Delphi menüleri

File menüsü:

New: Pencerenin new kısmında bulunan bazı seçenekler şunlardır:

Form seçeneği ile aktif projeye yeni bir form eklenir.

Application seçeneği aktif projeyi kapatarak yeni bir projenin başlatılmasını sağlar. Bu seçenek standart windows programını oluşturmak için kullanılır.

Component seçeneği ile delphi component dosyası oluşturulabilir.

DLL seçeneği ile DLL dosyaları gibi kütüphane dosyaları oluşturmak için kullanılır.

Text seçeneği ile yeni bir text dosyası eklenebilir. Bu dosyalar kod değil metin içerirler.

Unit seçeneği projeye yeni bir unit ekler.

ActiveX:

Active Form seçeneği ile active formlar oluşturmak için kullanılır.
Oluşturulan bu formlar web tarayıcılar tarafından çalıştırılabilir programlardır.

ActiveX Control seçeneği yeni activeX controller oluşturmak için kullanılır. Bu seçenek ile birlikte activex sihirbazları devreye girerek atılacak adımlar için programcıya yardımcı olurlar.

ActiveX Library seçeneği yeni activex kütüphaneleri oluşturmak için kullanılır.

Automation Object seçeneği yeni OLE elemanları oluşturmak için kullanılır. Bu seçenek ile birlikte OLE otomasyon sihirbazları devreye girerek atılacak adımları için programcıya yardımcı olurlar.

Property Page seçeneği activex kontrollerine ait özelikleri değiştirmek için kullanılabilecek Özellikler penceresi oluşturmak için kullanılır.

New application

Aktif projeyi kapatarak yeni bir projenin başlatılmasını sağlar.

New data module

Yeni bir modülü oluşturmak için kullanılır.

New form

Projeye yeni form eklenir.

Open

Diskte var olan DPR (delphi-project), PAS (delphi-unit), DFM (delphi-form) ve TXT (metin) dosyalarını açmaya yarar.

ReOpen

Son çalıştığınız projelerin listesi bu menü altında tutulacak. Böylece son kullandığınız bu menü altından kolayca açabilirsiniz.

Save.Save As,Save All

Projedeki dosyaları aynı isimle veya yeni isimle kaydeder. Projenizi kaydettiğinizde projedeki unitlerin isimi kaydettiğiniz isim olacaktır. Bu yüzden dosya isimlerinde bazı Türkçe karakterleri kullanamazsınız. Ayrıca kaydettiğiniz isimle Unit satırında tanımlanan isim aynı değilse delphi hata verecektir. Bu yüzden unit satırındaki ismi değiştirmemelisiniz.

Close,CloseAll

Aktif dosyayı veya projedeki bütün dosyaları kapatır.

Add To Project

Projeye var olan Unitlerden (.PAS) ekler.

Remove form Project

Açılan pencereden seçilen unit veya formu projeden çıkarır.

Edit menüsü:

Cut,Copy,Paste

Keme kopyalama ve yapıştırma işlemlerini yapar. Sadece kodları değil kontrolleride kesip yapıştırabilirsiniz. Bu işlem o kontrol ile aynı özelliklere sahip fakat isme sahip yeni kontroller oluşturulur.
Eğer bir kontrolün kopyasını çıkarırsanız oluşan yeni kontrol eski kontrole ait olay alt programlarını kullanacaktır.

Align to Grid

Seçilen kontrolün koordinatları form üzerindeki gridlere denk gelmiyorsa en yakın grid noktasına denk gelecek şekilde taşınır.

Brign To Front, Send To Back

Üst üste gelen kontrollerden birini öne ve arkaya almaya yarar.

Align

Seçilen kontrolleri aynı hizaya getirmeye yarar.

Size

Seçilen kontrollerin hepsinin aynı boya ve/veya aynı yüksekliğe sahip olmasını sağlar. Açılan pencerede seçile seçime hepsi en küçüğünün veya en büyüğünün boyuna getirilmesi sağlanır.

Scale

Form üzerindeki kontrollerin seçilen oranda büyültüp küçültmeye yarar. Bir seferde %25 ile %400 oranında ölçeklendirilebilir.

Tab Order

Form üzerindeki kontrollerin tab sırasını değiştirmeye yarar. Tab sırası kullanıcının kontroller arasında tab tuşu ile geçiş yaparken sırası ile hangi kontollere geçileceğini belirler. Açılan pencerede şu anki tab sırası görüntülenir. Aşağı/Yukarı düğmesi ile seçilen kontrolün tab sırası değiştirilebilir.

Lock Control

Form üzerindeki şu anki koordinatlarına kilitleyerek yerlerinin değiştirilmesini önler. Tekrar seçilirse bu kilit kalkar. Özellikle üzerinde bir çok kontrolün bulunduğu formlarda form tasarımı bittikten sonra yanlışlıkla kontrollerin yerinin değiştirilmesini önlemek için faydalı bir seçenektir. Özellikle bir çok kontrolün bulunduğu formlarda ekrana sığmayan kontrolleri de ekranda kaydırma çubuğu olmadan göstermek için kullanışlı bir seçenektir.

Search menüsü :

Find

Kod penceresinde herhangi bir ifadeyi bulmak için kullanılır.

Find in Files

Belirlenen bir dosyanın içinde bir ifade bulmak için kullanılır.

Replace

İstenilen ifadeleri bir diğeriyle değiştirmek için kullanılır. Aynı yerde geçen birden fazla olan ifadeler otomatik olarak tek adımda değiştirilir.

Search Again

Aranılan ifade F3 tuşu ile tekrar aratılır.

Incremental Search

Arama aşağı doğru devam ettirilir.

Goto Line Number

İstenile bir adıma tek adımda ulaşmak için kullanılır.

View Menüsü :

Project souce

Projenin oluşturulması için gerekli olan kodu kod penceresinde gösterir.
Bu kod, proje oluşturulurken delphi tarafından oluşturulur.

Project Manager

Project manager penceresini görüntüler. Project manager penceresi normalde ekranda görülmez. Bu pencerede programınıza ait formlar ve unitler gösterilir. Bu pencere içindeki dosyalar DPR uzantılı bir dosyada tutulur. Projeden bir dosyayı çıkarmak için remove seçeneğini, pencereye var olan dosyalardan eklemek içinde add seçeneğini kullanabilirsiniz. Projenizle ilgili ayarları da options seçeneği ile yapabilirsiniz. Ekranda görülmeyen bir formu görüntülemek için bu penceredeki view form seçeneği kullanılabilir.

Object Inspector

Object inspector penceresini görüntüler. F11 kısayol tuşu ile de ulaşabilirsiniz. Bu pencere form üzerine component paletten yerleştirilen her kontrolün özellik ve olaylarını belirler.

Alignment Palette

Bu pencere seçilen kontrolleri aynı hizaya getirmeye yarar.

Windows List

O anda açık olan pencerelerin listesini gösterir.Listeden seçili olan pencereye geçiş yapabilirsiniz.

Component List

Form üzerinde kullanabileceğimiz kontrol elemanlarını göstermeye ve istenilen form üzerine almaya yarar.

Break Points

Programı adım adım çalıştırırken programın bir noktaya kadar çalışıp geçici olarak durmasını sağlamak ve o noktadaki değişkenlerin değerlerini incelemek için belirli noktalara F5 tuşu ile breakpointler konabilir. Programınıza fazla sayıda breakpoint koyduysanız bunları takip etmek için bu menü seçeneğini kullanabilirsiniz.

Call Stack

Bu seçenekte adım adım çalıştırmada kullanılır. Şu anda çalışan fonksiyonun hangi fonksiyonlar tarafından çağrıldığını gösterir. Özellikle kendi kendini çağıran prosedürler söz konusu ile prosedürün kaç düzeyde kendini çağırdığını görmek için gereklidir.

Watches

Program adım adım çalıştırılırken değişkenleri izlemek üzere run-add watch seçeneği ile eklenmiş değişkenlerin bulunduğu pencereyi görüntüler. Bu pencerede değerleri izlemek üzere eklenmiş değişkenlerin değerleri görülmektedir. Bu değişkenlere yeni birisini eklemek için listedeki boş bir satırı çift tıklayabilir, listedeki değişkenlerden birini silmek için del tuşuna basabilirsiniz.

Toggle Form/Unit

Formlar ve unitler arasında geçiş yapmak için kullanılabilecek bir seçenek. F12 kısayol tuşu ile de kolayca geçebilirsiniz.

Unit

Şu anda projede mevcut olan unitleri bir pencerede listeleyerek bunları seçip görüntüleme imkanı verir.

Forms

Şu anda projede mevcut olan formları bir pencerede listeleyerek bunları seçip görüntüleme imkanı verir.

New Edit Window

Şu andaki kod penceresinin bir kopyasını yeni bir kod penceresi ile gösterir. Bunlardan birinde yaptığınız değişiklik diğerinde de anında yapılır. Bu yeni pencere size kodunuzun aynı anda göremediğiniz kısımlarını göme imkanı verir.

Project Menüsü :

Add to project

Projeye var olan unitlerden eklemeye yarar. Seçeneği seçtiğinizde karşınıza çıkan dialog penceresinde istediğiniz unit ilgili klasörden bulunup eklenebilir.

Remove from project

Açılan pencere vasıtasıyla seçilen formları veya unitleri projeden çıkarır.

Import type library

Bu seçenekle açılan pencere vasıtasıyla bilgisayarınızda kullandığınız ve kontrol elemanlarını içeren kütüphane dosyalarını projenize ekleyebilirsiniz. Buradaki add düğmesi bu pencereye yeni kütüphane dosyaları ekler. Remove düğmesi de penceredeki dosyaları kaldırır. Ok düğmesi ise seçili olan dosyayı bir unit olarak projeye ekler.

Compile

Programı çalıştırmadan derler. Örneğin bir DLL dosyası yazılırken bu dosya çalıştırılamaz. Bunun sadece derlenmesi gerekir.

Build All

Compile veya run seçeneği ile derlenen projede bütün unitler ve formlar derlenmez. Bunun yerine en son derlemeden sonra değişen dosyalar derlenir. Build all seçeneği değişmiş olsun olmasın bütün dosyaları yeniden derler.

Syntax Check

Bu seçenek Link yanmadan sadece yazım hataları olup olmadığını kontrol eder.

Information

Program derlendikten sonra proje hakkındaki teknik bilgiyi bir dialog penceresi ile görüntüler.

Options

Programla ilgili ayarların yapılabileceği pencereyi açar. Bu pencerede yapılan ayarlar sadece şu andaki projeyi etkileyecektir. Eğer bütün projeler için bu ayarların kullanılmasını istiyorsanız default düğmesini işaretlemelisiniz. Penceredeki forms tabı kısmı(bölümü)'nda main form kutusunda programın hangi formun ana form olacağı belirlenir. Normalde projede ilk oluşturulan form ana formdur ve ilk olarak bu formla program çalışmaya başlar.

Auto create forms listesinde hangi formların otomatik olarak oluşturulacağı belirlenir. Diğer listede (available forms listesi) bulunan formlar ise otomatik olarak oluşturulamaz. Bunları oluşturmak için create metodu kullanılmalıdır. Hangi formları otomatik olarak oluşturulup oluşturulamayacağını formları bu listeler arasında taşıyarak belirleyebilirsiniz.

Application tabında ise programın başlığı, help dosyasının ismi ve ikonu belirlenir. Proje için bir help dosyası belirlenmişse kullanıcı programı çalıştırırken F1 tuşuna basarsa bu help dosyası winhelp32 aracılığıyla görüntülenir ve kullanıcının üzerinde bulunduğu kontrolün helpcontextID özelliği ile belirlenen konu gösterilir.(Bu numaralı konunun ne olduğu help dosyası tasarlanırken belirlenir.)

Title kutusunda belirlenen başlık ise formun değil programın başlığıdır. Bu başlık program minimize edildiğinde, görev çubuğunda görülen isimdir.

Icon kutusunda belirlenen ikon da title özelliği gibi forma ait değil programa ait ikondur. Yukarıdaki durumlarda da yine bu ikonla temsil edilir.

Run :

Run

Programı derleyip çalıştırır. Program zaten derlenmişse sadece çalıştırır. Ayrıca programın çalışması kırılmışsa kaldığı yerden devam ettirir. F9 kısayol tuşu da kullanılabilir.

Parameters

Program komut satırından parametre alıyorsa tasarım aşamasında komut satırı parametreleri buradan verilir.

Program Reset

Programın çalışmasını durdurur. Ancak bu işlem bazen sitem kaynaklarının azalmasına sebep olabilir. Böyle bir durum söz konusu ise delphi sizi uyaracaktır. Program bu seçenek sonlandırıldığında açık olan dosyalar kapatılır, component library'den oluşturulmuş kontrollere verilen sistem kaynakları geri alınır ve değişkenler bellekten atılır. Ancak programınızda kendi kodlarınızla oluşturduğunuz sistem kaynağı tüketen bileşenlerin kullandığı kaynaklar serbest bırakılmayacaktır.

Hata Ayıklama İşlemleri

Eğer programın sonsuz döngüye girmesi sebebi ile normal yollardan kapatamıyor ve program reset menüsüne de ulaşamıyorsanız Ctrl+Alt+SysReg tuşlarına bamayı deneyin.

Menüdeki diğer seçenekler vasıtası ile program adım adım çalıştırılarak programdaki hatalar ayıklanabilir. Bahsedilen hatalar yazım hataları değil mantık hatalarıdır. Program adım adım çalıştırılarak değişkenlerin durumu bu seçenekler vasıtasıyla incelenebilir.

- Programı debug modunda çalıştırmak için F8(step over) veya F7(trace intro) tuşu kullanılabilir.
- Ayrıca belirli bir satıra kadar çalışması içinde F5(Add breakpoint) tuşu ile durulacak satır belirlendikten sonra F9 tuşu ile derlenir.
- Programa bu işlemler yapılmada başlansa bile program pause seçeneği ile programın çalışması durdurulup debug moduna geçilebilir.
- Bu modda değişkenlerin durumu fonksiyonların çağrılma sırası görülebilir.

Step Over(F8)

Programı bir onraki satıra kadar çalıştırır. Her satırı tek tek çalıştırıp eskisini görmek için kullanılabilir.

Trace Intro(F7)

Step over gibi satırları tek tek çalıştırmaya yarar, ancak step over'den farklı olarak eğer satırda bir fonksiyon çağrısı varsa fonksiyona gidilir.

Add Breakpoint(F5)

Programı F8 ile adım adım çalıştırmak her zaman kullanışlı değildir. Örneğin 1'den 1000'e kadar olan bir for-do döngüsünden F8 ile çıkmak zordur. Bu gibi durumlarda break point dediğimiz, programdaki bazı satırlara F5 ile kesme konur ve F9 ile program çalıştırılır. Kontrol kesme konan satıra program kırılır ve sonuçlar bu şekilde izlenebilir.

Show Execution Point

Kursörü şu anda çalışacak satıra götürür.

Add Watch

Program çalışırken izlenecek değişkenler bu seçenek ile eklenir. Bu yöntemle eklenen değişkenlerin değerleri watch list penceresinde görüntülenir.(Bu pencereyi view-watches menü seçeneği ile görüntüleyebilirsiniz.) Global ve o anda çalışan prosedürdeki local değişkenlerin şu naki değerleri pencerede görüntülenir.

Add watch seçeneği bir inputbox penceresi açarak izlenecek değişkeni girmenize imkan verir. Bu penceredeki expression kutusuna izlenecek değişkenin adı girilir.

Enabled seçeneği kaldırılırsa değişkenin değeri listede gösterilmez. Bir çok değişkeni birlikte izliyorsanız programın çalışması yavaşlayacaktır. O anda değerine ihtiyaç duymadığınız bazı değişkenleri bu yöntemle izlenmesini durdurabilirsiniz.
Radio düğmeleri ile değişkenin uygun başka bir formatta gösterilmesini sağlayabilirsiniz. Default seçeneği değişkeni orijinal haliyle gösterir. Örneğin integer bir değişkenin **hexadecimal** sistemde gösterilmesi için **hex integer** seçeneği kullanılabilir.

Memory Dump seçeneği değişkenin bellekteki halini gösterir. Örneğin bir string değişkenin bellekte durumunu görmek için bu seçeneği kullanabilirsiniz. String'in ilk byte'nın karakter sayısını gösterdiğini görebilirsiniz.

Evaluate/Modify

Bu seçenekle girilen değişkenin sadece o andaki değeri öğrenilebilir veya değiştirilebilir. Add watch seçeneğinden farklı olarak değişken debug penceresine eklenmez ve sürekli takip edilmez. Sadece seçildiği andaki değeri görüntülenir. Bu seçenek bir inputbox penceresi açarak değişkeni girmenize ve değiştirmenize imkan verir.

Expression kutusuna değişkenin adı girilir. Evalute komutu değişkenin şu anki değerini görüntüler. Bu değer değiştirilmek isteniyorsa New value kutusuna değişkenin almadı istenen yeni değer girildikten sonra Modify düğmesi kullanılır. Ayrıca view-call stack seçeneği ile de şu anda çalışan prosedürün kimler tarafından çağrıldığı görülebilir.

Component Menüsü :

New

Yeni bir component dosyası oluşturmak için kullanılır.

Install Component

Component palete yeni kontroller eklemek için kullanılır.

Configure Palette

Component palette yer alan kontrollerin yerlerini değiştirmek için veya silmek için kullanılır.

Import Activex Control

Activex (OCX) kontrollerini de delphi ile kullanabilirsiniz. Bu menü ile açılan aşağıdaki pencereyi kullanarak sisteminize kayıt yapılmış activex kontrollerinden seçtiklerinizi ekleyebileceğiniz gibi Add düğmesi ile de yeni OCX dosyalarını ekleyebilirsiniz.

Database Menüsü :

Explore

Database explorer programını çalıştırır.

SQL Monitor

SQL monitor programını çalıştırır.

Form Wizard

Programa database formları eklemek için yardımcı olur. Açılan pencereler takip edilerek formun tipini, kullanacağı database dosyasını ve bu dosyadaki anları seçmenizi sağlayarak bir database formu oluşturmanızı sağlar.

Tools Menüsü :

Environment Options

Bu seçenekle açılan pencereden editörle ilgili özellikler belirlenir.

Repository

Delphi kendi formlarınızı hazırlayarak bunları daha sonrada standart form olarak kullanmanıza imkan verir. Bu pencere vasıtası ile hazırlanan formlar galeriye eklenerek daha

sonra new-form seçeneği ile yeni bir form oluştururken buraya eklediğiniz formlardan birini seçebilmenizi sağlar.

Configure Tools

Tools menüsüne çok kullandığınız programlardan ekleyebilirsiniz. Bu menüye yeni program eklemek için tools menüsünün configure tools seçeneği kullanılır.

Sağ Fare Tuşunun Kullanımı

Windows altında çalışan programların çoğu sağ fare tuşunu destekler. Böylece sağ fare tuşu ile açılan popup menüler aracılığıyla yapılacak işlemlere oldukça hızlı errişim sağlanmış olur.

Form Üzerinde Sağ Fare Tuşu

Form üzerinde sağ fare tuşu tıklanacak olursa açılan popup menüdeki seçenekler edit menüsü ile aynıdır.

Align to Grid

Seçilen kontrolün koordinatları form üzerindeki gridlere denk gelmiyorsa en yakın grid noktasına denk gelecek şekilde taşınır.

Brign To Front, Send To Back

Üst üste gelen kontrollerden birini öne ve arkaya almaya yarar.

Align

Seçilen kontrolleri aynı hizaya getirmeye yarar. Açılan penceredeki radio düğmeleri kullanılarak seçili olan eleman sağdan,solan,üstten,merkezden,alttan aynı hizaya getirilebilir. Hatta seçili olan elemanlar formun tam ortasına alınabilir.
(Center in window seçeneği ile)

Size

Seçilen kontrollerin hepsinin aynı boya ve/veya aynı yüksekliğe sahip olmasını sağlar. Açılan pencerede seçile seçime hepsi en küçüğünün veya en büyüğünün boyuna getirilmesi sağlanır. Buradaki no change seçeneği ile elemanlar üzerinde bir işlem yapılmazken, shrink to smallest seçeneğiyle, seçili olan en küçük boyutlu elemanın boyutuna, grow to largest ile de en büyük elemanın boyutuna getirilir. Width, heigth seçenekleri ile de seçili olan elemanları genişlik ve uzunlukları değiştirilir.

Scale

Form üzerindeki kontrollerin seçilen oranda büyültüp küçültmeye yarar. Bir seferde %25 ile %400 oranında ölçeklendirilebilir. Bu pencerenin kutusuna girilen sayısal ifade seçili olan elemanların boyutlarını değiştirir.

Tab Order

Form üzerindeki kontrollerin tab sırasını değiştirmeye yarar. Tab sırası kullanıcının kontroller arasında tab tuşu ile geçiş yaparken sırası ile hangi kontollere geçileceğini belirler. Açılan pencerede şu anki tab sırası görüntülenir. Aşağı/Yukarı düğmesi ile seçilen kontrolün tab sırası değiştirilebilir.

Creation Order

Uygulamanın oluşturduğu visual olmayan elemanları sırası bu pencere ile düzenlenebilir.

Kod Penceresinde Sağ Fare Tuşu

Close Page :Seçeneği ile mevcut kod penceresi kapatılır.

Open file at cursor : Kursörün bulunduğu pozisyona istenilen dosya, aç dialog kutusu ile eklenir.

Topic search : Seçili olan deyim hakkında online yardım görüntülenir.

Toggle marker : Kod editörünün istenilen yerine işaret koyar. Bu işlem birinci defasında işaret koyarken ikinci defasında kaldırır.

Go to marker : Daha önce işaretlemiş yere direkt cursoru konumlandırır.
Toggle breakpoint : Kod editörünün istenilen noktasına, program kesme noktası bırakılır veya kaldırılır.
Run to cursor : Kursörün bulunduğu noktaya kadar programı çalıştırır.
Evaluate/Modify : Bu menü seçeneği ile evaluate/modify dialog penceresi aktif hale getirilerek, var olan bir ifadenin değeri değiştirilebilir yada değerlendirilebilir.
Add watch at cursor : Menü seçeneğiyle watch dialog penceresi açılır. Burada gözleme pencereleri oluşturulur veya değiştirilir.
Read only : Menü seçeneğiyle mevcut açık olan dosyaya sadece okunabilirlik özelliği verilir. Dosya üzerinde herhengi bir değişiklik yapılmaz.
Message view : Hata mesajlarını gösterecek olan pencereyi görüntüler yada saklar.
Properties : Bu seçenekle kod penceresine ait bazı ayarlamaların yapıldığı pencere görüntülenir. Burada genel olarak kod editörü ile ilgili yazı ayarlamaları, renk ayarlamaları ve görüntü ayarlamaları yapılır.

Project Inspector Penceresinde Sağ Fare Tuşu

Expand : Alt seçenekleri + ile temsil edilen özellikler için aktif hale gelir. Bu seçeneğin tıklanmasıyla alt seçeneklerde görülür.
Collapse : Alt seçenekleri açılmış özellikler için aktif hale gelir. Bu seçeneğin kullanılmasıyla alt seçenekli özellikler + halinde temsil edilir.
Saty on top : Object inspector penceresini daima en üstte tutar.
Hide : Object inspector penceresini gizler.
Help : Pencere ile ilgili yardım dosyasını açar.

Component Paleti Sağ Fare Tuşu

Configure : Component paletini yeniden belirlenen şekli ile kaydeder.
Show hints : Component paletinde bulunan elemanların isimlerini görüntüler.
Hide : Component paletini saklar.
Properties : Componentlerle ilgili bazı ayarlamaların yapıldığı pencereyi görüntüler.

Standart Kontrol Elemanları

Standart kontrol elemanlarının kullanım amaçlarını kısaca verelim:
TMainMenu : Menü çubuğunu tasarlamada kullanılır. Bu kontrol kullanıcının bilgi girişi yapmasına imkan veren en çok kullanılan elemanlardandır.
TPopupMenu : Sağ fare tuşu ile çalışan menüler hazırlamada kullanılır.
TLabel(Etiket) : Form üzerine açıklama yazmak yada kontrollere açıklama yazmada kullanılır.
Tedit(Metin kutusu) : Kullanıcının bilgi girmesi için kullanılır.
TMemo(çok satırlı bilgi girişi kutusu) : Kullanıcının birden fazla satıra sahip bilgileri girmesi için kullanılır. Edit kutusunun yapabildiği bütün işlemleri yapabilirken
TButton(Komut düğmesi) : Kullanıcının bir işi yaptırabilmesi için kullanılır.
TCheckBox(İşaret kutusu) : Bir seçeneği aktif veya pasif yaptırmak için kullanılır.
TRadioButton(Seçenek kutusu) : Birden fazla seçenekten birisinin seçilmesi gereken durumlarda kullanılır.
TLisbox(Liste kutusu) : Birden fazla elemanı listelemek ve düzenlemek için kullanılır.

TComboBox(Aşağı doğru açılan liste) : Kullanıcının hazır değerlerden birini seçebilmesi için kullanılır.
TScrollBar(Kaydırma çubuğu): Kaydırma işlemlerinde veya değer artırıp azaltma işlemlerinde kullanılır.
TRadioGroup : Şeçenek düğmelerini tasarım zamanı oluşturabilmek için kullanılır.
TPanel Kontrol Elemanı : Diğer kontrolleri gruplamak ve durum çubuğu oluşturmakta kullanılır.
TGroupBox(Gruplama kutusu) : Diğer kontrolleri (daha çok seçenek düğmelerini) gruplamakta kullanılır.

Tablo Özellikleri

Bu Bölümde saha geçerlilik kurallarının tanımlanması,Table lookup,secondary indexes(ikincil anahtarlar),referantial Integrity(tablo ilişkileri),Şifre güvenliği ve Tablo dilinin ayarlanması konularıolacak.

A) Sahaların Geçerlilik Kuralları:

Bir sahanın üzerinde dolaşırken eğer tipini tanımladıysanız sahanın tipine göre saha özelliklerini belirleyebileceğiniz metin kutuları,bir onay kutusu ve bir düğme kullanılabilir hale gelir.Bu metin kutularını kullanarak sahayı istediğiniz özelliklerde belirleyebilirsiniz.
Required Field: Tabloya giriş esnasında kullanıcı sahayı atladığında hata vermesini sağlar.Böylece kullanıcını hiç bir şekilde bu sahayı boş geçmemesini sağlarız.
Minimum Value:Sahanın tip tanımlamasına uygun bir şekilde sahanın alabileceği minimum değeri içerir. Örneğin bir sepetteki elma sayısı asla eksili bir değer olamaz. Bunun için minimum değerine 0 atamalıyız.
Maximum Value:Sahanın tip tanımlamasına uygun bir şekilde sahanın alabileceği maximum değeri içerir.Örneğin iskambik kağıdı no'su tutan bir saha için iskambil kağıdı sayısı 52'den fazla olamaz.olursa hile var demektir :)
Default Value:Bu sahanın varsayılan değeridir. Minimum ile maximum değerleri arasında bir değer almalıdır.
Picture: Bu sahanın görünüm ve giriş şeklidir.Bu özelliği kullanarak sahanıza yapılan girişin sizin belirlediğiniz kurallara uygun olup olmadığını hiç bir kod yazmadan kontrol edebilirsiniz.

Örnekler:
yazılabilir.

Örneklere bakarak sizde kendi giriş formüllerinizi yazabilirsiniz.Bunun için önce yukarıdaki resimde gördüğünüz Assist düğmesine basarak yandaki resimde olduğu gibi bir ekran çıkaracaksınız. Daha sonra picture yazan metin kutusuna düşündüğünüz formatı yazacaksınız. Doğru olup olmadığını test etmek istiyorsanız Verify Syntax düğmesine basmalısınız.
Eğer yazdığınız formatı denemek istiyorsanız Sample Value metin kutusuna formatta belirttiğiniz şekilde bir veri girmelisiniz. Eğer girdiğniz verinin formata uygun girilip girilmediğini merak ediyorsanız Test Value düğmesine basın.Eğer Value is Valid mesajını görürseniz girdiğiniz veri hazırladığınız formata uygun demektir.
Hazırladığınız bu formatı kaydetmek için add to list düğmesine basmalısınız.
Eğer listedeki bir format kullanmak istiyorsanız formatı seçip Use düğmesine basmalısınız.

B) Table Lookup:

Table lookup tanımlamak için Table Properties combo kutusundan table look up seçeneğini seçip define düğmesine basmalısınız

Karşınıza yukarıdaki gibi bir ekran çıkacak. Burada sağda tanımlı olan sizin şu anda üzerinde çalışmakta olduğunuz ve look up tanımlayacak olduğunuz tablodur. Buradan hangi sahayı look up saha olarak tanımlamak istiyorsanız o sahayı liste kutusundan seçip FieldName metin kutusunun solundaki düğmeye basarak metin kutusuna atamalısınız.

Soldaki sahada seçilen aliasın içerdiği tabloları görüyorsunuz. Bu tablolardan look up yapmaya uygun bir tanasini seçip look up field metin kutusunun sağındaki düğmeye tıklayarak metin kutusuna atamalısınız. Burada dikkat edeceğiniz husus FieldName ile LookUp field sahalarının tiplerinin birbirine uyması. aksi taktirde hata verecektir. Unutmayın ki lookup tablonun hep ilk satırı lookup field metin kutusuna gelir.Eğer Lookup tablosu olarak kullanacağınınz tablonun ilk sahası anahtar saha olursa bu lookup yapma işlemini hızlandırır.

Look up Type

Just Current Field: Sadece Tanımlanan saha look up dan gelen bilgiler tarafından doldurulur.

All Corresponding Fields:Sadece tanımlanan look up saha değil look up tabloda bulunan ve lookup saha içeren tablonun tipleri ve adları uyan sahalarına da varsayılan değerleri atar. Örneğin içinde soyadı ve adı bulunan bir tablodan lookup yaptığımızı düşünelim. Eğer lookup yapan tabloda soyadı ve adı sahaları varsa mutlaka bu iki değeri doldurulur.

Look up Access

Fill No Help: Sahaya bir veri girerken yardımcı olmaz.

Help And Fill:Sahaya veri girerken yadımcı olur.

C) Secondary Indexes:

İkincil indeks, kayıtlar arası aramaları hızlandırmak yada farklı bir sıralama kuralına göre kayıtları dizmek için tanımlanır. İkincil indeksi bir sahadan yaratabileceğiniz gibi birden çok sahanın bir kombinasyonu olarakta yaratabilirsiniz.Not olarak ikincil indeks ile sadece bir kaydın sıralama kuralını değiştirebilirsiniz. Onun fiziksel sırasını değil.

ikincil Indeks tanımlamak için Table Properties combo kutusundan table Secondary Indexes seçeneğini seçip define düğmesine basmalısınız.

Yukarıdaki ekran gibi bir ekran karşınıza çıkacak.Sağ tarafta üzerinde değişiklik yaptığınız tablonun sahaları görünmekte. Buradan birini yada birkaçını seçerek sağ ok düğmesine basarak Indexed Fields liste kutusuna ekleyebilirsiniz.

Alttaki change order adıyla tanımlanan düğmeleri kullanarak sahaların indekslenme sırasını değiştirebilirsiniz.

Index option grup kutusunda ise 4 özellik tanımlanmış.Bunlar:

Uniquie:İkincil indekste birden fazla kaydın aynı değeri içerip içeremeyeceğini belirtir.Eğer Uniquie işaretli ise ve Database Desktop tekrarlı bir kayda rastlarsa indeks uygulanmaz ve bir hata mesajı görüntülenir.Sahanın verisini değiştirip tekrar indekslemeye çalışabilirsiniz.

Descending:İkincil indeksin azalan yönde mi yoksa artan yönde mi sıralanacağını belirtir.Eğer işaretli değilse azalandan artana doğrudur.

Case Sensitive:Sıralamada büyük yada küçük harf oluşuna dikkat edilip edilmeyeceğini belirtir.
Eğer işaretliyse veri şu sırada konumlanır:
Abcd, aBcd, aaaa
Eğer işaretli değilse şu sırada konumlanır:
aaaa, Abcd, aBcd

Database Desktop tek sahaları,büyük küçük harf duyarlı indeksleri otomatik isimlendirir.Eğer büyük küçük harf duyarlılığı olmayan bir indeks kaydedecekseniz kaydederken bir isim vermelisiniz. Bu size büyük küçük harf duyarlılığı olan ve olmayan, aynı sahalara sahip, iki farklı indeks tanımlamanızı sağlar.

Maintained:İkincil indeksin korunup korunmayacağını belirtir.
Maintained indeksler her tablo değişikliğinde tekrar kaydedilir. Bu Sorgular gibi işlemleri hızlandırır.Maintain indeksler anahtar içeren tablolarda geçerlidir.Maintain olmayan indeksler sadece indeks kullanılırken güncellenir.Örneğin bir tabloya bağladığınızda yada bir sorguyu çalıştırdığınızda.

Bir maintain olmayan indeksi kullanan bir işlem yaptığınızda bu işlem diğerinden biraz daha uzun sürer. Çünkü ilk önce indeks tablonun değişen verilerine göre yeniden düzenlenip indeks kurallarını yeniden sıralanacaktır.Her halukarda eğer bir maintain olmayan indeks kullanmak isitiyorsanız değişmeyen bir tabloda kullanının.Mesela sadece okunabilir tablolarda maintained olmayan indeksler daha hızlıdır.

D) Referantial Integrity:

Referantial integrity tablolar arası ilişkileri tanımlar.tabloar arası ilişkilerin nasıl tanımlanacağı ve neden tanımlanması gerektiği bir önceki makalede ayrıntıları ile verilmiştir.Tablolar arası ilişkileri tanımlamak için Table Properties combo kutusundan Referantial Integrity seçeneğini seçip define düğmesine basmalısınız.

Yukarıdaki resimde gibi referantial integrity menüsü çıkacak. Fields liste kutusundan seçeceğimiz saha ile ilişkilendirmeyi düşündüğümüz tabloyu seçip table liste kutusunun yan tarafındaki düğmeye tıklamalıyız.yukarıdaki resimin sağ tarafında, üzerinde çalışmakta olduğumuz tablonun sahaları var.Bu sahalardan hangisini seçtiğimiz tablonun anahtar sahası ile ilişkilendirmek istiyorsak o sahayı seçmeliyiz. Böylece iki sahayı birbiri ile ilişkilendirmiş oluruz.

Update Rule:

Cascade:Eğer ana tabloda herhangi bir değişiklik olursa bu direk bağlı tabloya yansır.
Prohibit:Eğer ana tablodan bir veri silinecekse bağlı tabloda da bu veri kullanılmışsa silinmesine izin verilmez.

Strict Referantial integrity:Eğer bu işaretli ise; bu tablonun daha önceki paradox sürümleri tarafından kullanımı esnasında tablo ilişkisinin bozulmamasını sağlar.

E) Password Security:

Tablonuzun başkası tarafından görülmemesini istiyorsanız bir şifre belirtmelisiniz.(En azından Supervisor password'u bilmeyen biri tarafından).Bunun için Table Properties combo kutusundan Password Security seçeneğini seçip define düğmesine basmalısınız.
Karşınıza yandaki gibi bir ekran gelecek. Buradan bir şifre belirtebilirsiniz.Eğer Auxiliary Passwords düğmesine tıklayacak olursanız karşınıza aşağıdaki gibi bir ekran çıkacak. Bu ekrandan ilişkisel şifreler belirtebilirsiniz. Bu şekilde bir kişinin o tablo üzerindeki otoritesini de belirmiş olursunuz.
Not: Buraya belirteceğiniz şifre Ana şifreden mutlaka farklı olmalıdır aksi taktirde buraya şifre belirtmenin hiç bir anlamı yoktur. Tüm işlemler için aynı şifreyi belirtebileceğiniz gibi her işlem için ayrı ayrı şifreler de belirtebilirsiniz.Bunu new tuşu ile yeni bir şifre açıp şifreleyeceğniz sahaları seçip saha operasyonunu seçip şifrenizi girdikten sonra add düğmesine basmanız yeterli olacaktır.
Buradaki operrasyonlar:
Eklemek ve silmek,Sadece veri girmek, güncellemek,sadece okumak ve tüm işlemleri yapabilmek olarak sınıflandırılmıştır.
Not:Eğer gerçekten veri güvenliğine ihtiyacınız varsa Crypt veDeCrypt metotlarını kullanın.

F) Table Language:

Table Language kısaca tablonun dilini belirleyeceğiniz bölümdür.Standart olarak tablonun dili ingilizceye uygun bir kod sayfasıdır. Eğer Türkçe bir kod sayfası kullanmak istiyorsanız.Aşağıdakilerden birini kullanabilirsiniz.

Base Trk cp857
Paradox Turk
Pdox ANSI Turkish
Unutmamanız gereken asıl şey Türkçe büyük harfin küçük harfe yada küçük harfin büyük harfe çevrilirken yaşayacağınız problem.Bu tüm Database Desktop elemanları için geçerli bir sorundur. Büyük İ çevrilirken i ye çevrileceğine kod sayfasnıa göre farklı bir karaktere çevrilir. aynı şekilde I ise küçük ı ya çevrileceğine i'ye çevrilir. Bu bir fonksiyonla düzeltebilirsiniz.
Veri Tabanı ve DataBase Desktop
Veri Tabanı Nedir?

Veritabanı; kısaca verilerin tutulduğu yer anlamına gelir. Fakat bu tanımdan kasıt verilerin tutulduğu bir dosya olmayabilir.Örneğin BDE bir veritabanını her bir dosyaya bir tablo gelecek şekilde ayırmıştır.Fakat Access için tek bir dosya söz konusudur.

Tablo:Düşünün ki bir miktar saklamak istediğiniz veriniz var. Bunlar:

Can,Kaynak,jankaynak@hotmail.com,Turkware Software Co.,
Mustafa,Kapsal,mustafakapsal@turkware.com,Turkware Software Co. vs.

Eğer yukarıdaki verileri guruplamak istesek:

Soyadı Gurubu:Kaynak,Kapsal
Ad Gurubu:Can,Mustafa
Şirket Gurubu:Turkware Software Co.,Turkware Software Co.
E-Mail Gurubu:jankaynak@hotmail.com,mustafakapsal@turkware.com

Gördüğünüz gibi 4 ana başlık altında toplayabildim böylece daha düzenli oldular ve aradığımı çabuk bulabilecek hale getirdim.
Yani tablo haline getirdim. Tablonun anlamı aynı niteliği betimleyen verileri tek bir gurup altında toplamaktır.Bu guruplar benim tablomun sahalarıdır. Aşağıda yukarıdaki örnek bir tablo haline getirilmiştir.

yukarıdaki tabloda Soyadı,Adı,Şirket,E-Mail adları sahalarımı tanımlayan isimleri oluşturdu. Dikkat! bu bahsettiğim Soyadı,Adı,Şirket,E-Mail sahaları tablomun sadece sahalarını tanımlar tablomun verilerine dahil değildir.

Her satır bir kayıtı tanımlar. Kayıt, Saha adlarına uygun olarak girilmiş verilere denir.

Tabloda Birincil Anahtar:Tablo; veri tabanının en önemli özelliğine uymalıdır.Tabloda tek olma. yani tüm sahaları aynı iki kayıt bir veritabanında bulunamaz. Bunun için Veritabanını sıralayan bir sahaya daha ihtiyaca vardır. Tabi ki her zaman ayrı bir saha olacak diye bir kaide yok ama eğer sahanızda hiç tekrarı mümkün olmayan bir sahanız yoksa yeni bir saha yaratmanız kaçınılmazdır.Biz bu sahayı birincil anahtar olarak niteleriz. Yukarıdaki örneğe bir birincil saha ekleyelim.

yukarıdaki örnekte ID adında bir saha daha ekledim. Bu ID sahasını sıralama değişkeni olarak kullanacağım her eklediğim kayıt öncekilerden farklı bir sayı olacak. Böylece tabloda tek olama kuralını korumuş olacağım.

Yukarıda ID sahasının yanına parantez içinde bir yıldız koydum. Yıldız işareti o sahanı bir anahtar olduğunu belirtir.Bu yıldız tanımlaması tüm programcılar tarafından kullanılan bir standarttır.

Daha önce bahsettiğim gibi her zaman birincil anahtar eklemek için farklı bir saha kullanmayabiliriz. Bunu yerine basbaya e-mail sahasını da kullanabilirdim. Çünkü herkesin kendine ait bir e-maili olacağı için tabloda tek olma kuralı yine korunmuş olur.

Tablolar arası ilişkiler:

Bazen bir veriyi topluluğunu tek bir tabloda tutmak mümkün olmayabilir. Bunun gerekçesi tek bir öğe için birden çok kaydın olmasıdır. Bunu şöyle açıklayabiliriz:

Mesela benim birden çok telefonum var.bunu iki şekilde tutabilirim. birincisi:

yukarıdaki örnekteki gibi tutmak sağlıklı değildir. Çünkü eğer ben başka bir telefon sahibi daha olursam bunu nereye yazacağım. Bunun nihayi çözümü tablolara parçalamak olacaktır. Şöyle ki:

Gördüğünüz gibi yukarıdaki örnekteki gibi tanımlarsak iki ayrı tabloya ayırmış oluruz.Ama Telefon bilgilerini içeren tabloda bir şey eksik. Telefonları kime ait olduğu. Telefonların kime ait olduğunu belirten bir saha daha eklemeliyiz.Ekliyeceğimiz saha kişi bilgilerini içen tablodaki tekrarsız saha olmalı.Örneğimizi şu şekilde değiştiriyoruz.

Yukarıdaki örnekte gördüğünüz gibi telefon bilgilerini içeren tabloya eklediğimiz, kişi bilgilerini içeren tablodaki anahtar saha ile aynı niteliğe sahip saha;telefon bilgilerini kişi bilgileriyle eşleştirip anlaşılır bir kayıt kümesi oluşturdu.

İpucu: Eğer elinizde bir veri kümesi varsa ve hangilerinin hangi tabloda yer alacağına karar vermiyorsanız; şu yöntemi deneyin. Eğer bir kayıt için o saha birden çok bilgi içeriyorsa bu yeni bir tablodur.Bölün.

Eğer tabanınızda birkaç kayıt varsa veri bu işlemin işlerliliği fazladır.Fakat binlerce kayıtla uğraşıyorsanız bu tanımlarda size yetmeyebilir. Bu tabloların kendi kendine birbiri ile iletişim kurmasını istiyorsak ilişkilendirmeliyiz.

İlişkiler; bir veri tabanının içindeki tabloların birbiri ile aynı görevde olan sahalarının birbirine bağlanması şeklinde nitelendirilebilir.

İlişkiler ikiye ayrılır:

Bire-Çok ilişki
Bire-Bir ilişki
Bire-çok ilişki; bir tablodaki her bir veri için ilişkili tablodaki sahaya karşılık gelen birden çok veri varsa buna Bire-Çok ilişki denir.Örneğin Bir babaya karşılık iki çocuk varsa bu bire çok ilişkidir. Çünkü baba her ikisininde babasıdır.

Bire-Bir ilişki; Bir tablodaki her bir veri için diğer tablodaki sahaya karşılık gelen bir veri varsa buna Bire-Bir ilişki denir. Örneğin Bir kişinin bir saat takması gibi.
Bire-bir ilişkiler bir tablonun kendi sahaları arasındaki ilişkidir. Bire çok ilişkiler ise bir tablonun anahtar sahası ile başka bir tablonun aynı nitelikteki sahası ile ilişkisidir. Bu ilişkileri en iyi tasarlayabileceğiniz program MsAccess programıdır. Veritabanı uygulamalarında hiç faydalanmasanız bile sadece ilişkileri düzenleme ekranından faydalanabilirsiniz.Yandaki resim bire çok ilişkinin MSAccess programında nasıl görüntülendiğini açıklar.

Database Desktop

Database Desktop programı Delphi ile gelen bir programdır. Bu program yardımı ile tablolar ile ilgili her türlü işlemi yapabiliriz. Database Desktop pek çok veritabanı motorunu kullanmanıza izin verir.

Delphi kullanıcıları arasında en çok kullanılanı ise Paradox'tur. Çünkü Paradox büyük veri yapılarıyla çolayca başa çıkabilir. Bende bu yüzden şimdilik sadece Paradox kullanımını örnek göstererek Database Desktop'ı size anlatacağım.

Database Desktop'ta yeni bir tablo yaratmak için File/New/Table comutunu çalıştırmanız gerekiyor. Daha sonra karşınıza yandaki gibi bir ekran çıkacak buradan Paradox7'yi seçin.Eğer Daha önceki sürümleri kullanmak yada başka veritabanları kullanmak istiyorsanız yine buradan seçebilirsiniz ama ben Paradox7 ile örnekleyeceğim için Paradox7'yi seçmenizde fayda var.

Not:Paradox7 ve tüm Paradox sürümleri %100 güvenli değildirler. Şifreleme yaptığınızda bile eğer bir cracker Paradox supervisor şifrelerini biliyorsa kolaylıkla veritabanınızın içinde ne var diye bakabilir. Bunu önlemenin yolu Crypt ve DeCrypt yöntemeleri ile verilerinizi şifrelemektir.

OK tuşuna bastığınızda karşınıza yandaki menü gibi bir menü gelecek.Bu Menüde İlk olarak Field Name Type Size ve Key hücreleri olacak.
Field Name hücresine 25 karakter uzunluğunda Delphi değişken isimlendirme kurallarına uygun bir isim girmelisiniz.
Type Hücresi bir açılır menü şeklindedir. Menünün içeriği ve delphideki karşılığı:

Alpha (String)
Number(Float)
Money(Integer)
Short(ShortInt)
Long Integer(Integer)
#BCD(Double)
Date(DateTime)
Time(TDateTime)
@TimeStamp(TDateTime)
Memo(TMemo)
Formatted Memo(TRichEdit)
Graphic(TGraphic)
OLE(Tüm OLE Bileşenleri)
Logical(Boolean)
+AutoIncrement(Integer)
Binary(ikili veriler)
Bytes(Array of Char)

Alpha: Alpha tipi tüm yazdırılabilir karakterleri içerir.0-255 arasında boyutu olabilir. Delphi'de String tanımlı bir değişken tarafından kullanılabilir.

Number:Number tipi sadece pozitif/negatif işareti,ondalık işareti ve sayı içerebilir. Pozitif ve negatif ifadeler tutabilir.-10307den 10307'ye kadar değer alabilir.Delphi'de Float tanımlı bir değişken tarafından kullanılabilir.

Money:Money tipi tıpkı diğer tipler gibi sayı içerir fakat sadece gösterimi diğerlerinden farklıdır.Her üç hanede bir nokta ile ayrılır. fakat bu değerine etki etmez.Delphi'de Float tanımlı bir değişken tarafından kullanılabilir.

Short:Short Tipi -32,767'den 32,767'e kadar değer içerebilir.Delphi'de Integer tanımlı bir değişken tarafından kullanılabilir.

Long Integer:Long Integer tipi -2147483648'den 2147483647'e kadar değer içerebilir.Delphi'de LongInt tanımlı bir değişken tarafından kullanılabilir.

BCD(Binary Coded Decimal):BCD tipi ikili codlanmış sayısal veri içerir.Diğer sayısal tiplerin sağladığından daha hassas hesaplamalar yapmanız gerektiğinde BCD sahaları kullanın.Fakat hesaplamalar diğer veri tiplerinden daha yavaş olacaktır.

Date: Date tipi ile 1 Ocak 9999 M.Ö'dan 31 Aralık 9999 M.S'ye kadar ki tarihleri tutabilirsiniz.Tüm artık yılları düzgün bir şekilde tutar. Delphi'de TDateTime ile çevrimler sonucu kullanılabilir.

Time:Time tipi milisaniyeler halinde 24 saati adresler.Varsayılan saat tipini değiştirmek için Paradıx'u kullanmalısınız. Delphi'de TDateTime ile çevrimler sonucu kullanılabilir.

TimeStamp:TimeStamp tipi Time ve Date tiplerinde olan bütün özellikleri sağlayan birleşik bir tipdir. Delphi'de TDateTime ile çevrimler sonucu kullanılabilir.

Memo:Memo Tipi tüm yazdırılabilir ASCII kodları içerebilir (NULL dışında).Bu tip Mb uzantılı bir dosyanın içinde tutulur.Asıl tabloya bağlanır.Bu tipi delphide kullanmak için TMemo nesnesini kullanabilirsiniz.Boyut olarak 1MB dan 240 MB kadar bir boyut belirtebilirsiniz

Formatted Memo:Formatted Memo tipi bir Richeditin içerebileceği tüm içeriği sağlar.(renk,font,bold vs.).Tüm yazdırılabilir ASCII kodları içerebilir (NULL dışında).bu tip Mb uzantılı bir dosyanın içinde tutulur.Asıl tabloya bağlanır.Bu tipi Delphi'de kullanmak için TRichEdit nesnesini kullanabilirsiniz..Boyut olarak 1MB dan 240 MB kadar bir boyut belirtebilirsiniz

Graphic:Graphic tipi .BMP, .PCX, .TIF, .GIF, ve .EPS dosya formatlarını destekler. Bu dosya formatlarını BMP dosya formatınıa çevirip bu şekilde saklar.Graphic tipinin boyut tanımlamaya ihtiyacı yoktur çünkü tablodan farklı bir dosyada tutulurlar.Bu tipi Delphi'de kullanmak için TImage nesnesini kullanabilirsiniz.

OLE:OLE tipini her çeşit dosyayı tutmak için kullanabilirsiniz.OLE sahası bu tip verileri göstermek ve değiştirmek için bir yol sağlar.Fakat ne yazık ki Database Desktop bu içeriği gösteremez.OLE tipinin boyut tanımlamaya ihtiyacı yoktur çünkü tablodan farklı bir dosyada tutulurlar.Bu tipi Delphi'de kullanmak için OLE nesnesini kullanabilirsiniz.

Logical:Logical tipi TRUE yada FALSE tipinde iki değer tutabilir.Delphi'de Boolean tipi ile birlikte kullanabilirsiniz.

AutoIncrement:AutoIncrement tipi sürekli artan bir sayıyı ifade eder her kayıt eklendiğinde sahanın değeri bir artar. Böylece tekrarsız kayıtları oluşturulmuş olur. Genellikle Birincil Anahtar olarak kullanılır.Delphi'de LongInt tipi ile birlikte kullanabilirsiniz.

Binary:Binary tipi diğer tiplerle tutulamıyacak olan verilerin (genellikle kullanıcı tanımlı verilerin) tutulması için kullanılır tüm ASCII tabloyu kullanabilir.Bu tip Mb uzantılı bir dosyanın içinde tutulur.Asıl tabloya bağlanır.

Bytes:Bytes tipi genellikle Barkod yada manyetik şeritleri tutmak için kullanılır.Bu tip Mb uzantılı bir dosyanın içinde tutulur.Asıl tabloya bağlanır.

Size hücresine hangi veriyi girmişseniz o veri ile ilgili boyutu girmelisiniz. tanımlama ekranının alt kısmında bu konuda bilgi vermektedir.

Key hücresi Anahtar saha olup olmadığını denetler. Eğer bu sahanın üzerini çift tıklarsanız o sahayı Anahtar saha yapmış olursunuz.

Veritabanı Öğelerine Giriş

Tablolara erişmek için Delphi bir dizi öğe kullanir bir uygulamada eğer uygulama içinde tablonuza yeni kayit eklemek, tablonuzdan kayit silmek yada kayitlarinizi değistirmek istiyorsaniz en az üç adet öğelik combine kullanmaniz gerekecek. Bunlardan biri TTable ögesidir.

Table öğesi bir tabloya ulaşmak için kullanabileceğiniz en sorunsuz ve en kisa yoldur. Hiç bir kod yazmadan sadece Object Inspector yardımı ile bir tablo ile bağlanti kurabilirsiniz. Burada önemli olan tablonuzun bir alias altinda tanimli olmasidir.Eğer tanimli değilse bu tablonun içinde bulundugu dizinin yolunu girerekde tabloya ulaşmanız mümkün olabilir. Fakat bu programınızın taşınabilirliğini azaltır.

Bu öge çalisma aninda görünmeyen bir öğedir. Bu öğe ile bagladiginiz bir tabloyu elbette düzenlemek,tabloya kayit eklemek yadatablodan kayit silmek isteyeceksiniz. Ben bu bölümde TDBComponent'in bir üyesi olan TDBGrid'i kullanacagim. Çünkü bu öğe ile tablonuzda bulunan tüm kayitlari görebilir degistirebilir,ekleyebilir yada silebilirsiniz. Bu öge Delphi'nin **Data Controls** sekmesinin altindadir. Borland tablolarin birden fazla öge tarafindan kullanilabilirligini saglamak için degisik bir metot gelistirmistir. Bir TDataset kökenli veritabani ögesi ile (TTable,TQuery vs.) bir TDBcomponent ögeleri arasinda bir baglantinin kurulmasi için TDataSource ögesinin kullanilmasi gerekmektedir.

DataSource öğesi sadece ara istasyon olarak görev aln bir öğe pozisyonundadir ister yerel bir tablodan isterse uzak veritabani birimlerinden yada TClientDataSet ögesinden gelen bir veriyi isteyen Veritabani kontrollerine iletir. Istediginiz kadar çok kontrol baglayabilirsiniz. Simdi hep birlikte ilk veritabani uygulamamizi yaratalim. Bunun için TTable,TDataSource,TDBGrid ögelerini kullanacagiz. Ben bu uygulama örnegi için DBDemos aliasinin içinde olan bir tabloyu kullanmayi düsünüyüroum. Böylece sizde kendi bilgisayarinizda ayni uygulamayi gerçeklestirebilirsiniz.
yandaki sekildeki gibi bir form yaratin ve bu formun üzerine table,datasource ve dbgrid ögelerini yerlestirin. Formumuz çalistiginda table ve datasource öğeleri görünmez olacagi için bu öğeleri istediginiz bir yere yerlestirebilirsiniz. Table ögesini seçip object inspector'da özelliklerinin görüntülenmesini saglayin.Table ögesinin degistirilecek özellikleri asagidaki gibidir(ögenin özelliklerini degistirirken lütfen sirayi takip edin):

Table ögesinin bu özelliklerini sirasi ile ayarladiktan sonra Datasource ögesinin özelliklerini ayarlamamiz gerekiyor.Datasource ögesini seçerek özelliklerinin görüntülenmesini saglayin.Dataset özelligini Table1 olarak atayin. Böylece herhangi bir DBComponent'i Table1 ögesine baglama imkanina kavusacagiz.

DbGrid ögesini özelliklerinden Datasource özelligini Datasource1 olarak ayarlayarak uygulamamizi tamamliyoruz. Asagidaki sekilde bir görüntü olusmazsa mutlaka bir seyi eksik yapmissinizdir demektir

uygulamayi çalistirdiginizda bir kayit eklemek isterseniz klavye tuslarindan yaralanabilirsiniz.Insert tusu bir kayit eklemenizi F2 tusu üzerinde bulundugunuz kaydi degistirebilmenizi saglar.Ctrl+Delete tusu üzerinde bulundugunuz kaydi simenizi saglar. Kayitlar arasinda dolasmak istiyorsaniz ok tuslarini ve tab tusunu kullanabilirsiniz.Eger kayitlarin içinde daha rahat dolasmak istiyorsaniz TDbNavigator ögesini ekleyebilir ve Datasource özelligini Datasource1 yapabilirsiniz. Dikkat ettiyseniz bu uygulamada bir satir kod bile yazmadik. Bu Delphi'nin bize sagladigi öge gelistirme teorisinin bir armaganidir.Fakat her veritabani uygulamasi bu kadar kolay hazirlanmayabilir özellikle birbiri ile bagli tablolarda baglantili tablodaki bir veriye göre anatablodan bir veri seçmemiz gerektiginde bir kaç satir kod daha yazmamiz gerekir. Bunun için TQuery ögesinin kullanilmasi gerekir.
TQuery ögesi ile tablolar birbiri ile birlestirilebilir iç içe sorgular yaratilabilir. TQuery ögesi TTable ögesinden daha yavastir. Bunu çok kayitli bir veritabaninda rahatlikla görebilirsiniz.Fakat ag üzerinde bir tablonun kayitlarina ulasilacaksa yada bir internet sunucusundan kayitlar sorgulanacaksa TQuery ögesi Table ögesinden daha hizli olabilir. Bu olusumun sebebi Table nenesinin tablodaki tüm kayitlara erismesi sonucu veri trafigini çok arttirmasidir.Query ögesi ise ag üzerinde sorguyu Ulasilan sunucunun veritabani motoruna yollayip istenilen kayitlari sorgulamasini ister.Dönen kayit miktari, eger tüm kayitlarin listelenmesi istenmemisse, Table ögesinin döndürecegi miktardan daha azdir.
Size Veritabani uygulamasi gelistirme asamasinda verebilecegim tek ögüt gerekli olmadikça tüm kayitlari kullaniciya göstermeyin. sadece tabloda ekleme silme ve degistirme yapacakca tüm kayitlari görmesine gerek yok. Insanlarin dogasi geregi belirli bir miktar bilgiyi bir anda kafalarinda sorgulayabilirler. Bu miktardan fazlasi kullanici için gereksizdir. TQuery ögesi SQL dilini bilmenizi gerektirir. TQuery ögesi SQL anlatidiktan sonra açiklanacagi için burada deginmeyecegim.Table ögesi bittikten sonra DBComponent kontrollerinin kullanimina geçecegim ardindan SQL dilini altatan bir Makale dizisiyi yayinladiktan sonra Query ögesi ile Veritabani uygulamasi gelistirmek adli makale dizisine geri dönecegim.