Veri Kontrol Dili
(DCL-Data Control Language)

Arif GÜNEL

Veri Kontrol Dili (DCL-Data Control Language)

- *Hangi nesnelere , hangi kullanıcılar erişebilecek ve hangi kullanıcı hangi ifadeleri çalıştırabilecek türünden kısıtlama ve yetkilendirmelerin yapılmasını sağlayan komutlar.*
- *DCL, bir veri tabanı ile ilişkili kullanıcıları ve rollerin izinlerini değiştirmek için yani verilere erişim yetkilerini düzenlemek amacıyla kullanılır.*
- *Kısaca;*
  - ***Kullanıcı= Yapabilecekleri(İzinleri-Kısıtlamaları)***

- *Kullanıcıların yapabilecekleri işlemler;*
  - *Silme,*
  - *Değiştirme,*
  - *Görme*
  - *…..*

- *Neden yetkileri belirliyoruz?*
  - *Çünkü gereğinden fazla verilen izinler yani yetkiler veri güvenliğini riske atar,*
  - *Gereğinden az verilen yetkiler iş yapamaz hale getirir o yüzden gerektiği kadar yetki tanımlamak önemlidir.*

Üç temel komutu vardır:

Grant: *Belirtilen kişiye ya da gruba veri kullanma ve komut çalıştırma izni verir.*

Deny: *İşlem yapmak için verilen izinleri kısıtlar ya da kaldırır.*

Revoke: *Verilen-tanımlanan tüm izinleri ve kısıtlamaları iptal eder.*

DCL Komutlarını Kullanabilecek Yetkili Kullanıcılar

- ***sysadmin :*** *Sistem admin, genel yönetici*
- ***Dbcreator :*** *Veri tabanı tasarımcısı, oluşturma yetkisi*
- ***db_owner :*** *(Sahip) anlamında tüm yetkiye sahip*
- ***db_securityadmin :*** *Güvenlikten sorumlu olduğu için yetkileri kontrol eder.*

Kural (Role) Türleri

1.Server Role

2.Database Role

3.Application Role

Server Role

Sunucu tabanlı yetkiler(Admin)

- ***sysadmin*** *(System Administrator – Sistem Yöneticisi) : En yüksek yetkisi olandır. Sistem üzerinde tüm yetkilere sahiptir*
- ***securityadmin*** *(Security Administrator – Güvenlik Yöneticisi) : Server üzerinde kullanıcıların yetkilerini denetlemek, yönetmek ve şifrelerini sıfırlamak ya da istendiğinde değiştirme yetkisi olan*
- ***serveradmin*** *(Server Administrator – Server Yöneticisi) : SQL Server üzerinde yapı ayarları, başlat/durdur/yeniden başlat gibi yetkileri olan.*

- ***bulkadmin*** *(Bulk Insert Administrator – Çoklu Kayıt Yöneticisi) : Bulk Insert komutuna yetkisi olan*
- ***dbcreator*** *(Database Creator – Veritabanı Yöneticisi) : Database oluşturabilme, silebilme, düzenleyebilme yetkisi olan*
- ***diskadmin*** *(Disk Administrator – Dosya Yöneticisi) : Disk üzerinde bulunan dosyaları yönetme yetkisi olan*
- ***processadmin*** *(Process Administrator – İşlemci Yöneticisi) : SQL Server üzerinde çalışan işlemcileri kontrol etme yetkisi olan.*
- ***public*** *(Herkese Kısıtlı Hak) : SQL Server üzerinde standart ayarlarla giriş yapan herkesin rolüdür. Bu kural ile tüm kullanıcıların kısıtlı hakları vardır. Daha sonra bu kullanıcılara kural değişikliği yapılarak diğer kurallar atanabilir*

Database Role?

- ***db_owner** : Veri tabanında en yüksek yetkilidir/Veri tabanı sahibi. Silme, ekleme, düzenleme, başlatma/durdurma gibi yetkileri vardır*
- ***db_accessadmin** : Veri tabanında kullanıcılara veri tabanına erişim kural/yetki atayan yöneticidir*
- ***db_securityadmin** :Veri tabanındaki kural/yetkileri yönetir. Güvenlik yöneticisi.*
- ***db_datareader** : Veri tabanındaki kullanıcıların "SELECT" sorgusunu çalıştırmasına izin verir*
- ***db_datawriter** : Veri tabanındaki kullanıcıların "INSERT , DELETE , UPDATE sorgularını çalıştırmasına izin verir*

- ***db_ddladmin*** *: Veri tabanındaki kullanıcıların “DDL” komutlarını çalıştırmasına izin verir*
- ***db_denydatareader*** *: Veri tabanındaki kullanıcıların “SELECT” sorgusunu çalıştırmasını kısıtlar. Sadece okuma ile yetkili.*
- ***db_denydatawriter*** *: Veri tabanındaki kullanıcıların “INSERT , DELETE , UPDATE sorgularını çalıştırmasını kısıtlar. Veri tabanına veri girişi yapmakla yetkili.*
- ***db_backupoperator*** *: Veri tabanının yedeğini alma operatörü*

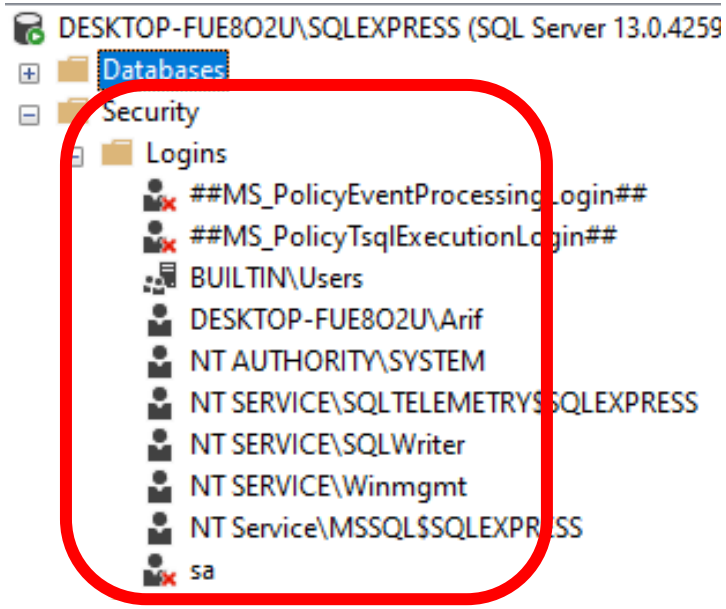
Application Role?

- *Eğer “guest” hesabı oluşturursanız veri tabanında, database için tek tek kullanıcı hesabı açıp tanımlamanıza gerek kalmaz.*
- *Bunu da aşağıdaki iki procedure komutu ile kullanılabilir ya da kullanılamaz olarak ayarlayabilirsiniz.*
  - ***sp_setapprole** : Kullanılabilir*
  - ***sp_unsetapprole** : Kullanılamaz*

Sunucuya dışarıdan erişmek için Login Oluşturalım

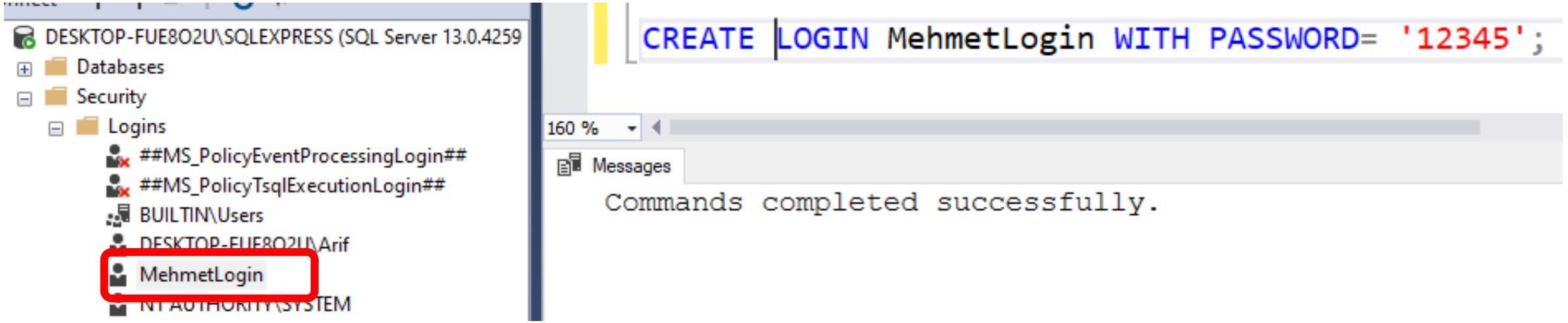
- *Sql Server'daki kullanıcıların hesaplarına Logins denmektedir.*
- *Her bir login'e ayrı yetkiler verilebilir, erişebileceği veri tabanları belirlenebilir.*
- *Sql Server'da güvenlik seviyesi iki katmandan oluşur Server ve Veri tabanı seviyesi.*
- *Login hesabı ilk önce server seviyesi oluşturulur, daha sonra ilgili veri tabanına mapped denilen ataması yapılır ve veri tabanına erişimi sağlanmış olur.*

Mehmet isminde Login oluşturalım.




```
CREATE LOGIN MehmetLogin WITH PASSWORD= '12345';
```

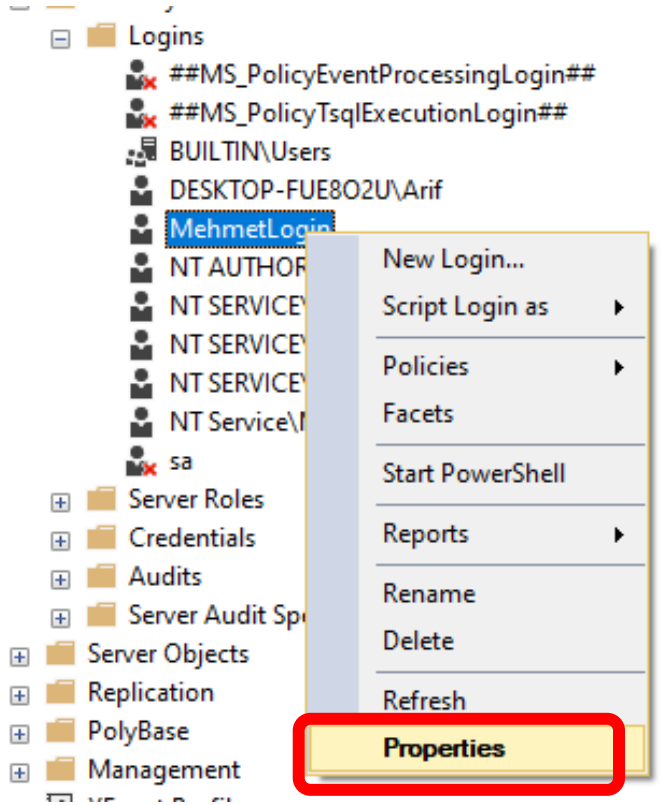
- *Kullanıcı adı Mehmet, parolası 12345*
- *Menüde sol tarafta Security altında Mehmet isimli bir kullanıcın olmadığı görülüyor kodu çalıştırınca refresh dersek MehmetLogin isimli kullanıcının oluştuğunu görürüz*



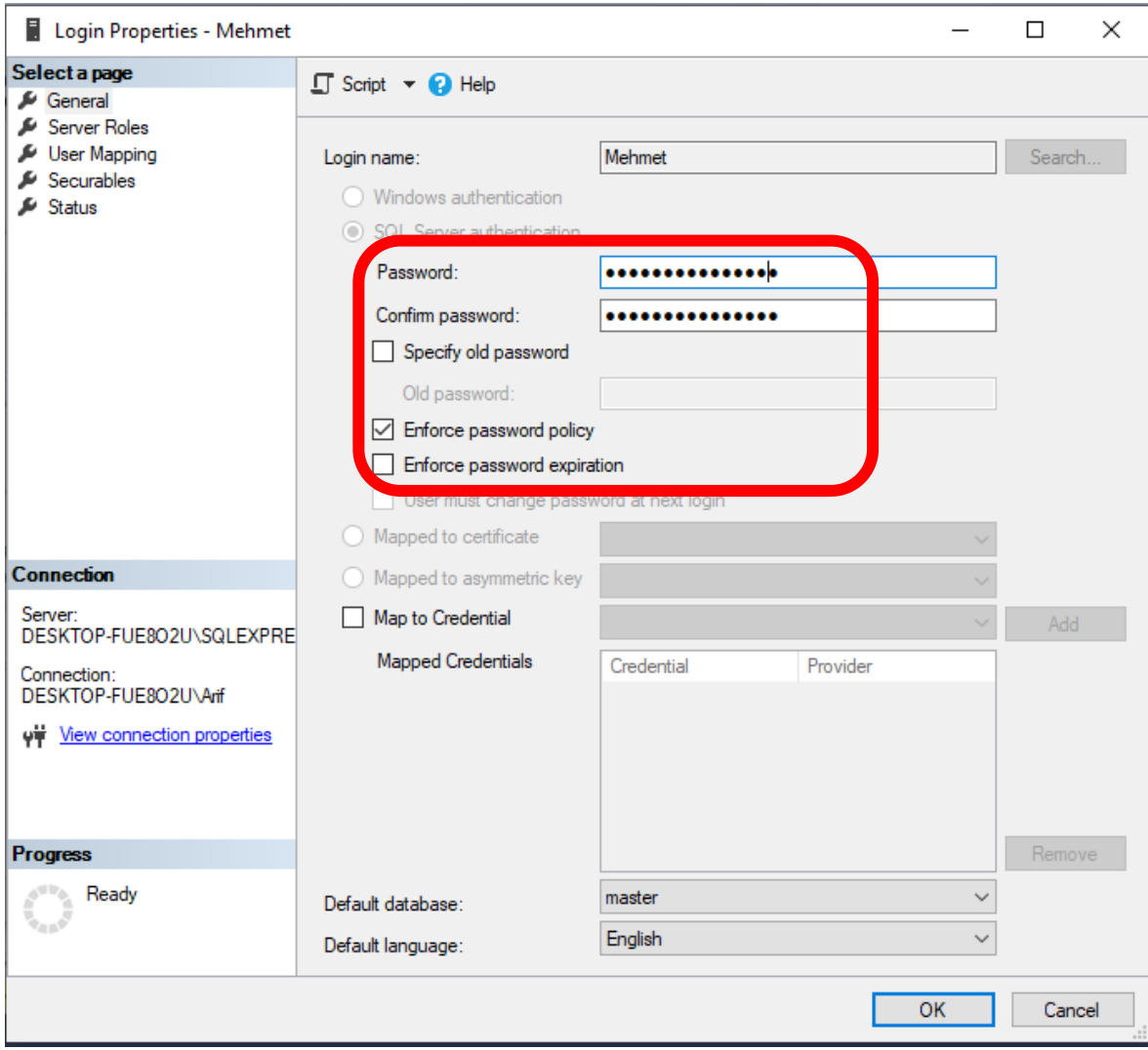
- *Kodu çalıştırınca MehmetLogin isimli yeni kullanıcı geliyor*

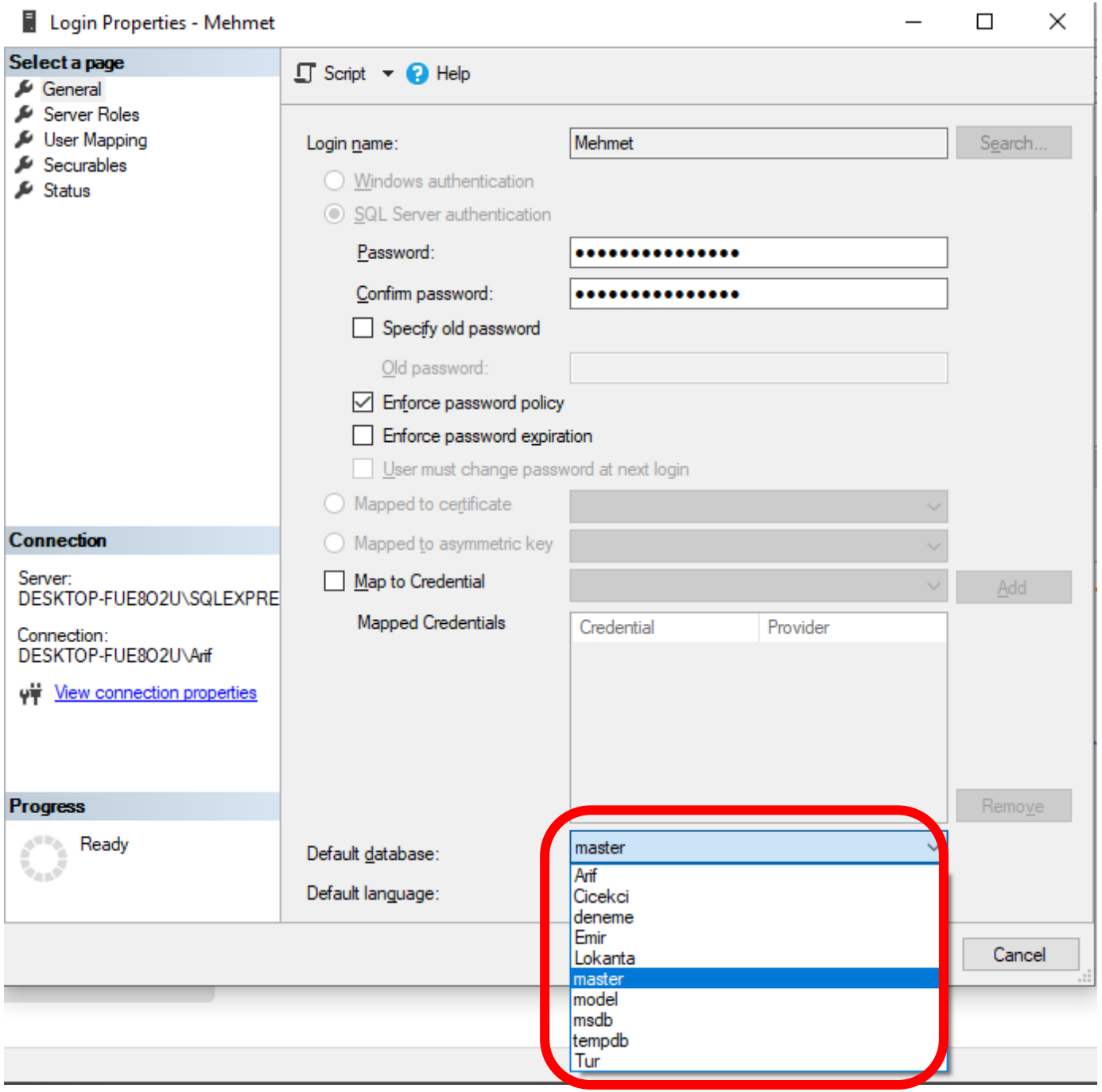
MehmetLogin isimli kullanıcının özelliklerine bakalım

- Üzerinde sağ tık Properties diyoruz



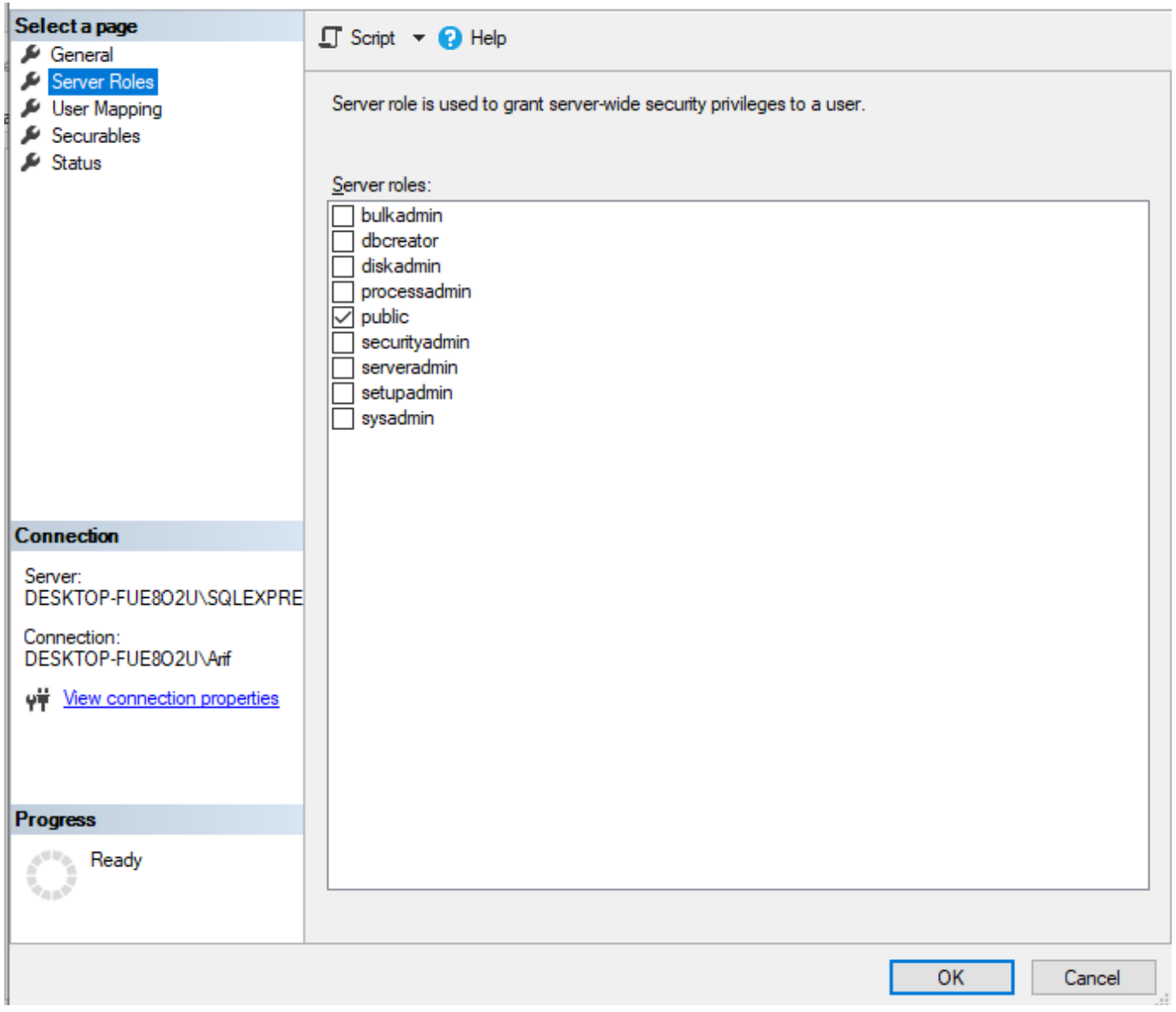
- *Bu kısımdan parola değişimi, belli aralıklarla güncelleme ve diğer ayarları yapabiliriz.*





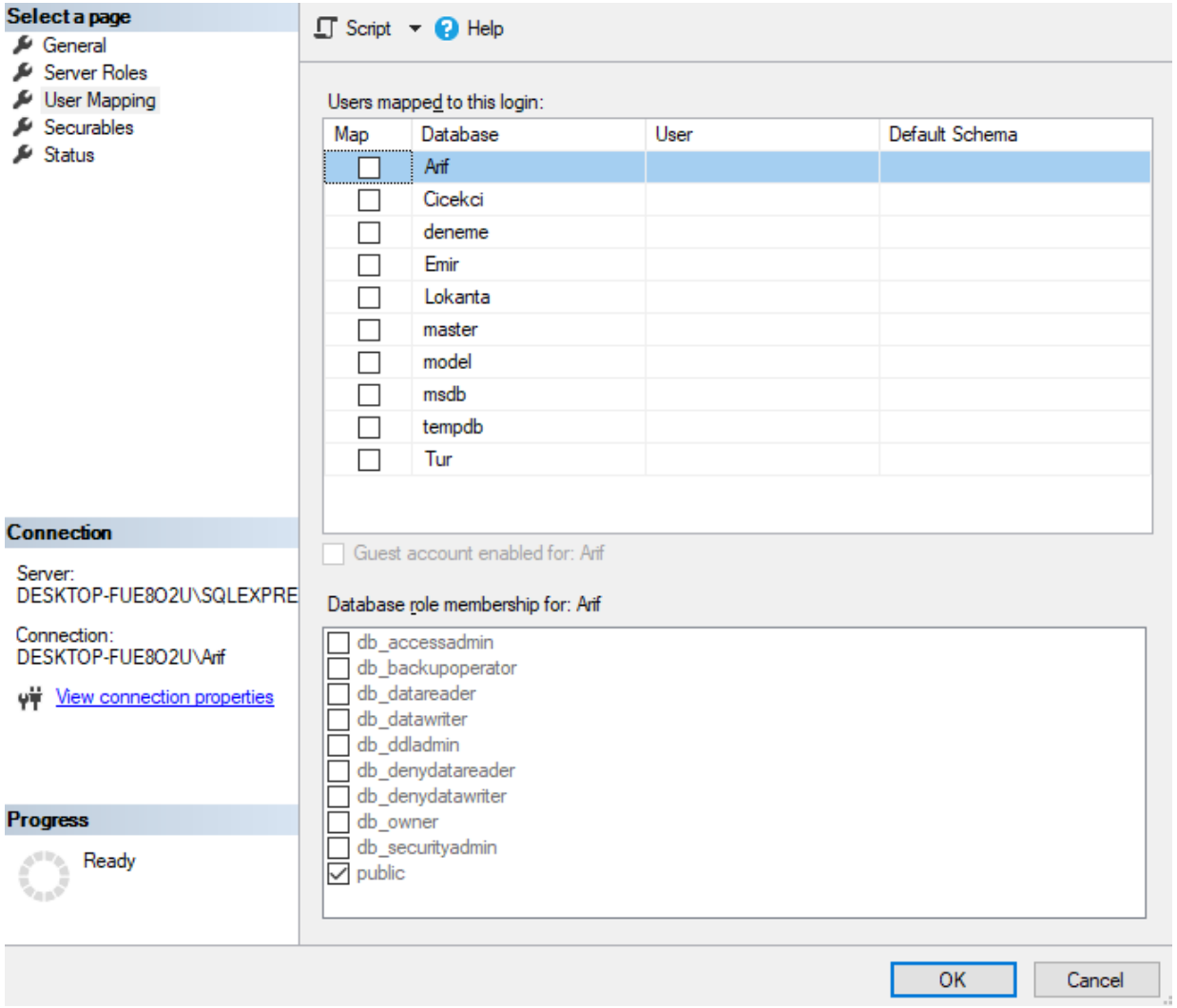
- *Loginin üzerinde işlem yapacağı veri tabanını seçiyoruz*
- *Master tüm veri tabanlarına erişim,*
- *Yada sadece bir tane veri tabanına erişim tanımlayabilirsiniz*
- ***Default Language** dil ayarı yapılıyor*

Server Role



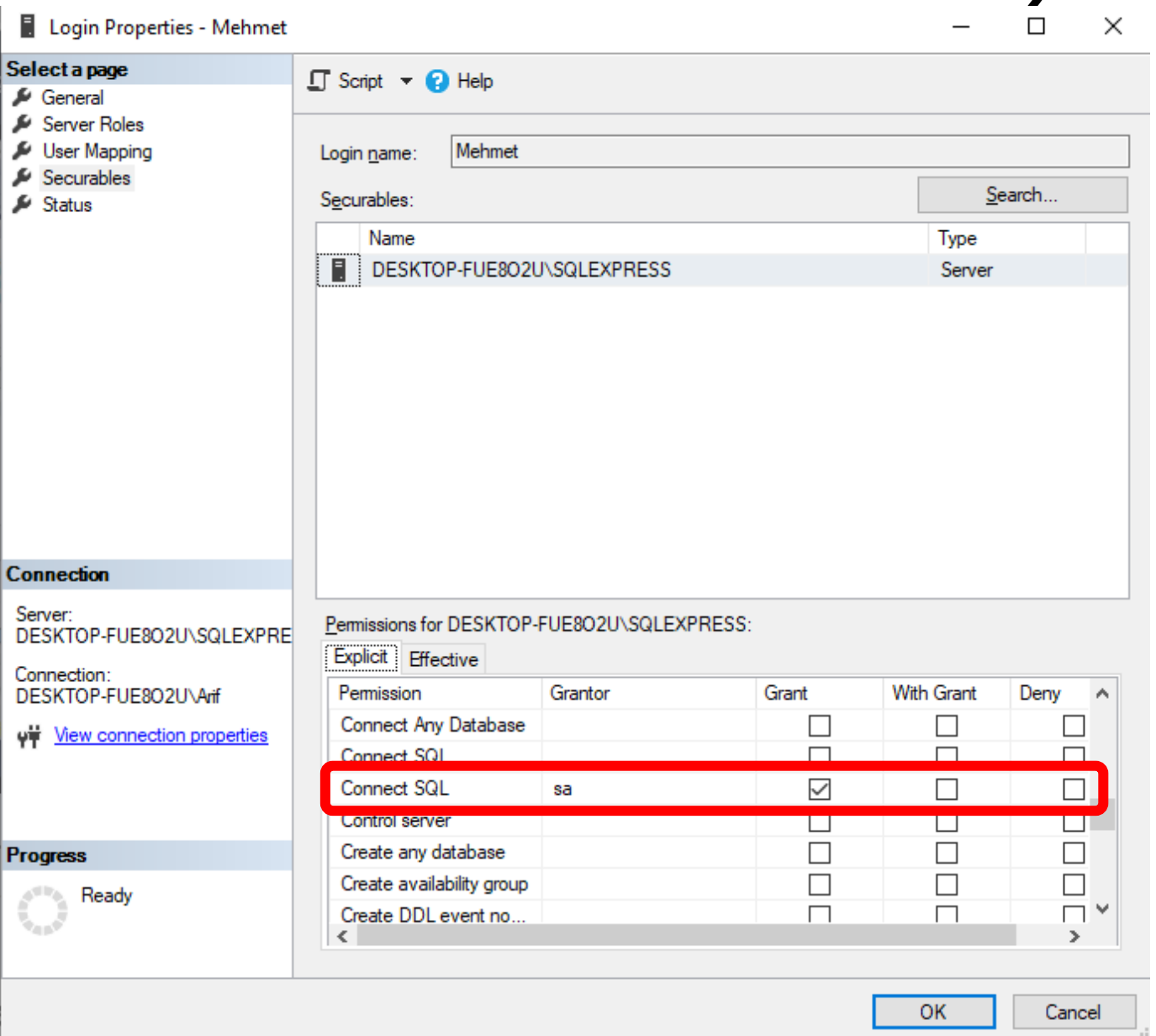
- *Server Roles kısmı bu kullanıcının yetkilerinin seviyesini belirleniyor sistem admin mi güvenlik yöneticisi mi gibi seviyesinin belirliyoruz*

User Mapping-Kullanıcılar Arası İlişki Kurmak



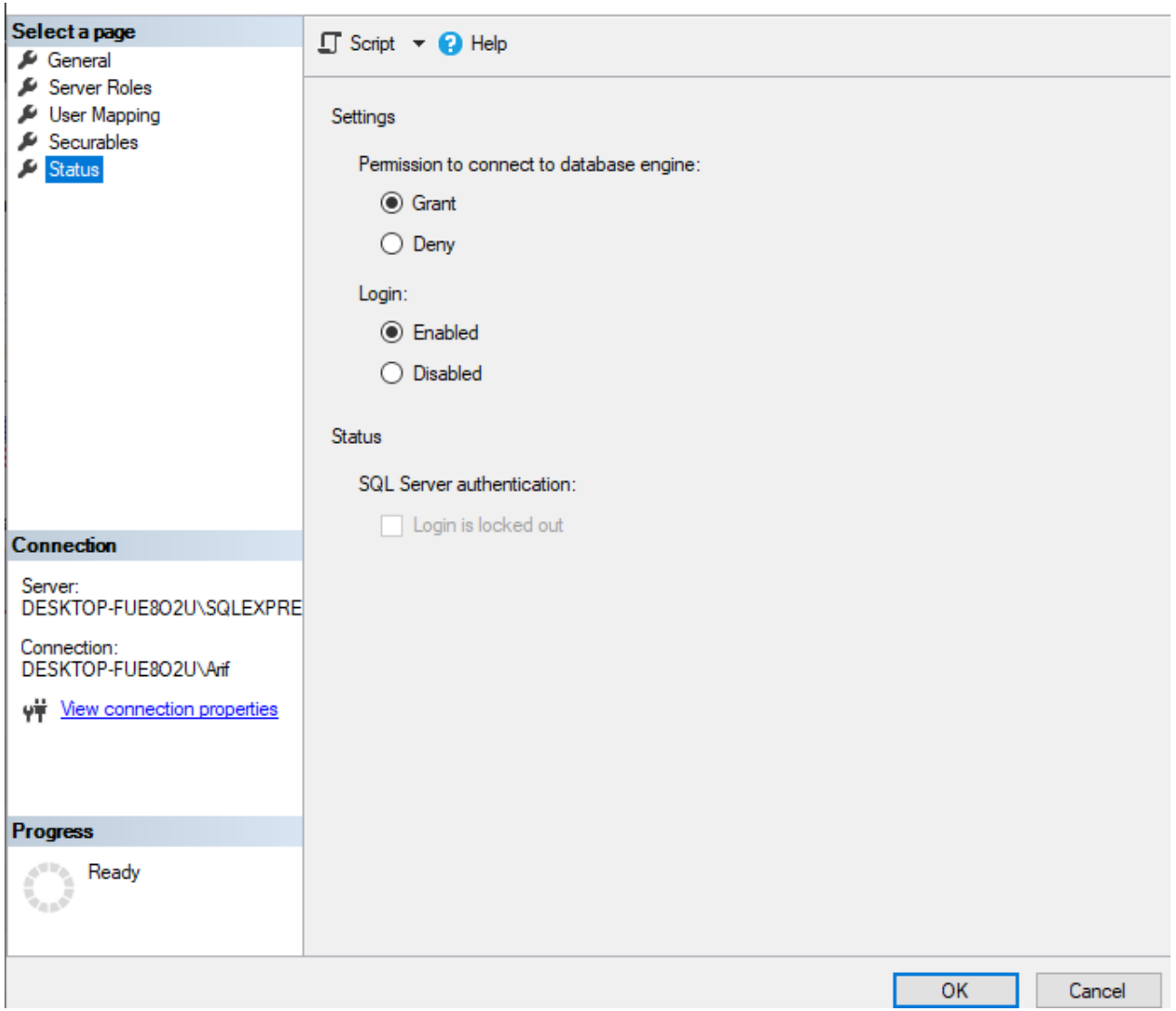
- *Veri tabanlarının yetkilendirme seviyelerini ayarladığımız kısım*
- *Login ve User arasında ilişki kurmak(mapping) için kullanabiliriz.*

Securables-Server Seviyeli Rol ve İzin Atanması



- *Permission(izin) kısmına **göz attığımızda** Connect SQL **Grant** yani Mehmet isimli kullanıcının SQL Sunucusuna(Server) bağlantı yetkisi tanımlanmıştır.*
- *Bu kısmı **Deny** yaparsak Mehmet isimli kullanıcı sunucuya bağlanamaz.*
- *Diğer özelliklere bakarsak sunucu tabanlı işlemler olduğunu görüyoruz*

Status



- *Settings kısmından MehmetLogin isimli kullanıcının veri tabanı motoruna bağlanma izninin olması(Grant) yada olmaması(deny) izinlerini ayarlıyoruz.*
- *Login kısmı ise etkin/geçerli(Enabled) ya da devre dışı(Disabled) yapıyoruz.*

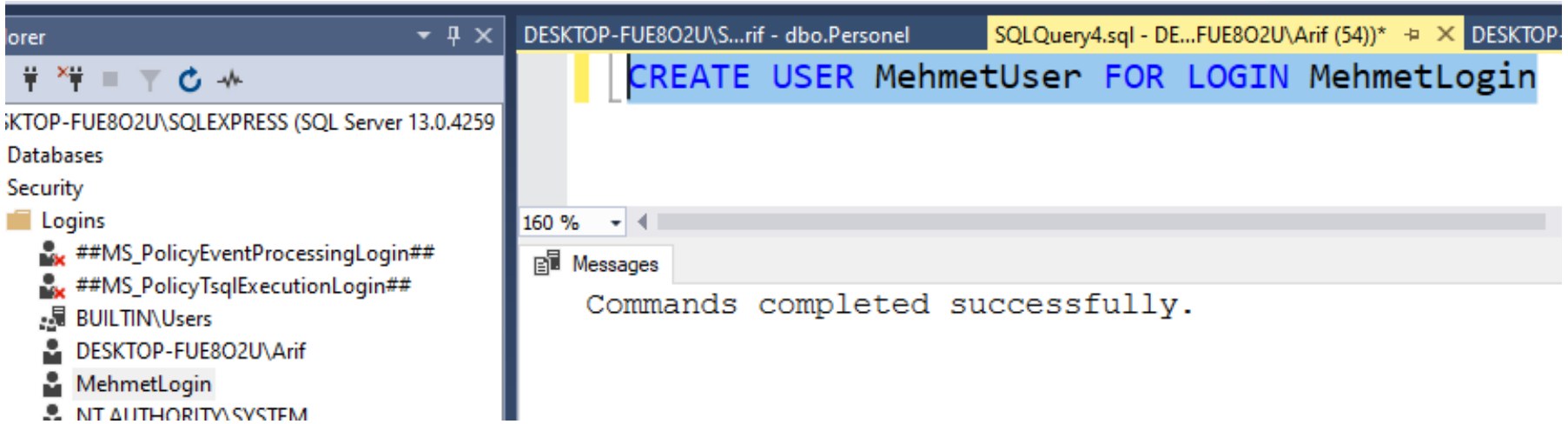
User-Kullanıcı oluşturma

```
CREATE LOGIN MehmetLogin WITH PASSWORD= '12345’;
```

- *İlk olarak server bağlantısı için bir login oluşturduk şimdi bu loginin altında bir User oluşturalım.*
- *Bunu şu şekilde düşünebiliriz siz bir yol oluşturuyorsunuz bu yolu bir çok kullanıcı kullanabiliyor.*
- *Bu durum çalışmaları yönetme düzenlemede yardımcı oluyor bir nevi takım oluşturup(login) bu takımda farklı görev/kısıtlamaya sahip oyuncular(user) oluşturuyorsunuz.*

MehmetLogin bağlantısının altında MehmetUser isimli kullanıcı oluşturuyoruz

- `CREATE USER MehmetUser FOR LOGIN MehmetLogin`



- Not: Login ile user aynı isimde ise FOR LOGIN yazmaya gerek yok
- `CREATE USER MehmetLogin ~~FOR LOGIN MehmetLogin~~`

Application Role - Uygulama Rolü

- *Application Role veri tabanı seviyesinde tanımlanan bir roldür.*
- *Application Rolleri kullanarak bir veri tabanına belli yöntem ve belirli kullanıcılarla erişim sağlanması için kullanılır.*
- *Uygulamayı etkinleştirmesini sağlamak için kullanılan kendi parolası vardır.*

```
CREATE APPLICATION ROLE AppRole
WITH PASSWORD = 'app_role_pwd',
DEFAULT_SCHEMA= AppRole;
```

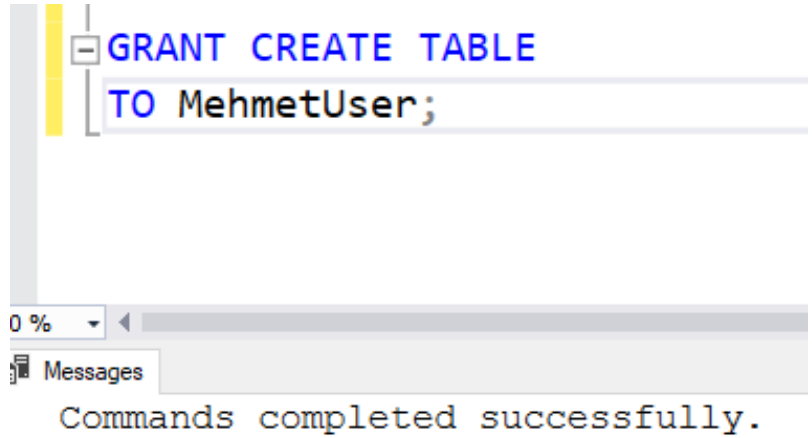
GRANT ile İzin Oluşturma

- *Daha önce tanımlamış olduğumuz veri tabanı kullanıcısına, veri tabanı rolü ya da uygulama rolüne izinler vermek için kullanılan komuttur.*
- *Kullanım kalıbı;*

```
GRANT ‘Hangi nesnelere verilecek izinler’
TO ‘İzin verilenler’
```

MehmetUser isimli kullanıcıya tablo oluşturma yetkisi verelim

```
GRANT CREATE TABLE
TO MehmetUser;
```



- ***MehmetUser isimli kullanıcıya tüm yetkilerin verilmesi***

```
GRANT ALL
TO MehmetUser;
```

Role' lere de yetki tanımlayabiliriz

- *Burada hem Role hem de user için tablo oluşturma izni veriyoruz.*
- *Birden fazla izin vermek için virgül(,) kullanıyoruz*

```
GRANT CREATE TABLE
TO MehmetUser, AppRole;
```

Birden fazla yetki verme

MehmetUser isimli kullanıcıya veri tabanı ve tablo oluşturma yetkisi verme.

```
GRANT CREATE DATABASE,CREATE TABLE
TO MehmetUser;
```

WITH GRANT OPTION ile Basamaklı Yetkilendirme

- *İş yerlerindeki birim sorumluları gibi düşünebiliriz. Birim sorumlusu yöneticiden aldığı yetkileri isterse ekibinde çalışanlara da verebilir.*
- *Bu deyim ile izin verilen bir kullanıcının bu nesne üzerinde aldığı izni bir başka kullanıcıya verebilmesi için kullanılır.*

Örnek sorgu : WITH GRANT OPTION

- *Bu durumda Mehmet ile belirtilen role sahip herkes, Personel tablosu üzerinde başkalarına da SELECT(Seçme) ve INSERT(Ekleme) erişim izni tanımlama hakkına sahip olacaktır.*

```
GRANT SELECT, INSERT ON Personel
TO Mehmet
WITH GRANT OPTION;
```

DENY ile Erişim Kısıtlama-Engelleme

- *Kullanıcıların erişimlerini kısıtlamak için kullanılır.*

- *Kalıbı;*

```
DENY { ALL veya izinler}
TO {kullanıcılar}
```

MehmetUser isimli kullanıcıya verilen tablo oluşturma yetkisinin kaldırılması

```
DENY CREATE TABLE
TO MehmetUser;
```

MehmetUser isimli kullanıcının tbPersonel isimli tablodan veri çekme yetkisinin kaldırılması

```
DENY SELECT ON tbPersonel
TO MehmetUser
```

REVOKE ile Erişim Tanımını Kaldırmak

- *Tüm kısıtlama ve izinleri iptal etmek için kullanılır.*
- *Bir nesneyi oluşturan kullanıcının, nesne üzerindeki yetkilendirme ve kullanım hakkı iptal edilemez. Yani üreten kimse hakkı saklıdır.*
- *REVOKE komutunu, sys_admin rolü ya da db_owner, db_securityadmin sabit veritabanı rollerine sahip kullanıcılar ve nesne için, dbo olan kullanıcı çalıştırabilir.*

REVOKE Kalıbı

```
REVOKE {ALL veya izinler}
{TO veya FROM} {hesaplar}
```

- *Mehmet ismine verilen tüm yetkileri kaldırmak için;*

```
REVOKE ALL TO Mehmet
```

MehmetUser isimli kullanıcının tbPersonel isimli tablodan veri çekme yetkisinin kaldırılmasının iptali

Alt kısımdaki sorgu ile yetkiyi iptal etmiştik.

```
DENY SELECT ON tbPersonel
TO MehmetUser
```

Şimdi var olan izin kısıtlamasını iptal edelim

```
REVOKE SELECT ON tbPersonel TO MehmetUser
```

İngilizce Kelimeler

- *Grant : İmtiyaz, ruhsat*
- *Deny : İzin vermemek, reddetmek*
- *Revoke : Geri almak, kaldırmak, iptal etmek*
- *securable : emniyetli olarak*
- *Explicit : Belirgin, aşikar, açık*
- *Permission : İzinler, erişim izni*
- *Grantor : Hibe eden, bağışlayan*
- *Effective : Etkili, geçerli*
- *Connect : Bağlantı, bağlamak*
- *Status : Durum, vaziyet*

Kaynaklar

- *https://www.veriyum.net/v1/mssql-server-role-nedir-nasil-olusturulur-ne-ise-yarar.html*
- *http://www.veritabani.gen.tr/2017/04/24/application-role-nedir/*
- *YAZILIMCILAR İÇİN İLERİ SEVİYE T-SQL PROGRAMLAMA-Cihan ÜNAL*
- *https://www.udemy.com/course/sql-server-veritabani-programlama/learn/lecture/7271902#overview*
- *https://www.sciencedirect.com/topics/computer-science/application-role*