

Veritabanı ve Yönetim Sistemleri

Öğr. Gör. M. Mutlu YAPICI

Ankara Üniversitesi
Elmadağ Meslek Yüksekokulu

Ders İzlenesi

Hafta	Modüller/İçerik/Konular
1. Hafta	Temel Kavramlar
2. Hafta	Veri Modelleri
3. Hafta	
4. Hafta	
5. Hafta	
6. Hafta	
7. Hafta	
8. Hafta	
9. Hafta	
10. Hafta	
11. Hafta	
12. Hafta	
13. Hafta	
14. Hafta	

Visual Studio ve SQL Bağlantısı

Veri taban işlemlerini SQL serverda nasıl yapacağımızı gördük. Yeni bir veri tabanı yaratıp tablolar ve bu tablolara da nitelikler ekleyerek veri tabanımızı oluşturabiliyoruz. VTYS ni kullanarak veri tabanımıza yeni veriler ekleyip bu verilerle ilgili işlemler de gerçekleştirebiliyoruz.

Şimdiye kadar tüm bu işlemleri VTYS kullanarak gerçekleştirdik. Şimdi de bir masa üstü uygulamasından bu veri tabanına bağlantıyı nasıl yaparız bir form uygulaması yaparak öğrenelim.

Visual Studio ve SQL Bağlantısı

Uygulamayı Visual Studio Windows Form uygulamasında yapacağız. Uygulamada kullanacağımız veri tabanı daha önce yarattığımız okul veri tabanı olabilir. Eğer böyle bir veri tabanınız yoksa, okul adında bir veri tabanı yaratıp altına bölümler ve öğrenciler isimli iki adet tablo ekleyin ve 1-n ilişkisi olacak şekilde veri tabanınızı oluşturun. Altına birkaç bölüm ve öğrenci ekleyebilirsiniz. İki tablonun da niteliklerini siz belirleyin.

Visual Studio ve SQL Bağlantısı

Benim kullanacağım veri tabanı görüntüsü aşağıdaki gibidir.

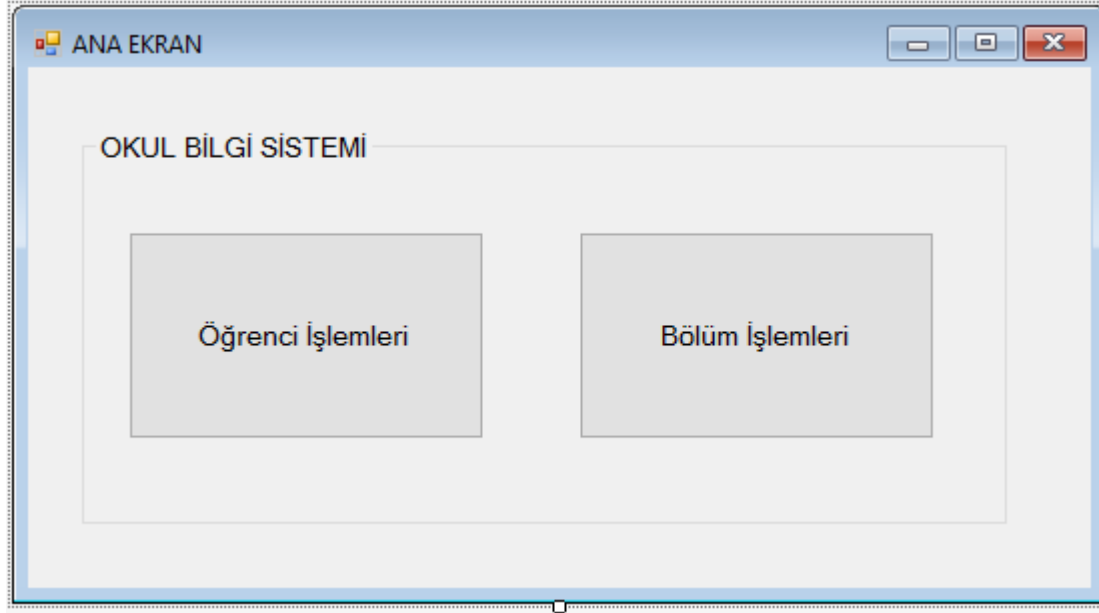
The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the server 'DESKTOP-5243N4C\SQLEXPRESS' with a database named 'UEOku'. Under 'Tables', the 'dbo.bolum' and 'dbo.ogrenci' tables are highlighted with a red box. On the right, the 'Results' pane shows two query results, both highlighted with red boxes. The top result is a table with columns 'tcno', 'adi', 'tel', 'cins', and 'fr_BNO'. The bottom result is a table with columns 'bno', 'adi', 'adres', and 'tel'.

	tcno	adi	tel	cins	fr_BNO
1	1234	Mutlu YAPICI	535	Bay	1
2	2345	Bill GATES	444	Bay	1
3	3344	Sharon Stone	333	Bayan	3
4	34445	Donald TRUMP	33	Bay	4

	bno	adi	adres	tel
1	1	Bilgisayar	EMYO	0312
2	3	Muhasebe	EMYO	NULL
3	4	Büro Yönetimi ve Sekreterlik	NULL	NULL

Visual Studio ve SQL Bağlantısı

Şimdi visual studio da bir Windows form uygulaması açın ve aşağıdaki gibi tasarımı yapın. Öğrenci İşlemleri butonundan öğrenciyi ekleyip, silip güncellediğimiz, Bölüm İşlemleri butonundan da bölümleri ekleyip, silip güncellediğimiz, işlemleri yapacağız.



Visual Studio ve SQL Bağlantısı

Öncelikle Bölüm İşlemleri Formunu yapalım. Projenize yeni bir form ekleyin ve tasarımını aşağıdaki gibi yapın. Bu form Ana Ekrandaki Bölüm İşlemleri Butonuna tıklayınca açılacak.

The screenshot shows a Windows form titled "BÖLÜM YÖNETİM EKRANI". Inside the form, there is a "GROUP BOX" containing three textboxes labeled "Bölüm Adı", "Bölüm Adresi", and "Bölüm Telefonu". A red arrow points from the text "TEXTBOX" to the three textboxes. Below the textboxes is a large empty rectangular area labeled "LIST VIEW". At the bottom of the form, there are four buttons: "Verileri Göster", "Bölüm Ekle", "Seçili Bölümü Güncelle", and "Seçili Bölümü Sil".

Visual Studio ve SQL Bağlantısı

Bölüm Yönetim formunun Ana Ekrandaki Bölüm İşlemleri Butonuna tıklayınca açılması için aşağıdaki komutları butonun click olayına yazmalısınız.

```
private void button2_Click(object sender, EventArgs e){  
    Form f = new bolum();  
    f.ShowDialog();  
}
```


Visual Studio ve SQL Bağlantısı

Bölüm Formunda öncelikle verileri veri tabanından alıp list view de gösterme işlemlerini yapalım.

Veri tabanı bağlantı işlemi için `SqlConnection` sınıfından bir bağlantı nesnesi yaratıp tüm bağlantılarda bu nesneyi kullanmalıyız. `SqlConnection` Nesnesinin kullanımı aşağıdaki gibidir.

```
SqlConnection baglanti = new SqlConnection(  
@"server =.\SQLEXPRESS; //Sunucu Adresi  
Initial Catalog = UEokul; //Veri tabanı Adı  
Integrated Security = SSPI" //Güvenlik kodu  
);
```

Bu nesneyi sürekli kullanacağımız için global olarak tanımladık.

Visual Studio ve SQL Bağlantısı

Önceki sunumda oluşturduğumuz SQL bağlantısı ile veri tabanından verileri alıp listelemenin birkaç farklı yolu vardır. Bunlardan en çok kullanılan ikisi `SqlDataReader` ve `SqlDataAdapter` dir. Önce `SqlDataReader` ile bu işlemin nasıl olduğuna bakalım. Her iki durumda da öncelikle

- veri tabanı bağlantımızı açacağız,
- sonra veri tabanına bir SQL sorgusu göndereceğiz yani SQL komutu göndereceğiz
- ve sonra da gelen verileri alıp ekranda istediğimiz nesne içerisinde göstereceğiz.

Visual Studio ve SQL Bağlantısı

Veri tabanına komut gönderebilmek için `SqlCommand` nesnesine ihtiyacımız olacak. Bu nesneyi sürekli kullanacağımız için bunu da global tanımlayalım. Daha sonra gelen verileri okuyabilmemiz için de `SqlDataReader` nesnesine ihtiyacımız olacaktır.

Okuma ve verileri alıp list view de yazdırma işlemlerini `private void` `BolumVerileriniOku()` isimli bir fonksiyon içerisinde yapalım.

Fonksiyona ait tüm kodlar bir sonraki sunumdadır.

Visual Studio ve SQL Bağlantısı

```
o private void BolumVerileriniOku(){
    baglanti.Open(); //önce veri tabanı bağlantısını açalım
    komut = new SqlCommand();//SQL Komut nesnesini oluşturalım

    ///komutun çalışacağı bağlantıyı tanımlayalım
    komut.Connection = baglanti;

    ///Göndereceğimiz SQL komutunu yazalım
    komut.CommandText = "SELECT * FROM bolum";

    ///komutu çalıştırıp sonuçları alalım
    SqlDataReader veriOkuyucu = komut.ExecuteReader();

    ////Şimdi de sonuçları List View İçerisine yazalım
    BU Kısım Bir Sonraki Sunumda onu Buraya Yazın

}
```

Visual Studio ve SQL Bağlantısı

- List View nesnesinin kullanımı için sonraki beş sunum sayfasına bakın.

- `private void` BolumVerileriniOku()**(DEVAMI)**

////Şimdi de sonuçları List View İçerisine yazalım

```
listView1.Columns.Add("BNO", 50, HorizontalAlignment.Center);  
listView1.Columns.Add("ADI", 150, HorizontalAlignment.Center);  
listView1.Columns.Add("ADRES", 250, HorizontalAlignment.Center);  
listView1.Columns.Add("TELEFON", 100, HorizontalAlignment.Center);
```

```
while (veriOkuyucu.Read())  
{
```

```
    int elemanSayisi = listView1.Items.Count;  
    listView1.Items.Add(veriOkuyucu["bno"].ToString());
```

```
listView1.Items[elemanSayisi].SubItems.Add(veriOkuyucu["adi"].ToString());
```

```
listView1.Items[elemanSayisi].SubItems.Add(veriOkuyucu["adres"].ToString());
```

```
listView1.Items[elemanSayisi].SubItems.Add(veriOkuyucu["tel"].ToString());
```

```
}
```

```
    baglanti.Close();////bağlantıyı kapatalım
```

```
}
```

List View Kullanımı

ListView en çok kullanılan kontrollerden biridir. ListBox ile benzer olmak ile birlikte, ListBox'un gelişmiş halide denebilir. Bu kontrol birçok farklı amaç için kullanışlıdır. Dizindeki klasörleri listeleyebilir veya veritabanı işlemlerini halledebiliriz. Birçok amaca hizmet eder. Üzerindeki ok işaretinden açılan sekmeden yeni sütunlar ve içerisine yeni elemanlar ekleyebiliriz.

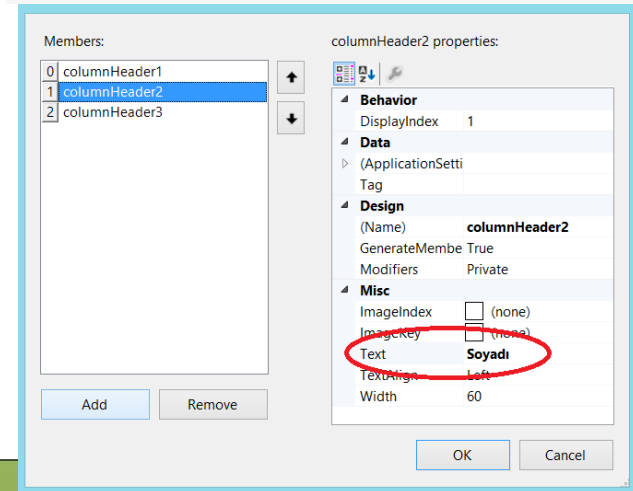
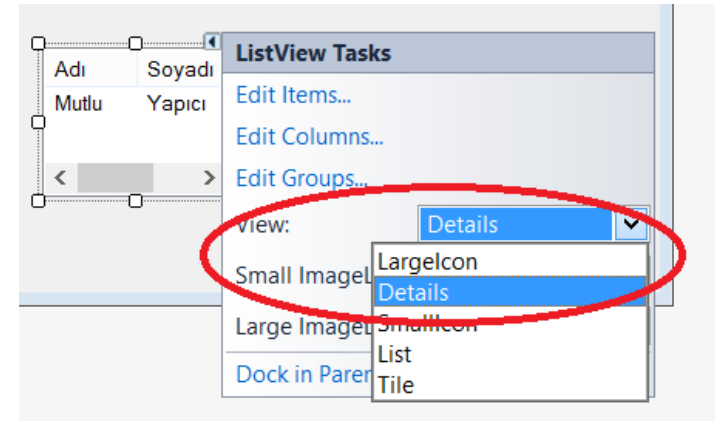
EditColumns sekmesinden yeni sütunlar ve bu sütunlara ait özellikleri ayarlayabiliriz. Aynı şekilde EditItems sekmesinden yeni elemanları ekleyebiliriz.

Tabi bu işlemlerin tamamını kod ile de yapabiliriz.

NOT: ListView de tüm sütunların görünebilmesi için yanda gösterilen View Listesinden **Details** seçilmelidir.

.Bu işlemi kod ile şu şekilde de gerçekleştirebiliriz.

```
listView1.View = View.Details;
```



List View Kullanımı

ListView'e eklediğimiz ilk elemanlar ana eleman oluyor yani ilk sütun ana eleman oluyor diğer sütunları eklemek için aşağıda gösterildiği gibi subItem sekmesine tıklayarak o satırdaki ana elemanın alt elemanlarını eklemelisiniz. Örneğin Ana Eleman Mutlu alt elemanlar Yapıcı ve 12345 gibi.

The screenshot displays two windows from Visual Studio. The 'ListViewItem Collection Editor' window shows a list of items with one item, '0 ListViewItem: {Mutlu}'. The 'ListViewSubItem Collection Editor' window shows a list of sub-items with two items: '0 ListViewItem: {Yapıcı}' and '1 ListViewItem: {12345}'. The 'SubItems' property in the 'ListViewItem' editor is circled in red. The 'Yapıcı' sub-item in the 'ListViewSubItem' editor is also circled in red.

Adı	Soyadı	TC No
Mutlu	Yapıcı	12345

List View Kullanımı

ListViewe kod ile yeni bir sütun eklemek için;

```
//Telefon ve Maaş isimlerinde, 70 genişliğinde, sola yatık şekilde 2 tane sütun(column)ekler.
```

```
listView1.Columns.Add("Telefon", 70, HorizontalAlignment.Center);  
listView1.Columns.Add("Maaş", 70, HorizontalAlignment.Right);
```

Tüm elemanlar listede 0. indexten başlarlar yani her satır bir indexle numaralandırılır. Eğer bir satırı silmek veya verilerini değiştirmek istersek bu index numarası ile o satıra ulaşabiliriz.

Adı	Soyadı	TC No	Telefon	Maaş
Mutlu	Yapıcı	12345		

Yine listViewe yeni bir veri eklemek istersek en son indexe eklemeliyiz bu sebeple veri eklemeden önce listede kaç eleman var sayısını almalıyız. Böylece bu bizim son ekleyeceğimiz değer index numarası olacaktır. Listedeki eleman sayısını şu şekilde alabiliriz: `int sıra = listView1.Items.Count;`

Sıra numarasını bildiğimize göre yeni elemanı ekleyebiliriz.

```
//ilk eklediğimiz ana itemdir, ilk sütundan sonra gelen sütunları doldurabilmek için altitem(subitem) kullanırız.
```

```
listView1.Items.Add("Mehmet");  
listView1.Items[sıra].SubItems.Add("Parlak");  
listView1.Items[sıra].SubItems.Add("123443");  
listView1.Items[sıra].SubItems.Add("312 312 12 12");  
listView1.Items[sıra].SubItems.Add("4000 TL");
```

Adı	Soyadı	TC No	Telefon	Maaş
Mutlu	Yapıcı	12345		
Mehmet	Parlak	123443	312 312 ...	4000 TL

List View Kullanımı

ListViewden seçili elemanı silmek için remove veya removeAt metodları kullanılabilir.

İndeksi belli olan seçili elemanı silmek için

```
listView1.SelectedItems[0].Remove(); // Yine yukarıda anlattığımız  
//gibi tek bir öge silmek istediğimiz için bu işlemiyapıyoruz.
```

Tüm seçili elemanları silmek için

```
foreach (ListViewItem item in ListView1.SelectedItems)  
{  
    item.Remove();  
}
```

Tüm listeyi silmek için listView1.Items.Clear(); komutu kullanılır.

Listview de dikkat ederseniz sadece il sütun seçiliyor. Tüm satırı seçmek için FullRowSelect Özelliği True olmalıdır.

```
listView1.FullRowSelect = true;
```

Adı	Soyadı	TC No	Telefon	Maaş
Mutlu	Yapıcı	12345		
Mehmet	Parlak	123443	312 312 ...	4000 TL

Adı	Soyadı	TC No	Telefon	Maaş
Mutlu	Yapıcı	12345		
Mehmet	Parlak	123443	312 312 ...	4000 TL

List View Kullanımı

ListViewde çok satır seçimi yapabilmek için MultiSelect özelliğini true yapmalıyız.

```
listView1.MultiSelect = false; //Birdeb fazla satır seçilemesin diye.
```

Yine listViewde grid dediğimiz çizgilerin görünmesini istiyorsak GridLines özelliğini true yapmalıyız

```
listView1. GridLines = true;
```

ListViewlerde eğer seçim yaptıktan sonra başka bir araca geçerse yani listView üzerinden ayrılırsak seçim kaybolur. Kaybolmasın yine de seçtiğim yer gri bir renkte seçili kalsın istiyorsak HideSelection özelliğini false yaparak kapatmalıyız.

```
listView.HideSelection = false; // !!! Normalde listview seçili satırdan (nesneden) ayrıldığınız zaman seçim olayı kalkar. Biz bunun önüne geçmek için bu özelliği kullandık. Artık nesneden ayrılısanızda arkada gri bir renk bırakır.
```

Visual Studio ve SQL Bağlantısı

Şimdi de Bölüm Formumuzdaki ikinci buton olan Bölüm Ekle işlemlerini yapalım.(Burada gelen veriler boş mu diye kontrol yapmalısınız)

```
private void button2_Click(object sender, EventArgs e){
    baglanti.Open();///Bağlantıyı açalım
    //SQL Komut nesnesini oluşturalım
    komut = new SqlCommand("INSERT INTO bolum
                            (adi,adres,tel)
                            VALUES('"+textBox1.Text+ "', '" +
                                textBox2.Text + "', '" +
                                textBox3.Text + "')",
                            baglanti);
    komut.ExecuteNonQuery(); ///Komutu Çalıştıralım
    baglanti.Close();///bağlantıyı kapatalım

    BolumVerileriniOku(); ///Verileri Tekrar Gösterelim
}
```

Visual Studio ve SQL Bağlantısı

Şimdi de Bölüm Formumuzdaki üçüncü buton olan Bölümü Güncelle işlemlerini yapalım.(Burada gelen veriler boş mu diye kontrol yapmalısınız)

```
private void button3_Click(object sender, EventArgs e){  
    ///güncelleme işleminde öncelikle seçili elemanın BNO  
    bilgisini almalıyız  
    int bno =  
    Convert.ToInt32(listView1.SelectedItems[0].Text);  
    baglanti.Open();///Bağlantıyı açalım  
    //SQL Komut nesnesini oluşturalım  
    komut = new SqlCommand("UPDATE bolum SET adi='" +  
    textBox1.Text + "',adres='" + textBox2.Text + "',tel='" +  
    textBox3.Text + "' WHERE bno="+bno+"", baglanti);  
    komut.ExecuteNonQuery();  
    baglanti.Close();///bağlantıyı kapatalım  
  
    BolumVerileriniOku();  
}
```

Visual Studio ve SQL Bağlantısı

Şimdi de Bölüm Formumuzdaki dördüncü buton olan Bölümü Sil işlemlerini yapalım.(Burada gelen veriler boş mu diye kontrol yapmalısınız)

```
private void button4_Click(object sender, EventArgs e){  
    ///Silinecek seçili elemanın BNO bilgisini almalıyız  
    int bno = Convert.ToInt32(listView1.SelectedItems[0].Text);  
  
    baglanti.Open();///Bağlantıyı açalım  
    //SQL Komut nesnesini oluşturalım  
    komut = new SqlCommand("DELETE FROM bolum WHERE  
bno=" + bno + "", baglanti);  
    komut.ExecuteNonQuery();  
    baglanti.Close();///bağlantıyı kapatalım  
  
    BolumVerileriniOku();  
}
```

Visual Studio ve SQL Bağlantısı

Veri tabanından verileri alıp listeleme işlemini `SqlDataReader` ile yaptık şimdi de öğrenci formu için bu listeleme işlemini `SqlDataAdapter` ile yapalım.

Önce Öğrenci İşlemleri İçin de Formumuzu tasarlayalım. Bölüm ile aynı formatta tasarlayabilirsiniz. Bir sonraki sunumda tasarım görünmektedir. Burada listeleme için List View elemanı yerine DataGrid View elemanını kullandığımızı dikkat edin.

ANA EKРАН daki Öğrenci İşlemlerine Tıklayınca ÖĞRENCİ formu açılacak.

Visual Studio ve SQL Bağlantısı

ÖĞRENCİ İŞLEMLERİ EKRANI

GROUP BOX

ÖĞRENCİ İŞLEMLERİ

TC No

Öğrenci Adı

Öğrenci Telefonu

Cinsiyet

COMBO BOX

Bölümü

COMBO BOX



TEXTBOX

DATA GRID VIEW

Öğrencileri Göster

Öğrenci Ekle

Seçili Öğrenciyi
Güncelle

Seçili Öğrenciyi Sil

Visual Studio ve SQL Bağlantısı

`SqlDataAdapter` ile verileri listelemek için yine öncelikle bir bağlantı başlatmalıyız. Buraya kadar `SqlDataReader` ile aynı, farklılık SQL komutunu gönderip, verileri alıp yazdırırken ortaya çıkmaktadır.

Normalde SQL komutunu `SqlCommand` ile gönderirken `SqlDataAdapter` kullanırken `SqlDataAdapter` ile göndereceğiz. Bunun için bir `SqlDataAdapter` nesnesi yaratmalıyız.

Verileri `DataGridView` e yüklemek içinse `DataSet` nesnesini kullanacağız.

Visual Studio ve SQL Bağlantısı

Öğrenci listeleme fonksiyonumuz aşağıdadır.

```
SqlConnection baglanti = new SqlConnection(
@"server =.\SQLEXPRESS; //Sunucu Adresi
Initial Catalog = UEokul; //Veri tabanı Adı
Integrated Security = SSPI" //Güvenlik kodu
);

    SqlCommand komut;
private void OgrenciBilgileriniListele()
{
    baglanti.Open(); ///önce veri tabanı bağlantısını açalım
    SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM
ogrenci",baglanti); ///Göndereceğimiz SQL komutunu yazalım
    DataSet ds = new DataSet();///data set nesnemizi oluşturalım
    da.Fill(ds, "ogrenci");///dataseti veri ile dolduralım
    dataGridView1.DataSource = ds.Tables["ogrenci"];

    baglanti.Close();///bağlantıyı kapatalım
}
```

Visual Studio ve SQL Bağlantısı

Veri ekleme güncelleme ve silme işlemleri bölümde olduğu gibi öğrencide de aynı yani sqlCommand ile komutu ayarlayıp ExecuteNonQuery() fonksiyonu ile göndereceksiniz. Bu sebeple aynı komutları tekrar düzenlemedim.

Eğer Takıldığınız yer olursa sorabilirsiniz. Güncelleme ve silme işleminde dataGridView içinden seçili öğrencinin tcno bilgisini almalısınız bunun için aşağıdaki komutu kullanabilirsiniz

```
string tcno = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();
```

SQL - Komutlar

- INSERT
- UPDATE
- DELETE
- ALTER TABLE
- DROP TABLE
- CREATE VIEW
- ALTER VIEW

SQL - INSERT Komutu

- Tabloya yeni kayıt eklemek için INSERT komutu kullanılır,
- INSERT INTO tablo_adi(alan listesi) values (yenikayıtlar verileri)

SQL - INSERT Komutu

- Egitselyazilim veri tabanına yeni bir kullanıcı eklemek için;
- `INSERT INTO kullanıcı(kul_adi, adi, soyadi,sifre) values ('ayse','Ayşe', 'Ak', 111);`

SQL – UPDATE Komutu

- Tablonun bir veya birden fazla kolonundaki veriyi yine tablo üzerinde güncelleyerek değiştirmek için UPDATE komutu kullanılır,
- UPDATE tabloadi SET kolon1=deger1, kolon2=deger2;

SQL – UPDATE Komutu

- Egitselyazilim veri tabanındaki Ayşe isimli kullanıcının kullanıcı adını değiştirmek için;
- UPDATE kullanıcı SET kul_adi=ayse_bote WHERE adi='Ayşe';

SQL – DELETE Komutu

- Tabloda bulunan kayıt ya da kayıtları silmek için kullanılır;
- `DELETE FROM tablo_adi WHERE alanadi='Ayşe';`

SQL – DELETE Komutu

- Egitselyazilim veri tabanındaki Ayşe isimli kullanıcıyı silmek için;
- `DELETE FROM kullanici WHERE adi='Ayşe';`

SQL – ALTER TABLE Komutu

- Tabloya alan ekleme, varolan alanı düzenleme ve silmek için ALTER TABLE komutu kullanılır,
- ALTER TABLE tablo_adi ADD alan_adi alan_turu;
- ALTER TABLE tablo_adi MODIFY alan_adi alan_turu;
- ALTER TABLE tablo_adi DROP COLUMN alan_adi;

SQL – ALTER TABLE Komutu

- Egitselyazilim veri tabanındaki kullanıcı tablosuna 50 karakterlik metin girişine imkân sağlayan e_posta isimli bir alan eklemek için;
- ALTER TABLE kullanıcı ADD e_posta VARCHAR(50),

SQL – ALTER TABLE Komutu

- Egitselyazilim veri tabanındaki kullanıcı tablosuna birden fazla alan eklemek için;
- ALTER TABLE kullanıcı ADD (e_posta VARCHAR(50), yas INT (3), yer VARCHAR(80));

SQL – ALTER TABLE Komutu

- Egitselyazilim veri tabanındaki kullanıcı tablosundaki yer alanının veri türüyle ilgili değişiklik yapmak için;
- ALTER TABLE kullanıcı MODIFY yer VARCHAR(100);

SQL – ALTER TABLE Komutu

- Egitselyazilim veri tabanındaki kullanıcı tablosundaki yer alanı kaldırmak için;
- ALTER TABLE kullanıcı DROP COLUMN yer;

SQL – DROP TABLE Komutu

- Veritabanında yer alan bir tabloyu kaldırmak için;
- DROP TABLE tabloadi;

SQL – DROP TABLE Komutu

- Egitselyazilim veri tabanındaki kullanıcı tablosunu kaldırmak için;
- DROP TABLE kullanıcı;

SQL – CREATE VIEW Komutu

- View oluşturmak için;
- CREATE view view_adi AS
SELECT alanadi,alanadi2 FROM tabloadi WH
ERE kosul;

SQL – CREATE VIEW Komutu

- Egitselyazilim veri tabanındaki kullanıcı tablosundaki Ahmet isimli kullanıcıların ad ve soyadları listeleyen bir view oluşturmak için;
- `CREATE view kul_adsoyad AS
SELECT adi,soyadi FROM kullanıcı WHERE
adi='Ahmet';`

SQL – ALTER VIEW Komutu

- View oluşturmak için;
- ALTER view view_adi AS
SELECT alanadi FROM tabloadi WHERE kos
ul;

SQL – CREATE VIEW Komutu

- Egitselyazilimdaki kul_adsoyad view'ını değiştirerek, veri tabanındaki tüm kullanıcıların ad ve soyadları listelemesi için;
- ALTER view kul_adsoyad AS
SELECT adi,soyadi FROM kullanıcı;

KAYNAKLAR

- Yrd. Doç. Dr. Altan MESUT
<http://altanmesut.trakya.edu.tr/vtys1/>
- Öğr. Gör. Dr. Sibel SOMYÜREK
http://sibelsomyurek.com/veritabani/ders_notlari.html
- Tokdemir, G. ve Çağıltay, N. E. (2010). *Veritabanı Sistemleri Dersi*. Seçkin yayıncılık, Ankara.