

Veritabanı Programlama

Bölüm 3

SQL Server ve Kurulumlar

Dr. Öğr. Üyesi Murat TAŞYÜREK (kayubmprogramlama1@gmail.com)

19 Ekim 2023

Kayseri Üniversitesi, Bilgisayar Mühendisliği Bölümü

T-SQL ile İlgili Kurallar

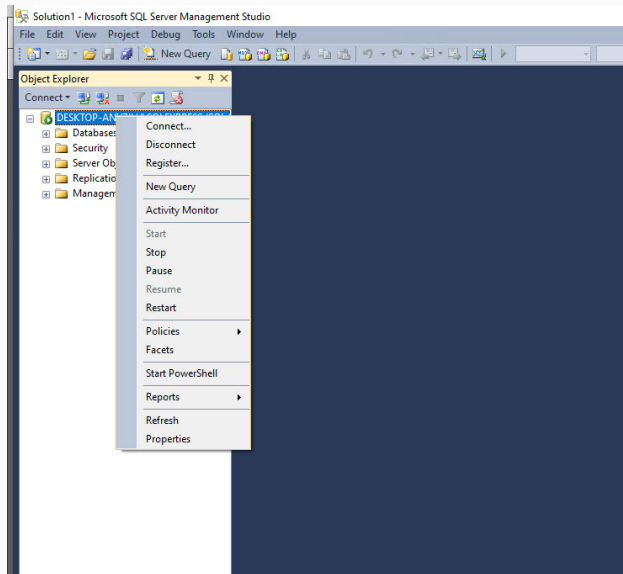
- T-SQL'de tablo, sütun, index, procedur ve diğer tüm nesnelerin isimleri belirlerken uyulması gereken standartlar vardır.
- Bu standartlara uyulmadığı takdirde geliştirdiğimiz program anlaşılır olmayacak ve beklenmedik hatalar ile karşılaşacağız.
- Nesne isimleri Harf (a-Z) ile ya da alt çizgi (_) ile başlamalıdır. Rakam ile başlamamalıdır.
- İsimler @, @@, #, ##, \$ gibi özel karakterler ile başlamamalıdır.
- SQL ve T-SQL mimarisinde tanımlanmış (SELECT, INSERT, UPDATE, DELETE ve benzeri gibi) kelimeler kullanılmamalıdır.
- Türkçe karakter ve boşluk kullanılmamalıdır.

- @ : Değişken tanımlarken değişken adının başında kullanılır.
- @@ : Sistem değişkenlerinin isimlerinin başında kullanılır.
- # ve ## : Geçici nesne tanımlamak için kullanılır.

T-SQL Değişken Tanımlama

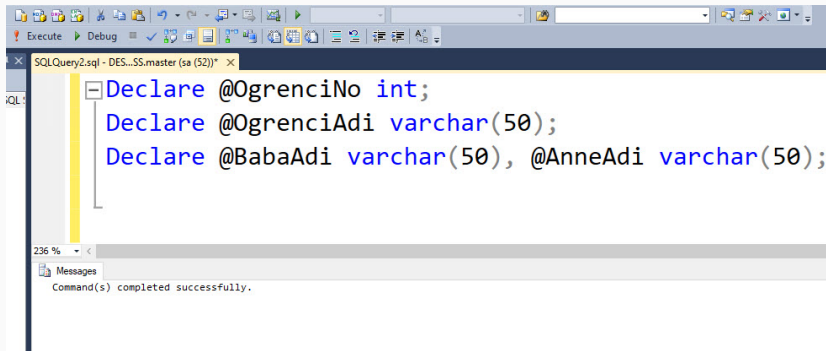
- Değişken tanımlama **DECLARE** komutu ile yapılır.
- Genel yazım şekli aşağıdaki gibidir.
- **DECLARE @degiskenadi veriTipi (boyut)**
- Veri tipi ise sayısal değerler için INT veya karakter türündeki veriler için VARCHAR(boyut) olabilir.
- Tablo oluşturulması sırasında sütun alanlarında kullanılanabilen veri tipleri değişkenler için de geçerlidir.
- Aynı satırda değişken tanımlama ve atama yapabilmek için ilk komuttan sonra noktalı virgül(;) kullanılması gerekir.

SQL Server New Query Window



T-SQL Değişken Tanımlama

- Değişken tek satırdada tanımlanabilir
- Ayrı ayrı satırdada tanımlanabilir.



The screenshot shows a SQL Server Enterprise Manager window with a query editor. The query text is as follows:

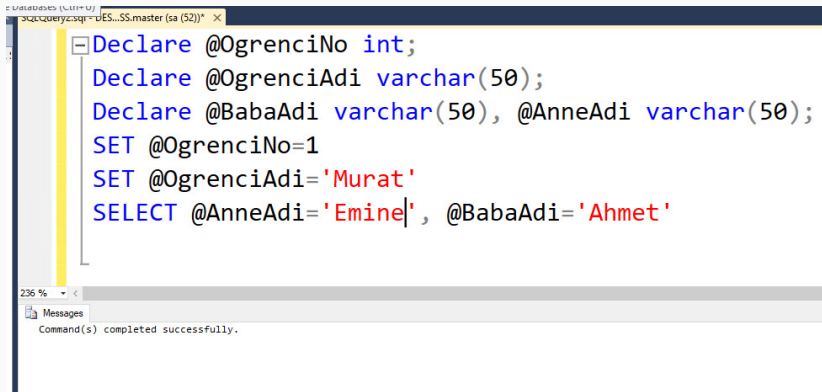
```
Declare @OgrenciNo int;  
Declare @OgrenciAdi varchar(50);  
Declare @BabaAdi varchar(50), @AnneAdi varchar(50);
```

Below the query editor, the Messages pane shows the following output:

```
Command(s) completed successfully.
```

T-SQL Değişken Değer Atama

- T-SQL'de değişkenlere değer atamak için **SET** ve **SELECT** terimi kullanılır.



The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays a query window titled "Query - SS.master (sa (52))". The query text is as follows:

```
Declare @OgrenciNo int;  
Declare @OgrenciAdi varchar(50);  
Declare @BabaAdi varchar(50), @AnneAdi varchar(50);  
SET @OgrenciNo=1  
SET @OgrenciAdi='Murat'  
SELECT @AnneAdi='Emine', @BabaAdi='Ahmet'
```

The bottom pane shows the "Messages" tab with the message: "Command(s) completed successfully."

T-SQL Değişken Değerini Yazdırma

- T-SQL'de değişkenlere atanan değerleri görüntülemek için **PRINT** ve **SELECT** terimi kullanılır.
- **PRINT** terimi ile ekrana yazılan değerler **Messages** tabında görüntülenir.
- **SELECT** terimi ile ekrana yazılan değerler **Results** tabında görüntülenir.
- **SELECT** terimi ile ekrana yazılan değerlerde **AS** terimi ile alias tanımlanarak kullanılabilir. Bu sayede değerler sanki bir tablodan veri çekiliyormuş gibi **C#** tarafında okunabilir.

T-SQL Print


```
SQLQuery2.sql - DES...SS.master (sa (52)) *  
--  
Declare @OgrenciNo int;  
Declare @OgrenciAdi varchar(50);  
Declare @BabaAdi varchar(50), @AnneAdi varchar(50);  
SET @OgrenciNo=1  
SET @OgrenciAdi='Murat'  
SELECT @AnneAdi='Emine', @BabaAdi='Ahmet'  
  
PRINT @OgrenciNo  
PRINT @OgrenciAdi  
SELECT @AnneAdi as AnneAdi, @BabaAdi as BabaBadi
```

236 %

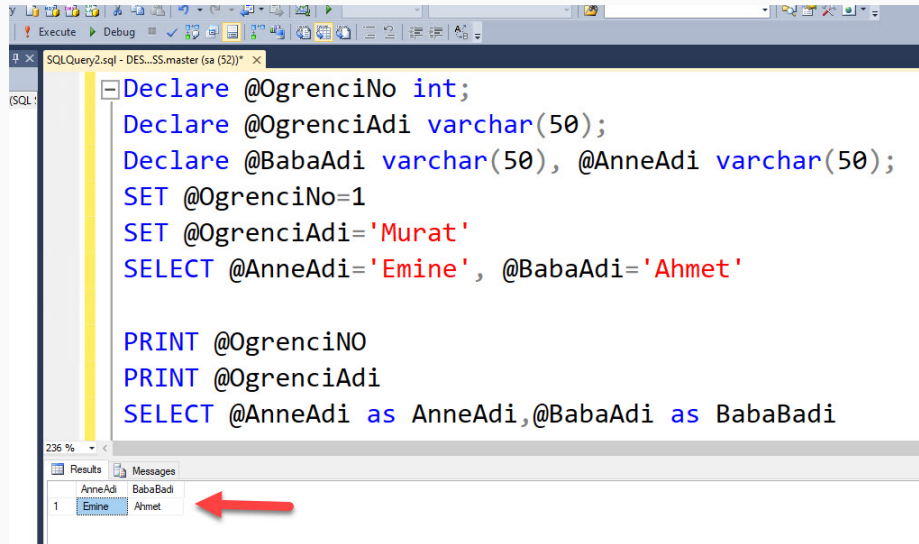
Results Messages

1	Murat
---	-------

(1 row(s) affected)



T-SQL SELECT



The screenshot shows a SQL Server Enterprise Manager window with a T-SQL query editor and a results pane. The query editor contains the following code:

```
Declare @OgrenciNo int;  
Declare @OgrenciAdi varchar(50);  
Declare @BabaAdi varchar(50), @AnneAdi varchar(50);  
SET @OgrenciNo=1  
SET @OgrenciAdi='Murat'  
SELECT @AnneAdi='Emine', @BabaAdi='Ahmet'  
  
PRINT @OgrenciNo  
PRINT @OgrenciAdi  
SELECT @AnneAdi as AnneAdi,@BabaAdi as BabaBadi
```

The results pane shows the output of the query. It has two tabs: "Results" and "Messages". The "Results" tab is active, displaying a table with two columns: "AnneAdi" and "BabaBadi". The table contains one row with the values "Emine" and "Ahmet". A red arrow points to the "Emine" value in the "AnneAdi" column.

	AnneAdi	BabaBadi
1	Emine	Ahmet

T-SQL IF Yapısı

- T-SQL'de programlama dillerine olduğu gibi **IF ELSE** yapısı vardır.
- **IF** koşulu sağlanırsa o koşua ait kod bloğu çalışır.
- Koşula ait çalışak kod tek satır ise **beging end** bloğuna gerek yoktur. Ancak birden fazla satır var ise **begin end** bloğu kullanılmalıdır.

IF (koşul İfadesi 1)

Bir SQL ifadesi ya da BEGIN...END bloğu

ELSE IF (Koşul İfadesi 2)

Bir SQL ifadesi ya da BEGIN...END bloğu

ELSE

Bir SQL ifadesi ya da BEGIN...END bloğu

T-SQL Örnek

- T-SQL'de tanımla bir değişken için değişkenin değeri 2'den küçük ise sadece ekrana merhaba yazdırsın.
- Değişkenin değeri 3 ile 7 arasında ise değişkenin karesini alsın ve ekrana bu değeri yazdırsın
- Değişkenin değeri 8 ile 10 arasında ise değişkenin küpünü alsın ve ekrana bu değeri yazdırsın
- Değişkenin değeri 10'den büyük ise bu değerinin 2 katını ekrana yazdırsın.
- ilgili örneği T-SQL de kodlayalım.

T-SQL Örnek Kod

```
SQLQuery2.sql - DES...SS.master (sa (52))> X<
-- Declare @Degisken INT;
-- SET @Degisken=3
-- IF @Degisken <2
--     SELECT @Degisken as DegiskeninDeri;
-- ELSE IF @Degisken >=3 AND @Degisken<=7
--     BEGIN
--         DECLARE @Karesi INT;
--         SET @Karesi=@Degisken*@Degisken
--         SELECT @Karesi as DegiskeninKaresi
--     END
-- ELSE IF @Degisken BETWEEN 8 AND 10
--     BEGIN
--         DECLARE @Kupu INT;
--         SET @Kupu=@Degisken*@Degisken*@Degisken
--         SELECT @Kupu as DegiskeninKupu
--     END
-- ELSE
--     SELECT @Degisken*2 AS Degiskenin2Kati
```

236 %

Results Messages

	DegiskeninKaresi
1	9

T-SQL Loops

- T-SQL'de For Loop döngüsü bulunmamaktadır. PL-SQL (Oracle)'da vardır ve kullanımı çok kolaydır.
- T-SQL'de For Loop döngüsünü While döngüsü ile sağlanıdır.
- While döngüsünün çalışma mantığı aşağıdaki gibidir.

```
WHILE koşul_ifadesi  
BEGIN  
    Tekrarlanacak işlemler  
END
```

- T-SQL'de 1'den 50'ye kadar olan sayılardan 7'ye tam olarak bölünenleri ekrana yazdıralım.
- 7'ye bölünen her bir sayı için Messages tabında X sayısı 7'ye tam olarak bölünüyor mesajı gösterelim.
- İlgili örneği T-SQL'de kodlayalım.

T-SQL Örnek 2 Kod

```
DECLARE @Degisken INT;  
SET @Degisken=1  
  
WHILE @Degisken < 50  
BEGIN  
    IF @Degisken%7=0  
    BEGIN  
        DECLARE @Metin VARCHAR(50)  
        SET @Metin=STR(@Degisken)+' i 7 ye tam olarak bölünüyor'  
        PRINT @Metin  
    END  
    SET @Degisken=@Degisken+1  
END;
```

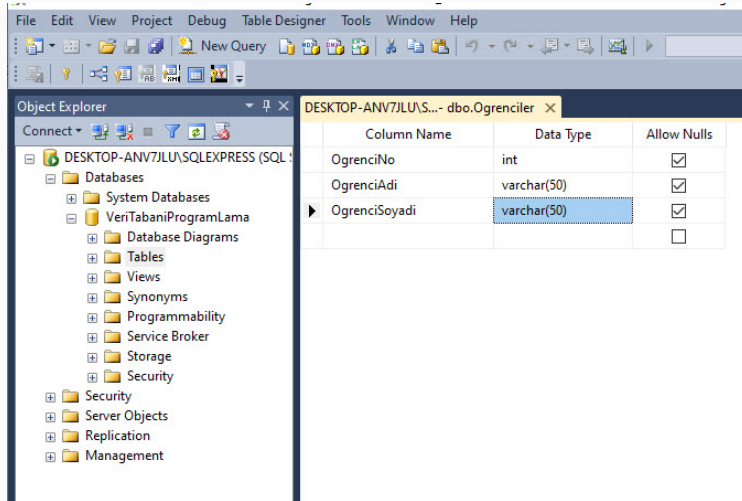
Messages

```
7 i 7 ye tam olarak bölünüyor  
14 i 7 ye tam olarak bölünüyor  
21 i 7 ye tam olarak bölünüyor  
28 i 7 ye tam olarak bölünüyor  
35 i 7 ye tam olarak bölünüyor  
42 i 7 ye tam olarak bölünüyor  
49 i 7 ye tam olarak bölünüyor
```


T-SQL Örnek 3

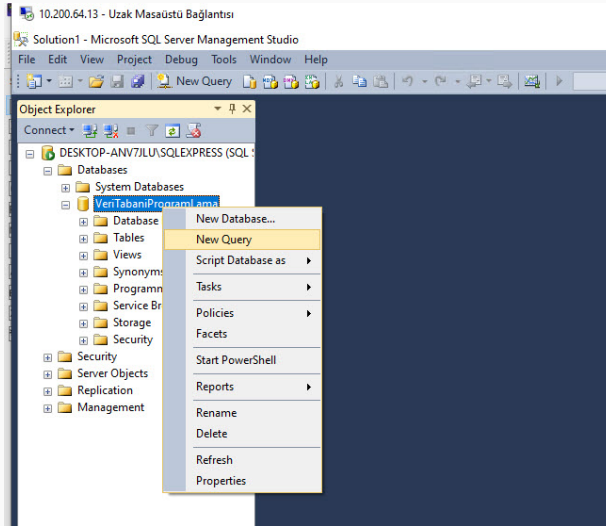
- SQL Server'da bir database oluşturun.
- Oluşturduğunuz database Ogrenciler isimli bir tablo oluşturun.
- Oluşturduğunuz tablonun OgrenicNO int, OgrenciAdi varchar(50) ve OgrenciSoyadi varchar(50) karakter olsun.
- 1'den 30'a kadar otomatik olarak öğrencileri ilgili tabloya kaydedin.
- Kayıt işlemi yaptıktan sonra ilgili kayıtları **öğrenci numarasına göre azalan şekilde sıralayım.**
- İlgili Örneği T-SQL'de kodlayalım.

T-SQL Örnek 3 Veritabanı

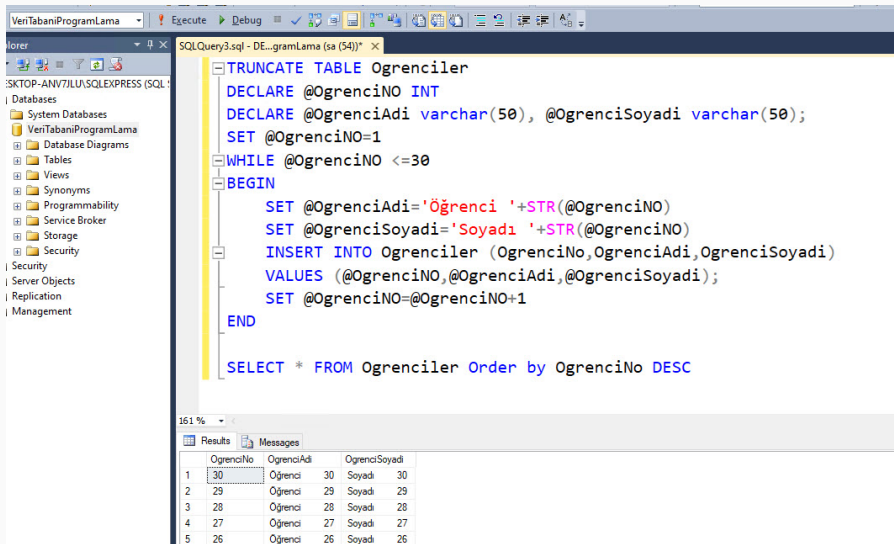


T-SQL Örnek 3 Veritabanı

- Yazacağımız sorguyu yeni veritabanında çalışacak şekilde ayarlıyoruz.



T-SQL Örnek 3 Kod



The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the 'VeriTabaniProgramLama' database selected under 'Databases'. The right pane shows a T-SQL script in the 'SQLQuery3.sql' window. The script performs a truncation, declares variables, and inserts data into the 'Ogrenciler' table. Below the script, the 'Results' pane shows the output of the query, displaying a table with 5 rows of student data.

```
TRUNCATE TABLE Ogrenciler  
DECLARE @OgrenciNO INT  
DECLARE @OgrenciAdi varchar(50), @OgrenciSoyadi varchar(50);  
SET @OgrenciNO=1  
WHILE @OgrenciNO <=30  
BEGIN  
    SET @OgrenciAdi='Öğrenci '+STR(@OgrenciNO)  
    SET @OgrenciSoyadi='Soyadı '+STR(@OgrenciNO)  
    INSERT INTO Ogrenciler (OgrenciNo,OgrenciAdi,OgrenciSoyadi)  
    VALUES (@OgrenciNO,@OgrenciAdi,@OgrenciSoyadi);  
    SET @OgrenciNO=@OgrenciNO+1  
END  
  
SELECT * FROM Ogrenciler Order by OgrenciNo DESC
```

	OgrenciNo	OgrenciAdi	OgrenciSoyadi
1	30	Öğrenci 30	Soyadı 30
2	29	Öğrenci 29	Soyadı 29
3	28	Öğrenci 28	Soyadı 28
4	27	Öğrenci 27	Soyadı 27
5	26	Öğrenci 26	Soyadı 26

T-SQL Case Yapısı

- Case komutu, birden fazla koşulun kontrol edilmesi gerektiğinde kullanılan bir komuttur.
- Case yapısı SQL sorgu ifadelerinin içinde de koşul amacıyla kullanılmaktadır.
- T-SQL'de bu yapının iki farklı kullanımı mevcuttur.

```
CASE Değer
  WHEN Değer ya da aralık THEN Sonuç_ifadesi_1
    WHEN Değer ya da aralık THEN Sonuç_ifadesi_2
    * * *
  ELSE Sonuç_ifadesi_n
END
```

```
CASE
  WHEN Koşul_1 THEN Sonuç_ifadesi_1
    WHEN Koşul_2 THEN Sonuç_ifadesi_2
    * * *
  ELSE Sonuç_ifadesi_n
END
```

T-SQL Örnek 4

- Oluşturduğunuz Öğrenciler tablosunda cinsiyet bilgisi için bir Cinsiyet varchar(1) sütunu ekleyelim.
- Öğrenci Numarası çift olanlar Bay (E) tek olanlar ise Bayan (B) olacak şekilde güncelleme komutları yazalım.
- Öğrenciler tablosundan sorgu çekerek öğrencilerin cinsiyetlerini detaylıca gösterelim.
- Öğrenci Numrası 7'nin katı ise öğreninin adı ve soyadı'ını birlikte gösterebiliriz.
- Öğrenci numrası 11'in katı ise sadece öğrencinin soyadını gösterebiliriz
- diğer durumlarda öğreninin adını gösterebiliriz
- İlgili Örneği T-SQL'de kodlayalım.

T-SQL Örnek 4 Veritabanı

10.200.64.13 - Uzak Masaüstü Bağlantısı

DESKTOP-ANV7JLU\SQLEXPRESS.VeriTabaniProgramLama - dbo.Ogrenciler - Microsoft SQL Server Management Studio

File Edit View Project Debug Table Designer Tools Window Help

New Query

Object Explorer

Connect

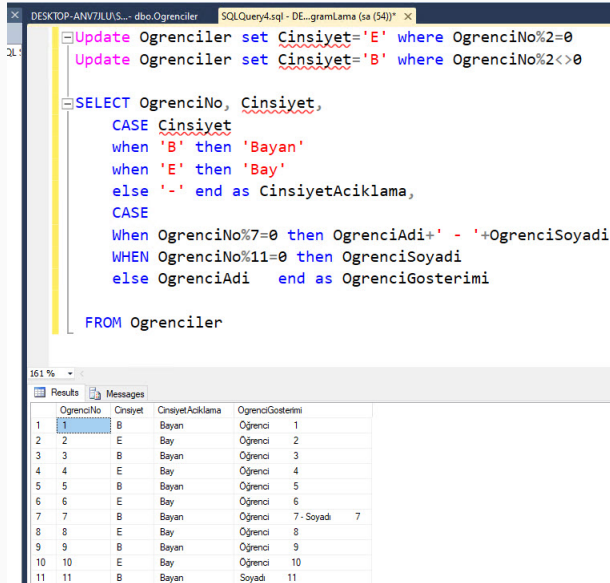
DESKTOP-ANV7JLU\SQLEXPRESS (SQL Server)

- Databases
 - System Databases
 - VeriTabaniProgramLama
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - dbo.Ogrenciler
 - Columns
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - Views
 - Synonyms
 - Programmability
 - Service Broker
 - Storage

DESKTOP-ANV7JLU\S...- dbo.Ogrenciler SQLQuery4.sql - DE...gramLama (sa)

Column Name	Data Type	Allow Nulls
OgrenciNo	int	<input checked="" type="checkbox"/>
OgrenciAdi	varchar(50)	<input checked="" type="checkbox"/>
OgrenciSoyadi	varchar(50)	<input checked="" type="checkbox"/>
Cinsiyet	varchar(1)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

T-SQL Örnek 4 Kod



The screenshot displays a SQL Server Enterprise Manager interface. At the top, a query window titled 'SQLQuery4.sql - DE...gramLama (sa (54))' contains the following T-SQL code:

```
Update Ogrenciler set Cinsiyet='E' where OgrenciNo%2=0
Update Ogrenciler set Cinsiyet='B' where OgrenciNo%2<>0

SELECT OgrenciNo, Cinsiyet,
CASE Cinsiyet
when 'B' then 'Bayan'
when 'E' then 'Bay'
else '-' end as CinsiyetAciklama,
CASE
When OgrenciNo%7=0 then OgrenciAdi+' - '+OgrenciSoyadi
WHEN OgrenciNo%11=0 then OgrenciSoyadi
else OgrenciAdi end as OgrenciGosterimi
FROM Ogrenciler
```

Below the query window, the 'Results' tab shows the output of the query. The results are displayed in a table with the following columns: 'OgrenciNo', 'Cinsiyet', 'CinsiyetAciklama', and 'OgrenciGosterimi'. The table contains 11 rows of data, with the first row highlighted.

OgrenciNo	Cinsiyet	CinsiyetAciklama	OgrenciGosterimi
1	B	Bayan	Oğrenci 1
2	E	Bay	Oğrenci 2
3	B	Bayan	Oğrenci 3
4	E	Bay	Oğrenci 4
5	B	Bayan	Oğrenci 5
6	E	Bay	Oğrenci 6
7	B	Bayan	Oğrenci 7 - Soyadı 7
8	E	Bay	Oğrenci 8
9	B	Bayan	Oğrenci 9
10	E	Bay	Oğrenci 10
11	B	Bayan	Soyadı 11