

Görsel Programlama

Ders 9

Dr. Öğr. Üyesi Mehmet Dinçer Erbaş
Bolu Abant İzzet Baysal Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Veritabanı sistemleri

- Veritabanı sistemleri, verinin organize edilmiş şekilde saklanmasına olanak verir.
- Bu sayede verinin saklanması ve sonradan ulaşılması kolay bir şekilde yapılır.
- Tüm uygulamaların veri işlemesi gerekir.
- Bu amaçla oluşturulmuş birçok yöntem ve veritabanı yönetim sistemi mevcuttur.
- Bu derste yapacağımız projelerde Microsoft Access veri tabanı kullanacağız.

Veritabanı sistemleri

- Table1 isminde aşağıda belirtilen yapıya sahip bir veritabanımız olduğunu varsayalım.

The screenshot displays the Microsoft Access application window. The title bar reads 'Database1: Database- E:\Gorsel Programlama\Database1.accdb (Access 2007 - 2016 file format) - Access'. The ribbon includes 'File', 'Home', 'Create', 'External Data', 'Database Tools', and 'Table Tools'. The 'Table Tools' ribbon is active, showing 'Design' and 'Table Tools' tabs. The 'Design' tab is selected, showing a 'Tell me what you want to do...' search bar and various tool icons. The 'Table Tools' ribbon is also visible, showing 'Field, Record & Table Events' and 'Relationships' sections. The 'All Access Objects' task pane on the left shows 'Tables' and 'Table1'. The main area displays the design view of 'Table1'. The table has four fields: 'OgrenciNo' (Number), 'Isim' (Short Text), 'Soyisim' (Short Text), and 'Bolum' (Short Text). The 'Field Name' column lists the fields, and the 'Data Type' column lists their respective data types. The 'Description (Optional)' column is empty. The 'Field Properties' pane at the bottom shows the 'General' tab with various properties for the selected field, including 'Field Size' (255), 'Format', 'Input Mask', 'Caption', 'Default Value', 'Validation Rule', 'Validation Text', 'Required' (No), 'Allow Zero Length' (Yes), 'Indexed' (No), 'Unicode Compression' (Yes), 'IME Mode' (No Control), 'IME Sentence Mode' (None), and 'Text Align' (General). A status bar at the bottom indicates 'Design view. F6 = Switch panes. F1 = Help.'

Field Name	Data Type	Description (Optional)
OgrenciNo	Number	
Isim	Short Text	
Soyisim	Short Text	
Bolum	Short Text	

Field Properties

General Lookup

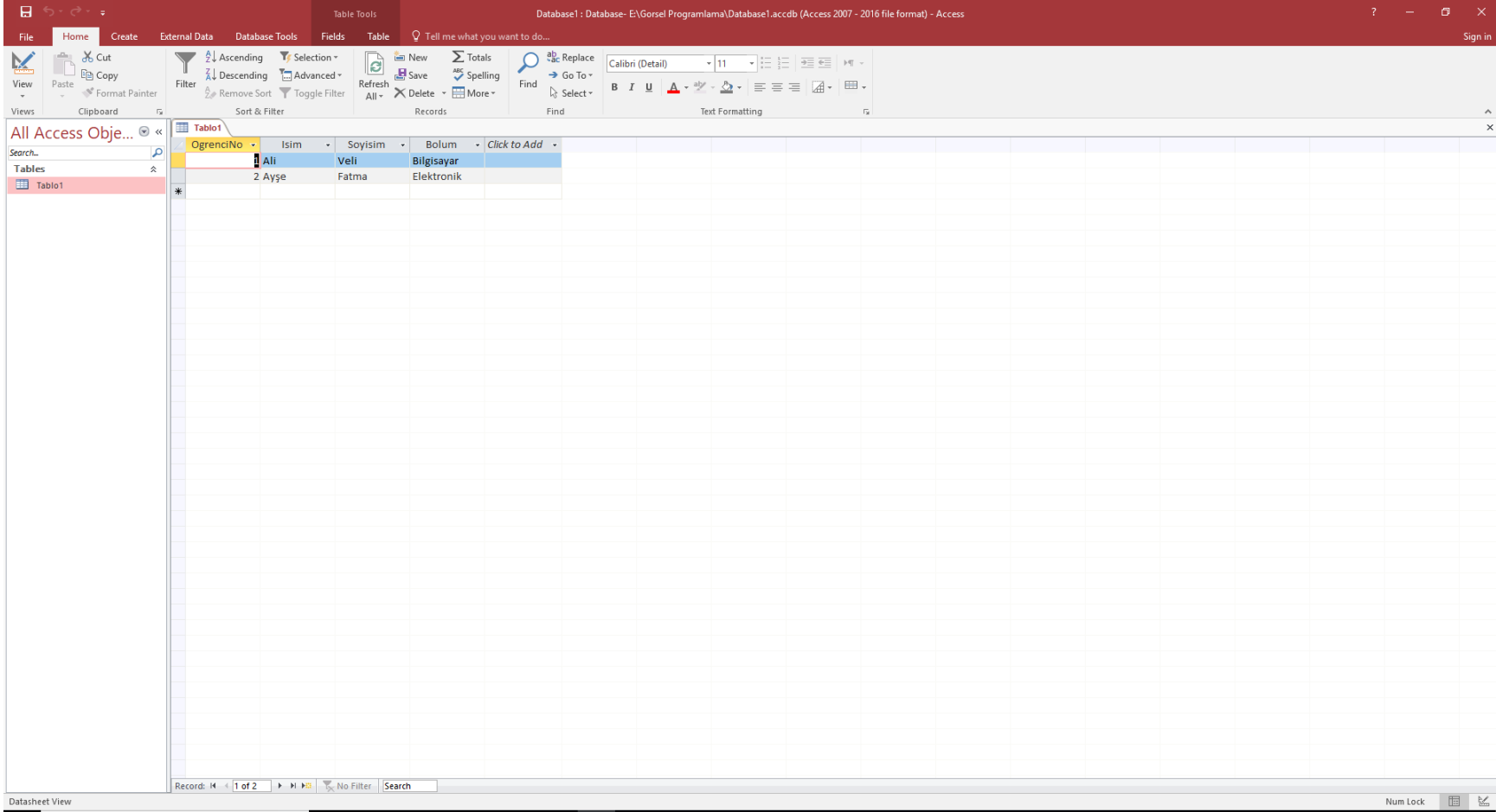
Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Text Align	General

The field description is optional. It helps you describe the field and is also displayed in the status bar when you select this field on a form. Press F1 for help on descriptions.

Design view. F6 = Switch panes. F1 = Help.

Veritabanı sistemleri

- Table1 isminde aşağıda belirtilen verileri içeren bir veritabanımız olduğunu varsayalım.



OgrenciNo	Isim	Soyisim	Bolum	Click to Add
2	Ali	Fatma	Elektronik	

Veritabanı sistemleri

- Projenizin konusu ile alakalı olarak aşağıda belirtilen işlemleri öğreneceğiz.
 - 📖 Bir veri tablosuna yeni veri ekleme
 - 📖 Bir veri tablosundaki verileri okuma
- Veritabanı işlemleri için aşağıda belirtilen kütüphane projemize eklenmelidir.
 - 📖 `using System.Data.OleDb;`
- Visual C# dilinde bir veritabanına erişebilmek için öncelikle bir bağlantı nesnesi yaratılmalıdır.
 - 📖 `System.Data.OleDb.OleDbConnection conn = new`
 - 📖 `System.Data.OleDb.OleDbConnection();`

Veritabanı sistemleri

- Veritabanına bağlantı sağlanabilmesi için gerekli bilgiler bağlantı nesnesine eklenmelidir.

```
 conn.ConnectionString =  
    @"Provider=Microsoft.ACE.OLEDB.12.0;" + @"Data source=  
    Database1.accdb";
```

- Veritabanı üzerinde herhangi bir işlem yapabilmemiz için öncelikle bağlantı açılmalıdır.

```
 conn.Open();
```

- Veritabanı ile bağlantı başarılı olarak açılmışsa artık yazma ve okuma işlemleri yapabiliriz.
- Bu amaçla bir SQL isteği metni oluşturmalıyız.

```
 SQL bir veritabanı yönetim dilidir (Structured Query Language)
```

Veritabanı sistemleri

- Ders kapsamında veritabanı üzerinde yapacağımız basit işlemler ve ilgili SQL komutları şu şekildedir.

📖 Okuma: Örneğin tabloda bulunan öğrencilerle ilgili bilgileri almak için

- `string sql = "SELECT * FROM Table1";`

📖 Sadece öğrencilerin isimlerini almak için

- `string sql = "SELECT Isim FROM Table1";`

📖 Sadece okulda bulunan departmanların listesini almak için

- `string sql = "SELECT Bolum FROM Table1";`

📖 Tabloda bulunan belli bir öğrencinin bilgilerini almak için

- `string sql = "SELECT * FROM Table1 WHERE Isim = 'Ali'";`

Veritabanı sistemleri

- SQL komutları

📖 Tabloda bulunan satırları saymak için

- `string sql = "SELECT COUNT(*) FROM Table1";`

📖 Belli bir bölümdeki öğrencileri saymak için

- `string sql = "SELECT COUNT(*) FROM Table1 WHERE Bolum = 'Makine'";`

📖 Veritabanına yeni bir satır eklemek için

- `string sql = "INSERT INTO Table1
(OgrenciNo,Isim,Soyisim,Bolum) VALUES
(4,'Ali','Veli','Bilgisayar')";`

📖 Veritabanından belli isme sahip öğrenciyi silmek için

- `string sql = "DELETE FROM Table1 WHERE Isim = 'Ali'";`

Veritabanı sistemleri

- SQL komutu oluşturulduktan sonra önceden oluşturduğumuz bağlantı ile SQL komutu kullanılarak veritabanı komutumuz oluşturulur.

 `OleDbCommand command = new OleDbCommand(sql, conn);`

- Okuma işlemleri için ayrıca bir okuyucu (reader) oluşturulmalıdır.

 `OleDbDataReader reader = null;`

- Daha sonra okuma komutu çalıştırılır.








 `reader = command.ExecuteReader();`

- İlgili satırların okunması sağlanır.

 `reader.Read();`

Veritabanı sistemleri

- Tablomuzdaki kayıtları okuyan ve öğrencilerin isimlerini yazan kod parçası


```
 string sql = "SELECT * FROM Table1";  
 OleDbDataReader reader = null;  
 OleDbCommand command = new OleDbCommand(sql, conn);  
 reader = command.ExecuteReader();  
 while (reader.Read())  
 {  
    • Console.WriteLine(reader[1].ToString() + " " +  
      reader[2].ToString());  
 }
```

Veritabanı sistemleri

- Yazma ve silme işlemleri için ilgili komutu gerçekleştirmemiz yeterlidir.

 `command.ExecuteNonQuery();`

- Yazılacak verinin her alanının belirtilmesi gerekmektedir.
- Tabloya yeni bir öğrenci ekleme işlemi şu şekilde yapılır:

 `sql = "INSERT INTO Table1 (OgrenciNo, Isim, Soyisim, Bolum)
VALUES (5, 'Ali', 'Veli', 'Bilgisayar')";`

 `command = new OleDbCommand(sql, conn);`

 `command.ExecuteNonQuery();`

- Veritabanı üzerinde işlemler tamamlandıktan sonra veritabanı kapatılmalıdır.

 `conn.Close();`