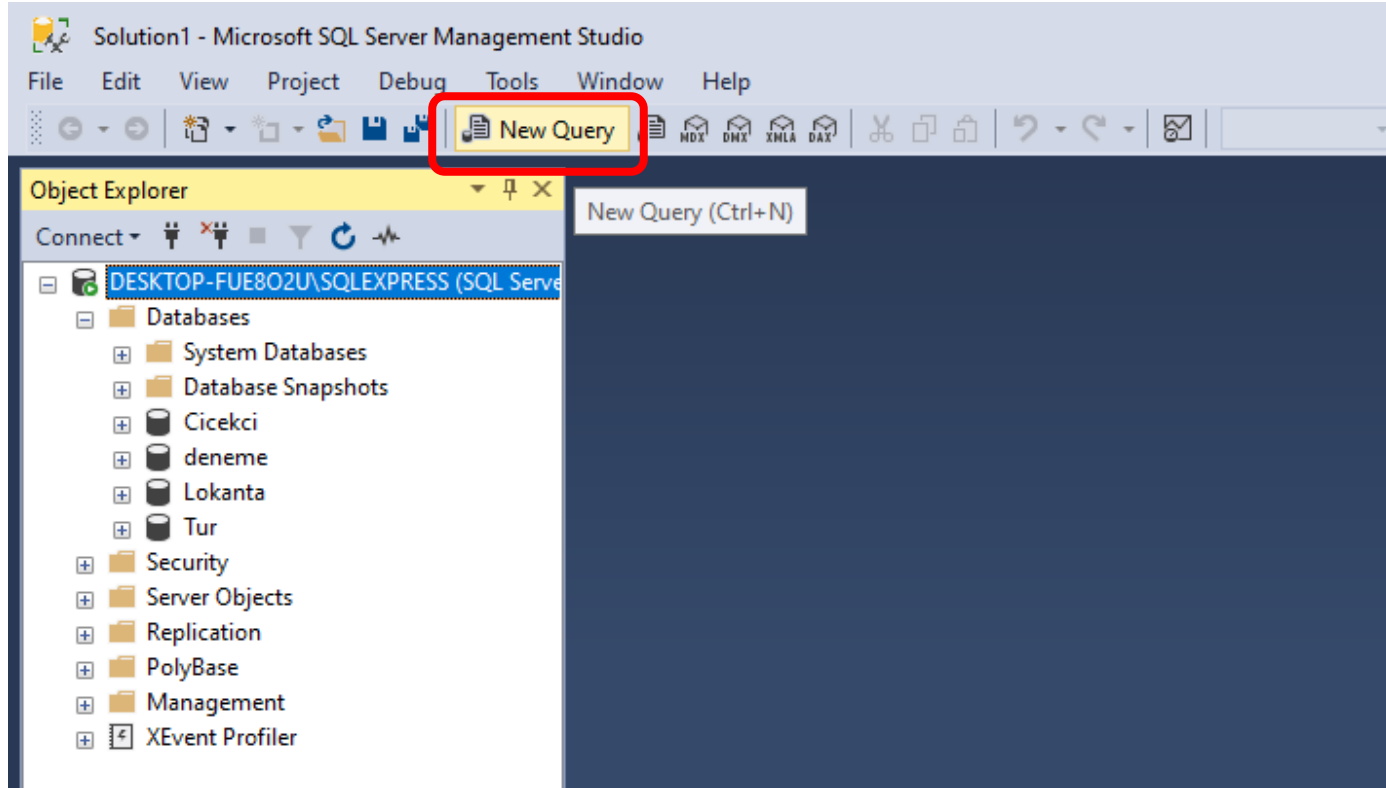
- *IDENTITY* sütunlar, genel olarak tablolardaki birincil anahtarlarda yeni sayısal değer oluşturmak için kullanılır.
- *IDENTITY* sütun ile *PRIMARY KEY* (birincil anahtar) kavramı, bir arada kullanıldığı için benzer gibi görünse de farklı kavramlardır.
- *IDENTITY* belirli bir sayıdan başlayarak belirli bir aralık ile artan ya da azalan sayısal değer üretir. *PRIMARY KEY* ise, bir sütunda bulunan değerlerin benzersiz olmasını sağlar.

T-SQL VE SORGU

- *SQL, sadece veri tabanı üzerinde işlem yapılmasını sağlayan bir alt dildir.*
- *SQL, veri tabanına veri ekleme, verilere erişime ve değiştirmede kullanılan yapısal bir sorgulama dilidir.*
- *SQL sorgulama dili ANSI (American National Standards Institute) standartları üzerine kurulmuştur. Bunun yanında kullanılan veri tabanı işleme programları kendilerine özgü bazı ek komutlar ve özellikler ekleyerek veri tabanı üzerinde daha etkin bir veir işleme özelliğine sahip olmuşlardır.*

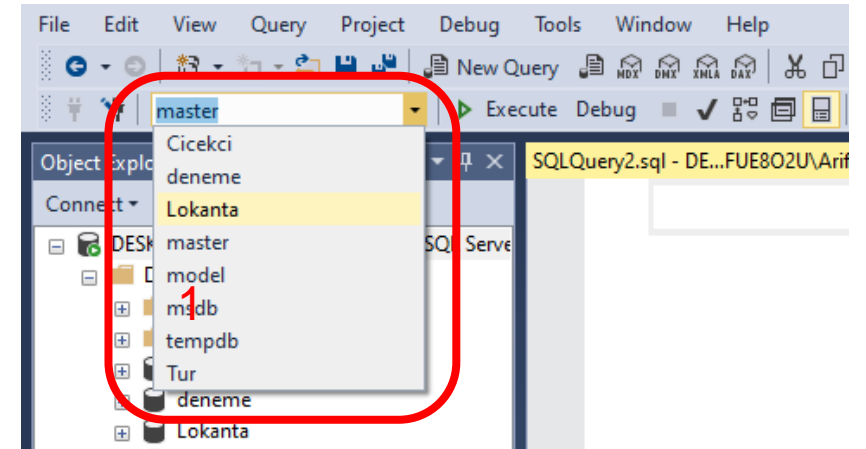
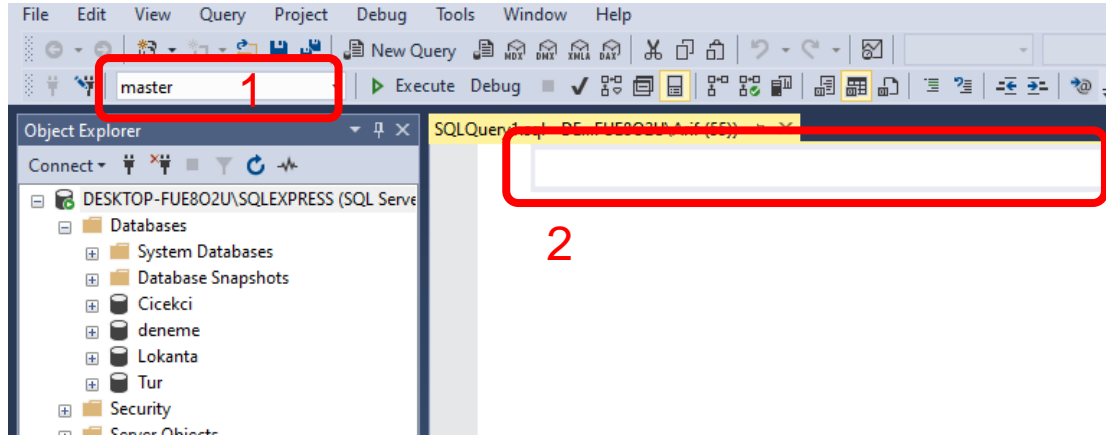
- *Gerek klasik pencere programlaması olsun gerekse web tabanlı programlar olsun, elde edilen bu verilerden isteğe uygun verileri çekmek için SQL (Strucruted Query Language - Yapısal Sorgulama Dili) adı verdikleri dili geliştirdiler.*
- *Yazılan uygulamalarda, veri tabanına kaydedilen verileri okumak, güncellemek veya yeni bir veri eklemek için SQL dili kullanılmaktadır.*

Sorgu Yazılması



Sorgu Yazılması

- Burada 1' nolu kısımda sorgu yazılacak veri tabanının seçilmesi gerekiyor. Default olarak master veri tabanı gelir.
- 2' nolu kısma ise sorgumuzu yazıyoruz.



master

SQLQuery2.sql - DE_FUE8O2U\Arif (53)**

```
Select* From Musteri
```

194 %

Msg 208, Level 16, State 1, Line 1
Invalid object name 'Musteri'.

Cicekci

SQLQuery2.sql - DE_FUE8O2U\Arif (53)**

```
Select* From Musteri
```

194 %

	No	Ad	Soyad	Kayıt Tarihi	TCKimlikNo
1	1	Arif	Günel	NULL	NULL

USE VE GO Komutu

- *SQL Server'da yığın kavramı, sorguların sırayla işleme alınmasını ifade eder.*
- *Çalışma sırasında SQL Server'a birden fazla gönderilen sorgu yığınlar halinde ele alınır.*
- *Bu yığınları birbirinden ayırt etmek için bazı komutlar kullanılır.*

USE Komutu

- *T-SQL kod bloklarının, hangi veri tabanı için çalıştırılacağını belirtmek için kullanılır.*
- *Kullanımı;*

USE VeriTabaniAdi

GO Komutu

- *Go komutu bir T-SQL ifadesi değildir, fakat TSQL kodlarında sıkça kullanılır.*
- *Normal sorgularda kullanılmaz.*
- *GO komutu öncesinde yazılmış olan tüm sql sorgularını belli işlemleri yapmak için SQL Servera gönderir.*
- *GO komutunu yazılan **scripti** servera göndermek için bir işaret olarak algılar.*
- *GO komutu bir yığının sonunu belirtmek için kullanılır.*
- *Bir yığın SQL Server'da işlenmeye başladığı anda önce ayrıştırılır (parse), sonra derlenir (compile) ve son olarak sorgu çalıştırılır (execute).*

Kullanımı;

Komutlar

GO

The screenshot displays the SQL Server Enterprise Manager interface. In the top toolbar, the 'master' database is selected. The query editor shows the following SQL code:

```
USE Cicekci
GO
Select* From Musteri
```

The Results pane displays the following data:

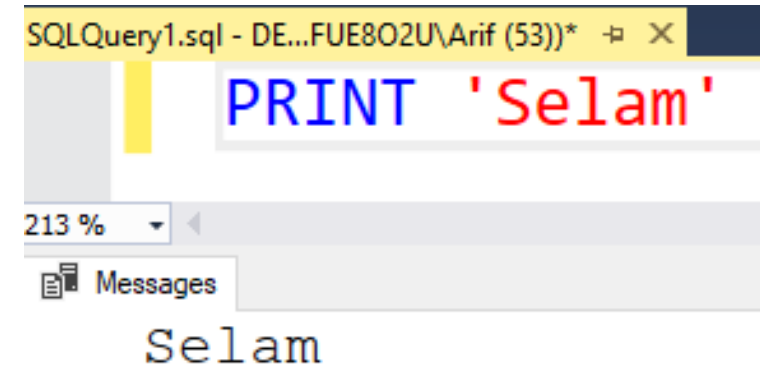
No	Ad	Soyad	Kayit Tarihi	TCKimlikNo
1	Arif	Günel	NULL	NULL

Dikkat edilmesi gereken noktalar:

- *GO komutu diğer T-SQL komutları ile aynı saatırda kullanılamaz*
- *GO komutundan önce tanımlanan bir değışken GO komutundan sonra kullanılamaz. Çünkü daha önceki kod yığınının sona erdiğini belirtmektedir.*
- *Kodun sonunda GO kullanılmadığı veya tüm sorgu bitmediğı sürece komutlar sunucuya gönderilmez.*

PRINT Komutu

- *T-SQL içerisinde kullanılan değişkenlerin değerleri ya da hatalar sonucu oluşacak hata mesajları gibi sonuçların sorgu çalıştığı anda gösterilmesini sağlar.*
- *Print ifadesinden sonra tırnak işareti ile direk metin yazdırabileceğimiz gibi var olan değişken üzerinden de yazdırabiliriz.*



Veri Tanımlama Dili

(DDL-Data Definition Language)

- *Verinin saklandığı ortam ile ilgili nesne oluşturma, silme ve nesneler üzerinde değişiklik yapmaya denir.*
- *Bir veri tabanı (database) veya tablo (table) oluşturmayı, silmeyi ve üzerinde değişiklik yapmayı sağlayan ifadelerdir.*
- *Üç temel komutu vardır: "**create**", "**alter**", "**drop**".*
- *Bunlar temel SQL sorgularıdır.*

CREATE

- *Veri tabanındaki nesneleri oluşturmak için kullanılır.*

ALTER

- *Nesneler üzerinde değişiklik yapmak için kullanılır.*

DROP

- *Nesneleri silmek için kullanılır.*

CREATE-OLUŞTUR

- *Veri tabanındaki nesnelerin oluşturulması için kullanıyoruz.*
- *Yeni bir veri tabanı, tablo vb nesneleri oluşturuyoruz.*
- *CREATE ifadesi genel bir oluşturucu ifadedir. CREATE ifadesinden sonra oluşturulacak nesnenin özelliklerine göre farklı parametreler alır.*

CREATE

ALTER-DEĞİŞTİR

- *İhtiyaç durumunda mevcut tablonun **değiştirilmesi** istenebilir.*
- *Bu durumda değiştirilecek kısım için tüm tabloyu silip create ile tekrar tablo yapmak yerine alter komutu kullanılarak değişiklik yapılabilir.*
- *Tabloya yeni alan ekleme, üzerinde değişiklik yapma(ismini), düzenleme gibi işlemlerde kullanılır.*

Tablolar üzerinde ALTER komutu ile yapılabilecek değişiklikler

1.Yeni bir sütun ekleme.

2.Sütun tanımı ve sütun değerinin değiştirilmesi.

3.Var olan bir sütunun silinmesi.

4.Tablo tanımının değiştirilmesi.

5.Tablo kısıtlarının değiştirilmesi, yeni kısıtların eklenmesi ya da var olan kısıtların düşürülmesi.

DROP-SİL

- *Var olan bir nesnenin bir kısmını silmek için.*
- *Mesela var olan bir tablonun bir alanını silmek için kullanıyoruz.*

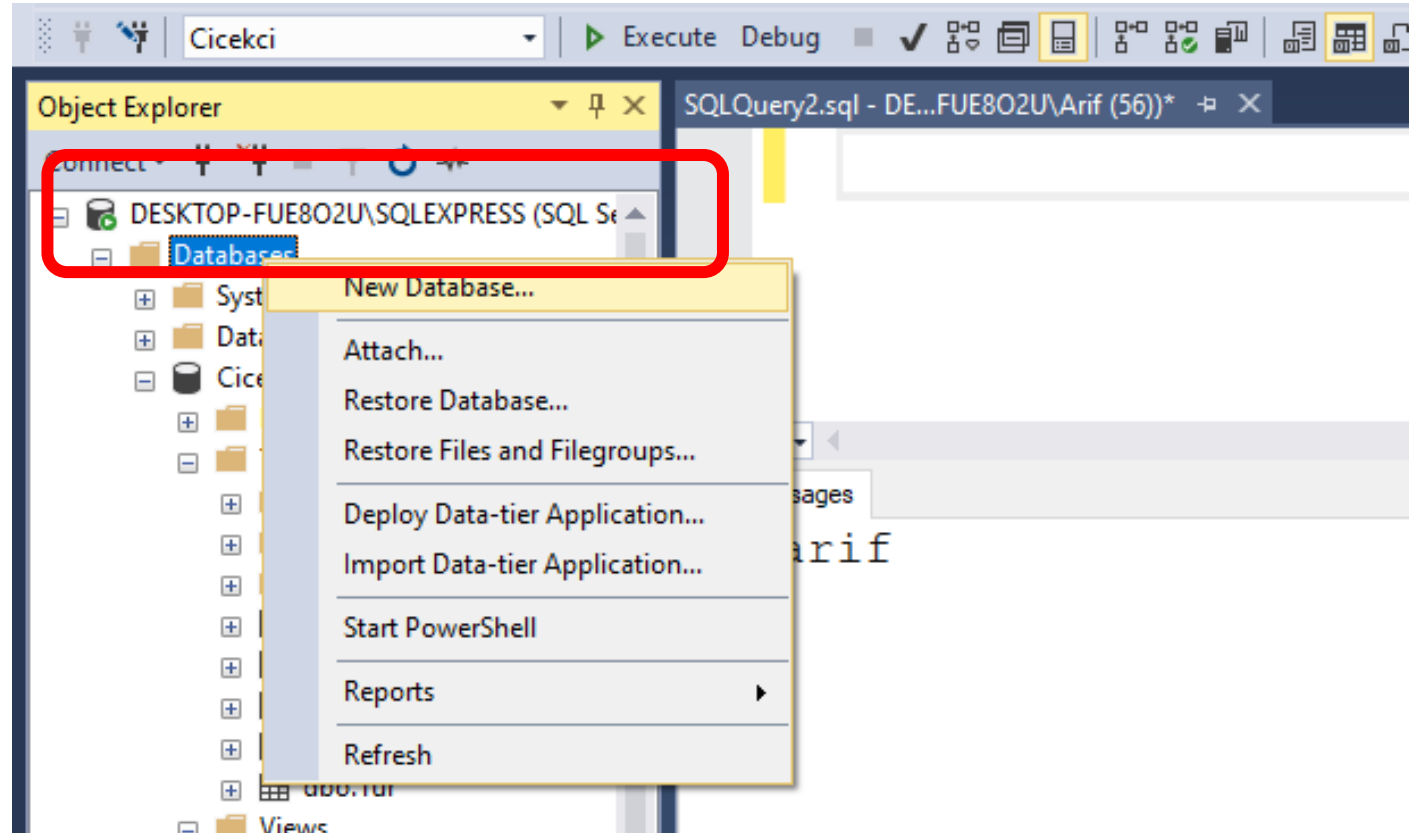
Yeni Veri Tabanı Oluşturma

Yeni veri tabanı oluşturma iki şekilde yapabiliriz.

- İlk olarak SQL Server Management Studio menüleri üzerinden yapabiliriz*
- Diğer CREATE SQL komutu ile*

SQL Server Management Studio menüleri üzerinden Yeni Bir Veri Tabanı Oluşturma

- Sol taraftaki **Object Explorer** kısmından database üzerinde sağ tıklayıp **New Database** diyoruz



- Burada veri tabanına isim veriyoruz
- İkinci kısımda görülenler veri tabanındaki dosyalar. Biri veri tabanın kendi dosyası diğeri de log dosyası

New Database

Select a page

- General
- Options
- Filegroups

Script Help

Database name: Deneme

Owner: <default>

☒ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
Deneme	ROWS...	PRIMARY	8	By 64 MB, Unlimited
Deneme_log	LOG	Not Applicable	8	By 64 MB, Unlimited

Connection

Server: DESKTOP-FUE802U\SQLEXPRESS

Connection: DESKTOP-FUE802U\Arif

[View connection properties](#)

Progress

Ready

Add Remove

OK Cancel

- Dosyaları boyutları hafızadaki alanları ve dosyanın yeri ile ilgili veriler mevcut

CREATE SQL komutu ile Veri Tabanı Oluşturma

CREATE

- İlk olarak veri tabanının ismi birincil olan data dosyasının özelliklerini oluşturuyoruz.

```
CREATE DATABASE Arif
ON PRIMARY
(
NAME=N'ARİF_Data',
FILENAME=N'C:\Program Files\Microsoft SQL
Server\MSSQL13.SQLEXPRESS\MSSQL\DATA\Arif_Data.mdf',
SIZE=20MB,
MAXSIZE=UNLIMITED,
FILEGROWTH=20MB
)
```

NAME : Veri dosyasının adı.

FILENAME : Veri dosyamızın işletim sistemindeki fiziksel adresi.

SIZE : Veri tabanı boyutunu.

MAXSIZE : Veri tabanı boyutunu limitsiz olarak belirtiyoruz. Limit, sabit diskin kapasite üst sınırı olacaktır.

FILEGROWTH : Veri tabanı genişlemesi gerektiğinde, her seferinde artım miktarı.

Sonraki kısımda Log dosyasını özelliklerini oluşturuyoruz

LOG ON

(

NAME=N'ARİF_Log' ,

FILENAME=N'C:\Program Files\Microsoft SQL
Server\MSSQL13.SQLEXPRESS\MSSQL\DATA\Arif_Log.ldf' ,

SIZE=20MB ,

MAXSIZE=20GB ,

FILEGROWTH=20MB

)

GO

ON

- *Veri tabanının veri ve log dosyasının yerini belirtmek için kullanılır. ON operatöründen sonra gelen PRIMARY anahtar kelimesini fark etmiş olmalısınız.*
- *Bunun anlamı, belirtilen dosya bilgilerinin fiziksel olarak birincil dosya grubuna ait olduğudur.*

NAME

- *Veri tabanının mantıksal ismini belirtir. SQL Server veri tabanı dosyasına ulaşmak için bu ismi kullanır. Veri tabanını ya da dosyayı yeniden boyutlandırmak gibi işlemler için bu isim kullanılır.*

FILENAME

- *Veri ve log dosyasının bulunduğu dosyaların işletim sistemi üzerindeki gerçek fiziksel dosya isimlerini belirtir.*
İsimlendirme veri ve log dosyası için farklı şekilde yapılır.
 - *Veri dosyası uzantısı: mdf*
 - *İkincil dosya uzantısı: ndf*
 - *Log dosyası uzantısı: ldf*

SIZE

- *Veri tabanının dosya boyutunu gösterir. Boyut varsayılan olarak MB (MegaByte) cinsinden belirtilir. Ancak boyutu KB (KiloByte), GB (GigaByte) ya da TB (TeraByte) cinsinden de belirtebilirsiniz. Dosya boyutu en az model veri tabanı boyutu kadar olması gerekir. Dosya boyutu belirtilirken tam sayı cinsinden, yani ondalık bölüm olmadan belirtilmesi gerekir.*

MAXSIZE

- *Veri tabanı dosya boyutunun genişleyebileceği en yüksek büyüklüğü belirtir.*
- *MAXSIZE parametresi de SIZE gibi varsayılan olarak MB cinsinden değer alır.*
- *Fakat dilerseniz KB, GB ya da TB cinsinden dosya boyutu belirtebilirsiniz. Bu parametre değerinin belirtilmesi zorunlu değildir. Eğer bir değer belirtilmezse varsayılan olarak bir MAXSIZE değeri atanmayacak ve bu durumda MAXSIZE değeri sabit disk (HDD)'in kapasitesiyle sınırlı olacaktır.*

- *MAXSIZE, üzerinde sürekli anlık işlem yapıldığı durumlarda çok kritik ve takip edilmesi gereken bir özelliktir. Eğer dosya boyutları (mdf, ndf), MAXSIZE'ın maksimum değerine ulaşırsa, kullanıcılar veri girişi yapmak istediğinde hata alırlar. Log tutma işleminde de maksimum değere ulaşıldığında log tutulamayacaktır. Bir veri tabanı veri dosyası (mdf), sadece veri giriş işlemlerinde bir girdi olduğu için hızlı büyümmez. Ancak log dosyaları her işlem için log dosyasına (ldf) loglama işlemi gerçekleştirdiği için sürekli büyüyecektir.*

FILEGROWTH

- *Veri tabanı dosyasının bir seferde ne kadar genişleyeceğini(büyüyeceğini) belirtmek için kullanılır.*
- *Veri tabanının başlangıç genişlik değeri SIZE ile olabilecek en yüksek değeri MAXSIZE arasında geçerlidir. Bu değeri KB, MB, GB ya da TB olarak belirleyebileceğimiz gibi yüzdesel olarak da belirleyebilirsiniz.*
- *Başlangıç değeri 100 MB olan bir veri tabanının her seferinde %10 büyümesini isterseniz, 10 MB büyüyerek 110 MB olacaktır.*

```
CREATE DATABASE Arif
ON PRIMARY
(
NAME=N'ARİF_Data',
FILENAME=N'C:\Program Files\Microsoft SQL
Server\MSSQL13.SQLEXPRESS\MSSQL\DATA\Arif_Data.mdf',
SIZE=20MB,
MAXSIZE=UNLIMITED,
FILEGROWTH=20MB
)
LOG ON
(
NAME=N'ARİF_Log',
FILENAME=N'C:\Program Files\Microsoft SQL
Server\MSSQL13.SQLEXPRESS\MSSQL\DATA\Arif_Log.ldf',
SIZE=20MB,
MAXSIZE=20GB,
FILEGROWTH=20MB
)
GO
```

NAME : Veri dosyasının adı.

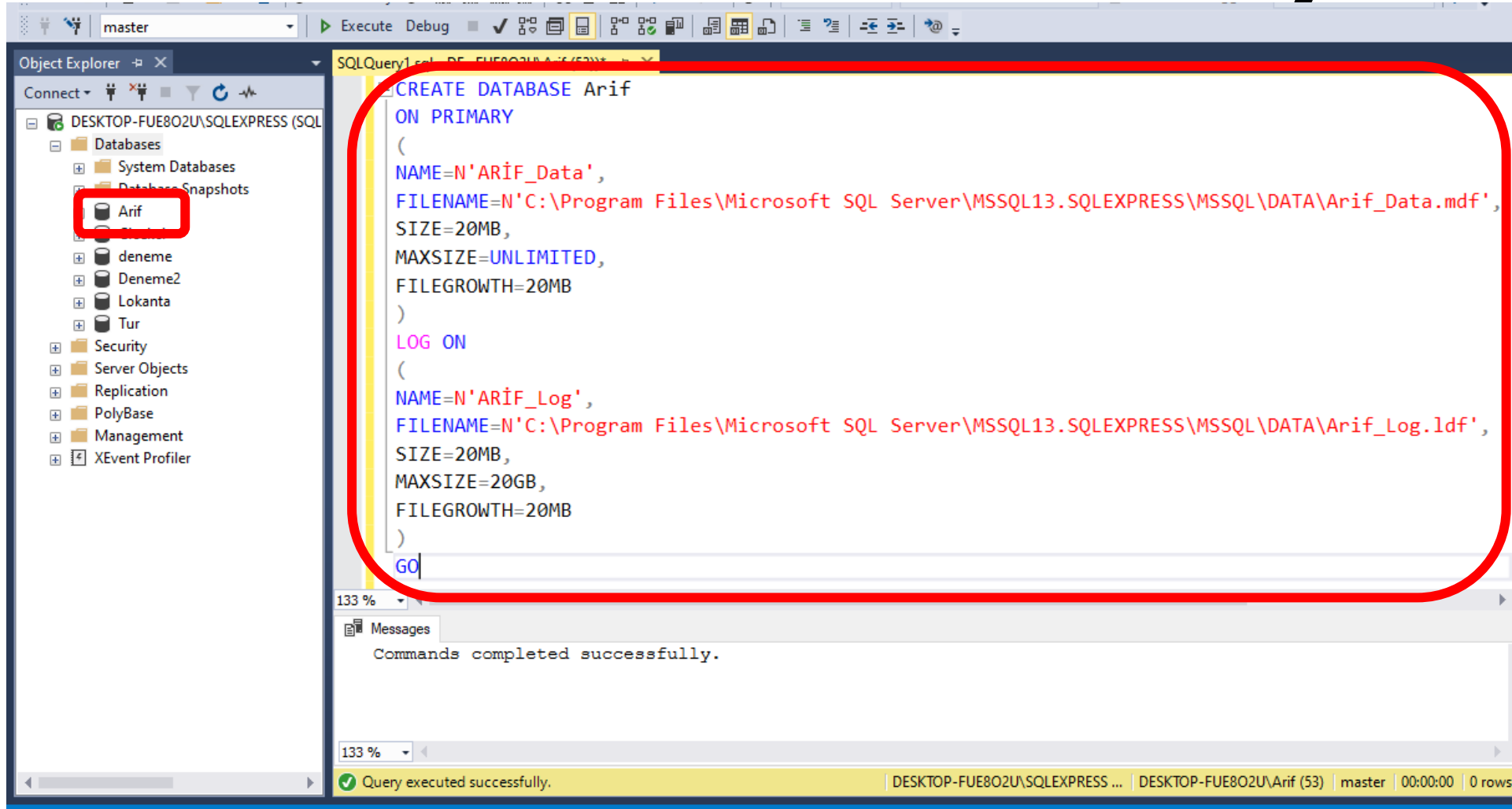
FILENAME : Veri dosyamızın işletim sistemindeki fiziksel adresi.

SIZE : Veri tabanı boyutunu.

MAXSIZE : Veri tabanı boyutunu limitsiz olarak belirtiyoruz. Limit, sabit diskin kapasite üst sınırı olacaktır.

FILEGROWTH : Veri tabanı genişlemesi gerektiğinde, her seferinde artım miktarı.

CREATE SQL komutu ile Veri Tabanı Oluşturma



Bilgisayardan dosya konumlarına baktığımızda Arif isimli data ve log dosyalarını görüyoruz

C:\Program Files\Microsoft SQL Server\MSSQL13.SQLEXPRESS\MSSQL\DATA				
Ad	Değişirme tarihi	Tür	Boyut	
Arif_Data	12.09.2020 15:03	SQL Server Databa...	20.480 KB	
Arif_Log	12.09.2020 15:03	SQL Server Databa...	20.480 KB	
Cicekci	10.09.2020 19:48	SQL Server Databa...	8.192 KB	
Cicekci_log	12.09.2020 15:03	SQL Server Databa...	8.192 KB	
deneme	10.09.2020 19:48	SQL Server Databa...	8.192 KB	
deneme log	12.09.2020 15:03	SQL Server Databa...	8.192 KB	

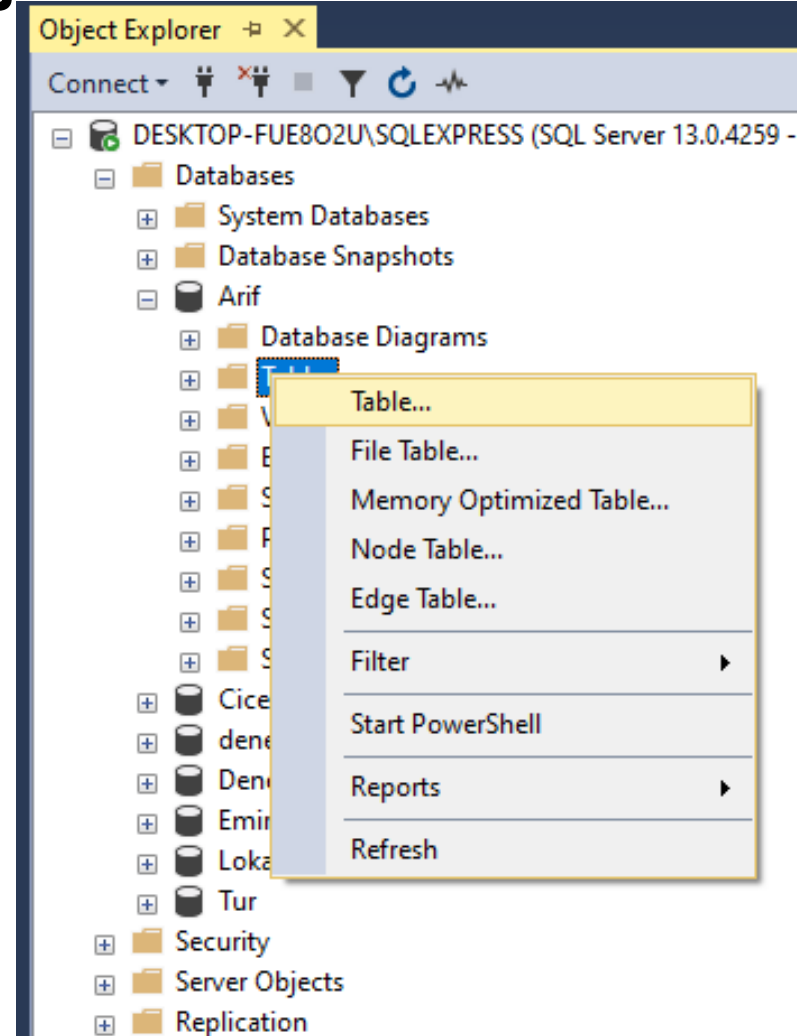
Tablo Oluşturma

Tablo oluşturmaya iki şekilde yapabiliriz

- 1. Menüler ile yeni tablo oluşturma*
- 2. T-SQL sorguları kullanarak tablo oluşturma*

Menüler ile yeni tablo oluşturma

- *Hangi veri tabanında tablo oluşturulacaksa onun altındaki «Table» üzerinde sağ tıklıyoruz*



- 3 adet alan oluşturuyoruz.
- No alanı boş geçilemez özelliğinde ad ve soyad boş geçilebilir

DESKTOP-FUE8O2U\S...Arif - dbo.Table_1* ✖ Object Explorer			
	Column Name	Data Type	Allow Nulls
	No	int	<input type="checkbox"/>
	Ad	varchar(50)	<input checked="" type="checkbox"/>
	Soyad	varchar(50)	<input checked="" type="checkbox"/>

- *Ctrl + S* ile isim verip kaydediyoruz

DESKTOP-FUE8O2U\S...Arif - dbo.Table_1* - Object Explorer			
	Column Name	Data Type	Allow Nulls
	No	int	<input type="checkbox"/>
	Ad	varchar(50)	<input checked="" type="checkbox"/>
	Soyad	varchar(50)	<input checked="" type="checkbox"/>
▶			<input type="checkbox"/>

Choose Name

Enter a name for the table:

Personel

OK Cancel

Object Explorer - DESKTOP-FUE8O2U\S...rif - dbo.Personel			
Connect [Icons]			
DESKTOP-FUE8O2U\SQLEXPRESS (S)			
Databases			
System Databases			
Database Snapshots			
Arif			
Database Diagrams			
Tables			
System Tables			
FileTables			
External Tables			
dbo.Personel			

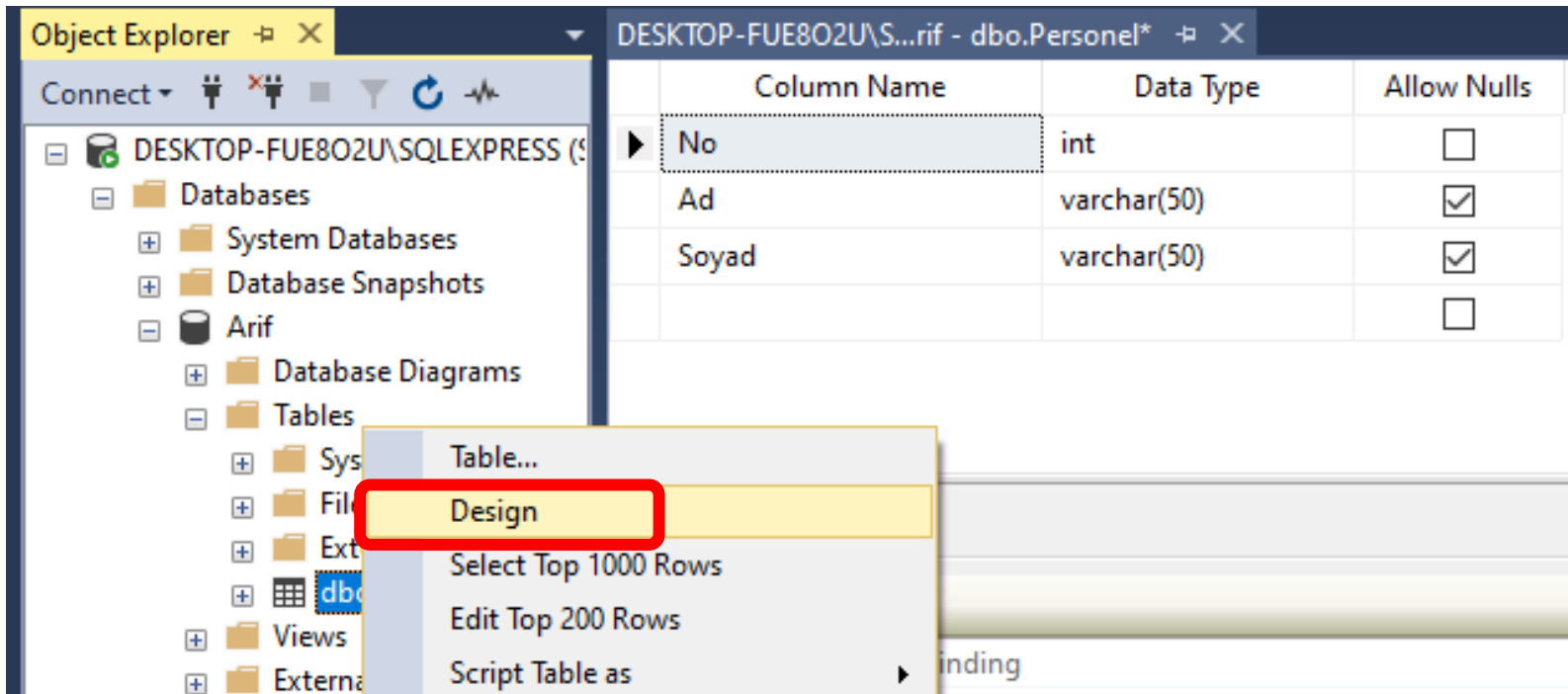
DESKTOP-FUE8O2U\S...rif - dbo.Personel			
	Column Name	Data Type	Allow Nulls
	No	int	<input type="checkbox"/>
	Ad	varchar(50)	<input checked="" type="checkbox"/>
	Soyad	varchar(50)	<input checked="" type="checkbox"/>
▶			<input type="checkbox"/>



ALTER

- *Menüler ile deęişiklik yapma*

Tabloda değişiklik yapma



No alanını birincil anahtar ve otomatik artan yapalım

DESKTOP-FUE802U\S...rif - dbo.Personel* -> X

Column Name	Data Type	Allow Nulls
No	int	<input type="checkbox"/>
Ad	varchar(50)	<input checked="" type="checkbox"/>
Soyad	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

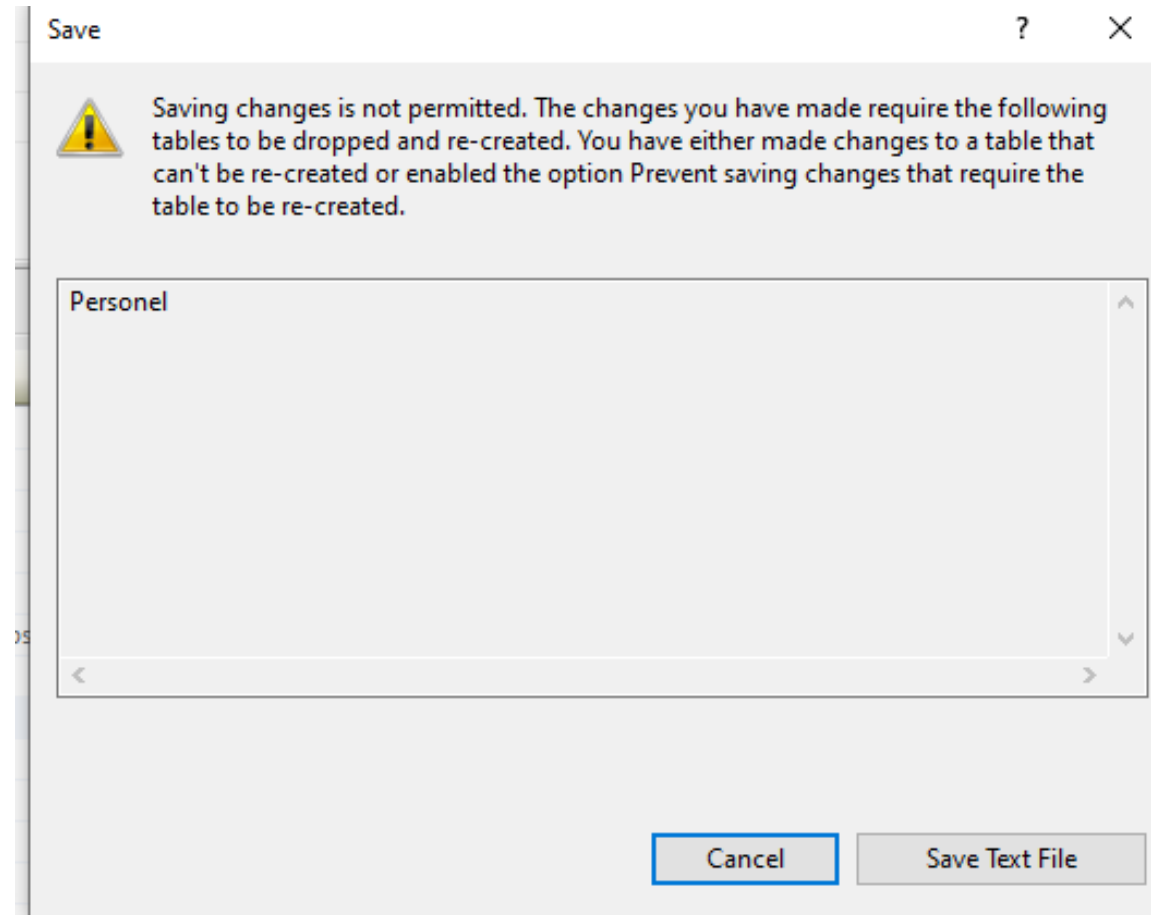
Column Properties

Default Value or Binding	
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No
Not For Replication	No

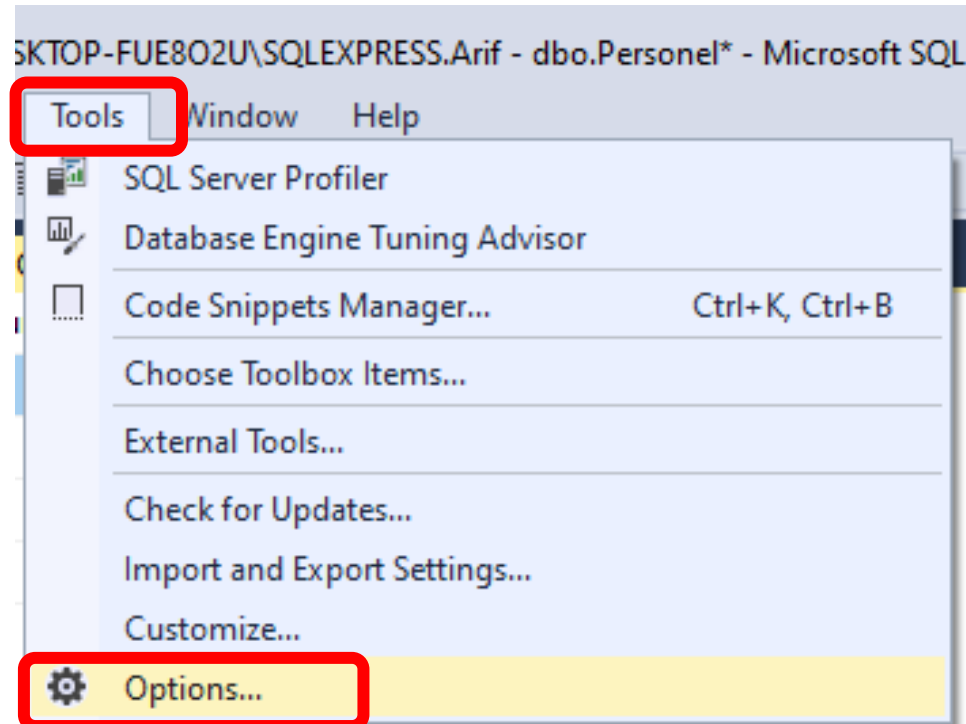
(Is Identity)

Column Name	Data Type
Set Primary Key	
Insert Column	
Delete Column	
Relationships...	
Indexes/Keys...	
Fulltext Index...	
XML Indexes...	
Check Constraints...	
Spatial Indexes...	
Generate Change Script...	
Properties	

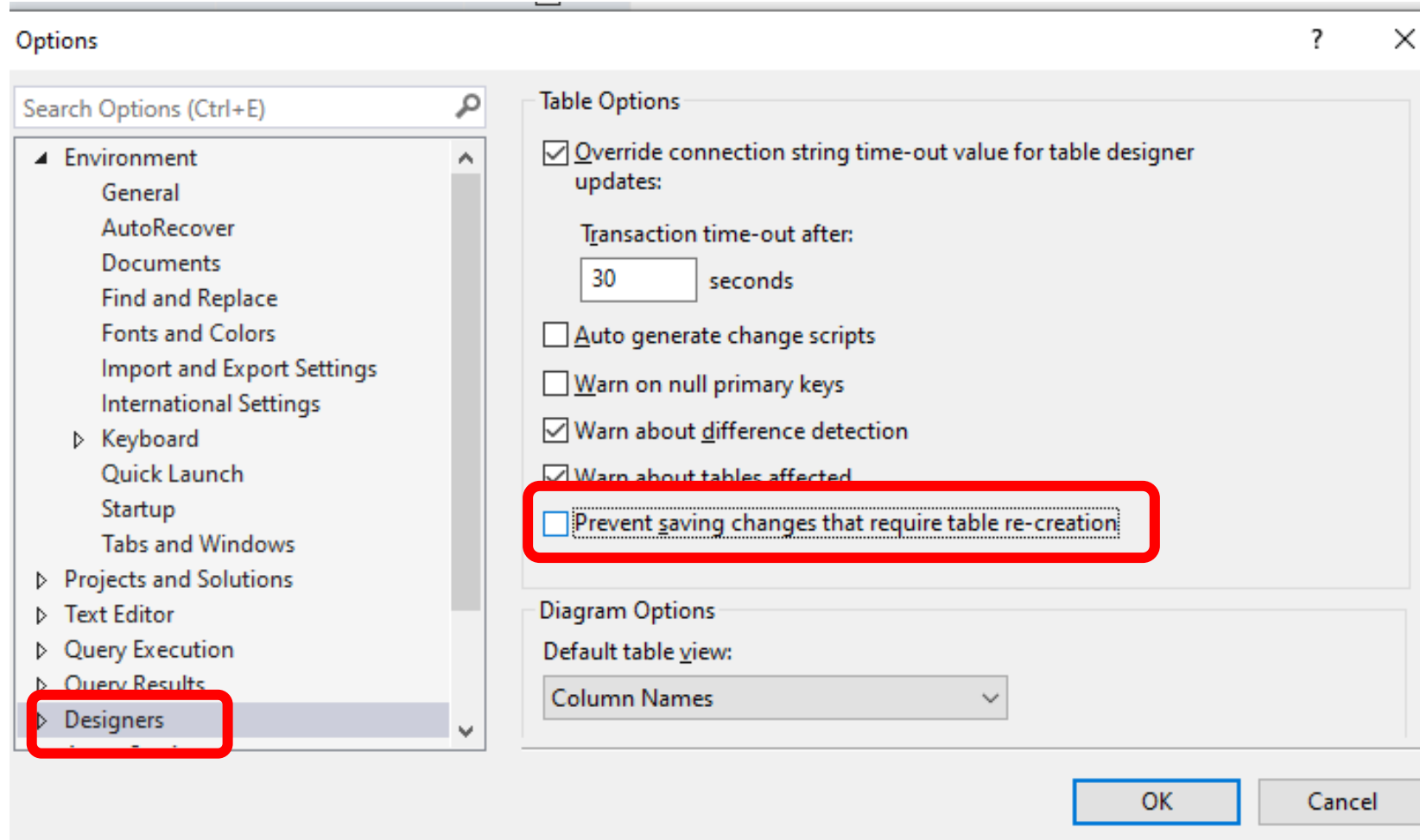
Tabloda deęişiklik yapma hatası alıyoruz



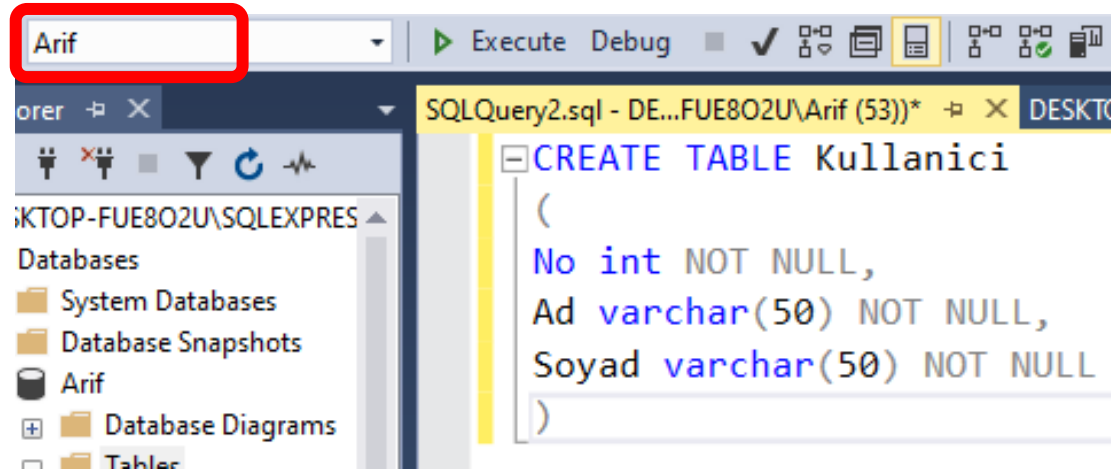
Tools -> Options



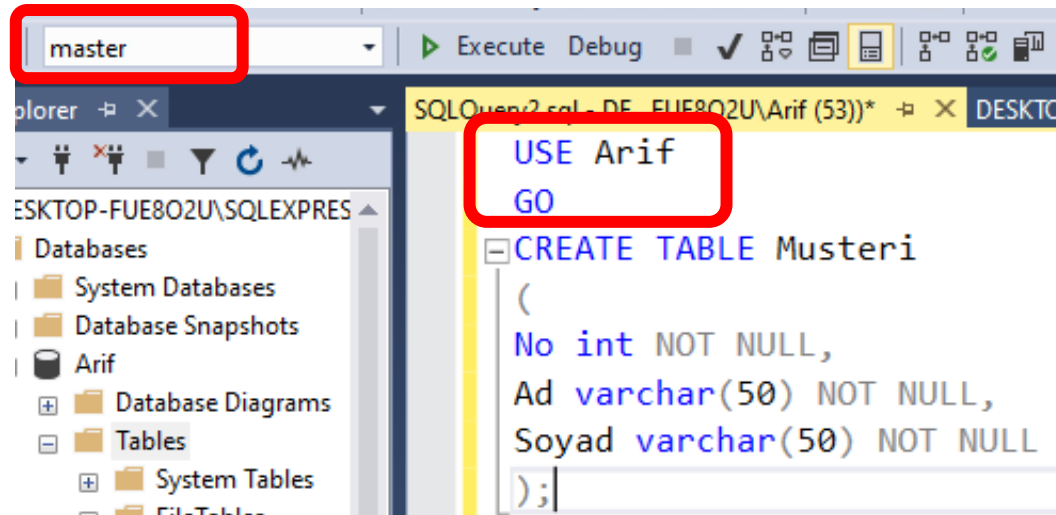
Yeniden oluşturmaya izin verelim ve tekrar kaydedelim.



Tablo oluşturmayı T-SQL ile Yapalım

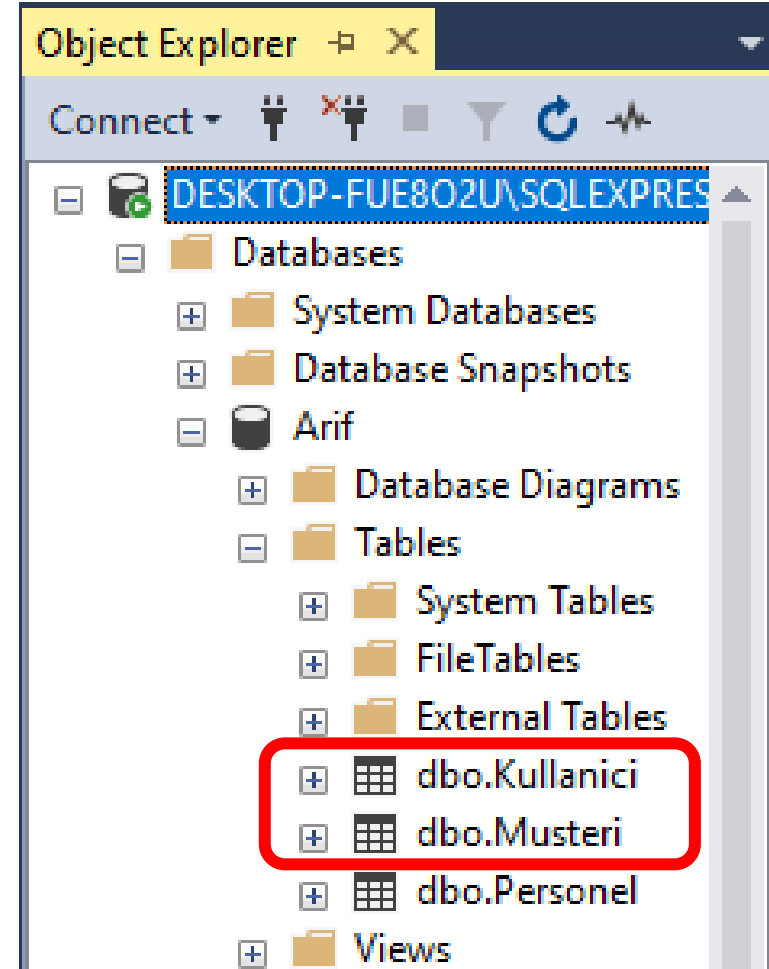


```
CREATE TABLE Kullanici
(
    No int NOT NULL,
    Ad varchar(50) NOT NULL,
    Soyad varchar(50) NOT NULL
)
```



```
GO
CREATE TABLE Musteri
(
    No int NOT NULL,
    Ad varchar(50) NOT NULL,
    Soyad varchar(50) NOT NULL
);
```

- *Kullanici ve Musteri*
isimli tabloların
oluştüğunu görüyoruz

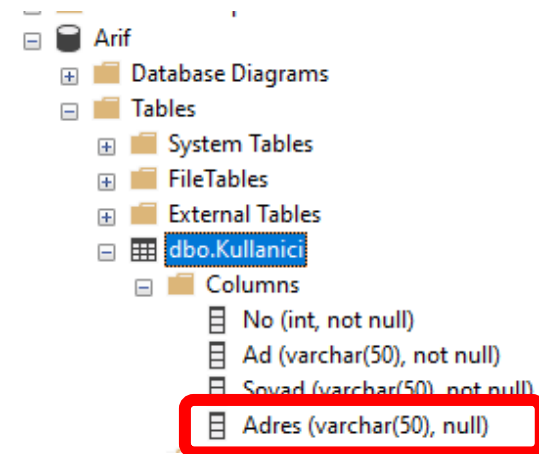
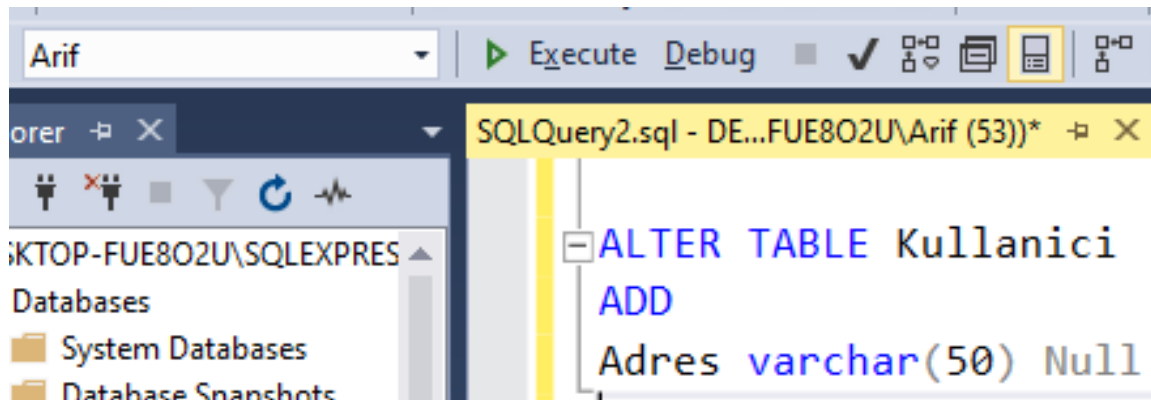


- *Sorgu ile deęişiklik yapma.*

ALTER

ALTER ile Kullanici isimli tabloya alan ekleyelim

```
ALTER TABLE Kullanici  
ADD  
Adres varchar(50) Null
```



	No	Ad	Soyad	Adres
	NULL	NULL	NULL	NULL

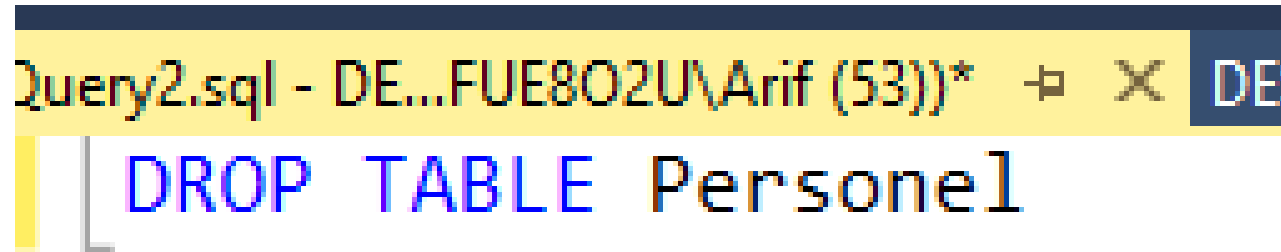
DROP

DROP ile Veri tabanını Silmek

- *DROP kelimesini kullanıp veri tabanı silmek hem basit hem de çok riskli bir işlemdir.*
- DROP DATABASE Deneme
- *Silme sırasında veri tabanı sunucu üzerinde kullanılması durumunda hata alınır,*

Drop ile personel tablosunu silelim

- DROP TABLE Personel

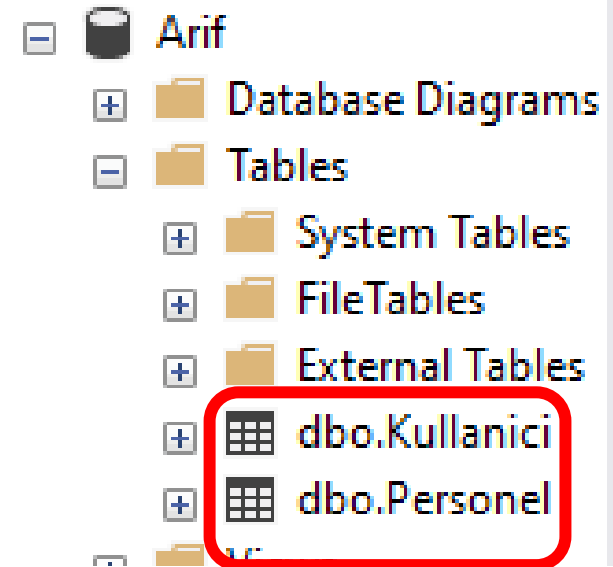
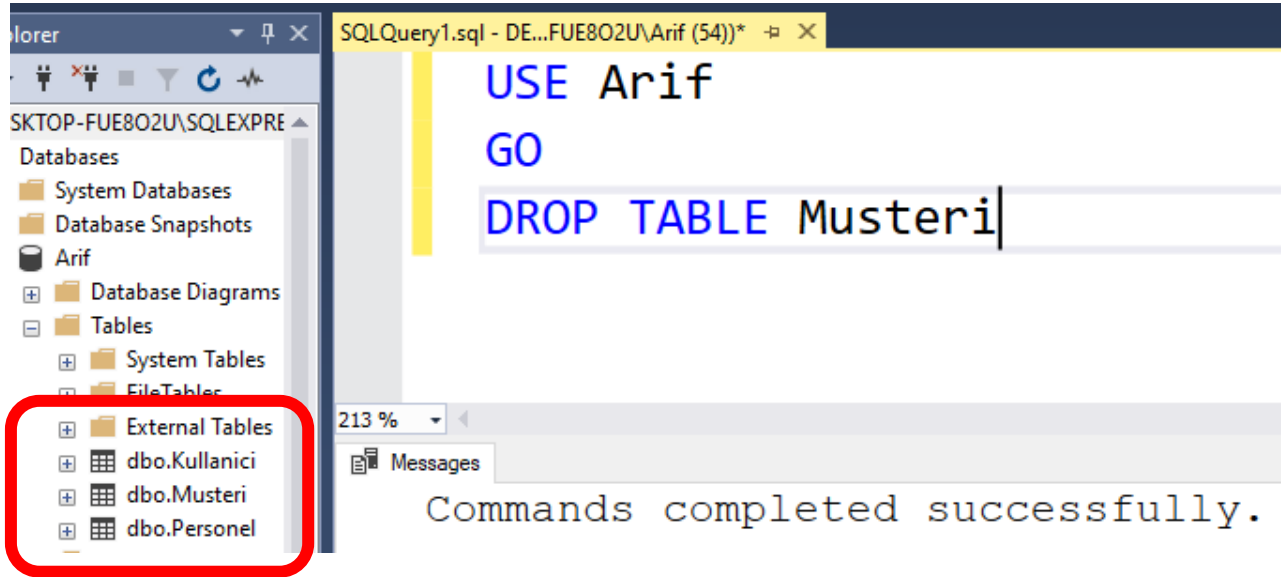


Query2.sql - DE...FUE8O2U\Arif (53))* X DE
| DROP TABLE Personel

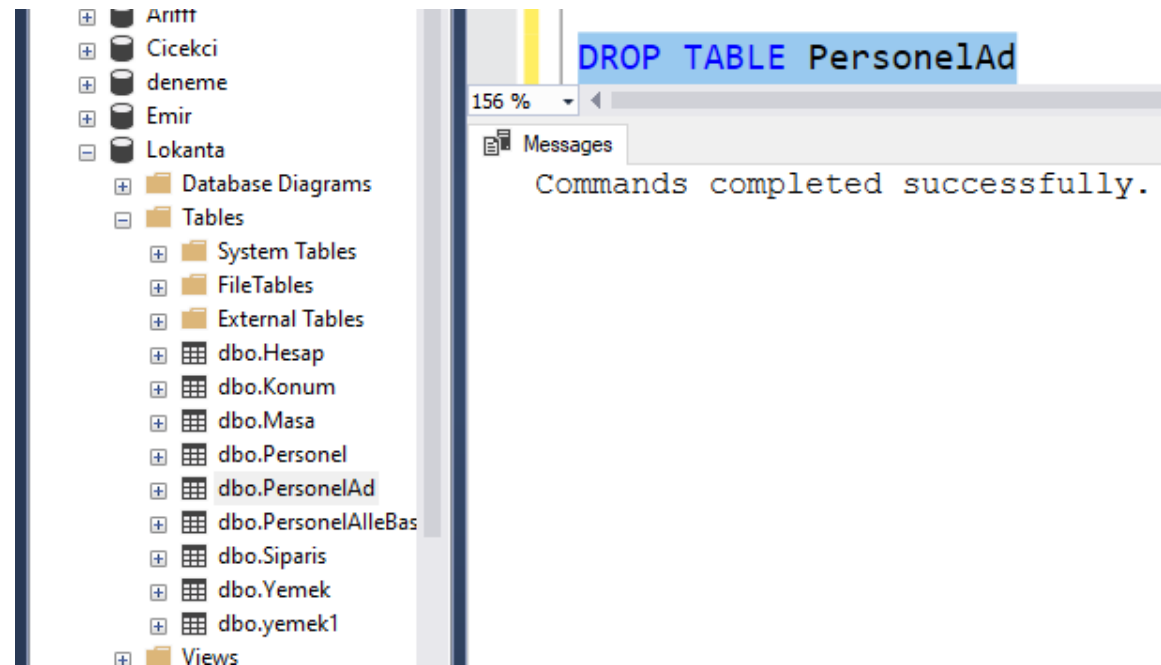
DROP

DROP TABLE Personel

- Bu şekilde bir kullanım yanlış veri tabanını da silebileceği için USE ve Go komutları ile hangi veri tabanının altındaki tablo olduğunu göstermek veri kayıplarının önüne geçilmesini sağlayacaktır.*



- DROP TABLE PersonelAd



Kaynaklar

- <http://www.yazilimutfagi.com/index.php/2012/04/20/go-komutunun-kullanimi-go-komutunu-kullanarak-bir-sorgu-birden-fazla-nasil-calistirilir/>
- <https://medium.com/gokhanyavas/t-sql-veri-tan%C4%B1mlama-dili-ddl-8fef1c39ce38>
- *İleri Seviye T-Sql Programlama - Cihan Özhan Kodlab Yayınları*