

VERİ TABANI NESNELERİ

CONNECTION NESNESİ :	Bağlantı nesnesidir. Veritabanına bağlantı kurma (Open()), açık olan bağlantıyı kapatma (Close()) fonksiyonları olan ayrıca DataAdaptor ve Command nesnelere tarafından kullanılan nesnedir.
Üyeleri :	
*ConnectionString:	Türü string'dir. Connection nesnesinin hangi aracı (Provider) ile hangi veritabanına, hangi kullanıcı, hangi şifre ile bağlanacağı bilgisinin olduğu metin türünden özelliğidir.
*Open() :	Open fonksiyonu Connection nesnesinin veritabanına bağlantı kurmasını sağlar. Bir hata var ise hata(Exception) fırlatır.
*Close() :	Close fonksiyonu Connection nesnesinin bağlı olduğu veritabanından bağlantıyı kesmesini sağlar. Bir hata var ise hata(Exception) fırlatır.

DATAADAPTOR NESNESİ :	Select sorgusu çalıştırmak için gerekli nesnedir. Sorgu sonucunu Fill metodu ile belirtilen (parametresi) yere doldurur.
Üyeleri :	
*Connection:	Dataadaptor nesnesinin çalışabilmesi için bir connection kullanması (*open) gerekir. Türü connection'dır.
*CommandText :	Dataadaptor nesnesinin çalıştıracağı sorgu cümlesi özelliğidir. Türü string'dir.
*Fill() :	Verilen sorgu çalıştırma fonksiyonudur. Parametre olarak verilen DATATABLE nesnesi sorgu sonucu dönen tablo halindeki verileri doldurur. *Not: Ekleme şeklinde doldurur. Gerekirse önceden DATATABLE temizlemek gerekir.

COMMAND NESNESİ :	Insert, Update, Delete sorgularını çalıştırmak için gerekli nesnedir.
Üyeleri :	
*Connection:	Command nesnesinin çalışabilmesi için bir connection kullanması (*open) gerekir. Türü connection'dır.

*CommandText :	Command nesnesinin çalıştıracağı sorgu cümlesi özelliğidir. Türü string'dir.
*ExecuteNonQuery() :	Verilen sorgu çalıştırma fonksiyonudur. Sorgu çalıştırdıktan sonra tabloda ya da tablolarda işlem gören satır sayısını (int) sayı olarak dönderir.

DATATABLE NESNESİ :	SELECT sorguları sonucu tablo halinde dönen verileri saklamak için kullanılan nesnedir.
----------------------------	---

DATASET NESNESİ :	<p>İçerisinde birden fazla DATATABLE nesnesi saklayabilen nesnedir. İçerisindeki DATATABLE'lara örnekteki gibi erişilir.</p> <p>Örnek:</p> <pre>DataSet ds = new Dataset(); DataTable dt1 =new DataTable(); DataTable dt2 =new DataTable(); ds.Tables.Add(dt1); //dt1 'i ds datasetine ekledim. ds.Tables.Add(dt2); //dt2 'i ds datasetine ekledim. Da.Fill(ds.Tables["dt1"]); //Fill metodu verileri dt1 e doldurur. Da.Fill(ds.Tables[0]); //Fill metodu verileri dt1 e doldurur.</pre> <p>Not: dt1 ds'ye ilk eklene Datatable olduğu için indeks (sıra numarası) 0 dir. dt Datatable'ına indeks ilede erişilebilir.</p>
--------------------------	--

SELECT Sorgusu	
SELECT * FROM tabloadı	
ORDER BY alan ASC yada DESC //ASC olursa küçükten büyüğe, DESC büyükten küçüğe sıralar	
WHERE şehir ="KONYA" //Tüm kayıtları değil şehiri KONYA olan kayıtları getirir	
GROUP BY şehir //Kayıtlarda SUM,AVG, COUNT gibi fonksiyonlar için gruplama yapar.	
<p>Örn: SELECT COUNT(şehir) FROM tablo_adi GROUP BY şehir = "KONYA"</p> <p>Sorgu şehri konya olan kaç kişi olduğunu verir.</p> <p>COUNT : Adet verir</p> <p>SUM : Toplam verir</p> <p>AVG : Ortalama verir.</p>	

`SELECT` departman, `AVG`(maas) `FROM` maaslar `GROUP BY` Departman

Sorgu her departmanın maaş ortalamasını verir.

INSERT Sorgusu

`INSERT INTO` tablo_adi(alan1,alan2,...) `VALUES`(değer1,değer2,..)

UPDATE Sorgusu

`UPDATE` tablo_adi `SET` alan1=değer1,alan2=değer2,.. `WHERE`

DELETE Sorgusu

`DELETE FROM` tablo adı `WHERE`kriter