Merhaba Arkadaşlar,  
**Mobilhanem.com** sitesi üzerinden SQL Dersleri Serisini birlikte işleyeceğiz. Bu derste **SQL DDL (Data Definition Language) komutlarını** işleyeceğiz ve en temel komutlardan olan SQL DDL komutlarını örnekler ile sizlere anlatmaya çalışacağız. O zaman haydi başlayalım.

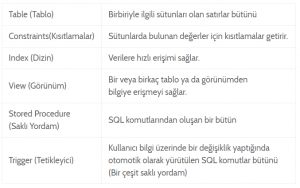
**SQL DDL(Data Definition Language) Nedir?**

Türkçe karşılığı Veri Tanımla Dili olan DDL veritabanın da ki verilerin tipiyle ilgilenir. Yani veri tabanında ki tabloların, tabloların birbiri ile olan ilişkileri tabloda ki verilerinin türleri ne olacak bunların hepsini DDL ile yapıyoruz. DDL temel olarak üç komuttan oluşur.

**CREATE : Nesne oluşturmak için kullanılır(Table/Database vb.)  
ALTER : Nesneler üzerinde değişiklik yapmak için kullanılır.  
DROP : Nesneleri silmek için kullanılır**

**CREATE KOMUTU**

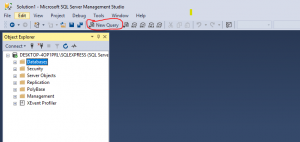
Veri tabanı ve veritabanı neslerini oluşturmak için kullanılır. Veri tabanı nesneleri nelerdir?



Biz bu yazıda veritabanı oluşturma ve veritabanı nesnesi olan tabloları oluşturacağız  
sonraki derslerde diğer nesneleri işleyeceğiz.

**Create Database( Veritabanı oluşturma)**

Bunun için ilk olarak SQL Server arayüzünü görelim;



Yeni sayfa için yukarıda işaretlediğim yeri tıklayarak açabilirsiniz.  
Genel olarak kullanım şekli şu şekildedir.

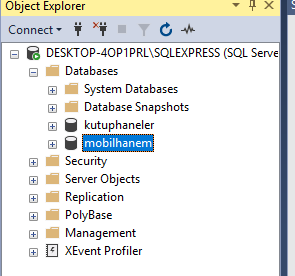
**CREATE DATABASE veritabanı\_ismi**

Ve bu kullanım bütün veritabanı yönetim sistemleri için aynıdır.

**Transact-SQL**

|  |  |
| --- | --- |
|  | **CREATE DATABASE mobilhanem** |

Hemen bakalım veritabanı oluşmuş mu?



Gördüğünüz gibi  mobilhanem varitbanını oluşturduk.  
Şimdilik veritabanı parametrelerini göstermeyeceğiz.

**Create Table(Tablo oluşturma)**

Tablo oluşturmadan önce tabloyu hangi veritabanı üzerinde oluşturacağımızı belirtmemiz lazım. Bunun için;  
Genel kullanım şekli şu şekildedir.

**use tablo\_ismi**

Hemen Yazalım

|  |  |
| --- | --- |
|  | **use mobilhanem** |

Tamam hangi veritabanı üzerinde oluşturacağımızı belirttik şimdi tablomuzu oluşturalım.  
Bunun için genel kullanım şu şekildedir.

**CREATE TABLE tablo\_ismi(sütün\_ismi1, veri\_tipi(boyut),sütün\_ismi2,veri\_tipi(boyut),sütün\_ismi2, veri\_tipi(boyut))**

Şeklinde kullanılır.  
O zaman örnek olarak mobilhanem.com yazarları tablosunu oluşturalım.

|  |  |
| --- | --- |
|  | **CREATE TABLE yazarlar(**  **yazar\_id int,**  **yazar\_isim varchar(15),**  **yazar\_soyad varchar(20),**  **yazar\_ders varchar(30));** |

Hemen anlatalım ne yaptık, ilk önce yazarlar isminde bir tablo oluşturduk, sonra bu veritabanı içine ilk olarak veri tipi integer olan bir bir yazar\_id isminde sütun oluşturduk. Sonrada diğer varchar() veri tipinde ki sütunlarımızı ekledik buradaki varchar(15) max 15 karakter alabileceğini belirtiyor siz istediğiniz veri uzunlukta alabilirsiniz.  
Temel basit şekilde tablo oluşturma bu şekildedir. Ama bizim bu tabloyu bazı özellikler eklememiz lazım, mesela yazar\_id her yazarda farklı olmalı ki ben onları ayırt edebileyim.

Kişinin tc numarası gibi, çünkü biz insanları ad veya soyad özelliklerine göre ayıramayız, kişilerin isimleri ve soyadları bazen aynı olabiliyor. Peki nasıl yapacağım diye soracaksınız hemen göstereyim.

**Primary Key İfadesi**

Primary key bir tabloda bir alanı, yani bir sütunu eşsiz bir değer yapmaya yarar. Bu ne demek oluyor eğer tablonun içinde yazar\_id = 15 olan bir yazar varsa siz ikinci bir yazar\_id numarası 15 olan bir yazar ekleyemezsiniz demek oluyor. Örneğin tc kimlik numaraları primary değerdir. Asla iki kişinin aynı tc kimlik numarası olamaz. Dolayısıyla tc kimlik numarası tutacak bir tablo muz varsa tc kimlik numarası sütunu primary olmalıdır.  
Hemen kullanalım;

|  |  |
| --- | --- |
|  | **CREATE TABLE yazarlar(**  **yazar\_id int primary key,**  **yazar\_isim varchar(15),**  **yazar\_soyad varchar(20),**  **yazar\_ders varchar(30));** |

Değişken tipinin yanına primary key yazarak bu sütünun primary\_key yani unique (benzersiz) bir değer olduğu bilgisini verdik.

NOT:

Bu şekilde bir hata mı aldınız

|  |  |
| --- | --- |
|  | Msg 2714, Level 16, State 6, Line 1  There is already an object named 'yazarlar' in the database. |

Hiç telaş yapmayın biz daha önce yazarlar isminde bir tablo oluşturduğumuz için hata verdi. Çünkü tablo isimleri tekildir**primary** gibi düşünün. Bir veritabanı içinde aynı isimde birden fazla tablo olamaz. Onun için ilk tablomuzu silelim ve tekrar çalıştıralım.

Şimdi sizin burada bana bir soru sormanız lazım; eğer yazar\_id numarası girilmez ise o zaman nasıl ayırt edeceğiz. Yani diğer alanları doldurup yazar\_id sütununu boş bırakırsa.  Bu durumda ayırt edemezsiniz bunun için yazar\_id kısmını zorunlu kılmalıyız. Çoğu zaman bir siteye veya bir yere kayıt olurken kullanıcı adı veya email adresi boş geçilmez şeklinde bir uyarı almışsınızdır. İşte bizde burada kullanıcıyı uyaracağız yazar\_id boş geçilemez diye. Nasıl mı **not null** anahtar kelimesi ile. Hemen kullanalım;

|  |  |
| --- | --- |
|  | **CREATE TABLE yazarlar(**  **yazar\_id int not null primary key,**  **yazar\_isim varchar(15),**  **yazar\_soyad varchar(20),**  **yazar\_ders varchar(30));** |

Bu şekilde halletmiş olduk. Artık yazar\_id sütunu hem primary hemde not null yani boş bırakılamaz.  Peki eksik bir şey var mı? Yok bitmedi daha şimdi biz yazar\_id kısmını aldık almasına ama bir şey kontrol etmeyi unuttuk. Günümüzde birçok zaman karşılaşmıştırsınız 11 karakterli tc numarasını veya 10 karakterli telefon numarasını girin diye veya siz sınav puanı gireceksiniz ama sınav puanı 0 -100 arasında bir değer girmelisiniz gibi. Evet bizde burada bunu kontrol etmeliyiz. bunun için;

**Check İfadesi**

|  |  |
| --- | --- |
| 1  2  3  4  5 | CREATE TABLE yazarlar(      yazar\_id int not null primary key check (yazar\_id <100),      yazar\_isim varchar(15),      yazar\_soyad varchar(20),      yazar\_ders varchar(30)); |

Evet burada yazar\_id numaramızın max 100 olabileceğini belirttik.  
Peki sadece 100 den büyük olduğu durumlarda mı hata vermeli? Bir kullanıcı id numarası negatif olamaz o zaman birden fazla durumu kontrol etmeliyiz bunun için hemen aklınıza şu şekilde bir kullanım gelmiştir.

|  |  |
| --- | --- |
|  | **CREATE TABLE yazarlar(**  **yazar\_id int not null primary key check (yazar\_id <100 and yazar\_id >0),**  **yazar\_isim varchar(15),**  **yazar\_soyad varchar(20),**  **yazar\_ders varchar(30));** |

tabi ki bu doğrudur. Yukarıda yazar\_id’nin primary olduğunu boş olamayacağını ve 100 den küçük 0 dan büyük olacağını belirttik.

Eğer birden fazla sütunun ayrı koşulları varsa ? Örneğin 2 sütunun da primary olması gerekliyse ? 3 sütunun not null olması gerekirse? tek tek mi yazmalıyız?

**Constarint İfadesi**

Constraint ifadesi birden fazla  sütünü kontrol edeceğimiz zaman kullanacağımız ifadedir. Hemen bir örnekle gösterelim.

|  |  |
| --- | --- |
|  | **CREATE TABLE yazarlar(**  **yazar\_id int not null,**  **yazar\_isim varchar(15),**  **yazar\_soyad varchar(20),**  **yazar\_ders varchar(30),**  **ders\_kodu int**  **constraint p\_key primary key(yazar\_id,ders\_kodu));** |

Burada aynı anda ders\_kodu ve yazar\_id sütunlarının primary key olduğunu belirttik. Sizden isteğim sizde aynı şekilde contsraint kelimesini check ifadesi ilk birlikte kullanmanız. Dilerseniz dersin altında yorum kısmında benimle paylaşırsanız birlikte bakarız 🙂  
Son olarak başka bir ifade daha gösterelim.

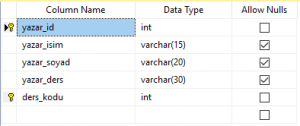
**Identity İfadesi**

Bazı durumlarda kendimiz değilde id numarsı vb. sütünları otomatik olarak eklemek isteyeceğiz. Bunun için identity ifadesini kullanacağız. Dilerseniz tabloya ekleme işlemini **IDENTITY(1,1)** ifadesi ile de tabloya otomatik olarak ekleneceğini belirleyebilirsiniz. Bu kullanış 1’den başla birer aralıklarla devam et anlamında kullanılmıştır.  
Hemen göstereyim;

|  |  |
| --- | --- |
|  | **CREATE TABLE yazarlar(**  **yazar\_id int not null IDENTITY(1,1),**  **yazar\_isim varchar(15),**  **yazar\_soyad varchar(20),**  **yazar\_ders varchar(30),**  **ders\_kodu int**  **constraint p\_key primary key(yazar\_id,ders\_kodu));** |

**ALTER KOMUTU**

Türkçe karşılığı değiştirmek ve güncellemek olan bu komut tablolar üzerinde güncelleme yapmak için kullanılır. Önce tablomuzu görelim.



Alter komutu üç farklı parametre alıyor:

* ADD
* DROP
* ALTER

O zaman öğrenmeye başlayalım:)

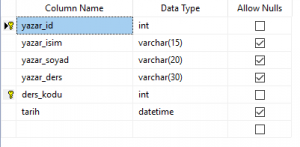
**ADD :** Bu parametre tabloya sütün ekleme işlemi yapmamızı sağlıyor.

**ALTER TABLE *tablo\_adı* ADD *sütün\_adı   sütün\_özelliği***

Şimdi yeni bir sütün ekleyelim, örneğin  tarihi sütünü ekleyelim;

|  |  |
| --- | --- |
|  | ALTER TABLE yazarlar ADD tarih datetime |

Şimdi tablomuzun son haline bakalım;



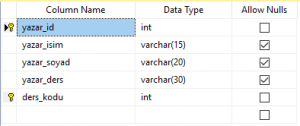
Gördüğünüz gibi tarih sütünü eklemiş olduk.

**DROP :**Bu parametre ile tablodan sütün silme işlemi yapmamızı sağlıyor.

**ALTER TABLE *tablo\_adı* DROP  COLUMN*sütün\_adi***

Şimdi tablomuzda ki bir sütünü silelim, ben az önce eklediğimiz tarih sütünü tekrar siliyorum.

|  |  |
| --- | --- |
|  | ALTER TABLE yazarlar DROP tarih |



Böylelikle*tarih* sütünü silinmiş oldu.

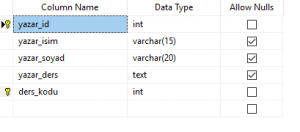
**ALTER:** Bu parametre ile tabloda belirtilen  sütünü güncellemenizi sağlıyor.

**ALTER TABLE *tablo\_adı* ALTER COLUMN *sütün\_adı   sütün özellikleri***

Şimdi tablo üzerinde ki*yazar\_ders* sütünü veri türü*varchar()* iken bunu *text* yapalım;

|  |  |
| --- | --- |
|  | ALTER TABLE yazarlar ALTER COLUMN yazar\_ders text |

Tablomuzun son haline bakalım:



Tablo üzerinde güncelleme işlemini de yapmış olduk böylelikle.

**DROP**

Bu komut oluşturmuş olduğumuz veritabanı ve tabloları silmeye yarıyor.

**Veritabanı Silme**

Var olan bir veritabanı silmek için genel kullanım:

**DROP DATABASE veritabanı\_adı**

|  |  |
| --- | --- |
|  | DROP DATABASE mobilhanem |

**Tablo Silme**

Var olan bir tabloyu silmek için genel kullanım:

**DROP TABLE tablo\_adı**

|  |  |
| --- | --- |
|  | DROP TABLE yazarlar |

Bu şekilde veritabanı ve tablo silme işlemlerini de öğrenmiş olduk.

Yazdığımız komutları Online çalıştırmak isterseniz [tıklayınız](https://www.tutorialspoint.com/execute_sql_online.php).

**Mobilhanem.com** da SQL Dersleri eğitim serisinde bir yazının daha sonuna geldik.Bu dersimizde SQL DDL komutlarını işledik. Bir sonraki derste**SQL DML Komutları** dersi işleyeceğim. Eksik yada yanlış gördüğünüz bir yer varsa dersin altındaki yorum kısmında veya iletişim bilgilerimden bana ulaşabilirsiniz bir sonraki derste görüşmek üzere.