

MICROSOFT ACCESS DERS NOTLARI

Access bir **veritabanı** programıdır. Veritabanı, bir konuyla ilgili çok sayıda kaydın tutulduğu bir bilgi havuzu olarak nitelendirilebilir. Veritabanı programları, veritabanı denilen bilgi havuzunu oluşturmak ve veritabanından istenilen bilgiyi istenildiği şekilde alıp kullanabilmeyi sağlar.

Access programına girdiğimizde ilk gelen ekranda 3 ayrı seçenek bulunur. Yeni bir veritabanı oluşturmak istiyorsak bunlardan **Boş veritabanı** seçeneğini seçip Tamam düğmesine basılır. Daha sonra veritabanına bir dosya ismi vermemizi isteyen bir ekran gelir. Burada veritabanının hangi klasöre kaydedeceğimizi seçer ve dosya adı olarak da bir isim yazarız. Örneğin dosya adı olarak **öğrenci** ismini verelim.

Açılan veritabanında şu bölümler bulunur :

- **Tablolar** : Veritabanının temel nesnesi tablolardır. Bilgilerin asıl tutulduğu yer tablodur. Diğer veritabanı nesneleri tablolar esas alınarak oluşturulur. Bir veritabanında birden çok tablo bulunabilir.
- **Sorgular** : Tablolardaki çok sayıda kayıt içerisinde istenilen kriterlere uyan kayıtları seçerek görebilmek için oluşturulan bir nesnedir.
- **Formlar** : Tablolara doğrudan bilgi girişi yapmak daha zor ve sıkıcı olabilir. Formlar tablolara bilgi girişini kolaylaştıran ve daha anlaşılır bir ekran görüntüsü ile çalışmayı sağlayan nesnelerdir. Paket programlardaki kullanıcı ara yüzü olarak düşünülebilir.
- **Raporlar** : Tablolardaki bilgileri kağıda dökülebilmek için değişik şekillerde sayfa dizaynları oluşturmak için kullanılır.
- **Makrolar** : Veritabanında birden çok adımdan oluşan bir işlemin bir seferde yapılabilmesini sağlayan küçük program parçalarıdır.
- **Modüller** : Makrolar ile aynı amaca sahip olmakla birlikte Visual Basic programlama dili komutları ile yazılan küçük program parçalarıdır.

TABLO OLUŞTURMAK

Veritabanında ilk önce tablo oluşturmak gerekir. Tablo oluşturmak için Tablolar bölümünde iken sağ taraftaki **Yeni** düğmesine basılır. Burada tablo oluşturma seçenekleri gelir. Tablo alanlarını ve alan özelliklerini tek tek kendimiz belirleyeceksek **Tasarım görünümü** seçeneği seçilerek Tamam düğmesine basılır.

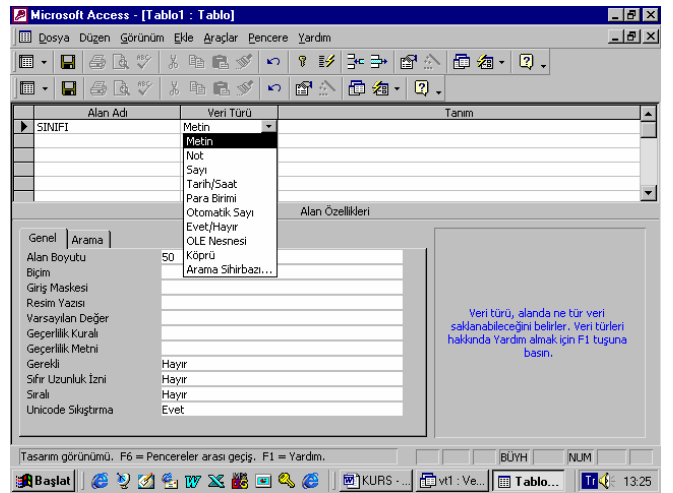
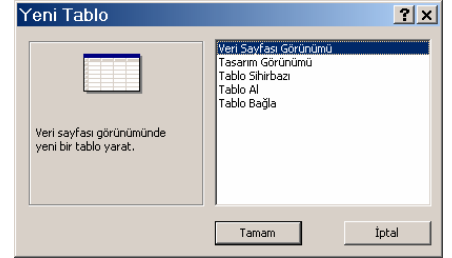
Tasarım görünümünde tablo hazırlarken tablonun başlıklarını oluşturacak her bir alanın **alan adını**, **veri türünü** ve **alanın özelliklerini** ayrı ayrı belirleriz.

Alan adı verilirken boşluk verilebilir, Türkçe noktalı harf kullanılabilir. Alan adı en fazla 64 karakter uzunluğunda olabilir.

Veri türleri

O alana girilecek bilginin türünü belirtir. Veri türleri şunlardır :

- **Metin** : Alfabetik ve sayısal türde her türlü bilgi girilebilen alan türüdür. En fazla 255 karakter bilgi alır.
- **Not** : Bir tablo alanı için metin türünün uzunluğu yetmiyorsa Not türü seçilebilir. Bu tür alanlara 65535 karaktere kadar bilgi girilebilir.
- **Sayı** : Tabloda üzerinde işlem yapılabilecek sayısal değerlerin girilebileceği alan türüdür. Sayısal alanların alan boyutunu belirlerken aşağıdaki boyut türleri seçilebilir :
 - Bayt : 0 ile 255 arasında değer girilebilir.
 - Tamsayı : -32768 ile 32767 arasında değer girilebilir.
 - Uzun tamsayı : -2.147.483.648 ile 2.147.483.647 arasında değer girilebilir.
 - Tek ve Çift : Çok büyük sayılar için ve ondalıklı sayılar için kullanılan veri tipleridir.
- **Tarih/Saat** : Tarih ve saat türündeki bilgiler için tanımlanır.



- **Para birimi** : Parasal değerler için tanımlanır. Bu tür alanlara 15 basamak tamsayı, ondalıktan sonra da 4 basamak bilgi girilebilir.
- **Otomatik sayı** : Bu tür alandaki sayı değeri girilen her kayıta otomatik olarak artar.
- **Evet/Hayır** : Mantıksal alan türüdür. İki ihtimalli bilgiler için kullanılır. Örneğin; Askerliğini yaptı/yapmadı, disiplin cezası var/yok gibi.
- **Köprü** : İnternet üzerinden bağlantı kurulacak web adresleri girilecekse bunun için köprü türü alan tanımlanır. Bu alandaki bilgiye tıklandığında eğer internet bağlantısı varsa belirtilen web adresine gidilebilir.
- **OLE nesnesi** : Veritabanına resim, ses, film gibi nesnelerin eklenmesi için kullanılır.

Alan biçimi

Bir alana girilen bilginin o alandaki yerleşimi ve görünümü alan özelliklerinde biçim özelliği ile gösterilir. Özellikle metin türündeki alanların biçimini tanımlarken aşağıdaki karakterler kullanılabilir :

KARAKTER	İŞLEVİ
@	Bir karakterlik bilgi için kullanılır.
<	Büyük harfle girilen bilginin küçük harfte görünmesini sağlar.
>	Küçük harfle girilen bilginin büyük harfle görünmesini sağlar.

Giriş Maskesi

Bir alana bilgi girişi yaparken kullanıcının bilgi girmesine yardımcı olmak ve sınırlama getirerek yanlış bilgi girilmesini önlemek için oluşturulan bir yapıdır. Giriş maskesi tanımlamak için şu karakterler kullanılabilir:

KARAKTER	İŞLEVİ
0	0 ile 9 arasında bir değer girilmesine izin verir. Bu tanımlama yapıldığında bilgi girişi zorunludur.
9	Bu tanımlama yapıldığı zaman bilgi girişi zorunlu değildir. Veya daha az rakam girilebilir. Rakamlar arasında boşluk verilebilir. Bu alana hiç bilgi girilmeden geçilebilir.
#	Tek bir rakam veya boşluk girilmesi için kullanılır. Bilgi girişi zorunlu değildir. Tanımlanandan daha az veri girilebilir.
L	Sadece alfabetik karakterlerin girişine izin verir. Tanımlandığı adet kadar bilgi girişi zorunludur. Karakterler arasında boşluk verilemez.
?	Sadece alfabetik bilgi veya boşluk girişine izin verir. Tanımlandığı adetten daha az bilgi girilebilir.
A	Sadece harf ve rakam girişine izin verir. Tanımlandığı sayı kadar bilgi girişi zorunludur.
a	Harf, rakam ve boşluk girişine izin verir. Tanımlandığı kadar bilgi girişi zorunlu değildir.
&	Herhangi bir karakter girişi yapılabilir. Tanımlandığı adet kadar bilgi girişi zorunludur.
C	Herhangi bir karakter girişi yapılabilir. Tanımlandığı adet kadar bilgi girişi zorunlu değildir.

Resim Yazısı

Bilgi girişi sırasında, bilgi alanları tanımlanmış adlarıyla görüntülenir. Eğer bilgi alanının başlığının başka bir metin olması isteniyorsa bu başlığın resim yazısı kutusuna yazılması gerekir.

Varsayılan değer

Hazırlanan bir tabloya bilgi girişi yapılırken o alanda başlangıçta görüntülenmesi istenen değerdir. Bilgi girişi yapılırken

Geçerlik Kuralı

Bir alana girilecek bilgiye giriş sınırlaması getirmek için kullanılan bir özelliktir. Örneğin 0 ile 100 arasında olmasını istiyorsak, “**BETWEEN 0 AND 100**” veya “**>=0 AND <=100**” şeklinde değer girilebilir.

Geçerlik metni

Geçerlik kuralı verilen alana bilgi girişi sırasında girilen bilgi bu kurala uymuyorsa ekranda görülecek hata uyarısı bu özellikte belirtilir.

Gerekli

Tablodaki bir alana bilgi girişini zorunlu tutmak istiyorsak bu özellik için **Evet** seçili olmalıdır.

Sıfır uzunluk izni

Bu özellik sadece metin türü alanlar için geçerlidir. Bu özellik **Evet** değerini taşırsa ilgili alana bilgi girişi yapmadan geçilebilir.

Sıralı

Tablo kayıtlarının o alandaki bilgiye göre sıralanıp sıralanmayacağı bu özellikte belirtilir. Burada **Evet (yineleme var)** seçili ise o alana göre sıralama olur ve o alanda aynı değeri taşıyan birden fazla kayıt bulunabilir. **Evet (yineleme yok)** seçili ise olana göre sıralama yine olur, ancak o alana aynı değeri taşıyan birden fazla kayıt girilemez.

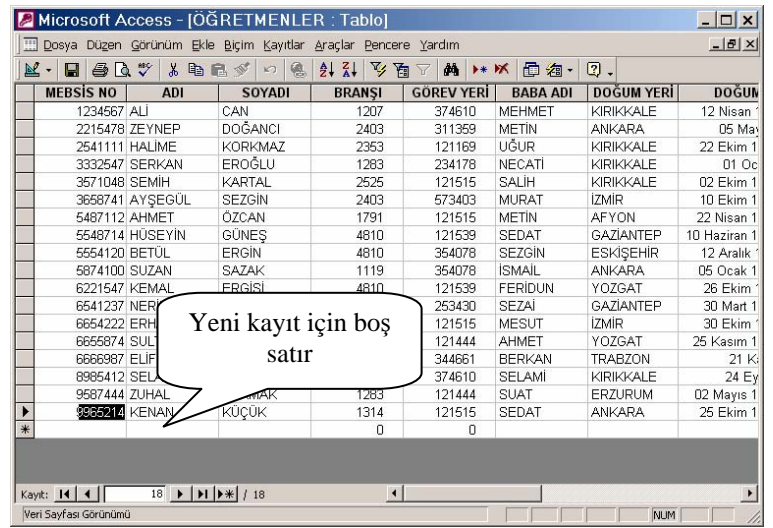
Unicode Sıkıştırma

Bu özellik 2 byte'lık bilgiyi 1 byte yer tutacak şekilde sıkıştırır. Bu özellik sadece metin türü alanlarda geçerlidir.

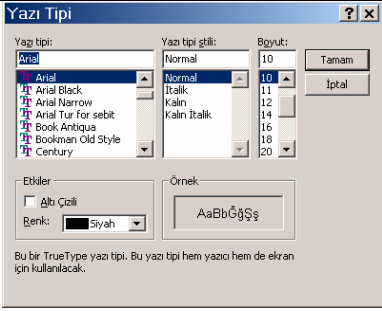

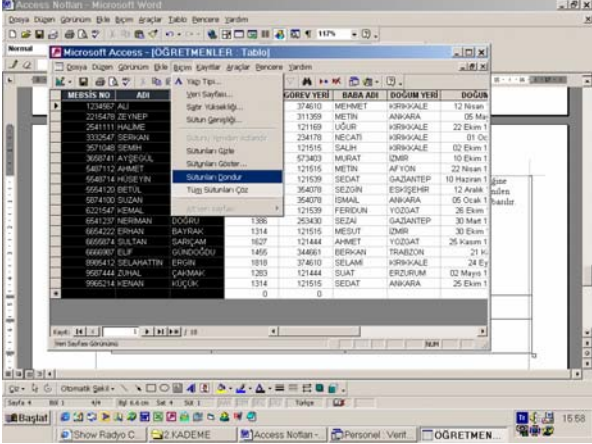
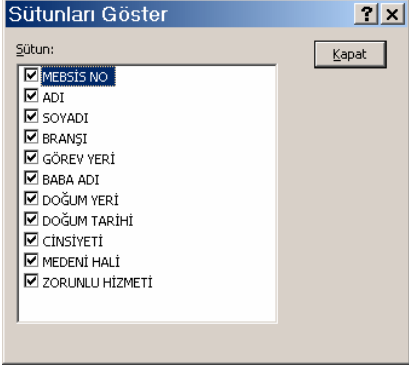
TABLOYA KAYIT GİRMEK

Access tablolarına bilgi girişi doğrudan tablodan yapılabilir. Bunun için bilgi girilecek tablo seçilip **Aç** düğmesine basılır. Tabloda her satır ayrı bir kayıt tutar. En son kayıttan sonraki boş satır yeni kayıt eklemek için kullanılır. Bu son satıra yeni bir kayıt girilmeye başlandığında o satırın bir altına tekrar boş bir satır ekler. Yeni kayıt girişi böylece devam eder. Her alana bilgi girildiğinde **Enter** tuşuna basarak bir sonraki alana geçilebilir.

Bir tablo üzerinde çalışırken tablo üzerinde ve kayıtlar üzerinde yapılabilecek bazı işlemleri şu şekilde açıklayabiliriz :



Tabloya yeni kayıt ekleme		Yeni kayıt eklemek için bu düğmeye basıldığında tablonun en son satırındaki boş kayıt satırına gider.
Tablodan kayıt silme		Tablodan herhangi bir kaydı silmek için o kayda gidip bu düğmeye basılır. Silme işlemine onay verilip silinir. Birden fazla kayıt silinecekse fareyi tablonun sol tarafından basılı tutup sürükleyerek seçilir ve yine bu düğmeye basılarak silinir.
Tablodaki kayıtları sıralatma		Tablonun hangi alanına göre sıralama yapılacaksa o alana geçilip bu düğmelerden birine basılır. (Artan sıralama veya azalana sıralama)
Tablodaki kayıtları seçime göre süzme		Tabloda herhangi bir alandaki bilgiye göre süzme yapılacaksa o alandaki bilgi fare ile sürükleyerek seçilir ve bu düğmeye basılır. Süzgeci kaldırmak için ise Süzgeci kaldır düğmesine basılır.

Tablodaki yazı tipini değiştirme		Biçim menüsü açılıp Yazıtipi seçeneğine girildiğinde bu ekran gelir. Buradan istenilen yazıtipi, yazıtipi stili, boyutu, rengi özellikleri seçilip Tamam düğmesine basılır.
Tablo sayfasının biçimini değiştirme		Biçim menüsünden Veri sayfası seçeneğine girildiğinde bu ekran gelir. Buradan istenilen özellikler belirlenip Tamam düğmesine basılır.
Tablodan istenilen bir sütunu dondurma		Sütunları dondurmak için tablonun sol tarafından istenilen sütunları seçtikten sonra Biçim menüsünden Sütunları dondur seçeneği seçilir. Tabloda sağa doğru kayarken bu sütunlar kaybolmaz.
Tablodan istenilen sütunları gizleme ve görüntüleme		Bunun için Biçim menüsünden Sütunları göster seçeneğine girildiğinde yandaki gibi bir liste gelir. Bu listeden istenilen sütunların işareti kaldırılıp gizlenebilir veya gizli sütunlar seçilerek geri getirilebilir.

ARAMA SİHİRBAZINI KULLANMAK

Bir tablonun tasarımı sırasında veri türü bölümünde listeyi açtığımızda **Arama Sihirbazı** adında bir tür görürüz. Bu aslında bir veri türü değildir. Amacı, tablo veya form üzerinde bu alana gelindiğinde bir liste kutusu görünmesini ve bu alana girilecek bilginin bu listeden seçilmesini sağlamaktır.

Arama sihirbazını kullanırken o alandaki liste kutusunda görülecek bilgiler iki şekilde oluşturulabilir. Birincisi bu bilgilerin elle yazılmasıdır. İkinci ise başka bir tablo oluşturup o tabloya girilen bilgilerden yararlanmaktır. İkinci yöntemde oluşturulan diğer tablo bir **Sözlük Tablosu** niteliği taşır. Yani bu tablo bir yardımcı tablodur ve devamlı açılıp kullanılan bir tablo değildir.

Şimdi bir örnek ile Arama Sihirbazının kullanımını açıklayalım :

BRANŞI ALANI İÇİN : Öğretmenler tablosunda BRANŞI alanına öğretmenin branş kodunun girileceğini varsayalım. Bunun için BRANŞLAR adında yardımcı bir tablo oluşturalım. Bu tabloda BRANS KODU ve BRANS ADI alanları olsun ve BRANS KODU alanı her branşta farklı olacağı için birinci anahtar olsun.

Daha sonra Öğretmenler tablosunun tasarımına geçip BRANŞI alanını veri türü listesini açarak Arama Sihirbazını seçelim. Sihirbazın ilk ekranında **Değerler tablo veya sorgudan alınsın** seçeneğini seçelim.

Bir sonraki adımda **Branşlar** tablosunu seçip yine İleri düğmesi ile devam edelim. Sonraki adımda **Branş Kodu** ve **Branş Adı** alanlarını şekilde görülen düğmeler ile sağ tarafa atalım.

Daha sonraki adımda Branş Kodu alanı birinci anahtar olduğu için listede görünmemektedir. Branş kodu alanındaki bilgi üzerinde çalıştığımız alana girileceği için bu seçeneği kaldırıyoruz. İleri ile devam.

Son adımda Etiketinin ne olmasını istersiniz sorusu geliyor. Bu soruyu o alan için başka bir etiket yazılabileceği gibi aynen de kalabilir. Son düğmesi ile işlem tamamlanır.

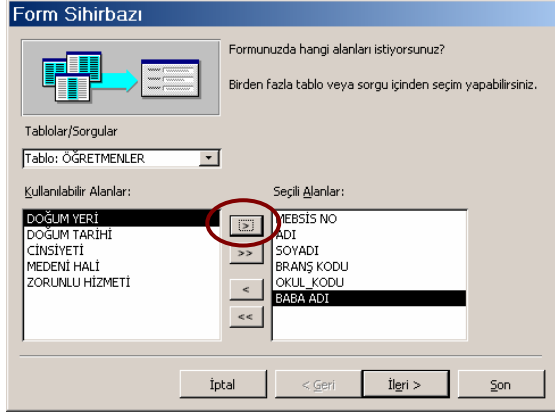
Bu işlem sonunda Öğretmenler tablosunun **Branş alanı** ile **Branşlar** tablosunun **Branş Kodu** alanı arasında bir ilişki kurulmuş olur. Bunu görmek için **Araçlar** menüsünden **İlişkiler** bölümüne girebilirsiniz.

FORM OLUŞTURMAK

Form, tabloya bilgi giriş yapmanın başka bir yoludur. Paket programlardaki kullanıcı ara yüzü gibi ekranlar oluşturulur.

Form Sihirbazı ile form oluşturma

- 1- Formlar bölümündeki **Yeni** düğmesine basılır.
- 2- Gelen ekranda önce alttaki liste kutusundan kendisi ile ilgili form hazırlanacak olan tablo seçilir. Daha sonra **Form sihirbazı** seçeneği seçilerek Tamam düğmesine basılır.



Formunuzda hangi alanları istiyorsunuz?
Birden fazla tablo veya sorgu içinden seçim yapabilirsiniz.

Tablolar/Sorgular
Tablo: ÖĞRETMENLER

Kullanılabilir Alanlar:

DOĞUM YERİ	DOĞUM TARİHİ	CİNSİYETİ	MEDENİ HALİ	ZORUNLU HİZMETİ
------------	--------------	-----------	-------------	-----------------

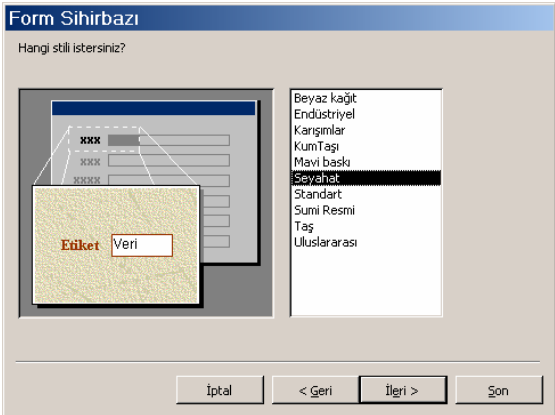
Seçili Alanlar:

MEBSİS NO	ADI	SOYADI	BRANŞ KODU	OKUL KODU	BABA ADI
-----------	-----	--------	------------	-----------	----------

İptal < Geri İleri > Son

- 4- Daha sonra alanların form üzerine nasıl yerleşeceğini belirleyen seçeneklerden biri seçilir ve yine İleri düğmesine basılarak devam edilir.

- 5- Sonra formun zemin rengi ve alanların renkleri ile ilgili seçenekler içeren **Hangi stili istersiniz** sorusu gelir. Bu ekranda istenilen stil seçilir ve İleri düğmesine basılarak devam edilir.



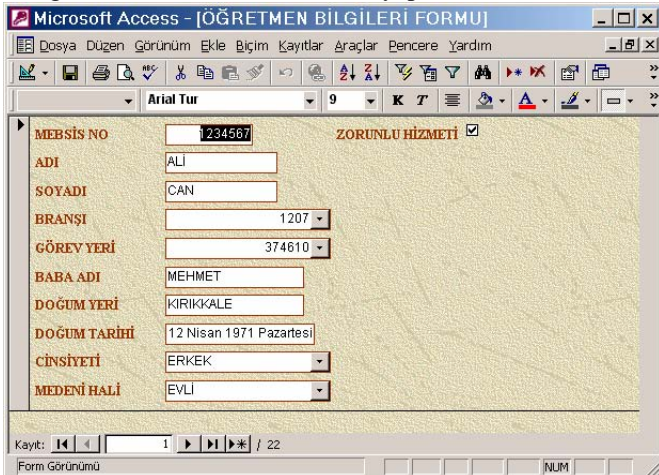
Hangi stili istersiniz?

Beyaz kağıt
Endüstriyel
Karışık
Kum Taşı
Mavi baskı
Seyahat
Standart
Sumi Resmi
Taş
Uluslararası

Etiket Veri

İptal < Geri İleri > Son

Form oluşup ortaya çıktığında artık bilgi girişi yapılabilir veya istenirse **Tasarım Görünümü**ne geçerek sihirbazın tasarladığı formun üzerinde düzenleme yapılabilir.



Microsoft Access - [ÖĞRETMEN BİLGİLERİ FORMU]

MEBSİS NO: 1234567 ZORUNLU HİZMETİ: [X]

ADI: ALI

SOYADI: CAN

BRANŞI: 1207

GÖREY YERİ: 374610

BABA ADI: MEHMET

DOĞUM YERİ: KIRIKKALE

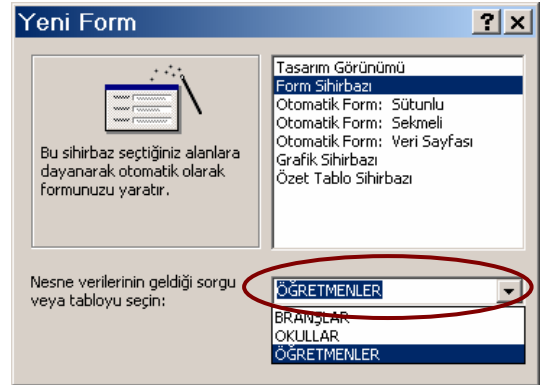
DOĞUM TARİHİ: 12 Nisan 1971 Pazartesi

CİNSİYETİ: ERKEK

MEDENİ HALİ: EVLİ

Kayıt: 1 / 22

Form Görünümü



Yeni Form

Bu sihirbaz seçtiğiniz alanlara dayanarak otomatik olarak formunuzu yaratır.

Nesne verilerinin geldiği sorgu veya tabloyu seçin:

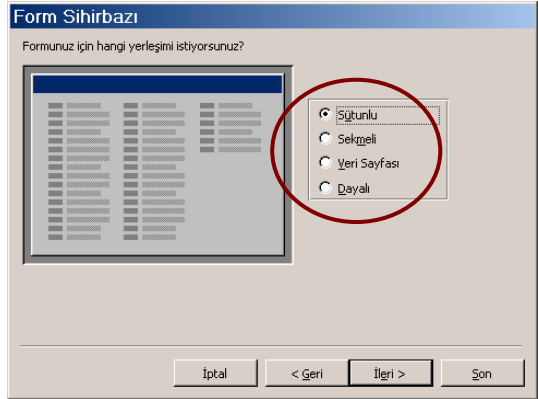
ÖĞRETMENLER

BRANŞLAR

OKULLAR

ÖĞRETMENLER

- 3- Sonraki ekranda form üzerinde görülecek alanlar sol taraftan sağ tarafa atılır. Bunun için iki bölme arasında bulunan sağa dönük oklar kullanılır. İleri düğmesine basılır.



Formunuz için hangi yerleşimi istiyorsunuz?

Sütunlu

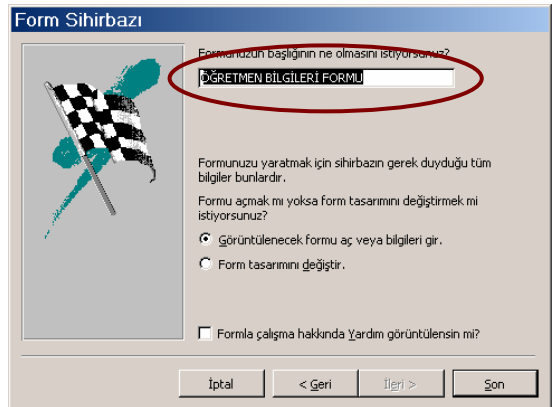
Sekmeli

Veri Sayfası

Dayalı

İptal < Geri İleri > Son

- 6- Son ekranda form başlığını yazıp Son düğmesine basarak formun oluşturulması sağlanır.



Formunuzun başlığının ne olmasını istiyorsunuz?

ÖĞRETMEN BİLGİLERİ FORMU

Formunuzu yaratmak için sihirbazın gerek duyduğu tüm bilgiler bunlardır.

Formu açmak mı yoksa form tasarımını değiştirmek mi istiyorsunuz?

Görüntülenecek formu aç veya bilgileri gir.

Form tasarımını değiştir.

Formla çalışma hakkında Yardım görüntülensin mi?

İptal < Geri İleri > Son

Alt Form İçeren Form Oluşturma

Birbiriyle ilişkili tablolara bilgi girişi yaparken her iki tablodaki bilginin birlikte görülebilmesi amacıyla form oluşturma sırasında alt form kullanılabilir.

Örneğin; Öğrenci bilgilerini içeren bir veritabanı çalışmasında öğrenci kişisel bilgileri ÖĞRENCİLER adındaki bir tabloda, bu öğrencilerin farklı derslerden aldıkları notlar da NOTLAR adındaki başka bir tabloda tutuluyor olsun. Öğrenci kişisel bilgilerinin tutulduğu tabloda her öğrencinin sadece 1 kaydı vardır. Ancak Notları tutan tabloda ise bir öğrencinin farklı derslerden notları olacağı için her öğrencinin aldığı ders kadar kaydı olacaktır. Bir form üzerinde öğrencinin OKUL NO, ADI, SOYADI gibi kişisel bilgileri ile birlikte her öğrencinin tüm derslerinden aldığı notları da birlikte görebilmek için alt form kullanmak gerekir.

Şimdi bu örneği uygulayarak anlatalım :

1- Formlar bölümünde Yeni düğmesine basıp ÖĞRENCİLER tablosunu seçin, Form sihirbazı seçeneğini de seçerek Tamam düğmesine basın.

2- Öğrenciler tablosundan OKUL NO, ADI, SOYADI alanlarını ortadaki ok ile sağ tarafa atın.

4- Bir sonraki adımda **göre ÖĞRENCİLER** ve **göre NOTLAR** şeklinde alt alta iki seçenek gelir. Burada Öğrenci bilgileri ana form, öğrencinin notları alt form olacağı için **göre ÖĞRENCİLER** seçili olmalıdır.

5- Yine aynı ekranda **Alt formları olan form** ve **Bağlı formlar** adında iki seçenek daha vardır. Bu örnekte ilkinii seçiyoruz. İstenirse diğeri de seçilebilir. İleri ile devam.

7- Daha sonra ana form için hangi stili istediğimiz sorulmaktadır. Buradan da istenilen seçilebilir.

3- Tablolar/Sorgular bölümünden NOTLAR tablosunu seçerek bu tablodaki tüm alanları da sağ tarafa atın. İleri ile devam.

6- Sonraki adımda alt form için nasıl bir yerleşim istendiği sorulmaktadır. Bu iki seçenekten yine istenilen seçilebilir.

8- Son aşamada ana form ve alt form için bir isim verilmesi istenir. Buralara da form içeriğine uygun birer isim yazılıp Son düğmesi ile işlem tamamlanır.

Alt Form İçeren Formun Kullanımı

OGRENCILER

OKUL NO: 101

ADI: ALI

SOYADI: CAN

NOTLAR:

Alt form

Ana formun kayıt düğmeleri

Alt formun kayıt düğmeleri

Ana formda bir öğrencinin okul numarası, adı ve soyadı görünürken alt formda bu öğrencinin aldığı tüm dersler ve bu derslere ait notları girilebilir ve incelenebilir.

Altındaki resimde de görüldüğü gibi veritabanı penceresinin Formlar bölümünde oluşturulan ana form ile alt form ayrı ayrı görülebilir. İstenirse ana formun tasarımı veya alt formun tasarımı değiştirilebilir.

Microsoft Access - [ÖĞRENCİ : Veritabanı]

Dosya Düzen Görünüm Ekle Araçlar Pencere Yardım

Nesneler

- Tablolar
- Sorgular
- Formlar
- Raporlar

Tasarım görünümünde form oluştur

Şablon kullanarak form oluştur

DERSLER

NOTLAR Altform2

OGRENCILER

ÖĞRENCİLERE GÖRE NOTLAR

Alt Form

Ana Form

Ana form ile alt formu birbirine bağlayan bir alan vardır. Örneğin yukarıdaki örnekte OKUL NO alanı hem Öğrenci bilgilerini gösteren ana formda hem de öğrenci notlarını gösteren alt formda mevcuttur. Alt forma bilgi girişi yapılırken ana formdaki OKUL NO bilgisi otomatik olarak alt formun OKUL NO alanına gelir. OKUL NO bilgisinin hem ana formda hem de alt formda görünmesi gereksizdir. Bu nedenle alt formun tasarımına girip OKUL NO alanını formdan silebiliriz.

KİŞİSEL BİLGİLERİ

OKUL NO: 101

ADI: ALI

SOYADI: CAN

NOTLAR:

DERS KODU	N1	N2	N3	N4	N5	N6	ORTALAMA
MAT03	100	12	22	54	58	54	50
MAT02	0	0	0	0	0	0	
TUR01	0	0	0	0	0	0	
TUR02	0	0	0	0	0	0	
*	0	0	0	0	0	0	

Kayıt: 1 / 4

Form Görünümü

SORGU OLUŞTURMAK

Sorgular, tablolardaki çok sayıda kayıt içerisinde istenilen kriterlere uyan kayıtları süzerek görmemizi sağlayan veritabanı nesneleridir.

Tasarım görünümünde sorgu oluşturma

- 1- Sorgular bölümünde **Yeni** düğmesine basılır.
- 2- Gelen ekranda **Tasarım görünümü** seçili iken

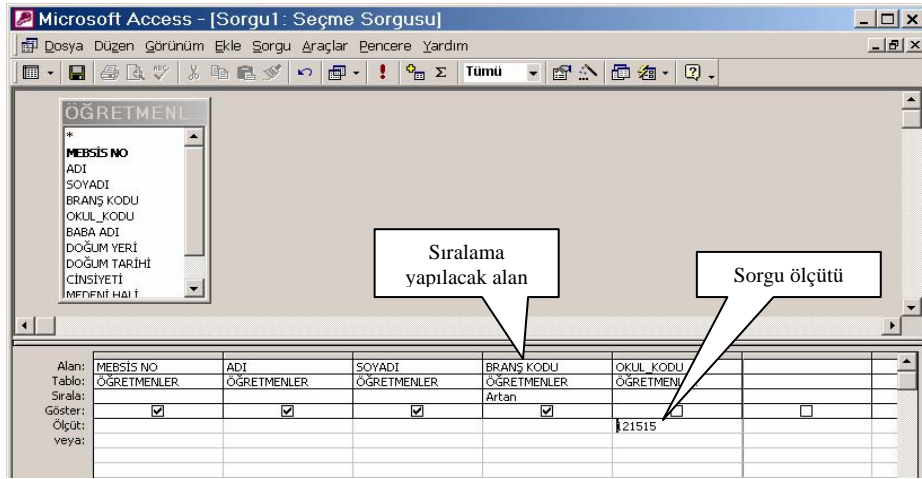
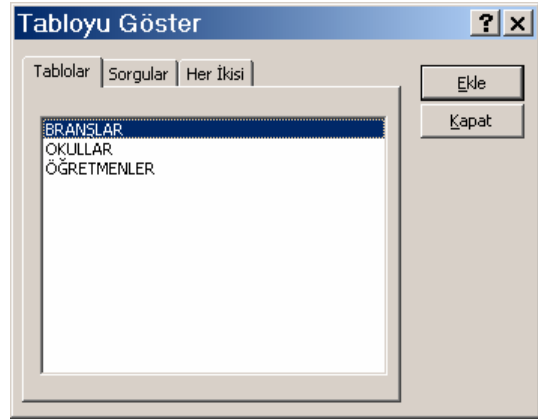
Tamam düğmesine basılır.

3- Sorgu tasarımına girildiğinde sorguya hangi tabloları ekleyeceğimizi soran bir pencere açılır. Bu pencerede sorguda kullanılacak tablolar ayrı ayrı veya birlikte seçilerek **Ekle** düğmesine basılır. Daha sonra bu pencere kapatılır.

4- Sorgu ekranında istenilen tablodan sorguda yer almasını istediğimiz alanlar ayrı ayrı alınarak alt taraftaki sorgu sütunlarına eklenir. Bunun için alan adından fare ile basılı tutup sürükleyerek aşağıdaki sütunlara indirilip bırakılabilir.

5- Sorgu sonucunun belli bir alandaki bilgilere göre sıralanmasını istiyorsak o alan için **Sırala** özelliği açılıp **Artan** veya **Azalan** seçeneği seçilebilir.

6- Sorgu sonucunda görünmesini istemediğimiz alanlar varsa bu alanlar için **Göster** kutucuğundaki işaret kaldırılabilir.



7- Sorgu sonucunda ilişkili tablodaki tüm kayıtları değil, belli bir kriterle uyan kayıtları görmek istiyorsak kriter belirtilecek alan için **Ölçüt** bölümüne istenilen kriter bilgisi yazılır. Eğer aynı alan için birden fazla ölçüt yazılacaksa alt alta bunlar yazılır. Bu durumda ölçütler arasındaki bağlaç **veya** olur. Eğer birden fazla alana ölçüt yazılırsa bu durumda ölçütler arasındaki bağlaç **ve** olur.

Örnek;

Alan:	MEBSİS NO	ADI	SOYADI	BRANŞ KODU	OKUL KODU		
Tablo:	ÖĞRETMENLER	ÖĞRETMENLER	ÖĞRETMENLER	ÖĞRETMENLER	ÖĞRETMENLER		
Sırala:				Artan			
Göster:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Ölçüt:					121515		
veya:					121539		

Bu sorgu tasarımının cümlesel ifadesi şu şekilde söylenebilir : Öğretmenler tablosundan **Okul kodu 121515 VEYA 121539** olan kayıtların MEBSİS NO, ADI, SOYADI, BRANŞ KODU alanlarındaki kayıtları **branş koduna göre artan sıralı** olarak görüntüle.

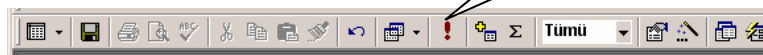
Örnek;

Alan:	MEBSİS NO	ADI	SOYADI	BRANŞ KODU	OKUL KODU		
Tablo:	ÖĞRETMENLER	ÖĞRETMENLER	ÖĞRETMENLER	ÖĞRETMENLER	ÖĞRETMENLER		
Sırala:				Artan			
Göster:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Ölçüt:				1314	121515		
veya:							

Bu sorgu tasarımının cümlesel ifadesi şu şekilde söylenebilir : Öğretmenler tablosundan **Okul kodu 121515 olan VE Branş kodu 1314** olan kayıtların MEBSİS NO, ADI, SOYADI alanlarındaki kayıtları **soyadına göre artan sıralı** olarak görüntüle.

Hazırlanan sorguyu çalıştırmak ve sonucunu görmek için araç çubuğundaki

basılabilir. Sorgu son halini almış ise kaydedilip saklanabilir.



Sorgu Ölçütlerinde Parametre Metni kullanımı

Sorgularda ölçüt verirken ölçüt ifadesini doğrudan yazıp kaydettiğimizde bu sorgu sadece o ölçüt için çalışır. Yukarıdaki örneklerde olduğu gibi okul kodu alanına ölçüt olarak tek bir okulun kodunu yazıp sorguyu böyle kullanırsak her seferinde sadece bu okuldaki öğretmenleri görüntüler. Bu şekildi düşündüğümüzde her okul için ayrı ayrı sorgu hazırlamak gerekecektir. Sorgu ölçütlerini genel hale getirebilmek için ölçüt alanına doğrudan bir ifade yazmak yerine **parametre metni** yazılıp kullanılabilir.

Alan:	MEBSİS NO	ADI	SOYADI	BRANŞ KODU	OKUL KODU
Tablo:	ÖĞRETMENLER	ÖĞRETMENLER	ÖĞRETMENLER	ÖĞRETMENLER	ÖĞRETMENLER
Sırala:			Artan		
Göster:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Ölçüt:					[Okul Kodunu Giriniz]
veya:					

Parametre metinleri **köşeli parantez** içerisinde yazılır. Parametre metni kullanıldığında sorguyu her çalıştırdığımızda bu metni içeren parametre giriş penceresi gelir. Bu pencerede her seferinde farklı bir değer girmek suretiyle sorguyu daha genel kullanabiliriz.

Parametre Değerini Gir

Okul Kodunu Giriniz

Sorguda Ölçüt yazarken kullanılabilecek bazı ifadeler

Sorguda herhangi bir alan için yazılan ölçüt ifadesi o alandaki bilgiye **tamamen eşit** olduğu taktirde sorgu çalışırken o kaydın görünmesini sağlar. Yani sorguda kullanılan alan adı ile ölçüt arasındaki karşılaştırma operatörü **eşittir** (=) işaretidir.

Bunun dışında sorgu ölçütünde şu işaretler de kullanılabilir :

Ölçüt Operatörü	Açıklaması
> büyüktür	Ölçüt yazılan alan türü: Sayı ise, belirtilen sayıdan büyük olan sayıların, Metin ise belirtilen harf veya harflerden sonra gelen metinlerin, Tarih ise belirtilen tarihten sonra gelen tarihlerin bulunduğu kayıtları getirir.
< küçüktür	Ölçüt yazılan alan türü: Sayı ise, belirtilen sayıdan küçük olan sayıların, Metin ise belirtilen harf veya harflerden önce gelen metinlerin, Tarih ise belirtilen tarihten önce gelen tarihlerin bulunduğu kayıtları getirir.
<= küçük eşit	Yukarıda belirtildiği gibi sayı, metin veya tarih türü alanlarda belirtilen ifadede küçük veya eşit olan kayıtları getirir.
>= büyük eşit	Yukarıda belirtildiği gibi sayı, metin veya tarih türü alanlarda belirtilen ifadede büyük veya eşit olan kayıtları getirir.
Between (Arasında)	Sayı, metin veya tarih türü alanlarda belirtilen iki değer arasındaki değeri taşıyan kayıtları getirir.
Like (İle başlar)	Ölçüt yazılan alanın veri türü metin ise, belirtilen harf veya harflerle başlayan kayıtları getirir.

Örnek Sorgu Ölçütleri

Örnek	Açıklaması
OKUL NO alanı için > 1000	Okul numarası 1000 den büyük olan kayıtları getirir.
HİZMET SÜRESİ alanı için >=10 and <=20	Hizmet süresi 10 a eşit veya büyük ve 20 ye eşit veya küçük olan kayıtları getirir.
ADI alanı için >= M	Adı M harfine eşit ve büyük olan kayıtları getirir. Yani adı M harfi ve sonraki harfler ile başlayan kayıtları getirir.
ADI alanı için Like "M**"	Bir önceki örnek ile aynı anlamı taşır.
OKUL NO alanı için Between 1000 and 2000	Okul numarası 1000 ile 2000 arasında olan kayıtları getirir.
DOĞUM TARİHİ alanı için Between #01/01/1980# and #01/01/193#	Doğum tarihi 01/01/1980 ile 01/01/1983 tarihleri arasında olan kayıtları getirir.
SOYADI alanı için Between "A" and "E"	Soyadı A harfi ile E harfi arasında başlayan kayıtları getirir.

İlişkili Tablolar İle Sorgu hazırlama

Veritabanında birbiriyle ilişkili birden fazla tabloyu kullanarak sorgu hazırlamak için sorgu tasarımına girildiğinde bu tablolar sorgu ekranına eklenmelidir. Tablolar sorgu ekranına getirildiğinde aralarındaki ilişkiler de burada görülebilir.

Birbiriyle ilişkili tablolardan ayrı ayrı alanları alıp sorguya eklemek mümkündür. Yukarıdaki ekranda yer alan

Microsoft Access - [Sorgu1: Seçme Sorgusu]

Dosya Düzen Görünüm Ekle Sorgu Araçlar Pencere Yardım

Tümü

BRANŞLAR

- * BRANŞ KODU
- BRANŞ ADI

ÖĞRETMENLER

- * MEBSİS NO
- ADI
- SOYADI
- BRANŞ KODU
- OKUL_KODU
- BABA ADI
- DOĞUM YERİ
- ...

OKULLAR

- * İLÇE_ADİ
- OKUL_KODU
- OKUL_ADİ

Alan:

MEBSİS NO	ADI	SOYADI	BRANŞ ADI	İLÇE_ADİ	OKUL ADI	OKUL_KODU
ÖĞRETMENLER	ÖĞRETMENLER	ÖĞRETMENLER	BRANŞLAR	OKULLAR	OKULLAR	ÖĞRETMENLER
Sırala:			Artan			
Göster:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Ölçüt:						
veya:						

[Okul Kodu Giriniz]

Hazır NUM

sorguda Branşlar tablosunun Branş Kodu alanı ile Öğretmenler tablosunun Branş Kodu alanı ilişkili olduğu için Branşlar tablosundaki Branş Adı alanı sorguya ekleniyor. Aynı şekilde Okullar tablosundaki Okul Kodu ile Öğretmenler tablosundaki Okul Kodu alanı ilişkili olduğu için Okullar tablosundan İlçe Adı ve Okul Adı alanları sorguya eklenebiliyor.

Bu ilişkilerle ilgili olarak şu cümle söylenebilir : Öğretmenler tablosunun Branş Kodu, Branşlar tablosundaki branş koduna eşit ise Branşlar tablosundaki Branş Adı alanını getir. Yine; Öğretmenler tablosundaki Okul Kodu, Okullar tablosundaki Okul Koduna eşit ise Okullar tablosundaki İlçe Adı ve Okul Adı alanlarını getir.

RAPOR OLUŞTURMAK

Rapor, tablolardaki bilgileri düzenli bir sayfa dizaynı oluşturarak kağıda dökmek için kullanılır. Raporlar doğrudan tabloyu esas alarak hazırlanırsa tablodaki tüm kayıtları getirir. Tablodan sadece istenilen kayıtları kağıda dökmek istiyorsak, raporu hazırlamak için bir **Sorguyu** esas almak daha uygun olacaktır.

Rapor sihirbazı ile rapor hazırlamak

- 1- Raporlar bölümünde **Yeni** düğmesine basılır.
- 2- Yandaki ekran geldiğinde önce alt taraftaki liste kutusu açılıp rapora kaynaklık edecek tablo veya sorgu seçilir. Daha sonra **Rapor sihirbazı** seçeneği seçilip Tamam düğmesine basılır.
- 3- Yine form sihirbazında olduğu gibi, tablo veya sorguda yer alan alanların listesinin bulunduğu ekran gelir. Buradan raporda kullanmak istediğimiz alanlar ortadaki sağa dönük oklar ile sağ tarafa atılır. İleri düğmesine basılarak devam edilir.

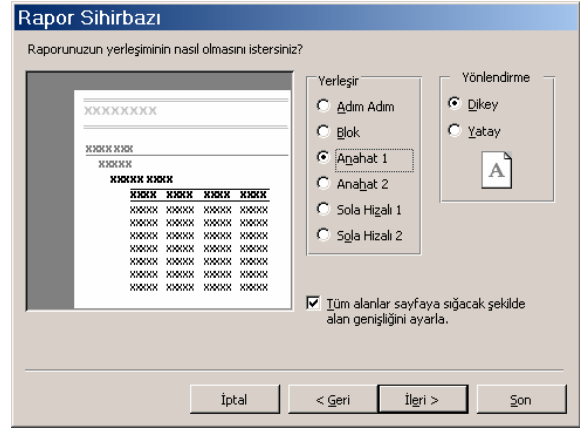
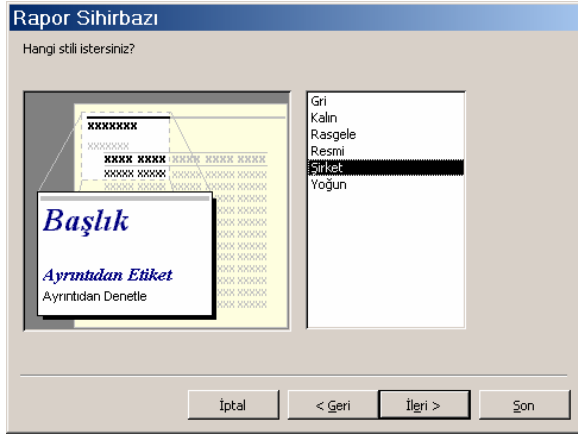
- 5- Bir sonraki ekranda, eğer rapor dökümünde gruplandırma yapmak istiyorsak hangi alana göre gruplandırma yapılacağı seçilir. İleri düğmesi ile devam edilir.

- 7- Daha sonra, raporun sayfa üzerindeki yerleşimini gösteren seçenekler yer alır. Bunlardan istenilen biri seçilir. Ayrıca sayfanın yatay veya dikey kullanımı da buradan değiştirilebilir. İleri düğmesi ile devam edilir.

- 4- Daha sonraki ekranda eğer raporu bir sorguya göre hazırlıyorsak ve sorguda birden fazla tablo varsa bu tablolardan hangisine göre listeleme olacağı sorulmaktadır. Buradan raporun özelliğine göre istenilen tablo seçilir. İleri düğmesine basılarak devam edilir.

- 6- Sonraki ekranda, rapordaki kayıtları belli alanlara göre sıralanmış olarak listelemek istiyorsak sıralama yapacağımız alanlar seçilebilir. Yine ileri düğmesi ile devam edilir.

8- Bir sonraki adımda raporun kağıda dökülüş stili ilgili seçenekler mevcuttur. Bunlardan istenilen biri seçilip yine devam edilir.



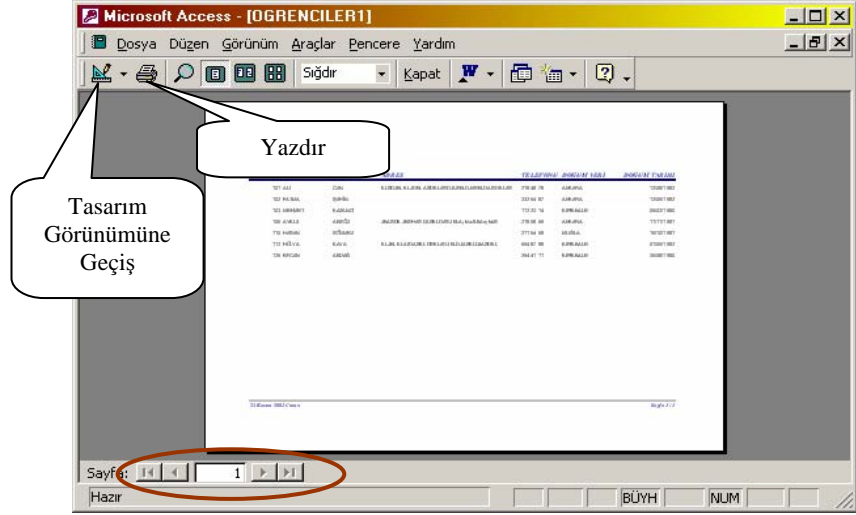
ile

9- Son adımda rapora bir ad verilir ve Son düğmesine basılarak raporun oluşması sağlanır.

Sorguya dayalı olarak hazırlanmış raporlarda, eğer sorguda **parametre metni** tanımlanmış ise, bu sorguyu esas alan raporun açılması sırasında da aynı parametre metni gelir.

Bir raporun **Baskı Önizleme** görüntüsü yandaki gibidir. Bu durumda **Yazdır** düğmesini kullanarak rapor yazıcıdan kağıda dökülebilir.

Sihirbaz kullanılarak oluşturulan bir raporun tasarım görünümüne geçerek rapor tasarımında değişiklik yapılabilir.



Yandaki örnek raporun tasarım görünümünde;

➤ Rapor üst bilgisi bölümünde görülen metin, bu raporun baskısı sırasında sadece ilk sayfanın üstünde görünür.

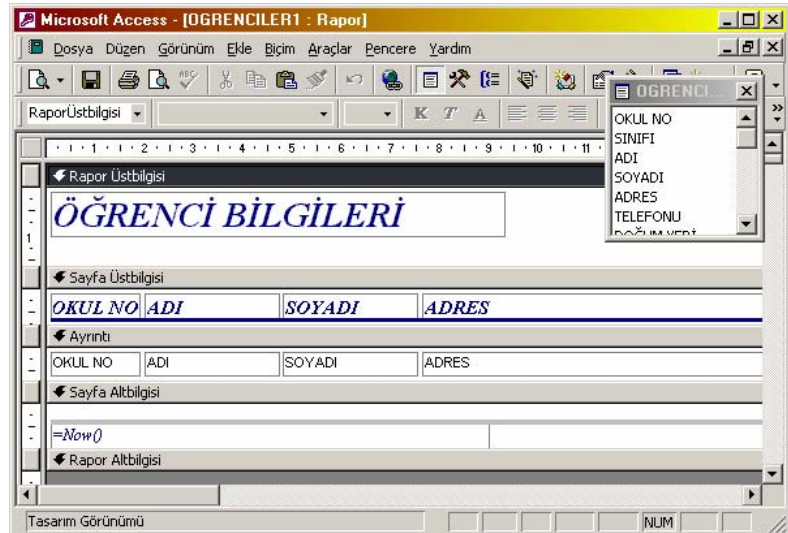
➤ Rapor alt bilgisi ise raporun son sayfasının altında çıkar.

➤ Ayrıntı bölümünde, görülen kutucuklar rapora kaynaklık eden tablonun alanlarıdır. Yani bu kısımda tablodan gelen bilgiler görülür.

➤ Sayfa üst bilgisindeki kutucuklar, ayrıntı bölümündeki bilgilerin başlıklarını oluşturan etiket metinleridir. Rapor tasarımında etiket metinleri istenildiği gibi değiştirilebilir.

➤ Sayfa alt bilgisi bölümünde her sayfanın altında görülmesi istenilen metinler bulunabilir. Örneğin rapor çıktısının alındığı tarih, sayfa numarası gibi.

Rapor tasarımında kullanmak istemediğimiz bölümleri alt kenarından fare ile tutup yukarı çekerek kapatabiliriz. Access'de **Veri Tabanı Tasarımı Oluşturmak ve Makro Atama**



Boş bir Veri Tabanı ile İşe başlayalım.Ä–nce **Dosya/Yeni Veritabanı** seçenekleri ile açılan pencereden **Boş Veri Tabanı** kullanın. Access veri tabanına vereceğiniz ismi verdikten sonra



Proje yönetim penceresine ulaşacaksınız.

Veri Tabanı Birçok bileşenden meydana gelebilir. Tablolarda Saklanacak Bilginin özellikleri ve bilgilerin kendisi bulundurulur.

Yukarıdaki Resimde Görünen pencerenin **Tablolar** kısmında iken **Yeni** düğmesini kullanarak yeni bir tablo oluşturalım.



Biz **Tasarım Görünümü** seçeneğini kullanarak tabloyu kendimiz tasarlıyalım.,

Üzerinde hesap yapılması gereken ve para olmayan alanlar bu tipten tanımlanırlar. Bu alana girilecek sayıların tipi ve sınırı *Genel* kısmındaki *Alan Boyu* ile belirlenir.

Bayt: 0 ile 255 arasında tam sayılar bu alanda tutulabilir. 1 byte yer işgal eder.

Tamsayı: -32,768 ile 32,767 arasındaki tam sayılar bu alanda tutulabilir. 2 byte yer işgal eder.

Uzun Tamsayı: -2,147,483,648 ile 2,147,483,647 arasındaki tam sayılar bu alanda tutulabilirler. 4 Byte yer işgal eder.

Tek: $-3,402823 \times 10^{38}$ ile $3,402823 \times 10^{38}$ arasındaki Ondalık sayılar bu alanda tutulabilir. 4 Byte yer işgal eder.

Çift: $-1.79769313486232 \times 10^{308}$ to $1.79769313486232 \times 10^{308}$ arasındaki ondalık sayılar bu alanda tutulabilir. 8 Byte yer işgal eder.

Girilen sayıların biçimi *Genel* kısmındaki *Biçim* kutusunu kullanarak açılan kutudan birisini seçebilirsiniz.

PARA BİRİMİ:

Aslında buda bir sayı veri tipidir. Ancak para hesapları düşünülerek geliştirilmiş. Para hesapları virgülden sonraki basamak sayı önemli değildir. Ā-nemli olan Virgülden önceki bütün basamakların tutulmasıdır. Be veri tipi tam kısmı 15 basamak ve virgülden sonrakikismi ise 4 basamak olan sayıları tutabilir. 8 Byte yer işgal eder.

OTOMATİK SAYI:

Bazı durumlarda her kayda bir numara girilmesini istediğinizde seçeceğiniz seçenektir. Sayının nasıl verileceğini ise *Genel* kısmındaki *Yeni Değerler* kutusundan belirleyebilirsiniz. 4 Byte yer işgal eder.

TARH/SAAT:

Doğum Tarihi, giriş saati gibi alanları text olarak tanımlarsınız bunlar üzerinde işlem yapamazsınız. Tarih veya saat ile ilgili bilgilerin bulunacağı alanları bu tipten tanımlamanız gerekir. 8 Byte yer işgal eder.

EVE/HAYIR:

Doğru/Yanlış, Evet/Hayır ve Açık/Kapalı gibi sadece iki durumlardan birinde bulunabilen alanları bu tipten tanımlamak gerekir. Çünkü bu alan sadece 1 bit (1/8 byte) yer işgal eder.

OLE NESNESİ:

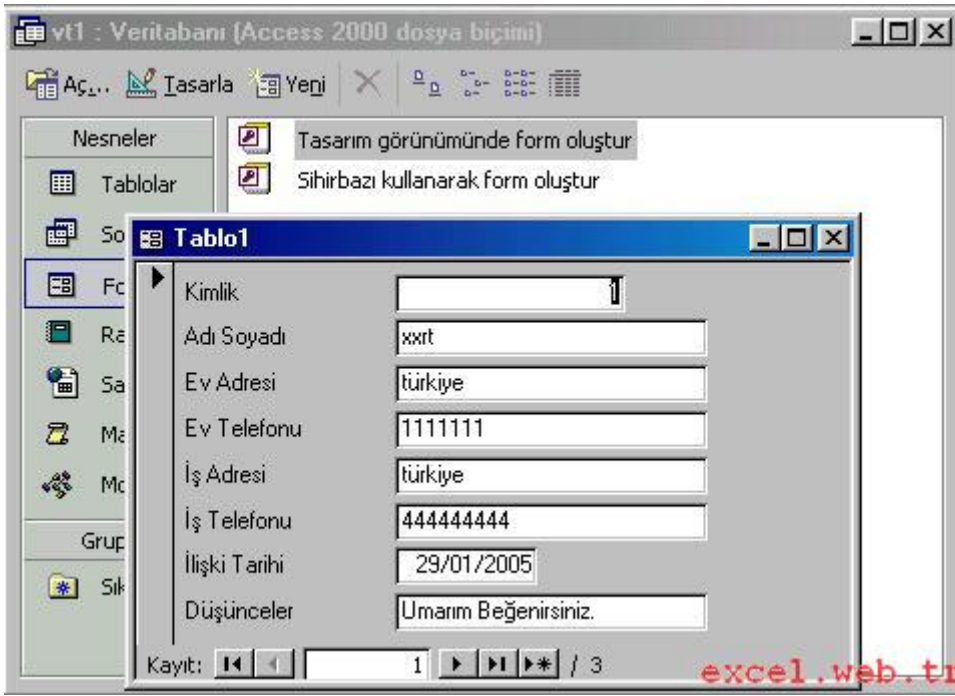
Veri tabanınıza resim, ses, animasyon, film gibi birçok şeyler ekliyebilirsiniz. Ā-rneğin paint Brushta çizilmiş bir resmi veri tabanınız ekliyebilirsiniz.

..-

Yine pencerenin *Genel* kısmındaki birçok özelliği kullanarak alanla ilgili bazı özellikleri kolayca belirleyebilirsiniz.

Eğer bir alana mutlaka bilgi girişi yapılması gerekiyorsa, örneğin müşterinin adı ve soyadının mutlaka girilmesi gerekiyorsa bu alan için *Gerekli* kutusunda evet seçeneğini verebilirsiniz. Diğer alanlarla ilgili özellikleri genişletebilirsiniz.

Ayrıca kullanıcının bilgi girişini kolaylaştırmak için bazı alanlara olabilecek değerlerin otomatik olarak yazılmasını sağlayabilirsiniz. Ā-rneğin satış tarihinin girilmesi gereken alan bu günün tarihini yazdırabilirsiniz. Bir alana varsayılan değeri vermek için pencerenin *Genel* kısmındaki *Varsayılan Değer* düğmesini kullanabiliriz.



ekrana gelecektir. Tasarım odunda ike ekrandaki alanların yerlerini ve etiketlerini değiştirebilirsiniz. Ayrıca yeni kontroller ekleyip kodlar oluşturabilirsiniz. Yukarıdaki Resim'de gördüğünüz alanlar iki parçadan oluşmaktadır. Soldakiler **Etiket** kutusu sağdakiler ise **Metin** Kutularıdır.

Tasarım Modu:

Tasarım Moduna alınan formda



excel.web.tr

Araç Kutusu Bulunmaktadır. Bu Araç kutusunu kullanarak yeni kontroller oluşturabiliriz.

..-

Araç çubuğunda bir kontrolü form üzerine yerleştirmek için o kontrolü araç çubuğundan seçtikten sonra formunuzun istediğiniz yerine fare ile çizmeniz yeterlidir.

Örnekte aşağıya doğru açılan bir kontrol yerleştirelim. Burada kontrolleri yerleştirirken dikkat edeceğimiz bir özellik

Yukarıdaki Resimde Yuvarlak içine alınan düğmenin basılı olmasına dikkat edin. Şimdi Araç Çubuğundan Açılan Listeyi seçelim ve formun üzerine çizelim. Çizdikten Sonra

Açılan Kutu Sihirbazı

Bu sihirbaz, içinden seçim yapabileceğiniz bir değerler listesi gösteren Açılan kutu öğesini oluşturur. Açılan kutu öğenizin, değerlerini nasıl almasını istersiniz?

☐ Açılan kutu, değerleri tablo veya sorgudan alsın.

☐ İstediğim değerleri ben yazacağım.

☒ Formumda, 1 değerine dayanan (Açılan kutu içinde seçili) bir kayıt bul.

İptal < Geri İleri > Son

Yukarıda görünen Sihirbaz devreyi girecektir. Aşağıya doğru açılan listeyi eleman bulmak için kullanacağımızdan yukarıdaki resimdeki pencereden son seçeneği işaretliyerek sonraki adıma geçelim..

Açılan Kutu Sihirbazı

Hangi alanlar, Açılan kutu listenizde içerilmesini istediğiniz değerleri içerir? Seçtiğiniz alanlar, Açılan kutu listenizde sütunlar haline gelir.

Kullanılabilir Alanlar:

Seçili Alanlar:

> >> < <<

Kimlik
Adı Soyadı
Ev Adresi
Ev Telefonu
İş Adresi
İş Telefonu
İlişki Tarihi
Düşünceler

İptal < Geri İleri > Son

Yukarıdaki görüleceği üzere İstenilen alanları sağ kısma alıyoruz. İleri tuşu ile devam ederek bu işlemi sonlandırdığımızda artık formumuzda aşağı doğru açılan bir liste kontrolü daha bulunuyor..

Tablo1

Kimlik	1
Adı Soyadı	xxrt Hızlı Giriş
Ev Adresi	türkiye
Ev Telefonu	1111111
İş Adresi	türkiye
İş Telefonu	444444444
İlişki Tarihi	29/01/2005
Düşünceler	Umarım Beğenirsiniz.

Kayıt: 1 / 3

excel.web.tr

Artık Buraya Kadar bazı özellikleri kendimiz dahada geliştirebiliriz.

KOD OLUŞTURMA :

Ä–nce Form üzerinde *İlişki Tarihi* kutusunu Tasarım Modunda iken seçin ve farenin sağ tuşu ile açılan Menüden

Tablo11 : Form

Form Üstbilgisi

Ayrıntı

Kimlik	Kimlik
Adı Soyadı	Adı Soyadı
Ev Adresi	Ev Adresi
Ev Telefonu	Ev Telefonu
İş Adresi	İş Adresi
İş Telefonu	İş Telefonu
İlişki Tarihi	İlişki Tarihi
Düşünceler	Düşünceler

Form Altbilgisi

Olay Oluştur...

Oluştur...

Değiştir

Sekme Sırası...

Kes

Kopyala

Yapıştır

Hizala

Boyut

Dolgu/Arka Plan Rengi

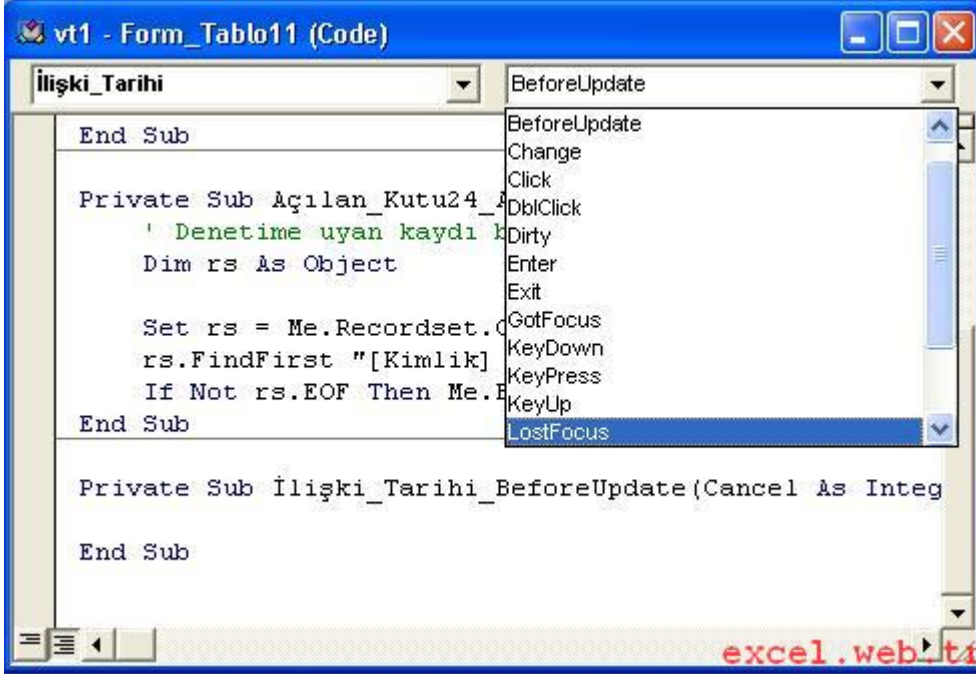
excel.web.tr

Olay Oluştur seçeneği seçin.Daha Sonra ekrana Gelen Tabloda



excel.web.tr

"Kod Oluştur" Seçeneğini işaretliyerek **Visual Basic** kodu oluşturalım.
Gelen Kod Penceresinde



Yordamdan LostFocus seçiyoruz. Bu pencere bir alt programdır. Visual dillerde program kodları olay dediğimiz alt programa yazılır. Bizim Yazacağımız kod giriş yapıldıktan sonra kontrol yapacağı için **LostFocus** olayına yazacağız. Aşağıdaki kodları oraya yazıyoruz.

```
[vb: 1:d453a7f944]Private Sub İlişki_Tarihi_LostFocus()
```

```
If Date < İlişki_Tarihi Then
```

```
MsgBox ("Girdiğiniz Tarihte Hata var")
```

```
End If
```

```
End Sub[/vb: 1:d453a7f944]
```

Buraya kadar Herşey tamam ise Tasarım modundan çıkın veya F5 Tuşu ile deneme yapın.



excel.web.tr

Burada ilişki Tarihine 1 yazdım ve Hata Mesajını aldık..

Sanırım buraya kadar bazı olayların gidişahı belli oldu.Sistem ve mantık aynı..

[Konuya Ait Örnek Çalışma İndirmek İçin](#)

Artık Bundan sonraki konular daha fazla ve değişik olarak bu sitemizde **devam edecektir**

Access Hikayeleri

Bu bölümde Access'le ilgili ipuçları, çözümler ve temel kurallarla ilgili bilgilerle beraber gelişmiş programlamaya daha çok ağırlık vermeye çalışacağız.

Başlıktaki 'hikaye' kelimesine gelince neden böyle bir başlık verdiğimi söyleyeyim. Bu yazı uzunca bir süredir Access ile çalışırken aldığım notların bir ürünü. Okuyuculardan gelen mesajlar doğrultusunda bu dizinin geri kalan kısmını şekillendireceğiz. Çok rasladığımız bazı sorunlar ve önemli ipuçları yazının devamında yer alıyor.

1. Otomasyon Hatası/ Derleme Hatası

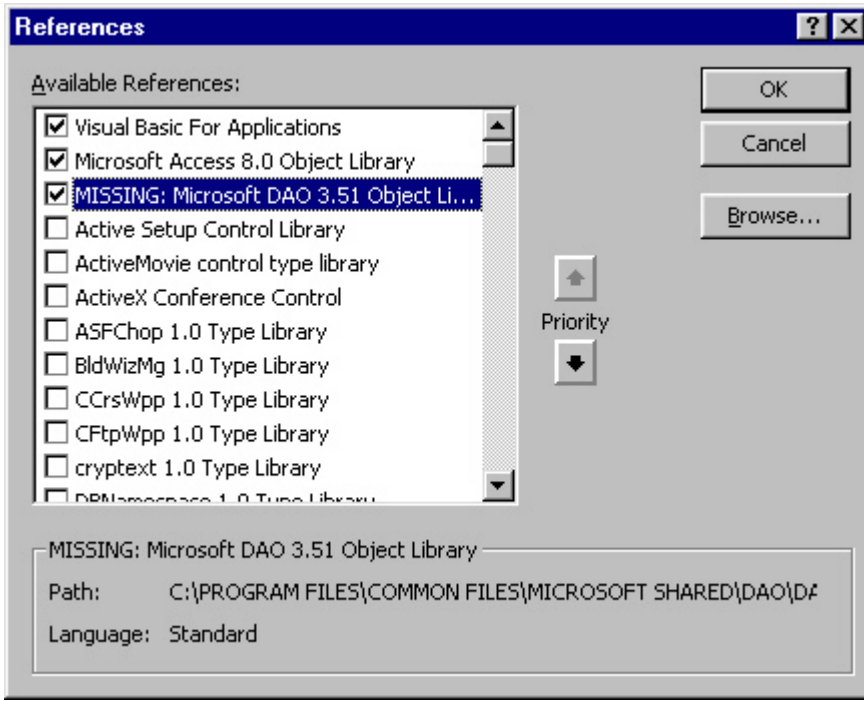
(Otomation Error/ Compile Error)

Bu çokça raslanan bir sorun. Otomasyon Hatası daha çok modüllerin derlenemediği durumlarda meydana çıkıyor. Bunun birçok sebebi olabilir. Bunları tam anlamıyla bilmesem de karşılaşp çözebildiklerimi anlatmayı yeğliyorum. Access'te kod yazmaya yeni başlamışsak bazı işlevlerin neden hata verdiğini anlamak için kendimizi epey yormamız gerekir. Mesela farklı dildeki (ya da konfigürasyondaki) iki Access'ten birinde Mid, InStr vs. komutlar çalışırken diğerinde hata oluşabiliyor. Sebebi çoğunlukla başvurulardaki bir eksikliktir.

References

(Başvurular)

Başvurular listesini herhangi bir modül açıkken "Tools/References" menüsünden görebiliyoruz. Burada "Missing:" (Kayıp) olarak belirtilmiş isimler arayın. Eğer varsa bunu kaldırıp yeniden yüklemeniz sorunu çözebilir. Altteki resimde başka bir bilgisayarda farklı klasör altında bulunan DAO350.DLL dosyası bulunamadığı için hata oluşmuş ve bu başvuru "MISSING:" olarak belirtilmiş. Daha sonra başvurulardan ayrıntılı olarak bahsedeceğiz.



DAO başvurusu bulunamadığı için dosya “MISSING:” (kayıp) olarak belirtilmiş.

2. Modül Penceresi Açılmıyor ve Kodlar Çalışmıyor

Bu problemin bazen doğrudan başvurularla ilgili olduğunu düşünebiliriz. Bazan da projemizdeki bir kod hatası buna sebebiyet verir. Herhangi bir modül sayfası açıkken “Debug/ Compile And Save All Modules” (Derle/ Tüm Modülleri Derle Ve Kaydet) menüsünü çalıştıralım. Bu işlem nerede hata olabileceğini birçok zaman gösterir. Eğer aynı adı taşıyan iki yordam varsa, ya da örneğin bir IF satırının bitirilmemiş olması veya “END SUB” kullanılmaması gibi hataları bu şekilde farkedebiliriz.

Bazı durumlarda derleme işlemi de sonuç vermeyebilir. O zaman bizim yapabildiğimiz son şey yeni bir dosya oluşturup hatalı olan dosyadan tüm modülleri, formları ve raporları kopyalamak. Bu esnada tüm modüller alınırken hatalı olanlarda ekrana ileti çıkıyor. Tabii ondan önce hatalı dosyamızı başka makinelerde test etmekte yarar var. Bazan başka bir yerde Compact (Veritabanı Düzenle) ve Repair (Veritabanı Onar) işlemleri sorunu giderebiliyor.

Bunun dışında bir de Türkçe karakter içeren nesne isimlerine dikkat etmek gerekli. Veritabanı içindeki nesne adlarında (Tablo, Sorgu, Form vs.) türkçe karakterler kullanmaktan olabildiğince kaçınmanızı öneririm.

3. Trim() ve Mid() İşlevleri Çalışmıyor

Bu hata çoğunlukla başvurularla ilgili. Trim işlevi VBA kütüphanesinin bir üyesi. Bunu Object Browser’dan (Nesne Yöneticisi) öğreniyoruz.

Bu dosyayı başvurular listesine tekrar yüklemek gerekebilir. Object Browser bileşeni ileride eklentiler yazarken bize epey yardımcı olacak. Gerekirse bunu daha sonra ayrıntılı olarak ele alabiliriz.

Object Browser

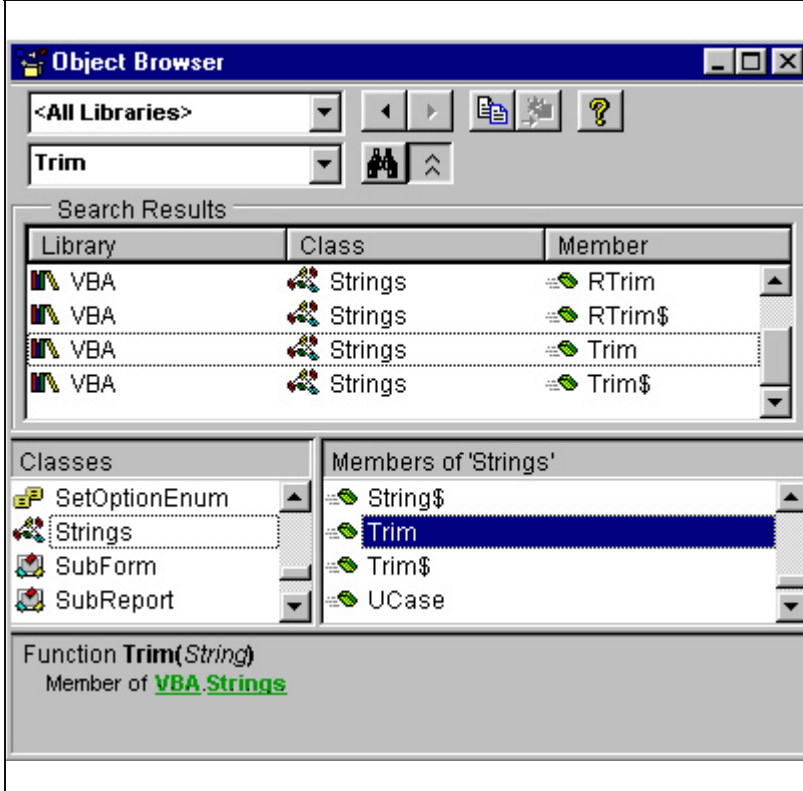
(Nesne Yöneticisi)

Trim komutunun bulunduğu kütüphaneyi öğrenmek için Object Browser’ı açıp Trim kelimesini aratıyoruz. Burada o an Access içinden kullanılacak tüm işlevlerin ve bileşenlerin listesini bulabiliriz. Aşağıda yer alan VBA yazısını tıklattınca VBA kütüphanesinin içeriğini görüyoruz. Bu kütüphanenin dosyası VBA332.DLL adlı dosya.

Library VBA

C:\Program Files\Common files\Microsoft Shared\Vba\Vba332.dll

Visual Basic For Applications



4. Program Penceresi Kapanmıyor

Bunun sebebi açılan kayıt kaynaklarının kapatılmaması ve kısmen de düzensiz bellek kullanımı olur çoğunlukla. Mesela,

```
Sub RecordSetDeneme()
Dim RS as recordset
Set RS=CurrentDb.OpenRecordset("Table1", dbOpenDynaSet)
...
RS.Close
End Sub
```

Bu yordam devredışı kalmadan önce mutlaka RS.Close komutu ile açık olan kayıt kaynağı serbest bırakılmalıdır. Eğer Database bileşeni tanımlanmış ise o da Nothing değerine eşitlenmeli.

```
Dim DB as Database
```

```
...
Set DB=Nothing
```

gibi. Aynı şekilde mantıksal değer ifade eden işlev parçacıklarının IF komutu içinde mutlaka TRUE ya da FALSE değerlerinden birine eşitlenmesi öneriliyor.

```
If DenemeFonksiyon() Then ... yerine
```

```
If DenemeFonksiyon()=True Then ... kullanılmalı.
```

Bu hatanın çözümüne The Access Web sitesinde rasladım. Her Accessçinin mutlaka bakması gereken bir site olduğunu söyleyeyim. (<http://mvps.org/access>)

5. "(AutoNumber)" / "(Otomatik Sayı)" Metnini Gizlemek

Veri türü otomatik değer olan bir veri alanını formdaki metin kutusuna eklediğimizde yeni kayıt anında "(AutoNumber)/(OtomatikSayı)" metni görünüyor. (Veri alanının adı "KayıtNo" olsun.) Bunu engellemek için önce yeni bir form açıp üzerinde bir metin kutusu nesnesi oluşturalım. Adı "txKayıtNo" olsun. Şimdi bunun içine veri alanının ismini

Burada dikkat etmemiz gereken şey, formda “KayıtNo” adlı başka herhangi bir nesnenin bulunmaması. Aksi halde kayıt verisi yerine aynı adlı nesnenin değeri görüntülenir.

6. Label/Metin Etiketi Nesnesinde Yeni Satır Kullanmak.

Label nesnelerinde başlığı yazarken yeni bir satıra başlamak için <Enter> yerine <Ctrl+Enter> tuşlarını kullanın. Aynı şekilde tablo görünümünde de bu tuş bileşimi yeni bir satıra başlamamızı sağlıyor. Bunu komut düğmeleri ve diğer nesneler için de deneyebilirsiniz.

Normalde Label ve CommandButton başlıklarında kısayol komutu için harflere altçizgi verirken "&" işaretini kullanıyoruz. Fakat bu işaretin kendisini nasıl göstereceğiz? Bunun için de “&&” yazmak yeterli.

7. Nesneyi Kendi Olay Yordamında Gizlemek.

Örneğin bir düğmenin "Click" yordamı içinden kendi görünür özelliğini kaldırmak istiyoruz (visible=false). Fakat odaklanılmış nesnelerde böyle bir işlemin geçersiz olduğuna dair hata mesajı çıkıyor. Bu satırdan önce başka bir nesneye odaklanmamız sorunu çözecektir. Yani;

Sub Button1_Click()

...

'Button1'in kendisini görünmez yapabilmek için Button2'ye odaklanalım.

Button2.SetFocus

Button1.Visible=false

...

End Sub

8. Formlara Eklenen Resimler Dosya Botunu Çok Büyütüyor

Form ya da raporlara özellikle büyük resimleri eklerken bunların BMP formatında olmasına dikkat edin. Ayrıca yüksek kalitede resim kullanmak, dosyanızın boyunun gereksiz yere artmasına neden olur. Daha önceden 300 KB'lık bir JPEG (16milyon renkli, 72 dpi) resmini dosyama eklemek istediğimde dosyanın boyutunun 2,5 MB arttığını gördüm. Aynı resmi aynı kalitedeki BMP dosyasına dönüştürüp aktardığımda dosyanın boyu sadece 350 kb. artmıştı.

9. SQL Kodlarında Tarih Kriterleri Sorun Çıkartıyor

Pek çoğumuz SQL kodlarında tarih arama kriterinin tanınmaması gibi bir sorunla karşılaşmışızdır.

“SELECT * FROM Table1 WHERE Tarih=#4.1.2000#”

gibi bir cümlede hata olduğu mesajı ile karşı karşıya kalırız. Sorun tarih ayıracının “.” yerine “/” olması gerektiğinden kaynaklanıyor olabilir. Ya da veriler yanlış geliyorsa gün ile ay kısmının yer değiştirmesi gerektiğini düşünebilirsiniz. Ve bunlar her bilgisayarda farklı olabiliyor. Bu sorunlardan kurtulmak için aşağıdaki şekilde deneyebilirsiniz.

Format(Date(), "#") ‘=36617 1.4.2000 için

Eğer tarihi belirtmek gerekiyorsa gün ile ayın hangisinin daha önce yazılacağına içimde hiç şüphe kalmadan şu komutu tercih ederim:

Format(DateSerial(2000, 4, 1), "#") ‘=36617

Artık tarih değerinin açıkça yazılması yerine doğrudan aynı değere karşılık gelen 36617 sayısı SQL cümlesi içinde yer alacak ve bizi sıkıntıdan kurtaracaktır.

“SELECT * FROM Table1 WHERE Tarih=36617”

gibi. Bu, hemen hemen tüm SQL uygulamaları için geçerli. 31.12.1899 günü 1 olacak şekilde her tarihin mutlaka bir sayısal değeri vardır. Veritabanları da tarih verilerini bu şekilde saklıyorlar, sadece gösterimde biçim uyguluyorlar. Tarihin saat kısmı ise tahmin edeceğimiz üzere ondalık olarak kaydediliyor.

10. Query/Sorgu Yazımında Hata

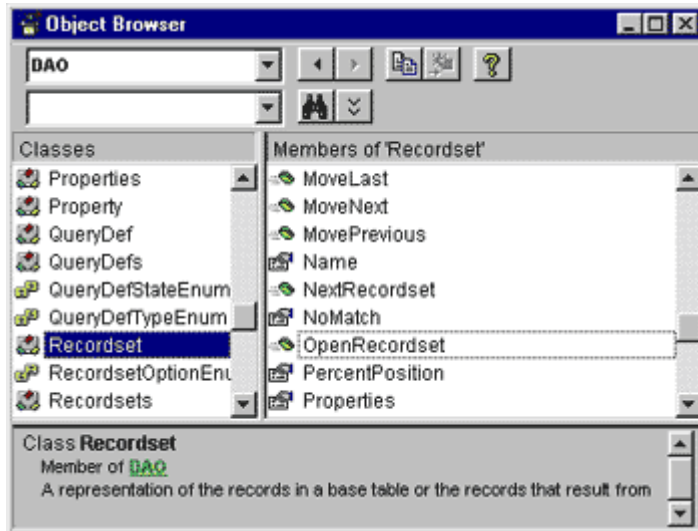
Access'in Türkçe sürümünde sorgu yazarken "AND" ve "IS NULL" yerine "BETWEEN", "REPEAT SECTION" gibi metinler otomatik olarak ekleniyor ve hatalı sorgu metinleri oluşuyor. Birçoğunuzun bildiği gibi bu hatalar servis güncellemeleri ile giderilebiliyor. En azından sorgu yazımında bu tür hataları önleyebiliyoruz. Sorgu yazarken Query Builder (Sorgu Sihirbazı) yerine SQL görünümünü kullanırsak bu tür kod değişiklikleri olmayacaktır. Fakat Query Builder ile yaptığımız ilk kayıta bu değişiklikler uygulanır.

Bu sayıda farklı birkaç konuyla ilgili bazı ipuçları vermeye çalıştık. Okuyuculardan gelen istek ve öneriler doğrultusunda da diğer bazı konulara değinmeye çalışacağız. İleriki aşamalarda Access'te başvuru kütüphaneleri, içine-ekleler ve sihirbazlar yazmayı deneyeceğiz ve Access 2000'e geçmeyi unutmayacağız...

Access'te Sınıf Modülleri

Class modülleri (Sınıf Modülleri) ayrı nesneler olarak kullanır, onlara istediğimiz gibi işlemler yaptırabilir ve onları çağırıp tekrar kullanabiliriz. Sıkça kullanacağımız bir işlemi sınıf modülüyle kolayca yapabiliriz. Bu konunun aslında doğrudan Visual Basic (VBA) ile ilgili olmasından dolayı biz biraz daha Access'e yaklaştırmaya çalışacağız. O yüzden okuyucunun sınıf modüllerini temel olarak bildiğini, en azından bu yazıdan bunun kısmen anlaşılabilirliğini varsayıyorum.

Sıkça kullandığımız "Recordset" tanımı Sınıf yani bir Class'tır. Nesne Gözetici (Object Browser) öyle söylüyor:



Class Recordset

Member of DAO

A representation of the records in a base table or the records that result from

Dim RS As Recordset

Set RS = CurrentDb.OpenRecordset("tablo1", dbOpenDynaset)

RS.AddNew

RS.Fields("isim") = "Hasan Baltacı"

RS.Update

RS.Close

Dim ile RS adlı değişkenin Recordset sınıfından olmasını sağladık. Addnew bu sınıfın metodlarından biri. Bu metod, RS'nin kayıt ekleme moduna geçmesini sağlıyor. Fields ise sınıfa ait bir özellik. Update ve Close ise diğer metodlardan.

LİSTE ÖZELLİKLİ BİR SINIF MODÜLÜ

Bizim sınıf modülümüz virgülle ayrılmış metinleri çözüp listeleyebiliyor ve sıralayabiliyor. Veritabanı penceresinde (Database Window) Modül (Module) sayfası açıkken Insert/ Class Module (Ekle/ Sınıf Modülü) menüsünü seçip ilk sınıf modülümüzü ekleyelim.

İlk bakışta herhangi bir modülden farkı yok.

Aşağıdaki kodu yeni bir sınıf modülü (Class Module) açıp yazın. Sonra bunu MyListBoxComma adı ile kaydedin. Unutmayın, modülü her zaman bu adla çağıracağız.

‘SINIF MODÜLÜ: MyListBoxComma

Option Compare Database

Option Explicit

Private Dizi() As String

Private Toplam As Integer

Private Ayirac As String * 1

Private Sub Class_Initialize() ‘Sınıf yükleniyor.

Toplam = 0

Ayirac = ","

End Sub

Public Sub AddItem(Strx As String) ‘Listeye eleman ekle.

Toplam = toplam + 1

ReDim Preserve dizi(1 To Toplam) As String

Dizi(Toplam) = Strx

End Sub

Public Property Get Seperator() As String ‘Ayıraç.

Seperator = Ayirac

End Property

Public Property Let Seperator(ByVal vNewValue As String)

Ayirac = vNewValue

End Property

Public Function Clear() ‘Listeyi temizle.

Erase Dizi

Toplam = 0

End Function

Public Property Get ListCount() As Integer ‘Listedeki eleman sayısı.

ListCount = toplam

End Property

Public Sub Sort() ‘Listeyi sırala.

...

...

End Sub

Private Function GetMin(sira As Integer) As Integer

‘Sıralamak için kullanılıyor. Sadece bu Class içinde

‘kullanılabilir. Çünkü Public olarak değil, Private olarak tanımlanmış.

...

...

End Function

Public Function SetComma(ByVal Strx As String)
'CommaText olarak girilen metni listeye dönüştürür.

...

...

End Function

Public Function GetComma() As String
'Tüm listeyi ayıraç kullanarak cümle haline getirir.

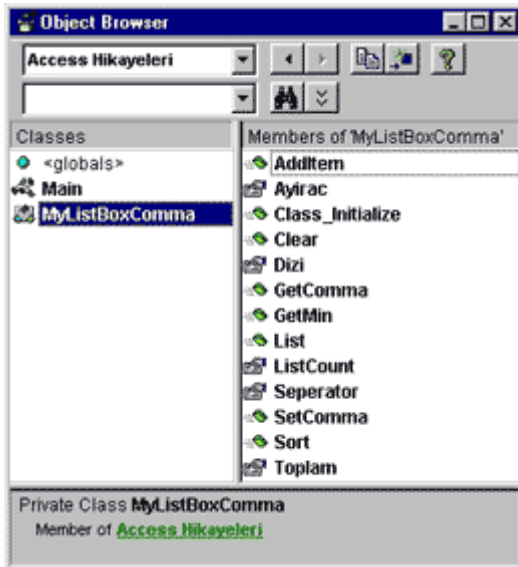
...

...

End Function

Bu modülden örneğin şu şekilde faydalanabiliriz:

Sınıfı MyListBoxComma adı ile kaydettiğimiz için bu adı kullanmalıyız.
'New ekini de kendi yazdığımız sınıfları kullanabilmek için yazmak zorundayız.
Dim Liste As New MyListBoxComma
Liste.AddItem "ocak"
Liste.AddItem "subat"
Liste.AddItem "mart" '... Tüm ayları yazın.
MsgBox liste.ListCount & " adet eleman:" & Chr(13) & liste.GetComma
Liste.Sort
Liste.Separator = Chr(13) 'Satirsonu işaretini(13) koyunca altalta listelenecek.
MsgBox "Listenin sirali hali:" & Chr(13) & liste.GetComma
Liste.Separator = "-"
Liste.SetComma ("sonbahar-kis-ilkbahar-yaz")
MsgBox Liste.ListCount & " adet oldu simdi."



Modülü biraz inceleyince Private ve Public şeklinde iki tanım görüyoruz. Bildiğiniz gibi Private ve Public tanımlamaları yordamların dışardan kullanılıp kullanılamayacağını belirliyorlar. Bilmeyenler için Public kullanmadan dışarıdan okuma ya da değiştirme yapılamayacağını ve Private kullanırsak sadece modülün kendi içinden okuma ve yazma yapabileceğimizi de söyleyeyim.

Burada normal işlev (function) ve yordamlardan (sub) farklı olarak Property Get ve Property Let tanımları da kullanılmış. Örneğin Separator (Ayıraç) özelliği için hem Get hem de Let yöntemi var. ListCount için ise sadece Property Get var. Bu durumda ListCount özelliği sadece okunabilir, Separator ise hem okunur hem de değiştirilebilir olmuştur.

Class_Initialize yordamı da anlayacağınız üzere bu modülün ilk çalışmaya başladığı yerdir. Fakat kullanılması zorunlu değildir. Aynı şekilde sınıflar bellekten atılırken varsa Class_Terminate adlı yordam çağrılır.

RAPOR YAZDIRIRKEN KOPYA SAYISINI SEÇMEK

“Rapor1” adlı raporu 3 defa yazdırmak için şu kodu deneyin:

```
DoCmd.OpenReport "Rapor1", acViewPreview  
DoCmd.PrintOut acPrintAll, , , 3
```

Örneğin 6-8 arası sayfaları 3 kez yazdırmak için ikinci satırı şu şekilde değiştirebilirsiniz:

```
DoCmd.PrintOut acPrintRange, 6, 8, , 3
```

PrintOut komutu aktif olan rapor, form ya da tabloyu yazdırır. Kullanmadan hemen önce yazdırmak istediğiniz belgenin aktif olmasını sağlayın.

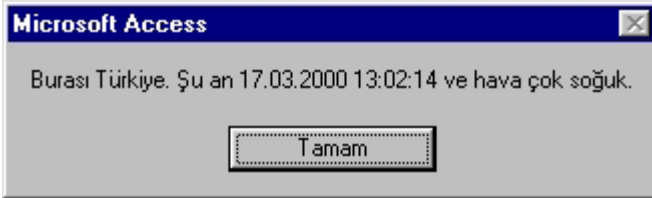
SINIRSIZ PARAMETRELİ İŞLEV (FONKSİYON) YAZMAK

Bazan öyle olur ki sınırsız parametre verebileceğimiz yordamlara ihtiyaç duyarız. İşte örneği:

```
Function FormatStr(ByVal MainStr As String, ParamArray StrList() As Variant) As String  
‘ParamArray en son parametrenin değişken uzunluklu bir dizi olmasını sağlıyor.  
If UBound(StrList) < 0 Then Exit Function  
Dim i As Integer  
For i = 0 To UBound(StrList)  
MainStr = Replace(MainStr, "%" & i + 1, StrList(i))  
Next  
FormatStr = MainStr  
End Function
```

Bunu şu şekilde kullanın:

```
MsgBox FormatStr("Burası %1. Şu an %2 ve hava %3.", "Türkiye", Now(), "çok güzel.")  
‘Resimdeki mesaja aldırmanın siz :)
```



ParamArray olarak tanımladığınız parametreyi en sona yerleştirin. Aksi halde hata oluşur.

DAHA GÜZEL KAYIT SİLME

Access’in kendi komut düğmesi sihirbazı ile gelen kayıt silme kodu ekrana uyarı iletisi çıkartıyor ve bu genelde iyi bir görüntü oluşturmuyor. Bunun yerine aşağıdaki kodu kullanabilirsiniz.

```
Public Sub KayıtSil(Optional ByVal frm As Form, Optional ShowMsg As Boolean)  
On Error Resume Next  
If ShowMsg Then  
If MsgBox("Kayıt silinecek. Devam edilsin mi?", vbCritical + vbYesNo) = vbNo Then Exit Sub  
End If  
If frm Is Nothing Then  
DoCmd.SetWarnings False 'Uyari mesajlarini gosterme.  
DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70  
DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70  
DoCmd.SetWarnings True 'Sonraki uyari mesajlarini gosterebilirsiniz.  
Exit Sub
```

```
Else  
If frm.NewRecord And (Not frm.Dirty) Then Exit Sub  
End If
```

'Yeni kayıt ya da kayıt degisikligi ise önce UNDO yap.

```
If frm.Dirty Then DoCmd.DoMenuItem acFormBar, acEditMenu, acUndo, , acMenuVer70
```

```
If Not frm.NewRecord Then 'Eski kayıt ise DELETE  
With frm.RecordsetClone  
.Bookmark = frm.Bookmark  
.Delete  
.MoveNext  
If .EOF Then .MoveLast  
frm.Bookmark = .Bookmark  
End With
```

```
End If  
End Sub
```

Kullanımı şöyle:

Call KayitSil (Me,True)

Eğer ikinci parametreyi vermezseniz ya da False olarak verirsiniz “Kayıt silinsin mi?” mesajı görünmez.

Call KayitSil (Me)

Aslında “Me”yi yani formun kendisini parametre olarak yollamanıza her zaman gerek yok.

Call KayitSil ()

ya da,

Call KayitSil (, True)

Bu şekilde isterseniz bir formdan ya da bir tablodan kendi mesajınızı göstererek kayıt silme işlemi yaptırabilirsiniz.

Gördüğünüz gibi bazı parametreleri vermemize her zaman gerek yok. Bunu yordamın başlığındaki “Optional” seçeneği ile sağlıyoruz. Optional belirttiğiniz ilk parametreden sonraki diğer parametrelerde de mutlaka aynısını kullanmanız gerekiyor.

Önceki sayıda SQL kodlarında tarih kullanımıyla ilgili verilen ipucunun Delphi’deki karşılığı soruluyor. VB’deki Format işlevi yerine FormatFloat bu işi görür:

```
Tarih:= #05.23.2000#;  
// ya da Tarih:=EncodeDate(2000,5,23); //23.05.2000  
ShowMessage(FormatFloat('#',Tarih)); //36669  
Query1.SQL.Text:='SELECT Table1.* FROM Table1 WHERE Table1.Tarih=36669';
```

Geçen ayki ipucuna bakmayanlar için Access ya da VB’de Format(Now,“#”) komutunun kullanılabileceğini ve SQL kodlarında da bu şekilde belirtme yapılabileceğini söyleyeyim.

Tüm örnekleri ve Access’te yapılmış ufak bir Sınıf modüllü oyunu Ailenizin Web Sitesi’nde (<http://www.pcmagazine.com.tr>) Dergide Adı Geçen Dosyalar bölümünde bulabilirsiniz. Gelecek sefer ilk başvuru kütüphanemizi (Reference Library) yazacağız.

ACCESS HİKAYELERİ

HİKAYE BU YA...

Bazan Access'in son üç sürümünü aynı anda kullanmak gerekiyor. Office'in en sevdiğim özelliklerinden biri yeni kurulumlarda eski sürümlere acıyıp onlara kıymaması. Yapmamız gereken tek şey yeni kurulacak klasörü eskisinden farklı belirlemek.

Daha önce Access2'de yapılmış programlarımızı Access97'ye dönüştürmemiz gerekiyor. Ağdaki makinaların bir kısmında MS'nin binbir programlı özel paketinden çıkan Office97 Türkçe sürümü yüklü. Bir kısmı ise mecburen aldığımız İngilizce sürüm.

Access'le çalışan tüm tanıdıklarım malum Türkçe meselesinden yakınıyor. Bizimkilerin sorunu da nedir bir bilseniz! Form adlarında yumuşak g ve diğer Türkçe karakterler var diye hiçbir kod çalışmıyor. Neyse ki farklı dildeki bir Windows'ta bunu düzeltebildik.

Şimdi gelgelerim yüzlerce sorgu (query), form, kod içinde bu formun adı kullanıldığı için arkadaşlar başlarının kısmen belada olduğunu düşünmeye başladılar. Bazan bu işler imkansız gibi görünse de Access'teki gizli yardımcılarımız yine yardımımıza koşuyor.

Tools/Analyze/Documenter menüsünü çalıştırdık. Documenter, dosyamızın içinde ne var ne yok bir bir gösteriyor. Bilmem hangi formun hangi düğmesinin başlığı nedir, hangi sorgunun sql kodu nasıldır gibisinden herşey var. Böylelikle Access dosyamızın içerisinde Türkçe karakter olan tüm değerleri kolay yoldan bulacağız.

İşin sırrı nerede? File menüsündeki "Save As Table" komutunu çalıştırmak gerekiyor ki o kadar zamanımızı alan bu raporu heba etmeyelim. Gerisi "Object Definitions" adlı tabloyu incelememize kalmış.

Baştaki konuya dönersek MDB dosyalarının çift tıklatılmasıyla hangi sürümün açılması gerektiği önceden bilinmeli. Bu durumda dosyaların açılma seçeneklerine bunları yerleştirmek lazım. Bir de bizim kendi programlarımız için verdiğimiz uzantının da aynı şekilde ayarlanması gerekiyor.

DAX uzantılı dosyamız bir şekilde HKEY_CLASSES_ROOT\DAX_auto_file dizinine yerleşmiş. Windows Gezini'nde sağ tuşla tıklatınca Aç komutu doğrudan Access2'yi açıyor. Şimdi biz buraya Access97 ve Access2000'i ekliyoruz. HKEY_CLASSES_ROOT\DAX_auto_file\shell\open\command anahtarının değeri Access2'yi gösteriyor. Yeni değerler oluşturduk:

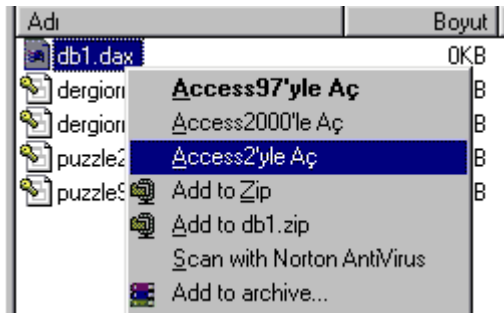
HKEY_CLASSES_ROOT\DAX_auto_file\shell\Access97 İle Aç\command

HKEY_CLASSES_ROOT\DAX_auto_file\shell\Access2000'le Aç\command

Her ikisinin Default değerine msaccess.exe dosyalarının yolunu yazdık.

Tabii bunun çalışması için HKEY_CLASSES_ROOT\dax anahtarında "DAX_Auto_File" yazıyor olmalı.

Eğer dosya Auto_File olarak tanımlı değilse daha kolay bir yolla aynı şeyleri yapmamıza izin veriyor Windows. Bildiğiniz gibi Windows Gezini'nde Görünüm/Seçenekler/Dosya Türleri menüsünden yapılabilir bu.



Sonuçta resimdeki gibi bir sağ fare tuşu menüsü hiç fena değil bence.

DOSYA:

[AccessHaziran2000.zip](#)

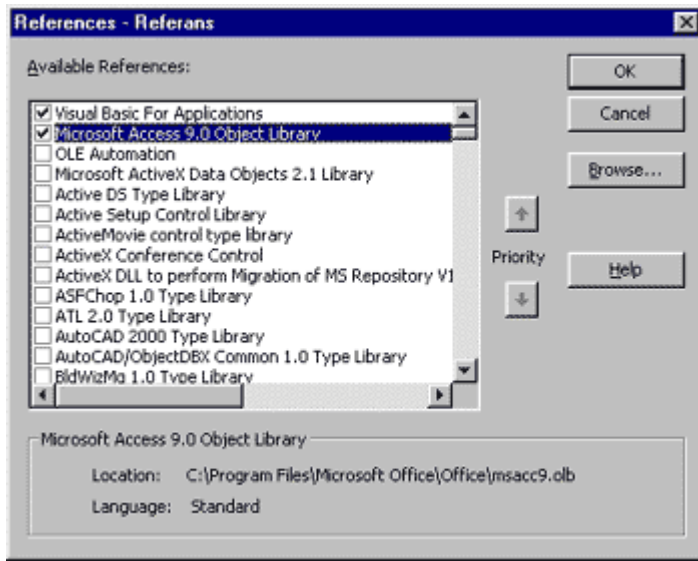
Access'te Başvuru Kitaplığı Yazmak

Bu ay biraz derinlere inip Access'teki References (Başvurular) listesine kendi başvuru kitaplığımızı ekleyeceğiz.

REFERENCE LIBRARY

Diğer Office uygulamalarında da olduğu gibi Access'te hem standart nesne kütüphaneleri hem de Activex kütüphaneleri kullanılabilir. Bu yazıyı yazdığım Word2000'de Visual Basic Editor'ü açtığımda Tools/ References menüsünden tüm başvuruları görebiliyorum. Aynıısı Access için geçerli olup Office'te programlama anında tüm bu kütüphanelerden istediğimiz gibi faydalanabiliyoruz.

Access'te boş bir dosya açıp modül sayfasında yeni bir modül oluşturalım. Tools/ References (Araçlar/ Başvurular) menüsü ile başvuruların listelendiği pencereyi görelim.



MSACC9.OLB dosyasının Microsoft Access 9.0 Object Library kütüphanesi olduğunu öğreniyoruz. Visual Basic For Applications ile beraber bu ikisi tüm Access uygulamalarında kullanılan dosyalar. Access bunları kaldırmamıza izin vermiyor.

Örneğin Access projenizde müzik çalmak istiyorsunuz ve elinizde bir Wav dosyası var. Bunun için eğer sisteminizde Microsoft Media Player yüklü ise büyük ihtimalle başvurular listesinde buna ait kütüphane görünecektir. O zaman çok kısa bir kodla müzik çalmayı başarabiliriz. Başvurular listesinden Windows Media Player'i seçip alttaki dört satırı çalıştırın.

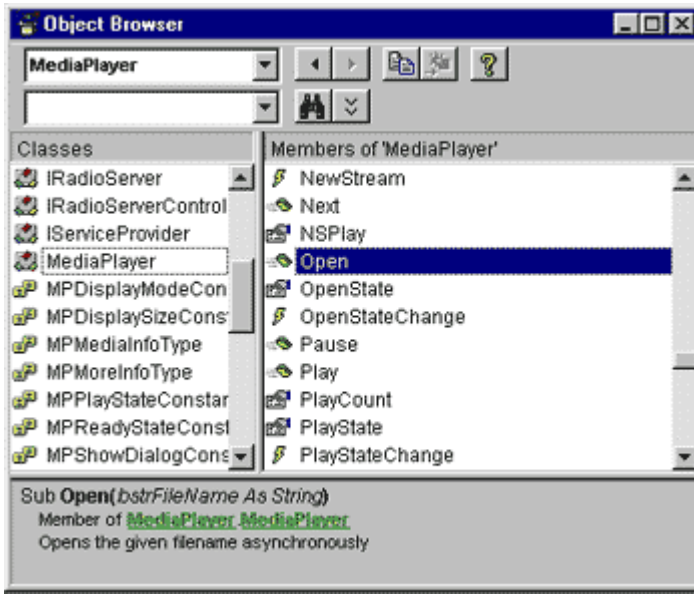
On Error Resume Next

Dim muzik As New MediaPlayer.MediaPlayer

muzik.Open "c:\windows\media\ding.wav"

muzik.Play

Gördüğünüz gibi çok kolay. Ama listede onlarca kütüphanenin hangisinin nasıl kullanıldığını nereden öğreneceğiz? Üstteki müzik çalma örneğini vermeden önce başvurular listesinden hangisini örnek vereyim diye aradım. Hiçbirinin nasıl kullanıldığını zaten bilmiyorum idiysem bana sadık dostumuz Object Browser (Nesne Gözetici) yardımcı olmuş demektir. F2'yle onu açtıktan sonra sol üstten Media Player'i seçerek içindekileri biraz incelemek gerekiyor.



İlk olarak müziği çalacak nesneyi bulmak lazım. Bunu sol alt listeden alıyoruz. İlk bakışta siz de aradığımız şeyin “Media Player” adında olduğunu anlayacaksınız. Bunu seçince sağdaki listeden de Open, Play gibi şeyler aramamız gerektiği de hemen aklımıza gelenlerden. Open’i seçtiğimizde alttaki kısımda nasıl kullanılacağı gösteriliyor. Bir dosya adını parametre olarak vermek gerekiyormuş. Sonra da Play’a bakacağız ve üstteki birkaç satırlık kodu çok kısa bir sürede keşfederek oluşturacağız. Eğer Media Player’iniz MP3 çalışıyorsa sizin de bu şekilde MP3 çalmanıza hiçbir engel yok. Hatta Access’te kendi müzik listenizi bile yapabiliyorsunuz bununla. Tabii VB’ciler tüm bunların VB’de de yapılabileceğini zaten biliyorlardır.

KENDİ BAŞVURU KÜTÜPHANENİ KENDİN YAP!

Geldik kendi başvuru dosyamızı yapmaya. Yeni bir Access dosyası açıp içine birer adet tablo, form ve rapor yerleştirelim. Örneğimizde üçü de “Kisiler” adında tablo, form ve rapor var. Sonra modül sayfasında yeni bir modül oluşturup aynen şunları yazıyorum:

Option Compare Database

Option Explicit

Public Function Topla(sayı1 As Long, sayi2 As Long) As Long

Topla = sayi1 + sayi2

End Function

Public Sub KisilerFormunuAc()

DoCmd.OpenForm "Kisiler"

End Sub

Public Sub KisilerRaporunuAc()

DoCmd.OpenReport "Kisiler"

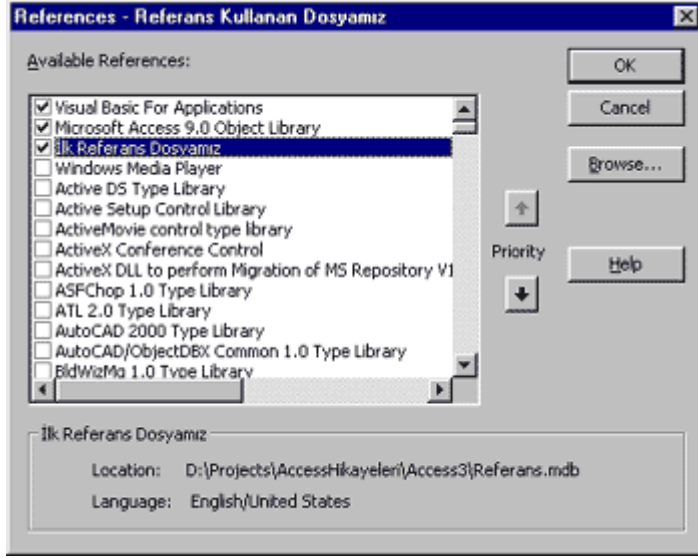
End Sub

Bir başvuru dosyasının içindeki bir formu, raporu veya diğer şeyleri açabilmek için aynı dosya içinde bu işleri yapan kodlarımızın olması gerekiyor. Örnekte rapor ve formu açabilmek için iki ayrı kod var. “Topla” adlı işlev ise genel bir yardımcı işlev görevi üstlenecek. Yani biz bu başvuru dosyasını nereye eklersek ekleyelim bu işlevi Access’in kendi işleviymiş gibi kullanıp, form ve raporu da açabileceğiz. Bu dosyamızın adı Referans.mdb olsun.

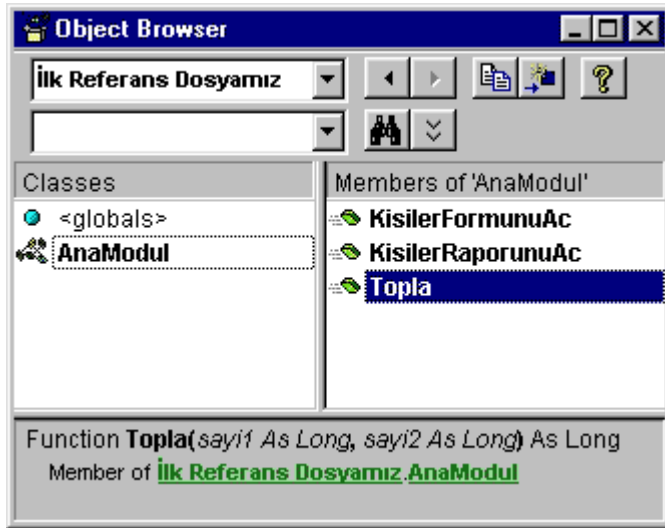
Zorunlu olmasa da bir ayar yapmamızda fayda var. Başvurular kısmında bu dosyanın görünen ismini belirlememiz gerekiyor. Yani projemizin adını. (Visual Basic Editor’de Tools/Properties/ProjectName.)

Application.SetOption "Project Name", "İlk Referans Dosyamız"

Bunu yaparak başvurular kısmında dosyamızın adının “İlk Referans Dosyamız” şeklinde görünmesini sağladık.



Şimdi bunu kullanmak için kapayıp yeni bir Access dosyası açıyoruz. Yeni bir modül sayfası açarak Başvurular bölümünde bu dosyayı bulup seçelim. Sonra Object Browser ile ilk başvuru kitaplığımızı inceleyelim.



İlk başvuru dosyamızdaki kullanılabilir işlevleri Object Browser ile görebiliyoruz.

Şimdi alttaki satırları çalıştırmayı deneyelim.

KisilerRaporunuAc
KisilerFormunuAc
MsgBox toplu(3, 5)

Böylelikle hem başka bir Access dosyası içindeki form ve raporu açabiliyoruz, hem de kendi ortak işlevimizi yazmış oluyoruz.

Tüm bunlardan sonra yapılması önerilen son bir işlem daha var. Başvuru dosyamızı derleyip kodlarını gizlemek ve kullanıma tam hazır hale getirmek. Şimdi Referans.MDB dosyasını tekrar açıp MDE dosyası oluşturalım. Tools/ Database Utilities/ Make MDE File (Araçlar/ Veritabanı Hizmet Programları/ MDE Dosyası Yap) seçeneği ile MDE dosyamızı oluşturduk. Bu şekilde hem kimse kendi özel kodlarımızı ve tasarımlarımızı göremeyecek ve değiştiremeyecek hem de bu dosyamız normal bir EXE gibi hızlı çalışacak.

Dosyamız hazır. Kendisini belirtmek için uzantısını örneğin REF olarak değiştirirelim. Şimdi asıl programımızdaki başvuru işaretini kaldırıp Referans.REF dosyasını yeniden yüklememiz gerekecek.

Bu ilk kütüphanemizi Access içinden bir DLL ya da OCX gibi kullanabiliriz artık.

Başvurularla ilgili genel olarak anlatacaklarım bu kadar. Sonrasında bununla ilgili birkaç ipucu vermekle yetineceğim. Bu özelliği kullanmaya karar vereceğiniz zaman CodeDB ve CurrentDB adlı iki nesneyi hatırlamanız faydalı olabilir. Bir başvuru dosyası içinde CurrentDB kullanırsanız bu, başvuruyu kullanan dosyayı işaret eder. Ama CodeDB kullanıyor iseniz bu da kodun bulunduğu dosyayı yani başvuru dosyasının kendisini işaret etmiş olur. Sıkça kullandığımız CurrentDB yerine CodeDB kullanmamızın sakıncası yoktur aslında. Halihazırda CodeDB zaten kodun yazılmış olduğu uygulamanın kendisini ifade ediyor.

İPUÇLARINA DEVAM - Bir Formu Veya Raporu Birden Çok Kere Açmak.

Access'te de VB ve diğer dillerde olduğu gibi bir formu ya da raporu aynı anda istediğimiz kadar açma şansımız var. İşte örnek:

Global Frm1 As Form_Form1

Global Frm2 As Form_Form1

Sub AyniFormu3kereAc()

DoCmd.OpenForm "Form1"

Forms("Form1").Caption = "Bu, Form1'in orijinali"

Set Frm1 = New Form_Form1

Set Frm2 = New Form_Form1

Frm1.Caption = "Bu, Form1'in ilk kopyası"

Frm2.Caption = "Bu da Form1'in ikinci kopyası"

Frm1.Visible = True

Frm2.Visible = True

MsgBox "FORM1'in 2 adet kopyası açıldı."

End Sub

Burada kullandığımız Form_Form1 tanımlaması Form1 formuna ait sınıfın adıdır. Geçen sayıda bahsettiğimiz sınıf modülleri konusunda bundan bahsetmek için acele etmedik. Normalde form ve raporlara ait kodların bulunduğu modüller aslında sınıftır ve bir sınıf ile yaptığımız herşeyi form ve raporlara ait modüllerle de yapabiliriz. Burada şunu unutmamakta yarar var. Eğer FRM1 ve FRM2'yi aynı yordam (SUB) içinde tanımlasaydık kod bitip bilgiler bellekten atılırken bu formlar kapanacaktı. Bunu önlemek için onları "Global" deyimi ile tanımladık.

Şimdi yeni bir form veya rapor oluşturup onun üzerinde deneme yapmaya çalışırken neden hata oluştuğunu sorabilirsiniz. Böyle bir durumda form veya raporunuza ait bir modülün varolmadığından şüphelenin. O zaman bunun kod kısmına bir harf de olsa birşeyler yazıp sınıf modülün oluşmasını sağlamalısınız. Çünkü kod olmadan sınıf modülü olmaz.

Gelecek sefer gelişmiş sorgu (query – sql) örneklerini incelemek istiyorum. Sadece Access'çilerin değil, SQL kullanan ve de kullanmak isteyenlerin bunu kaçırmamasını öneririm. Sonrasında Add-In yazma konusuna başlayabilmek için kod ile Access formlarının ve raporlarının nasıl düzenlendiğinin örneklerini inceleriz. Ama ondan önce ODE Tools'tan bahsetmeyi planlıyorum.

Güneşli güzel günler...

ACCESS HİKAYELERİ

ODE TOOLS GÜNLERİM

Sevgili Günlük. Access97 için ODE Tools'u satın almış bulunuyoruz. Yanlışlıkla getirdikleri yükseltme sürümünden sonra temiz bir sürüm elde ettik. Office97 Developer Edition adlı bu kutu içinde hem Office97 Professional var, hem de Office Developer Tools (ODE Tools). En önemlilerinden biri de Office programlama için iki güzel kitap. Access'te ya da Office'te programlama yapmaya çalışırken neyi yapıp neyi yapamayacağımızı bu kitaplar gayet iyi anlatıyor.

O güne kadar MDB dosyalarını EXE'ye dönüştürdüğünü zannettiğim Access Setup Wizard'ının arayüzü de Access'te yapılmış. EXE'ye dönüştürme de zaten MDE dosyaları kısmen EXE'ler gibi çalıştığı için sorun yok. Artı ve eksiler şöyle:

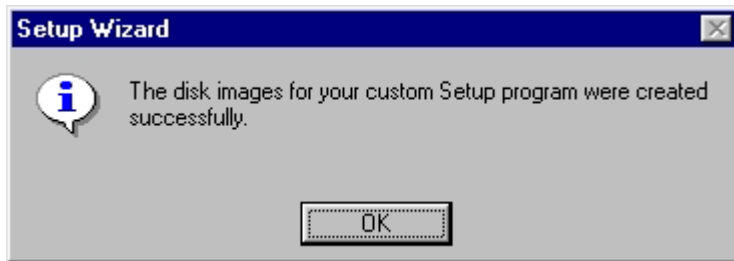
- Setup Wizard ile güzel bir kurulum programı hazırlıyoruz ve programımızı Access'in kendisi olmadan müşterimize kurabiliyoruz. Paket gerekli dosyaları bizim için topluyor. İçine bizim programımızla birlikte Access Run-Time modülünü koyuyor.
- Bir bilgisayarda sadece bir adet Run-Time modülü olması yeterli. Sonrasında istediğimiz kadar MDB, MDE dosyasını kopyalarak bunları çalıştırabiliyoruz.
- Haliyle müşterimize Access kurmaya gerek kalmıyor.
- Türkçe Access'te hazırlanan bir kurulumu İngilizce Access olan makineye kurduğumuzda maalesef onun dilini Türkçe'ye çeviriyor. Tersi de geçerli. Bir bakıyoruz form tasarımlarındaki Properties penceresinde Özellikler yazmaya başlamış. Çok acıklı değil ama neyse...
- Kurulum hazırlama yeteri kadar ayrıntılı. VB'dekinde olduğu gibi birçok seçenek burada da var.
- Ayrıca kurulumda ODBC ve diğer veritabanı seçenekleri var. İstenirse grafik (Chart), Outlook ve Exchange modülleri de dahil edilebiliyor.
- OCX kullanmış isek onu listeye kendimiz ekliyoruz ve Wizard, o OCX için gerekli diğer dosyaları kendisi buluyor.

Gelgelelim bazan kafa karıştırıcı şeyler de oluyor. Bir tanesini denerken kurulum sırasında bazı hatalar oluştu. Neden olduğunu zar zor anlayabildik. Meğer ki ayarları yaparken klasör ismi sorulan yerlerde klasörün tam adı yazılacak, ama isminin sonunda bölü işareti olmayacaktı. O kadar çok makine ve sistemde test ettim ki rasladığım problemlerin altından kalkmam epey vaktimi aldı. Sağolsun Microsoft'un destek sitesi böyle durumlarda ilaç gibi geliyor.

Epeydir, çözemediğim bir hata olduğunda en iyi yardımcılardan olan Altavista'ya o hata mesajının aynısını soruyorum. Akıllı AV şimdiye dek en az on adet büyük problemimin yarısından fazlasında, verdiği ilk cevapta çözüme ulaştırmıştı beni. Bunlar da MS'nin destek sitesi oluyor çoğunlukla. Sonradan Ask.com'u keşfettim. İlk sorduğum şey "where are you?" idi. Cevabı görünce gözlerim yaşarmıştı. ".. at AskJeeves!". Tabii bir de azar işittim. Soracak doğru dürüst birşeyim yok muymuş!

Kurulum paketi hazırlamaya devam ederken bir yandan da deneylere devam ediyorum. Zira paketi hazırlamak azıcık uzun sürüyor. Acaba "D" sürücüsüne kurulu yepyeni bir Windows'ta bu kurulumu çalıştırdığımda herşey yerli yerine oturacak mı? Cevap evet oldu. Son olarak merak ettiğim iki şey vardı. Kendi başvurularımızı eklediğimiz dosyalar sabit klasör adını arayacakları için başvuru problemleri ile karşılaşabilir miyiz? Bunun da cevabı evetti, maalesef ki. (Tabii o zaman bunun başka türlü çözebileceğini bilmiyordum.) Bir de Access 2000 yüklü makinelerde Access 97 için hazırlanmış kurulum sorun çıkarmadığı için testler dizisini sadece bir fireyle bitirmiş oldum.

Sonuç olarak Setup Wizard bize, dosyaları onarma (repair/compact) komutları dahil Programlar menüsündeki kısayolları da ayarlayan ve gerekirse Registry'ye kurulumla ilgili kayıtlar giren güzel bir kurulum paketi yapıyor.



EXE yapmasak da programlarımızı Access'e gerek kalmadan kurabileceğiz şimdi.

Hoşçakal Günlük.

DOSYA:

[AccessRef.zip](#)

HERKES İÇİN SQL

İyi sorgular yapabilmek için iyi hazırlanmış tablolarımızın olması gerekiyor. Dizinler, ilişkiler, geçerli değerler (indexes,

relationships, default value) gibi yardımcı özelliklerle veritabanımızın sağlamlığını artırabiliriz.

PERFORMANS, AMA NASIL?

Daha önce karşılaştığım bazı ayrıntıları anlatmakla yetineceğim şimdilik. Sorgu yaparken Access'i yoran şeylerden bir tanesi tablolarda tanımlı DevaultValue (Varsayılan) değeridir. Ekranda bir sürü açılan liste kutusu (combobox) varsa ve herbiri kendi listesini başka bir tablodan araştırarak getiriyorsa formumuzun pek hızlı olmasını beklememeliyiz her zaman.

Örneğin veritabanında, beraber çalıştığımız müşteri şirketler ve bunlara ait personel listesi olsun. Kisiler tablosundaki SirketNo adlı alan Sirketler tablosundaki No alanı ile bağlantılı. Yani biz kişilere ait şirket eşleşmesini şirketin ismi yerine numarasını vererek sağlıyoruz. Liste kutularındaki seçeneklerle gerekli eşleşmeleri ayarlayıp kolayca listeleme yapıyoruz.

Öncelikle bu tür eşleştirmelerde mutlaka dizin (indeks) ve ilişki (relationship) tanımlanmalıdır performans için. Aynı şekilde arama yapılacak alanlarda da dizin oluşturulmalıdır. (Sirketler tablosundaki No alanı.)

Asıl söylemek istediğime gelirim, özellikle şöyle bir ipucu işimize yarayabilir: Eğer varsayılan değeri (default value) sıfır olarak seçilmiş bir alana ait açılan liste kutusunda başka bir tablodaki kayıtları listeliyorsak ve bağlı arama alanı değeri hiçbir zaman sıfır olmayacaksa (mesela autonumber –otomatik sayı alanları için) boş yere sıfır değeri için arama yaptırıp zaman kaybedeceğiz. Zira açılır liste kutusu kendisinde kayıtlı 0 değerini bağlı tablodan arayacak ama hiçbir zaman otomatik sayı türündeki bir alanda bunu bulamayacaktır. Sıfır olmayacak alanlara bağlı sayısal alanlardaki, geçerli olan sıfır değerini kaldırarak formlarımızdaki kayıt geçişlerini bir nebze de olsa hızlandırabiliriz.

Bu, bir süre önce rasladığım küçük ama çok önemli olduğunu düşündüğüm bir ayrıntıydı. Bu ay aslında sorgulardan ve SQL'den bahsetmek istiyorum.

MAZERETİM VAR, SQL BİLMİYORUM

Access ile SQL bilmemize gerek kalmadan hemen her türlü sorgulama işlemini yapabiliriz. Ama yine de Access kullanmadan SQL kodu yazabilmenin inceliklerini birazcık da olsa öğrenmek gerekiyor.

Geçen haftalarda şirketin diğer departmanlarındaki arkadaşlarla bir saat kadar SQL çalıştık. Hepsi de SELECT, FROM, WHERE, ORDER BY başlıkları ile beraber MAX, MIN, SUM, COUNT, TOP, ASC, DESC komutlarını da anlayarak toplantıdan ayrıldılar. Toplantının sebebi uzak erişimli sunucularımızdaki verilere kendi başlarına sadece SQL kodu yazarak ulaşabilmelerini sağlamaktı. Her grup kendi yetki alanı dahilindeki veritabanlarında istediği SQL kodunu çalıştırarak istediği kayıtları listeleyebilmeliydi. Burada da bu ay sadece SELECT komutu dahilinde, yani kayıt listelemek için çalışalım.

SQL'de onu kullanan sistemin yapısına göre yazımda bazı farklılıklar gösterse de yaygın veritabanı sistemlerinin hemen hepsinde kayıt arama ile birlikte, yeni kayıt ekleme, silme, değiştirme ya da veritabanını yönetme gibi gelişmiş özellikler vardır. Mesela Access dosyasında istediğiniz bir tablonun alanlarını ya da dizinlerini SQL kodu ile değiştirebilir veya yeni bir tablo oluşturup bunun içine istediğiniz kayıtları SQL kullanarak ekleyebilirsiniz.

GENELDE SQL

Access bize sorgularda, formlarda, raporlarda, makrolarda ve modüllerde SQL kullanabileceğimizi söylüyor. Kayıt bulup getiren tipik bir SQL cümlesi şu şekildedir örneğin:

SELECT Ad, Soyadi, Yas, Tarih FROM Kisiler WHERE Ad="Ahmet" ORDER BY Tarih;

Bu komut (FROM) Kisiler adlı tablodan, (WHERE) Adı "Ahmet" olan kayıtları, (ORDER BY) Tarih alanına göre artan sıralayacak ve (SELECT) Ad, Soyadi, Yas, Tarih alanlarını seçecek şekilde getirecektir.

ORDER BY sıralama yapılacak alanı belirtir. Genelde en sonda kullanılıyor. "ORDER BY Tarih" demekle sonucu Tarih alanına göre artan sıralıyoruz. "ORDER BY Tarih ASC" ile arasında fark yoktur. ASC (Ascending) artan sıralamayı belirtiyor. Yerine DESC (Descending) kullansaydık ters sıralama yapılacaktı. Sadece bununla bitmiyor. Birkaç alana göre

de sıralama yapılabilir bununla. Mesela “.. ORDER BY Tarih DESC, Soyad ASC, Ad ASC” gibi. Yani Tarih’e göre ters sıraladıktan sonra önce Soyad’a ve ardından Ad’a göre düz sırala.

Aradığımız şeyi bulmak için WHERE komutunu kullanmamız gerekiyor.

.. WHERE ((Ad=’Ahmet’) AND (Soyad=’Kara’)) ..

şeklindeki bir kriter Ahmet Kara adlı kişinin kaydını arayacaktır. “WHERE Yas>=20” ile 20 yaş ve üzeri kişilerin kayıtlarını bulabiliriz. Ya da “Yas<>20” ile 20 yaşında olmayanlar bulunabilir. Eğer 20-40 yaş arası istiyorsanız “WHERE Yas BETWEEN 20 AND 40” demeniz yeterli.

İsterseniz içinde geçebilecek karakterlere göre de arama yaptırabilirsiniz.

.. WHERE (Ad Like “Ahm*”)..

“Ahm*”: Ahm ile başlayanlar,

“*Ahm”: Ahm ile bitenler,

“*ahm*”: Herhangi bir yerinde AHM olanlar (başında ve sonunda olanlar da dahil),

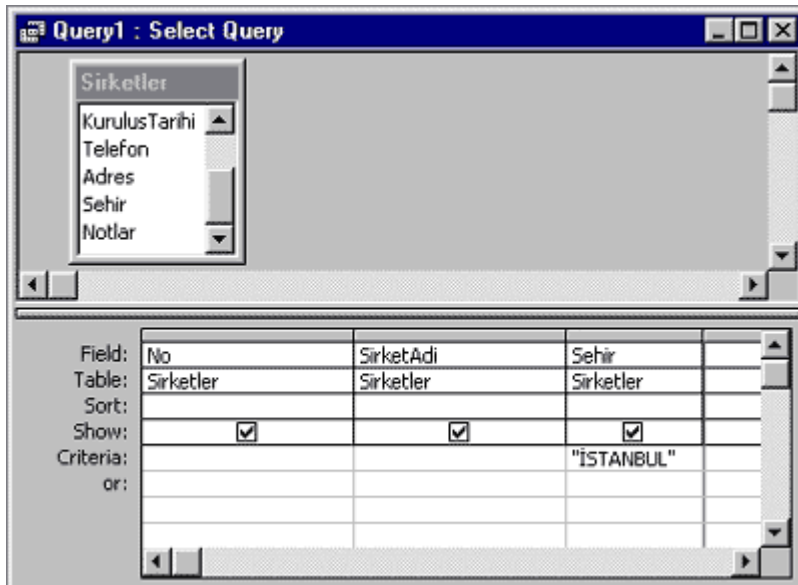
“*a?m*”: İçinde A olacak, sonraki bir adet harfin ne olduğu önemli değil ama onun yanında M harfi olmalı.

Bazı durumlarda çift tırnak yerine tek tırnak yazmanız gerekebileceğini hatırlamanız faydalı olacaktır. Bu örnekleri SQL cümlesi içindeki WHERE’in bulunduğu kısma yazarsanız çalışır. Tek başına bu kriterler çalışmaz.

FROM ile bildiğiniz gibi hangi tablodan kayıt alınacağı belirtiliyor. Şimdilik sadece tek tablo ile çalışıyoruz. SELECT ile bu tablo içindeki istediğimiz alanları görüntülüyoruz. “SELECT Ad, Soyad FROM ..” gibi bir cümle sadece Ad ve Soyad alanlarının getirileceğini belirtiyor. Bütün alanların gelmesini istiyorsanız “SELECT * FROM Kisiler ..” yazın, olsun bitsin.

Arama kriteri yoksa WHERE kullanmaya, sıralama yapılmayacaksa da ORDER BY kullanmaya gerek yoktur. Yani tek başına çalışabilecek en küçük SQL cümlesi “SELECT * FROM Kisiler” şeklinde olacaktır. Bu, Kisiler tablosundaki tüm kayıtları ve tüm alanları getirir.

ACCESS’TE SQL



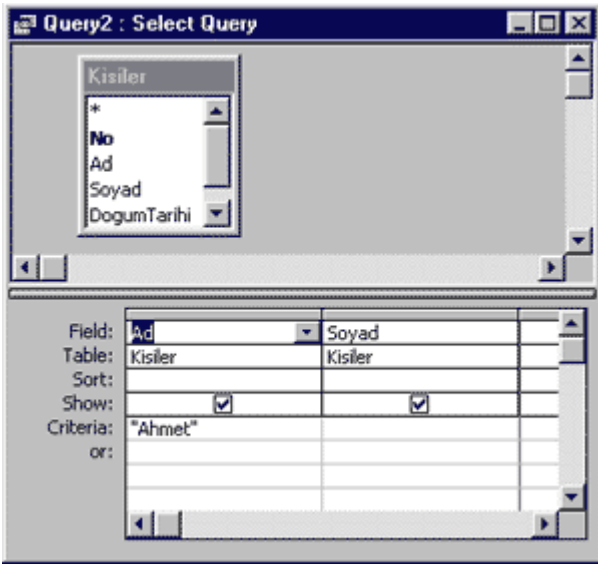
Veritabanı penceresinde Sorgu sayfasında yeni bir Sorgu oluşturalım. Sorguyu açtıktan sonra görüntülenene tablo seçme formundan sorgulamak istediğimiz tabloyu seçelim. (Yeni bir dosyada çalışıyorsanız, tabii ki önce bir tablo oluşturmalsınız.) Karşımıza, adının sonundaki sıfatı fazlasıyla hakeden Sorgu Sihirbazı çıktı.

Ben Kisiler tablosunu seçtim. Şimdi yukarıda öğrendiğimiz SQL kodlarını ilk olarak burada deneyebiliriz. Görünüm menüsünden SQL Görünümü'nü seçin.

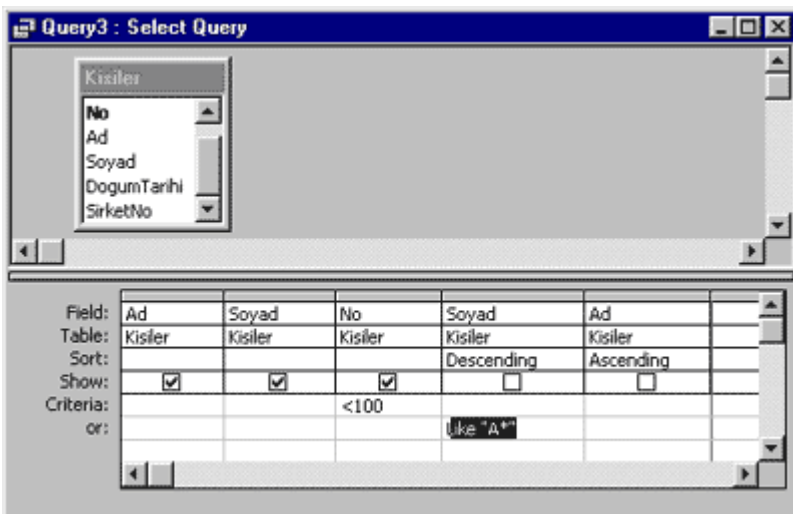
Bende “SELECT FROM Kisiler;” görünüyor. Bunu, öğrendiğimiz kurallara uyarak;

**SELECT Ad, Soyad
FROM Kisiler
WHERE Ad="Ahmet"**

olarak değiştiriyorum. Görünüm menüsünden Tasarım'a geçtiğimizde bakın nasıl bir görüntü çıkıyor (Query2):



SELECT kısmında * işareti de kullanabilirdik. Şöyle bir ayrıntıyı hatırlatmalıyım burada ki, bazı durumlarda tasarım görünümünde bütün alanlar olmamasına rağmen sorguyu çalıştırınca bütün alanlar listelenebiliyor. Bu, sorguların bütün alanları gösterme özelliğinin aktif olmasından dolayıdır. Görünüm menüsündeki Özellikler komutundan çıkan pencerede “Tüm Alanları Göster” (Output All Fields) kısmındaki değeri NO yaparsanız sadece sizin seçtiğiniz alanlar listelenecektir.



Üçüncü örneğimizde (Query3 resmi) Ad, Soyad, No, soyad, Ad alanları seçilmiş.

SELECT Ad, Soyad, [No]

FROM Kisiler
WHERE (([No])<100) AND (Soyad) Like "A*")
ORDER BY Soyad DESC, Ad;

1. No alanının parantez içine alınmasının sebebi “NO”nun özel bir kelime olup yorumlayıcı tarafından “Hayır” anlamında anlaşılmamasıdır.
2. En sağdaki Soyad ve Ad alanlarının ikinci kez kullanılmasının amacı sıralamayı sütunların sıralanmasından farklı yapabilmek. Yani önce Ad’a sonra Soyad’ göre sıralamak isteseydik baştaki Ad ve Soyad alanlarında sıralama seçerdik. Ama biz önce Soyad’ı istediğimiz için son tarafa bunları koyup Göster (Show) işaretini kaldırdık. (Aynı şeyi 2. soyad alanını birincinin olduğu yere taşıyarak da yapabilirdik. Zira, hem ORDER BY hem de SELECT için seçili alanların sırası aynı olacak.)
3. Sadece Göster (Show) değeri işaretli olan alanlar görüntülenir. Bunlar SELECT kısmına dahil edilir. (Tüm alanları gösterme özelliği kapalıysa tabii.)
4. Dördüncü sütundaki LIKE cümlecğini No alanının alt satırında kullanmamızın sebebi VEYA sorgusu elde edebilmek. Yani “ NO<100 OR SOYAD LIKE ‘A*’ ” gibi. İkisini aynı satırda kullanırsak VE sorgusu elde ederiz ki bu, cümlemizdeki “OR”un “AND”a dönüşeceği anlamına gelir.

Bu tasarım modunda ne kadar karışık ya da sade yaparsak yapalım Access evirip çevirip kendi bildiği gibi bir SQL kodu üretecektir.

SELECT Kisiler.Ad, Kisiler.Soyad, Kisiler.[No], *
FROM Kisiler
WHERE (((Kisiler.[No])<100)) OR (((Kisiler.Soyad) Like "A*"))
ORDER BY Kisiler.Soyad DESC , Kisiler.Ad;

Beşinci madde olarak da şunu ekleyeyim o zaman: Alan isimlerinin başına tablonun ismi yazılabilir. Bu, birden çok sayıda kullanılan tabloların olduğu sorgularda daha işe yarar olur.

EXCEL’DEKİ GİBİ OTOMATİK SAYI SERİLERİ OLUŞTURMAK

Access’te Excel’deki gibi artan satırlarda otomatik sayı oluşturmak ne iyi olurdu diye düşündüğünüz olduysa ilk fırsatta deneyin derim. Kayıtlar arasında aşağı giderek seri olabilecek sayılar girin. (Yani 1, 3, 5, 7 .. gibi.) Oklarla aşağı doğru gittikçe yeni değerleri anlayıp sizin yerinize Access yazacaktır.

KENDİ FORMÜLÜNÜ KENDİN HESAPLA

Kullanıcının kendi formülünü hesaplayabilmesini sağlamak istediğiniz oldu mu hiç? EVAL işlevi bu iş için biçilmiş kaftan. Siz formül metnini veriyorsunuz, o da hesaplıyor:

MsgBox Eval(“6^2 + 4*3 – Log(Exp(10))”)

OTOMATİK SAYI

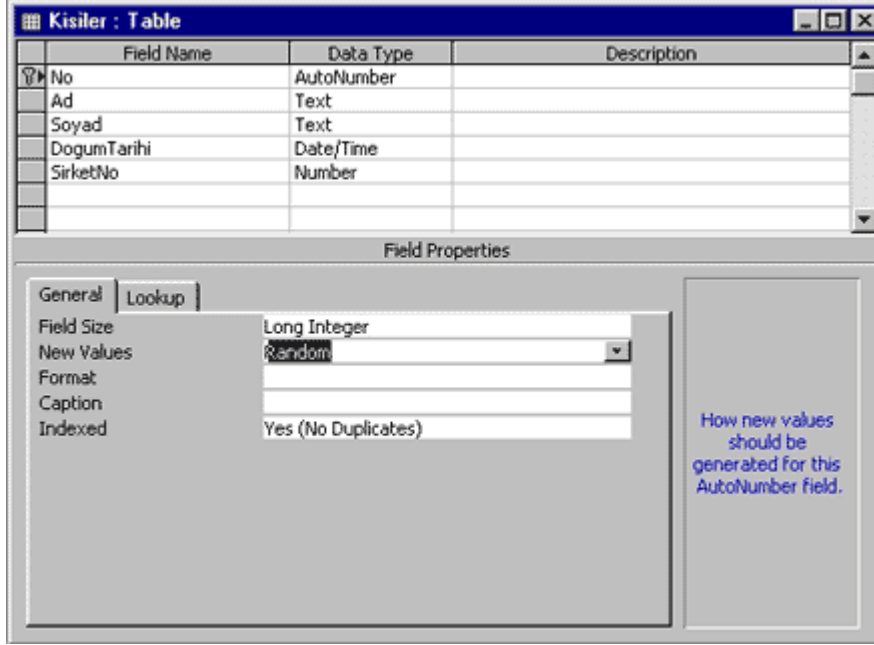
Bazan bu isim kendi kendine çalışan sayılar varmış gibi çağrışımlar uyandırıyor bende. Otomatik çamaşır makinası gibi. Gelgelelim bazı arkadaşlar bunun marifetlerinden pek muzdaripler. “Otomatik sayı, kayıt numarası neyse onu versin”, “Yeni kayda geçip vazgeçtiğimde numarayı alıyor, bir daha aynı numarayı kullanamıyoruz”, “Bütün kayıtları sildiğim halde 1’den başlamıyor” gibi yakınmalar oluyor.

Efendim, bu otomatik sayı öyle birşeydir ki görüntü açısından bir anlam taşıması için üretilmemiştir. Diğer bütün sistemlerde olduğu gibi en büyük amacı belli bir kayda özel bir kod vererek istendiğinde ona kolayca ulaşabilmeyi sağlamaktır. Mesela bütün alanların değerleri aynı olan iki kaydı birbirinden ancak Otomatik Sayı değerleri ile

ayırabilirsiniz. Yani arkaplan işlemleri için kullanılır. Ama isterseniz ekranda da değerini gösterin.

Bunun artan olacak şekilde değerler almasını ayarlayabileceğiniz gibi rasgele değerler almasını da sağlayabilirsiniz. Tablo tasarımı görünümünde New Values (Yeni Değerler) özelliğini Random (Rasgele) yaparak.

Yine de programı kullanacak kişiler açısından bir daha tekrarlanmayacak bir fiş numarası, kod numarası, etiket numarası gibi bir değer elde edilmiş olur. Access'teki replikasyon işleminde de kullanılabilir AutoNumber alanlar.



ACCESS HİKAYELERİ

FARE MENÜSÜYLE MP3 LİSTESİ

Şimdi bunun Access'le ne ilgisi var? Yok tabii. Ama dolaylı olarak Access'le çalışırken MP3 dinlemek istiyor insan. Hele bu eski makinemde Access'le Winamp'ı aynı anda çalıştırmak istemeyen biri olarak Media Player tercihim. Media Player madem MP3 çalışıyor, M3U da çalışıyor. Hatta M3U liste dosyasının özel bir biçimde yazılmış olmasına gerek yok. Yeni bir metin dosyasının içine MP3 dosyalarının isimlerini klasör adlarıyla birlikte altalta yazın. Metin dosyasını kapatıp uzantısını M3U yapın ve Media Player'da açın.

Yeni bir metin belgesi daha oluşturup şunları aynı şekilde yazın:

```
@echo off
dir %1\*.mp3 /on /b /a-d >mp3ler.m3u
echo Liste oluşturuldu... OK...
echo Bu pencereyi kapatabilirsiniz...
echo.
```

Sonra bunu mp3liste.bat adıyla c:\ içine kaydedin. Şimdi de Windows Gezgini'ni (Explorer) açıp Görünüm/Seçenekler menüsünü çalıştırın ve Dosya Türleri (FileTypes) sayfasına geçin. Listedeki Klasör'ü (Folder) seçin. Yeni (New) tuşuna basın. Üstteki yere "MP3 Listele", alttaki yere de "c:\mp3liste.bat %1" yazın ve Tamam (OK) tuşuna basın. Yine Tamam (OK) tuşlarıyla işlemleri onaylayın.

Böylece windows'ta herhangi bir klasörün üzerine geldiğinizde sağ fare menüsündeki MP3 Listele komutunu kullanarak o klasördeki MP3'leri aynı klasörün içinde "mp3ler.m3u" dosyasına kaydedebileceksiniz.

Eğer "dir %1*.mp3 /on /b /a-d /s >mp3ler.m3u" şeklinde "/s" eklerseniz tüm alt klasörleri arayarak klasör isimlerini de

dosyaların başına ekler.

Fakat C: D: gibi bir kök dizinde bunu yapıyorsanız bu kod çalışmaz. Çünkü kök dizinlerde parametre olarak \ işareti gelirken klasör isimlerinde bu ek yoktur. O zaman "dir %1*.mp3..." yerine "dir %1*.mp3..." yazmalısınız.



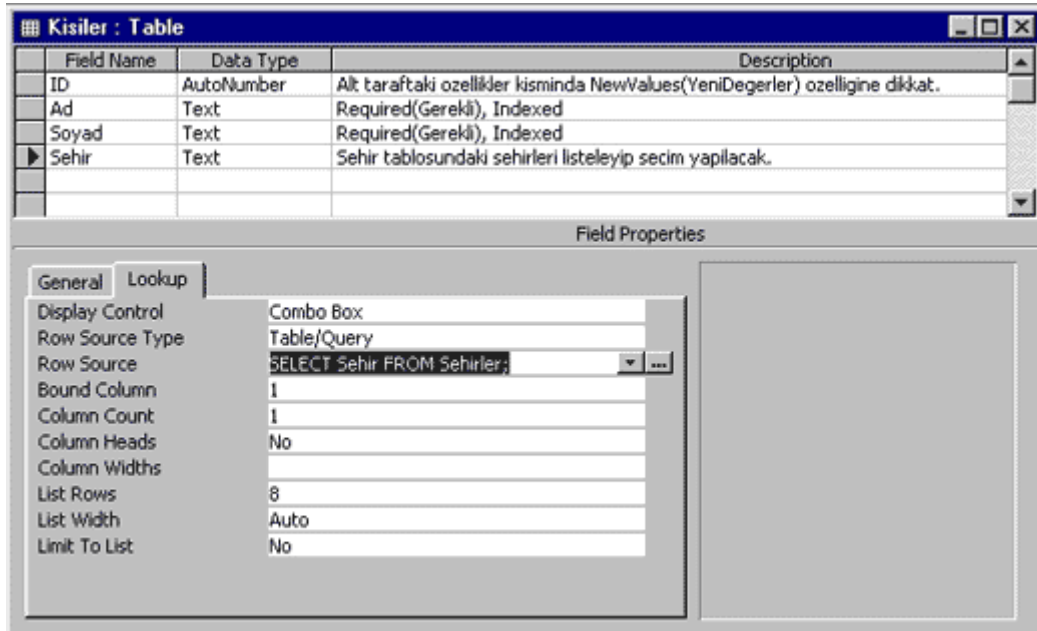
Biraz Keşif Biraz Kuşku

Şöyle elle tutulur ipuçları vermek bu ay vacip oldu. Access'in, sorgulama yeteneklerinin dışında, sorguları en umulmadık şekilde kullanabilme gibi yetenekleri de var. Gelgelelim işin ucunda biraz keşif ve biraz kuşku da var.

HAYAT AYRINTILARDA GİZLİ

Bazan en sıradan işlerde dikkat etmediğimiz bazı ayrıntılar birçok kolaylıklar getirebiliyor. Mesela tablo tasarımlarında pek dikkat etmediğimiz Lookup adlı bir özellik sayfası var. Bunu kullanarak tablolara kayıt girerken forma gerek kalmadan açılır listelerden seçim yaptırılabilir. Tablo görünümündeki süzme yetenekleri de cabası.

Tablo tasarımında iken bir alan tanımlayıp altta özellikler kısmındaki Lookup sayfasından ComboBox'u (Açılan Liste Kutusu) seçin. Sonra RowSource (Satır Kaynağı) kısmına gerekli SQL kodunu yazın. Yani seçim yaptırmak istediğiniz listenin kaynağını.



Tablolara kayıt girerken değerlerin listeden seçilmesini sağlayabilirsiniz.

Lookup sayfasındaki diğer özelliklere de dikkat etmekte fayda var. Örneğin Limit To List (Listeye Sınır) değerini YES olarak seçerseniz kayıt girişinde listedeki elemanlardan başka birşey yazılamaz. List Width (Liste Genişliği) içindeki Auto (Otomatik) değeri yerine bir sayı yazarsanız açılan listenin genişliğini belirlemiş oluyorsunuz.

Oradaki diğer özelliklerin fazlalığı dikkatinizden kaçmadıysa daha başka şeyler de anlatmamı bekliyor olmalısınız.

ColumnCount (sütun sayısı) değerinin varlığından Lookup işleminde birden fazla sütun gösterilebileceğini anlıyoruz.

Field Name	Data Type	Description
ID	AutoNumber	Alt taraftaki özellikler kısmında NewValues(YeniDeğerler) özelliğine dikkat.
Ad	Text	Required(Gerekli), Indexed
Soyad	Text	Required(Gerekli), Indexed
SehirID	Number	Bu alan şehirlerin numarasını tutuyor.

Property	Value
Display Control	Combo Box
Row Source Type	Table/Query
Row Source	SELECT ID, Sehir FROM Sehirler;
Bound Column	1
Column Count	2
Column Heads	No
Column Widths	0cm;8cm
List Rows	8
List Width	Auto
Limit To List	Yes

Seçim listesinde istediğiniz sayıda sütunu gösterebilirsiniz. Tabii ki bunlar SQL metni ile seçiliyor.

Resimde Column Widths (Sütun Genişlikleri) değerini “0cm;8cm” yapmamızın sebebi ilk sütunun gösterilmemesini istememiz. (İlk sütunun genişliği 0, ikincisinininki 8 cm.) Çünkü bir listeden herhangi bir şehir seçtiğimizde aslında onun numarası (yani görünmeyen kolondaki değer) kayıtlara girecektir.

BoundColumn değerine bakın. Onun 1 olması, SQL sorgusu ile gelen birinci sütunun kayda gireceğini gösteriyor. Eğer bu sütun, SQL sorgusu içinde ikinci sırada olsaydı BoundColumn değerini 2, sütun genişliklerini “8cm;0cm” olarak değiştirecektik. Bu arada ListWidth (Liste Genişliği) değeri “Auto” olduğu için açılan liste kutusunun genişliği aktif sütunun genişliği kadar olur. O yüzden sütun genişliklerindeki diğer değerler fazla bir önemi yok şu durumda. ListRows (Listedeki Satır Sayısı) ile de açılan listenin satır sayısı belirlenebiliyor.

Burada Access’çilerin koyduğu birkaç kural var. Mesela açılan listenin BoundColumn sırasındaki alanı görünmez olursa LimitToList değerini YES yapamıyoruz. ColumnHeads (Sütun Başlıkları) değerinin ne olduğunu da tahmin ediyorsunuzdur artık.

ID	Ad	Soyad	Sehir	Meslek	Cinsiyet	Askerlik	GirisTarihi
3	ahmet	beyaz	Bursa	Doktor	Erkek	<input checked="" type="checkbox"/>	
4	hülya	demir	Trabzon	Mühendis	Kadın	<input type="checkbox"/>	
10	murat	kalan	İstanbul	Öğrenci	Erkek	<input checked="" type="checkbox"/>	
2	ali	kara	Sehir	Telefon	Plaka	Bolge	02.0
8	ayşe	kural	Adana	332	01	Akdeniz	03.0
*	ber		Adıyaman	416	02	Güneydoğu	
			Afyon	272	03	İç Anadolu	
			Ağrı	472	04	Doğu Anadolu	
			Amasya	358	05		
			Ankara	312	06	İç Anadolu	
			Bolu	374			
			Bursa	224		Marmara	
			Diyarbakır	412			
			Eskişehir	222			
			Hakkari	438		Güneydoğu	
			İçel	324			
			İstanbul	212	34	Marmara	
			İzmir	232	35	Ege	
			Kocaeli	262	41	Marmara	

TABLO KULLANMADAN DEĞER LİSTESİ OLUŞTURMAK

Hepsi bu kadar mı? Değil tabii. Access’çiler ayrıntıları artırıp bu yazının uzaması için ellerinden geleni yapmışlar. RowSourceType (Satır Kaynağı Tipi) özelliğine bakın şimdi.

Şimdiye kadar SQL sorgusu ile aldığımız verileri listeleyebiliyorduk. Şu halde RowSourceType değerini “ValueList” (Değer Listesi) yapıp hemen altındaki RowSource kısmına da “İstanbul;İzmir;Ankara” yazın. Böylelikle ille bir tablodan veri alıp seçtirmeye gerek kalmıyor. Sütun sayısı ve genişliklerini de buna kolayca uyarlayabilirsiniz. Yani iki sütun gösterip “34;İstanbul;35;İzmir;06;Ankara” şeklinde bir listeyi kullanabilirsiniz. O zaman genişliği “0;8” şeklinde ayarlamalısınız.

RowSourceType kısmındaki “FieldList” (Alan Listesi) seçeneğinin ne işe yaradığını anlatmayacağım. Kolayca tahmin ettiğiniz şeyi isterseniz deneyin. Ya da ne olduğunu zaten biliyorsanız boşverin.

KENDİ LİSTENİ KENDİN YAP

“Kendin pişir, kendin ye!” serimizin bu bölümünde Lookup ile ilgili ufak bir ipucunu vermeden geçmeyeceğim.

Excel kullananlar bilirler. Uzunca bir listeyi yazarken listenin yukarısında daha önce yazdıklarınız aktif hücrede önerilir. Belki aynı şeyi yazmak istiyorsanız kolaylık olsun diye. Biz de bunun Access’çesini yapalım diyorum.

Mesela “Kisiler” adlı tablomuzun “Meslek” adlı bir alanı olsun. Ama ayrıca bir “Meslekler” tablosu yapmak istemiyoruz. İstedığımız şey tabloya kayıt girerken o ana kadar girdiklerimizden seçim yapabileceğimiz bir liste oluşturmak.

Önce “Kisiler” tablosunu tasarım görünümünde açıp “Meslek” adlı alan için dizin (indeks) oluşturalım. Ki, kayıt girişinde hızlı çalışsın. Sonra alttaki Lookup sayfasına geçip DisplayControl değerini ComboBox (Açılan Liste Kutusu) yapın. İki altındaki RowSource satırına “SELECT Meslek FROM Kisiler GROUP BY Meslek HAVING (Meslek Is Not Null) ORDER BY Meslek;” yazın (tırnak işaretleri olmayacak şekilde). Başka bir şey yapmadan kaydedin ve tabloyu kayıt listesi görünümünde açın. Artık Meslek alanı üzerinde, eski yazdıklarınızı sıralı bir liste halinde görüp onlardan seçim yapabilirsiniz. Fakat yeni girilen kayıtlar listede yer almayıp tabloyu sonraki açışınızda görüntülenir.



Kendi listesini kendi kayıtlarından seçen bir veri alanı.

GEÇERSİZ VERİLERİ ANINDA KONTROL EDİN

Bütün bu özellikler form ve raporlarda da aynı şekilde kullanılabilir. Ek olarak Validation Rule ve Validation Text yani geçerlilik kısıtlamalarından bahsetmek gerek. Bir veri alanının alabileceği değerler bu özellikleri kullanarak kısıtlanabilir. Mesela doğum tarihi bilgisini içeren bir alana 1900 yılından önce doğmuş kimse kalmadığını düşündüğünüz için oraya girilen değerin çok saçma olmamasını istiyorsunuz. Tarihin ille de 1900 yılından büyük olmasını sağlamak istediğinizde Validation Rule (Geçerlilik Kuralı) değerine “>#1.1.1900#” (tırnak işaretleri olmayacak şekilde) yazmalısınız. Uyarı mesajı olarak hemen alt satırdaki ValidationText (Geçerlilik Kuralı Metni) değerine doğru dürüst bir tarih değeri yazılmasını söyleyen bir uyarı metni ekliyorsunuz: “1900 tarihinden önce bir doğum tarihi girdiniz, buna katlanamam!”.

Bu ayarları tablonun tasarım görünümünde yaptığımızı tekrar hatırlatayım. Tüm alanlar için bu kuralı uygulayabilirsiniz. Ama benim asıl anlatmak istediğim bu değil. Bir adet de olsa tablonun kendisine geçerlilik kuralı tanımlama imkanımız var. Mesela kişinin cinsiyeti “Kadın” olarak seçildiğinde askerlik yapıldığına ilişkin bilgi giriliyorsa ikaz etmek gerekir.

Tablo tasarımında iken View/Properties komutu ile açılan Özellikler sayfasında bu değerler tanımlanabiliyor. Önce geçerlilik kuralını tam anlamıyla formülize etmemiz gerekiyor.

Bu işlemi kod ile yapmak istiyorsanız önce başvurular kısmında (Tools/References – Araçlar/Başvurular) Microsoft DAO 3.6 Object Library (DAO 3.6 ya da DAO 3.5) kütüphanesinin seçili olmasına dikkat edin. Herhangi bir modül sayfasında alttaki kodu çalıştırın.

Dim db As Database

Set db = CurrentDb

Dim td As TableDef

Set td = db.TableDefs("Kisiler")

td.ValidationRule = "Cinsiyet<>'K' or Askerlik=FALSE"

td.ValidationText = "Cinsiyet KADIN olarak seçildiğinde ASKERLİK bilgisi girilemez."

Tabloda birkaç alan arasında ortak geçerlilik kuralını bu şekilde kolayca tanımlayabiliriz. Alanların kendilerinde ise sadece kendileri için kural tanımlanabiliyor. Eğer tabloda yine kuralımız olacak ise onu da uygun bir kural metni oluşturup bu şekilde kaydedebilirsiniz. Mesela işe giriş ve işten çıkış tarihinin bir büyüklük sırası vardır. O halde bunun kuralını aynı tablomuza şu şekilde ekleyebiliriz:

td.ValidationRule = "(Cinsiyet<>'K' or Askerlik=FALSE) and (GirisTarihi<CikisTarihi)"

td.ValidationText = "Girilen değer kısıtlama kurallarına aykırı."

Ana koddaki son iki satırı bu şekilde değiştirmek yeterli. Sonrasında tabloya bu kurallara aykırı olabilecek kayıtlar girmeyi deneyin.

ÇEK BİR TABLO, FİLTRELİ OLSUN!

Tablo tasarımındaki güzellikler dışında süzme ve sıralama seçenekleri de yeterince cazip. Veri girişi görünümünde tabloyu istediğimiz gibi evirip çevirmemize izin veriyor Access.



Birazdan Ad alanı içinde 'ahm' olan kayıtlar görünecek.

Ad alanı üzerine gelip süzgeç kısmına "*ahm*" yazdığımızda içinde "ahm" olan isimlerin bulunduğu kayıtlar listelenecek. "*" işaretinin ve "Like" deyiminin nasıl kullanıldığını biliyorsunuzdur. Bu yazdığımız aslında ' Like "*ahm*" ' şeklinde kullanılıyor

Ek olarak süzgeç içinde birden fazla alan adını kullanabilirsiniz. Herhangi bir alanın üzerine gelin ve süzgeç kutusuna şunu yazın:

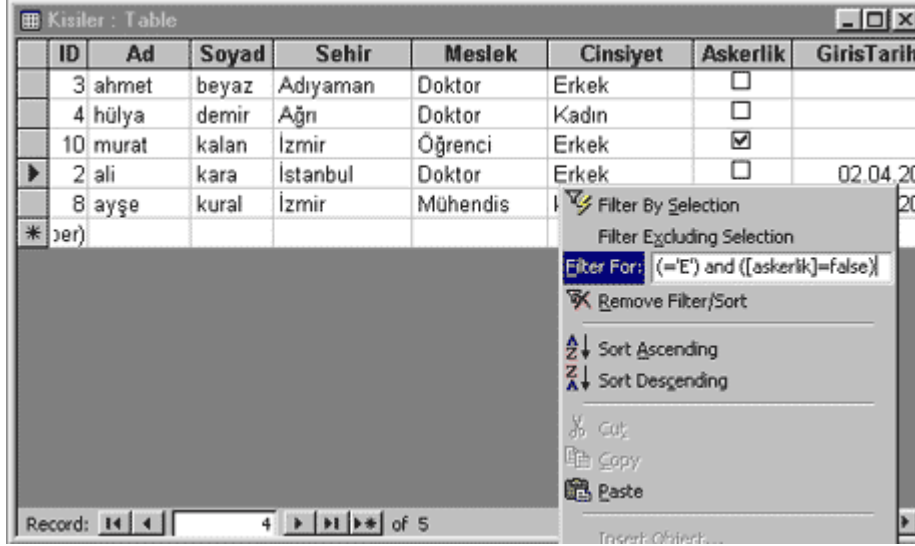
[cinsiyet] = 'e'

Ya da Cinsiyet alanı üzerine gelip şunu yazabilirsiniz:

(='e') and ([askerlik]=false)

Veyahut da CikisTarihi üzerine gelip şirkette bir yıldan uzun süre çalışmış kişileri şu şekilde kolayca bulabilirsiniz:

>=GirisTarihi + 365



Birden fazla veri alanı için süzleme yapılabilir.

Süzgeç menüsündeki diğer komutları da incelemenizi tavsiye ederim. “Filter By Selection” (Seçime Göre Süz), “Filter Excluding Selection” (Seçim Dışında Tutarak Süz), “Remove Filter/Sort” (Süzgeç/Sıralama Kaldır) komutlarını Türkçe anlamlarını vermek dışında anlatmaya gerek yoktur herhalde. Aynı şekilde “Sort Ascending” (Artan Sıralama) ve “Sort Descending” (Azalan Sıralama) seçenekleri de sıralama yapıyor bildiğiniz gibi.

Bu işlemleri aynı anda birden fazla sütun için de yapmak mümkündür. Sütun başlığından tutup birkaç alanı seçin. Fareyi seçili bölge içinde kalan verilerin üzerine getirip fare menüsünü açın. Orada sıralama komutlarını kullanarak seçili alanlara göre ardışık sıralama yaptırabilirsiniz. Menüün aşağısındaki HideColumns, ColumnWidths komutlarını da seçili tüm alanlar için uygulayabilirsiniz.



Birkaç alanı birden seçerek ardışık sıralama yaptırılabilir.

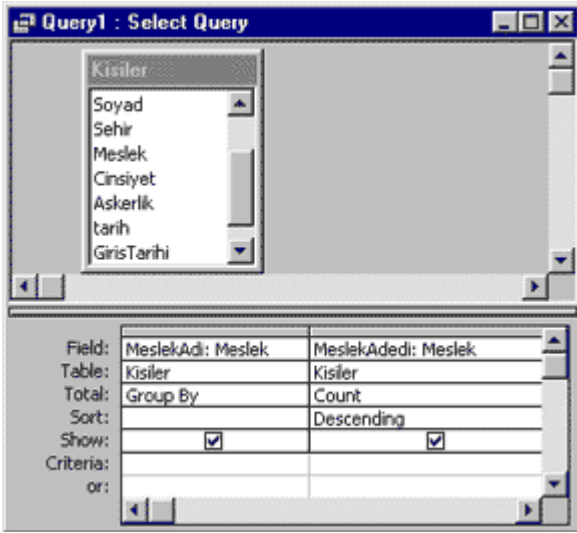
HideColumns (Sütunları Gizle) komutunu seçip de alanları gizleyip kaybedenler için Format/UnhideColumns (Biçim/Sütunları Göster) menüsünün kullanılabileceğini söyleyeyim.

Tablo tasarımlarında yapılan bu ayarlamaların bir faydası daha var. Zaten bütün bunlar form tasarımlarında da yapılabildiği için sihirbaz ile formunu hazırladığınız bir tabloda yaptığınız bu düzenlemeler forma aynı şekilde yansıtılıyor. Mesela Lookup ile listesini başka bir yerden alan veri alanları için aynı ayarların olduğu Combobox (Açılan Liste) yerleştiriliyor forma.

Hatırlarsanız geçen ay AutoNumber (Otomatik Sayı) hakkında bilgiler vermiştik. İşte orada bir noktaya değinmedik. Otomatik Sayı alanları için yeni kayıt anında vazgeçilmesi ile o numaranın bir daha kullanılmaması çok kullanıcıli ortamlarda da gerekli. Yani aynı anda beş kişinin kayıt girdiği bir tabloda birinin kayıttan vazgeçmesi otomatik sayının o anki değerinin bir daha kullanılmamasını gerektirir. Aksi halde yönetmek daha masraflı olsa gerek.

HERKES İÇİN SORGULAR

Sıkça ihtiyaç duyulan bir sorgu türü, kayıtlar içinde belli bir değeri gruplayarak herbirinin adedini olduğunu bulmak. Örneğin Kisiler tablosundaki Meslek bilgisini gruplayıp her bir meslekten kaç adet olduğunu öğrenmek ve buna göre sıralamak istiyoruz. Önce yeni bir sorgu oluşturup Kisiler tablomuzu ekleyelim. Sonra View (Görünüm) menüsünden Totals (Toplamlar) komutunu seçelim. Bunu yapınca alttaki kısımda Totals (Toplamlar) satırı görüntülendi. Resimdeki gibi değerleri girip sorguyu çalıştırınca mesleklerin listesini ve herbirinin adedini sıralı bir şekilde görebiliriz.



GROUP BY ve COUNT seçenekleri ile her bir Meslek grubunun sayısını öğrenebiliyoruz.

KOD İLE FORM VE RAPOR TASARLAMAK

Aslında bu başlık kod ile tasarım yapma konusunu anlatmak için biraz yetersiz kalıyor. Zira kod ile sadece formlar ve raporlar değil, makrolar, kodlar değiştirilebiliyor ve hatta veritabanının bütünü başaşağı edilebiliyor.

Bir formu (yani nesneyi) değiştirip kaydedebilmek için normalden farklı olarak yapacağımız tek şey onu normal görüntüleme modunda değil de tasarım modunda açmak ve işimiz bitince kapamak.

Dim f As Form

'Tasarım modunda aç.

DoCmd.OpenForm "form1", acDesign

Set f = Forms("form1")

'Gerekli değişiklikleri yap.

f.Caption = "Kisilerle ilgili deneme formu"

f.RecordSource = "Kisiler"

f!text1.ControlSource = "ad"

f!text2.ControlSource = "soyad"

f!text1.Top = 100

f!command0.Left = 10

fltext2.Visible = False

'Değişiklikleri kaydederek kapat.

DoCmd.Close acForm, "form1", acSaveYes

MDE olarak derlenmiş dosyalarda bu şekilde tasarım yapılamıyor.

Dim Frm As Form, Buton As CommandButton

Set Frm = CreateForm

Set Buton = CreateControl(Frm.Name, acCommandButton, , , 200, 200, 2000, 500)

Buton.Name = "Buton1"

Buton.Caption = "Yeni Buton"

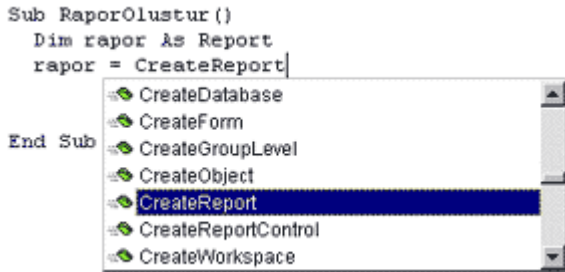
Frm.Caption = "Yeni Form"

DoCmd.Close acForm, Frm.Name, acSaveYes

'MsgBox Frm.Name & " adlı form oluşturuldu."

'Hatalı. Başına açıklama işareti koyduk.

Örnekteki gibi kolaylıkla yeni bir form da üretebiliyoruz. Bunu hem tasarım modunda hem de çalışma modunda yapma olanağımız var. Rapor yapmak için de CreateReport kullanılıyor.



Gelelim örnekteki hataya. Kodu çalıştırırken son satırda açık olmayan nesneye başvuru yapılamayacağına dair bir hata mesajı çıkıyor. Tanımlanmış olsalar da formlar kapatıldıkları anda kullanılamaz olurlar. Yani formu kapattığımız an Frm adlı değişken kaybolur. O halde son satırdaki “frm.Name” kısmını silmek gerekiyor.

İPİN UCUNDAN

Eğer bir formu ya da raporu gizli olarak açmak istiyorsanız DoCmd.OpenForm "Form1", acDesign, , , acHidden gibi altıncı parametrede acHidden değerini kullanın. Böylelikle kullanıcıya hiç göstermeden formu tasarım modunda açıp üzerinde istediğiniz değişiklikleri yaptıktan sonra kaydedip kapayabilirsiniz.

ASİSTAN ASİSTAN NERDESİN!

Office asistanından hoşlanıyorsanız onu programlamak da hoşunuza gidecektir. Öncelikle başvurular listesinden “Microsoft Office 9.0 Object Library” kütüphanesini seçmek gerekiyor. Ayrıca asistan da sistemde kurulu olmalı.

Application.Assistant.Visible = True

Application.Assistant.Animation = msoAnimationLookUpRight

Bu şekilde asistanı gösterip onu hareketlendirebilirsiniz. Asistanın Office’te yaptığı herşeyi kodla da yaptırabilirsiniz.


```

Sub Deneme()
    With Application.Assistant
        .Visible = True
        .Animation = msoAnimationLookUpRight
    End With

End Sub

```



BANA PROSEDÜRÜNÜ SÖYLE...

Diyelim ki DENEME adlı bir yordamımız var. Bunu normalde Call Deneme(...) ile çalıştırabiliriz. Ama RUN komutu bakın bunu nasıl yapıyor:

```

Sub Deneme(deger1 As String)
MsgBox deger1
End Sub

```

```

Sub RunIleCalistir()
Dim x
x = Run("Deneme", "Run ile deneme")
'Run komutu max 30 adet parametre alır.
End Sub

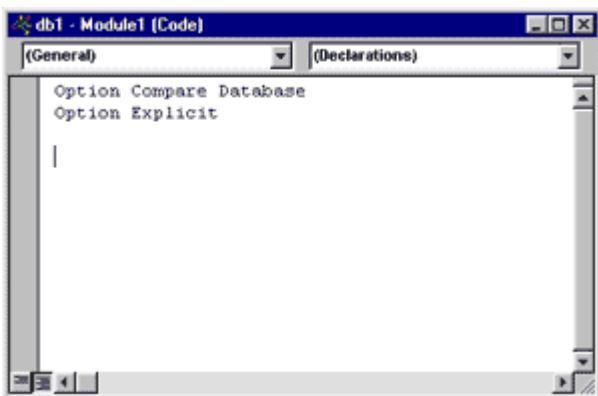
```

Kod Yazmaya Geçiş

Access modüllerinde kullandığımız dil VBA, yani Visual Basic for Applications. Office'in hemen her yerinde kullanılabilir. Hatta başka uygulamalarda da kısmen yer edinmiş görünüyor. AutoCad gibi bazı programlarda VBA kullanılabilir örnek.

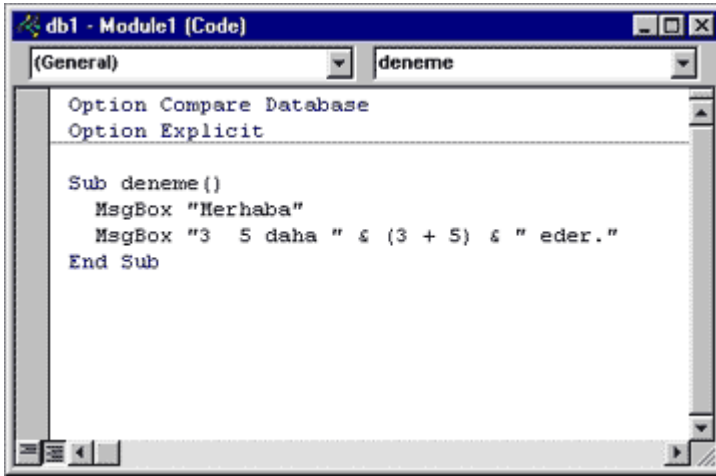
KOD DENEMEK İÇİN EN KISA YOL

Veritabanı ana penceresinde iken (Database Window) modül (Modules) sayfasına geçip yeni (New) bir modül oluşturalım. (Resim – db1 Module1 code)



Yeni bir modül penceresi.

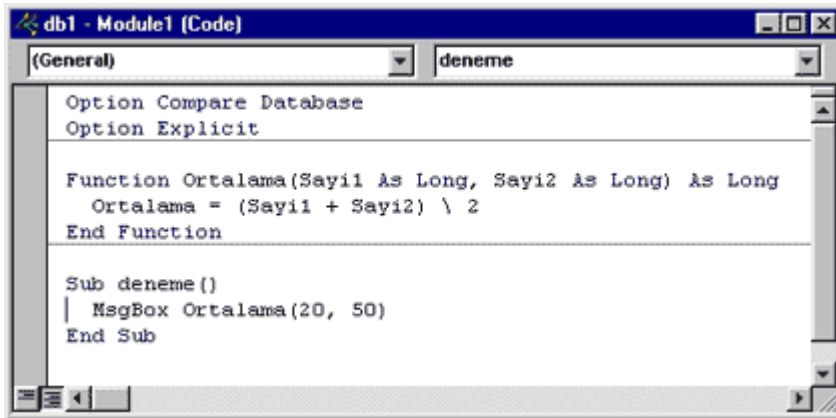
Herhangi bir kodu çalıştırmak için mutlaka "Sub" ile başlayan bir yordam oluşturmalıyız.



“Sub” ile “End Sub”un arasında herhangi bir yerde F5’e basın.

“Sub Deneme” yordamını aynen yazıp “Sub” ile “End Sub” arasında herhangi bir yerde F5 tuşuna basarsanız program çalışır ve iki kez mesaj gösterir. Bunun gibi End Sub’dan sonra yeni yordamlar ekleyebilirsiniz.

Ek olarak İşlev (Function) tanımından bahsetmek gerekiyor. Bu da “Sub” yerine “Function” yazılarak kullanılıyor ve “End Function” ile bitiyor. Tabii önemli bir farkı var. İşlevlerin asıl amacı geriye değer döndürmektir. Yani bir işlev çağırılırken bir şeye eşitlenir (istisna olabilir).



“Ortalama” adlı işlev örneğine bakın şimdi. Orada işlevi tanımladıktan sonra artık herhangi bir yerden bunu çağırıp kullanabiliyoruz. İsterseniz bunu sorgu yaparken SQL metinlerinizin içine de yazabilirsiniz.

İşlevler bir şart dışında, tek başlarına çalışmazlar. Eğer işlevlerin parametreleri yoksa bunlar “Sub” yordamları gibi çalışırlar.

Parametreler “Function” ya da “Sub” tanımlama başlığındaki parantezler arasında bulunan değerlerdir. İki parantezin arası boş ise parametre yoktur. Birden fazla parametre varsa aralarına virgül koyularak sırayla yazılırlar.

Şimdi varolan bir formunuzda düğme oluşturup “Tıklatıldığında” (OnClick) olay yordamına kod yazmak istiyoruz. Düğmenin ilgili özelliğini seçtiğinizde aşağıdaki gibi bir metin oluşturulur.

```
Option Compare Database
Option Explicit
Private Sub Buton_Click()
..
End Sub
```

“Private Sub” ile “End Sub” arasındaki kısma yazdığınız kodlar düğmeye basıldığında Access tarafından çalıştırılır. (“Private Sub” ve “Public Sub” deyimlerinden daha önce bahsetmiştik. Bu form için “Private Sub”un “Sub” kullanmaktan

farkı yok şimdilik.)

Şimdi bu düğme formu kapatma görevi üstlensin ve siz onun kod bölümüne şunu yazın:

DoCmd.Close

İlk kısımda modüllerle ilgili bahsettiğimiz yazım kuralları burada da geçerli. İsterseniz “Ortalama” işlevini bu düğmenin içinde kullanabilir, isterseniz işlevin kendisini bu formun içine alabilirsiniz. Bir veritabanı içindeki Genel (Global) yordamların aynı adı alamayacaklarını ve formlarda olduğu gibi raporlarda da aynı şekilde kod yazabileceğinizi söyleyeyim.

İPİN UCUNDAN AZICIK

Eğer sıkça kod yazmaya başlamışsanız Tools/Options/Form-Reports (Araçlar/Seçenekler/Form-Rapor) menüsündeki “Always use event procedures” (Her zaman olay yordamlarını kullan) seçeneğini aktif hale getirmeniz faydalı olacaktır. Bu, örneğin bir düğmenin tıklama (click) yordamına bir kod yazmak istediğinizde karşınıza herhangi bir soru çıkmadan, doğrudan modül sayfasında ilgili olay yordamının açılmasını sağlar.

LİSTEDE YOKSA EKLEYİVER

Açılan Kutu (ComboBox) nesnesini kullanırken yeni yazdığınız değer listede yoksa eklenmesini istiyorsanız aşağıdaki kodu deneyin. Önce nesnenin “Listede Yokken” (NotInList) olayının çalışması için kod sayfasını açın. Karşınıza alttaki satırlar gelecektir:

Option Compare Database

Option Explicit

Private Sub AcilanKutu1_NotInList(NewData As String, Response As Integer)

End Sub

“Private Sub..” ile “End Sub” satırları arasına Tablo1’deki kodu ekleyin.

Tablo1

```
If MsgBox("Yazılan değer listede yok. Eklensin mi?") = vbNo then exit sub
Dim db As Database
Set db = CurrentDb
db.Execute "insert into TABLO1(ALAN1) values (" & AcilanKutu1.Text & ")"
'db.Execute "insert into TABLO1(ALAN1) values (' " & AcilanKutu1.Text & " ')"
'Eger kayıt girilecek alan sayi degil de metin ise 2. satiri kullanin.
'Acilan kutudaki yeni metin ilgili tabloya eklenecek.
'TABLO1(ALAN1) yerine hangi tablonun hangi alanina ekleme yapilacaksa
'onlari yazin. KISILER(AdiSoyadi) gibi. tablo ya da alan isimleri
'hata verirse basina ve sonuna koseli parantez ekleyin.
'[kisi adlari tablosu]([adi soyadi]) gibi.

Dim deger
deger = AcilanKutu1.Text
AcilanKutu1.Undo
AcilanKutu1.Requery
Response = acDataErrAdded
'yeni kaydi ben simdi ekledim. hatayi yoksay.
AcilanKutu1.Value = deger
```

DEBUG WINDOW MU DESEM, HATA AYIKLAMA PENCERESİ Mİ DESEM...

Programı durdurup izlemek yapılan şeyleri gerektiğinde Locals Window ve Immediate Window'dan çokça faydalanabiliyoruz. Immediate Window programa ara verildiğinde (pause) o an çalıştırabileceğimiz kod satırları girmemizi sağlar. Tabii programın orjinal kodları değişmez, sadece o an için ayrı bir pencerede kod yazıp çalıştırma imkanımız oluyor.

LOCALS WINDOW İLE HERŞEYİ GÖRÜN

Şimdi herhangi bir veritabanı dosyamızı açıp içinde ne var ne yok, bakalım. Önce yeni bir modül sayfası açıp şu yordamı ekleyelim:

```
Sub Deneme()  
Dim DB as Database  
Set DB=CurrentDb()  
End Sub 'bu satıra gelip F9 tuşuna basın.
```

Locals Window'un amacı yereldeki aktif değişkenlerin (bellekteki bize ait bilgilerin) görüntülenmesini sağlamak. Bu durumda biz ne kadar değişken tanımlarsak, onu Locals Window'dan izleme olanağımız var. Öyleyse bir CurrentDB tanımlamak gerekiyor.

Locals Window'u ancak program çalıştığı durumda izleyebiliriz. Bunun için en son satırın üzerinde iken F9 tuşu ile bir "breakpoint" oluşturalım. Bunu yapınca satırın rengi değişecektir. (İptal etmek için yine F9'a basılıyor.)

F5 ile kodu çalıştıralım. F9 ile işaretlediğiniz satıra gelince program bekler. Breakpoint'lerin görevi bulunduğu satıra gelindiğinde programı durdurmaktır (pause). Şimdi View / Locals Window menüsünü açalım.

Benim örneğimde bellekte aktif olan iki nesne de pencerede görünüyor. Yanındaki (+) ile listeyi açarak FRM adıyla tanıttığım formun Caption değerini buluyorum. Orada "Form1" yazıyor.

Bu esnada yine View menüsünden Immediate Window'u açacağız. Oraya "? Frm.Caption" yazıp enter'a basın. Hemen altına Frm'nin başlığını yani "Form1" yazdı. Şimdi de (Frm.Caption="Forma yeni isim") yazın. Sonra yeniden "Frm.Caption"u yazdırmayı deneyin. Gördüğünüz gibi program durdurulmuş halde iken istediğimiz gibi müdahale yapabiliyoruz.

ADO GELDİ MERTLİK BOZULDU

Acces2000'de veritabanı işlemleri için ADO kütüphanesi kullanılıyor. Doğrudan bu seçili geldiği için biz babadan kalma DAO'muzu kullanacağız şimdilik. Referanslar kısmından ADO'yu kaldırıp DAO'yu seçin.

Örneğin dosyamızdaki tüm form ve raporları kodla listelemek istiyoruz. Şöyle bir yordam yazıp çalıştırın:

```
Sub Deneme()  
Dim db As Database  
Set db = CurrentDb()  
End Sub 'buraya gelip F9'a basın.
```

Şimdi Immediate Window'da şu satırları çalıştırın:

```
? db.containers("Forms").Documents.count  
? db.containers("Forms").Documents(0).Name  
Application.Assistant.Visible=True
```

Birincisinde dosyamda kaç adet form olduğunu, ikincisinde 1. formun adını öğrendim (0 numarası 1. formu işaret ediyor). Diğer nesnelerin de listesini bulmak istiyorsanız program bu halde dondurulmuş iken Locals Window'u kullanarak

DB/Container/Documents öğelerini incelemelisiniz. Üçüncü satırı da siz tahmin edin artık.

Şimdi Immediate Window’da şu komutu çalıştırın:

MsgBox "F5 tuşuna basıp programı devam ettir."

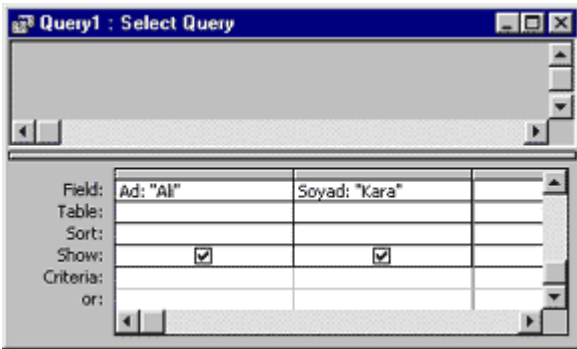
Bu arada (?) işaretinin “print” anlamında kullandığını söyleyeyim. İsterseniz doğrudan “print” yazabilirsiniz. Değer değiştirmek için de Immediate Window’a gerek yok her zaman. Locals Window’da ilgili değeri bulup seçerek değerini değiştirmeyi deneyin.

Bu konuyla ilgili bir de Watch Window var. Onun içine kendimiz değer tanımlı ekleyebiliyoruz. Örneğin biraz önceki örneğimizde bulunan DB değişkenini eklersek Locals Window’da olduğu gibi bunun tüm alt değerlerini görebiliyoruz. Yani bu, Locals Window’un bizim değiştirebildiğimiz hali sayılabilir. Ek olarak buna formül de eklenebiliyor.

HERKES İÇİN SORGULAR - TABLO OLMADAN SORU YAPMAK

Tablo olmadan sorgu olur mu dersenez, olur. Hani daha önce her SQL cümlesinde mutlaka olması gereken iki bileşenden, “Select” ve “From” deyimlerinden bahsetmiştik. Sonra da SQL’in standart olanı ve standart olmayanı olduğunu söylemiştik. Bu da standart olmayan bir SQL kodu işte.

SELECT "Ali" AS Ad, "Kara" AS Soyad;

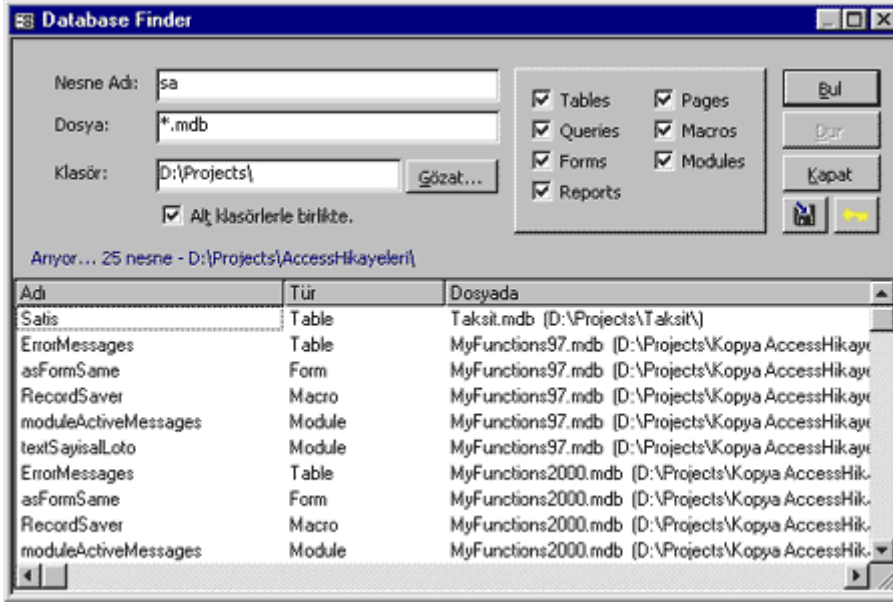


Tablo seçmeden de sorgu oluşturabilirsiniz.

ACCESS HİKAYELERİ İLK ADD-IN’İMİZ

Access, add-inleri çalıştırmak için dosyada bir işlev arar ve onu çalıştırır. O işlevi biz hazırlarız ve içine programın başlaması için gereken komutları yazmalıyız. Tabii add-in’imizi de Access’e tanıtmak için ayarlar gerekiyor.

Önce “uSysRegInfo” adlı bir tablo oluşturulur. Haliyle add-in’in ilk yüklenmesi için Access’in bu tabloyu kullandığını söylemeliyim. Fakat Add-in’ler registry’de kayıtlı olduğu için bu zorunlu değil ama kolaylaştırıcı. Yani Access add-in yüklerken ya da kaldırırken registry’deki birkaç değeri düzeltmekten başka birşey yapmıyor.



Gelecek yazı Access'te eklenti (add-in) oluşturma ile ilgili.

İlk Eklentimiz

EKLENTİ (ADD-IN) OLUŞTURMAK

1) Hazırlamış olduğunuz Access programınızın içinde aşağıdaki gibi uSysRegInfo adlı bir tablo oluşturun ve listedeki 4 kaydı tabloya girin.

2) Tabii öncelikle Add-in'imizin bir adı olması gerektiği için bizimkinin adını "Edinim" koydum.

Subkey	Type	ValName	Value
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\9.0\Access\Menu Add-Ins\Edinim	0		
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\9.0\Access\Menu Add-Ins\Edinim	1	Library	C:\Program Files\Microsoft Office\Office\Edinim.mda
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\9.0\Access\Menu Add-Ins\Edinim	1	Expression	=Edinim_Entry()
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\9.0\Access\Menu Add-Ins\Edinim	4	Quicktip	108409

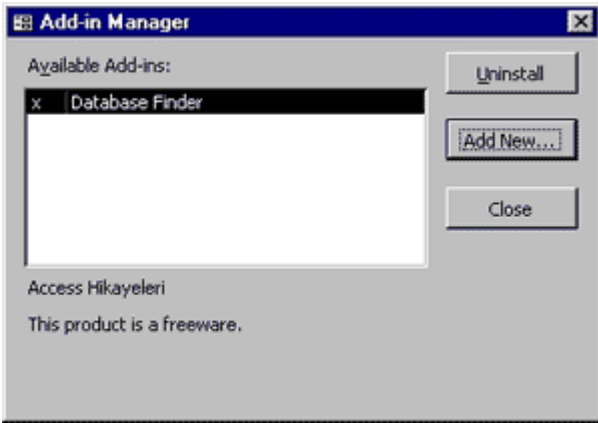
3) Type adlı alan sayı tipinde, diğer alanlar ise 255 uzunluklu metin alanları olmalı ve eğer Access2000 değil de Access97 kullanıyorsanız Subkey alanındaki değerlerde "9.0"ı değiştirip "8.0" yapmalısınız.

4) uSysRegInfo tablosundaki "=Edinim_Entry()" değeri bu add-in'in başlatılırken çağrılacağı işlevin adıdır. (Yani add-in başlarken ana formu açmak istiyorsanız, herhangi bir modülün içinde "Edinim_Entry" adlı parametresiz bir işlev (function) oluşturun ve kod kısmına ana formun açılmasını sağlayacak kodu yazın. Bu, add-in'in başlatılacağı yerdir.)

5) Kodlarımızın gizlenmesi ve programımızın bir EXE gibi hızlı çalışabilmesi için bu dosyayı önce MDE haline getirmeliyiz ve sonrasında dosyanın uzantısını MDA yapmalıyız. (Laf aramızda, MDB dosyalarının uzantısını MDA uzantılı yapmak da kâfi.)

6) Dosyayı uSysRegInfo tablosunda belirttiğiniz klasörün içine kopyalayın ve Access'i açıp Add-in yükleme kısmından bu dosyayı yükleyin. (Tools/Add-ins/Add-in Manager ile yapılıyor. Tahmin edeceğiniz gibi bu da Access'te yapılmış bir add-

in.)



Add-in yüklemek için Tools/Add-ins/Add-in Manager menüsünü çalıştırın.

Add-in'lerin amacı Access programcısına kolaylık sağlamak.

Önceki bilgilerimizi yoklayarak CodeDB ile CurrentDB'nin ne işe yaradıklarını hatırlayalım. Ek olarak fazla bir şey yapmaya gerek yok. Add-in ile normal bir Access programı ile yaptığınız her türlü işlemi yapabilirsiniz. Ama asıl amacı üzerinde çalışılan projeye yardımcı olmak. Örneğin geçen ay bahsettiğimiz Database Finder adlı ilk add-in'imiz, Access'te çalışırken diğer yandan bilgisayardaki tüm Access dosyalarını tarayarak aradığımız isimdeki nesneleri (tablo, form, sorgu, makro vs.) arkaplanda bulup listeliyordu. Aynı şekilde Access97'deki "SwitchBoard Manager" add-in'i Access içinden otomatik menü formları yapmamızı sağlıyordu.

İP ÜSTÜNE UN SERMİŞ

Bu ay ne ettiysem Hoca bana ipucu vermedi. Neymiş, ipin ucuna un sermiş. İş başa düşünce Bay Akses'inkileri aratmayacak ipuçları keşfettim, laf aramızda.

İKİ FORMUN KAYIT KÜMESİNİ BAĞLAMAK

Diyelim ki bir adet liste formunuz ve bir adet bu listeden seçerek ayrıntısını göstermek için kullanmak istediğiniz ayrıntılı bilgi formunuz var. Haliyle bu iki form, kayıtlarını aynı tablodan almak zorunda. Tek fark birinin liste halinde, diğerinin sayfa halinde göstermesi. Bu durumda kod yazıp "FindFirst" gibi bir komutla ayrıntı tablosundaki kayda ulaşmak gerekiyor. Fakat şu yolu denerseniz, öyle aramayla, bulmayla neyin uğraşmanıza gerek kalmayacak. Kisiler adlı formda bir kayıt seçip detayını görmek için Kisiler_Detay adlı formu açmak istediğinizi varsayarsak şu kodu bir defalık çalıştırmanız yeterli:

```
DoCmd.OpenForm "Kisiler_Detay", , , , acHidden
Set Forms("Kisiler_Detay").Recordset = Forms("Kisiler").Recordset
Forms("Kisiler_Detay").Visible = True
```

ad	soyad	sehir
ahmet	duman	istanbul
ayşe	kara	ankara
ali	beyaz	sivas
mehmet ali	yemişçi	adana
steve	soul	michigan
sibel		

Record: 14 of 6

Kişiyi ait bilgiler

sibel

ID: 6

Ad: sibel

Soyad:

Sehir:

Adres: yenice mah. beyaz

Dogum Tarihi:

Meslek:

Kayıt kümeleri birbirine bağlanmış iki formun ahenkli uyumu.

Her ne olursa olsun Recordset sınıflarının adresleri eşit olduğu sürece zaten açık olan kaydı bir daha aramaya gerek kalmıyor.

LDB DOSYASI İLE KULLANICILAR LİSTESİ

Access dosyalarının olduğu yerde aynı önadı taşıyan LDB uzantılı dosyayı hepiniz farketmişsinizdir. Birçoğumuz acaba Access dosyamıza başka kimler bağlı diye bulmak istemişsinizdir en az bir kere. Velhasıl, bizim aradığımız şey tam tamına LDB dosyasının kendisidir. Çalışma anında bunun kopyasını alarak içini okumayı deneyin ve bu dosyayı salt-okunur açmayan tüm bilgisayar adlarını ve kullanıcı adlarını orada görün.

TRUE -1 FALSE 0

Elinizde 10 adet boolean tipinde değişken var. Bunlardan kaçının TRUE olduğunu nasıl bulursunuz? Siz hiç düşünmeden bunların değerlerini toplamanızı öneriyorum.

Dim a as boolean, b as boolean f as boolean

msgbox ((a + b + c + d + e + f) * -1) & " adedi TRUE yapılmış."

Zira TRUE değerinin VBA'daki sayısal karşılığı -1, FALSE değerinin ise 0'dır. Bu haliyle TRUE olanları toplayıp -1 ile çarpmak yeterli oluyor. Eşit mi, değil mi sorularının çok olduğu kodlarda işinize yarayabilir.

Mesela 10 adet seçim kutusundan en az 3 tanesinin seçilmesi gerektiğini karşımızdaki kişiye şöyle anlatabiliriz:

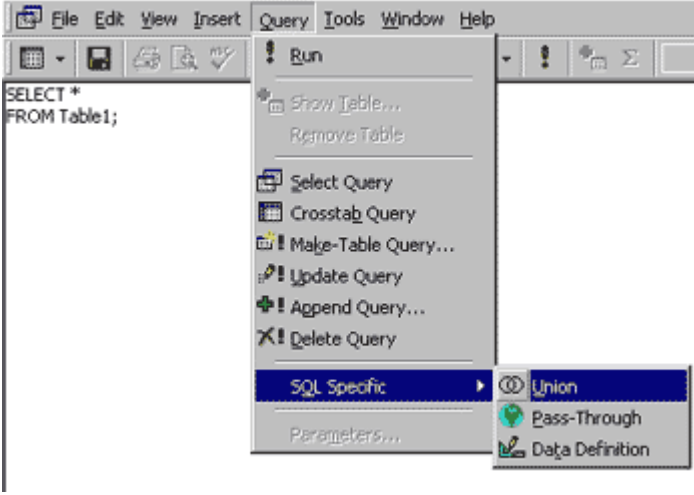
If (Check1 + Check2 + ... + Check10) > -3 Then _
MsgBox "En az üç seçenek işaretlemelisiniz."

Bir de şunu deneyin:

If (IsNull(Text1) + IsNull(Text2) ... + IsNull(Text10)) = -10 Then _
MsgBox "En az bir değer girmelisiniz."

HERKES İÇİN SORGULAR - UNION QUERY

Bu sorgu türü ile birbirinin aynı özelliklere sahip ya da birbirine benzer olan tabloları birleştirerek sorgu yapıp tek bir kayıt kümesi elde edebiliyorsunuz. Yeni bir sorgu oluşturup Sorgu/SQL Spesific menüsünün altındaki menüden en üsttekini seçin.



Sözelimi aynı alanları içeren Adresler1 ve Adresler2 adlı iki tablonuz var. Ama siz herhangi bir kişinin adresini bulmak için ikisinden aynı anda arama yapmak istiyorsunuz. Şu kodu yazın, olsun bitsin:

```
SELECT Ad, Soyad  
FROM  
(  
SELECT * FROM Adresler1  
UNION  
SELECT * FROM Adresler2  
) as Adresler  
WHERE Ad Like "*abc*"  
ORDER BY Ad, Soyad;
```

Sonrasında bu sorguyu tasarım moduna getirip üstünde çalışmak çok zevkli oluyor.

Windows 2000'den sonra sonra Türkçe karakterler konusunda olayların ters dönmeye başladığına şahit oluyorum. Artık yeni programlarda “i” ile “İ” ayrılıyor ve Türkçe’de nasılsa öyle oluyor. Yani “ID” adlı bir alanınız olduğunda “id” yazdığınızda büyük ihtimalle hata oluşacak, şu an benimkinden olduğu gibi.

BAŞKA BİR DOSYADAKİ TABLOYU SORGULAMAK

Ufak bir iş için bir tabloyu içeri bağlamayı içinize sindiremiyorsanız sizin için de ufakık bir çözümümüz var. Başka bir Access dosyasının içindeki bir tabloyu sorgu içinde kullanmak için şu iki örneği inceleyin:

```
SELECT Kisiler.*  
FROM [c:\belgelerim\db1.mdb].Kisiler;
```

```
SELECT Kisiler.*  
FROM Kisiler IN 'c:\belgelerim\db1.mdb';
```

Birincisinde sorgunun tasarım anında tablo özelliklerinde dosya ismi seçilebiliyor. Bunu yapınca tasarım görünümünde tablonun özelliklerinde Source (Kaynak) değerinin, yazdığımız dosyanın adı olduğunu göreceksiniz. Sorgu içinde tablo seçili iken View/Properties (Görünüm/Özellikler) menüsü ile özellikler sayfasını açabilirsiniz. İkinci örnek ise eskiden beri kullanılagelen bir “IN” kalıbı.

BAŞKA UYGULAMALARI YÖNETMEK

Söz başka dosyalardan açılmışken başka bir Access uygulamasının nasıl yönetildiğine bakınız şimdi.

```
Option Compare Database
Option Explicit
Dim a As Access.Application
Sub deneme()
Set a = Nothing
Set a = New Access.Application
a.OpenCurrentDatabase "c:\belgelerim\db1.mdb"
a.DoCmd.OpenForm "Form1"
a.DoCmd.Maximize
a.Visible = True
End Sub
```

Yeni bir Access uygulaması açıp onda bir dosya açarak kendi dosyamızda yapabileceğimiz herşeyi o uygulamada da yapabiliriz.

İpuçlarına Devam

Bu ayki yazının büyük kısmı Visual Basic'te kod yazımıyla ilgili. Bu yazıda, kod yazımıyla ilgili bazı sorularınıza cevap bulabilir ve yordamların nasıl kullanılacağına dair bazı temel bilgilere ulaşabilirsiniz. İlginç ipuçlarına da yer vermeyi ihmal etmedik yine.

YORDAMLAR VE GERÇEKLER

Yordamlar “Function” ve “Sub” türünde olabilir ve herbirinin mutlaka adı vardır. Önce şu örnek yordam başlıklarına göz atın:

- Sub Sub1()
- Function işlev1() As Date
- Function işlev1()
- Function işlev1(Parametre) as Integer
- Sub Sub1(parametre1 As String, parametre2 As Date)
- Public Function F(Ad As String, ParamArray Dizi()) As Long
- Private Static Function StatikBirİşlev() As Boolean
- Sub SubYeni(Ad As String, Optional Tarih As Date=36861)
- Private Sub Command1_Click()

“Sub” yordamları mutlaka “End Sub” satırı ile, “Function” yordamları da “End Function” satırı ile biter.

```
Private Sub MetinKutusu1_Click()
```

```
...
```

```
End Sub
```

ve

```
Function Degistir() as Boolean
```

```
...
```

```
End Function
```

gibi.

Örneklerde “Function” ve “Sub” deyimlerinden önce bazı tanımlamalar var. Bunlar yordamlara bazı ek özellikler katarlar. Mesela başında “Private” deyimi olan bir yordam, aktif olan modül sayfasının dışında bir yerde kullanılamaz. Ya da, başında Static olan bir yordamın içindeki tüm değişkenler statik olarak tanımlanmış olur.

Function yordamlarının asıl amaçlarından biri değer üretmek olduğu için başlık satırının sonunda mutlaka “AS” deyimi ile bir tip belirtilir. Eğer tipi belirtilmemiş bir işlev varsa bunun döndüreceği değer Variant (herhangi bir değişken tipinde olabilen) türünde olur.

Bir de parametresiz işlevler vardır ki, tipi belirtilmiş olsa bile, parametresiz bir işlevi isterseniz “Sub” yordamlar gibi kullanabileceğinizi geçen aylarda da söylemiştik netekim.

Örneğin,

işlev1() as String

şeklinde tanımlı bir işlevi;

Call işlev1

ya da doğrudan

işlev1

şeklinde çalıştırmak mümkündür.

Herhangi bir yerden bulduğunuz bir kod parçasını programınıza birkaç şekilde ekleme imkanınız var. Şu kod parçası aktif dosyamızın bulunduğu yerdeki tüm dosyaları listeleyip mesaj kutusunda gösteriyor:

```
Dim Dosyalar As String, Dosya As String, Klasor as String
Klasor = CurrentProject.Path
If Right(Klasor,1)<>"\" Then Klasor= Klasor & "\"
Dosya = Dir(Klasor)
Dosyalar = "Bu klasördeki dosyalar:" & vbCrLf
Dim i As Integer
i = 0
Do While Dosya <> ""
If Dosya <> "." And Dosya <> ".." Then
i = i + 1
Dosyalar = Dosyalar & vbCrLf & i & ". " & Dosya
End If
Dosya = Dir
Loop
MsgBox Dosyalar
```

Bunu, var olan bir yordamın içine ekleyebilir ya da kendi oluşturacağınız yeni bir yordamın içine koyabilirsiniz. Mesela bir düğmenin “Click” (Tıklatıldığında) yordamı kısmına. Eğer elinize geçen bir kod parçasında “Sub” ya da “Function” şeklinde yordam başlıkları varsa bunu kendi yordamınıza eklemek için başlıklarını ve son satırlardaki “End..” kısımlarını çıkarmalısınız. Veyahut, yordamı aynen kullanmalısınız.

DURUM ÇUBUĞUNDA MESAJ GÖSTERMEK

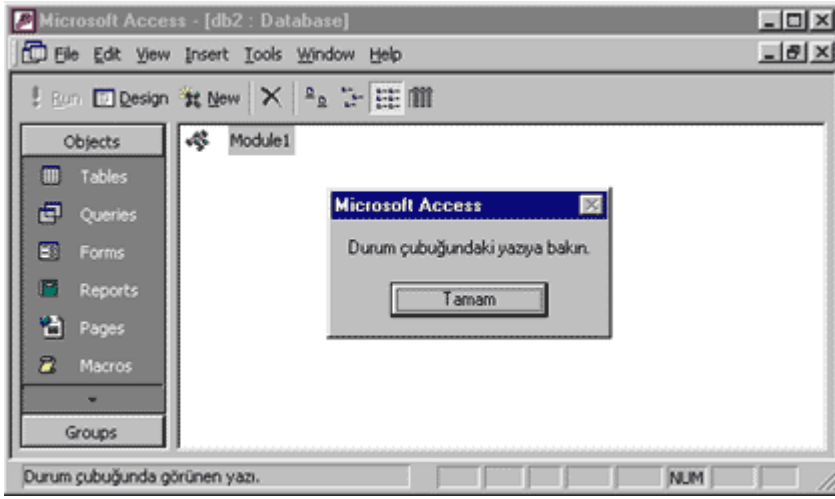
Uygulamalarımnda diğer profesyonel programlar gibi durum çubuğunu kullanmak ne iyi olurdu diyorsanız şu kısacık kodu deneyin:

Application.Echo True, "Durum çubuğunda görünen yazı."

MsgBox "Durum çubuğundaki yazıya bakın."

Application.Echo True

Bunu çalıştırınca Access’in durum çubuğunda (Status Bar) bir mesaj görüntülenecek. Kullanıcının bekletileceği yerlerde “İşlem devam ediyor...” gibi bir mesajı gösterebilirsiniz bu şekilde. İlk satırdaki “True” yerine “False” yazarsanız, Access penceresi silik görünür. Böyle bir durumda, işlem sonunda Echo’yu True ile çalıştırmayı ihmal etmeyin, yoksa işlem yapılabilir de, Access ekranında pasif bir pencere görüntüsünden başka bir görüntü olmayacaktır.



DoCmd nesnesinin Echo adlı metodu da aynı işi yapabiliyor. Fakat Application nesnesininkini kullanmamız öneriliyor.

DAHA KOLAY KAYIT SİLME

Geçen aylarda bu sayfalardan, kayıt silmek için ‘daha güzel kayıt silme’ başlığında bir kod örneği verdik. Gelgelelim aşağıdaki yeni işlevle daha kolay ve kısa bir şekilde kayıt silme işlemi yapabilirsiniz.

```
On Error Resume Next
If Me.NewRecord Then
Me.Undo
ElseIf MsgBox("Kayıt silinsin mi?", vbYesNo) = vbYes Then
Me.Recordset.Delete
End If
```

Bu kodu, kayıt silme görevi üstlenecek bir düğmenin “Click” (Tıklatıldığında) yordamındaki “Sub..” ile “End Sub” satırları arasında kalan kısma yazın.

ON ERROR GOTO 0

Hata gidermek için birçok yerde “ON ERROR RESUME NEXT” deyimi kullanılıyor. Hata giderme pozisyonundan ayrılmak gerektiği durumlarda “ON ERROR GOTO 0” (sıfır ile bitiyor) deyimini kullanmak gerekiyor

```
On Error Resume Next
'Hataları yoksayıp sonraki komuta geç.
MsgBox 3 / 0
MsgBox "Sıfıra bölme hatasını geçti."
On Error GoTo 0
'Hataları yoksayma.
MsgBox 3 / 0
'Simdi sıfıra bölme hatası oluşunca uyardı.
```

AH ŞU KISAYOL TUŞLARI!

Sonraki Kayıt Ctrl + Page Down
Önceki Kayıt Ctrl + Page Up
İlk Kayıt Ctrl + (Yukarı Ok)
Son Kayıt Ctrl + (Aşağı Ok)
Yeni Kayıt Ctrl + +
Kayıt Sil Ctrl + -
Önceki Kaydı Çoğalt Ctrl + '

Bu bilgilere, yardım bölümünde “keyboard shortcuts” başlığını aratarak da ulaşabilirsiniz.

FORM AÇIK MI?

Bir formun ya da raporun açık olup olmadığını bulmak için aşağıdaki kodları kullanabilirsiniz.

```
If SysCmd(acSysCmdGetObjectState, acForm, "Form1") <> 0 Then
MsgBox "Form açık."
Else
MsgBox "Form kapalı."
End If
```

Örnek, Form1 adlı formun açık olup olmadığını araştırıyor. Raporlar için “acForm” deyimini “acReport” olarak değiştirin.

DURUM ÇUBUĞUNDA İLERLEME ÇUBUĞU GÖSTERMEK

Access kendi işlemlerini yaparken bazan durum çubuğunda ilerleyen mavi çizgi beliriyor. Bundan, işlemin ne kadarının bittiğini anlayabiliyoruz. Tabii, bunun aynısını bizim kullanabilmemize de imkan var. Tablo1’deki kodu çalıştırırsanız resimdeki gibi mavi ilerleme çubuğu çıkacak.

TABLO1:

```
Sub Ilerleme_Cubugu()

Dim T As Long
T = Timer

SysCmd acSysCmdInitMeter, "İşlem devam ediyor. Lütfen bekleyin...", 10
'İlerleme çubuğunu başlat. 10 sayısı ile değer sınırı veriliyor.

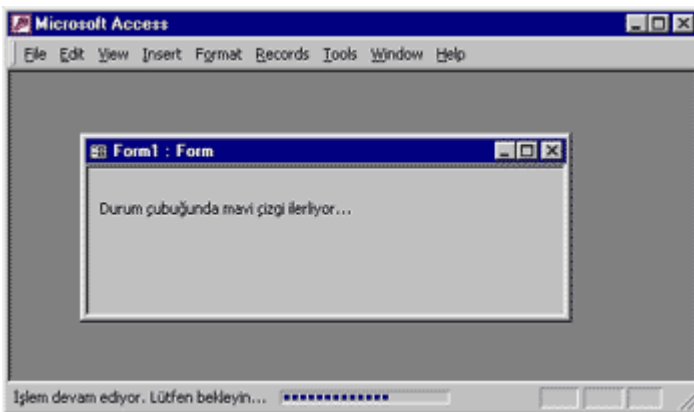
Do
If Timer - T >= 10 Then Exit Do
SysCmd acSysCmdUpdateMeter, Timer - T
'10 saniye boyunca her saniye için çubuğu 1 nokta ileri al.
DoEvents
'Ekranı dondurma
Loop

MsgBox "İşlem tamamlandı."

SysCmd acSysCmdClearStatus
'İlerleme çubuğunu kaldır.

End Sub
```

acSysCmdInitMeter ile ilerleme başlatılıyor ve çubuğu ilerletmek için acSysCmdUpdateMeter kullanılıyor. İşlem bitiminde acSysCmdClearStatus ile çubuğu kaldırmalısınız.



Mavi çizgi ilerliyor...

PENCERELERİ BOYUTLANDIRMAK

Access Visual Basic'teki gibi formları kolayca boyutlandırıp taşımamıza izin vermiyor. Bunun için MoveSize komutunu kullanmak gerekiyor. MoveSize ile bir formu belli bir noktaya taşıyabilir ya da taşırken boyutlarını yeniden ayarlayabilirsiniz.

Şu komut aktif pencereyi sol üst köşeye taşır:

DoCmd.MoveSize 0, 0

Bu ise sol üste taşırken genişliğini 4000, yüksekliğini 2000 olarak ayarlar:

DoCmd.MoveSize 0, 0, 4000, 2000

Tüm parametreleri belirtmek zorunda değilsiniz:

DoCmd.MoveSize , 1000, , 1500

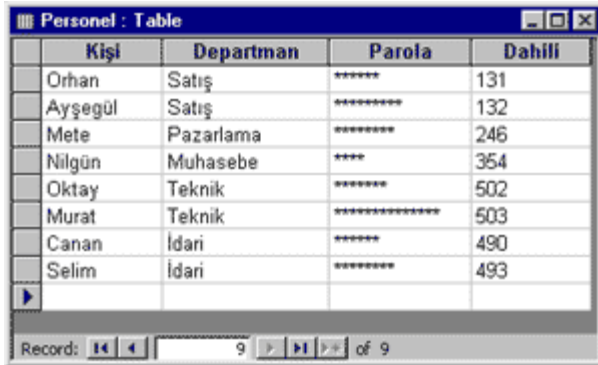
Bu kod sol ve genişlik değerlerini değiştirmeden, üst tarafı 1000 ve genişliği 1500 olarak ayarlayacak.

Parametreler aşağıdaki şekilde kullanılıyor:

DoCmd.MoveSize Sol, Üst, Genişlik, Yükseklik

PAROLA KAYITLARINI GİZLİ GÖSTERMEK

Access'teki form ve raporlardaki metin kutularında şifreli metin kullanabilmek için InputMask değerini "Password" yapmak yeterli oluyordu. Eğer kayıt tablosunda da parola bilgilerini (*) işaretli olarak görmek istiyorsanız tahmin ettiğiniz gibi veri alanının InputMask (Giriş Maskesi) değerini aynı şekilde ayarlamalısınız.



Kişi	Departman	Parola	Dahili
Orhan	Satış	*****	131
Ayşegül	Satış	*****	132
Mete	Pazarlama	*****	246
Nilgün	Muhasebe	****	354
Oktay	Teknik	*****	502
Murat	Teknik	*****	503
Canan	İdari	*****	490
Selim	İdari	*****	493

Record: 9 of 9

Tabloyu tasarım görünümünde açıp parola alanının InputMask değerine "Password" yazın.

Field Name	Data Type	Description
Kisi	Text	
Departman	Text	
Parola	Text	
Dahili	Text	

Field Properties	
General	Lookup
Field Size	50
Format	Text
Input Mask	Password
Caption	Parola
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes

A pattern for all data to be entered in this field

Eski Defterler

Yeni yıla girildiğinde eski defterler karıştırılır ve önceki yılın muhasebesi yapılır, adettir. Biz de eski defterleri şöyle bir yoklayalım dedik. Yeni şeyleri de ihmal etmedik tabii.

Bu dizinin ilk yazısının yazıldığı zamanlarda Türkçe Access97'deki problemlerin yamalarıyla meşgulduk. Yine o aylarda Access'in meşhur "Otomasyon Hatası" piyasada fınk atıyordu. Derken "Otomasyon Hatası"nın bize aslında "Otomasyon Nesnesi"ni ima etmeye çalıştığını farkettilik. Bu da Sınıf Modülleri (Class), Başvuru Kütüphaneleri (Reference Library) ve Add-in'ler gibi yapılara yöneltti bizi.

Yıllardır değişmeyen bir olgu, Türkiye'de kullanılan tarih biçiminin Windows standardıyla aynı olmamasından kaynaklanan SQL kodu yetmezliği. Gerek ASP veya VB kullanıp ADO ile SQL Server'dan ya da başka bir yerden veri almaya çalışırken, gerek Access'in kendi içinde, gerekse Delphi gibi programlarda Paradox ile çalışırken, SQL kodları içinde kullanılan tarih değerlerinin standart verilememesi, epeydir gördüğüm üzere birçok programcının başını ağrıtmıştır.

Bunun çok önemli olduğunu düşündüğüm için tarih değerlerinin aslında sayı olduklarına değinmiş ve SQL kodu içinde tarih değerinin sayısal karşılığını yazmanın garantili bir çözüm olduğundan bahsetmiştik. Şu an da bundan bahsediyor olmamın sebebi aynı problemin hala birçok yerde yaşandığını görmemdir.

Projelerimizi dağıtma zamanı geldiğinde Access'in kendisine gerek kalmadan Access uygulamalarını çalıştırabilmek için Microsoft'un çıkardığı ODE Tools adlı program ile tanışmıştık. Access dosyası EXE olur mu sorusu sıkça sorulmaya başlandığında, bu programın EXE yapmadığını ama Access dosyalarını EXE gibi çalıştırabilme yeteneğinin olduğunu ve de Access dosyalarının derlenip kodlarının tamamen gizlenebileceğini görmüştük.

SQL kodu yazmak, Access'te programlama sürecinin ayrılmaz parçalarından olduğu için bunu da yeteri kadar önemseyip ilginç ve işe yarar ipuçlarını inceledik.

Otomatik Sayı (Autonumber) alanları da birçoğumuzun hoşuna gitmeyen tavırlar gösteriyordu. Kayıtlar silindiğinde neden eski numaralar kullanılamaz oluyordu?!! Bunun, otomatik sayı alanlarının en temel özelliği olduğu açık. Veri alanının silinip yeniden oluşturulması öneriliyor çok zaman. Fakat bu her zaman yapılamayacak bir şeydi. Yine de Access dosyasının içeriğini kontrol edip belli bir noktasını bulmak bunu sağlıyor. Aşağıda Access dosyalarında otomatik sayı değerinin istenildiği gibi değiştirilebileceğine dair bir örnek bulacaksınız.

Birçok şeyin aslında ayrıntılarda olduğunu görüp belli noktaları yakaladıktan sonra işimize yarayacak şeyler bulmuştuk. Tablo görünümünde başka tablodaki kayıtlardan seçim yaptırabilmek bunlardan biriydi. Yine, tablo kullanmadan

elimizdeki değerleri sorgu (query) olarak kullanmak, birleşik sorgular, tablo bağlamaksızın (link) başka bir Access dosyasındaki tabloyu sorguya dahil etmek, tablo görünümünde süzme yetenekleri, geçerlilik kısıtlamaları (validation rules), sistem yazıtlarını resim olarak kullanmak gibi ayrıntılara rasladık.

Birçok okurumuzun istediği halde, Access'te kod yazmaya geçişten bahsetmemek olmazdı. En azından bir yerden bulunan kod parçasının bir Access programına nasıl ekleneceğinin bilinmesi şarttı.

OTOMATİK SAYI DEĞİŞİR AMA..

Otomatik sayı değerinin değiştirilmesini istemenin sebebi kayıtlara tanım numarası vermekten ziyade görünüşü güzel göstermek olsa gerek.

Aşağıdaki yöntemi sadece otomatik sayıların nasıl işlendiği konusunda bir fikir vermesi açısından dikkate alın. Kullanmanızı tavsiye etmiyorum. Zaten tabloda kayıtlar varken yeni alınacak otomatik sayı değerini sıfırlamak ya da değiştirmek anlam taşımaz.

Küçük bir dosyada otomatik sayı alanı içeren bir tablo oluşturuldu. Bu tabloya 15 kayıt eklendi, bazı kayıtlar silindi. Bu durumdayken dosyanın bir yedeği alındı. Sonra tablo açılıp yeni kayıt satırına gelinerek birşeyler yazıldı ve ESC tuşu ile kayıttan vazgeçildi. Böylelikle dosyanın bir yerine en son kullanılan otomatik sayı değeri yazılmış oluyor. Dosya kapatıldı ve küçük bir kod yazıp iki dosya arasında farklı olan kesimleri araştırdık. Sonuçta bu iki dosya arasında sadece 2 baytlık kesimin farklı olduğunu gördük.

Bunlardan birinin değerini LONG tipinde (4 bayt) okuduğumuzda en son seçtiğimiz otomatik sayı değerini elde ettik. Geriye bu değeri değiştirmek kaldı. Oraya ne yazılırsa otomatik sayı yeni değerini bu numaradan başlatıyordu. Burayı 0 olarak değiştirdikten sonra resimdeki gibi bir görüntü normaldi artık.

ID	ad	departman
2	tekin	teknik
3	idil	idari
5	pelin	pazarlama
6	inci	insan kaynakları
9	serhat	servis
12	bilgen	bilgi işlem
15	murat	muhasebe
1	deneme	

Otomatik sayı değeri artık 1'den başlıyor.

Eğer otomatik sayı alanı birincil anahtar (primary key) olarak belirlenmişse, sonraki kayıt girişlerinde çoğaltılmaması gereken dizin değeri hatası (duplicate unique index) ile karşılaşmak kaçınılmaz olur.

Siz en iyisi bu değeri arkaplan işlemleri için kullanın ya da kullanıcıya herhangi bir kod değeriymiş gibi gösterin. Ben olsam ekran başındaki kişiyi resimde olduğu gibi kandırırdım.

Kod	Ad
Kod 1	Ali
Kod 2	Veli
Kod 3	Kırkdokuz
Kod 4	Eli

Nasıl mı oldu diyorsanız, tablonun tasarım görünümünde otomatik sayı alanının Format (Biçim) özelliğinin değerini tam

olarak

“Kod “#

şeklinde belirtmeniz yeterli.

DEĞİŞİKLİKLER KAYDEDİLSİN Mİ

Diğer birçok programda olduğu gibi bilgilerinizin kaydedilmesinden önce uyarı mesajı göstermek istiyorsanız alttaki kodu deneyebilirsiniz. Herhangi bir formun güncelleştirme öncesi (BeforeUpdate) ile ilgili yordamının aşağıdaki gibi olmasını sağlayın. Artık eski kayıtlar üzerinde değişiklik yapıldığında soru sorulacak ve kullanıcı eğer hata yaptığını düşünüyorsa kaydı bu şekilde geri alabilecek. Formun BeforeUpdate yordamını oluşturun ve aynen şu şekilde olmasını sağlayın:

```
Private Sub Form_BeforeUpdate(Cancel As Integer)
If NewRecord = False Then
If MsgBox("Değişiklikler kaydedilsin mi?", _
vbYesNo) = vbNo Then
Cancel = True
DoCmd.RunCommand acCmdUndo
End If
End If
End Sub
```

FORMU TASARIM GÖRÜNÜMÜNDE AÇ

Formu, raporu, tabloyu, sorguyu ve diğer nesneleri tasarım görünümünde açmak için CTRL tuşunu basılı tutarak açmayı deneyin. CTRL+ENTER yapın ya da CTRL basılı iken fareyle çift tıklayın.

SQL SERVER TABLOSUNDA YENİ GİRİLEN KAYITLARA ULAŞMAK

ADO kullanırken VBA ya da ASP kodlarında SQL Server veritabanındaki bir tabloya yeni kayıt eklendiğinde bu kayda ulaşmak sorun olabiliyor. Aslında sorun birincil anahtar dizini (primary key) kullanılmamış olmasından kaynaklanıyor.

```
Tablo1.Addnew
Tablo1("Alan1")="Değer"
...
Tablo1.Update
'MsgBox Tablo1("Alan1") 'Access için.
Response.Write Tablo1("Alan1") 'ASP için.
```

Eğer tablonuzda herhangi bir birincil anahtar dizini yoksa bu kod son satırda hata verecektir.

HERKES İÇİN SORGULAR

“IN” DEYİMİ İLE ÇOKLU ARAMA KRİTERİ BELİRLEMEK

Sorgularda aynı alan için birçok arama kriteri “OR” deyimi ile tanımlanabiliyor. Tabii bununla bazan kod yeterince uzayabiliyor ve karmaşık bir hal alıyor. Onun yerine “IN” komutunu kullanmak daha kolay.

Kişiler tablosundaki kayıtlardan Meslek alanının değerinin ‘Öğretmen’, ‘Bankacı’, ‘Doktor’, ‘Muhasebeci’ olduğu kayıtları bulmak için OR komutunu şöyle kullanabilirsiniz:

```
SELECT * FROM Kisiler WHERE ( (Meslek='Öğretmen') OR (Meslek='Bankacı') OR (Meslek='Doktor') OR (Meslek= 'Muhasebeci') )
```

Aynı kod IN deyimi ile bakın nasıl oluyor:

```
SELECT * FROM Kisiler WHERE Meslek IN ('Öğretmen', 'Bankacı', 'Doktor', 'Muhasebeci')
```

MESAJ KUTUSUNDA YENİ SATIR GÖSTERMEK

Kullanıcıya uzun uzadıya olan mesajları daha anlaşılır bir şekilde göstermek istiyorsanız mesaj kutularında yeni satırlar kullanmayı deneyebilirsiniz.

**MsgBox "Bu mesaj" & vbCr & "birkaç " & vbCr & _
"satırdan" & vbCr & "oluşuyor.", vbOKOnly + vbExclamation**

VbCr deyimini satır sonu karakteridir. Yani Chr(13) ile aynı.



GÖRÜLEN KAYDI RAPORDA AÇ

Form üzerinde birçok kayıt olduğu halde sadece görünen kayda ilişkin bir rapor nasıl yazdırılır? Bunun için raporu açarken kullandığınız koda bir parametre eklemeniz yeterli.

DoCmd.OpenReport "Rapor1", asViewPreview, , "SıraNo=3"

gibi. Eğer bir metin alanına göre koşul belirlemek istiyorsanız değerin başında ve sonunda ' işareti olmalı.

DoCmd.OpenReport "Rapor1", asViewPreview, , "PersonelNo='Ahmet0001'"

Sondaki parametre aslında bir arama koşulu. Oraya istediğiniz kadar koşul içeren bir metin yazmanız mümkün.

DoCmd.OpenReport "Rapor1", asViewPreview, , "(SıraNo>100) AND (PersonelAdi>'K') AND (Yas<25)"

En son parametrede verilen süzme kriteri formları açarken de aynı şekilde kullanılabilir.

ACCESS PENCERESİNİ BOYUTLANDIRMAK

Özenle hazırladığımız form ve raporları tam ekran yapar, kullanıcının önüne koyarız. Fakat programın kullanıldığı bilgisayarda Access penceresi tam ekran ya da bizim istediğimiz boyutta açılmıyor her zaman. Lakin bunun da çözümü yok değil. Access penceresinde modül sayfasına gelip yeni bir modül oluşturun ve içinin tam olarak aşağıdaki gibi olmasını sağlayın. Sonra bu modülü "ModulYerlester" adıyla kaydedin.

```
Option Compare Database
Option Explicit

Public Type RECT
Left As Long
Top As Long
Right As Long
Bottom As Long
End Type

Public Const SW_RESTORE = 9
Public Declare Function ShowWindow Lib "user32" (ByVal hwnd As Long, ByVal nCmdShow As Long) As Long
Public Declare Function MoveWindow Lib "user32" (ByVal hwnd As Long, ByVal x As
```



```
Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal bRepaint As Long) As Long
```

```
Public Const SPI_GETWORKAREA = 48  
Public Declare Function SystemParametersInfo Lib "user32" Alias  
"SystemParametersInfoA" (ByVal uAction As Long, ByVal uParam As Long, ByRef  
lpvParam As Any, ByVal fuWinIni As Long) As Long
```

```
Public Sub MasaustuneYerlestir()  
'Kodu şimdi çalıştırmak için buraya gelip F5 tuşuna basın.  
Dim r As RECT  
SystemParametersInfo SPI_GETWORKAREA, 0, r, 0  
Call Yerlestir(r.Left, r.Top, r.Right - r.Left, r.Bottom - r.Top)  
End Sub
```

```
Public Sub Yerlestir(Sol As Long, Ust As Long, Sag As Long, Alt As Long)  
ShowWindow hWndAccessApp, SW_RESTORE  
MoveWindow hWndAccessApp, Sol, Ust, Sag, Alt, 1  
End Sub
```

Artık programınızın herhangi bir yerinde

Call MasaustuneYerlestir

ya da

Call Yerlestir(100, 100, 200, 200)

gibi komutları kolaylıkla kullanabileceksiniz.

ACCESS HİKAYELERİ

ESKİ DEFTERLER YENİ HİKAYELER

Access2000 projelerinin dağıtılması için Office 2000 Developer Tools v1.0'ı aldık. Fakat aldığımız gün MSDN'de bu paketin v1.5 sürümün çıktığını öğrendik. Elde bu paket varken MSDN'e bakma işi nereden çıkmıştı? Tabii hala normal bir kurulum hazırlayamadığımız için. Öyle ki, kurulum hazırlamaya çalışırken Access Run-Time için 500MB'lık bir klasör gereksiz yere kurulum paketine dahil ediliyor. Tabii bu, sadece Access Run-Time paketini eklerken oluyor. Bu paket eklenmezse proje dosyasıyla beraber birkaç dosya paketleniyor sadece. Boyu da 3 MB. civarında. Gerçi klasörün içinden gereksiz gördüğüm bazı dosyaları sildiğimde sorun çıkarmadı ama buna güvenmemek daha iyi.

Böyle durumlarda MSDN'nin sitesinden yardım almak gerekli. Orada anlatıldığına göre normalde En az 150 mb.'lik bir paket oluşturuluyormuş. Tabii hazırladığımız pakete Access Run-Time eklemek istediğimizde IE5'in kurulumunu da eşantiyon olarak ekliyor. Sonradan olacak ki bunun abartı olduğunun farkına varmışlar ve IE5'i pakete dahil etmeyen bir yama çıkarmışlar. Yama demek biraz zor gerçi. 10 mb. civarında bir büyüklüğü var ilk yükseltme programının. v1.5 sürümü ise 30 mb. civarında.

Bu olaydan da öğrendim ki, yeni bir ürün aldığımızda öncelikle Internet'ten o ürünle ilgili yeni güncelleştirmelerin olup olmadığını araştırmak gerekiyor.

* * *

Access kullanan bazı makinelerin SQL Server'ın olduğu bilgisayara isim ile ulaşmaları bazan sorun oluyor. Bu yüzden HOSTS dosyasında bir IP adresi ile bir bilgisayar adı tanımlamak gerekiyor. Böylelikle orada yazılı olan ad bizi her zaman yanındaki IP adresine yönlendiriyor.

Bu aşamadan sonra aynı programın her bir makineye kurulup ODBC ayarlarının yapılması gerekiyor. Zira Access içinden SQL'e ODBC ile bağlanılıyor. Tablolar kısmına SQL Server veritabanındaki tablolar yerleştirildi. Böylelikle aynı ayarın,

programın kurulacağı tüm makinelere yapılması şart oluyor. Bunu da kod ile yapmak çok masraflı değil.

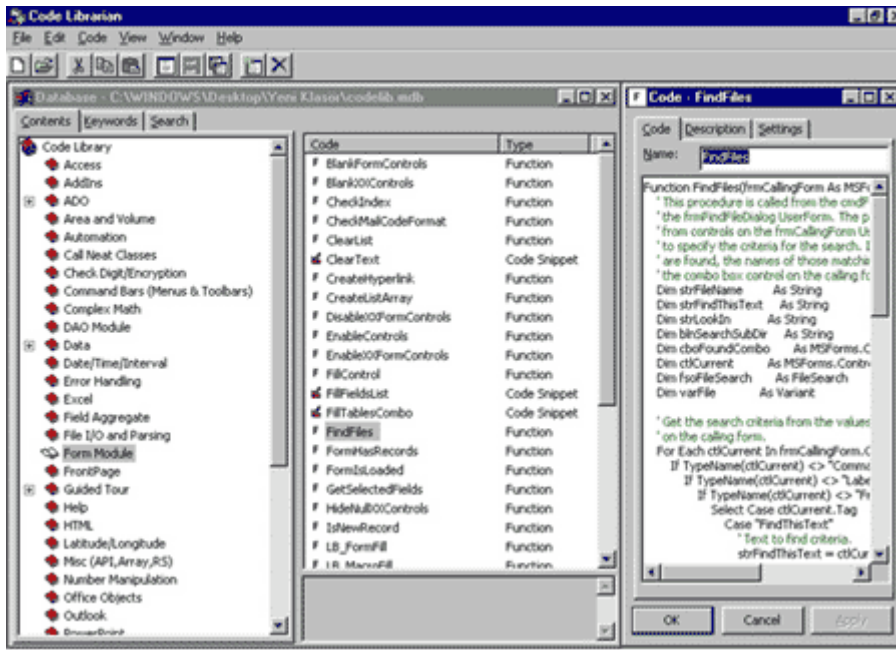
Yeni bir ODBC DSN bileşenini kod ile tanımlamak için Registry'deki HKLM\Software\ODBC altında bazı değişiklikler ve ODBC.INI dosyasına eklemeler yapmak yeterli.

HOSTS dosyasını da otomatik ayarlamak için önce dosyanın yerini bulmak lazım. Eğer sistem Win9x ise %WINDIR% klasörü altında HOSTS.SAM dosyası, eğer NT ya da Win2000 ise %SYSDIR%\DRIVER\ETC klasöründeki HOSTS dosyası değiştirilmeli.

Artık DSN ayarlarının da kurulum programları tarafından otomatik olarak yapılması fikri kulağa hoş geliyor aslında.

* * *

ODE Tools'un içinden çıkan bir başka eşantiyon da Code Librarian adlı yardımcı bir program. Bu program, kod yazarken büyük projeler arasında bir o yana bir bu yana koşuşturan programcılar için bir kod kütüphanesi oluşturmuş. Herhangi bir konuyla ilgili kod ihtiyacınız olduğunda gerekli başlığın altından aradığınız kodu kolayca bulabiliyorsunuz. Üzerine istediğiniz eklemeleri yapabileceğiniz gibi, var olan kodları da değiştirmeniz mümkün.



Eğer kodlarınız için kütüphane oluşturmayı düşünüyorsanız bu program birebir. Bu kadar hazır kod el altındayken insana kopya çekme fırsatı da doğmuyor değil açıkçası.

Tabii ki, PC Magazine okuyucuları ilk kez yayınlanan orjinal ipuçlarını bu sayfalarda görmeye devam edecekler.

DOSYA:

[AccessOcak2001.zip](#)

(Otomatik sayıların değiştirilmesi ile ilgili sonradan eklenen bir yazıya [buradan](#) ulaşabilirsiniz.)

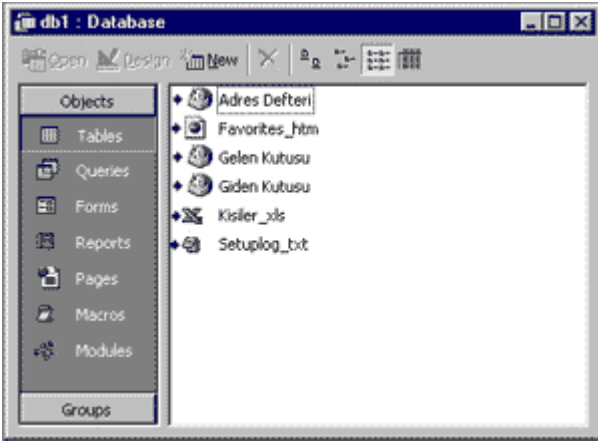
Bağlı Tablolar

Access başka veri kaynaklarından da kendisine tablo bağlanmasına izin veriyor. Altından kalkamadığı durumlarda ise yardımına ODBC koşuyor.

Tablo bağlama işleminin Access'in İngilizce sürümündeki adı Link. Farklı türdeki birçok veri tablosunu Access dosyalarının içine bağlamak mümkün. Örneğin Excel'de hazırlanmış bir tabloyu Access'in içinden rahatlıkla kullanabilirsiniz. Sabit sütunları olan bir Excel tablosunu Access'e bağlamak için Access'teki Dosya / Dış Veri Al / Bağlı Tablo (File / Get External Data / Link Tables) komutunu çalıştırıp bir Excel dosyası seçmek yeterli. Sonrasındaki soruları da geçerek Excel tablosunu kolayca bağlamış oluyorsunuz.

Bu özellik öncelikle Access tablolarını başka Access tabloları içine bağlamak için var. Access 2000 dosyası içine Access 97 sürümündeki bir dosyanın tablosu bağlanabiliyor. Böylece tabloyu uzaklarda aramadan doğrudan kullanabiliyorsunuz.

Bazan, ayrıntılara dikkat etmenin ne kadar önemli olduğundan bahsediyorum. Eğer tablo bağlama penceresindeki dosya türleri (file of types) listesine dikkat ederseniz orada metin dosyalarından tutun Exchange tablolarına kadar birçok seçenek göreceksiniz.



Bağlı tabloları herhangi bir Access tablosuymuş gibi kullanabilirsiniz.

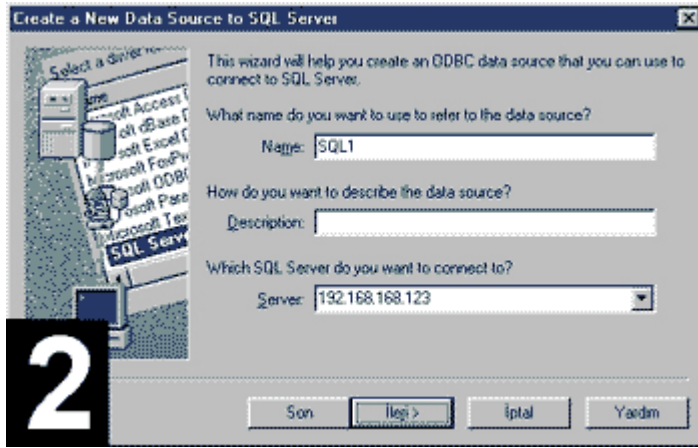
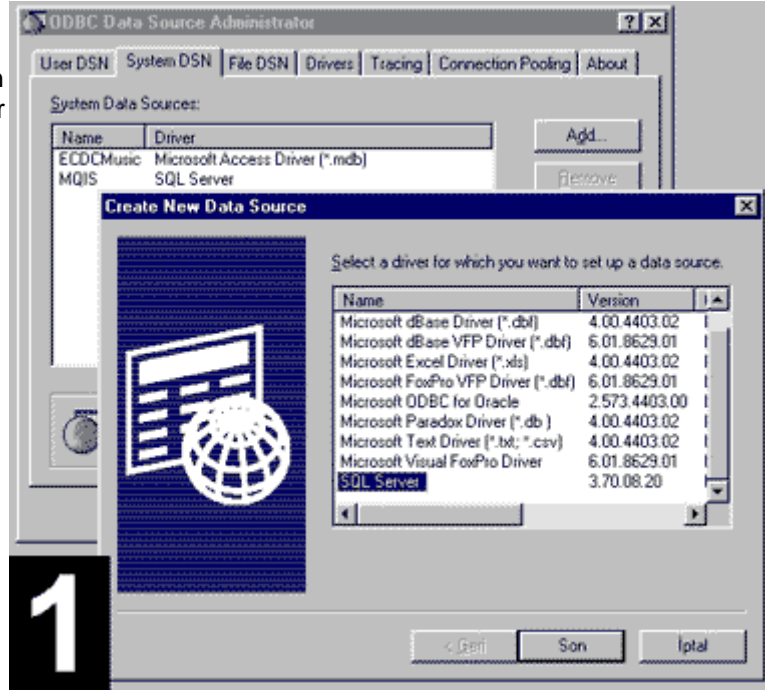
Outlook Express'teki adres defterimi başka bir yerden kullanabilmek hiç bu kadar kolay olmamıştı. Tablo bağlama penceresinde dosya türü olarak Exchange'i seçtikten sonra açılan pencereden Adres Defteri'ni seçip Access dosyasının içine bağladım. Aynı şey e-posta iletileri için de geçerli.

Kayıtlar içeren metin dosyaları için de hemen hemen aynı şeyler geçerli. Fakat bu dosyalar veritabanı yapısında olmadığı için tablo bağlarken bazı ayarlamalar gerekiyor. Virgülle, tab işaretiyle ayrılmış veya sabit uzunluklu alanlar içeren bir metin dosyasını, tablo bağlama penceresinden seçin. Açılan formlarla birlikte gerekli ayarlamaları kolayca yapabilirsiniz. Nihayetinde metin dosyasını (sadece okunur) bir tablo olarak kullanabilirsiniz.

SQL SERVER TABLOSUNU ACCESS'E BAĞLAMAK

Gelelim SQL Server tablolarının nasıl bağlandığına. Bunun için önce Denetim Masası'ndaki ODBC bölümünde bir ayar yapmak gerekli.

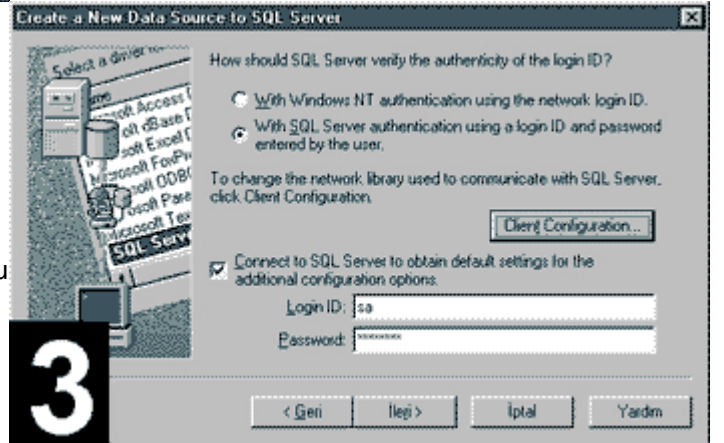
ODBC seçeneklerinde yeni bir System DSN (System Data Source) bileşeni oluşturup, bunu var olan SQL Server sunucunuzla ilişkilendirin. System DSN oluşturma ekranındaki ağ bağlantı seçeneklerinden TCP/IP'yi (Client Configuration – Network Libraries) seçmeyi ihmal etmeyin.

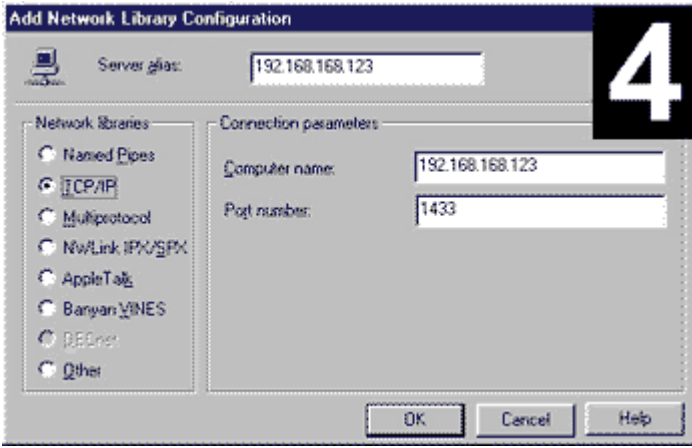


Yeni ODBC DSN'sini oluşturma işlemi tamamlandıktan sonra Access'teki tablo bağlama penceresinde dosya türleri listesinden ODBC Data Sources seçeneğini seçin. Yeni açılan pencerenin Machine Data Source sayfasında, az önce oluşturduğunuz yeni DSN'yi işaretleyin. Sonrasında çıkan seçenekleri de işaretleyip SQL Server tablonuzu kayıt ekleme, silme ya da değiştirme için rahatlıkla kullanabilirsiniz.

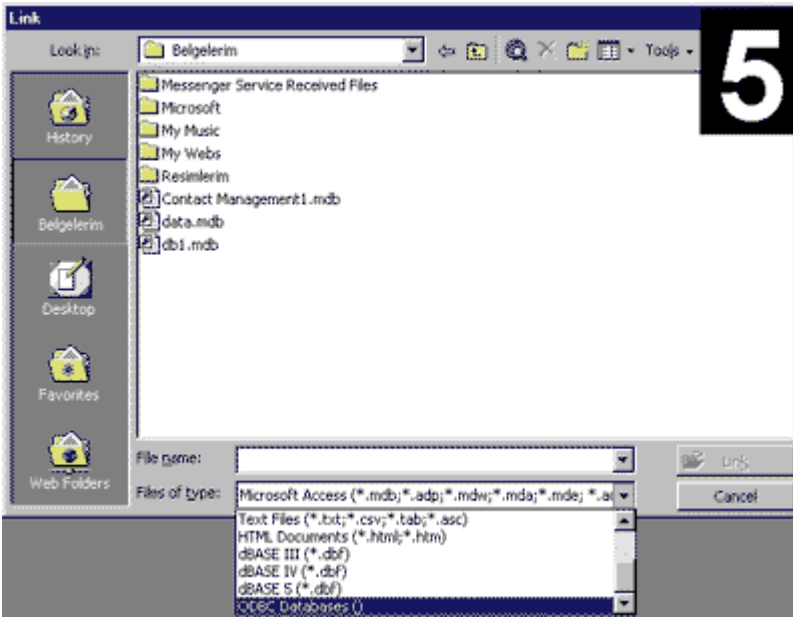
Sadece SQL Server değil, ODBC sürücüsü olan Oracle, Paradox, FoxPro, dBase ve diğer tüm veritabanı sistemlerindeki tabloları Access'in içinden ya da başka yerlerden kolaylıkla kullanmak mümkün.

Tablo bağlama işlemi yapmayacaksanız veya buna zorunlu değilseniz ODBC bağlantısı ile çalışmak öncelikli tercih olmamalı. Zira doğrudan bağlantı kurabilen ADO ve sair sistemler ODBC'den daha hızlı çalışır.





ODBC (Open Database Connectivity) farklı veritabanı sistemlerinin her yerden aynı şekilde kullanılabilmesine imkan veren bir ara bağlantı modülüdür. Bununla Access içine bağladığınız bir Paradox tablosunu Access'e hissettirmeden ve başka bir ayar yapmadan bir anda Oracle içinden kullanılabilmek mümkündür. Zaten Access kullandığı bir bağlı tablonun hangi sistem üzerinde olduğunu umursamaz.



SQL Server tablolarını bağlamak için ODBC Databases'i seçin.

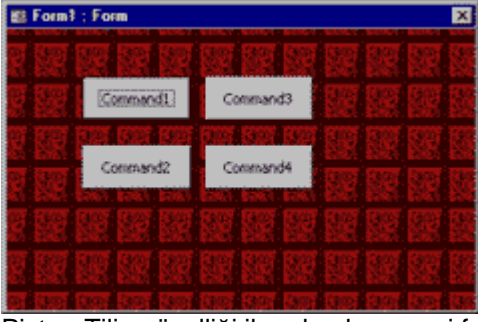
Bazı tabloların sadece okunur olarak görüntülenebileceğini unutmayın. Metin dosyalarını standart veritabanı tabloları gibi yönetmek kolay olmadığı için kayıt değişikliği yapılmasına izin verilmemesi normal. Aynı şekilde çok büyük metin dosyalarını kullanırken iyi bir performans beklememelisiniz. Zaten veritabanı sistemleri varken büyük metin dosyalarını tablo olarak kullanmaya ihtiyaç yoktur genelde.

FORM ÜZERİNE RESİM DÖŞEMEK

Formların ve raporların üzerine resim yerleştirmek için özellikler penceresinde birkaç seçenek var. Orada "picture" ile başlayan altı seçeneği inceleyip nasıl kullanıldıklarını ve ne işe yaradıklarını hemen öğrenebilirsiniz.

Resmi döşemek için Picture Tiling özelliği kullanılıyor. Picture özelliğine resim dosyasının konumu yazılıyor. Hemen altındaki Picture Type seçeneğindeki Embedded değeri bu resim dosyasının Access dosyasının içine fiziksel olarak dahil edileceğini belirtiyor. Aksi halde orada Linked değerinin seçili olması resim dosyasının Access dosyasına dahil edilmeyeceğini, resmin belirtilen disk konumundan her seferinde yeniden çağrılıp görüntüleneceğini işaret ediyor. Tabii bu seçenek, görüntüleme süresini biraz artırıyor. Resmin anında görüntülenmesi gerekiyorsa Embedded seçeneğini işaretleyin.

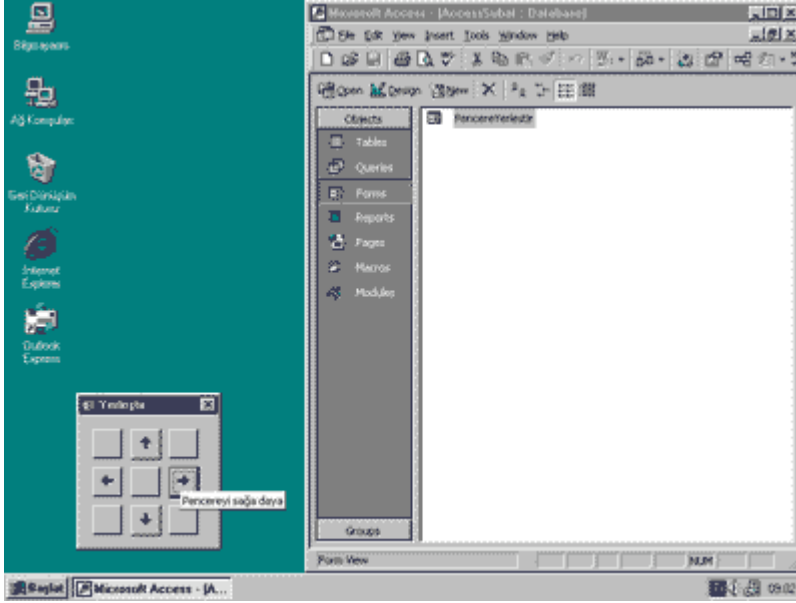
Raporlara özgü Picture Pages seçeneği ile arkaplan resminin sadece ilk sayfada ya da tüm sayfalarda görüntülenmesini sağlayabilirsiniz.



PictureTiling özelliği ile arkaplan resmi form üzerine döşeniyor.

ACCESS PENCERESİNİ YERLEŞTİRMEK

Geçen aylarda Access penceresinin masaüstünde kolayca yerleştirilebileceğiyle ilgili bir kod örneği vardı. Bu sayıdaki örnek ise eskisinin biraz daha gelişmiş hali. Kod biraz uzun olduğu için bu sayfalara kıyamıyorum. Dergide Adı Geçen Dosyalar bölümündeki dosyada kodu bulabilirsiniz.



Access penceresi yerine herhangi başka bir program penceresinin boyutlarını değiştirmek isterseniz koddaki hWndAccessApp yerine ilgili pencerenin Handle (hWnd) değerini yazmanız yeterli.

Bunun için FindWindow gibi bir API komutuna ihtiyacınız olabilir. Onu da şu şekilde yapabilirsiniz:

```
Public Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
```

```
Sub Deneme()  
Dim h As Long  
h = FindWindow("Notepad", "Adsız - Not Defteri")  
MsgBox "Not Defteri'nin HWND değeri = " & h  
End Sub
```

TUŞ BASILI MI DEĞİL Mİ

Windows'ta ve diğer programlarda olduğu gibi Ctrl, Shift ve Alt tuşlarını kısayollara eklemek ve işlem anında bunların basılı olup olmadığını kontrol etmek profesyonel bir görünüm kazandırabilir programlarımıza.

Örneğin bir sıralama işlemi yaptırırken Shift tuşu basılı ise sıralamayı tersten yaptırırız.

Yeni bir modül oluşturup en üstteki Option Compare Database ve varsa Option Explicit satırlarının altına şu satırları yerleştirin:

```
Public Declare Function GetKeyState Lib "user32" (ByVal nVirtKey As Long) As Integer
Public Const VK_SHIFT = &H10 'SHIFT
Public Const VK_CONTROL = &H11 'CTRL
Public Const VK_MENU = &H12 'ALT
```

Programınızın herhangi bir yerinde, basılı olup olmadığını bulmak istediğiniz tuşlar için şu kodu kullanabilirsiniz artık:

```
If GetKeyState(VK_SHIFT) < 0 Then MsgBox "SHIFT tuşu basılı."
If GetKeyState(VK_CONTROL) < 0 Then MsgBox "CTRL tuşu basılı."
If GetKeyState(VK_MENU) < 0 Then MsgBox "ALT tuşu basılı."
```

KAYIT SİLME KODUNUN SON SÜRÜMÜ

Daha önce iki kez farklı kayıt silme kod örnekleri vermiştim. O kadar satırlık kod parçası iki satıra sığabiliyor. İşte son sürüm kayıt silme kodu:

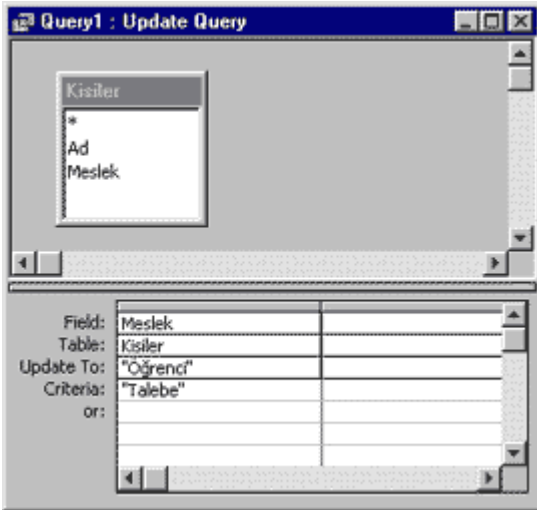
```
DoCmd.SetWarnings False
DoCmd.RunCommand acCmdDeleteRecord
```

HERKES İÇİN SORGULAR

Sıkça sorulan bir SQL sorusu, belli kayıtların bulunup değiştirilmesi ile ilgili.

Access'te yeni bir sorgu oluşturup Query (Sorgu) menüsünden Update Query'yi (Değiştirme Sorgusu) seçin.

Değiştirmek istediğiniz veri alanlarını alt taraftaki sütunlara yerleştirip UpdateTo kısmına yeni değerleri yazın. Sadece belli kayıtları değiştirmek istiyorsanız Criteria (Koşul) alanını doldurmayı unutmayın. Aksi halde bütün kayıtlarınız değiştirilir.



Resimdeki sorgu Meslek alanındaki "Talebe" değerlerini "Öğrenci" olarak değiştiriyor.

UPDATE Kisiler SET Meslek = "Öğrenci" WHERE Meslek="Talebe";

Değiştirilecek alan için koşul belirtmek zorunda değilsiniz. Bu kod Kisiler tablosunda Meslek alanı değerinin "Ev Hanımı" olduğu kayıtlardaki Cinsiyet alanına "Kadın" değerini ve Askerlik alanına "Muaf" değerini yazıyor:

UPDATE Kisiler SET Cinsiyet="Kadın", Askerlik="Muaf" WHERE Meslek="Ev Hanımı";

Koşul belirtmeye gerek yok tabii. Şu SQL kodu ile tüm müşteri kayıtlarındaki MektupGonder alanı True (Evet) olarak güncelleştirilir:

UPDATE Musteriler SET MektupGonder = TRUE;

ODE TOOLS

Şu sıralar **ODE Tools** ile ilgili sorular çoğaldı. İşte kısa cevaplar:

- ODE Tools 2000 = Microsoft Office 2000 Developer Tools.
- ODE Tools ile Access kurulu olmayan bilgisayarlarda Access'te yapılmış programları çalıştırabilirsiniz.
- Access dosyaları EXE yapılmıyor. Ama kodlar ve tasarım bilgileri tamamen gizlenip dosya derlenebiliyor. Bunun için ODE Tools'a gerek yok. Dosyayı MDE dosyası yapmalısınız.
- Program ile Access projeleri için kurulum paketi (setup) hazırlanıyor. Tabii sadece bu işi yapmıyor. İçerisinde Office'te programlama ile ilgili başka eklentiler de var.
- ODE Tools'u bir defa satın almış olmakla, bununla hazırladığınız kurulum programlarınızı sınırsız sayıda ve tamamen ücretsiz dağıtma hakkınız olur. Kurulum programlarınız her bilgisayara, yasal bir kısıtlama olmaksızın kurulabilir.
- Ayrı olarak satılabiliyor ve ayrıca Office 2000 Developer Edition paketi içinde yer alıyor. Office 97 için de aynı şekilde sürümleri mevcut. Fakat ODE Tools 97 şu an piyasada değil.
- Programı Linksoft'tan satın alabilirsiniz. Tel: 0(216) 522 00 22.
- Lisans kontrollerinde öncelikle faturaların incelendiğini unutmayın.
- Programı satın almadan önce PC Magazine'in Temmuz 2000 ve Ocak 2001 tarihli sayılarındaki ODE Tools incelemelerine gözatmanızı tavsiye ederim.

SORULARLA İLGİLİ

Okurlarımızın gönderdiği soruların hemen hepsine cevap yazmaya çalışırım. Fakat bir yığın sorunun tamamına kısa sürede cevap yazmak mümkün değil. Çoğunlukla araştırma yapmak gerekli ve mesai sırasında işten çokça vakit ayırmak pek doğru olmasa gerek. Gelen mesajları vaktimin yettiği kadarıyla akşamları evden yanıtlayabiliyorum genelde.

Sadece bana gönderdiğiniz postalarda değil, başka yerlere gönderdiğiniz postalarda da mesajlarınızın içindeki soruları kısa tutmanız ve birçok soruyu tek bir mesaj içinde toplamamanız cevap alma sürenizi azaltabilir. Ayrıca kullandığınız Access'in (yani probleme konu olan programın) hangi sürüm ve dilde olduğunu belirtirseniz daha iyi olur.

Haber grupları sorulara cevap bulmak için daha efektif oluyor. Oralarda çok daha kısa sürede ve ayrıntılı çözümler bulunabiliyor. Zira yüzlerce kişi sorunuzu okuyor ve problemin çözümünü bilenler mesajınızı cevaplıyorlar.

Sorularınızın birçoğu haber gruplarında geçmiş tarihlerde başkaları tarafından sorulup cevaplanmış olabiliyor. Problem için öncelikle Internet'ten ve haber gruplarından araştırma yapmak daha hızlı bir çözüm olabilir.

Internet'te de Access'le ilgili Web sayfaları mevcut. Bir daha ki ay için Access'le ilgili kişisel web sitelerinin adreslerini gönderirseniz bu bölümde yayınlayacağız. Bu arada Access'le ilgili kapsamlı bir site olan The Access Web sitesini de sıkça ziyaret etmenizde fayda var.

<http://www.mvps.org/access>

Mvps.org sitesinde Access dışında VB, API, VC gibi konularda çok kapsamlı birkaç bölüm daha var. Site, Microsoft tarafından MVP ünvanı (Microsoft's Most Valuable Professional - bir nevi fahri Microsoft mühendisliği) verilmiş kişiler tarafından oluşturulmuş. Oradaki diğer bağlantılar arasında başka MVP'lerin siteleri için de adresler verilmiş. Outlook Express için epey kapsamlı bilgiler içeren bir MVP sitesi bile varmış nitekim.

Okurlarımızdan gelen bazı soruların cevapları ise eski sayılarımızda yer alabiliyor. Mayıs 2000'de başlayan bu dizinin önceki yazılarını PC Magazine Online sitesindeki Eski Sayılar Arşivi bölümünde bulabilirsiniz. Bu yazıları da elinizin altında bulundurmanız faydalı olacaktır.

PC Magazine sitesindeki Dergide Adı Geçen Dosyalar bölümünde ise aynı tarihten başlayacak şekilde bazı aylara ait Access'le ilgili örnek çalışma dosyaları yer alıyor.

DOSYA:

[AccessNisan2001.zip](#)

Serkan Şahinoğlu
PC Magazine, Nisan 2001

<http://BilgiTeknoloji.net> 



DB1.MDB

Projenizin başından sonuna kadar nelere dikkat etmelisiniz? Bitmiş bir Access projesini nasıl dağıtmalı? İşte yeni teknikler ve yeni ipuçları.

Bu sayfalarda Access projelerinizin başından sonuna dek dikkat etmeniz gerekebilecek önemli bazı ayrıntıları, uygulayabileceğiniz ilginç teknikleri ve önemli ipuçlarını bulacaksınız. İşte yeni teknikler ve yeni kotarılmış ipuçları!

...

BİR ACCESS PROJESİ

Access dosyasını düzenli tutmamak, masamızı sürekli dağınık bırakmak gibi sonradan can sıkıcı olabilir. Bir projeye başlamadan önce yapılması önerilen ilk işlerden biri, proje için anlaşılır ve kullanışlı dosya ve nesne adları belirlemektir.

Dosya gruplarından oluşmuş projeler gibi Access dosyaları da obje gruplarından oluşur. Formlar, raporlar, modüller görünür objedirler. Bununla beraber Access kendi içinde görünen ve görünmeyen birçok objeyi barındırır.

Container (Konteyner) küme nesneleri, içlerinde bildiğimiz objeleri, yani formları, raporları, tabloları, indeksleri ve diğer nesneleri taşıyabilirler (DAO.Database.Containers). Bunlar dışında bizim bilmediğimiz birçok görünmeyen objenin de bu tip konteynerler içinde bulunması muhtemel.

Gelgelelim, bu gibi zor işleri Access yapıyor ve bize olabildiğince az iş bırakıyor. Çoğunlukla seçenekler arasında dolaşıp fareyle işaretleyerek bir Access projesi üretebiliyoruz. İşte bu kolaylıklar yüzündendir ki, Access'in programlama aracı olamayacağı yönünde bir görüş de var nitekim. Access'i yapanlar da "yapsak bir türlü, yapmasak iki türlü" diye dert yanmaktadırlar herhalde!

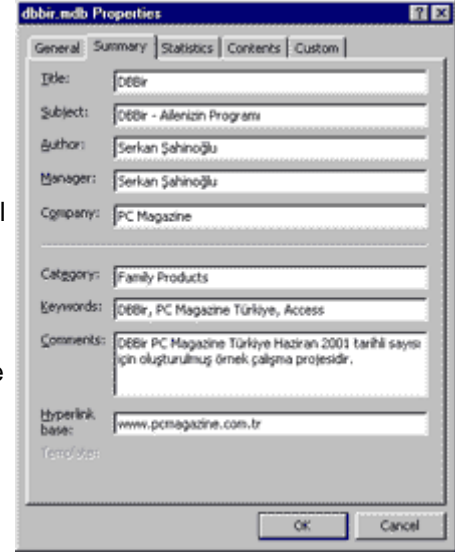
DOSYAYA KİMLİK

Bir Access dosyasını oluşturduğunuzda bu dosyaya adınızın kaydedildiğini unutmayın.

File > Database Properties (Dosya > Veritabanı Özellikleri) menüsünü açın. Gerekli gördüğünüz alanları doldurabilirsiniz. Buraya yazacağınız bilgiler dosyanız için genel amaçlı açıklamalar niteliğinde olmalıdır.

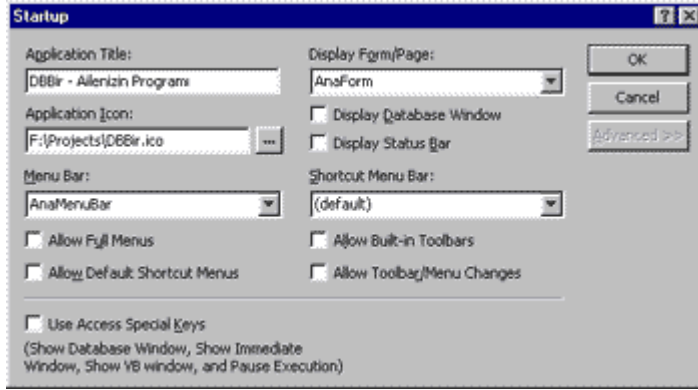
Siz hiçbir şey yazmış olmasanız dahi, dosyayı ilk oluşturduğunuzda bilgisayarınızda kayıtlı olan isminiz ve diğer bazı bilgileriniz Office'in diğer üyelerinde olduğu gibi Access'te de dosya özellikleri sayfasına kaydedilecektir. Bu özellik hemen her Office bileşeninde olmasına rağmen pek bilinmiyor ya da kullanılmıyor.

File > Database Properties menüsünü açıp özet bilgileri yazın.

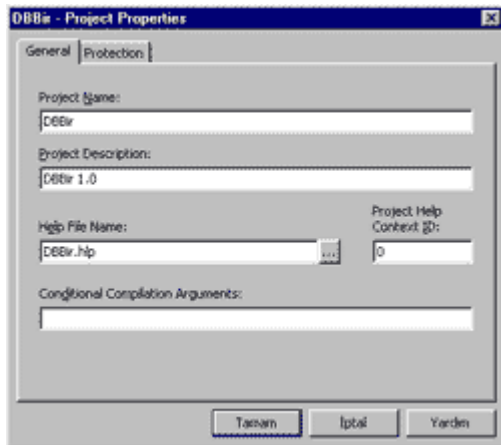


Buraya kadar aslında bir Office bileşeni olarak bu Access dosyası hakkında genel bilgiler yazmış oldunuz. Uygulama ile ilgili bilgileri ise başka iki bölümde yazmanız gerekiyor.

Tools > Startup (Araçlar > Başlangıç) menüsünü açarak Application Title (Uygulama Başlığı) kutusuna bu uygulamanın Windows'ta görüntülenecek açık adını yazın. Başlat panelinde buraya yazdığınız isim görüntülenecektir.



Son olarak Tools > Macro > Visual Basic Editor (Araçlar > Makro > Visual Basic Düzenleyicisi – ALT+F11) menüsü ile Visual Basic Editor'ü açın. Tools > DBBir Properties (Araçlar > DBBir Özellikleri) menüsü ile proje adını yazabilirsiniz. Project Name (Proje Adı) kutusuna uygun bir isim yazmanız gerekiyor.



EN KOLAY SIRALAMA

Bazı programlarda bir listenin başlığını tıklattınca ilgili veri alanına göre sıralama yapılıyor. Windows Gezini ya da Outlook Express'te bu şekilde kolayca sıralama yapılabilir. Access'te bunun aynısını yapmamız için bir engel yok tabii ki.

İş biraz uzun ama yaptığınıza değecek cinsten. Önce resimdeki gibi bir form oluşturun. Aşağı ve yukarı ok işaretlerini Webdings yazıtipinden alacağız. “5” ve “6” karakterleri bu ok işaretlerine karşılık geliyor. İki Label (metin etiketi) oluşturup bunları yeşil olarak renklendirdikten sonra adlarını labelYonASC ve labelYonDESC olarak ayarlayın.

Üst şeritteki altı çizili label (etiket) nesnelerinin adları sıra ile labelID, labelKitapAdı, labelYazar, labelYayıncı, labelYıl, labelÇeviren, labelGrup, labelGirisTarihi olacak. Yine bu etiketlerin Tag özelliklerini sıra ile ID, KitapAdı, Yazar, Yayıncı, Yıl, Çeviren, Grup, GirisTarihi olarak ayarlayın.

Aynı şekilde bu Label nesnelerinin HyperlinkAddress özelliklerini de “ ” (sadece bir adet boşluk) olarak ayarlayın. Son olarak Tablo1'deki uzun kodu bu formun kod sayfasında en üstteki

Option Compare Database
Option Explicit
'(Bu satır yoksa ekleyebilirsiniz.)

satırlarının altına ekleyin.

Nihayetinde listede üstteki altı çizili isimleri fareyle tıklatarak kolaylıkla artan veya azalan sıralama yaptırabilirsiniz. Doğruca azalan sıralama yapmak için tıklama anında Shift tuşunu basılı tutmak yeterli.

ID	Kitap Adı	Yazar	Yayıncı	Yıl	Çeviren	Grup	Giriş Tarihi
5	Kemal ve Sıyah - 2	Stendhal	MEB	1997	Nurullah Ataç	Dünya Edebi	02.04.1999
1	Kemal ve Sıyah - 1	Stendhal	MEB	1997	Nurullah Ataç	Dünya Edebi	12.03.2001
8	Bitmeyen Oğun	Peter Hopkirk	Yeni Yüzyıl	1995	Mehmet Hamancı	Tarih	05.02.2001
10	Windows95 Programcı Kılavuzu	Microsoft Corpore	Arkadaş	1996	Ceyhan Temüroğlu	Bilgisayar	15.12.1998
6	Kadınlar Duygusal Zeka	Mariella Sartorius	Varlık	1999	Şebnem Can Erendor	Psikoloji	01.06.2000
9	Nineteen Eighty-Four	George Orwell	Penguin	1954		Diğer	13.02.1997
7	Oracle7 Workgroup Server	Dr. Yalçın Özkan	Ant	1996		Bilgisayar	05.05.2001
4	Delikanlı - 3	Dostoyevski	MEB	1997	Servet Lünel	Dünya Edebi	02.09.2000
3	Delikanlı - 2	Dostoyevski	MEB	1997	Servet Lünel	Dünya Edebi	01.03.2001
2	Delikanlı - 1	Dostoyevski	MEB	1997	Servet Lünel	Dünya Edebi	14.02.2001

Yazar alanına göre ters sıralı.

Form üzerinde kolay sıralama için gerekli kod

```
Private Declare Function GetKeyState Lib "user32" (ByVal nVirtKey As Long) As Integer  
Private Const VK_SHIFT = &H10
```

```
Private Sub Sirala(labelx As Label)
```

```
On Error Resume Next
```

```
Dim fieldx As String
```

```

fieldx = labelx.Tag

If ((OrderBy = fieldx) Or (GetKeyState(VK_SHIFT) < 0)) Then
'Shift tuşu basılı ise ters sırala.
OrderBy = fieldx & " DESC"
labelYonASC.Visible = False
labelYonDESC.Visible = True
labelYonDESC.Left = labelx.Left + labelx.Width - 60
labelYonDESC.Top = labelx.Top - 20
Else
OrderBy = fieldx
labelYonDESC.Visible = False
labelYonASC.Visible = True
labelYonASC.Left = labelx.Left + labelx.Width - 60
labelYonASC.Top = labelx.Top - 20
End If

OrderByOn = True

End Sub

Private Sub Form_Error(DataErr As Integer, Response As Integer)
Select Case DataErr
Case 7983: 'Hyperlink Hatası
Response = acDataErrContinue
End Select
End Sub

Private Sub labelCeviren_Click()
Siralama labelCeviren
End Sub

Private Sub labelGirisTarihi_Click()
Siralama labelGirisTarihi
End Sub

Private Sub labelGrup_Click()
Siralama labelGrup
End Sub

Private Sub labelID_Click()
Siralama labelID
End Sub

Private Sub labelKitapAdi_Click()
Siralama labelKitapAdi
End Sub

Private Sub labelYayinci_Click()
Siralama labelYayinci
End Sub

Private Sub labelYazar_Click()
Siralama labelYazar
End Sub

Private Sub labelYil_Click()
Siralama labelYil
End Sub

'Her bir başlık etiketinin Click (Tıklatıldığında) olay
'yordamını ve formun Error yordamını Özellikler

```


'penceresinde bu yordamlarla ilişkilendirin.

KAYIT SEÇİCİLERDEN SIKILDIYSANIZ

Formlarda sol tarafta bulunan kayıt seçicilerden sıkıldıysanız aşağıdaki adımları takip ederek kendi kayıt seçicilerinizi yapabilirsiniz. Öncelikle Otomatik Sayı (AutoNumber) benzeri bir veri alanı kullanılması gerekiyor. Bu örnekte ID adlı bir AutoNumber alan olduğu varsayılıyor.

Formunuzun sol tarafına bir metin kutusu (Textbox) yerleştirin. Boyunu yeteri kadar kısaltın. Özellikler (Properties) penceresinden, aşağıdaki değerleri ayarlayın:

Ad (Name) = Secici
Enabled (Seçili?) = No (Hayır)
Locked (Kilitli) = Yes (Evet)
BackStyle (Arkaplan Biçimi) = Transparent (Saydam)
BorderStyle (Kenarlık Biçimi) = Transparent (Saydam)
FontName (Yazıtipi) = Webdings

Bu değerleri atadıktan sonra formun Current olay yordamını aşağıdaki gibi düzenleyin:

```
Private Sub Form_Current()  
Secici.ControlSource = "=IIf([ID]= " & _  
Nz(Me!ID, 0) & ", '8', '4')"  
End Sub
```

ID	Kitap	Yazar	Yayıncı	Yıl	Çeviren
1	Kırmızı ve Siyah - 1	Stendhal	MEB	1997	Nurullah Ataç
2	Delikanlı - 1	Dostoyevski	MEB	1997	Servet Lünel
3	Delikanlı - 2	Dostoyevski	MEB	1997	Servet Lünel
4	Delikanlı - 3	Dostoyevski	MEB	1997	Servet Lünel
5	Kırmızı ve Siyah - 2	Stendhal	MEB	1997	Nurullah Ataç
6	Kadınlar da Duygusal Z	Mariella Santorius	Varlık	1999	Şebnem Can Erendor
7	Özdece? Workshop Se	Dr. Yalçın Özkan	Anı	1996	
8	İstanbul'da Doğuşunda	Peter Hopkirk	Yeni Yüzyıl	1995	Mehmet Hamancı
9	Nineteen Eighty-Four	George Orwell	Penguin	1954	
10	Microsoft Windows 95	Microsoft Corpora	Akıadag	1996	Ceyhan Temürücü

Record: 14 of 10

Sol tarafta kendi kayıt seçicimiz duruyor.

Resimdeki gibi kayıt seçicimiz sol tarafta duruyor. Metin kutusunun üzerindeki resmin görüntülenmesi için Webdings yazıtı sistemde yüklü olmalı.

LİSTEYİ SÜZ

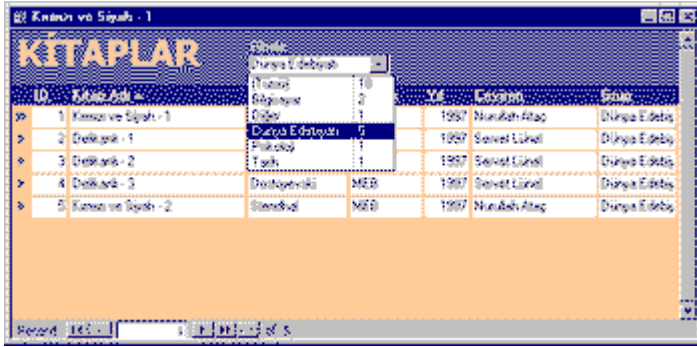
Kayıtlarınızı belli gruplara göre ayırmak istiyorsanız Access'in süzme özelliğini kolaylıkla kullanabilirsiniz. Bizim yöntemimizde bir Combobox (Açılan Liste Kutusu) nesnesi gerekiyor. Adı cmbFiltre. Açılıştaki ilk değeri görüntülemek için listenin Default Value (Geçerli Değer) özelliğini "(Tümü)" olarak atayın. RowSource (Kayıt Kaynağı) özelliğini de alttaki şekilde ayarlayın:

```
(
SELECT Grup as GrupAdi,
COUNT(Grup) AS Adet
FROM Kitaplar
GROUP BY Grup
)
UNION
(
SELECT "(Tümü)" as GrupAdi,
```

COUNT(ID) as Adet
FROM Kitaplar
);

Sonra bu listenin AfterUpdate (Güncelleştirme Sonrası) olay yordamını şu şekilde düzenleyin:

```
Private Sub cmbFiltre_AfterUpdate()  
If IsNull(cmbFiltre) Then Exit Sub  
Dim orderByx As String, orderbyonx As Boolean  
'Eski sıralama korunacak.  
orderByx = OrderBy  
orderbyonx = OrderByOn  
If cmbFiltre = "(Tümü)" Then  
Me.RecordSource = "Kitaplar"  
Else  
Me.RecordSource = "SELECT * FROM Kitaplar " & _  
"WHERE Grup='" & cmbFiltre & "'" & _  
End If  
OrderBy = orderByx  
OrderByOn = orderbyonx  
End Sub
```



KAYDA GİT

DBBir programımıza şimdi de kayda gitme özelliği eklememiz gerekiyor. Önce txGit adlı bir metin kutusu (Textbox) oluşturun. Buna yazdığınız metni Kitap adları arasında bulmak için şu kodu kullanın:

```
txKitapAdi.SetFocus  
DoCmd.FindRecord txGit, acAnywhere, , acDown, , False
```

Kolaylık olması için F5 tuşuna basınca bu metin kutusuna odaklanmayı sağlayalım. Formun KeyPreview adlı özelliğini True, formdaki KeyDown (Tuşa Basarken) yordamını da alttaki gibi yapın:

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)  
Select Case KeyCode  
Case vbKeyF5: txGit.SetFocus  
End Select  
End Sub
```

Bu örnek görüntülenen kayıt kümesi üzerinde aranan değeri bulup getirmeye ilgilidir. Peki görünen kaydın ayrıntılı bilgilerini başka bir formda göstermek için ne yapmalıyız?

Bunun birkaç yolu var. Bizim yöntemimizde önce ayrıntılı bilgilerin yer aldığı detay formu oluşturulmalı. Ardından bu detay formunun RecordSource (Kayıt kaynağı) özelliğini şu şekilde ayarlayın:

```
SELECT Kitaplar.* FROM Kitaplar  
WHERE ((ID=[Forms]![Kitaplar_Liste]![ID]));
```

Ana formumuzun adını Kitaplar_Liste olarak belirlediğim için bu kodda da aynı adı kullandım. Tabii detay formunu açmadan önce bu liste formunun açık olması gerekiyor. Bu formu açtığınızda liste formunda görünen kaydın ayrıntısı açılacaktır.

ODE TOOLS İLE PROJENİZİ DAĞITIN

Geldik DBBir projesinin dağıtım paketinin hazırlanmasına. Amacımız DBBir projesinin kod ve tasarım bilgilerini tamamen gizlemek ve programın Access kurulu olmayan bilgisayarlarda çalıştırılmasını sağlamak.

Tasarım bilgilerini gizlemek için Tools > Database Utilities > Make MDE File (Araçlar > Veritabanı Seçenekleri > MDE Dosya Yap) menüsü ile MDE dosyasını oluşturun. Bunu yaptığınızda derlenmiş yeni MDE dosyasında tasarım ve kod bilgileriniz tamamen gizlenmiş olacaktır.

Dağıtım işini ise ODE Tools ile yapacağız. Öncelikle ODE Tools'un bilgisayarınızda kurulu olması gerekiyor (Bilgi için Nisan 2001 tarihli PC Magazine'deki Access yazısı).

Öncelikle birkaç ayrıntının aklınızın bir köşesinde bulunmasında fayda var.

Kurulumu hazırladığınız bilgisayarda Access'in hangi dilde olduğuna dikkat edin. Programınız hep o dili kullanacaktır.

ODE Tools ile hazırladığınız kurulum programı iki temel bileşenden oluşur. Sizin program dosyalarınız ve Access Runtime

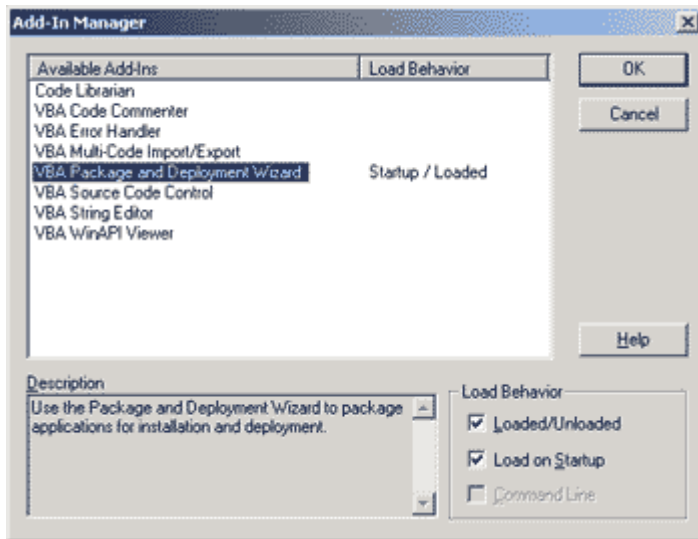
Access Runtime programınızı çalıştıracak modüldür ve her bilgisayara sadece bir kez kurulması yeterlidir. Zaten bu modül üzerinde tek başına 70 MB. yer kaplıyor. Bu durumda hazırladığınız paket ortalama 75 MB. büyüklüğünde olacaktır. İsterseniz kurulum hazırlarken Access Runtime modülünü eklemiyorsunuz. Bu durumda 3 MB. civarında bir paket çıkıyor ortaya.

Bir kereye mahsus olarak 75 MB. büyüklüğünde bir paket hazırlayıp sonraki paketlerinize Access Runtime eklemeyebilirsiniz.

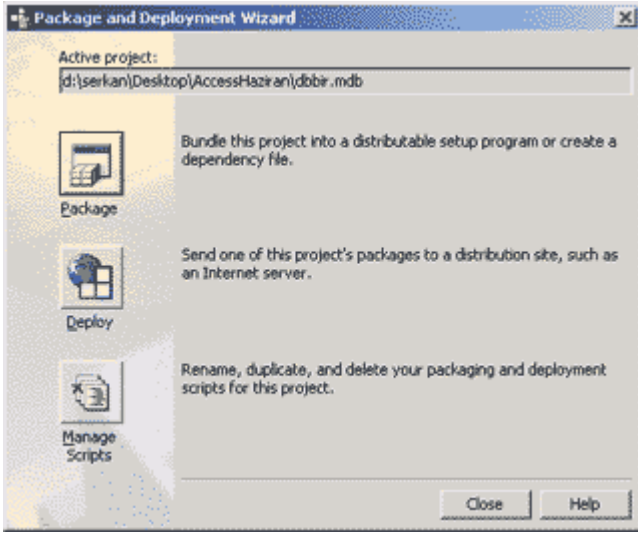
KURULUM PAKETİ HAZIRLAMA

ODE Tool'un bir önceki sürümü kurulumdan sonra kendisini Programlar menüsüne yerleştiriyordu. ODE Tools 2000 kurulduğunda ise kendisini bulabilmek için biraz çaba sarfetmek gerekiyor.

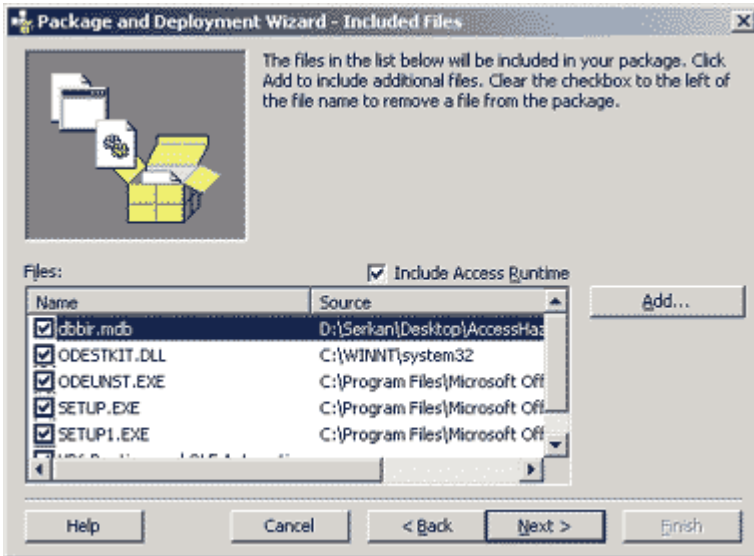
ALT+F11 ile Visual Basic Editor'ü açıp Add-Ins > Add-In Manager menüsü ile VBA package And Deployment Wizard'ı yüklemek gerekiyor. Bu haliyle Add-Ins menüsü altında Package And Deployment Wizard menüsü belirecektir. Bunu çalıştırmadan önce Access dosyasının Exclusive (tek kullanıcı oturumu) modda açılması gerekli.



Dosyayı kapatın ve dosya açma penceresinde dosyayı seçip sağ alttaki düğmeyi Open (Aç) durumundan Exclusive Open haline getirin. Exclusive olarak açtıktan sonra Visual Basic Editor'e geçip ilgili menüyü çalıştırın. Açılan formdan Package'yı tıklayın.



Devam eden sayfalarda önerilen komutları seçin. Gerekli görmüyorsanız ilerleyen sayfalardaki seçeneklerin biri ile Access Runtime'i kurulum paketinize dahil etmeyebilirsiniz.



Zira ilk seferde onlarca megebayt büyüklüğünde bir paketle karşı karşıya kalabilirsiniz. Öncelikle Microsoft.com sitesinde önerilen bazı güncellemeleri yapmanız gerekebilir.

MOD2000: Access 2000 Run-Time Minimum Now Available for Download

<http://support.microsoft.com/support/kb/articles/Q243/9/57.ASP>

Kurulum hazırlama anında hatalarla karşılaşırsanız dosyanızda gereksiz yüklenmiş olan bazı başvuruları kaldırmayı deneyin. Visual Basic Editor > Tools > References.

Aynı şekilde hazırlama işlemi çok uzun sürüyorsa paketinize yüzlerce dosyanın ekleniyor olması muhtemeldir. Hazırlık aşamasında "Include Access Runtime" seçeneğini işaretlerseniz "C:\Program Files\Microsoft Office\ODETools\V9\Runtime" klasörünün tamamı paketinize dahil edilir. Ben bu klasörün yedeğini aldıktan sonra içindeki gereksiz olduğunu düşündüğüm bazı dosya ve klasörleri sildim.

Herşey olması gerektiği gibi ise ODE Tools ile, nasıl yapıldığını anlatmama gerek kalmayacak kadar kolay bir şekilde dağıtım paketinizi (setup) hazırlayabilirsiniz.

HERKES İÇİN SORGULAR
GRUPLAYARAK TOPLAMLAR ALMAK

Sorgularla ilgili en çok sorulan sorulardan biri gruplamalar hakkında.

- Her ürün ana grubunun içinde kaç adet ürün var?
- Her güne ait toplam satış tutarını nasıl hesaplayacağım?

Yukarıdakiler gibi sorularınız varsa GROUP BY deyimini kullanmanız gerekiyor demektir. Bunu en kolay yapabileceğimiz y Access'teki Query Builder.

Yeni bir Query (Sorgu) oluşturup içine Kitaplar tablosunu ekleyin. Veri alanlarını alttaki sütunlara yerleştirip View > Totals (Görünüm > Toplamlar) menüsünü seçin ve Query1 resmindeki gibi sorguyu düzenleyin.

Bu haliyle sadece grup adını ve o grupta kaç kitap olduğunu bulmuş oluyoruz. Geçen aylarda bahsettiğimiz UNION sorgu ile toplam kitap adedini de bu sorguya dahil edelim.

SELECT GrupAdi, Adet

FROM


(
SELECT Grup as GrupAdi,
COUNT(Grup) AS Adet, 0 AS Sira
FROM Kitaplar
GROUP BY Grup

UNION

SELECT "(TOPLAM)" as GrupAdi,
COUNT(ID) as Adet, 1 AS Sira
FROM Kitaplar
)

AS Liste

ORDER BY Sira, Adet DESC;



GrupAdi	Adet
Dünya Edebiyatı	5
Bilgisayar	2
Tarih	1
Psikoloji	1
Diğer	1
(TOPLAM)	10

Query2 resminde gördüğünüz gibi her bir grup adını ve içinde kaç adet kitap olduğunu listelemekle beraber en altta da toplam kitap adedini göstermiş oluyoruz.

Sorguda kullandığımız Sira adlı alan UNION sorgu içindeki farklı iki kümeyi birbirinden ayırmak içindi. İlk kümeyi 0, ikincisini 1 yapıp bu alanı en altta ORDER BY içine yerleştirmek yeterli oluyor.

Genel olarak adet bulma işlemleri için COUNT deyimini kullanmalısınız. Toplam hesaplamak için ise SUM deyimini kullanılır. AVG, MIN, MAX, STDEV, FIRST, LAST gibi deyimlerle de ortalama, en küçük, en büyük, standart sapma, ilk, son gibi değerler elde edebilirsiniz.

AÇILIŞTA TAM EKRAN OLSUN

tam ekran açılmasını tercih ederim. Bunun için Makrolar (Macros) sayfasında AutoExec adlı yeni bir makro oluşturmak yeterli.

Access açıldığında her zaman AutoExec adlı bir makro arar ve bulursa onu çalıştırır. Windows'taki Autoexec.bat dosyası gibi. Bu makroyu oluşturduktan sonra ilk satırda Action sütununa gelip listeden "Maximize"yi seçin ve makroyu kaydedip kapatın. Bundan sonraki her açılışta veritabanı penceresi tam ekran açılacaktır.

Aynı şekilde Tools > Startup (Araçlar > Başlangıç) menüsündeki seçenekleri kullanarak projenizin başlığını, başlangıçta açılmasını istediğiniz formları burada belirleyebilirsiniz. Gerekliyse veritabanı pencerenizi gizlemeniz, kısayol tuşlarını kısıtlamanız mümkün. Böylece gizlediğiniz veritabanı penceresini kimsenin görmemesini sağlamış oluyorsunuz.

DOSYANIN İÇİNİ KİMSE GÖRMESİN

AutoExec makrosu ve diğer başlangıç seçeneklerini bir şekilde es geçme şansımız vardır. Bir Access dosyasını açarken klavyede Shift tuşunu basılı tutarsanız AutoExec makrosu çalıştırılmaz. Aynı şekilde diğer başlangıç seçenekleri de geçerli olmayacaktır. Buna Tools > Startup menüsündeki Application Title (Uygulama Başlığı) özelliği ve veritabanı penceresinin görüntülenmeme seçeneği de dahil.

Fakat bunu da gidermenin yolu var. Aşağıdaki kodu kullanarak dosyanızın Shift tuşu ile açılıp içindeki tabloların, sorguların ve diğer nesnelerin görüntülenmesini engelleyebilirsiniz. Öncelikle Başvurular (References) listesinden DAO'yu yüklemiş olmanız gerekiyor.

Dim db As Database

Set db = OpenDatabase("c:\ldb1.mdb")

Dim prop

Set prop = db.CreateProperty("AllowByPassKey", DB_BOOLEAN, False)

db.Properties.Append prop

db.Close

Bu şekilde Access dosyasının içinde, geçerli olarak False değerli AllowByPassKey adlı bir Property oluşturuluyor. Böylelikle Shift tuşu basılı olarak açılrsa da başlangıç seçeneklerinde belirlediğiniz veya gizlediğiniz özellikler kimse göremeyecektir.

DOSYANIN İÇİNİ KİMSE GÖRSÜN!

Üstteki kodu kullanarak Shift ile dosyanızın açılmasını engellediğinizi kesinlikle sağlayabileceğinizi mi düşünüyorsunuz? Maalesef bunu da aşmanın yolu yok değil. Dosyanız MDE dosyası olsa da.

Dim db As Database

Set db = OpenDatabase("c:\ldb1.mdb")

db.Properties("AllowByPassKey") = True

db.Close

Ama unutmayın ki, bunu ancak Access'te kod yazmayı bilen bir kişi yapabilir. Kullanıcılarınız Access'te kod yazmayı bilmiyorsa üstteki örneği de denemeye uğraşmazlar herhalde.

Programınızı güvenli dağıtmak istiyorsanız tek önerim MDE dosyası olarak derlemeniz. Ne kadar kısıtласanız ya da parola koysanız da, bunu çözen programlar Internet'te mevcut.

ETİKET DEDİĞİN YAPIŞKAN OLMALI

Form üzerine yeni bir metin kutusu (Textbox) yerleştirdiğimizde geçerli olarak yanına bir de metin etiketi (Label) iliştiliyor. Kutuyu taşısanız da bu yapışkan etiket onunla hareket ediyor. Buna gerek olmadığı durumlarda etiketi seçip silebiliyoruz. Fakat bazen buna tekrar ihtiyacımız olabiliyor.

Forma yeni bir metin etiketi (Label) yerleştirin. Bunu seçip klavyede CTRL+C tuşlarına basarak hafızaya alın. Sonra yapıştırmak istediğiniz metin kutusunu seçerek CTRL+V ile etiketi kutuya yapıştırın.

Yazıdaki örnekleri PC Magazine Online sitesinde Dergide Adı Geçen Dosyalar bölümündeki dosyada bulabilirsiniz.

Güneşli güzel günler...

DOSYA:

[accessdb1.zip](#)

- Access XP Özel -

Access XP ile Yeni Deneyimler

Yeni Access beraberinde neler getiriyor? Access XP'nin derinliklerine inip, yenilikleri ve yeni Access'le yapabileceklerimizi araştırdık.



Nihayet piyasaya çıkan Office XP ile birlikte Access de XP sürümü ile karşımıza çıkıyor. Öncekine nazaran görünürde çok değişiklik içermeyip, yeteneklerin artırılması ve sunulan yeni bileşenler, görünüşüne aldandığım yeni Access konusundaki fikirlerimi yavaş yavaş değiştirdi.

Bununla birlikte Office'ten aldığı özellikler ve beraberinde sunulan diğer araçlar, önceki Access'te bazı eksikliklerin olduğunu hissettiriyor.

Access XP ya da diğer adıyla Access 2002'ye derinlemesine bir dalış yapıyor, getirilen yenilikleri, yapılan değişiklikleri ve y Access ile yapabileceklerimizi inceliyoruz.

NELER YENİ

Access 2002 yeni Office'in Professional ve Developer adlı sürümleri içinde yer alıyor ve ayrı kutu olarak satılıyor. Access'te yapılan programları sınırsız dağıtmak için kullanılan Office Developer Tools ise tek satılmayıp sadece Office Developer pa içinde geliyor.

Access 2002'nin masaüstü uygulamaları alanında üç ana hedefi olduğu belirtiliyor:

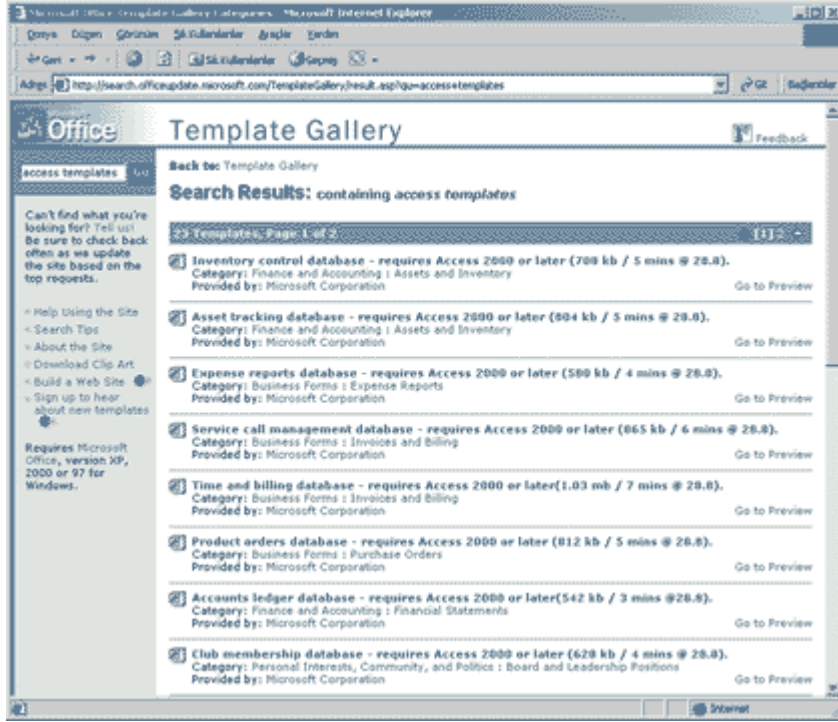
- * Web tabanlı erişim, analiz ve raporlama araçları
- * Birlikte çalışma ve paylaşım desteği
- * Hızlı geliştirme ve yönetme

Eski sürümle uyumlu çalışabilen Access XP, Web ağına biraz daha kök salmış görünüyor. XML desteği ile birlikte dinamik sayfalar sunabilen bu yeni sürüm, Internet ve yerel ağ ortamında veri paylaşım ve sunum olanağını üst düzeylere çıkarıyor. Veri sayfalarının (Data Access Pages) yetenekleri geliştirilerek daha kullanışlı web formlarının ve akıllı web grafiklerinin oluşturulmasına olanak sağlanıyor.

SQL Server uyumluluğu daha artırılmış, veri sayfalarıyla SQL Server verilerinin çevrimdışı olarak kullanılabilmesi sağlanmış. Çevrimiçi özellikleriyle canlı veriler ve canlı grafikler sunmak artık mümkün oluyor.

Yeni Pivot Table ve Pivot Chart adlı bileşenler sorgulama ve raporlama işlerimize farklı bir boyut getiriyor. Piyasadaki diğer raporlama araçlarında önceden beri yer alan Pivot nesneleri artık Access'te en ön sıralarda yerlerini alıyorlar.

Office'in bu sürümüyle birlikte, çok yararlı olabilecek kaynakların, dokümanların, şablonların, resim galerilerinin bulunduğu Web servisleri oluşturulmuş. Access'te ofisinizde kullanılacak bir program yapmak istiyorsanız Office XP'nin Web sitesindeki Template Gallery'de bulabileceğiniz, çalışır halde onlarca şablon dosyası var.



YENİ PIVOT BİLEŞENLERİ VE CANLI GRAFİKLER

Access'teki Crosstab Query (Çarpraz Sorgu) nesnelerini kullanmış olanlar Pivot Table'a çabuk alışacaktır. Adında Table kelimesi geçse de bu nesne formlar bölümünde yer alıyor. Form sihirbazı ile yeni bir Pivot Table oluşturulabiliyor. İlk aşamada hangi tablonuz üzerinde sorgu işlemi yapacağınızı seçiyorsunuz. Sonrasında veri alanlarını bir aşağı, bir yukarı taşıyarak alanları birbiri ile istediğiniz şekilde karşılaştırıp gruplayabiliyor, süzebiliyor, genel toplamları, adetleri anında görebiliyorsunuz.

The screenshot shows a PivotTable in Microsoft Access. The PivotTable is based on the 'Satışlar' table. The PivotTable is set to show data for 'Beyaz Eşya' (White Goods) by 'Ay' (Month). The PivotTable has two columns: 'Adet' (Quantity) and 'Tutar' (Amount). The data is summarized by month, with a 'Grand Total' row at the bottom.

Ay	Adet	Tutar
1	710	16259
2	840	19236
3	650	14885
4	550	12595
5	570	13053
6	700	16030
7	600	13740
8	650	14885
9	640	14656
10	490	11221
11	430	9847
12	390	8931
Grand Total	7220	165338

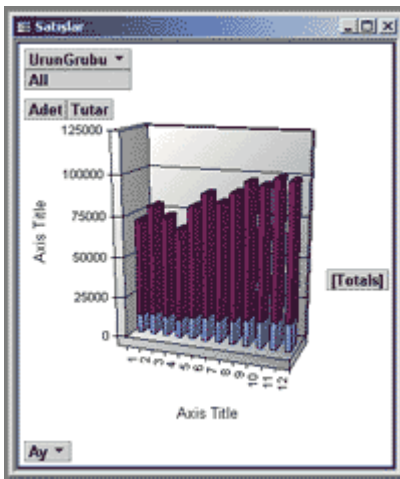
Üst taraftaki seçenek yardımıyla sadece Beyaz Eşya ile ilgili bilgiler listelendi.

The screenshot shows a PivotTable in Microsoft Access. The PivotTable is based on the 'Satışlar' table. The PivotTable is set to show data for all product groups ('UrunGrubu') by 'Ay' (Month). The PivotTable has two columns: 'Tutar' (Amount) and 'Adet' (Quantity). The data is summarized by month, with a 'Total' row at the bottom.

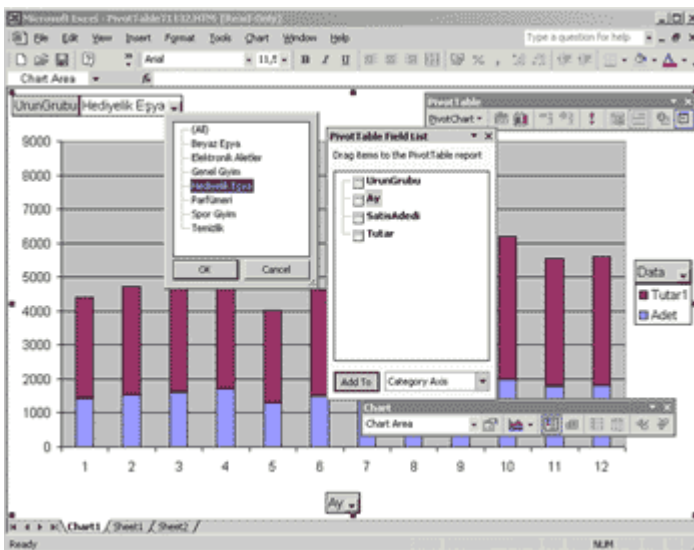
UrunGrubu	Ay	Tutar	Adet
Beyaz Eşya	1	16259	710
Beyaz Eşya	2	19236	840
Beyaz Eşya	3	14885	650
Beyaz Eşya	4	12595	550
Beyaz Eşya	5	13053	570
Beyaz Eşya	6	16030	700
Beyaz Eşya	7	13740	600
Beyaz Eşya	8	14885	650
Beyaz Eşya	9	14656	640
Beyaz Eşya	10	11221	490
Beyaz Eşya	11	9847	430
Beyaz Eşya	12	8931	390
Total		165338	7220
Elektronik Aletler		31941	5070
Genel Giyim		230136	53520
Hediyeleş Eşya		44604	21240
Parfümler		64791	28170
Spor Giyim	1	16767	2070
Spor Giyim	2	16848	2080
Spor Giyim	3	20250	2500
Spor Giyim	4	18387	2270
Spor Giyim	5	23895	2950
Spor Giyim	6	25920	3200
Spor Giyim	7	25110	3100
Spor Giyim	8	28350	3500
Spor Giyim	9	30780	3800
Spor Giyim	10	32481	4010

Drop Filter Fields Here																			
		UrunGrubu		Beyaz Eşya		Elektronik Aletler		Genel Giyim		Hediyelik Eşya		Parfümeri		Spor Giyim		Temizlik		Grand Total	
Ay		Tutar	Adet	Tutar	Adet	Tutar	Adet	Tutar	Adet	Tutar	Adet	Tutar	Adet	Tutar	Adet	Tutar	Adet	Tutar	Adet
1		16259	710	3643	610	13201	3070	2982	1420	4692	2040	16767	2070	2760	1150	60504	11070		
2		19236	840	3654	580	17544	4080	3192	1520	4784	2080	16848	2080	3120	1300	68378	12480		
3		14895	660	2142	340	16090	3600	3360	1600	3887	1690	20250	2500	3600	1500	63174	11780		
4		12595	550	1890	300	14061	3270	3570	1700	3542	1540	18387	2270	2880	1200	56925	10830		
5		13553	570	2690	460	16905	3950	2730	1300	6003	2610	23895	2950	3360	1400	68924	13240		
6		16030	700	3087	490	18060	4200	3150	1500	6440	2800	25820	3200	3480	1450	76167	14340		
7		13740	600	3276	520	17630	4100	3990	1900	4646	2020	25110	3100	3624	1510	72016	13750		
8		14895	660	1764	280	19950	4500	4620	2200	5313	2310	28390	3600	3000	1290	77282	14690		
9		14696	640	2016	320	20940	4800	5250	2500	5773	2510	30780	3800	3744	1560	82869	16130		
10		11221	490	2457	390	21543	5010	4000	2000	6003	2610	32491	4010	3504	1460	81409	15070		
11		9847	430	2835	450	25929	6030	3759	1790	6693	2910	32886	4060	3024	1260	84973	16930		
12		8931	380	2079	330	30143	7010	3801	1810	7015	3050	28036	3560	2640	1100	83445	17250		
Grand Total		165338	7220	31941	5070	230136	53520	44604	21240	64791	28170	300510	37100	38736	16140	876056	168460		

Pivot Table ile Pivot Chart nesneleri birbiri ile tam uyumlu çalışıyorlar. Öyle ki birinden diğerine tek tıklamayla geçiş yapabiliyorsunuz. Bu iki bileşen aslında görünüm türü olarak kullanılıyor. Access formları ile çalışırken Görünüm menüsünün Tasarım, Normal, Veri Sayfası komutlarının altına Pivot Table ve Pivot Chart eklenmiş.

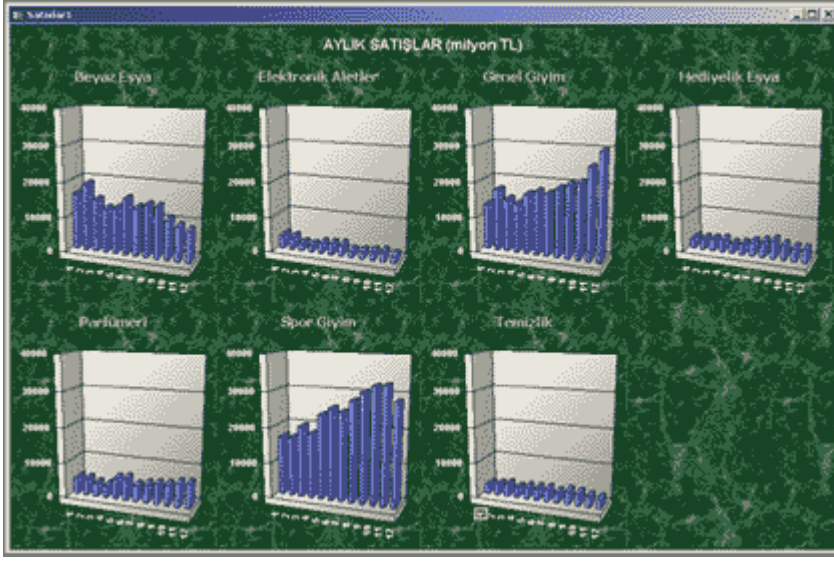


Yazılanlardan bu iki nesnenin sadece Access'e ait olduğunu düşünmeyin. Access'teki Pivot Chart çalışmasını bir anda Excel'e göndermek mümkün.

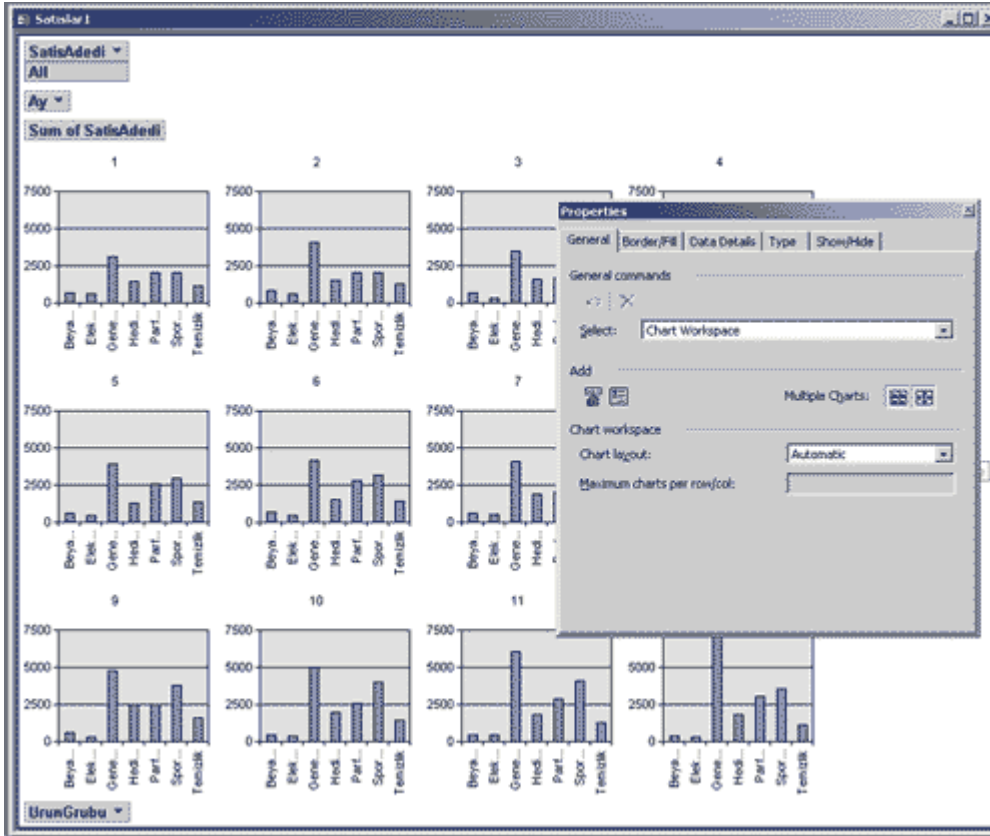


Veri alanlarını bir o yana bir bu yana taşıyabilme dışında Pivot grafiklerindeki Multiple Charts özelliği hoşunuza gidecektir. İstedığınız veri alanına göre çoklu grafik gösterebiliyorsunuz. Aylık satış grafiklerini her ürün grubu için ayrı grafik olacak şekilde

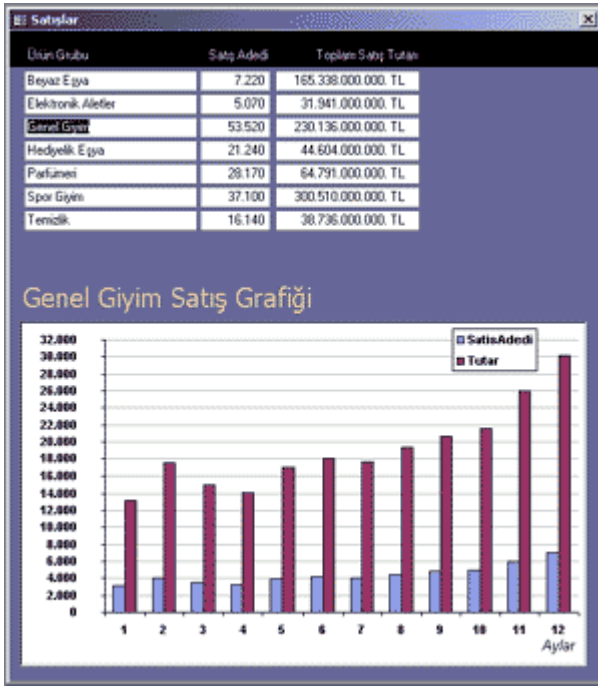
yanyana göstermek mümkün.



Multiple Chart görünümündeyken tek grafik üzerinde yazıtipi, renk gibi özellikler ayarlanıp grafiğin özellikler penceresindeki "Duplicate Chart Format" ile bu grafiğin biçimi tüm grafiklere uygulanabiliyor.



Form ve raporlardaki grafik nesneleri daha geliştirilmiş. Formdaki kayıt listenizde hangi kaydın üzerinde bulunuyorsanız alttaki grafik o kayda göre anında güncelleniyor. Kayıt değişikliklerinde de grafikteki değişim aynen uygulanıyor.



Form üzerinde anında güncellenen akıllı grafikler

AKILLI VERİ SAYFALARI

Access 2000 ile gelen Data Access Pages (Veri Sayfaları) bileşenleri Access 2002'de biraz daha ön plana çıkmış görünümüne göre daha akıllı bu nesnelerle daha güzel ve yetenekli veri sayfaları hazırlayabiliyorsunuz. Bir yanda verileriniz varken hemen yan tarafta akıllı Pivot Chart grafiklerini göstermek mümkün oluyor. Web'e hızlı uyum için form ve raporlar veri sayfaları olarak kaydedilip bu şekilde kullanılabilirler.



Yukarıda belirttiğimiz gibi veri sayfaları (data access pages) ile Internet sayfanızda veri yayınlamak, grafik yayınlamak, veri giriş formları oluşturmak ya da çevrimdışı veri girişi sağlayıp veri değişikliklerini tek seferde kaydetmek mümkün.

İlk denemelerinizde sayfadaki kayıtlarda değişiklik yapılamadığını düşünebilirsiniz. Data Access Page'deki kayıtlarda değişiklik yapılabilmesi için DataPageSize özelliğinin 1 olması ve herhangi bir grup seviyelendirmesinin yapılmaması gerekiyor.

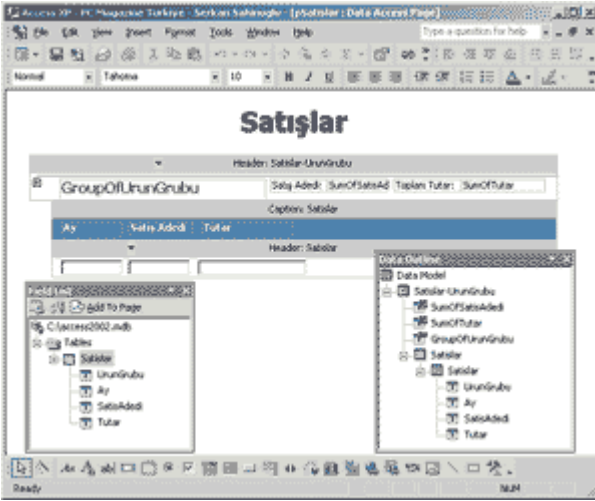
Data Access sayfaları geçerli olarak yerel ağ için tasarlanmış. Sayfanızı Internet'te yayınlamanız için bir yığın ayar gerekiyor. Sonraki yazılarımızda bunu anlatmamı bekleyebilir ya da

<http://support.microsoft.com/support/kb/articles/Q264/0/80.ASP>

adresinden bunun nasıl yapılacağını öğrenebilirsiniz. Bu sayfalarda Access formlarında olan birçok özellik var. DLookup gibi Domain Aggregate Function adı verilen işlevleri, BeforeUpdate, BeforeInsert, AfterInsert gibi olay yordamlarını kullanabiliyorsunuz. Fakat bunların bir kısmı sadece Access içinde iken geçerlidir.

Bu sayfalar ayrıca SQL Server 2000 ile tam uyumlu çalışabiliyorlar. Artık kullanıcının sayfanın görünüşünde daha fazla düzenleme yapılmasına olanak sağlanmış. Aynı şekilde veri alanları için kolayca alt toplam gibi bilgiler görüntülenebiliyor.

Sayfa tasarımında yeni Data Outline bileşeni bağlı veritabanından kolayca sorgu nesneleri oluşturabilmemize yardımcı oluyor.



Data Access Page nesneleri Access'teki Page görünümünde ya da Internet Explorer'ın 5.0 ve daha yeni sürümlerinde görüntülenebilirler.

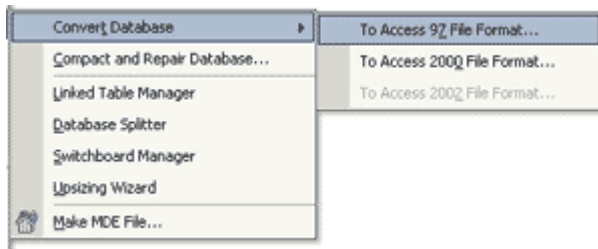
YENİ ÖZELLİKLER

Yeni eklentilerin dışında sıkça kullandığımız bazı bölümler biraz daha kullanışlı hale getirilmiş. Ayrıca altyapı ile ilgili değişiklikler söz konusu.

Öncelikle Access 2002'nin kendine ait bir dosya yapısı var. Tabii eski sürüme de tam destek amaçlanmış. Access 2000 dosyası Access 2002 üzerinde açılabilir ve çoğunlukla sorunsuz kullanılabilir. Ürün belgelerinde Access 97'ye fazla destek verilmediği söylene de elimdeki sürüm ile Access 97 ile yapılmış programları birkaç hata mesajı sonrasında çalıştırabildim. Access 97 dosyasını açarken 2002 biçimine dönüştürmek de mümkün oldu. Hiç dönüştürme işlemi olmadan Access 97 dosyasındaki bir tabloyu Access 2002 dosyasına bağlayıp (Link) sorunsuz kullanabildim.

İlk kurulumdan sonra Access XP'de yeni bir dosya oluşturduğunuzda geçerli olarak eski biçimde, yani Access 2000 türünde dosya oluşturuyor. Tools / Options / Advanced sayfasındaki Default File Format seçeneğini değiştirdiğiniz takdirde yeni dosyalarınızı Access 2002 türünde oluşturabiliyorsunuz.

Access 2002 biçimindeki dosyayı Tools / Convert Database menüsü ile Access 97 ya da Access 2000 biçimine dönüştürebiliyorsunuz. Ayrıca eski sürümlerden dönüştürme işlemi yaparken oluşan hataları bir tablonun içine kaydediyor (Conversion Errors). Böylelikle hata giderme aşamalarında zaman kazanılıyor.



Tablo bağlama özelliğinin yetenekleri artırılmış. Dosya açma penceresinden DBase, Paradox, Excel dosyası ya da metin dosyalarını açmak istediğinizde bu dosyalara otomatik bağlantı (link) oluşturuluyor.

Eski sürüme tam destek verildiği söylene de bu, önceki Access'le tam uyumlu çalışabileceğiniz anlamına gelmiyor. Yeni Access'i başka bir klasöre kurduğumda eskisine pek dokunmasa da Access 2000 bazı yeteneklerini kaybetti. Örneğin Linked Table Manager ve diğer bazı sihirbazlar artık çalışmıyorlar. Ayrıca Microsoft Access 9.0 Object Library de kurulum esnasında kaldırılmış.

İki sürüm birlikte kurulu iken ikisi arasında Add-in'ler konusunda anlaşmazlık çıktı. Eski sürümde bir Add-in yüklü iken yenisinde de bunu yüklemeye çalıştığımda onu yüklüyümüş gibi gösterdi ama seçtiğim halde Add-in'ler menüsüne getiremedi.

HIZLI VE BECERİKLİ

Yeni bir dosya oluştururken bunu, var olan başka bir dosyadan türetmeniz mümkün. Task Pane'nin New File kısmında New From Existing File ile var olan bir dosyayı şablon olarak kullanıp yeni dosyanızı içi dolu bir şekilde oluşturmuş oluyorsunuz.

Önemli değişikliklerden bir diğeri de artık tasarım anında birden fazla geri alma işleminin yapılabilmesi. Zira daha önce sadece bir kez geri alma işlemi yapılabiliyordu.

Yine form ve raporların olay yordamı tanımlama seçenekleri (Events) artırılmış. Tasarım için bazı kolaylıklar getirilmiş. Buradaki form ve raporların yerleştirdiğiniz nesneleri hassas boyutlandırmaya çalıştığınızda bu özelliğin artık olmadığını düşünebilirsiniz. Hassas boyutlandırma özelliği yine var ama eskisinde Shift basılı tutulup oklar kullanılırken, şimdi Ctrl ve Shift tuşlarını beraber basılı tutmak gerekiyor.

Tablo tasarımında Allow Zero Length (Sıfır Uzunluk İzni) değeri artık geçerli olarak True (Evet) geliyor. Zaten bu gereklilik daha önceki seferinde bunu False'tan True'ya çevirmek bana zor gelmeye başlamıştı.

Yazdırma işleriyle fazla uğraşan programcılar için, yazıcı ayarlarının artık değiştirilebiliyor olması sevindirici bir haber olacaktır. Printer ve Printers koleksiyonları ile yazdırma işlemlerinizi yönetebiliyorsunuz. Bir formu yazdırmadan önce kenar boşluğu ayarları dahil yazıcı ekranında gördüğünüz birçok ayarı kod ile yapabiliyorsunuz. (Form.Printer)

Ayrıca raporların Baskı Önizleme penceresindeki yakınlaştırma seçeneklerine %500 ve %1000 değerleri eklenmiş. Eskiden en fazla %200 yakınlaştırma yapılabiliyordu bildiğiniz gibi.

Bul/Değiştir ekranında boş (Null) değerlerin aranabiliyor olması, formlarda olduğu gibi raporlara OpenArgs özelliğinin eklenmesi gibi bazı değişiklikler de var.

SPESİFİKASYONLAR

Yeni Access dosyalarında eskisine nazaran altyapıda çok az değişiklik var.

- Veritabanı dosyası büyüklüğü aynı (2 gb.) kalacak şekilde bir tabloda saklanabilecek veri büyüklüğü sınırı 1 gigabayttan 2 gigabayta çıkarılmış.
- Sorgu metinlerindeki koşul ifadeleri sayısı 40 iken 99 sınırına çıkarılmış.
- İç içe kullanılabilen form veya rapor adedi 3'ten 7'ye çıkarılmış.

XP'nin proje dosyasındaki (ADP) öznitelikler, veritabanı dosyasınıninkilerle (MDB) genel olarak aynı. Tek fark XP'nin MDB veritabanı tablosunda en fazla 255 veri alanı olabilirken ADP proje dosyasına SQL Server 6.5 için 255, SQL Server 7.0 ve 2000 için 1024 adet veri alanı oluşturulabiliyor olması. (ek: [Access XP ile Access 2000 Spesifikasyonları](#))

SONUÇ

Zaten becerikli ve lüks bir masaüstü uygulaması olan Office, XP sürümü ile yeteneklerini ve lüksünü daha fazla artırmış bulunuyor. Bu yeteneklerin ve lüksün artırılması ve getirilen yeni bileşenler, yeni Access'e XP soyadını hakettiriyor.

İlk açıldığında sadece ekran tasarımı değişikliğinden ibaret gibi görünse de kullanıldıkça yenilikler farkediliyor. Ama birçok bölümde zaten hiçbir değişiklik yapılmamış. Kalite açısından belli bir doygunluk düzeyine ulaşan Access'in yeni bir sürümü bazı bölümlerde çok fazla değişiklik beklememek daha doğru olur.

Kullandıkça sizin de farkedeceğiniz şeylerden biri, uzakdoğu dil desteğinin birçok yerde kendini hissettirmesi. Tablo, form ve nesneleri tasarlarlarken bununla ilgili ayarlarla karşılaşılıyor.

Gelişmiş yeni çözümleriyle Access XP, hızlı geliştirme ortamını ve yetenekli veritabanı ve Internet uygulamaları geliştirme gücünü bir arada sunuyor.

EK: [Access XP ile Access 2000 Spesifikasyonları](#)

- Access XP Özel -

Access XP ile Yeni Deneyimler

Yeni Access beraberinde neler getiriyor? Access XP'nin derinliklerine inip, yenilikleri ve yeni Access'le yapabileceklerimizi araştırdık.



Nihayet piyasaya çıkan Office XP ile birlikte Access de XP sürümü ile karşımıza çıkıyor. Öncekine nazaran görünürde çok fazla değişiklik içermeyip, yeteneklerin artırılması ve sunulan yeni bileşenler, görünüşüne aldandığım yeni Access konusundaki ilk fikirlerimi yavaş yavaş değiştirdi.

Bununla birlikte Office'ten aldığı özellikler ve beraberinde sunulan diğer araçlar, önceki Access'te bazı eksikliklerin olduğunu hissettiriyor.

Access XP ya da diğer adıyla Access 2002'ye derinlemesine bir dalış yapıyor, getirilen yenilikleri, yapılan değişiklikleri ve yeni Access ile yapabileceklerimizi inceliyoruz.

NELER YENİ

Access 2002 yeni Office'in Professional ve Developer adlı sürümleri içinde yer alıyor ve ayrı kutu olarak satılıyor. Access'te yapılan programları sınırsız dağıtmak için kullanılan Office Developer Tools ise tek satılmayıp sadece Office Developer paketi içinde geliyor.

Access 2002'nin masaüstü uygulamaları alanında üç ana hedefi olduğu belirtiliyor:

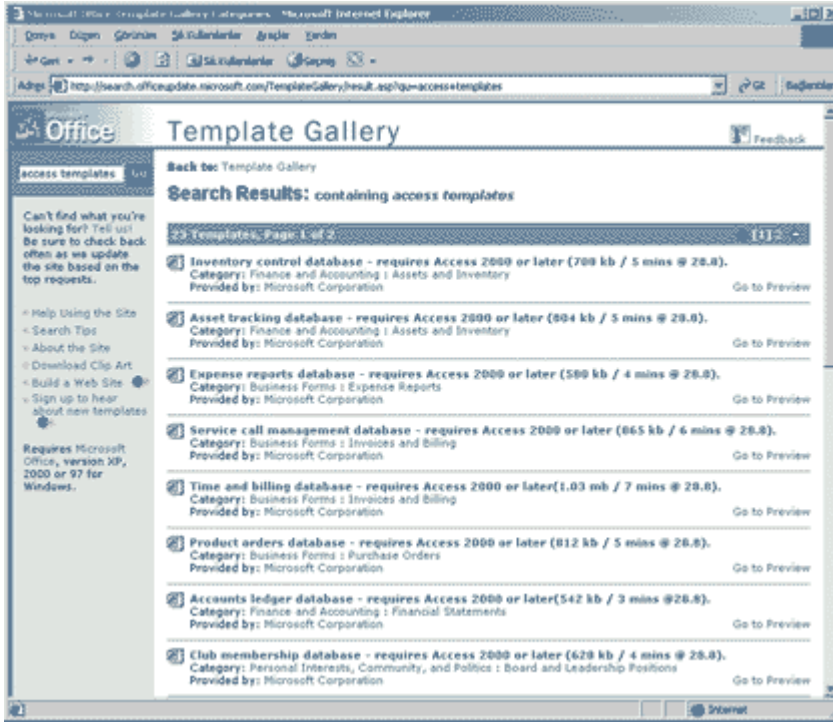
- * Web tabanlı erişim, analiz ve raporlama araçları
- * Birlikte çalışma ve paylaşım desteği
- * Hızlı geliştirme ve yönetim

Eski sürümle uyumlu çalışabilen Access XP, Web ağına biraz daha kök salmış görünüyor. XML desteği ile birlikte dinamik sayfalar sunabilen bu yeni sürüm, Internet ve yerel ağ ortamında veri paylaşım ve sunum olanağını üst düzeylere çıkarıyor. Veri sayfalarının (Data Access Pages) yetenekleri geliştirilerek daha kullanışlı web formlarının ve akıllı web grafiklerinin oluşturulmasına olanak sağlanıyor.

SQL Server uyumluluğu daha artırılmış, veri sayfalarıyla SQL Server verilerinin çevrimdışı olarak kullanılabilmesi sağlanmış. Çevrimiçi özellikleriyle canlı veriler ve canlı grafikler sunmak artık mümkün oluyor.

Yeni Pivot Table ve Pivot Chart adlı bileşenler sorgulama ve raporlama işlerimize farklı bir boyut getiriyor. Piyasadaki diğer bazı raporlama araçlarında önceden beri yer alan Pivot nesneleri artık Access'te en ön sıralarda yerlerini alıyorlar.

Office'in bu sürümüyle birlikte, çok yararlı olabilecek kaynakların, dokümanların, şablonların, resim galerilerinin bulunduğu Web servisleri oluşturulmuş. Access'te ofisinizde kullanılacak bir program yapmak istiyorsanız Office XP'nin Web sitesindeki Template Gallery'de bulabileceğiniz, çalışır halde onlarca şablon dosyası var.



YENİ PIVOT BİLEŞENLERİ VE CANLI GRAFİKLER

Access'teki Crosstab Query (Çarpraz Sorgu) nesnelerini kullanmış olanlar Pivot Table'a çabuk alışacaktır. Adında Table kelimesi geçse de bu nesne formlar bölümünde yer alıyor. Form sihirbazı ile yeni bir Pivot Table oluşturulabiliyor. İlk aşamada hangi tablonuz üzerinde sorgu işlemi yapacağınızı seçiyorsunuz. Sonrasında veri alanlarını bir aşağı, bir yukarı taşıyarak alanları birbiri ile istediğiniz şekilde karşılaştırıp gruplayabiliyor, süzebiliyor, genel toplamaları, adetleri anında görebiliyorsunuz.

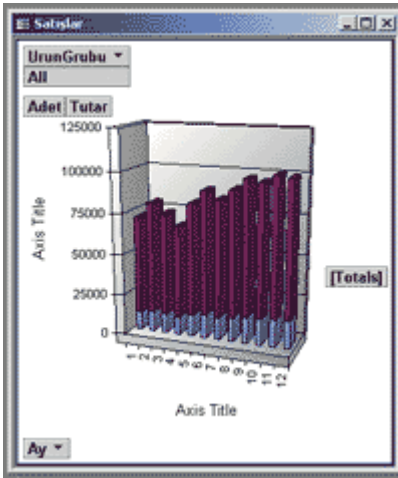
Satışlar			
Drop Filter Fields Here			
UrunGrubu	Ay	Adet	Tutar
Beyaz Eşya	1	710	16259
	2	840	19236
	3	650	14885
	4	550	12595
	5	570	13053
	6	700	16030
	7	600	13740
	8	650	14885
	9	640	14656
	10	490	11221
	11	430	9847
	12	390	8931
Grand Total		7220	165338

Satışlar			
Drop Filter Fields Here			
		Drop Column Fields Here	
UrunGrubu	Ay	Tutar	Adet
Beyaz Eşya	1	16259	710
	2	19236	840
	3	14885	650
	4	12595	550
	5	13053	570
	6	16030	700
	7	13740	600
	8	14885	650
	9	14656	640
	10	11221	490
	11	9847	430
	12	8931	390
Total		165338	7220
Elektronik Aletler		31941	5070
Genel Giyim		230136	53520
Hediyelik Eşya		44604	21240
Parfümleri		64791	28170
Spor Giyim	1	16767	2070
	2	16848	2080
	3	20250	2500
	4	18387	2270
	5	23895	2950
	6	25920	3200
	7	25110	3100
	8	28350	3500
	9	30780	3800
	10	32481	4010

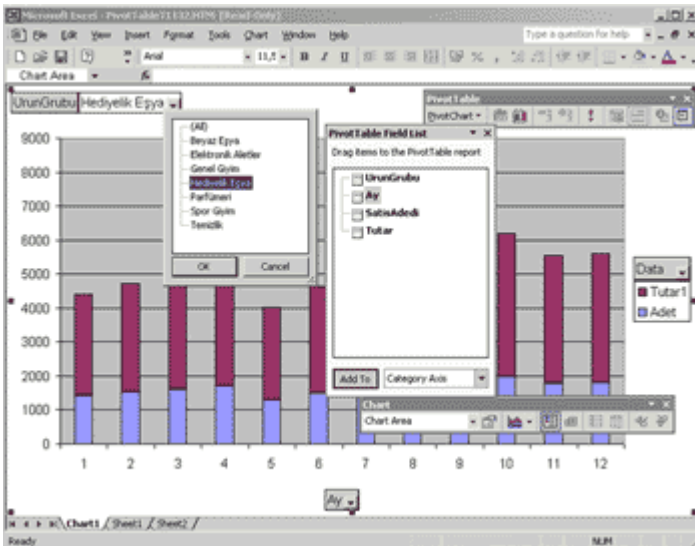
Üst taraftaki seçenek yardımıyla sadece Beyaz Eşya ile ilgili bilgiler listelendi.

Drop Filter Fields Here																		
UrunGrubu ▾																		
Bazay Eyya		Elektronik Aletler				Genel Ciyim		Hediyelek Eyya		Parkman		Spor Ciyim		Ternizlik		Grand Total		
Ay	Tutar	Adet	Tutar	Adet	Tutar	Adet	Tutar	Adet	Tutar	Adet	Tutar	Adet	Tutar	Adet	Tutar	Adet	Tutar	Adet
1	16259	710	3843	610	13201	3070	2062	1420	4092	2040	16767	2070	2760	1190	60504	11070		
2	19236	840	3654	580	17544	4080	3192	1520	4784	2080	16848	2080	3120	1300	68378	12480		
3	14895	650	2142	340	16060	3600	3360	1600	3087	1690	20290	2500	3600	1500	63174	11780		
4	12595	550	1890	300	14061	3270	3570	1700	3542	1540	18387	2270	2880	1200	56925	10630		
5	13053	570	2096	460	16905	3950	2730	1300	6003	2610	23696	2950	3360	1400	68524	13240		
6	16030	700	3087	490	18060	4200	3150	1500	6440	2800	25920	3200	3480	1450	76167	14340		
7	13740	600	3276	520	17630	4100	3990	1900	4646	2020	25110	3100	3624	1510	72016	13750		
8	14895	650	1764	280	19350	4500	4620	2200	5313	2310	28390	3500	3000	1250	77262	14690		
9	14656	640	2016	320	20640	4800	5250	2500	5773	2510	30780	3800	3744	1560	82899	16130		
10	11221	490	2457	390	21543	6010	4200	2000	6003	2610	32491	4010	3504	1460	81409	15970		
11	9647	430	2835	450	25929	6030	3759	1790	6693	2910	32886	4060	3024	1260	84873	16930		
12	8931	390	2079	330	30143	7010	3801	1810	7015	3090	28836	3560	2640	1100	83445	17250		
Grand Total	165338	7220	31941	5070	230136	53520	44604	21240	64791	28170	300510	37100	38736	16140	876056	168460		

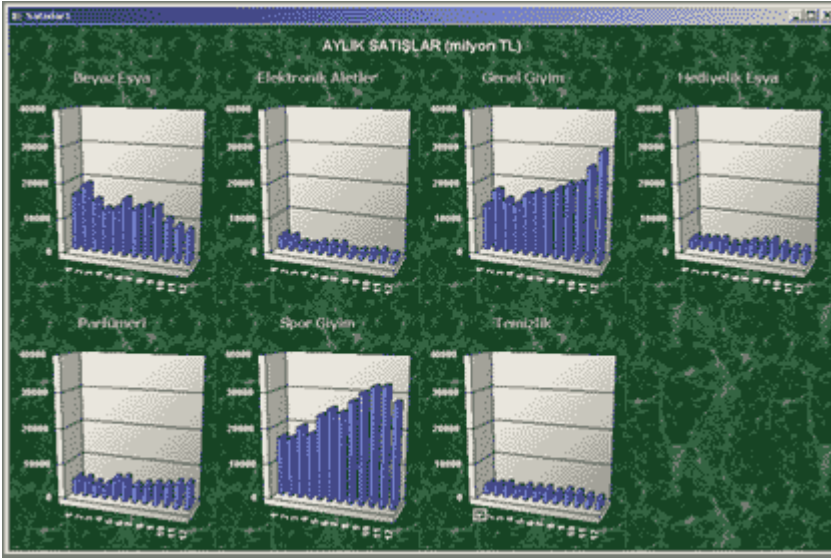
Pivot Table ile Pivot Chart nesneleri birbiri ile tam uyumlu çalışıyorlar. Öyle ki birinden diğerine tek tıklamayla geçiş yapabiliyorsunuz. Bu iki bileşen aslında görünüm türü olarak kullanılıyor. Access formları ile çalışırken Görünüm menüsündeki Tasarım, Normal, Veri Sayfası komutlarının altına Pivot Table ve Pivot Chart eklenmiş.



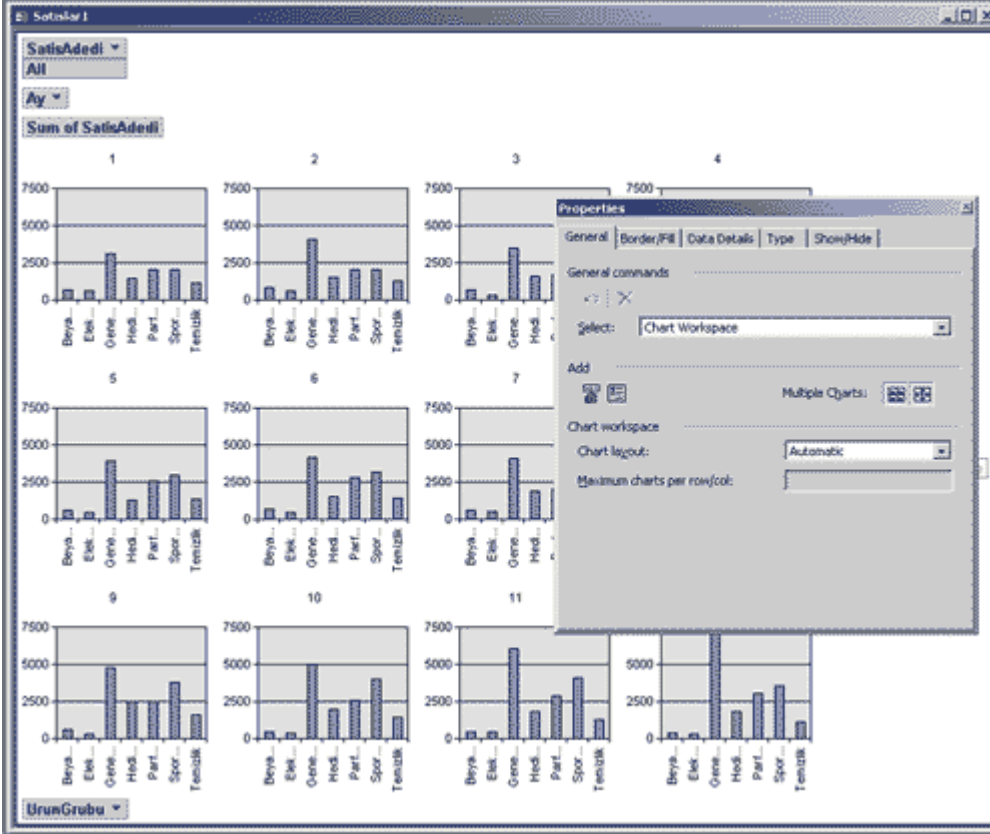
Yazılanlardan bu iki nesnenin sadece Access'e ait olduğunu düşünmeyin. Access'teki Pivot Chart çalışmasını bir anda Excel'e göndermek mümkün.



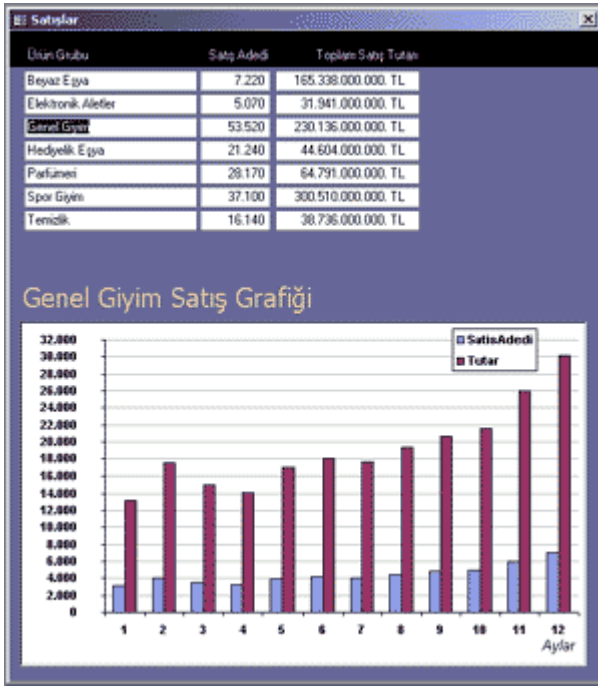
Veri alanlarını bir o yana bir bu yana taşıyabilme dışında Pivot grafiklerindeki Multiple Charts özelliği hoşunuza gidecektir. İstediğiniz veri alanına göre çoklu grafik gösterebiliyorsunuz. Aylık satış grafiklerini her ürün grubu için ayrı grafik olacak şekilde yanyana göstermek mümkün.



Multiple Chart görünümündeyken tek grafik üzerinde yazı tipi, renk gibi özellikler ayarlanıp grafiğin özellikler penceresindeki "Duplicate Chart Format" ile bu grafiğin biçimi tüm grafiklere uygulanabiliyor.



Form ve raporlardaki grafik nesneleri daha geliştirilmiş. Formdaki kayıt listenizde hangi kaydın üzerinde bulunuyorsanız alttaki grafik o kayda göre anında güncelleniyor. Kayıt değişikliklerinde de grafikteki değişim aynen uygulanıyor.



Form üzerinde anında güncellenen akıllı grafikler

AKILLI VERİ SAYFALARI

Access 2000 ile gelen Data Access Pages (Veri Sayfaları) bileşenleri Access 2002'de biraz daha ön plana çıkmış görünüyor. Eskisine göre daha akıllı bu nesnelerle daha güzel ve yetenekli veri sayfaları hazırlayabiliyorsunuz. Bir yanda verileriniz varken hemen yan tarafta akıllı Pivot Chart grafiklerini göstermek mümkün oluyor. Web'e hızlı uyum için form ve raporlar veri sayfası olarak kaydedilip bu şekilde kullanılabilirler.

Satışlar		
Hediyelik Eşya Satış Adedi: 21.240 Toplam Tutarı: 44.604 milyon TL.		
Parfümeri Satış Adedi: 28.170 Toplam Tutarı: 64.791 milyon TL.		
Elektronik Aletler Satış Adedi: 5.070 Toplam Tutarı: 31.941 milyon TL.		
Ay	Satış Adedi	Tutar
1	600	3843 milyon TL.
2	500	3604 milyon TL.
3	340	2142 milyon TL.
4	300	1890 milyon TL.
5	460	2996 milyon TL.
6	490	3067 milyon TL.
7	520	3276 milyon TL.
8	280	1764 milyon TL.
9	320	2016 milyon TL.
10	390	2457 milyon TL.
11	460	2836 milyon TL.
12	300	1879 milyon TL.
Genel Giyim Satış Adedi: 53.520 Toplam Tutarı: 230.136 milyon TL.		
Temizlik Satış Adedi: 16.140 Toplam Tutarı: 38.736 milyon TL.		
Spor Giyim Satış Adedi: 37.100 Toplam Tutarı: 300.510 milyon TL.		
Beyaz Eşya Satış Adedi: 7.220 Toplam Tutarı: 165.338 milyon TL.		

Yukarıda belirttiğimiz gibi veri sayfaları (data access pages) ile Internet sayfanızda veri yayınlamak, grafik yayınlamak, veri girişi formları oluşturmak ya da çevrimdışı veri girişi sağlayıp veri değişikliklerini tek seferde kaydetmek mümkün.

İlk denemelerinizde sayfadaki kayıtlarda değişiklik yapılamadığını düşünebilirsiniz. Data Access Page'deki kayıtlarda değişiklik yapılabilmesi için DataPageSize özelliğinin 1 olması ve herhangi bir grup seviyelendirmesinin yapılmaması gerekiyor.

Data Access sayfaları geçerli olarak yerel ağ için tasarlanmıştır. Sayfanızı Internet'te yayınlamanız için bir yığın ayar gerekiyor. Sonraki yazılarımda bunu anlatmamı bekleyebilir ya da

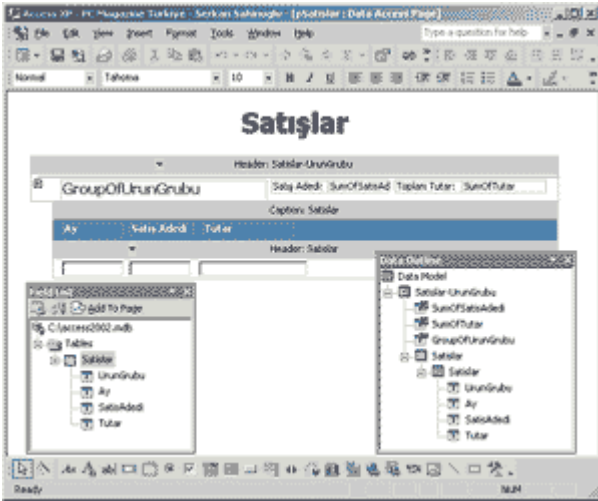
<http://support.microsoft.com/support/kb/articles/Q264/0/80.ASP>

adresinden bunun nasıl yapılacağını öğrenebilirsiniz. Bu sayfalarda Access formlarında olan birçok özellik var. DLookup gibi Domain Aggregate Function adı verilen işlevleri, BeforeUpdate, BeforeInsert, AfterInsert gibi olay yordamlarını kullanabiliyorsunuz. Fakat bunların bir kısmı sadece Access içinde iken geçerlidir.

Bu sayfalar ayrıca SQL Server 2000 ile tam uyumlu çalışabiliyorlar. Artık kullanıcının sayfanın görünüşünde daha fazla düzenleme yapılmasına olanak sağlanmış. Aynı şekilde veri alanları için kolayca alt toplam gibi bilgiler görüntülenebiliyor.

Sayfa tasarımında yeni Data Outline bileşeni bağlı veritabanından kolayca sorgu nesneleri oluşturabilmemize

yardımcı oluyor.



Data Access Page nesneleri Access'teki Page görünümünde ya da Internet Explorer'ın 5.0 ve daha yeni sürümlerinde görüntülenebiliyorlar.

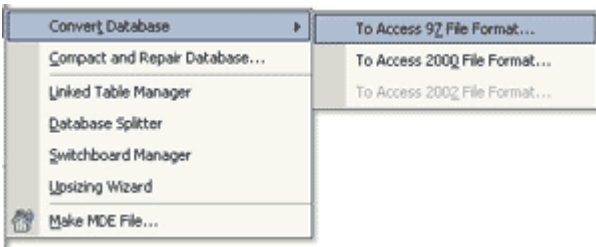
YENİ ÖZELLİKLER

Yeni eklentilerin dışında sıkça kullandığımız bazı bölümler biraz daha kullanışlı hale getirilmiş. Ayrıca altyapı ile ilgili değişiklikler söz konusu.

Öncelikle Access 2002'nin kendine ait bir dosya yapısı var. Tabii eski sürüme de tam destek amaçlanmış. Access 2000 dosyası Access 2002 üzerinde açılabilir ve çoğunlukla sorunsuz kullanılabilir. Ürün belgelerinde Access 97'ye fazla destek verilmediği söylense de elimdeki sürüm ile Access 97 ile yapılmış programları birkaç hata mesajı sonrasında çalıştırabildim. Access 97 dosyasını açarken 2002 biçimine dönüştürmek de mümkün oldu. Hiç dönüştürme işlemi olmadan Access 97 dosyasındaki bir tabloyu Access 2002 dosyasına bağlayıp (Link) sorunsuz kullanabildim.

İlk kurulumdan sonra Access XP'de yeni bir dosya oluşturduğunuzda geçerli olarak eski biçimde, yani Access 2000 türünde dosya oluşturuyor. Tools / Options / Advanced sayfasındaki Default File Format seçeneğini değiştirdiğiniz takdirde yeni dosyalarınızı Access 2002 türünde oluşturabiliyorsunuz.

Access 2002 biçimindeki dosyayı Tools / Convert Database menüsü ile Access 97 ya da Access 2000 biçimine dönüştürebiliyorsunuz. Ayrıca eski sürümlerden dönüştürme işlemi yaparken oluşan hataları bir tablonun içine kaydediyor (Conversion Errors). Böylelikle hata giderme aşamalarında zaman kazanılıyor.



Tablo bağlama özelliğinin yetenekleri artırılmış. Dosya açma penceresinden DBase, Paradox, Excel dosyası ya da metin dosyalarını açmak istediğinizde bu dosyalara otomatik bağlantı (link) oluşturuluyor.

Eski sürüme tam destek verildiği söylense de bu, önceki Access'le tam uyumlu çalışabileceğiniz anlamına gelmiyor. Yeni Access'i başka bir klasöre kurduğumda eskisine pek dokunmasa da Access 2000 bazı yeteneklerini kaybetti. Örneğin Linked Table Manager ve diğer bazı sihirbazlar artık çalışmıyorlar. Ayrıca Microsoft Access 9.0

Object Library de kurulum esnasında kaldırılmış.

İki sürüm birlikte kurulu iken ikisi arasında Add-in'ler konusunda anlaşmazlık çıktı. Eski sürümde bir Add-in yüklü iken yenisinde de bunu yüklemeye çalıştığımda onu yüklüymüş gibi gösterdi ama seçtiğim halde Add-in'ler menüsüne getiremedi.

HIZLI VE BECERİKLİ

Yeni bir dosya oluştururken bunu, var olan başka bir dosyadan türetmeniz mümkün. Task Pane'nin New File kısmında New From Existing File ile var olan bir dosyayı şablon olarak kullanıp yeni dosyanızı içi dolu bir şekilde oluşturmuş oluyorsunuz.

Önemli değişikliklerden bir diğeri de artık tasarım anında birden fazla geri alma işleminin yapılabilmesi. Zira daha önce sadece bir kez geri alma işlemi yapılabiliyordu.

Yine form ve raporların olay yordamı tanımlama seçenekleri (Events) artırılmış. Tasarım için bazı kolaylıklar getirilmiş. Bunun yanında forma yerleştirdiğiniz nesneleri hassas boyutlandırmaya çalıştığınızda bu özelliğin artık olmadığını düşünebilirsiniz. Hassas boyutlandırma özelliği yine var ama eskisinde Shift basılı tutulup oklar kullanılırken, şimdi Ctrl ve Shift tuşlarını beraber basılı tutmak gerekiyor.

Tablo tasarımında Allow Zero Length (Sıfır Uzunluk İzni) değeri artık geçerli olarak True (Evet) geliyor. Zaten bu gerekliydi. Her seferinde bunu False'tan True'ya çevirmek bana zor gelmeye başlamıştı.

Yazdırma işleriyle fazla uğraşan programcılar için, yazıcı ayarlarının artık değiştirilebiliyor olması sevindirici bir haber olacaktır. Printer ve Printers koleksiyonları ile yazdırma işlemlerinizi yönetebiliyorsunuz. Bir formu yazdırmadan önce kenar boşluğu ayaları dahil yazıcı ekranında gördüğünüz birçok ayarı kod ile yapabiliyorsunuz. (Form.Printer)

Ayrıca raporların Baskı Önizleme penceresindeki yakınlaştırma seçeneklerine %500 ve %1000 değerleri eklenmiş. Eskisinde en fazla %200 yakınlaştırma yapılabiliyordu bildiğiniz gibi.

Bul/Değiştir ekranında boş (Null) değerlerin aranabiliyor olması, formlarda olduğu gibi raporlara OpenArgs özelliğinin eklenmesi gibi bazı değişiklikler de var.

SPEŞİFİKASYONLAR

Yeni Access dosyalarında eskisine nazaran altyapıda çok az değişiklik var.

- Veritabanı dosyası büyüklüğü aynı (2 gb.) kalacak şekilde bir tabloda saklanabilecek veri büyüklüğü sınırı 1 gigabayttan 2 gigabayta çıkarılmış.
- Sorgu metinlerindeki koşul ifadeleri sayısı 40 iken 99 sınırına çıkarılmış.
- İç içe kullanılabilen form veya rapor adedi 3'ten 7'ye çıkarılmış.

XP'nin proje dosyasındaki (ADP) öznitelikler, veritabanı dosyasınınkilerle (MDB) genel olarak aynı. Tek fark XP'nin MDB veritabanı tablosunda en fazla 255 veri alanı olabilirken ADP proje dosyasına SQL Server 6.5 için 255, SQL Server 7.0 ve 2000 için 1024 adet veri alanı oluşturulabiliyor olması. (ek: [Access XP ile Access 2000 Spesifikasyonları](#))

SONUÇ

Zaten becerikli ve lüks bir masaüstü uygulaması olan Office, XP sürümü ile yeteneklerini ve lüksünü daha fazla artırmış bulunuyor. Bu yeteneklerin ve lüksün artırılması ve getirilen yeni bileşenler, yeni Access'e XP soyadını hakettiriyor.

İlk açıldığında sadece ekran tasarımı değişikliğinden ibaret gibi görünse de kullanıldıkça yenilikler farkediliyor. Ama

birçok bölümde zaten hiçbir değişiklik yapılmamış. Kalite açısından belli bir doygunluk düzeyine ulaşan Access'in yeni bir sürümünde, bazı bölümlerde çok fazla değişiklik beklememek daha doğru olur.

Kullandıkça sizin de farkedeceğiniz şeylerden biri, uzakdoğu dil desteğinin birçok yerde kendini hissettirmesi. Tablo, form gibi nesneleri tasarlarırken bununla ilgili ayarlarla karşılaşılıyor.

Gelişmiş yeni çözümleriyle Access XP, hızlı geliştirme ortamını ve yetenekli veritabanı ve Internet uygulamaları geliştirme gücünü bir arada sunuyor.

EK: [Access XP ile Access 2000 Spesifikasyonları](#)

YARDIMCI İŞLEVLER

Bu ay, dokunulmazlığı olan problemlere dokunup gizlenmiş ipuçlarına kalem savurduk. Müzmin sorunlara uzman çözümleri Access Hikayeleri'nde.

Geçen aylar her zamanki gibi yorucuydu. Access sayfalarının fazla olması birçok okurumuzun isteğini karşılamış olsa da benim açımdan yorulmayla birlikte bazı eksiklikleri de beraberinde getirdi.

Access XP yazısından bir önceki yazıyı, yayınlandıktan sonra gözden geçirdiğimde programların dağıtımıyla ilgili karamsar bir tablo çizdiğimi farkettilim. Access dosyalarına verilen parolaların Internet'te bulunabilen bazı programlar tarafından çözülebildiğinden bahsetmiştim. Bunu şimdi biraz aydınlatmak gerektiğini düşünüyorum.

Dosyanıza parola verdiğinizde bu parola sadece dosyanızın yetkili açılabilir ya da kullanılabilir olmasını sağlayacaktır. MDE'den MDB'ye dönüştürme gibi bir uygulama hiç varolmayıp parola korumalı MDE dosyaları da açılıp çalıştırılabilir ve MDB'deki parola mantığı bunda da aynen geçerlidir. MDE dosyalarında formlar, raporlar, kodlar görüntülenemez ve değiştirilemez. Tablo ve sorgular ise değiştirilebilir.

Veritabanı için tanımlanan parola koruması sadece dosyanın yetkili açılmasını sağlayabilir. Sonrası için ise güvenlik ayarlarının tanımlandığı bölümde kullanıcı ve grup yetkilendirmesinin yapılması yeterli olacaktır.

YENİ İHTİYAÇLARA YENİ ÇÖZÜMLER

Bu ay yeni hazırladığım yardımcı işlevlerin sizin de işinize yarayacağından eminim. Sadece ihtiyacım olduğu zamanlarda yaptığım üzere geçen ay çalıştığım bir proje için bazı işlevler yazmak durumunda oldum.

Tasarladığım formlar yüzlerce veri ve formül alanı içeriyordu. Sabitler ve hesaplanan formüller sorgularda işlenip formda gösteriliyor. Ama müşteriye lazım olan yeni formülleri de forma yerleştirmek gerekiyor. Onlarca form ve her bir formda ortalama 300 adet sabit ve formül alanları. Bu durumda bir süre sonra sorgulardaki bazı alanlar gözden kaçıp forma yerleştirilmeyebilir.

Bunun için NotUsedFields adlı bir işlev yazdım. Bu işlev formun kayıt kaynağındaki tüm veri alanlarının formda kullanılıp kullanılmadığını buluyor. Böylelikle yüzlerce kutu arasında kaybolmuyorum. Aynı şekilde iptal edilmiş veri alanlarının da form üzerinde hala varolup olmadığının kontrol edilmesi gerekiyordu. SelectInvalidFieldObjects işlevi de bu işi yapıyor.

Hazır başlamışken bir veri alanını arama zahmetinden kurtarıp form üzerinde doğrudan gösteren FindField, tab duraklarını ayarlarken büyük kolaylıklar sağlayan SetTabOrder ve benim en çok hoşuma giden SelectControls işlevlerini yazdım. Bununla form üzerindeki bütün metin kutularını, etiketleri, düğmeleri ve hemen tüm diğer nesneleri bir anda seçmek mümkün. Shift tuşu basılı olduğunda da buna uyup eski seçimleri kaldırıyor.

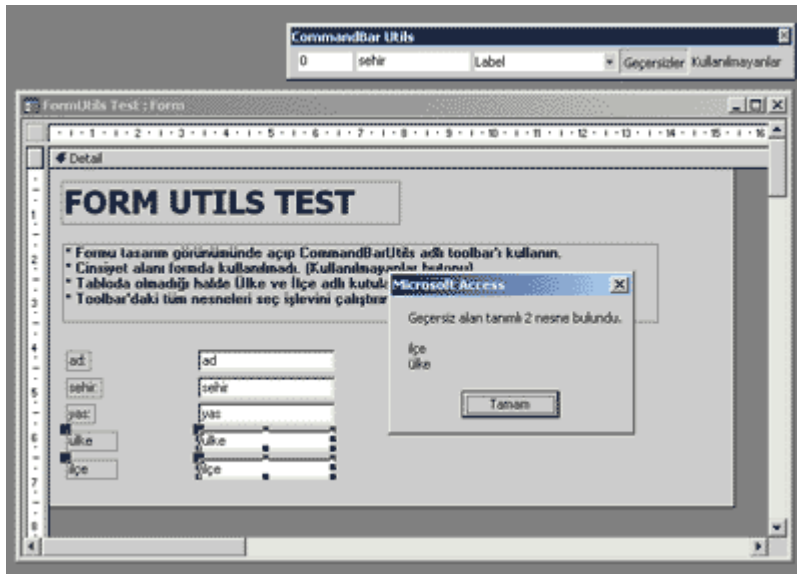
Bu işlevlerin bazılarında çalışma kipinde olup olmadığımızı tespit etmek gerekiyordu. Form nesnesinin CurrentView özelliği bunu bize veriyor. Sıfır ise tasarım kipinde demektir. 1 olduğunda ise çalışma anı demektir.

Yaptığım projede herbiri ortalama 500 nesne içeren 20 civarında form ve rapor bulunuyor. Tablo ve sorguların herbiri de ortalama 200 veri alanı içeriyor.

Formüllerin anlık olarak hesaplanması için en iyi yol Excel'deki gibi bağlantılı formül alanları kullanmak. Access'te bunları yapmak Excel'deki kadar kolay sayılır. Ekran üzerinde bir değeri değiştirdiğinizde buna bağlı onlarca formülü yeniden hesaplayabiliyorsunuz. Denetim kaynağı özelliğine bir formül yazmanız yeterli. Fakat formlarda uğraşmak yerine formülleri sorgularda yapmayı yeğledim.

Sorgularda formül tanımladığınızda Excel sayfalarında olduğu gibi bir değer değiştiğinde aynı satırda buna bağlı tüm değerler de anında değişebiliyor. Bunu yapmamın sebebi sorgu nesnelerini arkaplanda açıp grafikler için bazı değerleri seri halinde değiştirip sonuçlar elde etmektir. Bu iş için en iyi yol buydu.

Projemde kullandığım bazı teknikleri bu ay sizlerle paylaşıyorum.



MENÜ ARAÇ ÇUBUKLARINI PROGRAMLAYIN

Bu işlevleri kullanışlı hale getirmek için araç çubuklarına yerleştirmek gerekiyordu. Formdaki nesnelere tab numarası vermek için seçili objelerin başlangıç tab sırasını yazabileceğimiz bir kutuya ihtiyaç vardı. Toolbar'da oluşturduğum Utils barına msoControlEdit türünde bir CommandBarControl nesnesi yerleştirdim. Türüne göre tüm nesneleri seçebilmek için açılır liste kutusu (msoControlDropdown) gerekiyordu. Geçersiz ve kullanılmayan veri alanları için de birer CommandBarButton yeterliydi.



Fakat bir sorun komutların uygulanacağı formun belirlenmesiydi. Ne yazık ki, form üzerindeki Properties penceresi açık olduğunda Screen.ActiveForm deyimi geçerli bir form belirtmedi. Bu durumda eğer açık olan birden fazla form var ise aktif form seçilmeliydi. Sadece bir form açık ise aktif olsun ya da olmasın o seçilmeliydi. Hiç form açık değilse uyarı yapılmalıydı. Bunun için GetActiveForm işlevini yazarak her seferinde uzun kontrol kodları yazmaktan kurtuldum.

Tüm bu işlevleri kullanabilmek için sadece bir kez araç çubuğu oluşturup bunun üzerindeki nesnelere

yazdığım işlevleri bağlamak yeterli oldu.

CreateCommandBar yordamını bir modül sayfasına yerleştirip yordam içindeki herhangi bir satırda F5 tuşuna basarak onu çalıştırın. Bunu sadece bir kez yapın. Toolbar nesnelerinin kullanılabilmesini Office'te ortak kullanılan Microsoft Office 9.0 Object Library kütüphanesi sağlıyor. Başvurular listesini açıp bunun seçili olmasını sağlayın.

```
Sub CreateCommandBar()
```

```
Dim cb As CommandBar
```

```
'CommandBar varsa bul, yoksa ekle.
```

```
Set cb = FindCommandBar(cb_FormUtils, True)
```

```
'''SetTabOrder
```

```
Dim ed As CommandBarControl
```

```
If FindCommandBarControl("SetTabIndex", cb) = False Then
```

```
Set ed = cb.Controls.Add(msoControlEdit)
```

```
ed.Width = 50
```

```
ed.Caption = "SetTabIndex"
```

```
ed.TooltipText = "Formdaki seçili nesnelerin TabIndex değerlerini" & _  
vbCr & "bu numaradan başlayarak yeniden düzenle"
```

```
ed.OnAction = "commandbar_SetTabOrder"
```

```
End If
```

```
'''FindField
```

```
If FindCommandBarControl("FindField", cb) = False Then
```

```
Set ed = cb.Controls.Add(msoControlEdit)
```

```
ed.Width = 100
```

```
ed.Caption = "FindField"
```

```
ed.TooltipText = "Veri alanını bul"
```

```
ed.OnAction = "commandbar_FindField"
```

```
End If
```

```
'''SelectControls
```

```
Dim combo As CommandBarComboBox
```

```
If FindCommandBarControl("SelectControls", cb) = False Then
```

```
Set combo = cb.Controls.Add(msoControlDropdown)
```

```
combo.Width = 125
```

```
combo.Caption = "SelectControls"
```

```
combo.TooltipText = "Bu Türdeki Nesneleri Seç"
```

```
combo.AddItem "Label"
```

```
combo.AddItem "TextBox"
```

```
combo.AddItem "CheckBox"
```

```
combo.AddItem "ComboBox"
```

```
combo.AddItem "ListBox"
```

```
combo.AddItem "CommandButton"
```

```
combo.AddItem "ToggleButton"
```

```
combo.AddItem "Line"
```

```
combo.AddItem "Rectangle"
```

```
combo.AddItem "Image"
```

```
combo.OnAction = "commandbar_SelectControls"
```

```
End If
```

```
Dim bt As CommandBarButton
```

```
'''Geçersizler
```

```
If FindCommandBarControl("Geçersizler", cb) = False Then
```

```
Set bt = cb.Controls.Add(msoControlButton)
```

```
bt.Style = msoButtonCaption
```

```
bt.Caption = "Geçersizler"
```

```
bt.TooltipText = "Geçersiz alan tanımlı nesneleri seç"
```

```

bt.OnAction = "commandbar_InvalidFields"
End If

'''Kullanılmayanlar
If FindCommandBarControl("Kullanılmayanlar", cb) = False Then
Set bt = cb.Controls.Add(msoControlButton)
bt.Style = msoButtonCaption
bt.Caption = "Kullanılmayanlar"
bt.TooltipText = "Formda kullanılmayan veri alanlarını bul"
bt.OnAction = "commandbar_NotUsedFields"
End If

cb.Visible = True

End Sub

```

Bu modülün ilk halinde çalışırken her seferinde nesneler tekrarlanıyordu. Bunu önlemek için FindCommandBar ve FindCommandBarControl adlı işlevleri yazdım. Bu ikisi ile hem menü çubuğu içindeki nesnelerin olup olmadığını, hem de barın kendisinin var olup olmadığını kontrol edebiliyorum. Ana kodu çirkinleştirmemek için FindCommandBar'da tanımladığım AutoCreate parametresi ile barı otomatik oluştuyorum.

```

Function FindCommandBar(BarName As String, Optional AutoCreate As Boolean = False)
As CommandBar
'CommandBar'ın olup olmadığını bulur.
'Bulamadığında AutoCreate=True ise oluşturur.
On Error GoTo son
Set FindCommandBar = CommandBars(BarName)
Exit Function
son:
Err.Clear
If AutoCreate Then Set FindCommandBar = CommandBars.Add(BarName)
End Function

Function FindCommandBarControl(ControlName As String, cb As CommandBar) As Boolean
'CommandBarControl nesnesinin olup olmadığını bulur.
On Error GoTo son
Dim ct As CommandBarControl
Set ct = cb.Controls(ControlName)
FindCommandBarControl = True
Exit Function
son:
Err.Clear
End Function

```

CommandBar nesnelerinin çalıştıracağı işlev OnAction özelliği ile belirtiliyor. Bu özelliğe, çalıştırılacak işlevin adını yazmak gerekiyor.

```

Public Function commandbar_SetTabOrder()
Dim f As Form
If GetActiveForm(f) Then
SetTabOrder f, CommandBars(cb_FormUtils).Controls("SetTabIndex").Text
End If
End Function

Public Function commandbar_SelectControls()
Dim t As Integer
t = GetControlType(CommandBars(cb_FormUtils).Controls("SelectControls").Text)

If t = 0 Then Exit Function

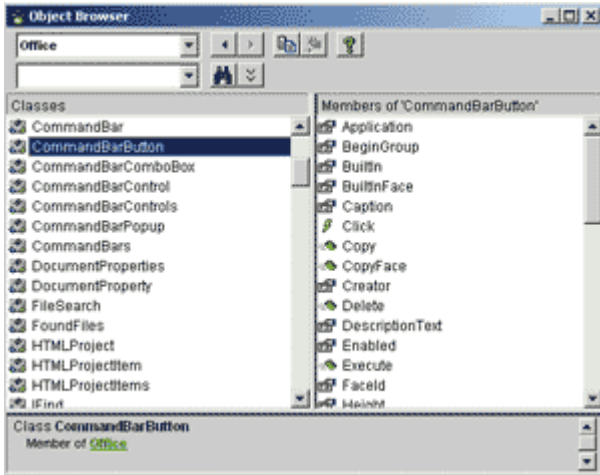
```

```

Dim f As Form
If GetActiveForm(f) Then
SelectControls f, t, GetKeyState(VK_CONTROL) < 0
'''CTRL basılı ise önceki seçim iptal edilmesin.
End If
End Function

```

CommandBar içindeki nesnelerin tamamı CommandBarControl tipindedir. Fakat bazıları özelleştirilebilir olduğundan bunlar için yeni tipler tanımlanmış. Örneğin CommandBarComboBox nesnesi aynı zamanda bir CommandBarControl'dur ama bundan daha farklı metot ve özellikler içerir. Object Browser penceresinde gördüğümüz gibi Button, Popup gibi nesneler de var.



SEÇ BUL DÜZENLE VS.

Asıl işleri yapacak işlevlerimiz de hazır. Aktif formu bul, tab duraklarını düzenle, belli türdeki tüm nesneleri seç, veri alanını formda bul, geçersiz veri alanlarını bul, formda kullanılmayan veri alanlarını bul. Tabii bu yazıda bunların tamamı yer almıyor.

```

Function GetActiveForm(F As Form) As Boolean
'Aktif olan ya da tek başına açık olan formu getirir.
'Sonuç form F değeridir. İşlev True ise geçerli
'bir form bulunmuş demektir.

```

```

GetActiveForm = False
If Forms.Count = 0 Then
MsgBox "Aktif form bulunamadı."
Exit Function
ElseIf Forms.Count = 1 Then
Set f = Forms(0)
Else
On Error Resume Next
Set f = Screen.ActiveForm
If Err.Number = 2475 Then
MsgBox "Aktif form bulunamadı."
Err.Clear
Exit Function
End If
On Error GoTo 0
End If
GetActiveForm = True
End Function

```

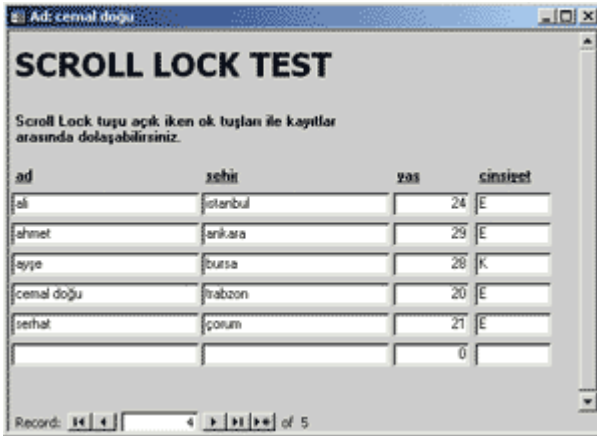
ACCESS HİKAYELERİ

Bu ay yardımcı işlemlere el attık. Yazının konusu form düzenlerken kullanabileceğiniz işlevler içeriyordu. Yeni yazdığım üç yararlı başka işlev de bu bölümde.

Ok tuşları ile formdaki kayıtlarda hareket edin. Tutumlu işlev ile yazıcıya gönderdiğiniz formları yazdırırken gereksiz toner masrafını önleyin. Çalışma anında tüm nesneleri belli bir oranda yeniden boyutlandırın.

AYIN TUŞU

Klavyedeki Scroll Lock tuşu kendi ışığını açıp kapamaktan başka ne işe yarar diye düşündünüz mü hiç? Boş kalsın diye Excel'de sayfayı kaydırma moduna geçme özelliği vermişler. Access'te de işe yaraması için işte size işlev. Kayıt sayfanız liste halindeyken Scroll Lock tuşu açık olduğunda, aşağı/yukarı oklarıyla kayıtlar arasında dolaşabilirsiniz.



Bir modül sayfasında en üste aşağıdaki kodları yerleştirin:

```
Private Declare Function GetKeyState Lib "user32" (ByVal nVirtKey As Long) As Integer
Private Const VK_SCROLL = &H91
```

```
Public Sub ScrollMode(frm As Form, KeyCode As Integer, Shift As Integer)
```

```
'Serkan Sahinoglu, 2001, http://www.bilgiteknoloji.net
```

```
If GetKeyState(VK_SCROLL) = 1 And frm.DefaultView = 1 Then
```

```
    Select Case KeyCode
```

```
        Case vbKeyUp
```

```
            SendKeys "^{PGUP}"
```

```
            KeyCode = 0
```

```
        Case vbKeyDown
```

```
            SendKeys "^{PGDN}"
```

```
            KeyCode = 0
```

```
'SOL ve SAĞ okları kullanmak istemiyorsanız
```

```
'alttaki iki Case bloğunu silin.
```

```
        Case vbKeyRight
```

```
            SendKeys "{TAB}"
```

```
KeyCode = 0  
Case vbKeyLeft  
SendKeys "+{TAB}"  
KeyCode = 0
```

```
End Select  
End If
```

```
End Sub
```

Kullanmak istediğiniz formun KeyPreview (olay yordamı listesinde alttan itibaren altıncı sırada) özelliğini aktif ettikten sonra formun KeyDown (tuşa basılırken) olay yordamını şu şekilde düzenleyin:

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)  
ScrollMode Me, KeyCode, Shift  
End Sub
```

Bundan sonra Scroll Lock tuşu açık ise klavyedeki ok tuşlarını kullanarak kayıtlar arasında hareket edebilirsiniz.

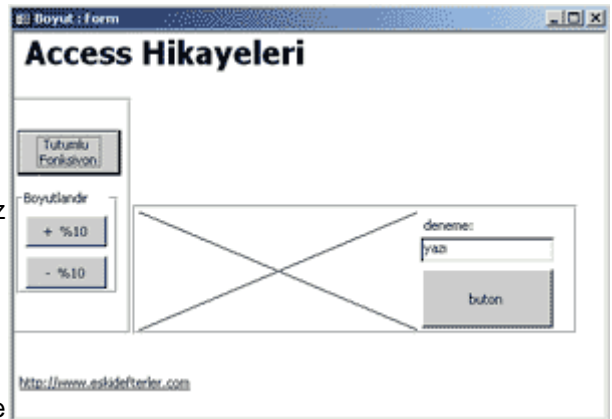
TUTUMLU İŞLEV

Programınızda güzel bir form hazırladınız, zevkinize göre istediğiniz renklerle süslediniz. Yeri geldiğinde bu formu yazıcıya gönderiyorsunuz. Bu halde iken formdaki verilerden çok, diğer renkler için bir sürü toner gidiyor.

Böyle durumlarda aşağıdaki tutumlu işlevimiz gereksiz toner masrafını ortadan kaldırıyor. Kullanmak istediğiniz kod kısmına ya da bir düğmenin Tıklatıldığında (OnClick) eylemine

Tutumluİşlev Me

yazdıktan sonra aşağıdaki işlevi genel bir modülün içine yerleştirin:



```
Public Sub Tutumluİşlev(frm As Form)  
'Serkan Sahinoglu, 2001, http://bilgiteknoloji.net
```

```
If MsgBox("Bu işlemi yaptıktan sonra eski görüntüye dönmek için pencereyi kapayıp tekrar açmalısınız. Devam edilsin mi?", vbYesNo) = vbNo Then Exit Sub
```

```
On Error Resume Next
```

```
Dim i As Integer
```

```
DoCmd.Hourglass True
```

```
'Tüm bölgeler beyaz. Toplam 9 bölge vardır.  
For i = 0 To 8  
frm.Section(i).BackColor = 16777215 'beyaz  
Next
```

```
'Tüm nesneleri yeniden renklendir.  
For i = 0 To frm.Controls.Count
```

```
frm.Controls(i).BackColor = 16777215 'beyaz
frm.Controls(i).ForeColor = 0 'siyah
Next

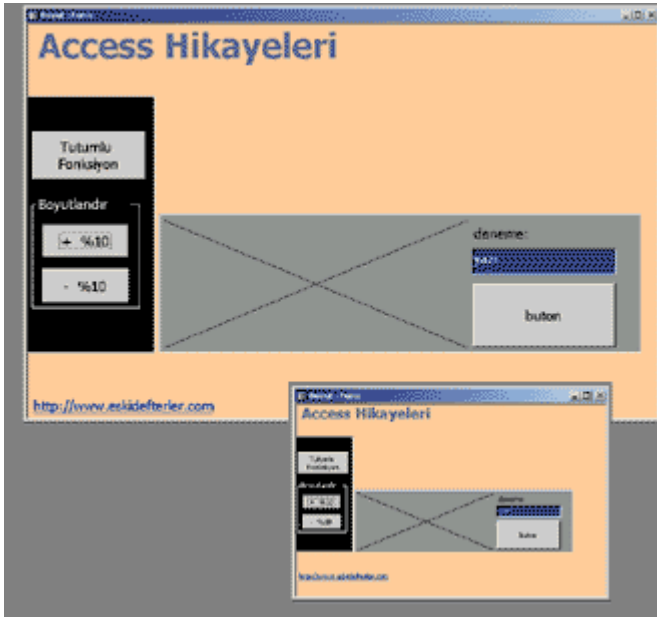
DoCmd.Hourglass False

End Sub
```

ayın işlevi

TÜM NESNELERİ YENİDEN BOYUTLANDIRIN

İşte size formlarınızı içindekilerle birlikte yeniden boyutlandırmanıza yarayan bir işlev. Bununla kullanıcılarınızın kendi ekranlarına göre boyutlandırma yapabiliyorsunuz. Alın, kullanın.



Formun OnOpen (Açıldığında) olay yordamına

Boyutlandır Me, 1.25

'Tüm nesneler 1.25 kat büyüyecek.

yazın. Tabii önce Boyutlandır işlevini genel bir modül sayfasına ekleyin.

Kod biraz hassas olduğu için hata oluştuğunda formunuzun içindeki bölgeleri biraz genişletmeyi deneyin. Diğer boyutlandırma seçenekleri için koddaki Oran adlı değeri değiştirebilirsiniz.

Formunuzu standart olarak 800*600 için hazırlamışsanız;

800*600 ekranda oran = 1

640*480 ekranda oran = 0.8

1024*768 ekranda oran = 1.28

değerlerini kullanın.

```
Public Sub Boyutlandır(frm As Form, Oran As Double)
'Serkan Sahinoglu, 2001, http://www.bilgiteknoloji.net

If Oran = 1 Then Exit Sub

Dim i As Integer
On Error Resume Next 'Hataları yoksay.

frm.Width = frm.Width * Oran 'En fazla 31680.

'Tüm bölgeler
For i = 0 To 8
frm.Section(i).Height = frm.Section(i).Height * Oran
Next

'Tüm nesneler
For i = 0 To frm.Controls.Count - 1
If frm.Controls(i).ControlType <> acPage Then
With frm.Controls(i)
.Width = .Width * Oran
.Height = .Height * Oran
.Left = .Left * Oran
```



```
.Top = .Top * Oran
.FontSize = .FontSize * Oran
'Bazı yazıtipleri küçülmüyor.
'Arial ve Times New Roman küçülüyor.
End With
End If
Next
End Sub
```

DOSYA:

[AccessEkim2001ornek.zip](#)

- veritabanı özel -

Gelişmiş Sorgular

Sayfa 1 [Sayfa 2](#) [Sayfa3](#)

Hangi veritabanını kullanırsanız kullanın, bu ay anlatılan birçok sorgu türüne ihtiyacınız olabilir. Bu yazıda SQL Server, Paradox gibi sistemlerde kullanabileceğiniz gelişmiş sorgu örnekleriyle birlikte, Access'in kendisine özel sorgu türlerinin ilginç örneklerini bulacaksınız. Access yerine bu ay veritabanı inceliklerinin yer aldığı Veritabanı özel köşesi ile karşınızdayız.

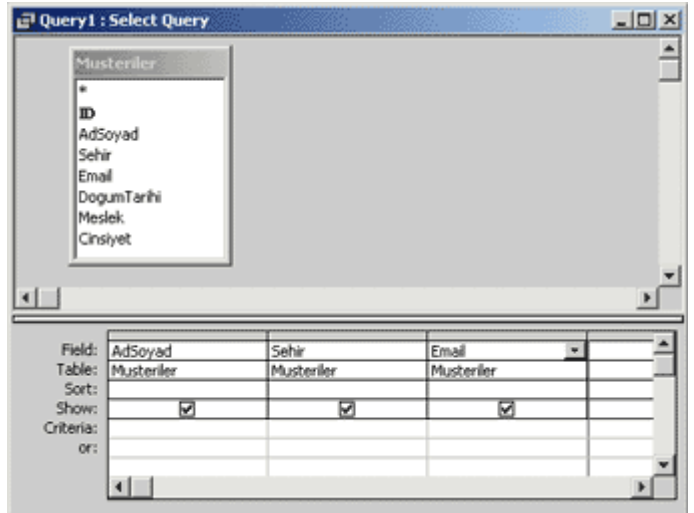
Yazıda anlatılan sorgu türlerinin birkaçı dışında hemen hepsini Access, Delphi, VB ya da ASP ile çalışırken aynı şekilde kullanabilirsiniz. İhtiyacınıza göre yapılandırabileceğiniz karmaşık sorgular, çarpaz sorgular, iç içe sorgular, birleşmiş sorgular yazının devamında.

SORGULARA BAŞLAYIN

Access, sorgulara başlamak için ya da gelişmiş sorgular hazırlamak için, en iyi sorgu tasarım araçlarından birini içerir. Access'teki Sorgu Sihirbazı'na (Query Builder) hakim birisi için en karmaşık sorguları hazırlamak çocuk oyuncağı sayılır.

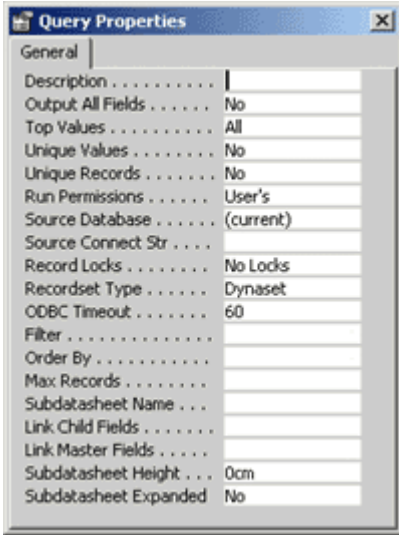
Yeni bir sorgu oluşturma aşamasında, açılan listedeki tabloları seçip sorguya eklemek gerekiyor. Sorguda kullanılacak bir veya daha fazla tablo burada seçiliyor. Tablo eklemeyi daha sonra yapmak için Görünüm > Tablo Göster (View > Show Table) menüsü kullanılıyor.

Görünen tablolardaki alanları çift tıklatıp aşağıdaki bölgeye ekleyin ve Görünüm > Tablo Görünümü menüsü ile sorgunun sonucunu görün.



SORGU ÖZELLİKLERİ

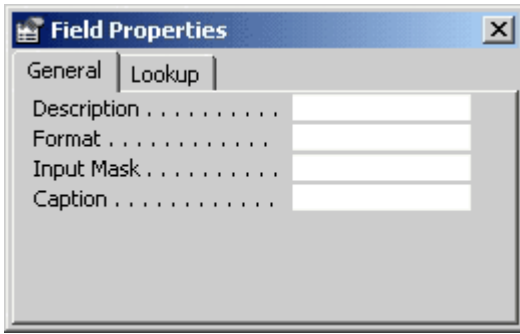
Sorgu Sihirbazı diğer bileşenler gibi özelliklerin listelendiği bir pencereye sahiptir. Sorgu tasarımında iken Görünüm > Özellikler (View > Properties) menüsü ile özellikler penceresi açılıyor. Yeri geldiğinde bu pencere işinize çok yarayabilir.



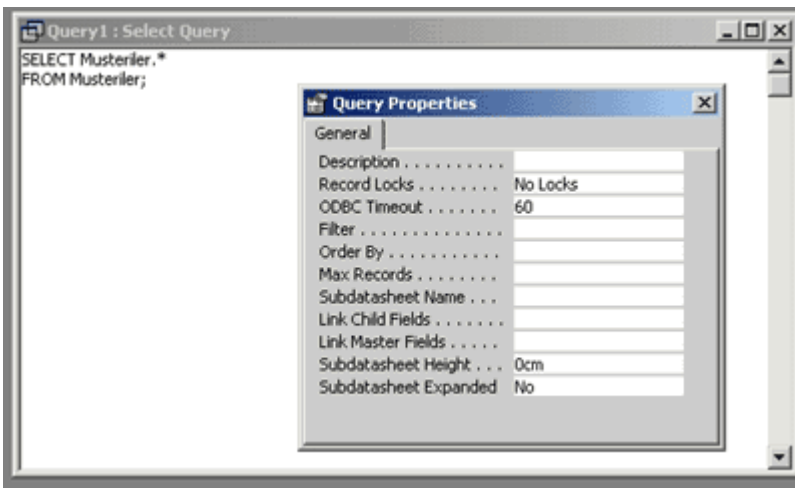
Özellikler penceresi seçili olan nesne için yerine göre dört farklı liste sunuyor. Eğer sorguya eklediğiniz bir tabloyu seçmişseniz bu pencere o tablonun özelliklerini listeliyor.



Üstteki boş alanı seçtiğinizde sorgunun, alt tarafa yerleştirdiğiniz bir veri alanını seçtiğinizde o veri alanının özelliklerini listeliyor ve değiştirebilmenizi sağlıyor. Fakat veri alanı kısmında tüm alanları listelemek için kullandığınız bir etiket olduğunda (Musteriler.* gibi) veri alanının değil –ki bu zaten veri alanı değildir– sorgunun özellikleri listelenir.



Dördüncü durum ise sql görünümünün açık olması halinde geçerli. Bu görünümde de başka özellikler listeleniyor.



Özellikler penceresi tasarım anında sorgunun koduna müdahale etmeye gerek kalmadan önemli bazı ayrıntıların değiştirilebilmesini sağlıyor.

Örneğin sorgu özellikleri penceresi açıkken Top Values değeri ile kayıt kümesindeki kayıtları sınırlandırabiliyorsunuz. Bunun sql'deki karşılığı TOP deyimi. "SELECT TOP 50 * FROM Musteriler" gibi bir kod kayıt kümesinin ilk 50 kaydını getirir. Tasarım anında sql kodu görünümü açmak yerine özellikler penceresindeki bu değeri değiştirmeniz yeterli.

Bu listedeki diğerlerine bakarsanız yüzdeli değerler göreceksiniz. Bu durumda kayıt kümesinin ilk yüzde 10'luk dilimini listelemeniz mümkün. Görünüm menüsünden sql görünümüne geçtiğinizde şuna benzer bir kod göreceksiniz:

SELECT TOP 10 PERCENT Musteriler.*

FROM Musteriler;

Percent deyimini öncesindeki sayı kadar ilk yüzdelik dilimi listeliyor. Sıralamayı değiştirmek istediğinizde ise OrderBy deyimini kullanın.

Özellikler listesindeki özelliklerin bazıları sql kodunu etkilemez. Örneğin kayıt kümesini sıralamak için bu listedeki Order By özelliğine güvenmemek gerekiyor. Özellikler listesindeki Order By özelliği sorgunun kodunu etkilemeyip kayıtların listelenmesi aşamasında devreye giriyor. Bazı durumlarda sorguyu çalıştırırken açılışta sizden bir parametre istenirse sorgunun tasarımında bu listedeki Order By'ın değerinden şüphelenmelisiniz. Zira sorgunun sonucunu incelerken fare menüsündeki sıralama komutlarını kullandığınızda oluşan sıralama bilgisi buraya kaydediliyor. Daha sonra sorgunun ismini bir şekilde değiştirdiğinizde bu sorunla karşılaşabilirsiniz. ("Order By deyimini" ile "Order By özelliği" arasındaki farka dikkat!)

Kayıtları sıralamak için alt tarafa yerleştirdiğiniz her bir veri alanının altındaki Sırala (Sort) listesinden Artan ya da Azalan (Ascending – Descending) değerini seçmelisiniz.

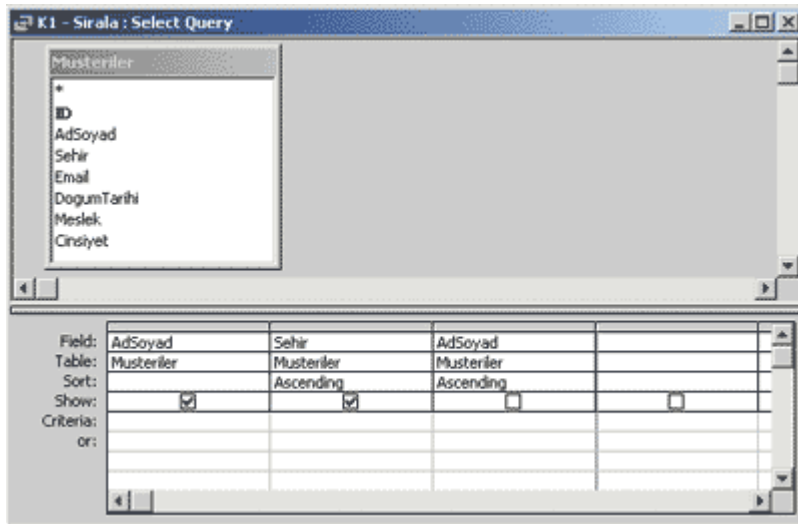
Üretilcek sql kodundaki Order By bölümü bu seçimler sonrasında oluşur. Çoklu sıralama yapmak için ilgili alanların altındaki listeden gerekli olanını seçin.

Veri alanlarının listelenmesindeki sıralama ile kayıtlardaki sıralamanın farkı olması gerektiğinde bazı alanları alttaki bölgeye tekrar yerleştirmeniz gerekebilir.

Bu ay bahsettiğim sorgu örneklerini Internet'ten yükleyebileceğiniz bir dosyaya yerleştirdim ve kolay anlaşılması için sorgu isimlerinin başına grup numarası verdim. Bu yazıda da aynı isimleri kullandım.

LİSTELENECEK ALANLARI SEÇİN

Resimdeki K1-Sırala sorgusunu inceleyin.



Sütunların ilkinin Adsoyad alanı olması gerektiği için sıralama koşulundan dolayı bu alanı ikinci kez ekledim ve uygun sıralamayı belirttim. Alt taraftaki Göster (Show) seçeneğini kaldırıp bu alanın gösterilmemesini, sadece sıralama amaçlı kullanılmasını sağladım.

AdSoyad alanını iki defa eklemek, kayıt kümesinde bu alanın iki kez yer alacağı anlamına gelmiyor. Show satırındaki kutunun seçili olmadığı alanlar görüntülenmez.

Query Properties penceresindeki diğer seçeneklerden biri de sizin yerleştirmenize gerek kalmadan tüm alanların listelenmesini sağlayan Output All Fields özelliği.

Bu özelliği aktif ettiğinizde tüm veri alanları kayıt kümesinde listelenir.

```
SELECT AdSoyad, Sehir, *  
FROM Musteriler  
ORDER BY Sehir, AdSoyad;
```

Yeni bir sorguda iki veri alanını alt tarafa yerleştirdikten sonra Output All Fields özelliğini Yes olarak işaretledim. Sorgunun sql kodu yukarıdaki gibi oldu.

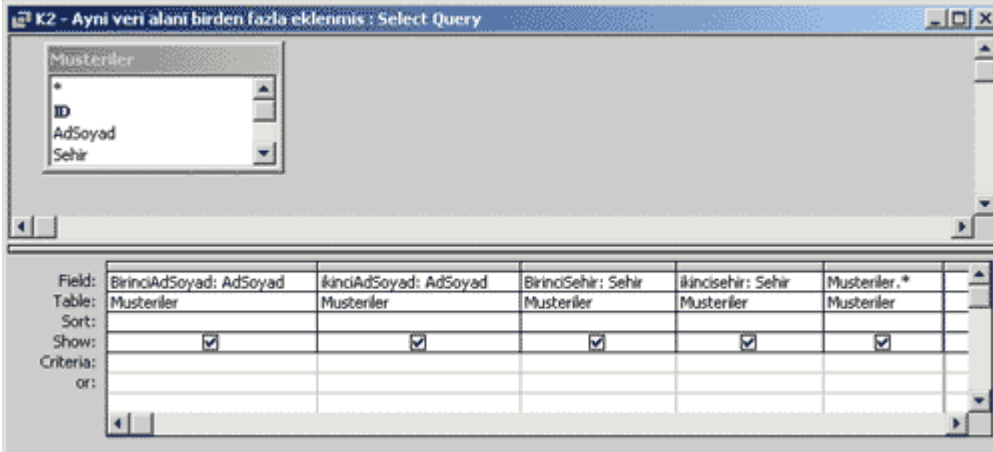
Tasarım görünümünde tabloların üst tarafında bulunan yıldız işareti ile buradaki birbirinden biraz farklı.

Output All Fields özelliğini hiç kullanmadan tablodaki yıldız işaretini ya da bazı veri alanlarını fareyle sürükleyip aşağıdaki alana dizebilirsiniz. Tablonun üstündeki yıldızla birlikte diğer veri alanlarını da aynı zamanda aşağıya ekleyebilirsiniz. Tablodaki yıldız işareti diğer alanları ekleseniz de tüm alanları tekrar görüntüler.

```
SELECT Musteriler.*  
FROM Musteriler;
```

Bu kodları henüz göremediyseniz sorgu tasarımı View (Görünüm) menüsünde üstteki üç komutu inceleyin. Tasarım, SQL, Tablo görünümleri (Design, SQL, Datasheet) ile sorguyu üç farklı halde görüntüleyebilirsiniz.

Bir veri alanının birden çok kez eklendiği K2 sorgusunun resmini inceleyin.



Bu tür bir sorguyu çalıştırdığınızda isimlerin tekrarlanmasından dolayı sütun başlıklarını Access kendine göre yeniden adlandıracaktı. Böyle durumlarda altta veri alanının adından önce yeni ad yazılıyor ve ikisinin arasına üstüste iki nokta işareti ekleniyor. Bu, veri alanının yeni bir adla gösterilmesini sağlıyor. Bunun SQL dilindeki karşılığı ise AS deyimi.

```
SELECT AdSoyad AS BirinciAdSoyad, AdSoyad AS ikinciAdSoyad, Sehir AS BirinciSehir, Sehir AS  
ikincisehir, Musteriler.*  
FROM Musteriler;
```

AS deyimi standart SQL dilinde de var olup birçok yerde buna benzer şekilde kullanılıyor. Sadece veri alanı için değil, tablo isimleri için de bunu kullanabilirsiniz.

Bu deyim SQL Server'daki View objelerinde de kullanılabilir. Örneğin Haberler adlı tablonuzda Türkçe ve İngilizce haberler yer aldığında bunları ASP sayfalarında pratik olarak kullanmak için View nesnelerinden faydalanabilirsiniz.

İngilizce ve Türkçe haber bilgilerini aynı tabloda tuttuğunuzda muhtemelen Baslik, Baslik_Eng, Haber, Haber_Eng gibi isimlerde veri alanlarınız olacak.

Haberler_Eng adlı bir view oluşturup içine şuna benzer bir sql kodu ekleyin:

```
SELECT Baslik_Eng as BASLIK, Haber_Eng as HABER
```

FROM Haberler

Türkçe ASP sayfasını hazırladıktan sonra bunun kopyasını alıp alan isimlerinin olduğu bloklarda değişiklik yapmak yerine sadece Recordset nesnesinin açıldığı yere Haberler_Eng yazmak yeterli olur.

Peki Output All Fields özelliği ne için var? Sorgu tasarım görünümünde tablonun üstündeki yıldız aşağı sürüklediğinizde oluşan sql kodu şu şekilde idi:

```
SELECT Musteriler.*  
FROM Musteriler;
```

Output All Fields ise şunu üretiyor:

```
SELECT *  
FROM Musteriler;
```

Tek başına yıldız işareti olmadan ALL deyimi de geçerli.

```
SELECT ALL *  
FROM Musteriler;
```

Yıldız işareti, seçilmemiş olan alanları sorgunun devamına ekliyor. Bu işaret hangi sırada olursa olsun kullanılmamış alanlar son sıradan itibaren dizilir.

```
SELECT *, Adsoyad, Email  
FROM Musteriler;
```

ya da

```
SELECT Adsoyad, Email, *  
FROM Musteriler;
```

deyimleri, sırasıyla AdSoyad, Email, ID, Sehir, DogumTarihi, Meslek, Cinsiyet alanlarını listeleyecek ve alan tekrarı olmayacak.

```
SELECT Adsoyad, Email, Musteriler.*  
FROM Musteriler;
```

kullandığınızda ise AdSoyad ve Email alanları tekrar edecekti.
Buraya kadarki sorgular için örnek dosyada K grubundaki sorguları inceleyin.

Sorguların performansı için, gerekli olmayan veri alanlarını sorguya dahil etmemek daha iyidir. Örneğin "SELECT * FROM Musteriler" gibi bir deyim yerine tüm alanları listelemeyip sadece gerekli alanların seçildiği "SELECT AdSoyad, Email FROM Musteriler" gibi sorgular yazmanız önerilir.

Sayfa 1 [Sayfa 2](#) [Sayfa3](#)

DOSYA:
[GelistmisSorgularMDB.zip](#)

**Otomatik Sayı Değişmez
Dediniz ve Kaybettiniz!**

Otomatik Sayı (Autonumber) alanları ile ilgili birçok yerde hala aynı sorular soruluyor.

- Kayıtlar silindiğinde neden eski numaralar kullanılamaz oluyor?
- Bu alanı 1'den itibaren yeniden sıralayamaz mıyım?

Verilen cevaplar hep aynı. Herhalde bunu bilmeyen de kalmamıştır. Bu veri alanının temel özelliği her yeni kayıt anında otomatik artması. Örneğin Access içinde iken yeni kayıt moduna geçildiği anda bu değer otomatik olarak artırılır. ADO ile SQL Server'a bağlanırken konumlandırıcı tipine göre ekleme komutu gönderildikten sonra bu değerler değişik seçeneklere göre artırılabilirler. (*)

Numaraları 1'den itibaren sıralamak için veri alanı silinip yeniden oluşturulabilir. Ama eğer bu alan başka bir veri alanı ile bağlantılıysa veritabanında karmaşa oluşacaktır.

Yandaki resme bakıp aldanmayın. Normal şartlarda Otomatik Sayı alanı değiştirilmez. Benim yaptığım şey sadece veritabanı dosyasına dışarıdan müdahale etmektir.

ID	ad	departman
2	tekin	teknik
3	idil	idari
5	pelin	pazarlama
6	inci	insan kaynakları
9	serhat	servis
12	bilgen	bilgi işlem
15	murat	muhasese
1	DENEME1	
2	DENEME2	
3	DENEME3	
4	DENEME4	
5	DENEME5	
6	DENEME6	

Bu işlemin nasıl yapıldığını anlatıyorum fakat kullanmanızı tavsiye etmiyorum ve problem çıkmayacağını garantilemiyorum.

Deneme amaçlı bir veritabanı ve bu veritabanı dosyasında otomatik sayı alanı içeren bir tablo oluşturdum. Bu tabloya 15 kayıt ekledim ve bazı kayıtları sildim. Bu durumdayken dosyanın bir yedeğini aldım.

Normal şartlarda Otomatik Sayı değerinin veritabanı dosyası içinde bir yerlerde tutulması gerekir. Bu durumda otomatik sayı ile ilgili bir işlem sonrasında veritabanı dosyasında bir yere bu bilgi yazılmalıdır.

Bu teoriden yola çıkarak tabloda yeni kayıt satırına gelip birşeyler yazdım ama sonra ESC tuşu ile kayıttan vazgeçtim. Böylelikle son kullanılan otomatik sayı değeri dosyanın bir yerine yazılmış olur.

Dosya kapatıp orjinal ve yeni dosyaların arasındaki farkları program kodu yazarak araştırdım.

```
Sub FarklariBul()  
    Open "c:\autonumber1.mdb" For Binary As #1  
    Open "c:\autonumber2.mdb" For Binary As #2  
    Dim i As Long  
    Dim deger1 As Byte, deger2 As Byte  
    For i = 1 To FileLen("c:\autonumber1.mdb")  
        Get #1, i, deger1  
        Get #2, i, deger2  
        If deger1 <> deger2 Then Debug.Print i, deger1, deger2  
    Next  
    Close #1  
    Close #2  
End Sub
```


İlk denemelerimde birçok fark olduğunu gördüm. Bunun sebebi veritabanı penceresinde farklı sayfa ve nesnenin seçili olmasıydı. Bu durumda farkı daha iyi algılayabilmek için tablolar sayfasının açık olması ve de kontrol edeceğim tablonun üzerine konumlanmış olmam gerekiyordu. En son denememde sadece iki fark olduğunu gördüm.

Konum (bayt)	Birinci dosyadaki ASCII kod	İkinci dosyadaki ASCII kod
3.587	1	2
106.517	15	16

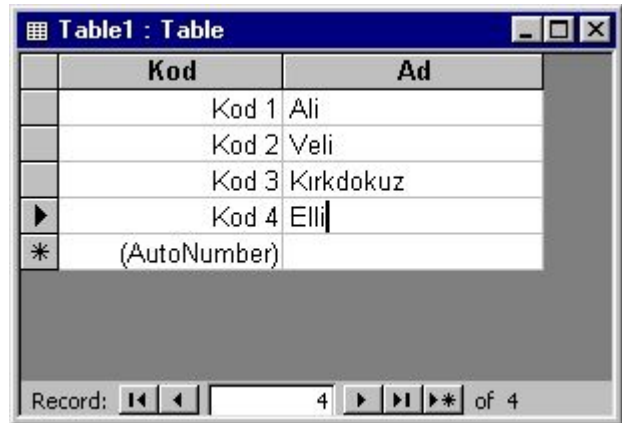
Her iki satır için ayrı ayrı değer aramaya başlamak gerekiyor. Otomatik sayı değerimiz en son 16 olduğuna göre bu değeri oluşturan baytları bulmak gerekiyordu.

106517. bayttan sonraki 4 bayt (LONG tipinde), 16 sayısını yani son kullanılan otomatik sayı değerini veriyordu. Bunun yerine bizim istediğimiz yeni değeri yazmamız yeterli oluyor.

```
Sub Getir()  
    Open "c:\autonumber1.mdb" For Binary As #1  
    Dim deger As Long  
    Get #2, 106517, deger  
    MsgBox deger  
    Close #2  
End Sub  
  
Sub YenisiniYaz()  
    Open "c:\autonumber2.mdb" For Binary As #2  
    Dim deger As Long  
    deger = 0  
    Put #2, 106517, deger  
    Close #2  
End Sub
```

Oraya ne yazılırsa otomatik sayı yeni değerini bu numaradan başlatıyordu. Burayı 0 olarak değiştirdikten sonra üst resimdeki gibi bir görüntü normaldi artık.

Bu işlemten sonra Veritabanı Düzenle (Compact) komutu kullanıldığında Otomatik Sayı normal hale dönecektir. Siz en iyisi bu değeri arkaplan işlemleri için kullanın ya da kullanıcıya herhangi bir kod değeriymiş gibi gösterin. Ben olsam ekran başındaki kişiyi resimde olduğu gibi kandırırdım. Nasıl mı oldu diyorsanız, tablonun tasarım görünümünde otomatik sayı alanının Format (Biçim) özelliğinin değerini tam olarak "Kod" "# şeklinde belirtmeniz yeterli.



Kod	Ad
Kod 1	Ali
Kod 2	Veli
Kod 3	Kırkdokuz
Kod 4	Elli
(AutoNumber)	

Değiştirilmiş otomatik sayı değeri ile ilgili örnek dosyayı yükleyip inceleyebilirsiniz.
[autonumber_mdb.zip](#) (17 KB)

- veritabanı özel -

Gelişmiş Sorgular



[Sonraki sayfa](#)

Hangi veritabanını kullanırsanız kullanın, bu ay anlatılan birçok sorgu türüne ihtiyacınız olabilir. Bu yazıda SQL Server, Paradox gibi sistemlerde kullanılabileceğiniz gelişmiş sorgu örnekleriyle birlikte, Access'in kendisine özel sorgu türlerinin ilginç örneklerini bulacaksınız. Access yerine bu ay veritabanı inceliklerinin yer aldığı Veritabanı özel köşesi ile karşınızdayız.

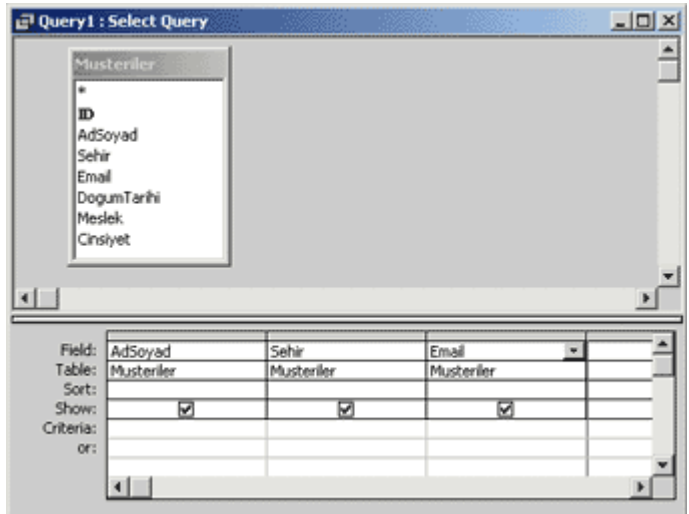
Yazıda anlatılan sorgu türlerinin birkaçı dışında hemen hepsini Access, Delphi, VB ya da ASP ile çalışırken aynı şekilde kullanabilirsiniz. İhtiyacınıza göre yapılandırabileceğiniz karmaşık sorgular, çarpaz sorgular, iç içe sorgular, birleşmiş sorgular yazının devamında.

SORGULARA BAŞLAYIN

Access, sorgulara başlamak için ya da gelişmiş sorgular hazırlamak için, en iyi sorgu tasarım araçlarından birini içerir. Access'teki Sorgu Sihirbazı'na (Query Builder) hakim birisi için en karmaşık sorguları hazırlamak çocuk oyuncağı sayılır.

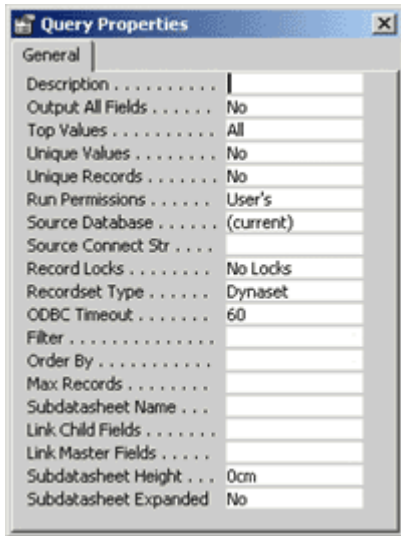
Yeni bir sorgu oluşturma aşamasında, açılan listedeki tabloları seçip sorguya eklemek gerekiyor. Sorguda kullanılacak bir veya daha fazla tablo burada seçiliyor. Tablo eklemeyi daha sonra yapmak için Görünüm > Tablo Göster (View > Show Table) menüsü kullanılıyor.

Görünen tablolardaki alanları çift tıklayıp aşağıdaki bölgeye ekleyin ve Görünüm > Tablo Görünümü menüsü ile sorgunun sonucunu görün.

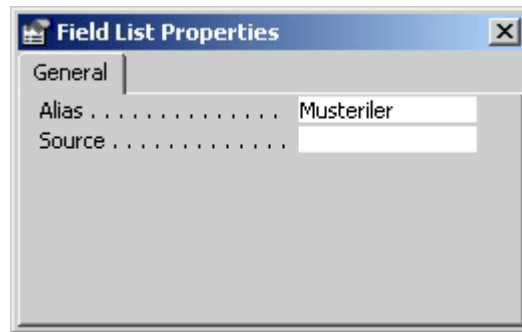


SORGU ÖZELLİKLERİ

Sorgu Sihirbazı diğer bileşenler gibi özelliklerin listelendiği bir pencereye sahiptir. Sorgu tasarımında iken Görünüm > Özellikler (View > Properties) menüsü ile özellikler penceresi açılıyor. Yeri geldiğinde bu pencere işinize çok yarayabilir.

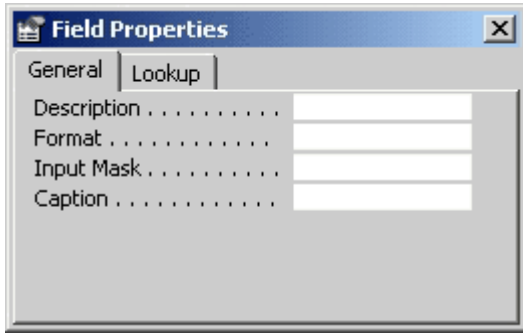


Özellikler penceresi seçili olan nesne için yerine göre dört farklı liste sunuyor. Eğer sorguya eklediğiniz bir tabloyu seçmişseniz bu pencere o tablonun özelliklerini listeliyor.

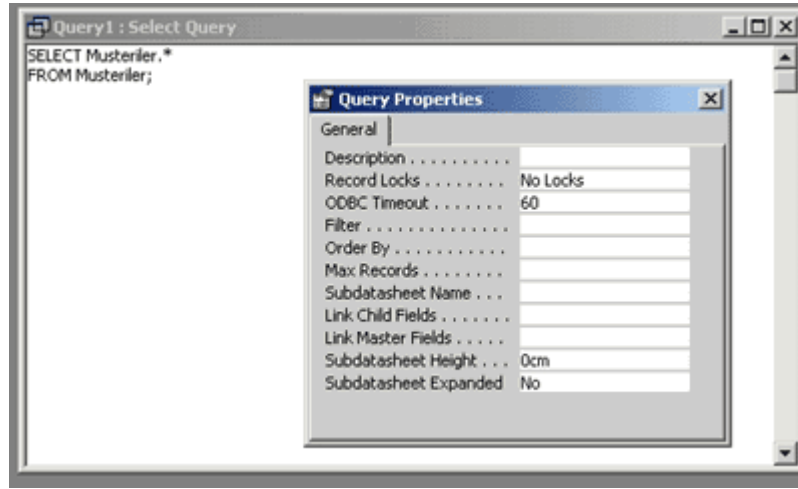


Üstteki boş alanı seçtiğinizde sorgunun, alt tarafa yerleştirdiğiniz bir veri alanını seçtiğinizde o veri alanının özelliklerini listeliyor ve değiştirebilmenizi sağlıyor. Fakat veri alanı kısmında tüm alanları listelemek için

kullandığınız bir etiket olduğunda (Musteriler.* gibi) veri alanının değil –ki bu zaten veri alanı değildir– sorgunun özellikleri listelenir.



Dördüncü durum ise sql görünümünün açık olması halinde geçerli. Bu görünümde de başka özellikler listeleniyor.



Özellikler penceresi tasarım anında sorgunun koduna müdahale etmeye gerek kalmadan önemli bazı ayrıntıların değiştirilebilmesini sağlıyor.

Örneğin sorgu özellikleri penceresi açıkken Top Values değeri ile kayıt kümesindeki kayıtları sınırlandırabiliyorsunuz. Bunun sql'deki karşılığı TOP deyimi. "SELECT TOP 50 * FROM Musteriler" gibi bir kod kayıt kümesinin ilk 50 kaydını getirir. Tasarım anında sql kodu görünümü açmak yerine özellikler penceresindeki bu değeri değiştirmeniz yeterli.

Bu listedeki diğerlerine bakarsanız yüzdeli değerler göreceksiniz. Bu durumda kayıt kümesinin ilk yüzde 10'luk dilimini listelemeniz mümkün. Görünüm menüsünden sql görünümüne geçtiğinizde şuna benzer bir kod göreceksiniz:

**SELECT TOP 10 PERCENT Musteriler.*
FROM Musteriler;**

Percent deyimi öncesindeki sayı kadar ilk yüzdelik dilimi listeliyor. Sıralamayı değiştirmek istediğinizde ise OrderBy deyimini kullanın.

Özellikler listesindeki özelliklerin bazıları sql kodunu etkilemez. Örneğin kayıt kümesini sıralamak için bu listedeki Order By özelliğine güvenmemek gerekiyor. Özellikler listesindeki Order By özelliği sorgunun kodunu etkilemeyeip kayıtların listelenmesi aşamasında devreye giriyor. Bazı durumlarda sorguyu çalıştırırken açılışta sizden bir parametre istenirse sorgunun tasarımında bu listedeki Order By'ın değerinden şüphelenmelisiniz. Zira sorgunun sonucunu incelerken fare menüsündeki sıralama komutlarını kullandığınızda oluşan sıralama bilgisi buraya kaydediliyor. Daha sonra sorgunun ismini bir şekilde değiştirdiğinizde bu sorunla karşılaşabilirsiniz. ("Order By deyimi" ile "Order By özelliği" arasındaki farka dikkat!)

Kayıtları sıralamak için alt tarafa yerleştirdiğiniz her bir veri alanının altındaki Sırala (Sort) listesinden Artan ya da Azalan (Ascending – Descending) değerini seçmelisiniz.

Üretilcek sql kodundaki Order By bölümü bu seçimler sonrasında oluşur. Çoklu sıralama yapmak için ilgili alanların altındaki listeden gerekli olanını seçin.

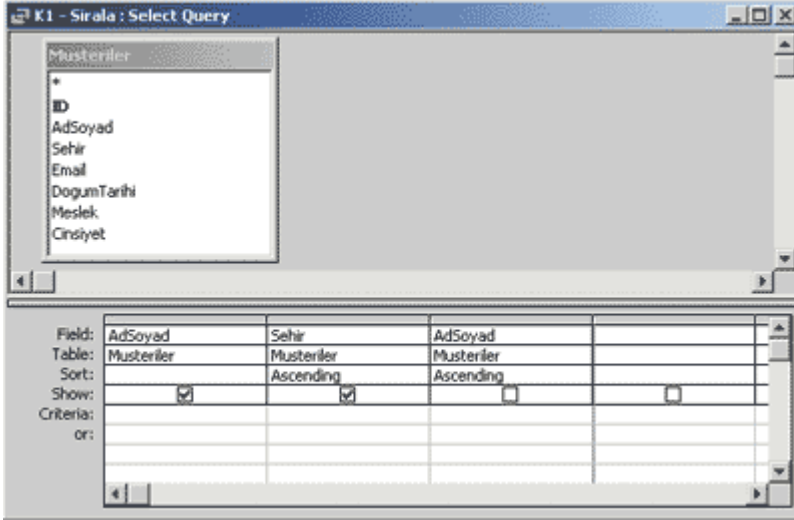
Veri alanlarının listelenmesindeki sıralama ile kayıtlardaki sıralamanın farkı olması gerektiğinde bazı alanları alttaki

bölgeye tekrar yerleştirmeniz gerekebilir.

Bu ay bahsettiğim sorgu örneklerini Internet'ten yükleyebileceğiniz bir dosyaya yerleştirdim ve kolay anlaşılması için sorgu isimlerinin başına grup numarası verdim. Bu yazıda da aynı isimleri kullandım.

LİSTELENECEK ALANLARI SEÇİN

Resimdeki K1-Sırala sorgusunu inceleyin.



Sütunların ilkinin Adsoyad alanı olması gerektiği için sıralama koşulundan dolayı bu alanı ikinci kez ekledim ve uygun sıralamayı belirttim. Alt taraftaki Göster (Show) seçeneğini kaldırıp bu alanın gösterilmemesini, sadece sıralama amaçlı kullanılmasını sağladım.

AdSoyad alanını iki defa eklemek, kayıt kümesinde bu alanın iki kez yer alacağı anlamına gelmiyor. Show satırındaki kutunun seçili olmadığı alanlar görüntülenmez.

Query Properties penceresindeki diğer seçeneklerden biri de sizin yerleştirmenize gerek kalmadan tüm alanların listelenmesini sağlayan Output All Fields özelliği.

Bu özelliği aktif ettiğinizde tüm veri alanları kayıt kümesinde listelenir.

```
SELECT AdSoyad, Sehir, *  
FROM Musteriler  
ORDER BY Sehir, AdSoyad;
```

Yeni bir sorguda iki veri alanını alt tarafa yerleştirdikten sonra Output All Fields özelliğini Yes olarak işaretledim. Sorgunun sql kodu yukarıdaki gibi oldu.

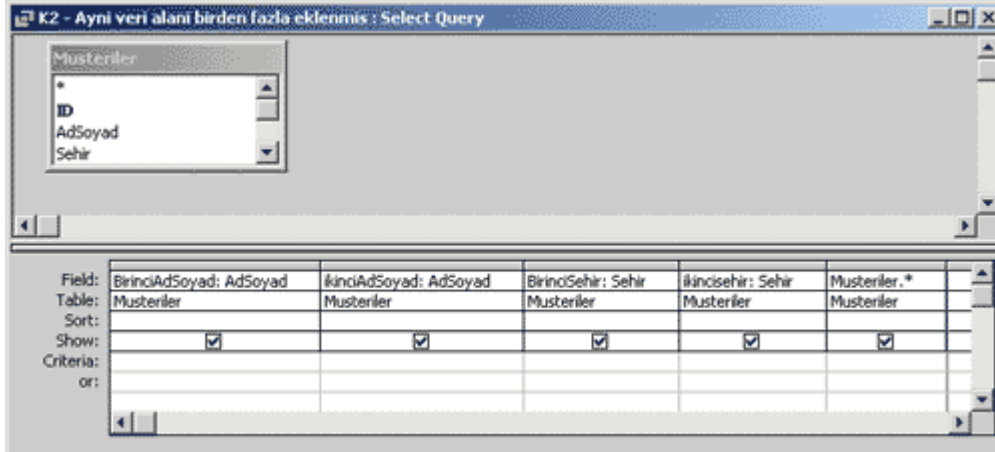
Tasarım görünümünde tabloların üst tarafında bulunan yıldız işareti ile buradaki birbirinden biraz farklı.

Output All Fields özelliğini hiç kullanmadan tablodaki yıldız işaretini ya da bazı veri alanlarını fareyle sürükleyip aşağıdaki alana dizebilirsiniz. Tablonun üstündeki yıldızla birlikte diğer veri alanlarını da aynı zamanda aşağıya ekleyebilirsiniz. Tablodaki yıldız işareti diğer alanları ekleseniz de tüm alanları tekrar görüntüler.

```
SELECT Musteriler.*  
FROM Musteriler;
```

Bu kodları henüz göremediyseniz sorgu tasarımında View (Görünüm) menüsünde üstteki üç komutu inceleyin. Tasarım, SQL, Tablo görünümleri (Design, SQL, Datasheet) ile sorguyu üç farklı halde görüntüleyebilirsiniz.

Bir veri alanının birden çok kez eklendiği K2 sorgusunun resmini inceleyin.



Bu tür bir sorguyu çalıştırdığınızda isimlerin tekrarlanmasından dolayı sütun başlıklarını Access kendine göre yeniden adlandıracaktı. Böyle durumlarda altta veri alanının adından önce yeni ad yazılıyor ve ikisinin arasına üstüste iki nokta işareti ekleniyor. Bu, veri alanının yeni bir adla gösterilmesini sağlıyor. Bunun SQL dilindeki karşılığı ise AS deyimi.

**SELECT AdSoyad AS BirinciAdSoyad, AdSoyad AS ikinciAdSoyad, Sehir AS BirinciSehir, Sehir AS ikincisehir, Musteriler.*
FROM Musteriler;**

AS deyimi standart SQL dilinde de var olup birçok yerde buna benzer şekilde kullanılıyor. Sadece veri alanı için değil, tablo isimleri için de bunu kullanabilirsiniz.

Bu deyim SQL Server'daki View objelerinde de kullanılabilir. Örneğin Haberler adlı tablonuzda Türkçe ve İngilizce haberler yer aldığında bunları ASP sayfalarında pratik olarak kullanmak için View nesnelerinden faydalanabilirsiniz.

İngilizce ve Türkçe haber bilgilerini aynı tabloda tuttuğunuzda muhtemelen Baslik, Baslik_Eng, Haber, Haber_Eng gibi isimlerde veri alanlarınız olacak.

Haberler_Eng adlı bir view oluşturup içine şuna benzer bir sql kodu ekleyin:

**SELECT Baslik_Eng as BASLIK, Haber_Eng as HABER
FROM Haberler**

Türkçe ASP sayfasını hazırladıktan sonra bunun kopyasını alıp alan isimlerinin olduğu bloklarda değişiklik yapmak yerine sadece Recordset nesnesinin açıldığı yere Haberler_Eng yazmak yeterli olur.

Peki Output All Fields özelliği ne için var? Sorgu tasarım görünümünde tablonun üstündeki yıldız aşağı sürüklediğinizde oluşan sql kodu şu şekilde idi:

**SELECT Musteriler.*
FROM Musteriler;**

Output All Fields ise şunu üretiyor:

**SELECT *
FROM Musteriler;**

Tek başına yıldız işareti olmadan ALL deyimi de geçerli.

SELECT ALL *

FROM Musteriler;

Yıldız işareti, seçilmemiş olan alanları sorgunun devamına ekliyor. Bu işaret hangi sırada olursa olsun kullanılmamış alanlar son sıradan itibaren dizilir.

**SELECT *, Adsoyad, Email
FROM Musteriler;**

ya da

**SELECT Adsoyad, Email, *
FROM Musteriler;**

deyimleri, sırasıyla AdSoyad, Email, ID, Sehir, DogumTarihi, Meslek, Cinsiyet alanlarını listeleyecek ve alan tekrarı olmayacak.

**SELECT Adsoyad, Email, Musteriler.*
FROM Musteriler;**

kullandığınızda ise AdSoyad ve Email alanları tekrar edecekti.
Buraya kadarki sorgular için örnek dosyada K grubundaki sorguları inceleyin.

Sorguların performansı için, gerekli olmayan veri alanlarını sorguya dahil etmemek daha iyidir. Örneğin "SELECT * FROM Musteriler" gibi bir deyim yerine tüm alanları listelemeyip sadece gerekli alanların seçildiği "SELECT AdSoyad, Email FROM Musteriler" gibi sorgular yazmanız önerilir.

[Sonraki sayfa](#)

Serkan Şahinoğlu, PC Magazine, Aralık 2001

<http://www.eskidefterler.com>

- veritabanı özel -

Gelişmiş Sorgular

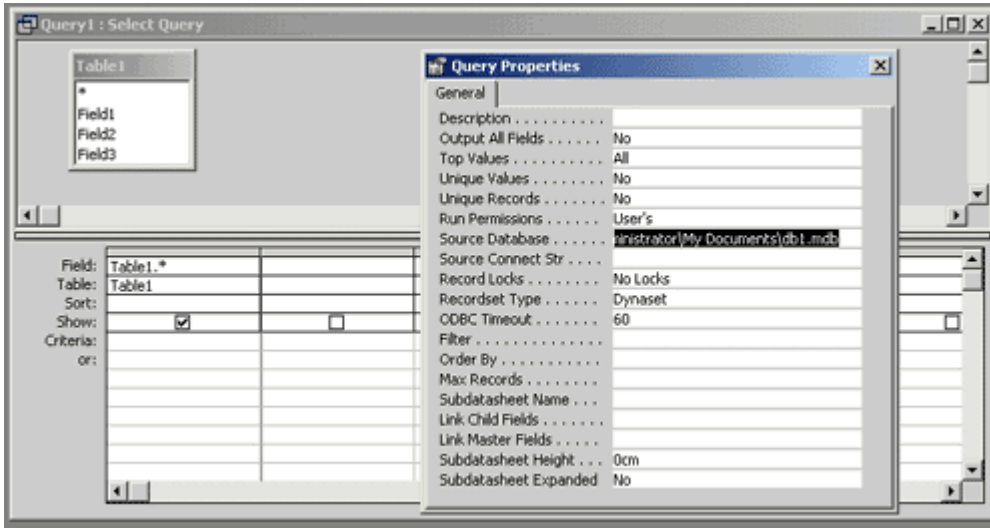


Sayfa2

[Önceki sayfa](#) - [Sonraki sayfa](#)

BAŞKA VERİTABANLARINI SORGULAYIN

Sorgu Sihirbazı başka Access dosyalarından sorgu yapılmasına olanak veriyor. Sorgu tasarımı iken Query Properties penceresini açıp SourceDatabase özelliğine konum belirterek bir dosya ismi yazın. Ne yazık ki buraya dosyanın yolunu el ile girmek gerekiyor.



Bu özelliği ayarladıktan sonra artık tüm işlemler seçtiğiniz dosya üzerinde yapılacak. Görünüm > Tablo Göster (View > Show table) menüsünü çalıştırdığınızda bu dosya içindeki tabloları göreceksiniz.

**SELECT Table1.*
FROM Table1 IN 'C:\My Documents\db1.mdb';**

Başka bir dosyayı tek başına sorgulamak yerine aktif dosyayla beraber sorgulama yapmak istiyorsanız Query Properties'teki Source Database özelliğini boş bırakmak gerekiyor. Sorgu Sihirbazı bu işlemler için hazırlık yaparken maalesef kolaylıklar sunmuyor.

Görünüm > SQL Görünümü (View > SQL View) ile SQL kod sayfasını açıp şu şekilde bir kod yazın:

**SELECT Table1.*
FROM [C:\My Documents\db1.mdb].Table1;**

Görünüm menüsünden tasarım görünümüne geçip Table1 tablosunu seçin ve özellikler penceresindeki Source değerine bakın. Sorgunun özellikler listesindeki seçeneği kullanmayıp bu şekilde farklı dosyalardan sorgulama yapılabilir.

**SELECT Musteriler.ID, Musteriler.AdSoyad, Table1.Field2, Table1.Field3
FROM Musteriler
INNER JOIN [C:\My Documents\db1.mdb].Table1 ON Musteriler.ID = Table1.Field1;**

Sorgu Sihirbazı'nda sadece Access dosyaları mı sorgulanıyor? Hayır. Paradox, Dbase ve Excel dosyalarını doğrudan sorgulamak mümkündür.

Bilgisayarınızda Delphi ya da başka bir Borland ürünü kurulu ise C:\Program Files\Common Files\Borland Shared\Data klasörü altında bir deneme veritabanı olabilir. Bu klasörün içinde Paradox ve Dbase tabloları olması gerekir.

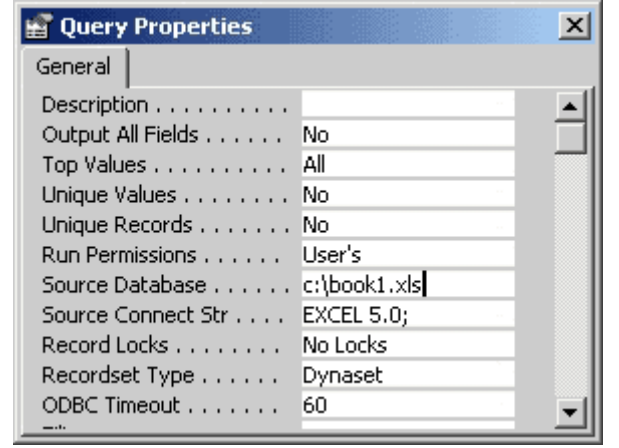
Sorgunun Source Database özelliğine bu klasörün adını ve Source Connect Str özelliğine Paradox 4.x; yazın. View > Show Tables menüsünü çalıştırdığınızda bu klasördeki Paradox tabloları listelenecektir. Bu tabloları kendi sorgularınızda istediğiniz şekilde kullanmanız mümkündür.

Dbase tabloları için ise Source Database alanını boş bırakıp Source Connect Str'ye

dBASE IV; Database=C:\Program Files\Common Files\Borland Shared\Data;

yazın.

Excel dosyasını sorgulamak için ise Source Database özelliğine sorgulanacak Excel dosyasının yolunu, Source Connect Str'ye EXCEL 5.0; yazmak yeterli olur. Örnek dosyadaki P grubu sorgularını inceleyin.



SQL SERVER'İ SORGULAMA

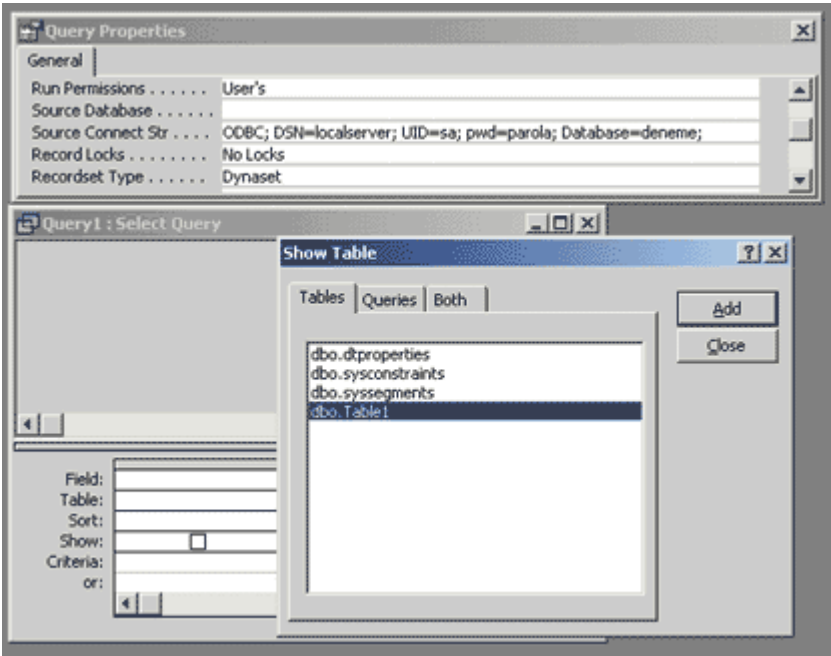
Source Database'in altında Source Connect Str özelliği var. Kimi durumlarda bu ikisi aynı anda kullanılamayabilir. Source Connect Str özelliği ODBC de dahil dış bir veritabanında çalışmayı sağlar. ODBC'de tanımladığınız bir veritabanını buraya aşağıdaki şekilde ekleyin.

ODBC;DSN=odbcadi;UID=kullanıcı adı;PWD=parola;DATABASE=veritabanı;

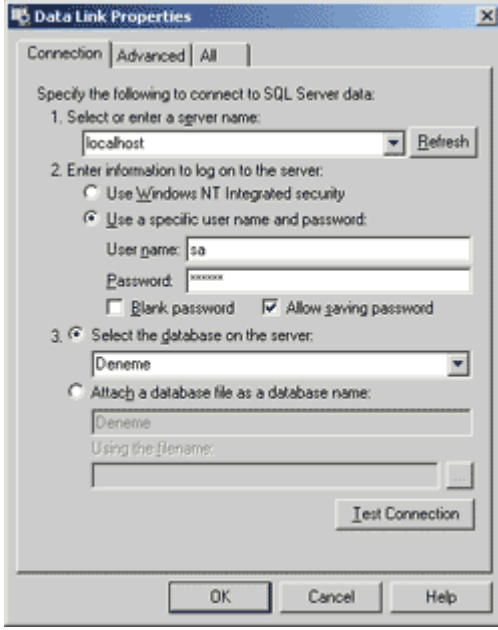
ya da

ODBC;DSN=odbcadi;

gibi bir değer girebilirsiniz. ODBC'de yaptığınız ayarlar ve buradaki kod doğruysa o ODBC üzerinde sorgu yapabileceksiniz.



SQL Server'la çalışmanın daha kolay yolu ise Access'in ADP dosyalarını kullanmak. File > New (Dosya > Yeni) menüsünden Project - New Database komutu ile bir ADP dosyası oluşturun.



ADP proje dosyaları SQL Server ile çalışmayı çok kolaylaştıran güçlü araçlar sunar. SQL Server'daki Enterprise Manager'dan biraz farklı olarak tasarım ve analiz için gelişmiş özellikler barındırır. Enterprise Manager ise SQL Server'ın yönetimi konusunda yetkindir.

ADP dosyası ile SQL Server'daki Table, View, Trigger, Stored Procedure, Diagram (RelationShip) nesnelerini düzenleyebilir, bunları Access form ve raporları ile bağlayıp kendi programınızda kolaylıkla kullanabilirsiniz.

Gerektiğinde File > Connection (Dosya > Bağlantı) menüsü ile bağlantıyı değiştirebilirsiniz.

Resimdeki Data Link Properties başlıklı pencerede Localhost kendi makinemde kurulu SQL Server'ı işaret ediyor. Kullanıcı adı ve parola bilgilerini girip bağlanmak istediğiniz veritabanını seçtikten sonra Test Connection ile bağlantıyı deneyin.

Veritabanı sunucuları arasında yapılan karşılaştırmalarda kaydetme veya sorgulama hızları arasındaki farkların ön planda tutulduğunu görüyoruz. Çok yüksek hızların veya diğer bazı şeylerin gerçek zorunluluk olmadığı durumlarda, veritabanları arasında yapılacak tercihler için önemli etkenlerden biri de Access'teki ADP dosyaları gibi yaygın ve güçlü yardımcı araçların var olmasıdır. Kullanılmayan güç ve hız sahibi olma ile çok kullanılır ve yetenekli araçlara sahip olma arasında yapılacak tercihler, fiyatlar arasında yapılacak tercihler kadar önemlidir kanaatimce.

[Önceki sayfa](#) - [Sonraki sayfa](#)

Serkan Şahinoğlu, PC Magazine, Aralık 2001

<http://www.eskidefterler.com>

- veritabanı özel -

Gelişmiş Sorgular



Sayfa3

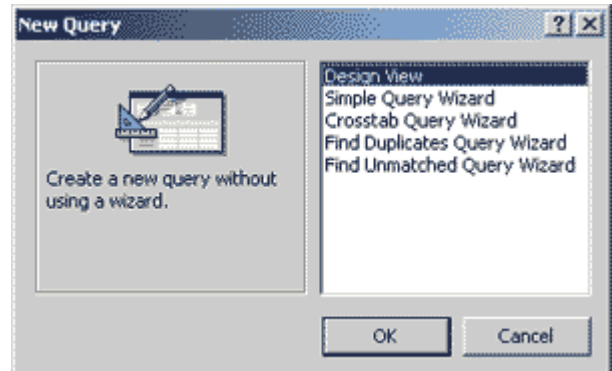
[Önceki sayfa](#)

GELİŞMİŞ SORGULAR

Access'teki sorgu hazırlama sihirbazları bazı sorguları yapmanıza yardımcı olabilirler. Yeni sorgu hazırlama anında karşınıza gelen pencereden istediğinize uygun olanını seçmelisiniz.

Sorguyu kendiniz hazırlayacaksanız listeden Design View'i seçin. Listede bulunan diğerleri çarpaz sorgu, çoğaltılmış alanları bulan sorgu, bir tabloda olup diğer tabloda olmayan kayıtları bulan sorgu türleri yapmayı sağlar. Sorgunuzu kendiniz hazırlamayı düşünmüyorsanız bu sihirbazlar size çok yardımcı olabilirler.

Standart SQL dili, veritabanında gelişmiş sorgular yapmak için birçok komut içerir. Bu komutlar birçok işlemi kolaylaştırır. Örneğin kayıtları düz bir liste şeklinde sıralamak için seçim (select) sorguları kullanılıyor. Sadece listelemek için değil, kayıt ekleme, silme, değiştirme, grupta, sayma, birleştirme gibi sorgular da



oluşturulabiliyor.

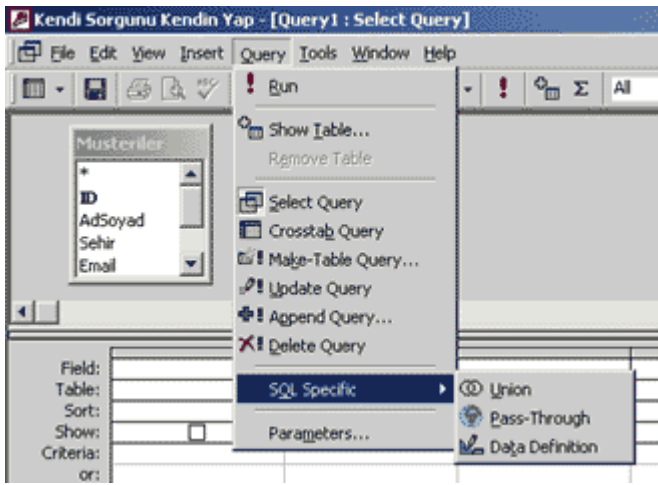
SQL sorgularını kullanarak birkaç tabloyu birleştirip veri alanları gruplayabilir, adetleri, ortalama değerleri, en büyük ve en küçük değerleri, bir tabloda olup diğerinde olmayan kayıtları, hangi kayıtların çoğaltılmış olduğunu bulabilir ve birçok istatistik amaçlı bilgiyi elde edebilirsiniz.

SQL dili sadece verileri analiz etmek için kullanılmıyor tabii ki. Veritabanında yönetim amaçlı değişiklikler de yapılabilir. Tablo oluşturma, değiştirme ve silme gibi işlemler de yapılabilir örneğin.

Access bunlara ek olarak kendine özgü bazı yeteneklerini sorgulara katabilir. VBA'daki bazı işlevlerle birlikte modüller içinde hazırladığınız işlevleri Access sorgularının içinde kullanabilirsiniz.

Örnek dosyaya eklediğim MainModule modülünü inceleyin. Oradaki AdBul ve SoyadBul işlevlerini M1 sorgusunda müşteri alanındaki AdSoyad alanından ad ve soyad bilgisini ayırmak için kullandım.

Sorgu işlemleri için Access 9 farklı türde sorgu nesnesi sunuyor.



- Select (Seçim sorgusu)
- Crosstab (Çarpaz sorgu)
- Make Table (Tablo yapma sorgusu)
- Update (Güncelleştirme sorgusu)
- Append (Ekleme sorgusu)
- Delete (Silme sorgusu)
- Union (Birleştirme sorgusu)
- Pass Through (Doğrudan geçişli sorgu)
- Data Definition (Veri tanımlama sorgusu)

Seçim sorguları genel amaçlı listeleme işlemleri için kullanılıyor. Çarpaz sorgular ise veri alanlarını gruplayıp çarpazlamak ve satır-sütunlara dönüştürmek için çok faydalıdır. Müşteri tablonuzdaki şehir ve cinsiyet bilgilerini gruplayıp her bir şehirdeki kadın ve erkek müşterilerinizin sayısını görebilirsiniz.

Tablo yapma ya da diğer değiştirme sorguları çalıştırılarak veritabanında ilgili işlemler yaptırılır. Aşağıdaki kodlar sırasıyla tablo yapma, güncelleştirme, ekleme ve silme sorgularına örnektir. Dosyadaki örnekler M grubunda.

```
SELECT AdSoyad, Sehir  
INTO YeniTablo  
FROM Musteriler;
```

Müşteriler tablosundaki AdSoyad ve Sehir alanları seçilip yeni oluşturulacak YeniTablo adlı bir tabloya yerleştirilecek. Güncelleştirme sorgusu ise aşağıdaki şekilde.

```
UPDATE Musteriler  
SET (Meslek = "Talebe")  
WHERE (Meslek="Öğrenci");
```

Ekleme sorgusu:

```
INSERT INTO Sehirler ( Sehir )  
SELECT Musteriler.Sehir  
FROM Musteriler LEFT JOIN Sehirler  
ON Musteriler.Sehir = Sehirler.Sehir  
WHERE (((Sehirler.Sehir) Is Null));
```

Silme sorgusu:

DELETE FROM Musteriler WHERE ID=10;

UNION VE SUB-QUERY

Birden çok seçim sorgusunu birleştirmek gerektiğinde Union sorguları devreye girer. Farklı tablolarda bulunan listeleri birleştirmek için Union sorguları idealdir.

```
SELECT * FROM Musteriler1  
UNION  
SELECT * FROM Musteriler2
```

Union sorgularda en önemli kural birleştirilecek kümelerde aynı sıraya gelen alanların aynı türde olması gerektiğidir. Kümelerin toplamında varsa tekrar edilmiş kayıtlar çıkarılır. Fazlalıkların aynı tabloda olması durumunda da bu geçerlidir ve ilk alandan itibaren sıralı listeleme sözkonusudur.

Farklı adlardaki veri alanlarına sahip tabloları birleştirmek istediğinizde alanları AS deyiimi ile yeniden isimlendirip birleştirmek mümkündür. Kayıtların hangi tablodan geldiğini anlamak için ise yeni bir alan oluşturmak yeterli. Bu alana tablonun ismini yazabilirsiniz örneğin.

```
SELECT "Musteriler1" AS TabloAdi, AdSoyad, Sehir FROM Musteriler1  
UNION  
SELECT "Musteriler2" AS TabloAdi, MusteriAdi AS AdSoyad, Sehir FROM Musteriler2
```

Eğer kişi adları farklı alanlardan oluşuyorsa bunları tek alanda birleştirebiliriz.

```
SELECT Ad & " " & Soyad AS AdivSoyadi, Sehir
```

UNION ile oluşturulmuş bir sorguyu sıralamak için başka bir yöntem kullanmak gerekiyor. Zira SELECT cümlelerinde kullanacağınız ORDER BY blokları bir işe yaramayacaktır. İççe SELECT cümleleriyle bunu halledebilirsiniz. Önce şunu inceleyin:

```
SELECT * FROM (SELECT * FROM (SELECT * FROM Musteriler));
```

Bu cümle verileri sorgulamak için mantıklı olmasa da örnek olması açısından bu şekilde verdim. İççe sorgular genellikle çok katmanlı sorgulamalar gerektiğinde işe yararlar.

Geçen aylardaki Access XP yazısında karşılaştırmalı spesifikasyonlar tablosunu hatırlıyor musunuz? Orada tek kümede iççe kullanılabilecek sorgu adedinin 50 olduğu yazıyordu. Bu yazıyı yazarken hemen üstteki örnek cümleye benzer bir kod denedim. İççe 50 SELECT'in olduğu SQL cümlesi sorunsuz çalıştı. 51 olduğunda ise hata verdi. Örnek dosyadaki L grubu sorgularını inceleyebilirsiniz.

Bu örnekte alt seviyedeki SELECT cümlesinin FROM grubunda olması gibi, alt SELECT cümlelerini doğrudan veri alanları ile karşılaştırmak da mümkündür.

```
SELECT Musteriler.*  
FROM Musteriler  
WHERE ( DogumTarihi > (SELECT AVG(DogumTarihi) FROM Musteriler) )  
ORDER BY DogumTarihi DESC;
```

L3 sorgusunda yaş ortalamasından daha genç olan müşteriler listeleniyor. WHERE satırında DogumTarihi alanı yeni bir SELECT cümlesi ile karşılaştırılıyor. Bu SELECT cümlesi ortalama doğum tarihini buluyor.

Burada en önemli kural alt sorgunun yalnızca 1 kayıt getirmesinin gerektiği. Getirilen bu değer sorgudaki doğum tarihi alanı ile karşılaştırılıyor.

PASS THROUGH VE DATA DEFINITION

Pass-Through sorgular ise bağılı olunan veritabanına doğrudan komutlar göndererek hızlı sonuç üretirler. Örneğin ODBC ayarlarında bir SQL Server'a bağlantı açmışsanız SQL Server'da çalıştırdığınız komutları Access'ten gönderebilirsiniz. Bu, SQL sunucusu üzerindeymiş gibi çalışmayı sağlar.

Yeni bir sorgu oluşturup Query > SQL Spesific > Pass Through menüsünü seçin. View > Properties komutu ile Özellikler penceresini açın. ODBC Connect Str özelliğini aşağıdakine benzer şekilde ayarlayın.

```
ODBC;DSN=LocalServer;UID=sa;PWD=parola;DATABASE=Deneme;
```

Hemen altındaki ReturnRecords özelliğini No olarak belirtin. Sonrasında SQL Server üzerindeymiş gibi çalışabilirsiniz. Access'te tanınmayan şu komut SQL Server'da yeni bir View oluşturmamı sağladı:

```
CREATE VIEW dbo.VIEW1  
AS  
SELECT ID, Deneme  
FROM dbo.Table1
```

Doğrudan geçişli sorgular aynı zamanda kayıt listesi de getirebilir. ReturnRecords özelliğinin Yes olarak ayarlanması gerekir. Bu yöntem SQL Server'a bağılı Access programınızda yaptığınız sorguların hızını artırır. Zira normalde yapılan sorgularda sonuçlar ham veriler alınıp işlendikten sonra elde ediliyordu. Burada ise ham veriler SQL Server'da işlenip sadece sonuç verileri Access'e gönderiliyor.

Bir de Data Definition sorguları vardır ki bunlar veritabanı üzerinde yapılandırma işlemleri için kullanılırlar. Yeni bir tablo oluşturma, bir tablonun alanlarını değiştirme, indeksleri yapılandırma gibi işlemler yapılabilir.

```
CREATE TABLE Tablo1 (Ad CHAR(10), Soyad CHAR(10), DogumTarihi DATETIME, Sira NUMERIC );
```

```
DROP TABLE Tablo1;
```

```
CREATE INDEX idxAd ON Tablo1(Ad);
```

SQL'DE TARİH SORUNSALI VE PARAMETRELER

Bu dizinin ilk yazısında tarihlerin aslında sayı olduklarından ve SQL kodlarında tarih değerlerinin sayısal karşılıklarının kullanılabileceğinden bahsetmiştik. (Daha fazla bilgi için derginin Mayıs 2000 tarihli sayısına bakınız.)

```
MsgBox CLng(Date)
```

VBA ve ASP'de kullanabileceğiniz CLng işlevi ile tarih bilgisinin sayısal karşılığını bulup bunu SQL kodunda kullanmak olanaklıdır.

```
Dim rs  
Set rs=CreateObject("Adodb.RecordSet")  
rs.open "SELECT * FROM Tablo1 WHERE SonTarih >=" & CLng(Date), [connection]
```

RecordSet'in Open eyleminde kullanılacak SQL kodu aşağıdakine benzer olacaktır.

```
SELECT * FROM Tablo1 WHERE SonTarih >= 37207
```

Programlama aşamasında tarih probleminin kurtulmanın bir başka yolu da sorgularda parametre kullanmak. Örnek dosyada da bulabileceğiniz aşağıdaki kodu inceleyin.

```
Sub SorguyaParametreGonder( )
```

```
Dim komut As New ADODB.Command  
Set komut.ActiveConnection = CurrentProject.Connection  
komut.CommandType = adCmdTable  
komut.CommandText = "[01 - Tarih parametresi]"
```

```
`Sorgunun ismi
Dim parametre1 As New ADODB.Parameter
parametre1.Name = "tarih1"
parametre1.Direction = adParamInput
parametre1.Type = adDate
parametre1.Value = #1/1/1970#
```

```
Dim parametre2 As New ADODB.Parameter
parametre2.Name = "tarih2"
parametre2.Direction = adParamInput
parametre2.Type = adDate
parametre2.Value = #1/31/1970#
```

```
`Parametreleri komuta ekle
komut.Parameters.Append parametre1
komut.Parameters.Append parametre2
```

```
Dim rs As New ADODB.Recordset
Set rs = komut.Execute
```

```
MsgBox rs.GetString
```

End Sub

Bu yordam parametrelili O1 sorgusuna tarih parametrelerini gönderiyor ve sonucu elde ediyor.

Activex kütüphaneleri her platformda aynı şekilde kullanıldığına göre yukarıdaki koddaki ADO yöntemlerini ASP'de de kullanabilirsiniz.

Projenizde ADO yerine eski DAO'yu tercih ediyorsanız aşağıdaki daha kolay kodu deneyin.

```
Sub SorguyaParametreGonder_DAO_kullan()
Dim db As DAO.Database
Set db = CurrentDb
Dim qd As DAO.QueryDef
Set qd = db.QueryDefs("O1 - Tarih parametresi")
```

```
`Sorgunun ismi
qd.Parameters("Tarih1") = #1/1/1970#
qd.Parameters("Tarih2") = #1/31/1970#
```

```
Dim rs As DAO.Recordset
Set rs = qd.OpenRecordset
rs.MoveLast
```

```
MsgBox rs.RecordCount
```

End Sub

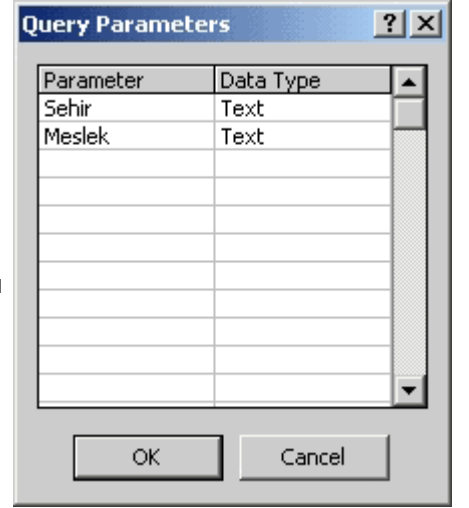
Parametre nedir? Yazının buraya kadarki kısmı bu sorunun cevabı bulmak için yeterli değil. Parametreleri 'belli bir kalıba gönderilen dış değerler' olarak tanımlayabiliriz. Access sorgularındaki parametreler de sorgu kalıbına gönderilen dış değerleri ifade ederler. Sorgunun koduna köşeli parantez içinde yazdığınız ve sorguda tanımlı olmayan isimleri Access parametre olarak varsayar. (Uzun tablo ya da veri alanı adlarında da köşeli parantez kullanmanız gerekebilir.)

SELECT * FROM Musteriler WHERE AdSoyad=[pAdSoyad]

Parametreleri tablo ya da veri alanlarından farklı olarak isimlendirmek gerekiyor. Ben genel olarak parametre alanlarının başına p harfi ekliyorum. Eğer sorguya Parameters deyimi ekliyorsanız farklı isimlendirme zorunluluğu yoktur.

Query > Parameters menüsü ile açtığınız parametreler listesine yeni parametreleri ekleyin. Bunlar veri alanlarının isimleri ile aynı olsa da köşeli parantez içinde kullandığınız deyimlere değer gönderirler. Bu listedekilerle kod içinde kullandığınız parametrelerin birbirini karşılamaları zorunlu değildir. O2 sorgusunu aşağıdaki gibi düzenledim.

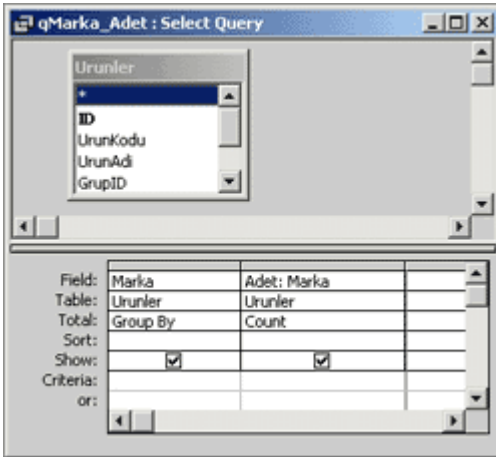
PARAMETERS Sehir Text (255), Meslek Text (255);
SELECT Musteriler.*
FROM Musteriler
WHERE ((Sehir=[Sehir]) AND (Meslek=[Meslek]));



Yukarıdaki kodu ve diğer SQL kodlarını Access sorgularının SQL görünümüne geçerek kod sayfasına yazabilirsiniz. Buraya yazdıklarınıza göre tasarım görünümünde karşınıza yeni bir görüntü çıkarabilir.

GRUPLA - BİRLEŞTİR - VE SAİR

Geçen ayki ASP yazısında SQL'deki GROUP BY ve JOIN deyiminin beraber iyi iş çıkardıklarını söylemiştim. Eski Defterler sitesindeki örnek ürünler sayfasını incelediyseniz orada gruplara göre adetler listesi olduğunu görmüşsünüzdür. O listeyi GROUP BY deyimi ile kolay bir şekilde oluşturmuştum.

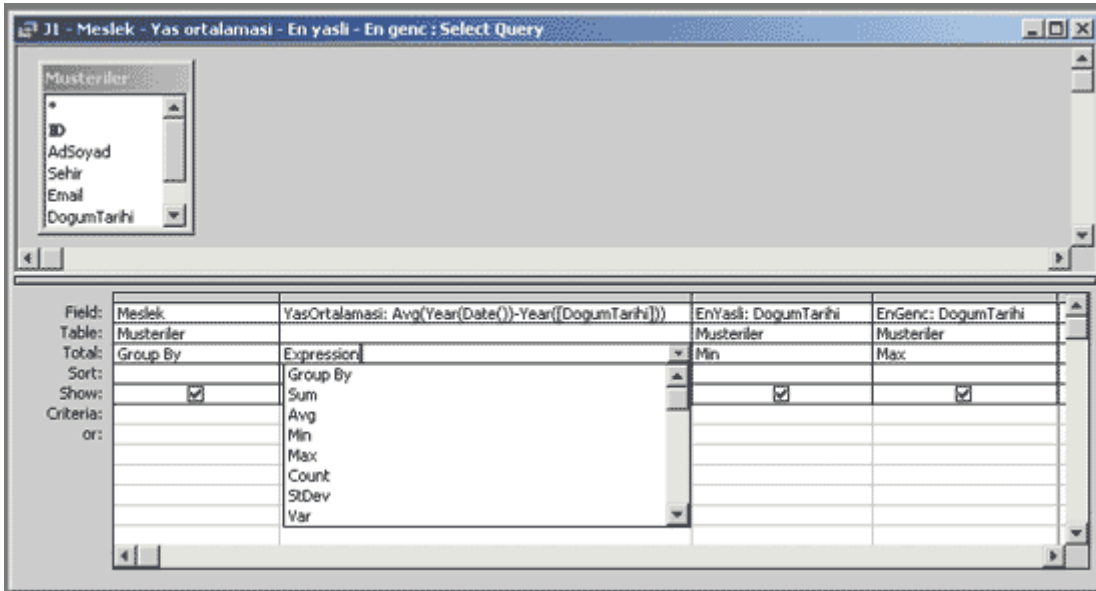


SELECT Marka, Count(Marka) AS Adet
FROM Urunler
GROUP BY Marka;

Yeni bir sorgu oluşturup ürünler tablosunu ekledim ve Marka alanını alt tarafa iki kez yerleştirdim. Görünüm > Toplamlar (View > Totals) menüsü ile Group By özelliğini aktif ettikten sonra altta ikinci sütundaki Marka alanının Total (Toplam) hücrelerini Count (Adet) olarak ayarladım. Alan adının başına "Adet:" ekleyerek ismini değiştirdim.

Group By deyimindeki alanların tamamına göre gruplama yapılır. Yukarıdaki örnekte sadece Marka alanına göre gruplama söz konusu. Bunu yaptıktan sonra her bir grup içinde ayrı ayrı incelemeler yapılıyor gerekli değerler hesaplanıyor.

Bu deyimler GROUP BY olmadan da kullanılabilir. Kayıt kümesinin tamamı bir grup olarak kabul edilir.



Aşağıdaki iki deyimın tasarım görünümünde biraz farklılıkları vardır.

SELECT Count(ID) AS Adet FROM Musteriler;

SELECT Count(*) AS Adet FROM Musteriler;

Tasarım anında Totals özelliği aktifse aşağıdaki listede bulunan Musteriler.* sütunu için hata iletisi gösterilecektir. Bu deyim varken alanların tümünün listesi üretildiği için Group By özelliği aktif edilemez.

**SELECT Sehir, Count(*) AS Adet
FROM Musteriler
GROUP BY Sehir;**

Count deyimi group by dahilindeki her bir grup için kayıt adedini bulur. Ama Null değerleri bu sayıya dahil etmez. Amacınız tüm kayıtları saymak ise * kullanın.

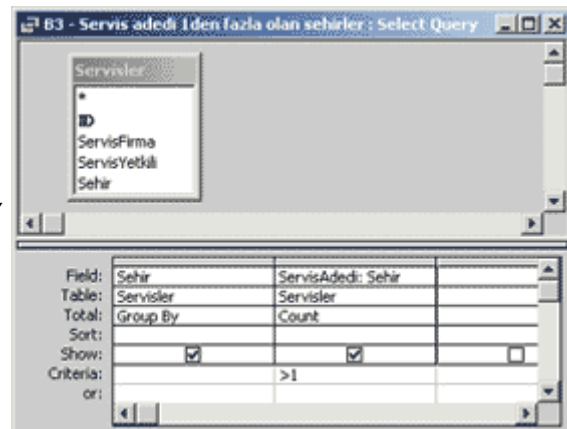
Yukarıdaki kod her bir şehirdeki müşteri adedini listeliyor. Şehirlerde email adresi olanların sayısını ayrı olarak listelemek istediğinizde Count'un parantezleri içindeki * işareti yerine Email yazın. Bunun sebebi yukarıda belirttiğim gibi Count deyiminin Null olan alanları sayıya dahil etmemesi.

**SELECT Sehir, Count(*) AS ToplamAdet, Count(Email) as EmailOlanlar
FROM Musteriler
GROUP BY Sehir;**

Group By'dan yararlanarak ilginç sorgular üretebiliriz. Yetkili servis adedinin birden fazla olduğu şehirleri bulmak için B3 sorgusunu oluşturdum. Resimde de gördüğümüz gibi Count ile tanımlı sütunun koşul bölmesine >1 yazmak yeterli oluyor.

Örnek dosyada A-J gruplarındaki sorgularda GROUP BY ve JOIN deyimleri kullanılıyor.

Birkaç tablodaki alanları birleştirip listelemek gerektiğinde JOIN deyimi devreye girer. JOIN kullanılmadan yapılan sorgular ekli tüm tabloların kayıt adetlerinin çarpımı kadar kayıt oluşur.



SONUÇ YERİNE

Konu bitmedi. JOIN ile ilgili ayrıntıları ve bu ay yazamadığım birçok ayrıntıyı gelecek yazımda anlatmayı düşünüyorum.

Bu ay ne ASP ne de Acces yazısı yazmak yerine veritabanı yazısı hazırladım. Bu, diğer dizilerin sonlandırıldığı anlamına gelmiyor tabii. Posta kutumda cevap bekleyen soruları ancak uzun bir sürede cevaplandırabilmeminkiyle aynı sebepten dolayı, ASP yazısını hazırlamadım. Zira vaktimi en çok alan yazılardan biriydi bu. Access, ASP ve Programcılık üçgeninde bazı şeyleri aktarmaya devam edeceğim.

Bu ay hazırladığım dosyayı www.eskidefterler.com adresinden yükleyebilirsiniz.

AÇILIŞ EKRANINI DEĞİŞTİRİN

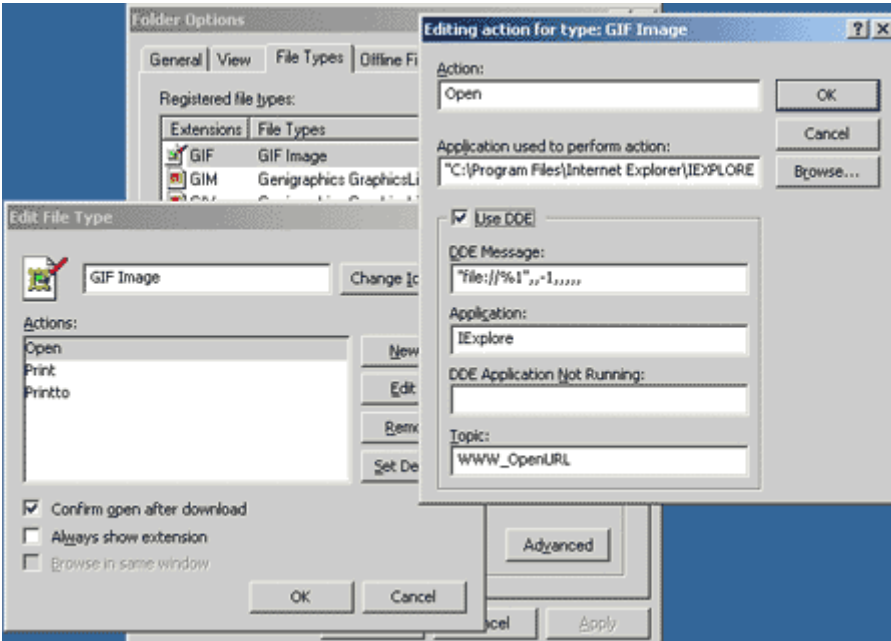
F@T@LITY adlı okurumuz Access'in açılışındaki ekranı değiştirmek için güzel bir ipucu göndermiş. Access dosyanızın bulunduğu yere aynı adlı bir BMP resim dosyası koyduğunuzda dosyayı açarken Access'in açılış ekranı yerine bu BMP dosyası kendi açılış ekranınız olarak gösteriliyor.

INTERNET EXPLORER YENİ PENCERE AÇMASIN

Bilgisayarınızda Internet Explorer kurulu ise bazı dosyalar çift tıklatıldığında IE ile açılır çoğunlukla. HTM dosyaları, gif ya da jpeg resim dosyaları için bazı bilgisayarlarda bu geçerlidir. Bu tür dosyaları çokça inceliyorsanız bazan IE herbiri için yeni bir pencere açıyor. Eğer dosyaların çift tıklatıldığında Internet Explorer'ın aynı pencereyi kullanmamasını istiyorsanız aşağıdaki adımları takip edebilirsiniz.

Windows Gezgini'nde iken Tools > Folder Options (Araçlar > Klasör Seçenekleri) menüsünü açın. File Types (Dosya Türleri) sekmesinde dosyanın uzantısını bulun. Advanced (Gelişmiş) düğmesini tıklatın. Açılan pencereden Open (Aç) komutunu seçip Edit (Düzenle) düğmesini tıklatın.

Use DDE (DDE kullan) seçeneğini aktif hale getirip alt taraftaki DDE Message kutusuna (birinci kutu) "file://%1",-1,,,, yazın. İkinci kutuya IExplore, dördüncü kutuya da WWW_OpenURL yazın.



Bundan sonra bu tür dosyaları açtığınızda her seferinde yeni bir Internet Explorer penceresi açılmayıp dosyalar aynı pencerede görüntülenecek.

DOSYA:

YARDIMCI İŞLEVLER



Bu ay, dokunulmazlığı olan problemlere dokunup gizlenmiş ipuçlarına kalem savurduk. Müzmin sorunlara uzman çözümleri Access Hikayeleri'nde.

Geçen aylar her zamanki gibi yorucuymdu. Access sayfalarının fazla olması birçok okurumuzun isteğini karşılamış olsa da benim açımdan yorulmayla birlikte bazı eksiklikleri de beraberinde getirdi.

Access XP yazısından bir önceki yazıyı, yayımlandıktan sonra gözden geçirdiğimde programların dağıtımıyla ilgili karamsar bir tablo çizdiğimi farkettim. Access dosyalarına verilen parolaların Internet'te bulunabilen bazı programlar tarafından çözülebildiğinden bahsetmiştim. Bunu şimdi biraz aydınlatmak gerektiğini düşünüyorum.

Dosyanıza parola verdiğinizde bu parola sadece dosyanızın yetkili açılabilir ya da kullanılabilir olmasını sağlayacaktır. MDE'den MDB'ye dönüştürme gibi bir uygulama hiç varolmayıp parola korumalı MDE dosyaları da açılıp çalıştırılabilir ve MDB'deki parola mantığı bunda da aynen geçerlidir. MDE dosyalarında formlar, raporlar, kodlar görüntülenemez ve değiştirilemez. Tablo ve sorgular ise değiştirilebilir.

Veritabanı için tanımlanan parola koruması sadece dosyanın yetkili açılmasını sağlayabilir. Sonrası için ise güvenlik ayarlarının tanımlandığı bölümde kullanıcı ve grup yetkilendirmesinin yapılması yeterli olacaktır.

YENİ İHTİYAÇLARA YENİ ÇÖZÜMLER

Bu ay yeni hazırladığım yardımcı işlevlerin sizin de işinize yarayacağından eminim. Sadece ihtiyacım olduğu zamanlarda yaptığım üzere geçen ay çalıştığım bir proje için bazı işlevler yazmak durumunda oldum.

Tasarladığım formlar yüzlerce veri ve formül alanı içeriyordu. Sabitler ve hesaplanan formüller sorgularda işlenip formda gösteriliyor. Ama müşteriye lazım olan yeni formülleri de forma yerleştirmek gerekiyor. Onlarca form ve her bir formda ortalama 300 adet sabit ve formül alanları. Bu durumda bir süre sonra sorgulardaki bazı alanlar gözden kaçıp forma yerleştirilmeyebiliyor.

Bunun için NotUsedFields adlı bir işlev yazdım. Bu işlev formun kayıt kaynağındaki tüm veri alanlarının formda kullanılıp kullanılmadığını buluyor. Böylelikle yüzlerce kutu arasında kaybolmuyorum. Aynı şekilde iptal edilmiş veri alanlarının da form üzerinde hala varolup olmadığının kontrol edilmesi gerekiyordu. SelectInvalidFieldObjects işlevi de bu işi yapıyor.

Hazır başlamışken bir veri alanını arama zahmetinden kurtarıp form üzerinde doğrudan gösteren FindField, tab duraklarını ayarlarken büyük kolaylıklar sağlayan SetTabOrder ve benim en çok hoşuma giden SelectControls işlevlerini yazdım. Bununla form üzerindeki bütün metin kutularını, etiketleri, düğmeleri ve hemen tüm diğer nesneleri bir anda seçmek mümkün. Shift tuşu basılı olduğunda da buna uyup eski seçimleri kaldırıyor.

Bu işlevlerin bazılarında çalışma kipinde olup olmadığımızı tespit etmek gerekiyordu. Form nesnesinin CurrentView özelliği bunu bize veriyor. Sıfır ise tasarım kipinde demektir. 1 olduğunda ise çalışma anı demektir.

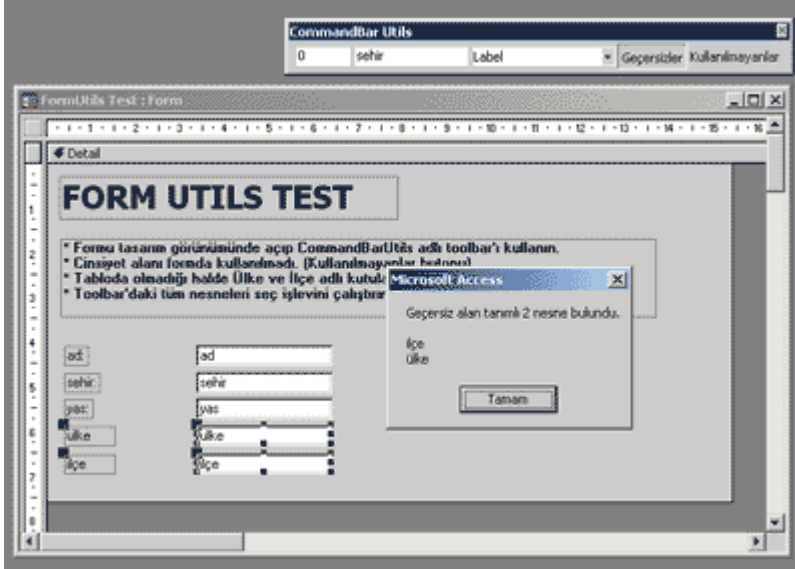
Yaptığım projede herbiri ortalama 500 nesne içeren 20 civarında form ve rapor bulunuyor. Tablo ve sorguların herbiri de ortalama 200 veri alanı içeriyor.

Formüllerin anlık olarak hesaplanması için en iyi yol Excel'deki gibi bağlantılı formül alanları kullanmak. Access'te bunları yapmak Excel'deki kadar kolay sayılır. Ekran üzerinde bir değeri değiştirdiğinizde buna bağlı onlarca formülü

yeniden hesaplabiliyorsunuz. Denetim kaynağı özelliğine bir formül yazmanız yeterli. Fakat formlarda uğraşmak yerine formülleri sorgularda yapmayı yeğledim.

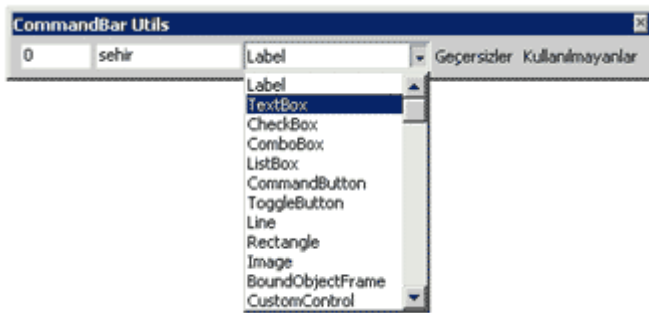
Sorgularda formül tanımladığınızda Excel sayfalarında olduğu gibi bir değer değiştiğinde aynı satırda buna bağlı tüm değerler de anında değişebiliyor. Bunu yapmamın sebebi sorgu nesnelerini arkaplanda açıp grafikler için bazı değerleri seri halinde değiştirip sonuçlar elde etmektir. Bu iş için en iyi yol buydu.

Projemde kullandığım bazı teknikleri bu ay sizlerle paylaşıyorum. Kodların çalıştırılabilir ve kullanılabilir halini Eski Defterler sitesinde bulacaksınız.



MENÜ ARAÇ ÇUBUKLARINI PROGRAMLAYIN

Bu işlevleri kullanışlı hale getirmek için araç çubuklarına yerleştirmek gerekiyordu. Formdaki nesnelere tab numarası vermek için seçili objelerin başlangıç tab sırasını yazabileceğimiz bir kutuya ihtiyaç vardı. Toolbar'da oluşturduğum Utils barına msoControlEdit türünde bir CommandBarControl nesnesi yerleştirdim. Türüne göre tüm nesneleri seçebilmek için açılır liste kutusu (msoControlDropdown) gerekiyordu. Geçersiz ve kullanılmayan veri alanları için de birer CommandBarButton yeterliydi.



Fakat bir sorun komutların uygulanacağı formun belirlenmesiydi. Ne yazık ki, form üzerindeki Properties penceresi açık olduğunda Screen.ActiveForm deyimi geçerli bir form belirtmedi. Bu durumda eğer açık olan birden fazla form var ise aktif form seçilmeliydi. Sadece bir form açık ise aktif olsun ya da olmasın o seçilmeliydi. Hiç form açık değilse uyarı yapılmalıydı. Bunun için GetActiveForm işlevini yazarak her seferinde uzun kontrol kodları yazmaktan kurtuldum.

Tüm bu işlevleri kullanabilmek için sadece bir kez araç çubuğu oluşturup bunun üzerindeki nesnelere yazdığım işlevleri bağlamak yeterli oldu.

CreateCommandBar yordamını bir modül sayfasına yerleştirip yordam içindeki herhangi bir satırda F5 tuşuna basarak onu çalıştırın. Bunu sadece bir kez yapın. Toolbar nesnelerinin kullanılabilmesini Office'te ortak kullanılan Microsoft Office 9.0 Object Library kütüphanesi sağlıyor. Başvurular listesini açıp bunun seçili olmasını sağlayın.

```
Sub CreateCommandBar()
```

```
Dim cb As CommandBar
```

```
'CommandBar varsa bul, yoksa ekle.
Set cb = FindCommandBar(cb_FormUtils, True)

'''SetTabOrder
Dim ed As CommandBarControl
If FindCommandBarControl("SetTabIndex", cb) = False Then
Set ed = cb.Controls.Add(msoControlEdit)
ed.Width = 50
ed.Caption = "SetTabIndex"
ed.TooltipText = "Formdaki seçili nesnelerin TabIndex değerlerini" & _
vbCr & "bu numaradan başlayarak yeniden düzenle"
ed.OnAction = "commandbar_SetTabOrder"
End If

'''FindField
If FindCommandBarControl("FindField", cb) = False Then
Set ed = cb.Controls.Add(msoControlEdit)
ed.Width = 100
ed.Caption = "FindField"
ed.TooltipText = "Veri alanını bul"
ed.OnAction = "commandbar_FindField"
End If

'''SelectControls
Dim combo As CommandBarComboBox
If FindCommandBarControl("SelectControls", cb) = False Then
Set combo = cb.Controls.Add(msoControlDropdown)
combo.Width = 125
combo.Caption = "SelectControls"
combo.TooltipText = "Bu Türdeki Nesneleri Seç"
combo.AddItem "Label"
combo.AddItem "TextBox"
combo.AddItem "CheckBox"
combo.AddItem "ComboBox"
combo.AddItem "ListBox"
combo.AddItem "CommandButton"
combo.AddItem "ToggleButton"
combo.AddItem "Line"
combo.AddItem "Rectangle"
combo.AddItem "Image"
combo.OnAction = "commandbar_SelectControls"
End If

Dim bt As CommandBarButton

'''Geçersizler
If FindCommandBarControl("Geçersizler", cb) = False Then
Set bt = cb.Controls.Add(msoControlButton)
bt.Style = msoButtonCaption
bt.Caption = "Geçersizler"
bt.TooltipText = "Geçersiz alan tanımlı nesneleri seç"
bt.OnAction = "commandbar_InvalidFields"
End If

'''Kullanılmayanlar
If FindCommandBarControl("Kullanılmayanlar", cb) = False Then
Set bt = cb.Controls.Add(msoControlButton)
bt.Style = msoButtonCaption
bt.Caption = "Kullanılmayanlar"
bt.TooltipText = "Formda kullanılmayan veri alanlarını bul"
bt.OnAction = "commandbar_NotUsedFields"
End If
```

```
cb.Visible = True

End Sub
```

Bu modülün ilk halinde çalışırken her seferinde nesneler tekrarlanıyordu. Bunu önlemek için FindCommandBar ve FindCommandBarControl adlı işlevleri yazdım. Bu ikisi ile hem menü çubuğu içindeki nesnelerin olup olmadığını, hem de barın kendisinin var olup olmadığını kontrol edebiliyorum. Ana kodu çirkinleştirmemek için FindCommandBar'da tanımladığım AutoCreate parametresi ile barı otomatik oluşturuyorum.

```
Function FindCommandBar(BarName As String, Optional AutoCreate As Boolean = False) As CommandBar
'CommandBar'ın olup olmadığını bulur.
'Bulamadığında AutoCreate=True ise oluşturur.
On Error GoTo son
Set FindCommandBar = CommandBars(BarName)
Exit Function
son:
Err.Clear
If AutoCreate Then Set FindCommandBar = CommandBars.Add(BarName)
End Function
```

```
Function FindCommandBarControl(ControlName As String, cb As CommandBar) As Boolean
'CommandBarControl nesnesinin olup olmadığını bulur.
On Error GoTo son
Dim ct As CommandBarControl
Set ct = cb.Controls(ControlName)
FindCommandBarControl = True
Exit Function
son:
Err.Clear
End Function
```

CommandBar nesnelerinin çalıştıracağı işlev OnAction özelliği ile belirtiliyor. Bu özelliğe, çalıştırılacak işlevin adını yazmak gerekiyor.

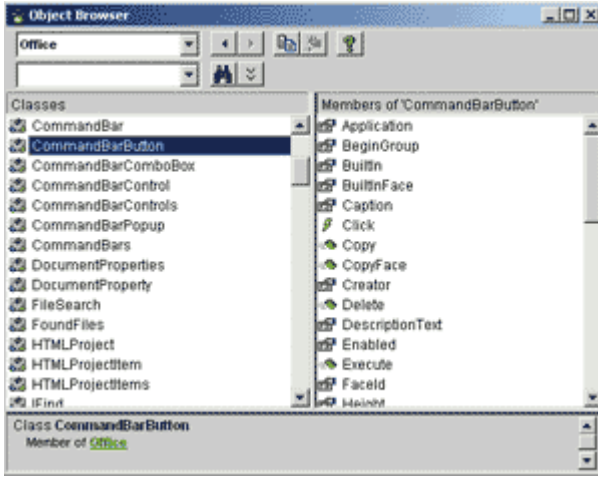
```
Public Function commandbar_SetTabOrder()
Dim f As Form
If GetActiveForm(f) Then
SetTabOrder f, CommandBars(cb_FormUtils).Controls("SetTabIndex").Text
End If
End Function

Public Function commandbar_SelectControls()
Dim t As Integer
t = GetControlType(CommandBars(cb_FormUtils).Controls("SelectControls").Text)

If t = 0 Then Exit Function

Dim f As Form
If GetActiveForm(f) Then
SelectControls f, t, GetKeyState(VK_CONTROL) < 0
'''CTRL basılı ise önceki seçim iptal edilsin.
End If
End Function
```

CommandBar içindeki nesnelerin tamamı CommandBarControl tipindedir. Fakat bazıları özelleştirilebilir olduğundan bunlar için yeni tipler tanımlanmış. Örneğin CommandBarComboBox nesnesi aynı zamanda bir CommandBarControl'dur ama bundan daha farklı metot ve özellikler içerir. Object Browser penceresinde gördüğünüz gibi Button, Popup gibi nesneler de var.



SEÇ BUL DÜZENLE VS.

Asıl işleri yapacak işlevlerimiz de hazır. Aktif formu bul, tab duraklarını düzenle, belli türdeki tüm nesneleri seç, veri alanını formda bul, geçersiz veri alanlarını bul, formda kullanılmayan veri alanlarını bul. Tabii bu yazıda bunların tamamı yer almıyor.

Function GetActiveForm(F As Form) As Boolean
'Aktif olan ya da tek başına açık olan formu getirir.
'Sonuç form F değeridir. İşlev True ise geçerli
'bir form bulunmuş demektir.

```
GetActiveForm = False
If Forms.Count = 0 Then
MsgBox "Aktif form bulunamadı."
Exit Function
ElseIf Forms.Count = 1 Then
Set f = Forms(0)
Else
On Error Resume Next
Set f = Screen.ActiveForm
If Err.Number = 2475 Then
MsgBox "Aktif form bulunamadı."
Err.Clear
Exit Function
End If
On Error GoTo 0
End If
GetActiveForm = True
End Function
```

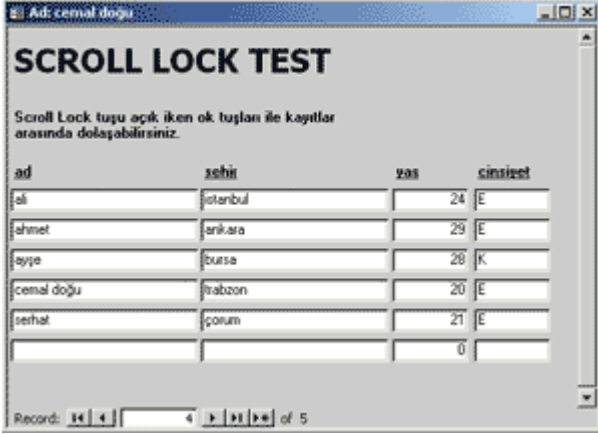
ACCESS HİKAYELERİ

Bu ay yardımcı işlevlere el attık. Yazının konusu form düzenlerken kullanabileceğiniz işlevler içeriyordu. Yeni yazdığım üç yararlı başka işlev de bu bölümde.

Ok tuşları ile formdaki kayıtlarda hareket edin. Tutumlu işlev ile yazıcıya gönderdiğiniz formları yazdırırken gereksiz toner masrafını önleyin. Çalışma anında tüm nesneleri belli bir oranda yeniden boyutlandırın.

AYIN TUŞU

Klavyedeki Scroll Lock tuşu kendi ışığını açıp kapamaktan başka ne işe yarar diye düşündünüz mü hiç? Boş kalmaması diye Excel'de sayfayı kaydırma moduna geçme özelliği vermişler. Access'te de işe yaraması için işte size işlev. Kayıt sayfanız liste halindeyken Scroll Lock tuşu açık olduğunda, aşağı/yukarı oklarıyla kayıtlar arasında dolaşabilirsiniz.



Bir modül sayfasında en üste aşağıdaki kodları yerleştirin:

```
Private Declare Function GetKeyState Lib "user32" (ByVal nVirtKey As Long) As Integer
Private Const VK_SCROLL = &H91
```

```
Public Sub ScrollMode(frm As Form, KeyCode As Integer, Shift As Integer)
'Serkan Sahinoglu, 2001, http://www.EskiDefterler.com
```

```
If GetKeyState(VK_SCROLL) = 1 And frm.DefaultView = 1 Then
Select Case KeyCode
Case vbKeyUp
SendKeys "^PGUP"
KeyCode = 0
Case vbKeyDown
SendKeys "^PGDN"
KeyCode = 0
```

```
'SOL ve SAĞ okları kullanmak istemiyorsanız
'alttaki iki Case bloğunu silin.
```

```
Case vbKeyRight
SendKeys "{TAB}"
KeyCode = 0
Case vbKeyLeft
SendKeys "+{TAB}"
KeyCode = 0
```

```
End Select
End If
```

```
End Sub
```

Kullanmak istediğiniz formun KeyPreview (olay yordamı listesinde alttan itibaren altıncı sırada) özelliğini aktif ettikten sonra formun KeyDown (tuşa basılırken) olay yordamını şu şekilde düzenleyin:

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
ScrollMode Me, KeyCode, Shift
```

End Sub

Bundan sonra Scroll Lock tuşu açık ise klavyedeki ok tuşlarını kullanarak kayıtlar arasında hareket edebilirsiniz.

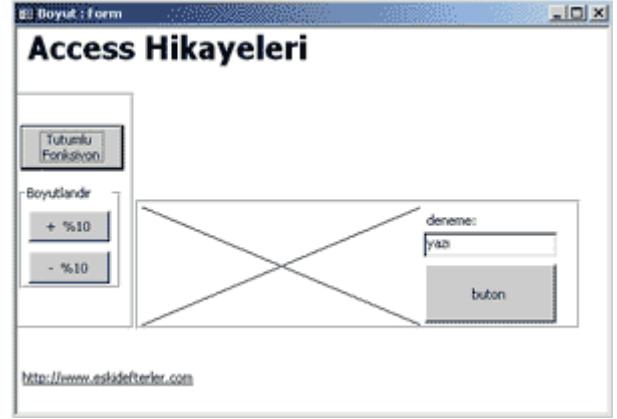
TUTUMLU FONKSİYON

Programınızda güzel bir form hazırladınız, zevkinize göre istediğiniz renklerle süslediniz. Yeri geldiğinde bu formu yazıcıya gönderiyorsunuz. Bu halde iken formdaki verilerden çok, diğer renkler için bir sürü toner gidiyor.

Böyle durumlarda aşağıdaki tutumlu işlevimiz gereksiz toner masrafını ortadan kaldırıyor. Kullanmak istediğiniz kod kısmına ya da bir düğmenin Tıklatıldığında (OnClick) eylemine

Tutumlu İşlev Me

yazdıktan sonra aşağıdaki işlevi genel bir modülün içine yerleştirin:



Public Sub Tutumlu İşlev(frm As Form)

'Serkan Sahinoglu, 2001, <http://www.EskiDefterler.com>

If MsgBox("Bu işlemi yaptıktan sonra eski görüntüye dönmek için pencereyi kapayıp tekrar açmalısınız. Devam edilsin mi?", vbYesNo) = vbNo Then Exit Sub

On Error Resume Next

Dim i As Integer

DoCmd.Hourglass True

'Tüm bölgeler beyaz. Toplam 9 bölge vardır.
For i = 0 To 8
frm.Section(i).BackColor = 16777215 'beyaz
Next

'Tüm nesneleri yeniden renklendir.
For i = 0 To frm.Controls.Count
frm.Controls(i).BackColor = 16777215 'beyaz
frm.Controls(i).ForeColor = 0 'siyah
Next

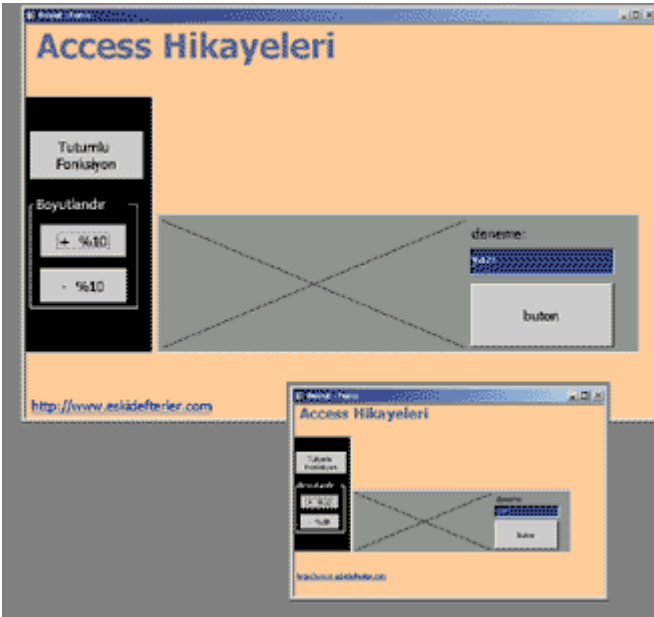
DoCmd.Hourglass False

End Sub

ayın işlevi

TÜM NESNELERİ YENİDEN BOYUTLANDIRIN

İşte size formlarınızı içindekilerle birlikte yeniden boyutlandırmanıza yarayan bir işlev. Bununla kullanıcılarınızın kendi ekranlarına göre boyutlandırma yapabiliyorsunuz. Alın, kullanın.



Formun OnOpen (Açıldığında) olay yordamına

Boyutlandır Me, 1.25

'Tüm nesneler 1.25 kat büyüyecek.

yazın. Tabii önce Boyutlandır işlevini genel bir modül sayfasına ekleyin.

Kod biraz hassas olduğu için hata oluştuğunda formunuzun içindeki bölgeleri biraz genişletmeyi deneyin. Diğer boyutlandırma seçenekleri için koddaki Oran adlı değeri değiştirebilirsiniz.

Formunuzu standart olarak 800*600 için hazırlamışsanız;

800*600 ekranda oran = 1

640*480 ekranda oran = 0.8

1024*768 ekranda oran = 1.28

değerlerini kullanın.

```
Public Sub Boyutlandir(frm As Form, Oran As Double)
'Serkan Sahinoglu, 2001, http://www.EskiDefterler.com

If Oran = 1 Then Exit Sub

Dim i As Integer
On Error Resume Next 'Hataları yoksay.

frm.Width = frm.Width * Oran 'En fazla 31680.

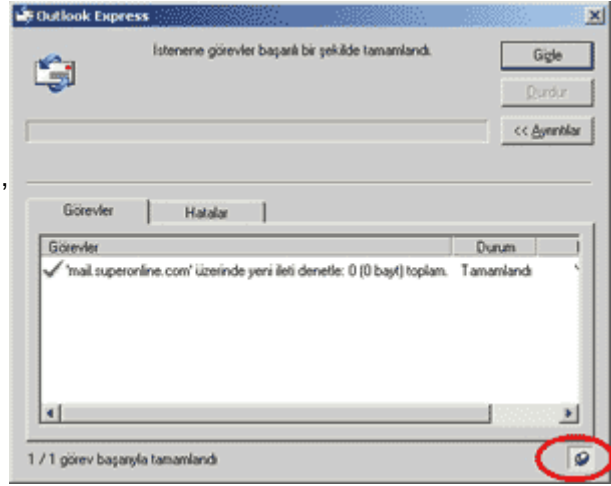
'Tüm bölgeler
For i = 0 To 8
frm.Section(i).Height = frm.Section(i).Height * Oran
Next

'Tüm nesneler
For i = 0 To frm.Controls.Count - 1
If frm.Controls(i).ControlType <> acPage Then
With frm.Controls(i)
.Width = .Width * Oran
.Height = .Height * Oran
.Left = .Left * Oran
.Top = .Top * Oran
.FontSize = .FontSize * Oran
'Bazı yazıtipleri küçülüyor.
'Arial ve Times New Roman küçülüyor.
End With
End If
Next

End Sub
```

NE İŞE YARAR BU MERET

Outlook Express kullanıyorsanız e-posta alma ve gönderme anında açılan formun sağ alt köşesindeki küçük resim dikkatinizi çekmiştir. Ama bunun ne işe yaradığı konusunda ekranda fazla bir ipucu yok. Efendim, bu öyle bir şeydir ki, Internet Explorer'daki dosya yükleme penceresinde bulunan « çıkışta bu iletişim kutusunu kapat » seçeneği ile aynı işi yapar. Yani bunu basılı bırakırsanız bu pencere işlem bittikten sonra ekranda kalacaktır.



DOSYA:

[AccessEkim2001ornek.zip](http://www.eskidefterler.com/AccessEkim2001ornek.zip)

Serkan Şahinoğlu, PC Magazine, Ekim 2001

<http://www.eskidefterler.com>

DB1.MDB



Projenizin başından sonuna kadar nelere dikkat etmelisiniz? Bitmiş bir Access projesini nasıl dağıtmalı? İşte yeni teknikler ve yeni ipuçları.

Bu sayfalarda Access projelerinizin başından sonuna dek dikkat etmeniz gerekebilecek önemli bazı ayrıntıları, uygulayabileceğiniz ilginç teknikleri ve önemli ipuçlarını bulacaksınız. İşte yeni teknikler ve yeni kotarılmış ipuçları!

...

BİR ACCESS PROJESİ

Access dosyasını düzenli tutmamak, masamızı sürekli dağınık bırakmak gibi sonradan can sıkıcı olabilir. Bir projeye başlamadan önce yapılması önerilen ilk işlerden biri, proje için anlaşılır ve kullanışlı dosya ve nesne adları belirlemektir.

Dosya gruplarından oluşmuş projeler gibi Access dosyaları da obje gruplarından oluşur. Formlar, raporlar, modüller görünür objedirler. Bununla beraber Access kendi içinde görünen ve görünmeyen birçok objeyi barındırır.

Container (Konteyner) küme nesneleri, içlerinde bildiğimiz objeleri, yani formları, raporları, tabloları, indeksleri ve diğer nesneleri taşıyabilirler (DAO.Database.Containers). Bunlar dışında bizim bilmediğimiz birçok görünmeyen objenin de bu tip konteynerler içinde bulunması muhtemel.

Gelgelelim, bu gibi zor işleri Access yapıyor ve bize olabildiğince az iş bırakıyor. Çoğunlukla seçenekler arasında dolaşıp fareyle işaretleyerek bir Access projesi üretebiliyoruz. İşte bu kolaylıklar yüzündendir ki, Access'in programlama aracı olamayacağı yönünde bir görüş de var nitekim. Access'i yapanlar da "yapsak bir türlü, yapmasak iki türlü" diye dert yanmaktadırlar herhalde!

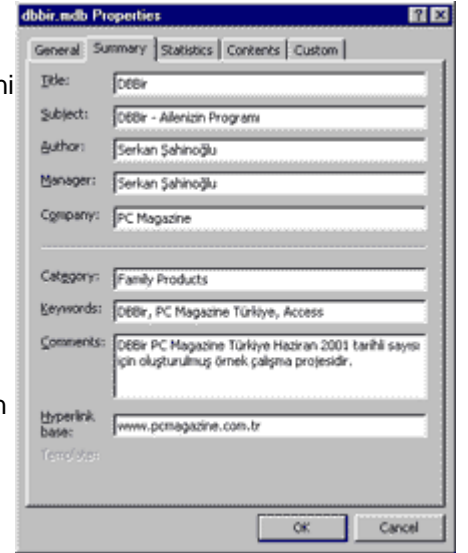
DOSYAYA KİMLİK

Bir Access dosyasını oluşturduğunuzda bu dosyaya adınızın kaydedildiğini unutmayın.

File > Database Properties (Dosya > Veritabanı Özellikleri) menüsünü açın. Gerekli gördüğünüz alanları doldurabilirsiniz. Buraya yazacağınız bilgiler dosyanız için genel amaçlı açıklamalar niteliğinde olmalıdır.

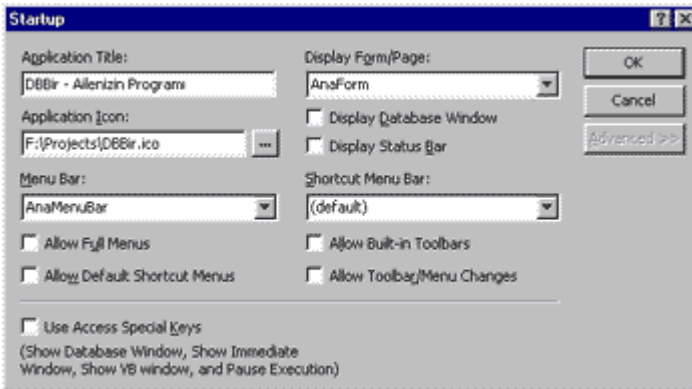
Siz hiçbir şey yazmış olmasanız dahi, dosyayı ilk oluşturduğunuzda bilgisayarınızda kayıtlı olan isminiz ve diğer bazı bilgileriniz Office'in diğer üyelerinde olduğu gibi Access'te de dosya özellikleri sayfasına kaydedilecektir. Bu özellik hemen her Office bileşeninde olmasına rağmen pek bilinmiyor ya da kullanılmıyor.

File > Database Properties menüsünü açıp özet bilgileri yazın.

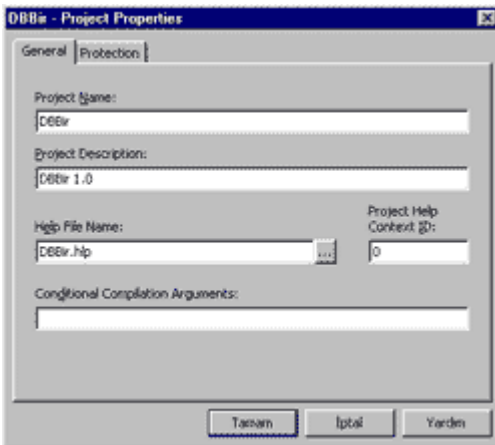


Buraya kadar aslında bir Office bileşeni olarak bu Access dosyası hakkında genel bilgiler yazmış oldunuz. Uygulama ile ilgili bilgileri ise başka iki bölümde yazmanız gerekiyor.

Tools > Startup (Araçlar > Başlangıç) menüsünü açarak Application Title (Uygulama Başlığı) kutusuna bu uygulamanın Windows'ta görüntülenecek açık adını yazın. Başlat panelinde buraya yazdığınız isim görüntülenecektir.



Son olarak Tools > Macro > Visual Basic Editor (Araçlar > Makro > Visual Basic Düzenleyicisi – ALT+F11) menüsü ile Visual Basic Editor'ü açın. Tools > DBBir Properties (Araçlar > DBBir Özellikleri) menüsü ile proje adını yazabilirsiniz. Project Name (Proje Adı) kutusuna uygun bir isim yazmanız gerekiyor.



EN KOLAY SIRALAMA

Bazı programlarda bir listenin başlığını tıklatınca ilgili veri alanına göre sıralama yapılıyor. Windows Gezgini ya da Outlook Express'te bu şekilde kolayca sıralama yapılabilir. Access'te bunun aynısını yapmamız için bir engel yok tabii ki.

İş biraz uzun ama yaptığınıza değecek cinsten. Önce resimdeki gibi bir form oluşturun. Aşağı ve yukarı ok işaretlerini Webdings yazıtipinden alacağız. "5" ve "6" karakterleri bu ok işaretlerine karşılık geliyor. İki Label (metin etiketi) oluşturup bunları yeşil olarak renklendirdikten sonra adlarını labelYonASC ve labelYonDESC olarak ayarlayın.

Üst şeritteki altı çizili label (etiket) nesnelerinin adları sıra ile labelID, labelKitapAdi, labelYazar, labelYayıncı, labelYıl, labelCeviren, labelGrup, labelGirisTarihi olacak. Yine bu etiketlerin Tag özelliklerini sıra ile ID, KitapAdi, Yazar, Yayıncı, Yıl, Ceviren, Grup, GirisTarihi olarak ayarlayın.

Aynı şekilde bu Label nesnelerinin HyperlinkAddress özelliklerini de " " (sadece bir adet boşluk) olarak ayarlayın. Son olarak Tablo1'deki uzun kodu bu formun kod sayfasında en üstteki

Option Compare Database

Option Explicit

'(Bu satır yoksa ekleyebilirsiniz.)

satırlarının altına ekleyin.

Nihayetinde listede üstteki altı çizili isimleri fareyle tıklatarak kolaylıkla artan veya azalan sıralama yaptırabilirsiniz. Doğruca azalan sıralama yapmak için tıklama anında Shift tuşunu basılı tutmak yeterli.

Yazar alanına göre ters sıralı.

Form üzerinde kolay sıralama için gerekli kod

```
Private Declare Function GetKeyState Lib "user32" (ByVal nVirtKey As Long) As Integer
Private Const VK_SHIFT = &H10
```

```
Private Sub SiraLa(labelX As Label)
```

On Error Resume Next

Dim fieldx As String
fieldx = labelx.Tag

If ((OrderBy = fieldx) Or (GetKeyState(VK_SHIFT) < 0)) Then
 'Shift tuşu basılı ise ters sırala.
 OrderBy = fieldx & " DESC"
 labelYonASC.Visible = False
 labelYonDESC.Visible = True
 labelYonDESC.Left = labelx.Left + labelx.Width - 60
 labelYonDESC.Top = labelx.Top - 20
Else
 OrderBy = fieldx
 labelYonDESC.Visible = False
 labelYonASC.Visible = True
 labelYonASC.Left = labelx.Left + labelx.Width - 60
 labelYonASC.Top = labelx.Top - 20
End If

OrderByOn = True

End Sub

Private Sub Form_Error(DataErr As Integer, Response As Integer)
 Select Case DataErr
 Case 7983: 'Hyperlink Hatası
 Response = acDataErrContinue
 End Select
End Sub

Private Sub labelCeviren_Click()
 Siralama labelCeviren
End Sub

Private Sub labelGirisTarihi_Click()
 Siralama labelGirisTarihi
End Sub

Private Sub labelGrup_Click()
 Siralama labelGrup
End Sub

Private Sub labelID_Click()
 Siralama labelID
End Sub

Private Sub labelKitapAdi_Click()
 Siralama labelKitapAdi
End Sub

Private Sub labelYayinci_Click()
 Siralama labelYayinci
End Sub

Private Sub labelYazar_Click()
 Siralama labelYazar
End Sub

Private Sub labelYil_Click()
 Siralama labelYil
End Sub

'Her bir başlık etiketinin Click (Tıklatıldığında) olay
'yordamını ve formun Error yordamını Özellikler
'penceresinde bu yordamlarla ilişkilendirin.

KAYIT SEÇİCİLERDEN SIKILDIYSANIZ


Formlarda sol tarafta bulunan kayıt seçicilerden sıkıldıysanız aşağıdaki adımları takip ederek kendi kayıt seçicilerinizi yapabilirsiniz. Öncelikle Otomatik Sayı (AutoNumber) benzeri bir veri alanı kullanılması gerekiyor. Bu örnekte ID adlı bir AutoNumber alan olduğu varsayılıyor.

Formunuzun sol tarafına bir metin kutusu (Textbox) yerleştirin. Boyunu yeteri kadar kısaltın. Özellikler (Properties) penceresinden, aşağıdaki değerleri ayarlayın:

Ad (Name) = Secici
Enabled (Seçili?) = No (Hayır)
Locked (Kilitli) = Yes (Evet)
BackStyle (Arkaplan Biçimi) = Transparent (Saydam)
BorderStyle (Kenarlık Biçimi) = Transparent (Saydam)
FontName (Yazıtipi) = Webdings

Bu değerleri atadıktan sonra formun Current olay yordamını aşağıdaki gibi düzenleyin:

```
Private Sub Form_Current()  
Secici.ControlSource = "=IIf([ID]=\" & _  
Nz(Me!ID, 0) & \", '8', '4')\"  
End Sub
```



ID	Kitap	Yazar	Yayıncı	Yıl	Çeviren
1	Kırmızı ve Siyah - 1	Stendhal	MEB	1997	Nurullah Ataç
2	Dehşetli - 1	Dostoyevski	MEB	1997	Servet Lünel
3	Dehşetli - 2	Dostoyevski	MEB	1997	Servet Lünel
4	Dehşetli - 3	Dostoyevski	MEB	1997	Servet Lünel
5	Kırmızı ve Siyah - 2	Stendhal	MEB	1997	Nurullah Ataç
6	Kadınlarda Duygusal Zor	Mariella Santorus	Varlık	1999	Şebnem Can Erendor
7	Oracle7 Workgroup Se	Dr. Yalçın Özkan	Anı	1996	
8	İstanbulun Doğusunda	Peter Hopkirk	Yeni Yüzyıl	1995	Mehmet Hamancı
9	Nineteen Eighty-Four	George Orwell	Penguin	1954	
10	Microsoft Windows 95	Microsoft Corpora	Akıadaç	1996	Ceyhan Temirci

Sol tarafta kendi kayıt seçicimiz duruyor.

Resimdeki gibi kayıt seçicimiz sol tarafta duruyor. Metin kutusunun üzerindeki resmin görüntülenmesi için Webdings yazıtipi sistemde yüklü olmalı.

LİSTEYİ SÜZ

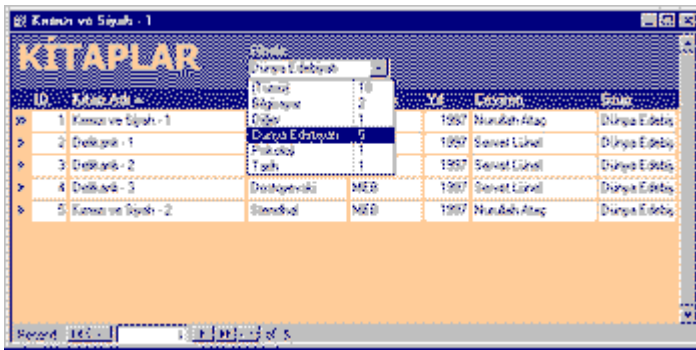
Kayıtlarınızı belli gruplara göre ayırmak istiyorsanız Access'in süzme özelliğini kolaylıkla kullanabilirsiniz. Bizim yöntemimiz için bir Combobox (Açılan Liste Kutusu) nesnesi gerekiyor. Adı cmbFiltre. Açılıştaki ilk değeri görüntülemek için listenin Default Value (Geçerli Değer) özelliğini "(Tümü)" olarak atayın. RowSource (Kayıt Kaynağı) özelliğini de alttaki şekilde ayarlayın:

```
(  
SELECT Grup as GrupAdi,  
COUNT(Grup) AS Adet  
FROM Kitaplar  
GROUP BY Grup  
)  
UNION
```

```
(
SELECT "(Tümü)" as GrupAdi,
COUNT(ID) as Adet
FROM Kitaplar
);
```

Sonra bu listenin AfterUpdate (Güncelleştirme Sonrası) olay yordamını şu şekilde düzenleyin:

```
Private Sub cmbFiltre_AfterUpdate()
If IsNull(cmbFiltre) Then Exit Sub
Dim orderbyx As String, orderbyonx As Boolean
'Eski sıralama korunacak.
orderbyx = OrderBy
orderbyonx = OrderByOn
If cmbFiltre = "(Tümü)" Then
Me.RecordSource = "Kitaplar"
Else
Me.RecordSource = "SELECT * FROM Kitaplar " & _
"WHERE Grup='" & cmbFiltre & "'"
End If
OrderBy = orderbyx
OrderByOn = orderbyonx
End Sub
```



KAYDA GİT

DBBir programımıza şimdi de kayda gitme özelliği eklememiz gerekiyor. Önce txGit adlı bir metin kutusu (Textbox) oluşturun. Buna yazdığınız metni Kitap adları arasında bulmak için şu kodu kullanın:

```
txKitapAdi.SetFocus
DoCmd.FindRecord txGit, acAnywhere, , acDown, , False
```

Kolaylık olması için F5 tuşuna basınca bu metin kutusuna odaklanmayı sağlayalım. Formun KeyPreview adlı özelliğini True, formdaki KeyDown (Tuşa Basarken) yordamını da alttaki gibi yapın:

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
Case vbKeyF5: txGit.SetFocus
End Select
End Sub
```

Bu örnek görüntülenen kayıt kümesi üzerinde aranan değeri bulup getirmeye ilgilidir. Peki görünen kaydın ayrıntılı bilgilerini başka bir formda göstermek için ne yapmalıyız?

Bunun birkaç yolu var. Bizim yöntemimizde önce ayrıntılı bilgilerin yer aldığı detay formu oluşturulmalı. Ardından bu detay formunun RecordSource (Kayıt kaynağı) özelliğini şu şekilde ayarlayın:

```
SELECT Kitaplar.* FROM Kitaplar
WHERE ((ID=[Forms]![Kitaplar_Liste]![ID]));
```

Ana formumuzun adını Kitaplar_Liste olarak belirlediğim için bu kodda da aynı adı kullandım. Tabii detay formunu açmadan önce bu liste formunun açık olması gerekiyor. Bu formu açtığınızda liste formunda görünen kaydın ayrıntısı açılacaktır.

ODE TOOLS İLE PROJENİZİ DAĞITIN

Geldik DBBir projesinin dağıtım paketinin hazırlanmasına. Amacımız DBBir projesinin kod ve tasarım bilgilerini tamamen gizlemek ve programın Access kurulu olmayan bilgisayarlarda çalıştırılmasını sağlamak.

Tasarım bilgilerini gizlemek için Tools > Database Utilities > Make MDE File (Araçlar > Veritabanı Seçenekleri > MDE Dosyası Yap) menüsü ile MDE dosyasını oluşturun. Bunu yaptığınızda derlenmiş yeni MDE dosyasında tasarım ve kod bilgileriniz tamamen gizlenmiş olacaktır.

Dağıtım işini ise ODE Tools ile yapacağız. Öncelikle ODE Tools'un bilgisayarınızda kurulu olması gerekiyor (Bilgi için Nisan 2001 tarihli PC Magazine'deki Access yazısı).

Öncelikle birkaç ayrıntının aklınızın bir köşesinde bulunmasında fayda var.

Kurulumu hazırladığınız bilgisayarda Access'in hangi dilde olduğuna dikkat edin. Programınız hep o dili kullanacaktır.

ODE Tools ile hazırladığınız kurulum programı iki temel bileşenden oluşur. Sizin program dosyalarınız ve Access Runtime.

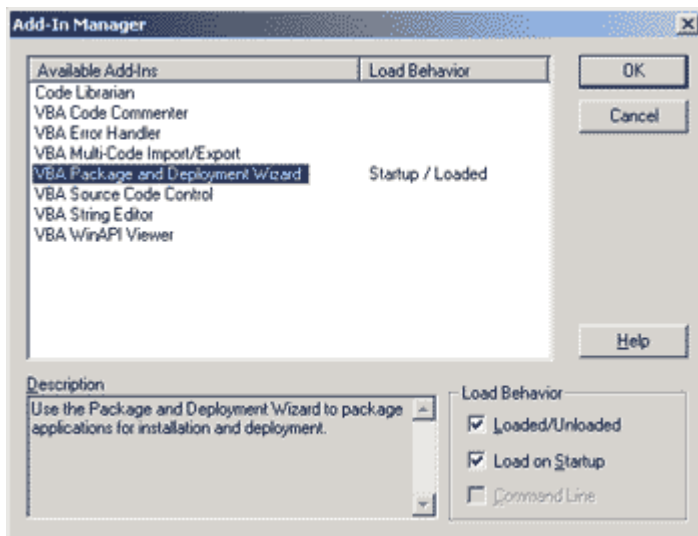
Access Runtime programınızı çalıştıracak modüldür ve her bilgisayara sadece bir kez kurulması yeterlidir. Zaten bu modül disk üzerinde tek başına 70 MB. yer kaplıyor. Bu durumda hazırladığınız paket ortalama 75 MB. büyüklüğünde olacaktır. İsterseniz kurulum hazırlarken Access Runtime modülünü eklemiyorsunuz. Bu durumda 3 MB. civarında bir paket çıkıyor ortaya.

Bir kereye mahsus olarak 75 MB. büyüklüğünde bir paket hazırlayıp sonraki paketlerinize Access Runtime eklemeyebilirsiniz.

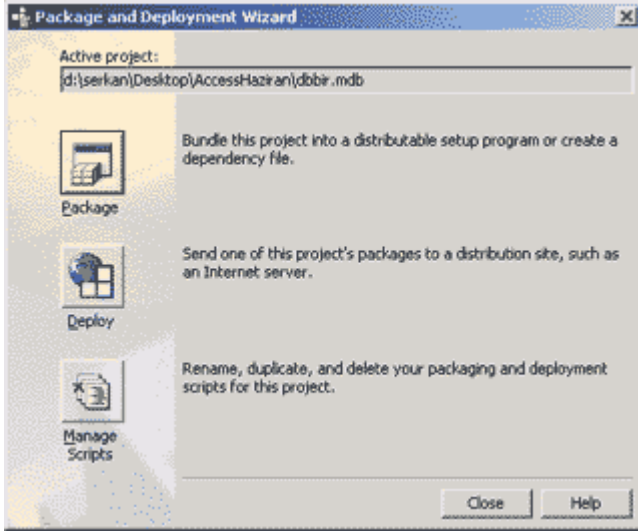
KURULUM PAKETİ HAZIRLAMA

ODE Tool'un bir önceki sürümü kurulumdan sonra kendisini Programlar menüsüne yerleştiriyordu. ODE Tools 2000 kurulduğunda ise kendisini bulabilmek için biraz çaba sarfetmek gerekiyor.

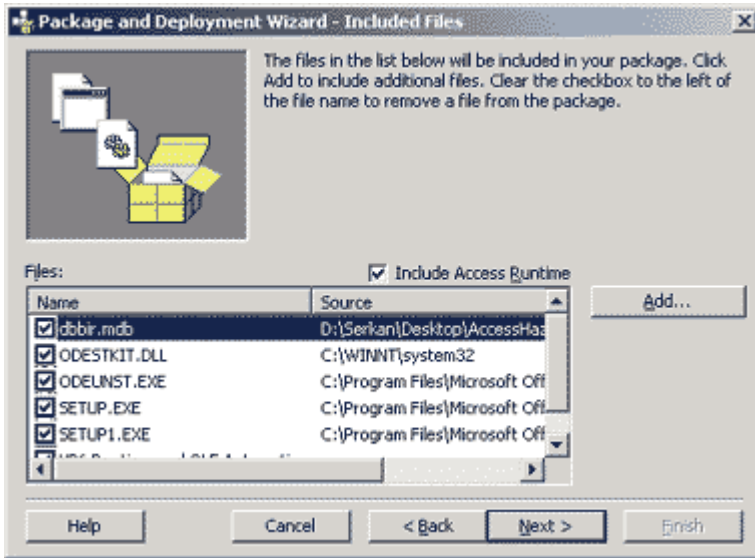
ALT+F11 ile Visual Basic Editor'ü açıp Add-Ins > Add-In Manager menüsü ile VBA package And Deployment Wizard'ı yüklemek gerekiyor. Bu haliyle Add-Ins menüsü altında Package And Deployment Wizard menüsü belirecektir. Bunu çalıştırmadan önce Access dosyasının Exclusive (tek kullanıcı oturumu) modda açılması gerekli.



Dosyayı kapatın ve dosya açma penceresinde dosyayı seçip sağ alttaki düğmeyi Open (Aç) durumundan Exclusive Open haline getirin. Exclusive olarak açtıktan sonra Visual Basic Editor'e geçip ilgili menüyü çalıştırın. Açılan formdan Package'yi tıklayın.



Devam eden sayfalarda önerilen komutları seçin. Gerekli görmüyorsanız ilerleyen sayfalardaki seçeneklerin biri ile Access Runtime'i kurulum paketinize dahil etmeyebilirsiniz.



Zira ilk seferde onlarca megebayt büyüklüğünde bir paketle karşı karşıya kalabilirsiniz. Öncelikle Microsoft.com sitesinde önerilen bazı güncellemeleri yapmanız gerekebilir.

MOD2000: Access 2000 Run-Time Minimum Now Available for Download

<http://support.microsoft.com/support/kb/articles/Q243/9/57.ASP>

Kurulum hazırlama anında hatalarla karşılaşırsanız dosyanızda gereksiz yüklenmiş olan bazı başvuruları kaldırmayı deneyin. Visual Basic Editor > Tools > References.

Aynı şekilde hazırlama işlemi çok uzun sürüyorsa paketinize yüzlerce dosyanın ekleniyor olması muhtemeldir. Hazırlık aşamasında "Include Access Runtime" seçeneğini işaretlerseniz "C:\Program Files\Microsoft Office\ODETools\V9\Runtime" klasörünün tamamı paketinize dahil edilir. Ben bu klasörün yedeğini aldıktan sonra içindeki gereksiz olduğunu düşündüğüm bazı dosya ve klasörleri sildim.

Herşey olması gerektiği gibi ise ODE Tools ile, nasıl yapıldığını anlatmama gerek kalmayacak kadar kolay bir şekilde dağıtım paketinizi (setup) hazırlayabilirsiniz.

HERKES İÇİN SORGULAR

GRUPLAYARAK TOPLAMLAR ALMAK

Sorgularla ilgili en çok sorulan sorulardan biri gruplamalar hakkında.

- Her ürün ana grubunun içinde kaç adet ürün var?
- Her güne ait toplam satış tutarını nasıl hesaplayacağım?

Yukarıdakiler gibi sorularınız varsa GROUP BY deyimini kullanmanız gerekiyor demektir. Bunu en kolay yapabileceğimiz yer Access'teki Query Builder.

Yeni bir Query (Sorgu) oluşturup içine Kitaplar tablosunu ekleyin. Veri alanlarını alttaki sütunlara yerleştirip View > Totals (Görünüm > Toplamlar) menüsünü seçin ve Query1 resmindeki gibi sorguyu düzenleyin.

Bu haliyle sadece grup adını ve o grupta kaç kitap olduğunu bulmuş oluyoruz. Geçen aylarda bahsettiğimiz UNION sorgu türü ile toplam kitap adedini de bu sorguya dahil edelim.

SELECT GrupAdi, Adet

FROM

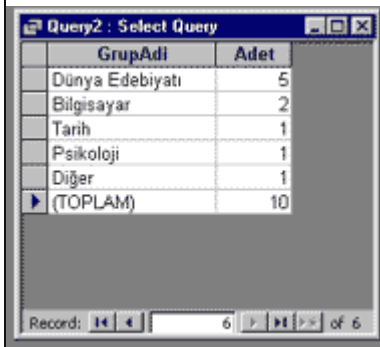
```
(  
SELECT Grup as GrupAdi,  
COUNT(Grup) AS Adet, 0 AS Sira  
FROM Kitaplar  
GROUP BY Grup
```

UNION

```
SELECT "(TOPLAM)" as GrupAdi,  
COUNT(ID) as Adet, 1 AS Sira  
FROM Kitaplar  
)
```

AS Liste

ORDER BY Sira, Adet DESC;



GrupAdi	Adet
Dünya Edebiyatı	5
Bilgisayar	2
Tarih	1
Psikoloji	1
Diğer	1
(TOPLAM)	10

Query2 resminde gördüğünüz gibi her bir grup adını ve içinde kaç adet kitap olduğunu listelemekle beraber en altta da toplam kitap adedini göstermiş oluyoruz.

Sorguda kullandığımız Sira adlı alan UNION sorgu içindeki farklı iki kümeyi birbirinden ayırmak içindi. İlk kümeyi 0, ikincisini de 1 yapıp bu alanı en altta ORDER BY içine yerleştirmek yeterli oluyor.

Genel olarak adet bulma işlemleri için COUNT deyimini kullanmalısınız. Toplam hesaplamak için ise SUM deyimini kullanılıyor. AVG, MIN, MAX, STDEV, FIST, LAST gibi deyimlerle de ortalama, en küçük, en büyük, standart sapma, ilk, son gibi değerleri elde edebilirsiniz.

AÇILIŞTA TAM EKRAN OLSUN

Üzerinde uzun süredir çalıştığınız projenizin hep aynı şekilde açılması kolaylık getirecektir. Ben veritabanı penceresinin hep tam ekran açılmasını tercih ederim. Bunun için Makrolar (Macros) sayfasında AutoExec adlı yeni bir makro oluşturmak yeterli.

Access açıldığında her zaman AutoExec adlı bir makro arar ve bulursa onu çalıştırır. Windows'taki Autoexec.bat dosyası gibi. Bu makroyu oluşturduktan sonra ilk satırda Action sütununa gelip listeden "Maximize"yi seçin ve makroyu kaydedip kapatın. Bundan sonraki her açılışta veritabanı penceresi tam ekran açılacaktır.

Aynı şekilde Tools > Startup (Araçlar > Başlangıç) menüsündeki seçenekleri kullanarak projenizin başlığını, başlangıçta açılmasını istediğiniz formları burada belirleyebilirsiniz. Gerekliyse veritabanı pencerenizi gizlemeniz, kısayol tuşlarını kısıtlamanız mümkün. Böylece gizlediğiniz veritabanı penceresini kimsenin görmemesini sağlamış oluyorsunuz.

DOSYANIN İÇİNİ KİMSE GÖRMESİN

AutoExec makrosu ve diğer başlangıç seçeneklerini bir şekilde es geçme şansımız vardır. Bir Access dosyasını açarken klavyede Shift tuşunu basılı tutarsanız AutoExec makrosu çalıştırılmaz. Aynı şekilde diğer başlangıç seçenekleri de geçerli olmayacaktır. Buna Tools > Startup menüsündeki Application Title (Uygulama Başlığı) özelliği ve veritabanı penceresinin görüntülenme seçeneği de dahil.

Fakat bunu da gidermenin yolu var. Aşağıdaki kodu kullanarak dosyanızın Shift tuşu ile açılıp içindeki tabloların, sorguların ve diğer nesnelerin görüntülenmesini engelleyebilirsiniz. Öncelikle Başvurular (References) listesinden DAO'yu yüklemiş olmanız gerekiyor.

Dim db As Database

Set db = OpenDatabase("c:\db1.mdb")

Dim prop

Set prop = db.CreateProperty("AllowByPassKey", DB_BOOLEAN, False)

db.Properties.Append prop

db.Close

Bu şekilde Access dosyasının içinde, geçerli olarak False değerli AllowByPassKey adlı bir Property oluşturuluyor. Böylelikle Shift tuşu basılı olarak açılrsa da başlangıç seçeneklerinde belirlediğiniz veya gizlediğiniz özellikler kimse göremeyecektir.

DOSYANIN İÇİNİ KİMSE GÖRSÜN!

Üstteki kodu kullanarak Shift ile dosyanızın açılmasını engellediğinizi kesinlikle sağlayabileceğinizi mi düşünüyorsunuz? Maalesef bunu da aşmanın yolu yok değil. Dosyanız MDE dosyası olsa da.

Dim db As Database

Set db = OpenDatabase("c:\db1.mdb")

db.Properties("AllowByPassKey") = True

db.Close

Ama unutmayın ki, bunu ancak Access'te kod yazmayı bilen bir kişi yapabilir. Kullanıcılarınız Access'te kod yazmayı bilmiyorsa üstteki örneği de denemeye uğraşmazlar herhalde.

Programınızı güvenli dağıtmak istiyorsanız tek önerim MDE dosyası olarak derlemeniz. Ne kadar kısıtласınız ya da parola koysanız da, bunu çözen programlar Internet'te mevcut.

ETİKET DEDİĞİN YAPIŞKAN OLMALI

Form üzerine yeni bir metin kutusu (Textbox) yerleştirdiğimizde geçerli olarak yanına bir de metin etiketi (Label) iliştiliyor. Kutuyu taşısanız da bu yapışkan etiket onunla hareket ediyor. Buna gerek olmadığı durumlarda etiketi seçip silebiliyoruz. Fakat bazan buna tekrar ihtiyacımız olabiliyor.

Forma yeni bir metin etiketi (Label) yerleştirin. Bunu seçip klavyede CTRL+C tuşlarına basarak hafızaya alın. Sonra yapıştırmak istediğiniz metin kutusunu seçerek CTRL+V ile etiketi kutuya yapıştırın.

Yazıdaki örnekleri PC Magazine Online sitesinde Dergide Adı Geçen Dosyalar bölümündeki dosyada bulabilirsiniz.

MICROSOFT PHOTO EDITOR PROBLEMİ

Office ile birlikte gelen ama geçerli kurulum listesinde işaretlenmediği için genelde yüklenmeyen Microsoft Photo Editor resimleri kesip biçme işlemleri için orta halli ama çok pratik bir program sayılabilir.

Birkaç kez bu programı çalıştırdığımda Başlat panelinde küçültülmüş olarak kalmaya başladı. Ne kadar tıklasam yüzünü görmek nasip olmuyordu. Velhasıl, Başlat panelinde küçültülmüş halde durduğu bir zamanda üzerinde sağ tuşu tıklatıp Ekranı Kapla menüsünü seçince hiç olmazsa tam ekran çalışabiliyor.

Güneşli güzel günler...

DOSYA:

[accessdb1.zip](#)

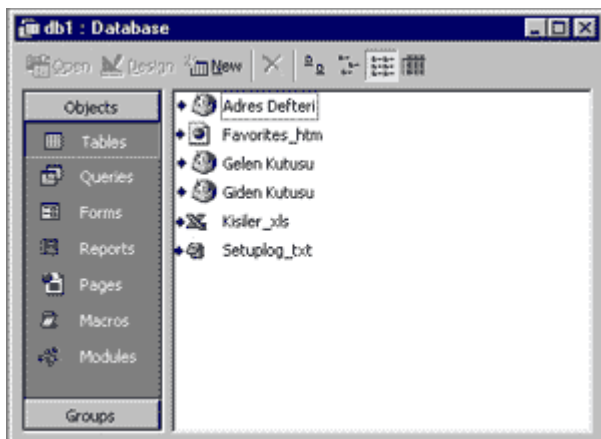
Bağlı Tablolar

Access başka veri kaynaklarından da kendisine tablo bağlanmasına izin veriyor. Altından kalkamadığı durumlarda ise yardımına ODBC koşuyor.

Tablo bağlama işleminin Access'in İngilizce sürümündeki adı Link. Farklı türdeki birçok veri tablosunu Access dosyalarının içine bağlamak mümkün. Örneğin Excel'de hazırlanmış bir tabloyu Access'in içinden rahatlıkla kullanabilirsiniz. Sabit sütunları olan bir Excel tablosunu Access'e bağlamak için Access'teki Dosya / Dış Veri Al / Bağlı Tablo (File / Get External Data / Link Tables) komutunu çalıştırıp bir Excel dosyası seçmek yeterli. Sonrasındaki soruları da geçerek Excel tablosunu kolayca bağlamış oluyorsunuz.

Bu özellik öncelikle Access tablolarını başka Access tabloları içine bağlamak için var. Access 2000 dosyası içine Access 97 sürümündeki bir dosyanın tablosu bağlanabiliyor. Böylece tabloyu uzaklarda aramadan doğrudan kullanabiliyorsunuz.

Bazan, ayrıntılara dikkat etmenin ne kadar önemli olduğundan bahsediyorum. Eğer tablo bağlama penceresindeki dosya türleri (file of types) listesine dikkat ederseniz orada metin dosyalarından tutun Exchange tablolarına kadar birçok seçenek göreceksiniz.



Bağlı tabloları herhangi bir Access tablosuymuş gibi kullanabilirsiniz.

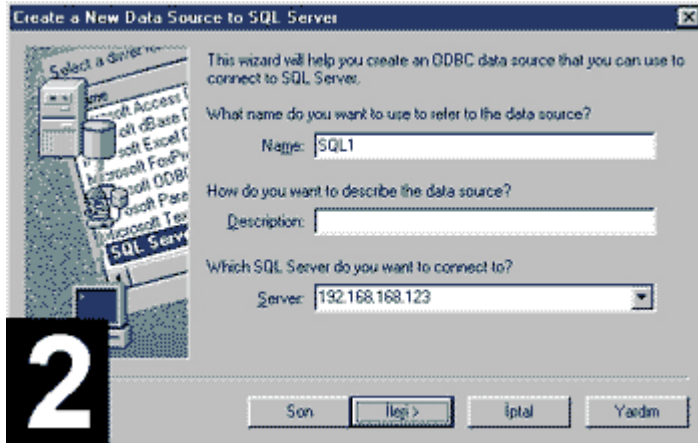
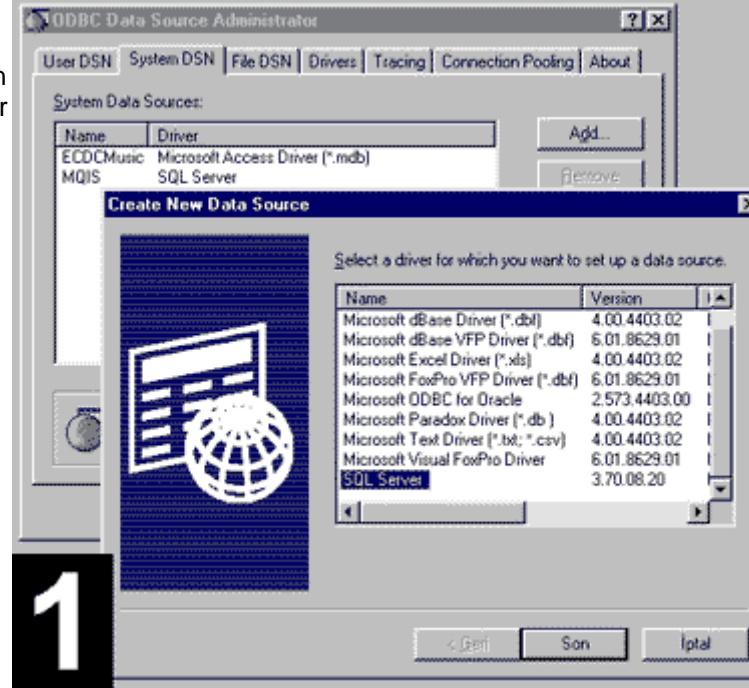
Outlook Express'teki adres defterimi başka bir yerden kullanabilmek hiç bu kadar kolay olmamıştı. Tablo bağlama penceresinde dosya türü olarak Exchange'i seçtikten sonra açılan pencereden Adres Defteri'ni seçip Access dosyasının içine bağladım. Aynı şey e-posta iletileri için de geçerli.

Kayıtlar içeren metin dosyaları için de hemen hemen aynı şeyler geçerli. Fakat bu dosyalar veritabanı yapısında olmadığı için tablo bağlarken bazı ayarlamalar gerekiyor. Virgülle, tab işaretiyle ayrılmış veya sabit uzunluklu alanlar içeren bir metin dosyasını, tablo bağlama penceresinden seçin. Açılan formlarla birlikte gerekli ayarlamaları kolayca yapabilirsiniz. Nihayetinde metin dosyasını (sadece okunur) bir tablo olarak kullanabilirsiniz.

SQL SERVER TABLOSUNU ACCESS'E BAĞLAMAK

Gelelim SQL Server tablolarının nasıl bağlandığına. Bunun için önce Denetim Masası'ndaki ODBC bölümünde bir ayar yapmak gerekli.

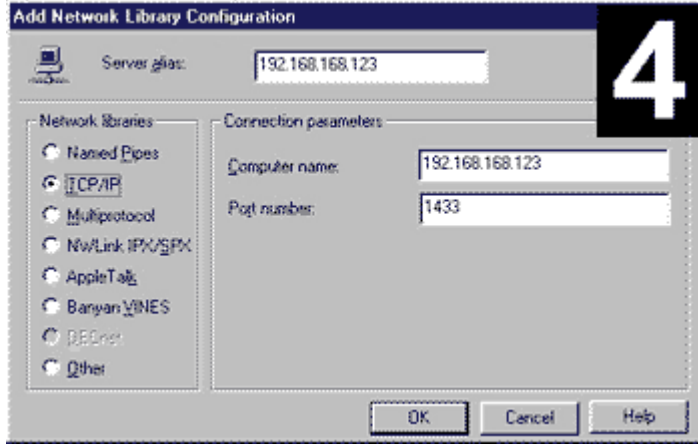
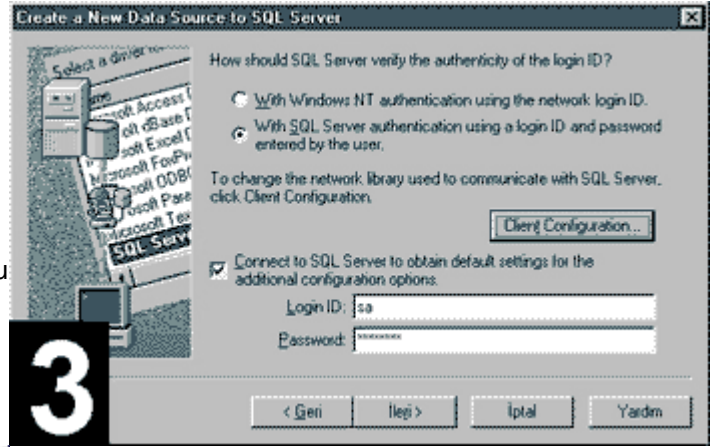
ODBC seçeneklerinde yeni bir System DSN (System Data Source) bileşeni oluşturup, bunu var olan SQL Server sunucunuzla ilişkilendirin. System DSN oluşturma ekranındaki ağ bağlantı seçeneklerinden TCP/IP'yi (Client Configuration – Network Libraries) seçmeyi ihmal etmeyin.



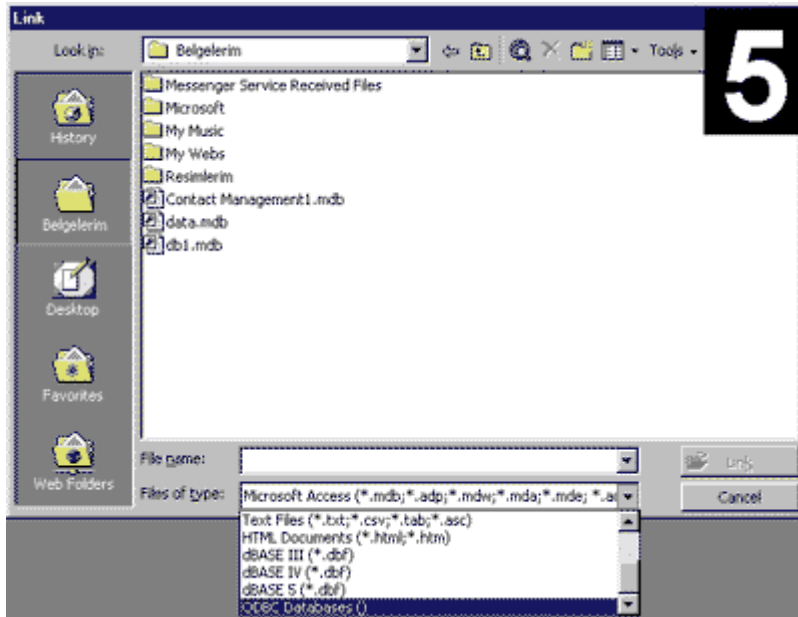
Yeni ODBC DSN'sini oluşturma işlemi tamamlandıktan sonra Access'teki tablo bağlama penceresinde dosya türleri listesinden ODBC Data Sources seçeneğini seçin. Yeni açılan pencerenin Machine Data Source sayfasında, az önce oluşturduğunuz yeni DSN'yi işaretleyin. Sonrasında çıkan seçenekleri de işaretleyip SQL Server tablonuzu kayıt ekleme, silme ya da değiştirme için rahatlıkla kullanabilirsiniz.

Sadece SQL Server değil, ODBC sürücüsü olan Oracle, Paradox, FoxPro, DBase ve diğer tüm veritabanı sistemlerindeki tabloları Access'in içinden ya da başka yerlerden kolaylıkla kullanmak mümkün.

Tablo bağlama işlemi yapmayacaksanız veya buna zorunlu değilseniz ODBC bağlantısı ile çalışmak öncelikli tercih olmamalı. Zira doğrudan bağlantı kurabilen ADO ve sair sistemler ODBC'den daha hızlı çalışır.



ODBC (Open Database Connectivity) farklı veritabanı sistemlerinin her yerden aynı şekilde kullanılabilmesine imkan veren bir ara bağlantı modülüdür. Bununla Access içine bağladığınız bir Paradox tablosunu Access'e hissettirmeden ve başka bir ayar yapmadan bir anda Oracle içinden kullanabilmek mümkündür. Zaten Access kullandığı bir bağlı tablonun hangi sistem üzerinde olduğunu umursamaz.



SQL Server tablolarını bağlamak için ODBC Databases'i seçin.

Bazı tabloların sadece okunur olarak görüntülenebileceğini unutmayın. Metin dosyalarını standart veritabanı tabloları gibi yönetmek kolay olmadığı için kayıt değişikliği yapılmasına izin verilmemesi normal. Aynı şekilde çok büyük metin dosyalarını kullanırken iyi bir performans beklememelisiniz. Zaten veritabanı sistemleri varken büyük metin dosyalarını tablo olarak kullanmaya ihtiyaç yoktur genelde.

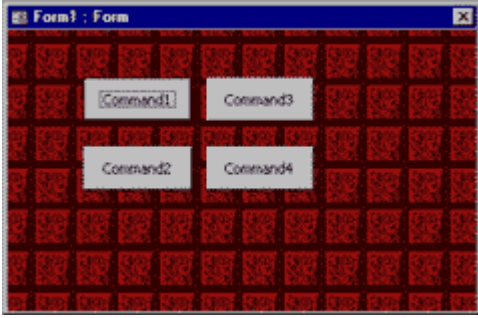
FORM ÜZERİNE RESİM DÖŞEMEK

Formların ve raporların üzerine resim yerleştirmek için özellikler penceresinde birkaç seçenek var. Orada

“picture” ile başlayan altı seçeneği inceleyip nasıl kullanıldıklarını ve ne işe yaradıklarının hemen öğrenebilirsiniz.

Resmi döşemek için Picture Tiling özelliği kullanılıyor. Picture özelliğine resim dosyasının konumu yazılıyor. Hemen altındaki Picture Type seçeneğindeki Embedded değeri bu resim dosyasının Access dosyasının içine fiziksel olarak dahil edileceğini belirtiyor. Aksi halde orada Linked değerinin seçili olması resim dosyasının Access dosyasına dahil edilmeyeceğini, resmin belirtilen disk konumundan her seferinde yeniden çağrılıp görüntüleneceğini işaret ediyor. Tabii bu seçenek, görüntüleme süresini biraz artırıyor. Resmin anında görüntülenmesi gerekiyorsa Embedded seçeneğini işaretleyin.

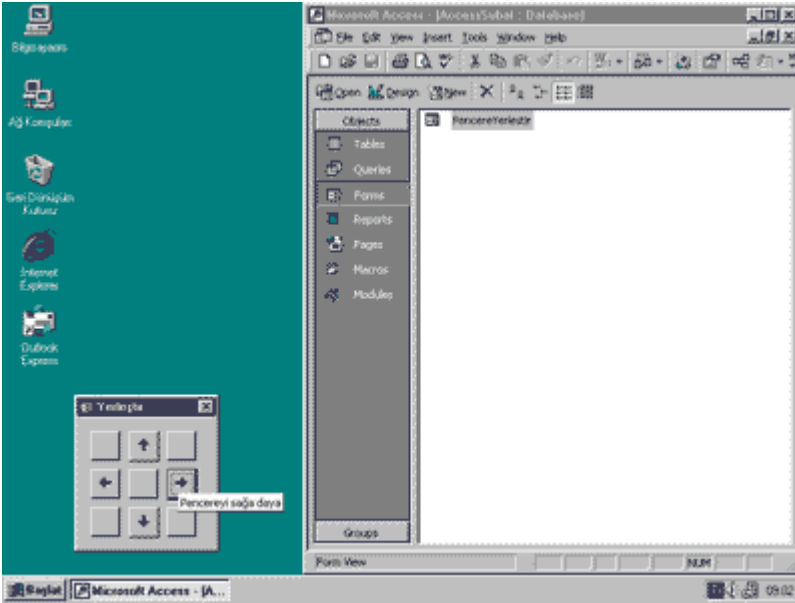
Raporlara özgü Picture Pages seçeneği ile arkaplan resminin sadece ilk sayfada ya da tüm sayfalarda görüntülenmesini sağlayabilirsiniz.



PictureTiling özelliği ile arkaplan resmi form üzerine döşeniyor.

ACCESS PENCERESİNİ YERLEŞTİRMEK

Geçen aylarda Access penceresinin masaüstünde kolayca yerleştirilebileceğiyle ilgili bir kod örneği vardı. Bu sayıdaki örnek ise eskisinin biraz daha gelişmiş hali. Kod biraz uzun olduğu için bu sayfalara kıyamıyorum. Dergide Adı Geçen Dosyalar bölümündeki dosyada kodu bulabilirsiniz.



Access penceresi yerine herhangi başka bir program penceresinin boyutlarını değiştirmek isterseniz koddaki hWndAccessApp yerine ilgili pencerenin Handle (hWnd) değerini yazmanız yeterli.

Bunun için FindWindow gibi bir API komutuna ihtiyacınız olabilir. Onu da şu şekilde yapabilirsiniz:

```
Public Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
```

```
Sub Deneme()
```

```
Dim h As Long
h = FindWindow("Notepad", "Adsız - Not Defteri")
MsgBox "Not Defteri'nin HWND değeri = " & h
End Sub
```

TUŞ BASILI MI DEĞİL Mİ

Windows'ta ve diğer programlarda olduğu gibi Ctrl, Shift ve Alt tuşlarını kısayollara eklemek ve işlem anında bunların basılı olup olmadığını kontrol etmek profesyonel bir görünüm kazandırabilir programlarımıza.

Örneğin bir sıralama işlemi yaptırırken Shift tuşu basılı ise sıralamayı tersten yaptırırım.

Yeni bir modül oluşturup en üstteki Option Compare Database ve varsa Option Explicit satırlarının altına şu satırları yerleştirin:

```
Public Declare Function GetKeyState Lib "user32" (ByVal nVirtKey As Long) As Integer
Public Const VK_SHIFT = &H10 'SHIFT
Public Const VK_CONTROL = &H11 'CTRL
Public Const VK_MENU = &H12 'ALT
```

Programınızın herhangi bir yerinde, basılı olup olmadığını bulmak istediğiniz tuşlar için şu kodu kullanabilirsiniz artık:

```
If GetKeyState(VK_SHIFT) < 0 Then MsgBox "SHIFT tuşu basılı."
If GetKeyState(VK_CONTROL) < 0 Then MsgBox "CTRL tuşu basılı."
If GetKeyState(VK_MENU) < 0 Then MsgBox "ALT tuşu basılı."
```

KAYIT SİLME KODUNUN SON SÜRÜMÜ

Daha önce iki kez farklı kayıt silme kod örnekleri vermiştim. O kadar satırlık kod parçası iki satıra sığabiliyor. İşte son sürüm kayıt silme kodu:

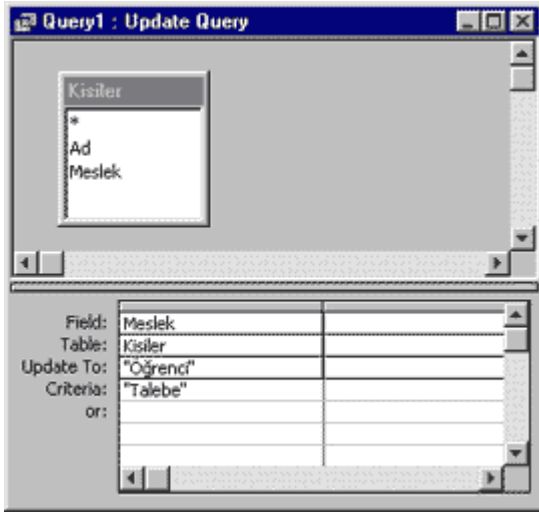
```
DoCmd.SetWarnings False
DoCmd.RunCommand acCmdDeleteRecord
```

HERKES İÇİN SORGULAR

Sıkça sorulan bir SQL sorusu, belli kayıtların bulunup değiştirilmesi ile ilgili.

Access'te yeni bir sorgu oluşturup Query (Sorgu) menüsünden Update Query'yi (Değiştirme Sorgusu) seçin.

Değiştirmek istediğiniz veri alanlarını alt taraftaki sütunlara yerleştirip UpdateTo kısmına yeni değerleri yazın. Sadece belli kayıtları değiştirmek istiyorsanız Criteria (Koşul) alanını doldurmayı unutmayın. Aksi halde bütün kayıtlarınız değiştirilir.



Resimdeki sorgu Meslek alanındaki "Talebe" değerlerini "Öğrenci" olarak değiştiriyor.

UPDATE Kisiler SET Meslek = "Öğrenci" WHERE Meslek="Talebe";

Değiştirilecek alan için koşul belirtmek zorunda değilsiniz. Bu kod Kisiler tablosunda Meslek alanı değerinin "Ev Hanımı" olduğu kayıtlardaki Cinsiyet alanına "Kadın" değerini ve Askerlik alanına "Muaf" değerini yazıyor:

UPDATE Kisiler SET Cinsiyet="Kadın", Askerlik="Muaf" WHERE Meslek="Ev Hanımı";

Koşul belirtmeye gerek yok tabii. Şu SQL kodu ile tüm müşteri kayıtlarındaki MektupGonder alanı True (Evet) olarak güncelleştirilir:

UPDATE Musteriler SET MektupGonder = TRUE;

ODE TOOLS

Şu sıralar **ODE Tools** ile ilgili sorular çoğaldı. İşte kısa cevaplar:

- ODE Tools 2000 = Microsoft Office 2000 Developer Tools.
- ODE Tools ile Access kurulu olmayan bilgisayarlarda Access'te yapılmış programları çalıştırabilirsiniz.
- Access dosyaları EXE yapılmıyor. Ama kodlar ve tasarım bilgileri tamamen gizlenip dosya derlenebiliyor. Bunun için ODE Tools'a gerek yok. Dosyayı MDE dosyası yapmalısınız.
- Program ile Access projeleri için kurulum paketi (setup) hazırlanıyor. Tabii sadece bu işi yapmıyor. İçerisinde Office'te programlama ile ilgili başka eklentiler de var.
- ODE Tools'u bir defa satın almış olmakla, bununla hazırladığınız kurulum programlarınızı sınırsız sayıda ve tamamen ücretsiz dağıtma hakkınız olur. Kurulum programlarınız her bilgisayara, yasal bir kısıtlama olmaksızın kurulabilir.
- Ayrı olarak satılabiliyor ve ayrıca Office 2000 Developer Edition paketi içinde yer alıyor. Office 97 için de aynı şekilde sürümleri mevcut. Fakat ODE Tools 97 şu an piyasada değil.
- Programı Linksoft'tan satın alabilirsiniz. Tel: 0(216) 522 00 22.
- Lisans kontrollerinde öncelikle faturaların incelendiğini unutmayın.
- Programı satın almadan önce PC Magazine'in Temmuz 2000 ve Ocak 2001 tarihli sayılarındaki ODE Tools incelemelerine gözetmanızı tavsiye ederim.

SORULARLA İLGİLİ

Okurlarımızın gönderdiği soruların hemen hepsine cevap yazmaya çalışırım. Fakat bir yığın sorunun tamamına kısa sürede cevap yazmak mümkün değil. Çoğunlukla araştırma yapmak gerekli ve mesai sırasında işten çokça vakit ayırmak pek doğru olmasa gerek. Gelen mesajları vaktimin yettiği kadarıyla akşamları evden yanıtlayabiliyorum genelde.

Sadece bana gönderdiğiniz postalarda değil, başka yerlere gönderdiğiniz postalarda da mesajlarınızın içindeki soruları kısa tutmanız ve birçok soruyu tek bir mesaj içinde toplamamanız cevap alma sürenizi azaltabilir. Ayrıca kullandığınız Access'in (yani probleme konu olan programın) hangi sürüm ve dilde olduğunu belirtirseniz daha

iyi olur.

Haber grupları sorulara cevap bulmak için daha efektif oluyor. Oralarda çok daha kısa sürede ve ayrıntılı çözümler bulunabiliyor. Zira yüzlerce kişi sorunuzu okuyor ve problemin çözümünü bilenler mesajınızı cevaplıyorlar.

Sorularınızın birçoğu haber gruplarında geçmiş tarihlerde başkaları tarafından sorulup cevaplanmış olabiliyor. Problem için öncelikle Internet'ten ve haber gruplarından araştırma yapmak daha hızlı bir çözüm olabilir.

Internet'te de Access'le ilgili Web sayfaları mevcut. Bir daha ki ay için Access'le ilgili kişisel web sitelerinin adreslerini gönderirseniz bu bölümde yayınlayacağız. Bu arada Access'le ilgili kapsamlı bir site olan The Access Web sitesini de sıkça ziyaret etmenizde fayda var.

<http://www.mvps.org/access>

Mvps.org sitesinde Access dışında VB, API, VC gibi konularda çok kapsamlı birkaç bölüm daha var. Site, Microsoft tarafından MVP ünvanı (Microsoft's Most Valuable Professional - bir nevi fahri Microsoft mühendisliği) verilmiş kişiler tarafından oluşturulmuş. Oradaki diğer bağlantılar arasında başka MVP'lerin siteleri için de adresler verilmiş. Outlook Express için epey kapsamlı bilgiler içeren bir MVP sitesi bile varmış nitekim.

Okurlarımızdan gelen bazı soruların cevapları ise eski sayılarımızda yer alabiliyor. Mayıs 2000'de başlayan bu dizinin önceki yazılarını PC Magazine Online sitesindeki Eski Sayılar Arşivi bölümünde bulabilirsiniz. Bu yazıları da elinizin altında bulundurmanız faydalı olacaktır.

PC Magazine sitesindeki Dergide Adı Geçen Dosyalar bölümünde ise aynı tarihten başlayacak şekilde bazı aylara ait Access'le ilgili örnek çalışma dosyaları yer alıyor.

INTERNET EXPLORER MENÜLERİ NİYE ÇALIŞMIYOR

Kimi zaman Internet Explorer'daki menüler durup dururken açılmaz oluyor. Aksi gibi açık olan diğer pencerelerin bir kısmında da aynı şey oluyor. Menü çubuğunu bir kenara yaslayıp kısaltarak menü oku ile menüyü açmak mümkün. Daha kolayı, iki kez F11 tuşuna basmak yeterli.

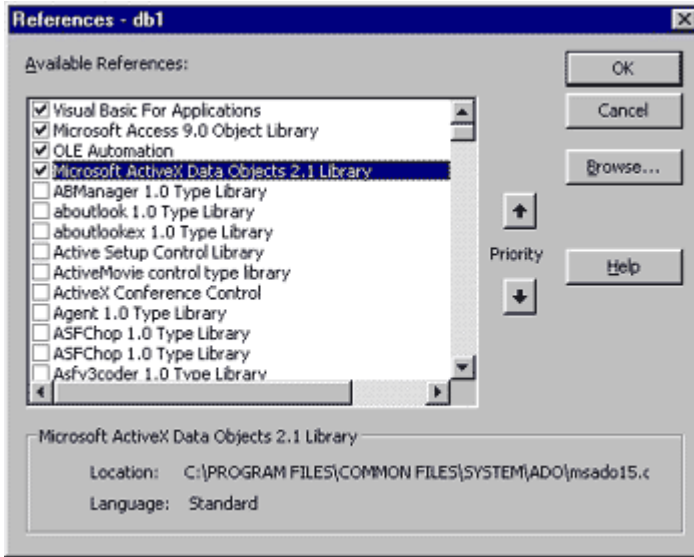
DOSYA:

[AccessNisan2001.zip](#)

DAO Gitti ADO Geldi

DAO gideli, ADO geleli epey oldu. Birçokları için artık eskimiş ADO'yu evirip çeviriyoruz bu ay.

Access'teki başvurular listesinde varsayılan olarak ADO seçili. Emektar DAO'nun yerinde yeller esiyor gördüğünüz gibi. Zaten Microsoft'un artık DAO'nun geliştirilmeyeceğini duyurduğunu biliyoruz. Madem öyle, işte ADO'nun bir kısım marifetleri.



Bir Access 2000 dosyasını ilk oluşturduğunuzda başvurular listesinde geçerli olarak ADO (Microsoft Activex Data Objects 2.x Library) seçili geliyor. Office 2000 ilk kurulduğunda 2.1 sürümü görünüyor. Bu yazının yazıldığı sıralarda ADO'nun 2.5 sürümü epey yaygındı.

ADO, Microsoft'un sitesinden ücretsiz yüklenebiliyor. Veyahut MS'nin bazı kurulum paketlerinin içinde ADO'nun kurulum dosyası çıkabiliyor. Visual Studio 6.0 Service Pack 4 kurulum dosyaları arasında yer alıyor bu dosya. (PC Magazine Offline Ekim 2000 CD'sinde Visual Studio 6 SP4'ü bulabilirsiniz.)

ADO'nun kurulum dosyasının adı MDAC_TYP.EXE ve dosyanın büyüklüğü 8 mb. civarında. ADO kullandığınız projenizi dağıtmak istediğinizde bu kurulum dosyasını da vermeniz gerekebileceğini unutmayın. Tabii ki sınırsız dağıtılabilir.

ADO'nun en büyük yeteneklerinden biri Internet veya yerel ağ ortamında bile SQL Server sunucusuna kolaylıkla bağlanabilmesi.

ADO İLE BAŞLANGIÇ

DAO'nun kolay nesnelere yeterince alışmıştık.

```
Dim DB as Database  
Dim Rs as Recordset  
Set DB= CurrentDB()  
Rs.FindFirst  
Rs.FindNext  
...
```

şeklindeki deyimler neredeyse tamamen değişti. Access'te, ASP'de ya da herhangi bir VBA platformunda ADO'yu aşağıda anlatıldığına benzer şekillerde kullanabilirsiniz.

ADO ile bir tabloyu kullanmak için aktif proje dosyasının Connection nesnesine bir başvuru yapılıyor.

```
Dim rs As New ADODB.Recordset  
rs.Open "Table1", CurrentProject.Connection
```


Bir tabloyu ya da veri kümesini açmak ve kullanmak için eskiden olduğu gibi birçok alternatif var. Ama ADO'da bu alternatifleri biraz daha iyi bilmemiz gerekiyor.

Bir tabloyu sadece okunabilir olarak açmak:

```
rs.Open "Table1", CurrentProject.Connection, adOpenKeyset, adLockReadOnly
```

Bir tabloyu kayıt ekleme, silme ya da değiştirme için açmak:

```
rs.Open "Table1", CurrentProject.Connection, adOpenKeyset, adLockOptimistic
```

Sadece okunabilir bir SQL kodu açmak:

```
rs.Open "SELECT * FROM Table1 WHERE ...", CurrentProject.Connection, adOpenKeyset, adLockReadOnly
```

Kayıt eklenebilecek, silinebilecek ya da değiştirilebilecek şekilde bir SQL kodu açmak:

```
rs.Open "SELECT * FROM Table1 WHERE ...", CurrentProject.Connection, adOpenKeyset, adLockOptimistic
```

KAYIT DEĞİŞİKLİKLERİ

Kayıt ekleme yapılabilmesi için tablo açılırken adLockOptimistic parametresi kullanılmalı.

```
rs.Open "Table1", CurrentProject.Connection, _  
adOpenKeyset, adLockOptimistic  
rs.AddNew  
rs("Field1") = "Test"  
rs("Field2") = 1234  
rs.Update
```

Kayıt silme eskisi gibi:

```
rs.Delete
```

Kayıt değiştirmek için ise ADO'nun DAO'ya göre bir farkı var. Değişikliği yapmadan önce artık Edit metodu kullanılmıyor ve doğrudan değişikliğe geçiliyor.

```
rs.Open ...  
rs("Field1") = "Test"  
rs("Field2") = 1234  
rs.Update
```

ADO İLE SQL KODU ÇALIŞTIRMAK

Artık Database nesnesi yerine Connection nesnesi kullanılıyor. CurrentProject.Connection deyimini kullanarak bir SQL kodunu çalıştırmayı (Execute) deneyin.

```
Dim conn As ADODB.Connection
```

```
Set conn = CurrentProject.Connection  
conn.Execute "UPDATE Table1 SET ad = 'deneme' WHERE (ID='1');"
```

ya da

```
CurrentProject.Connection.Execute "UPDATE Table1 SET ad = 'deneme' WHERE (ID='1');"
```

SQL İLE ARAMA YAPMAK

SQL kodu içinde arama yapmak için WHERE dışında LIKE deyiminin de kullanılabileceğini biliyorsunuzdur. Eskiden LIKE deyiminde metin alanlarında ayrıntılı arama için * işareti kullanılırdı. Şimdiyse bunun yerine % işareti kullanılıyor. (SELECT deyiminin yanında * işareti kullanılır. Bu değişiklik sadece LIKE deyimini için geçerli.)

Şu deyim ile bir SQL kodunu kayıt değişiklikleri yapılabilecek şekilde açabilirsiniz:

```
rs.Open "SELECT * FROM Table1 WHERE (AdiSoyadi Like 'A%')";  
CurrentProject.Connection, adOpenKeyset, adLockOptimistic
```

En sondaki adLockOptimistic deyimini adLockReadOnly yaparsanız kayıt kümesi sadece okunur olacaktır.

LIKE deyiminin yanındaki 'A%' ifadesi ile AdiSoyadi alanında 'A' ile başlayan kayıtlar seçiliyor. Değiştirip '[A-K,U-Z]%' şeklinde kullanmanız da mümkün. Yani hem A-K arası, hem de U-Z arası değerlerin olduğu kayıtlar elde ediliyor.

LIKE deyiminden sonra bir de şunu yazmayı deneyin:

```
'%A_[M,T]%'
```

Bu şekilde yapıldığında herhangi bir yerinde 'A' ve yanında herhangi bir karakter olup onun yanında da 'M' veya 'T' olan değerlerin bulunduğu kayıtlar listelenecektir. Mütevazı LIKE deyiminin eskiden kalma daha birçok yeteneği var. Buna ilk olarak Word6.0'da şahit olmuştum. LIKE ile ilgili arama yeteneklerini Word'ün bul/değiştir sayfasında görmelisiniz.

TABLO YA DA KAYIT KÜMESİ ÜZERİNDE ARAMA YAPMAK

Kayıt kümesinde arama yapmaktan kasıt, SQL kodu kullanmadan, elde ettiğimiz kayıt kümesi (RecordSet) açıldıktan sonra arama yapılması. Eskiden FindFirst, FindNext gibi komutlar kullanıyorduk. Bu komutların tümü Find adlı yeni komut altında toplanmış. Find komutu değişik parametreler alarak eski komutların işini yapabiliyor.

Arama işlemini başlatmadan önce tabloda kayıt olup olmadığı kontrol ediliyor. Zira kayıt yoksa MOVEFIRST komutu hata verir. Zaten herhangi bir anda ilk kaydı bulmak için tablonun başına gidilmesi gerekli.

```
Dim rs as New ADODB.Recordset  
rs.Open "Kisiler", CurrentProject.Connection, _  
adOpenKeyset, adLockOptimistic  
If rs.EOF = True Then
```

```
MsgBox "Tabloda kayıt yok."
Else

rs.MoveFirst
rs.Find Kriter, 0
'Eski FindFirst ile aynı.

If rs.EOF = True Then
MsgBox "Kayıt bulunamadı."
Else
'Tüm kayıtları sırayla bul ve mesaj göster.
Do While Not rs.EOF
MsgBox "Kayıt bulundu." & vbCrLf & vbCrLf & _
rs("ID") & " " & rs("Ad") & " " & rs("Soyad")
'Sonraki kayıttan itibaren tekrar ara.
rs.Find Kriter, 1
Loop
End If

End If

rs.Close
Set rs = Nothing
```

İlk kaydı bul (FindFirst):

```
rs.MoveFirst
rs.Find “Kriter”, 0, adSearchForward
```

Sonrakini bul (FindNext):

```
rs.Find “Kriter”, 1, adSearchForward
```

Öncekini bul (FindPrevious):

```
rs.Find “Kriter”, 1, adSearchBackward
```

Son kaydı bul (FindLast):

```
rs.MoveLast
rs.Find “Kriter”, 0, adSearchBackward
```

DAO'da kayıt bulunamadığında RecordSet nesnesine ait NoMatch özelliği kullanılabilirken ADO'da böyle bir özellik yok. Aradığınız kaydın bulunup bulunamadığını anlamak için, eğer ileri doğru arama yapıyorsanız kümenin sonunda (rs.EOF=True ?) olup olmadığını, geri doğru arama yapıyorsanız kümenin başında (rs.BOF=True ?) olup olmadığını kontrol etmelisiniz.

BOŞ KAYITLAR GELSİN

NULL deyimini boş değerleri yani ‘hiçbirşey’i ifade ediyor. Eskiden bir veri alanının değerinin boş olup

olmadığını bulmak için 'VeriAlani1 IS NULL' yazılıyordu. Ya da boş olmayan kayıtlar 'VeriAlani1 IS NOT NULL' ile listelenebiliyordu. ADO'nun SQL kodlarında ise bu deyimle beraber eşitlik işaretleri kullanılıyor.

'VeriAlani1 = NULL'
'VeriAlani1 <> NULL'

Son olarak NEW ya da CREATEOBJECT deyimi ile oluşturduğunuz Connection ve Recordset nesnelerini NOTHING ile silmeyi unutmayın.

Set rs = Nothing
Set conn = Nothing

ADO o kadar ayrıntılı ki tam olarak anlatana kadar neredeyse bir dergi dolusu yazı çıkar.

BİR YILIN TÜM GÜNLERİNİ LİSTELEMELİK

Bir yılın tüm günlerini bulmak için yerine göre birkaç yöntem önerilebilir. Bu yöntemi kullanmak için iki küçük tablo hazırlamak gerekiyor.

Aylar ve Gunler adlı iki tablo oluşturun. Aylar tablosunda Ay adlı yeni bir sayı alanı oluşturup, tabloya sırayla 1-12 arası sayıları girin. Gunler tablosunda da Gun adlı aynı özellikte bir alan oluşturup bu tabloya 1-31 arası sayıları girin.

Yeni bir sorgu açıp menüden Görünüm / SQL Görünümü (View / SQL View) komutunu seçin. Varsa yazılı olan şeyleri silip şu kodu ekleyin:

```
SELECT Tarih FROM  
(  
SELECT DateSerial([HANGİ YIL?],[ay],[gun]) AS Tarih  
FROM Aylar, Gunler  
) as Tarihler  
GROUP BY Tarih  
ORDER BY Tarih;
```

Tarih
01.01.2001
02.01.2001
03.01.2001
04.01.2001
05.01.2001
06.01.2001
07.01.2001
08.01.2001
09.01.2001
10.01.2001
11.01.2001
12.01.2001
13.01.2001
14.01.2001
15.01.2001
16.01.2001
17.01.2001
18.01.2001
19.01.2001
20.01.2001
21.01.2001

2001 yılının tüm günleri.

Sorguyu bir adla kaydedip kapadıktan sonra veri görünümünde açıp bir yıla ait tüm günleri listeleyebilirsiniz. Alttaki SQL metnini ise bir yıla ait çalışma günlerini (haftaiçi olan günleri) bulmak için aynı şekilde kullanabilirsiniz.

```
SELECT Tarih FROM
(
SELECT DateSerial([HANGİ YIL?],[ay],[gun]) AS Tarih
FROM Aylar, Gunler
) AS Tarihler
WHERE NOT Weekday([Tarih]) IN (1,7)
GROUP BY Tarih
ORDER BY Tarih;
```

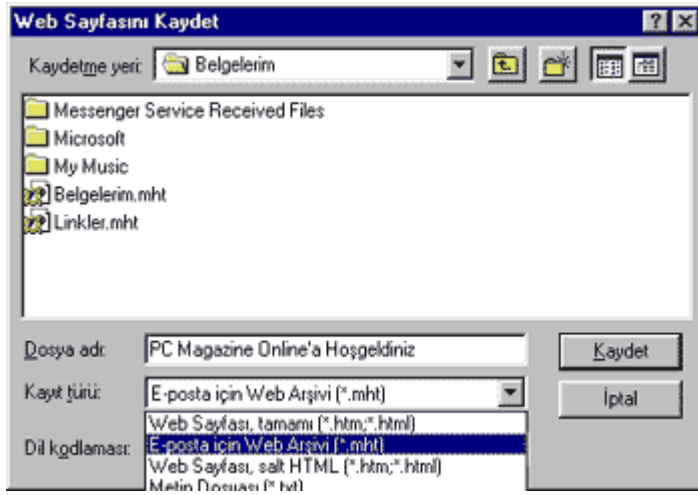
Tarih
01.01.2001
02.01.2001
03.01.2001
04.01.2001
05.01.2001
08.01.2001
09.01.2001
10.01.2001
11.01.2001
12.01.2001
15.01.2001
16.01.2001
17.01.2001
18.01.2001
19.01.2001
22.01.2001
23.01.2001
24.01.2001
25.01.2001
26.01.2001
29.01.2001

2001 yılında haftaiçi olan günler.

WEB SAYFASINI KAYDET

Internet Explorer’da gezinirken beğendiğiniz bir sayfayı tüm bileşenleri ile kaydedebiliyorsunuz. Ama bu özelliği çokça kullanıyorsanız etrafta bir sürü dosya ve aynı adı taşıyan klasörler birikmiş olabilir. Çünkü geçerli olarak sayfa içindeki resimler ve diğer ekler bir klasör içine yerleştiriyor. Gelgelelim sayfayı kaydederken değiştireceğiniz bir seçenekle herşeyi tek bir dosyada toplayabilirsiniz.

Dosya menüsünde sayfayı kaydetme komutunu çalıştırın ve açılan sayfanın alt kısmında, kayıt türü seçeneği olarak MHT uzantısını seçin. Böylelikle sayfadaki yazı ve resimlerin tamamı tek dosya içinde tutulacak ve kalabalıktan kurtulacaksınız. Tabii ki Internet Explorer bu dosyaları tekrar görüntüleyebiliyor. Üzerlerine çift tıklamanız yeterli.



Bu yazıdaki bazı örnekleri PC Magazine Online sitesinde Dergide Adı Geçen Dosyalar bölümünde bulabilirsiniz.

ACCESS HİKAYELERİ

Bir sektörün büyüklüğünün bir kanıtını literatürü etkileyebilmiş olmasında ararım. İyi veya kötü yönde olsun, dilimizi büyük ölçüde etkilediğini düşündüğüm bir yazılım sektörü bu.

ADINIZI GİRMİYİ UNUTTUNUZ

Yaptığınız programı sevdirmenin yollarından biri belki kullanıcıya hitap şeklindeki incelikler olsa gerek.

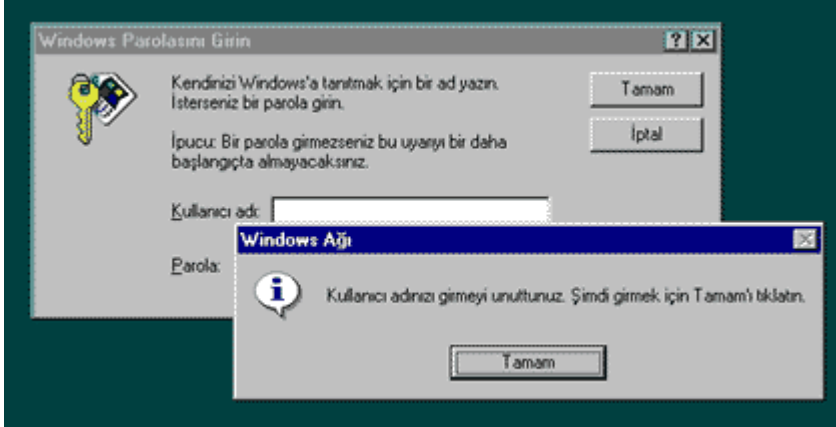
Kullanıcı etkileşimli programlar yapıyorsanız gösterilecek mesajları, komut metinlerini öncelikle kendinize yönlendirerek ufak denemeler yapabilirsiniz.

“Adınızı girmeyi unuttunuz. Lütfen girin.”

(Benim adımdan sana ne? Hem unuttuğumu da nereden çıkardın??!)

Unutmayın ki programınız bir gün paranoid bir kullanıcının eline geçebilir. Programın karşısındaki

kişiyi kendisinin birşeyi unuttuğunu söylediğinizde programınız parantez içindekinden farklı sözlerle de muhatap olabilir.



Sizce gerçekten unutmuş olabilir miyim?!!

Resimdeki unutma hadisesi Windows 95'in eski bazı sürümlerinde var. Güncel sürümlerde buna rastlamadım.

ŞİFRE Mİ PAROLA MI

Dikkat ettiyseniz Windows'ta kullanıcı hesaplarının olduğu yerlerin neredeyse hiçbir yerinde Şifre kelimesi kullanılmıyor. Kullanıcı adı ile birlikte Parola sözcüğü yer alıyor. Şifre kelimesi ise güvenlik amacıyla değiştirilmiş olan şeyler sözkonusu olduğunda kullanılıyor.

Baktığım sözlükte bu iki sözcük şöyle açıklanmış:

ŞİFRE = [Kökeni: Fransızca] [Türü: İsim] Gizli haberleşmeye yarayan işaretlerin tümü.
"Kumandan şifreyi söyledi" İstanbul mümessilliği şifresiyle Mustafa Kemal Paşa'ya bekledikleri malûmatı ilettiğim." (Y. K. Karaosmanoğlu)

*şifre anahtarı= şifrede kullanılan işaretleri gösteren liste

*şifreyi çözmek= bir şifrede kullanılan işaretlerin anlamını bulmak

PAROLA = [Kökeni: İtalyanca]

1. [Türü: İsim] Askerlerin veya gizli derneklerin toplantılarına katılan kimselerin birbirlerini tanımalarını sağlayan ve kendi aralarında önceden kararlaştırdıkları kelime veya söz.

"Bir asker uzaktan, görünmeyen bir yerden parola soruyordu." (M. Ş. Esenal)

"Korkulu mıntakalara geldiğimiz vakit, dağıldık, uzaktan ses parolamız Bağdad, cevabı Şam kelimesiydi." (F. R. Atay)

2. [Kullanım alanı: Askerlik] Varılmak istenen amacı özetleyen söz.

3. Gizlilik ortamında insanların birbirini tanımalarını ve anlaşmalarını sağlayan işaret.

"Eşinin balkona asacağı çamaşırların parolasından İngiliz polisinin o gün kendini evde arayıp aramadığını ve civarda nöbet.." (H. Taner)

Buradan anladığım kadarıyla "Şifre" kelimesi çözülmesi gereken gizli bir şeyi (kilit), varılmak istenen amacı ifade ediyor. Parola ise şifreyi çözen bir anahtar anlamı üstleniyor.

DİKKATSİZ MESAJLAR

Windows'ta olduğu gibi farklı dillerde piyasaya sürülecek bir program yazıyorsanız ya da tüm mesaj

metinlerini bir yerden yönetiyorsanız biraz daha kontrollü davranmanız gereklidir. Şu mesaj Windows 95'in güncelleştirilmemiş bazı eski sürümlerinde var:

“Bu klasörü silmek ve Tüm içeriği Geri Dönüşüm Kutusu'na taşınır mı? istediğinizden emin misiniz?”



Belli ki mesaj içine bazı parametreler yerleştirilmiş ama metnin parçaları doğru yerlerde değil.



Delphi kodlarının kağıda dökülmüş hali. İşte dört sayfanın beşinci sayfası!)

TIKLASIN FARELER

İlkokul yıllarında dillerin nasıl geliştiğini araştırırken doğadaki seslerin dilimizi ve yazımızı etkilediğini öğrenmiştik. Bakınız bu bilgisayar faresi dilimize ne kadar da kolay adapte olmuş! Artık bilgisayar ya da Internet dediniz mi akla ‘tık’ geliyor, farenin iyisi ‘tık’ının şahaneliliğinden, Internet bağlantısının iyisi, reklamındaki ‘tık’ların o yana bu yana ahenkli uçuşmasından anlaşıyor.

Microsoft da Türkçe programlarındaki metinlerde ‘tıklama’ eylemini ‘tıklatın’ şeklinde kullanıyor.

“Lütfen buraya tıklayın.”

“Denetim Masası simgesine çift tıklayın.”

Gördüğüm kadarıyla üstteki iki cümlemin Microsoftçası şöyle:

“Lütfen burayı tıklatın.”

“Denetim Masası simgesini çift tıklatın.”

Microsoft bu mesajları yorumlarken tıklayan şeyin biz değil, fare olduğunu düşünmüş olabilir. Yani ‘siz tıklama işini fareye yaptırın’ demek isteniyor. Biz de diyoruz ki, ‘Sayın kullanıcı. Fareyi eline al ama onu kullanma, koş burayı sen tıkla!’. Acaba gerçekten öyle mi?

KULLANICI HATA YAPMAZ – MI ACABA

Kimilerince programı kullanan kişi hata yapmaz. Hatayı programı üreten kişi yaptırır. Kullanıcı ancak bir işlem yaparken o işlemin yetki dahilinde geçerliliği konusunda hata yapmış olabilir bu görüşe göre.

Geçen yıllarda bu şekilde bazı tartışmalar yapıyordu. Sonrasında yazılım firmaları kullandıkları dilde bazı iyileştirmeler yapmışlardı. Aynı tartışmalar Macintosh için de geçerliydi.

YARDIM DOSYALARI AMA NASIL

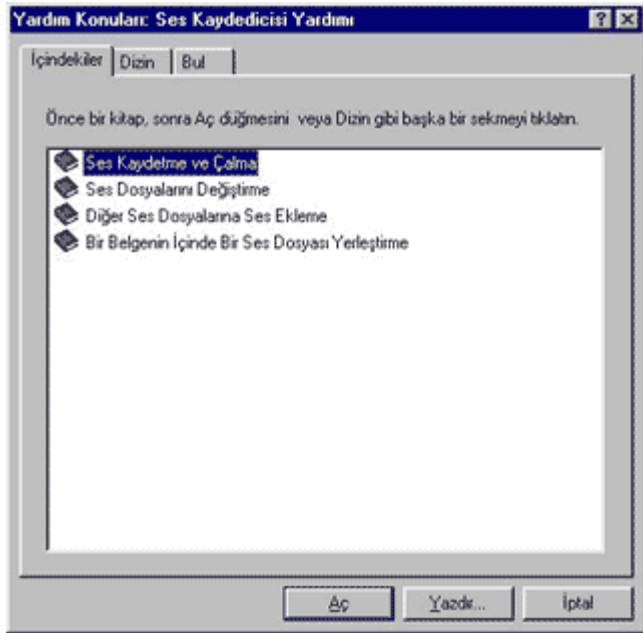
Türkiye’de üretilmiş yazılım programlarının hemen hepsindeki yardım dokümanlarında dikkatimi çeken birşey var. Aynı şey birçok web sayfası için de geçerli. Yardım dokümanları hazırlanırken aklımıza ilk gelen şey ekranlarda görünen nesnelerin ne işe yaradığını anlatmaktır.

Microsoft gibi büyük şirketlerin dokümanlarında dikkat ettiyseniz görünür işlevlerin ne işe yaradığı değil, **ihtiyaçların o ürün ile nasıl karşılanacağı** konusu ön plandadır.

Bizim yardım dokümanlarımızda ise ekrandaki komutların ne işe yaradığından başlanıp küçük ipuçlarıyla yardım belgeleri tamamlanmış olur genelde.

Bu yazıyı yazdığım Word programının yardım belgelerinde önce Word’ün ne işe yaradığından, önceki sürümlere göre farklılıklarından başlanıp Word’ün önemli yetenekleri ile devam edilmiş.

Çok kapsamlı bir program olduğu için bunun böyle yapılmış olduğu söylenebilir. Öyleyse Windows’taki küçük Ses Kaydedicisi’nin yardım dosyasına bir gözatalım.



Şu dört konu başlığına bakın:

- **Ses Kaydetme ve Çalma**
- **Ses Dosyalarını Değiştirme**
- **Diğer Ses Dosyalarına Ses Ekleme**
- **Bir Belgenin İçinde Bir Ses Dosyası Yerleştirme**

Gördüğünüz gibi son konu olarak başka uygulamalara nasıl ses eklenebileceğini anlatmayı ihmal etmemişler. Yani konular sadece görünen işlevlerle ve menülerin ve komutların nasıl kullanıldığıyla sınırlı değil.

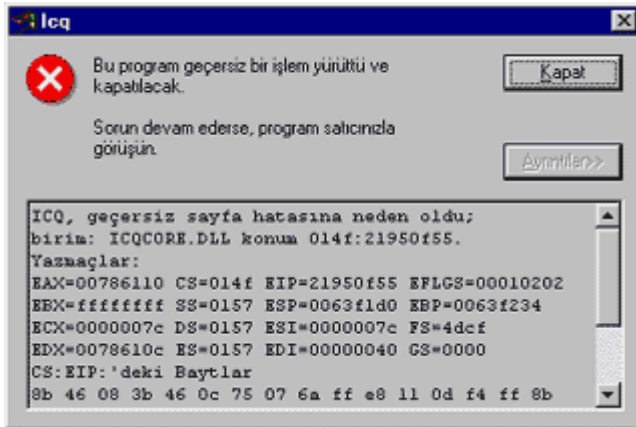
Bir muhasebe programında ‘FATURA MENÜSÜ NASIL KULLANILIR’ konusunu değil de ‘FATURA NASIL HAZIRLANIR’ konusunu öncelikle işleyen yardım belgelerinin olmasını daha doğru buluyorum. Tabii programınızla beraber yardım belgeleri vermiyorsanız böyle bir sorunuz da yoktur haliyle.



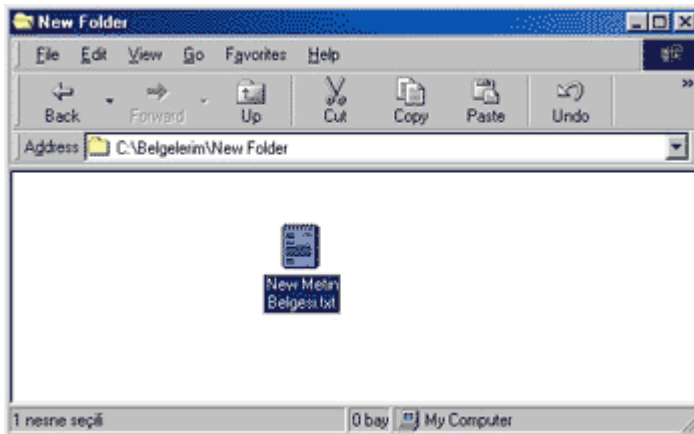
Az da olsa birşeyler çağrıştırıyor.



Mesaj metninin dikkatli bir şekilde yazıldığı belli oluyor.



Virgül gereksiz gibi duruyor ama mesaj olması gerektiği gibi.



Hangi dili kullanacağına karar veremeyen bir işletim sisteminden.

DOSYA:

[AccessSubat2001.zip](#)

Eski Defterler

Yeni yıla girildiğinde eski defterler karıştırılır ve önceki yılın muhasebesi yapılır, adettir. Biz de eski defterleri şöyle bir yoklayalım dedik. Yeni şeyleri de ihmal etmedik tabii.

Bu dizinin ilk yazısının yazıldığı zamanlarda Türkçe Access97'deki problemlerin yamalarıyla meşgulduk. Yine o aylarda Access'in meşhur "Otomasyon Hatası" piyasada fink atıyordu. Derken "Otomasyon Hatası"nın bize aslında "Otomasyon Nesnesi"ni ima etmeye çalıştığını farkettilik. Bu da Sınıf Modülleri (Class), Başvuru Kütüphaneleri (Reference Library) ve Add-in'ler gibi yapılara yöneltti bizi.

Yıllardır değişmeyen bir olgu, Türkiye'de kullanılan tarih biçiminin Windows standardıyla aynı olmamasından kaynaklanan SQL kodu yetmezliği. Gerek ASP veya VB kullanıp ADO ile SQL Server'dan ya da başka bir yerden veri almaya çalışırken, gerek Access'in kendi içinde, gerekse Delphi gibi programlarda Paradox ile çalışırken, SQL kodları içinde kullanılan tarih değerlerinin standart verilememesi, epeydir gördüğüm üzere birçok programcının başını ağrıtmıştır.

Bunun çok önemli olduğunu düşündüğüm için tarih değerlerinin aslında sayı olduklarına değinmiş ve SQL kodu içinde tarih değerinin sayısal karşılığını yazmanın garantili bir çözüm olduğundan bahsetmiştilik. Şu an da bundan bahsediyor olmamın sebebi aynı problemin hala birçok yerde yaşandığını görmemdir.

Projelerimizi dağıtma zamanı geldiğinde Access'in kendisine gerek kalmadan Access uygulamalarını çalıştırabilmek için Microsoft'un çıkardığı ODE Tools adlı program ile tanışmıştık. Access dosyası EXE olur mu sorusu sıkça sorulmaya başlandığında, bu programın EXE yapmadığını ama Access dosyalarını EXE gibi çalıştırabilme yeteneğinin olduğunu ve de Access dosyalarının derlenip kodlarının tamamen gizlenebileceğini görmüştük.

SQL kodu yazmak, Access'te programlama sürecinin ayrılmaz parçalarından olduğu için bunu da yeteri kadar önemseyip ilginç ve işe yarar ipuçlarını inceledik.

Otomatik Sayı (Autonumber) alanları da birçoğumuzun hoşuna gitmeyen tavırlar gösteriyordu. Kayıtlar silindiğinde neden eski numaralar kullanılamaz oluyordu?! Bunun, otomatik sayı alanlarının en temel özelliği olduğu açık. Veri alanının silinip yeniden oluşturulması öneriliyor çok zaman. Fakat bu her zaman yapılamayacak bir şeydi. Yine de Access dosyasının içeriğini kontrol edip belli bir noktasını bulmak bunu sağlıyor. Aşağıda Access dosyalarında otomatik sayı değerinin istenildiği gibi değiştirilebileceğine dair bir örnek bulacaksınız.

Birçok şeyin aslında ayrıntılarda olduğunu görüp belli noktaları yakaladıktan sonra işimize yarayacak şeyler bulmuştuk. Tablo görünümünde başka tablodaki kayıtlardan seçim yaptırabilmek bunlardan biriydi. Yine, tablo kullanmadan elimizdeki değerleri sorgu (query) olarak kullanmak, birleşik sorgular, tablo bağlamaksızın (link) başka bir Access dosyasındaki tabloyu sorguya dahil etmek, tablo görünümünde süzme yetenekleri, geçerlilik kısıtlamaları (validation rules), sistem yazıtiplerini resim olarak kullanmak gibi ayrıntılara rasladık.

Birçok okurumuzun istediği halde, Access'te kod yazmaya geçişten bahsetmemek olmazdı. En azından bir yerden bulunan kod parçasının bir Access programına nasıl ekleneceğinin bilinmesi şarttı.

OTOMATİK SAYI DEĞİŞİR AMA..

Otomatik sayı değerinin değiştirilmesini istemenin sebebi kayıtlara tanım numarası vermekten ziyade görünüşü güzel göstermek olsa gerek.

Aşağıdaki yöntemi sadece otomatik sayıların nasıl işlendiği konusunda bir fikir vermesi açısından dikkate alın. Kullanmanızı tavsiye etmiyorum. Zaten tabloda kayıtlar varken yeni alınacak otomatik sayı değerini sıfırlamak ya da değiştirmek anlam taşımaz.

Küçük bir dosyada otomatik sayı alanı içeren bir tablo oluşturuldu. Bu tabloya 15 kayıt eklendi, bazı kayıtlar silindi. Bu durumdayken dosyanın bir yedeği alındı. Sonra tablo açılıp yeni kayıt satırına gelinerek birşeyler yazıldı ve ESC tuşu ile kayıttan vazgeçildi. Böylelikle dosyanın bir yerine en son kullanılan otomatik sayı değeri yazılmış oluyor. Dosya kapatıldı ve küçük bir kod yazıp iki dosya arasında farklı olan kesimleri araştırdık. Sonuçta bu iki dosya arasında sadece 2 baytlık kesimin farklı olduğunu gördük.

Bunlardan birinin değerini LONG tipinde (4 bayt) okuduğumuzda en son seçtiğimiz otomatik sayı değerini elde ettik. Geriye bu değeri değiştirmek kaldı. Oraya ne yazılırsa otomatik sayı yeni değerini bu numaradan başlatıyordu. Burayı 0 olarak değiştirdikten sonra resimdeki gibi bir görüntü normaldi artık.

ID	ad	departman
2	tekin	teknik
3	idil	idari
5	pelin	pazarlama
6	inci	insan kaynakları
9	serhat	servis
12	bilgen	bilgi işlem
15	murat	muhasebe
1	deneme	

Otomatik sayı değeri artık 1'den başlıyor.

Eğer otomatik sayı alanı birincil anahtar (primary key) olarak belirlenmişse, sonraki kayıt girişlerinde çoğaltılmaması gereken dizin değeri hatası (duplicate unique index) ile karşılaşmak kaçınılmaz olur.

Siz en iyisi bu değeri arkaplan işlemleri için kullanın ya da kullanıcıya herhangi bir kod değeriymiş gibi gösterin. Ben olsam ekran başındaki kişiyi resimde olduğu gibi kandırırdım.

Kod	Ad
Kod 1	Ali
Kod 2	Veli
Kod 3	Kırkdokuz
Kod 4	Elli

Nasıl mı oldu diyorsanız, tablonun tasarım görünümünde otomatik sayı alanının Format (Biçim) özelliğinin değerini tam olarak

“Kod “#

şeklinde belirtmeniz yeterli.

DEĞİŞİKLİKLER KAYDEDİLSİN Mİ

Diğer birçok programda olduğu gibi bilgilerinizin kaydedilmesinden önce uyarı mesajı göstermek istiyorsanız alttaki kodu deneyebilirsiniz. Herhangi bir formun güncelleştirme öncesi (BeforeUpdate) ile ilgili yordamının aşağıdaki gibi olmasını sağlayın. Artık eski kayıtlar üzerinde değişiklik yapıldığında soru sorulacak ve kullanıcı eğer hata yaptığını düşünüyorsa kaydı bu şekilde geri alabilecek. Formun BeforeUpdate yordamını oluşturun ve aynen şu şekilde olmasını sağlayın:

Private Sub Form_BeforeUpdate(Cancel As Integer)

If NewRecord = False Then

If MsgBox("Değişiklikler kaydedilsin mi?", _
vbYesNo) = vbNo Then

Cancel = True

DoCmd.RunCommand acCmdUndo

End If

End If

End Sub

FORMU TASARIM GÖRÜNÜMÜNDE AÇ

Formu, raporu, tabloyu, sorguyu ve diğer nesneleri tasarım görünümünde açmak için CTRL tuşunu basılı tutarak açmayı deneyin. CTRL+ENTER yapın ya da CTRL basılı iken fareyle çift tıklatın.

SQL SERVER TABLOSUNDA YENİ GİRİLEN KAYITLARA ULAŞMAK

ADO kullanırken VBA ya da ASP kodlarında SQL Server veritabanındaki bir tabloya yeni kayıt eklendiğinde bu kayda ulaşmak sorun olabiliyor. Aslında sorun birincil anahtar dizini (primary key) kullanılmamış olmasından kaynaklanıyor.

Tablo1.Addnew

Tablo1("Alan1")="Değer"

...

Tablo1.Update

'MsgBox Tablo1("Alan1") 'Access için.

Response.Write Tablo1("Alan1") 'ASP için.

Eğer tablonuzda herhangi bir birincil anahtar dizini yoksa bu kod son satırda hata verecektir.

HERKES İÇİN SORGULAR

"IN" DEYİMİ İLE ÇOKLU ARAMA KRİTERİ BELİRLEMEK

Sorgularda aynı alan için birçok arama kriteri "OR" deyimi ile tanımlanabiliyor. Tabii bununla bazan kod yeterince uzayabiliyor ve karmaşık bir hal alıyor. Onun yerine "IN" komutunu kullanmak daha kolay.

Kişiler tablosundaki kayıtlardan Meslek alanının değerinin 'Öğretmen', 'Bankacı', 'Doktor', 'Muhasebeci' olduğu kayıtları bulmak için OR komutunu şöyle kullanabilirsiniz:

SELECT * FROM Kisiler WHERE ((Meslek='Öğretmen') OR (Meslek='Bankacı') OR

(Meslek='Doktor') OR (Meslek= 'Muhasebeci'))

Aynı kod IN deyiimi ile bakın nasıl oluyor:

SELECT * FROM Kisiler WHERE Meslek IN ('Öğretmen', 'Bankacı', 'Doktor', 'Muhasebeci')

MESAJ KUTUSUNDA YENİ SATIR GÖSTERMEK

Kullanıcıya uzun uzadıya olan mesajları daha anlaşılır bir şekilde göstermek istiyorsanız mesaj kutularında yeni satırlar kullanmayı deneyebilirsiniz.

**MsgBox "Bu mesaj" & vbCr & "birkaç " & vbCr & _
"satırdan" & vbCr & "oluşuyor.", vbOKOnly + vbExclamation**

VbCr deyiimi satır sonu karakteridir. Yani Chr(13) ile aynı.



GÖRÜNEN KAYDI RAPORDA AÇ

Form üzerinde birçok kayıt olduğu halde sadece görünen kayda ilişkin bir rapor nasıl yazdırılır? Bunun için raporu açarken kullandığınız koda bir parametre eklemeniz yeterli.

DoCmd.OpenReport "Rapor1", asViewPreview, , "SiraNo=3"

gibi. Eğer bir metin alanına göre koşul belirlemek istiyorsanız değerin başında ve sonunda ' işareti olmalı.

DoCmd.OpenReport "Rapor1", asViewPreview, , "PersonelNo='Ahmet0001'"

Sondaki parametre aslında bir arama koşulu. Oraya istediğiniz kadar koşul içeren bir metin yazmanız mümkün.

DoCmd.OpenReport "Rapor1", asViewPreview, , "(SiraNo>100) AND (PersonelAdi>'K') AND (Yas<25)"

En son parametrede verilen süzme kriteri formları açarken de aynı şekilde kullanılabilir.

ACCESS PENCERESİNİ BOYUTLANDIRMAK

Özenle hazırladığımız form ve raporları tam ekran yapar, kullanıcının önüne koyarız. Fakat programın kullanıldığı bilgisayarda Access penceresi tam ekran ya da bizim istediğimiz boyutta açılmıyor her zaman. Lakin bunun da çözümü yok değil. Access penceresinde modül sayfasına gelip yeni bir modül

oluşturun ve içinin tam olarak aşağıdaki gibi olmasını sağlayın. Sonra bu modülü “ModulYerlestir” adıyla kaydedin.

```
Option Compare Database
Option Explicit

Public Type RECT
Left As Long
Top As Long
Right As Long
Bottom As Long
End Type

Public Const SW_RESTORE = 9
Public Declare Function ShowWindow Lib "user32" (ByVal hwnd As Long, ByVal nCmdShow As Long) As Long
Public Declare Function MoveWindow Lib "user32" (ByVal hwnd As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal bRepaint As Long) As Long

Public Const SPI_GETWORKAREA = 48
Public Declare Function SystemParametersInfo Lib "user32" Alias "SystemParametersInfoA" (ByVal uAction As Long, ByVal uParam As Long, ByRef lpvParam As Any, ByVal fuWinIni As Long) As Long

Public Sub MasaustuneYerlestir()
'Kodu şimdi çalıştırmak için buraya gelip F5 tuşuna basın.
Dim r As RECT
SystemParametersInfo SPI_GETWORKAREA, 0, r, 0
Call Yerlestir(r.Left, r.Top, r.Right - r.Left, r.Bottom - r.Top)
End Sub

Public Sub Yerlestir(Sol As Long, Ust As Long, Sag As Long, Alt As Long)
ShowWindow hwndAccessApp, SW_RESTORE
MoveWindow hwndAccessApp, Sol, Ust, Sag, Alt, 1
End Sub
```

Artık programınızın herhangi bir yerinde

Call MasaustuneYerlestir

ya da

Call Yerlestir(100, 100, 200, 200)

gibi komutları kolaylıkla kullanabileceksiniz.

METİN DOSYALARINI KOLAY BİRLEŞTİRMEK

Metin dosyalarını kolay yoldan birleştirmek için MS-DOS Komut İstemi’ndeki (Command Prompt) COPY komutu kullanılabilir.

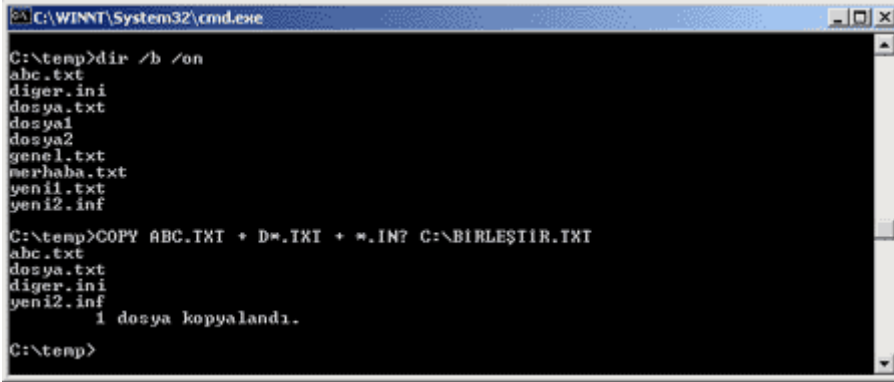
MS-DOS Komut İstemi’ni açıp şöyle bir komut yazın:

copy *.txt c:\birlestir.txt

Bu komut o klasördeki tüm TXT uzantılı dosyaları birleştirecek ve C:\ üzerinde BİRLEŞTİR.TXT adlı bir dosya oluşturarak içine bunları yazacak.

Ya da şunu deneyin:

copy abc.txt + d*.txt + *.in? c:\birlestir.txt



```
C:\WINNT\System32\cmd.exe
C:\temp>dir /b /on
abc.txt
diger.ini
dosya.txt
dosya1
dosya2
genel.txt
merhaba.txt
yeni1.txt
yeni2.inf

C:\temp>COPY ABC.TXT + D*.TXT + *.IN? C:\BIRLESTIR.TXT
abc.txt
dosya.txt
diger.ini
yeni2.inf
1 dosya kopyalandi.

C:\temp>
```

KLASÖRÜ WINDOWS GEZGİNİNDE AÇ

Bazan farenin o yana bu yana gidememesinden şikayetçiyseniz herhalde klavye kullanmaya da alışkınsınızdır. Klavyeyi ve klasör penceresi yerine Windows Gezgini'ni kullanmayı tercih eden biri iseniz klasörü açarken Shift tuşunu basılı tutmak işinize yarayacaktır.

Bahsedilen konularla ilgili kodların ve örneklerin tümünün bulunduğu dosyayı PC Magazine Online sitesinde Adı Geçen Dosyalar bölümünde bulabilirsiniz.

ACCESS HİKAYELERİ

ESKİ DEFTERLER YENİ HİKAYELER

Access2000 projelerinin dağıtılması için Office 2000 Developer Tools v1.0'ı aldık. Fakat aldığımız gün MSDN'de bu paketin v1.5 sürümünün çıktığını öğrendik. Elde bu paket varken MSDN'e bakma işi nereden çıkmıştı? Tabii hala normal bir kurulum hazırlayamadığımız için. Öyle ki, kurulum hazırlamaya çalışırken Access Run-Time için 500MB'lık bir klasör gereksiz yere kurulum paketine dahil ediliyor. Tabii bu, sadece Access Run-Time paketini eklerken oluyor. Bu paket eklenmezse proje dosyasıyla beraber birkaç dosya paketleniyor sadece. Boyu da 3 MB. civarında. Gerçi klasörün içinden gereksiz gördüğüm bazı dosyaları sildiğimde sorun çıkarmadı ama buna güvenmemek daha iyi.

Böyle durumlarda MSDN'nin sitesinden yardım almak gerekli. Orada anlatıldığına göre normalde En az 150 mb.'lik bir paket oluşturuluyormuş. Tabii hazırladığımız pakete Access Run-Time eklemek istediğimizde IE5'in kurulumunu da eşantıyon olarak ekliyor. Sonradan olacak ki bunun abartı olduğunun farkına varmışlar ve IE5'i pakete dahil etmeyen bir yama çıkarmışlar. Yama demek biraz zor gerçi. 10 mb. civarında bir büyüklüğü var ilk yükseltme programının. v1.5 sürümü ise 30 mb. civarında.

Bu olaydan da öğrendim ki, yeni bir ürün aldığımızda öncelikle Internet'ten o ürünle ilgili yeni güncelleştirmelerin olup olmadığını araştırmak gerekiyor.

* * *

Access kullanan bazı makinelerin SQL Server'ın olduğu bilgisayara isim ile ulaşmaları bazan sorun oluyor. Bu yüzden HOSTS dosyasında bir IP adresi ile bir bilgisayar adı tanımlamak gerekiyor. Böylelikle orada yazılı olan ad bizi her zaman yanındaki IP adresine yönlendiriyor.

Bu aşamadan sonra aynı programın her bir makineye kurulup ODBC ayarlarının yapılması gerekiyor. Zira Access içinden SQL'e ODBC ile bağlanılıyor. Tablolar kısmına SQL Server veritabanındaki tablolar yerleştirildi. Böylelikle aynı ayarın, programın kurulacağı tüm makinelere yapılması şart oluyor. Bunu da kod ile yapmak çok masraflı değil.

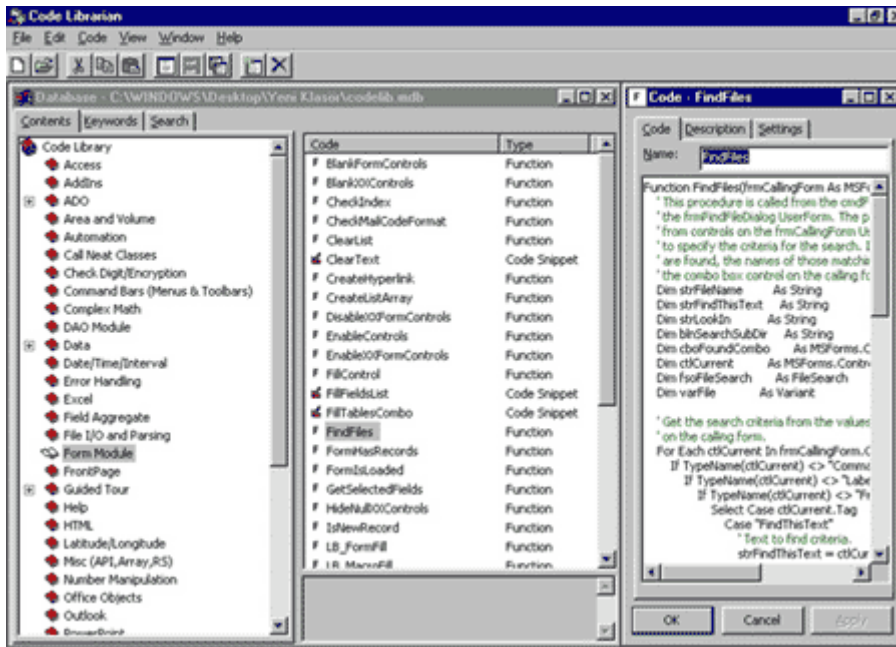
Yeni bir ODBC DSN bileşenini kod ile tanımlamak için Registry'deki HKLM\Software\ODBC altında bazı değişiklikler ve ODBC.INI dosyasına eklemeler yapmak yeterli.

HOSTS dosyasını da otomatik ayarlamak için önce dosyanın yerini bulmak lazım. Eğer sistem Win9x ise %WINDIR% klasörü altında HOSTS.SAM dosyası, eğer NT ya da Win2000 ise %SYSDIR%\DRIVER\ETC klasöründeki HOSTS dosyası değiştirilmeli.

Artık DSN ayarlarının da kurulum programları tarafından otomatik olarak yapılması fikri kulağa hoş geliyor aslında.

* * *

ODE Tools'un içinden çıkan bir başka eşantıyon da Code Librarian adlı yardımcı bir program. Bu program, kod yazarken büyük projeler arasında bir o yana bir bu yana koşuşturan programcılar için bir kod kütüphanesi oluşturmuş. Herhangi bir konuyla ilgili kod ihtiyacınız olduğunda gerekli başlığın altından aradığınız kodu kolayca bulabiliyorsunuz. Üzerine istediğiniz eklemeleri yapabileceğiniz gibi, var olan kodları da değiştirmeniz mümkün.



Eğer kodlarınız için kütüphane oluşturmayı düşünüyorsanız bu program birebir. Bu kadar hazır kod el altındayken insana kopya çekme fırsatı da doğmuyor değil açıkçası.

Tabii ki, PC Magazine okuyucuları ilk kez yayınlanan orijinal ipuçlarını bu sayfalarda görmeye devam edecekler.

DOSYA:

[AccessOcak2001.zip](#)

(Otomatik sayıların değiştirilmesi ile ilgili sonradan eklenen bir yazıya [buradan](#) ulaşabilirsiniz.)

İpuçlarına Devam

YORDAMLAR VE GERÇEKLER

Yordamlar “Function” ve “Sub” türünde olabilir ve herbirinin mutlaka adı vardır. Önce şu örnek yordam başlıklarına göz atın:

- Sub Sub1()
- Function işlem1() As Date
- Function işlem1()
- Function işlem1(Parametre) as Integer
- Sub Sub1(parametre1 As String, parametre2 As Date)
- Public Function F(Ad As String, ParamArray Dizi()) As Long
- Private Static Function StatikBirİşlev() As Boolean
- Sub SubYeni(Ad As String, Optional Tarih As Date=36861)
- Private Sub Command1_Click()

“Sub” yordamları mutlaka “End Sub” satırı ile, “Function” yordamları da “End Function” satırı ile biter.

Private Sub MetinKutusu1_Click()

...

End Sub

ve

Function Degistir() as Boolean

...

End Function

gibi.

Örneklerde “Function” ve “Sub” deyimlerinden önce bazı tanımlamalar var. Bunlar yordamlara bazı ek özellikler katarlar. Mesela başında “Private” deyimi olan bir yordam, aktif olan modül sayfasının dışında bir yerde kullanılamaz. Ya da, başında Static olan bir yordamın içindeki tüm değişkenler statik olarak tanımlanmış olur.

Function yordamlarının asıl amaçlarından biri değer üretmek olduğu için başlık satırının sonunda mutlaka “AS” deyimi ile bir tip belirtilir. Eğer tipi belirtilmemiş bir işlem varsa bunun döndüreceği değer Variant (herhangi bir değişken tipinde olabilen) türünde olur.

Bir de parametresiz işlevler vardır ki, tipi belirtilmiş olsa bile, parametresiz bir işlevi isterseniz “Sub” yordamlar gibi kullanabileceğinizi geçen aylarda da söylemiştik netekim.

Örneğin,

işlev1() as String

şeklinde tanımlı bir işlevi;

Call işlev1

ya da doğrudan

işlev1

şeklinde çalıştırmak mümkündür.

Herhangi bir yerden bulduğunuz bir kod parçasını programınıza birkaç şekilde ekleme imkanınız var. Şu kod parçası aktif dosyamızın bulunduğu yerdeki tüm dosyaları listeleyip mesaj kutusunda gösteriyor:

```
Dim Dosyalar As String, Dosya As String, Klasor as String
Klasor = CurrentProject.Path
If Right(Klasor,1)<>"\" Then Klasor= Klasor & "\"
Dosya = Dir(Klasor)
Dosyalar = "Bu klasördeki dosyalar:" & vbCrLf
Dim i As Integer
i = 0
Do While Dosya <> ""
If Dosya <> "." And Dosya <> ".." Then
i = i + 1
Dosyalar = Dosyalar & vbCrLf & i & ". " & Dosya
End If
Dosya = Dir
Loop
MsgBox Dosyalar
```

Bunu, var olan bir yordamın içine ekleyebilir ya da kendi oluşturacağınız yeni bir yordamın içine koyabilirsiniz. Mesela bir düğmenin “Click” (Tıklatıldığında) yordamı kısmına. Eğer elinize geçen bir kod parçasında “Sub” ya da “Function” şeklinde yordam başlıkları varsa bunu kendi yordamınıza eklemek için başlıklarını ve son satırlardaki “End..” kısımlarını çıkarmalısınız. Veyahut, yordamı aynen kullanmalısınız.

DURUM ÇUBUĞUNDA MESAJ GÖSTERMEK

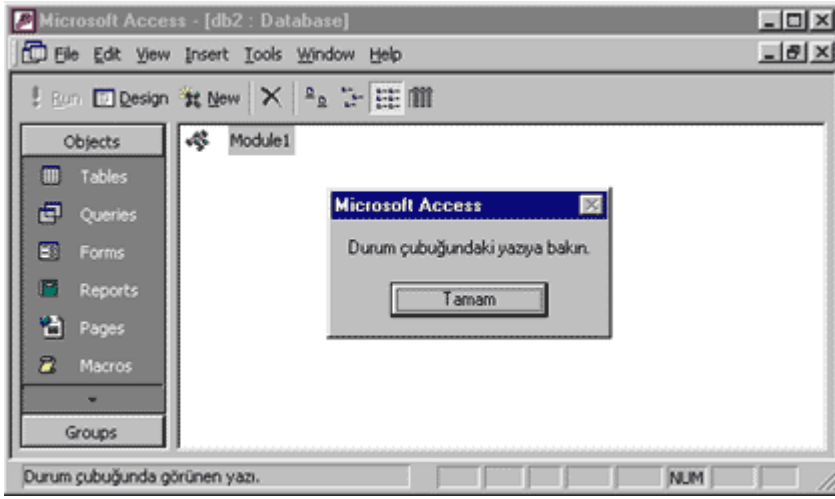
Uygulamalarımızda diğer profesyonel programlar gibi durum çubuğunu kullanmak ne iyi olurdu diyorsanız şu kısacık kodu deneyin:

Application.Echo True, "Durum çubuğunda görünen yazı."

MsgBox "Durum çubuğundaki yazıya bakın."

Application.Echo True

Bunu çalıştırınca Access’in durum çubuğunda (Status Bar) bir mesaj görüntülenecek. Kullanıcının bekletileceği yerlerde “İşlem devam ediyor...” gibi bir mesajı gösterebilirsiniz bu şekilde. İlk satırdaki “True” yerine “False” yazarsanız, Access penceresi silik görünür. Böyle bir durumda, işlem sonunda Echo’yu True ile çalıştırmayı ihmal etmeyin, yoksa işlem yapılabilsede, Access ekranında pasif bir pencere görüntüsünden başka bir görüntü olmayacaktır.



DoCmd nesnesinin Echo adlı metodu da aynı işi yapabiliyor. Fakat Application nesnesininkini kullanmamız öneriliyor.

DAHA KOLAY KAYIT SİLME

Geçen aylarda bu sayfalardan, kayıt silmek için ‘daha güzel kayıt silme’ başlığında bir kod örneği verdik. Gelgelelim aşağıdaki yeni işlevle daha kolay ve kısa bir şekilde kayıt silme işlemi yapabilirsiniz.

```
On Error Resume Next  
If Me.NewRecord Then  
Me.Undo  
ElseIf MsgBox("Kayıt silinsin mi?", vbYesNo) = vbYes Then  
Me.Recordset.Delete  
End If
```

Bu kodu, kayıt silme görevi üstlenecek bir düğmenin “Click” (Tıklatıldığında) yordamındaki “Sub..” ile “End Sub” satırları arasında kalan kısma yazın.

ON ERROR GOTO 0

Hata gidermek için birçok yerde “ON ERROR RESUME NEXT” deyimi kullanılıyor. Hata giderme pozisyonundan ayrılmak gerektiği durumlarda “ON ERROR GOTO 0” (sıfır ile bitiyor) deyimini kullanmak gerekiyor

```
On Error Resume Next  
‘Hataları yoksayıp sonraki komuta geç.  
MsgBox 3 / 0  
MsgBox "Sıfıra bölme hatasını geçti."  
On Error GoTo 0  
‘Hataları yoksayma.  
MsgBox 3 / 0  
‘Şimdi sıfıra bölme hatası oluşunca uyardı.
```

AH ŞU KISAYOL TUŞLARI!

Sonraki Kayıt Ctrl + Page Down
Önceki Kayıt Ctrl + Page Up
İlk Kayıt Ctrl + (Yukarı Ok)
Son Kayıt Ctrl + (Aşağı Ok)
Yeni Kayıt Ctrl + +
Kayıt Sil Ctrl + -
Önceki Kaydı Çoğalt Ctrl + '

Bu bilgilere, yardım bölümünde “keyboard shortcuts” başlığını aratarak da ulaşabilirsiniz.

FORM AÇIK MI?

Bir formun ya da raporun açık olup olmadığını bulmak için aşağıdaki kodları kullanabilirsiniz.

```
If SysCmd(acSysCmdGetObjectState, acForm, "Form1") <> 0 Then  
MsgBox "Form açık."  
Else  
MsgBox "Form kapalı."  
End If
```

Örnek, Form1 adlı formun açık olup olmadığını araştırıyor. Raporlar için “acForm” deyimini “acReport” olarak değiştirin.

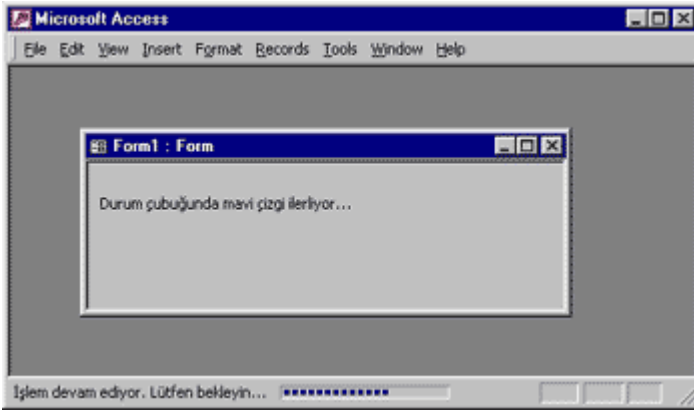
DURUM ÇUBUĞUNDA İLERLEME ÇUBUĞU GÖSTERMEK

Access kendi işlemlerini yaparken bazan durum çubuğunda ilerleyen mavi çizgi beliriyor. Bundan, işlemin ne kadarının bittiğini anlayabiliyoruz. Tabii, bunun aynısını bizim kullanabilmemize de imkan var. Tablo1’deki kodu çalıştırırsanız resimdeki gibi mavi ilerleme çubuğu çıkacak.

TABLO1:

```
Sub Ilerleme_Cubugu()  
  
Dim T As Long  
T = Timer  
  
SysCmd acSysCmdInitMeter, "İşlem devam ediyor. Lütfen bekleyin...", 10  
'İlerleme çubuğunu başlat. 10 sayısı ile değer sınırı veriliyor.  
  
Do  
If Timer - T >= 10 Then Exit Do  
SysCmd acSysCmdUpdateMeter, Timer - T  
'10 saniye boyunca her saniye için çubuğu 1 nokta ileri al.  
DoEvents  
'Ekranı dondurma  
Loop  
  
MsgBox "İşlem tamamlandı."  
  
SysCmd acSysCmdClearStatus  
'İlerleme çubuğunu kaldır.  
  
End Sub
```

acSysCmdInitMeter ile ilerleme başlatılıyor ve çubuğu ilerletmek için acSysCmdUpdateMeter kullanılıyor. İşlem bitiminde acSysCmdClearStatus ile çubuğu kaldırmalısınız.



Mavi çizgi ilerliyor...

PENCERELERİ BOYUTLANDIRMAK

Access Visual Basic'teki gibi formları kolayca boyutlandırıp taşımamıza izin vermiyor. Bunun için MoveSize komutunu kullanmak gerekiyor. MoveSize ile bir formu belli bir noktaya taşıyabilir ya da taşırken boyutlarını yeniden ayarlayabilirsiniz.

Şu komut aktif pencereyi sol üst köşeye taşır:

DoCmd.MoveSize 0, 0

Bu ise sol üste taşırken genişliğini 4000, yüksekliğini 2000 olarak ayarlar:

DoCmd.MoveSize 0, 0, 4000, 2000

Tüm parametreleri belirtmek zorunda değilsiniz:

DoCmd.MoveSize , 1000, , 1500

Bu kod sol ve genişlik değerlerini değıştirmeden, üst tarafı 1000 ve genişliği 1500 olarak ayarlayacak.

Parametreler aşağıdaki şekilde kullanılıyor:

DoCmd.MoveSize Sol, Üst, Genişlik, Yükseklik

PAROLA KAYITLARINI GİZLİ GÖSTERMEK

Access'teki form ve raporlardaki metin kutularında şifreli metin kullanabilmek için InputMask değerini "Password" yapmak yeterli oluyordu. Eğer kayıt tablosunda da parola bilgilerini (*) işaretili olarak görmek istiyorsanız tahmin ettiğiniz gibi veri alanının InputMask (Giriş Maskesi) değerini aynı şekilde ayarlamalısınız.

Personel : Table				
	Kişi	Departman	Parola	Dahili
	Orhan	Satış	*****	131
	Ayşegül	Satış	*****	132
	Mete	Pazarlama	*****	246
	Nilgün	Muhasebe	****	354
	Oktay	Teknik	*****	502
	Murat	Teknik	*****	503
	Canan	İdari	*****	490
	Selim	İdari	*****	493

Record: 14 9 of 9

Tabloyu tasarım görünümünde açıp parola alanının InputMask değerine "Password" yazın.

Field Name	Data Type	Description
Kisi	Text	
Departman	Text	
Parola	Text	
Dahili	Text	

Field Properties	
General	Lookup
Field Size	50
Format	
Input Mask	Password
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes

A pattern for all data to be entered in this field

BAĞLANTIYI YENİ PENCEREDE AÇ

Internet Explorer’da bağlantıyı yeni pencerede açmak için sağ fare tuşu menüsü kullanılabilir. Bunu el ile yapmak için bağlantıyı tıklarırken Shift tuşunu basılı tutmayı deneyin.

İlk Add-in'imiz

ADD-IN OLUŞTURMAK

1) Hazırlamış olduğunuz Access programınızın içinde aşağıdaki gibi uSysRegInfo adlı bir tablo oluşturun ve listedeki 4 kaydı tabloya girin.

2) Tabii öncelikle Add-in’imizin bir adı olması gerektiği için bizimkinin adını “Edinim” koydum.

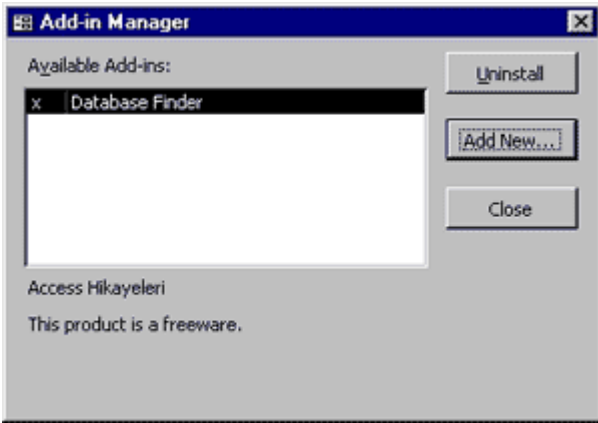
Subkey	Type	ValName	Value
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\9.0\Access\Menu Add-Ins\Edinim	0		
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\9.0\Access\Menu Add-Ins\Edinim	1	Library	C:\Program Files\Microsoft Office\Office\Edinim.mda
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\9.0\Access\Menu Add-Ins\Edinim	1	Expression	=Edinim_Entry()
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\9.0\Access\Menu Add-Ins\Edinim	4	Quicktip	108409

3) Type adlı alan sayı tipinde, diğer alanlar ise 255 uzunluklu metin alanları olmalı ve eğer Access2000 değil de Access97 kullanıyorsanız Subkey alanındaki değerlerde “9.0”ı değiştirip “8.0” yapmalısınız.

4) uSysRegInfo tablosundaki “=Edinim_Entry()” değeri bu add-in’in başlatılırken çağrılacağı işlevin adıdır. (Yani add-in başlarken ana formu açmak istiyorsanız, herhangi bir modülün içinde “Edinim_Entry” adlı parametresiz bir işlev (function) oluşturun ve kod kısmına ana formun açılmasını sağlayacak kodu yazın. Bu, add-in’in başlatılacağı yerdir.)

5) Kodlarımızın gizlenmesi ve programımızın bir EXE gibi hızlı çalışabilmesi için bu dosyayı önce MDE haline getirmeliyiz ve sonrasında dosyanın uzantısını MDA yapmalıyız. (Laf aramızda, MDB dosyalarının uzantısını MDA uzantılı yapmak da kâfi.)

6) Dosyayı uSysRegInfo tablosunda belirttiğiniz klasörün içine kopyalayın ve Access’i açıp Add-in yükleme kısmından bu dosyayı yükleyin. (Tools/Add-ins/Add-in Manager ile yapıyor. Tahmin edeceğiniz gibi bu da Access’te yapılmış bir add-in.)



Add-in yüklemek için Tools/Add-ins/Add-in Manager menüsünü çalıştırın.

Add-in’lerin amacı Access programcısına kolaylık sağlamak.

Önceki bilgilerimizi yoklayarak CodeDB ile CurrentDB’nin ne işe yaradıklarını hatırlayalım. Ek olarak fazla bir şey yapmaya gerek yok. Add-in ile normal bir Access programı ile yaptığımız her türlü işlemi yapabilirsiniz. Ama asıl amacı üzerinde çalışılan projeye yardımcı olmak. Örneğin geçen ay bahsettiğimiz Database Finder adlı ilk add-in’imiz, Access’te çalışırken diğer yandan bilgisayardaki tüm Access dosyalarını tarayarak aradığımız isimdeki nesneleri (tablo, form, sorgu, makro vs.) arkaplanda bulup listeliyordu. Aynı şekilde Access97’deki “SwitchBoard Manager” add-in’i Access içinden otomatik menü formları yapmamızı sağlıyordu.

İP ÜSTÜNE UN SERMİŞ

Bu ay ne ettiysem Hoca bana ipucu vermedi. Neymiş, ipin ucuna un sermiş. İş başa düşünce Bay Akses’inkileri aratmayacak ipuçları keşfettim, laf aramızda.

İKİ FORMUN KAYIT KÜMESİNİ BAĞLAMAK

Diyelim ki bir adet liste formunuz ve bir adet bu listeden seçerek ayrıntısını göstermek için kullanmak istediğiniz ayrıntılı bilgi formunuz var. Haliyle bu iki form, kayıtlarını aynı tablodan almak zorunda.

Tek fark birinin liste halinde, diğerinin sayfa halinde göstermesi. Bu durumda kod yazıp “FindFirst” gibi bir komutla ayrıntı tablosundaki kayda ulaşmak gerekiyor. Fakat şu yolu denerseniz, öyle aramayla, bulmayla neyin uğraşmanıza gerek kalmayacak. Kisiler adlı formda bir kayıt seçip detayını görmek için Kisiler_Detay adlı formu açmak istediğinizi varsayarsak şu kodu bir defalık çalıştırmanız yeterli:

```
DoCmd.OpenForm "Kisiler_Detay", , , , acHidden
Set Forms("Kisiler_Detay").Recordset = Forms("Kisiler").Recordset
Forms("Kisiler_Detay").Visible = True
```

ad	soyad	sehir
ahmet	duman	istanbul
ayşe	kara	ankara
ali	beyaz	sivas
mehmet ali	yemişçi	adana
steve	soul	michigan
sibel		

Record: 14 of 6

Kişiyi ait bilgiler

ID: 6

Ad: sibel

Soyad:

Sehir:

Adres: yenice mah. beyaz

Dogum Tarihi:

Meslek:

Kayıt kümeleri birbirine bağlanmış iki formun ahenkli uyumu.

Her ne olursa olsun Recordset sınıflarının adresleri eşit olduğu sürece zaten açık olan kaydı bir daha aramaya gerek kalmıyor.

LDB DOSYASI İLE KULLANICILAR LİSTESİ

Access dosyalarının olduğu yerde aynı önadı taşıyan LDB uzantılı dosyayı hepimiz farketmişsinizdir. Birçoğumuz acaba Access dosyamıza başka kimler bağlı diye bulmak istemiştik en az bir kere. Velhasıl, bizim aradığımız şey tam tamına LDB dosyasının kendisidir. Çalışma anında bunun kopyasını alarak içeriğini okumayı deneyin ve bu dosyayı salt-okunur açmayan tüm bilgisayar adlarını ve kullanıcı adlarını orada görün.

TRUE -1 FALSE 0

Elinizde 10 adet boolean tipinde değişken var. Bunlardan kaçının TRUE olduğunu nasıl bulursunuz? Siz hiç düşünmeden bunların değerlerini toplamanızı öneriyorum.

Dim a as boolean, b as boolean f as boolean

...

msgbox ((a + b + c + d + e + f) * -1) & " adedi TRUE yapılmış."

Zira TRUE değerinin VBA'daki sayısal karşılığı -1, FALSE değerinininki ise 0'dır. Bu haliyle TRUE olanları toplayıp -1 ile çarpmak yeterli oluyor. Eşit mi, değil mi sorularının çok olduğu kodlarda işinize yarayabilir.

Mesela 10 adet seçim kutusundan en az 3 tanesinin seçilmesi gerektiğini karşımızdaki kişiye şöyle anlatabiliriz:

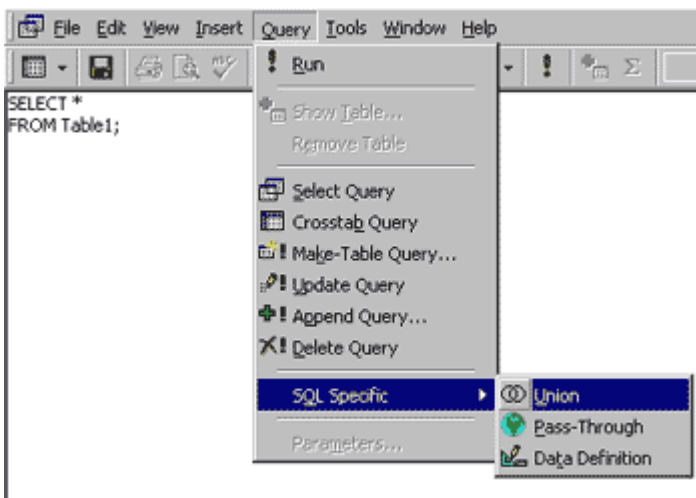
**If (Check1 + Check2 + ... + Check10) > -3 Then _
MsgBox "En az üç seçenek işaretlemelisiniz."**

Bir de şunu deneyin:

**If (IsNull(Text1) + IsNull(Text2) ... + IsNull(Text10)) = -10 Then _
MsgBox "En az bir değer girmelisiniz."**

HERKES İÇİN SORGULAR - UNION QUERY

Bu sorgu türü ile birbirinin aynı özelliklere sahip ya da birbirine benzer olan tabloları birleştirerek sorgu yapıp tek bir kayıt kümesi elde edebiliyorsunuz. Yeni bir sorgu oluşturup Sorgu/SQL Spesifik menüsünün altındaki menüden en üsttekini seçin.



Sözelimi aynı alanları içeren Adresler1 ve Adresler2 adlı iki tablonuz var. Ama siz herhangi bir kişinin adresini bulmak için ikisinden aynı anda arama yapmak istiyorsunuz. Şu kodu yazın, olsun bitsin:

```
SELECT Ad, Soyad  
FROM  
(  
SELECT * FROM Adresler1  
UNION  
SELECT * FROM Adresler2  
) as Adresler  
WHERE Ad Like "*abc*"  
ORDER BY Ad, Soyad;
```

Sonrasında bu sorguyu tasarım moduna getirip üstünde çalışmak çok zevkli oluyor.

Windows 2000'den sonra sonra Türkçe karakterler konusunda olayların ters dönmeye başladığına şahit oluyorum. Artık yeni programlarda “i” ile “İ” ayrılıyor ve Türkçe’de nasılsa öyle oluyor. Yani “ID” adlı bir alanınız olduğunda “id” yazdığınızda büyük ihtimalle hata oluşacak, şu an benimkinde olduğu gibi.

BAŞKA BİR DOSYADAKİ TABLOYU SORGULAMAK

Ufak bir iş için bir tabloyu içeri bağlamayı içinize sindiremiyorsanız sizin için de ufaklık bir çözümümüz var. Başka bir Access dosyasının içindeki bir tabloyu sorgu içinde kullanmak için şu iki örneği inceleyin:

```
SELECT Kisiler.*  
FROM [c:\belgelerim\db1.mdb].Kisiler;
```

```
SELECT Kisiler.*  
FROM Kisiler IN 'c:\belgelerim\db1.mdb';
```

Birincisinde sorgunun tasarım anında tablo özelliklerinde dosya ismi seçilebiliyor. Bunu yapınca tasarım görünümünde tablonun özelliklerinde Source (Kaynak) değerinin, yazdığımız dosyanın adı olduğunu göreceksiniz. Sorgu içinde tablo seçili iken View/Properties (Görünüm/Özellikler) menüsü ile özellikler sayfasını açabilirsiniz. İkinci örnek ise eskiden beri kullanılagelen bir “IN” kalıbı.

BAŞKA UYGULAMALARI YÖNETMEK

Söz başka dosyalardan açılmışken başka bir Access uygulamasının nasıl yönetildiğine bakınız şimdi.

```
Option Compare Database  
Option Explicit  
Dim a As Access.Application  
Sub deneme()  
Set a = Nothing  
Set a = New Access.Application  
a.OpenCurrentDatabase "c:\belgelerim\db1.mdb"  
a.DoCmd.OpenForm "Form1"  
a.DoCmd.Maximize  
a.Visible = True  
End Sub
```

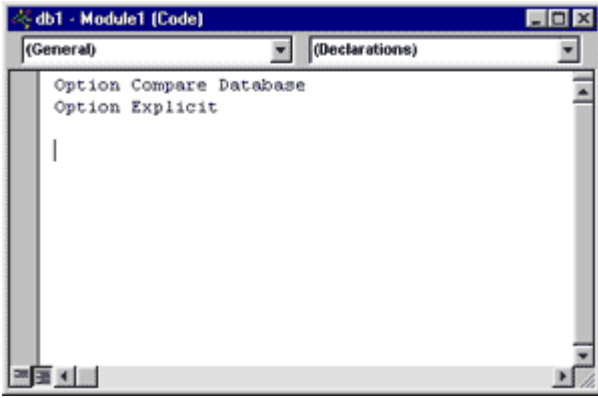
Yeni bir Access uygulaması açıp onda bir dosya açarak kendi dosyamızda yapabileceğimiz herşeyi o uygulamada da yapabiliriz.

Kod Yazmaya Geçiş

Access modüllerinde kullandığımız dil VBA, yani Visual Basic for Applications. Office’in hemen her yerinde kullanılabiliyor. Hatta başka uygulamalarda da kısmen yer edinmiş görünüyor. AutoCad gibi bazı programlarda VBA kullanılabiliyor örneğin.

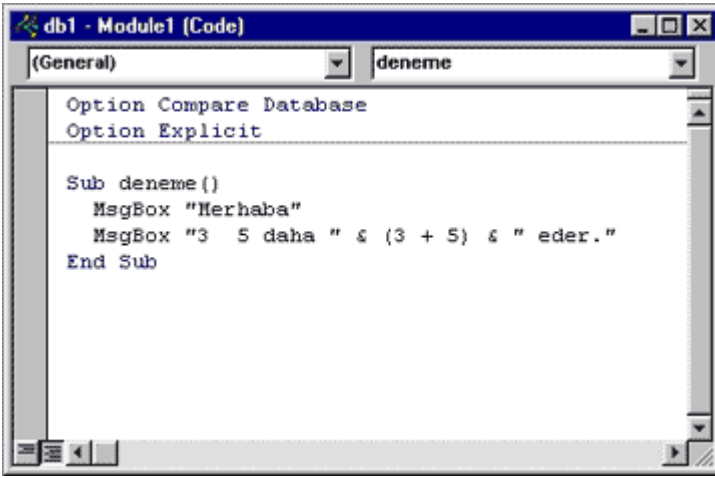
KOD DENEMEK İÇİN EN KISA YOL

Veritabanı ana penceresinde iken (Database Window) modül (Modules) sayfasına geçip yeni (New) bir modül oluşturalım. (Resim – db1 Module1 code)



Yeni bir modül penceresi.

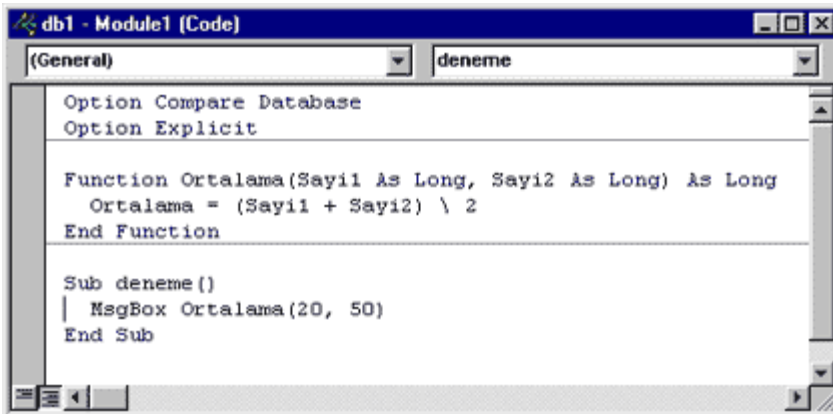
Herhangi bir kodu çalıştırmak için mutlaka “Sub” ile başlayan bir yordam oluşturmalıyız.



“Sub” ile “End Sub”un arasında herhangi bir yerde F5’e basın.

“Sub Deneme” yordamını aynen yazıp “Sub” ile “End Sub” arasında herhangi bir yerde F5 tuşuna basarsanız program çalışır ve iki kez mesaj gösterir. Bunun gibi End Sub’dan sonra yeni yordamlar ekleyebilirsiniz.

Ek olarak İşlev (Function) tanımından bahsetmek gerekiyor. Bu da “Sub” yerine “Function” yazılarak kullanılıyor ve “End Function” ile bitiyor. Tabii önemli bir farkı var. İşlevlerin asıl amacı geriye değer döndürmektir. Yani bir işlev çağrılırken bir şeye eşitlenir (istisna olabilir).



“Ortalama” adlı işlev örneğine bakın şimdi. Orada işlevi tanımladıktan sonra artık herhangi bir yerden bunu çağırıp kullanabiliyoruz. İsterseniz bunu sorgu yaparken SQL metninizin içine de yazabilirsiniz.

İşlevler bir şart dışında, tek başlarına çalışmazlar. Eğer işlevlerin parametreleri yoksa bunlar “Sub” yordamları gibi çalışırlar.

Parametreler “Function” ya da “Sub” tanımlama başlığındaki parantezler arasında bulunan değerlerdir. İki parantezin arası boş ise parametre yoktur. Birden fazla parametre varsa aralarına virgül koyularak sırayla yazılırlar.

Şimdi varolan bir formunuzda düğme oluşturup “Tıklatıldığında” (OnClick) olay yordamına kod yazmak istiyoruz. Düğmenin ilgili özelliğini seçtiğinizde aşağıdaki gibi bir metin oluşturulur.

Option Compare Database

Option Explicit

Private Sub Buton_Click()

..

End Sub

“Private Sub” ile “End Sub” arasındaki kısma yazdığınız kodlar düğmeye basıldığında Access tarafından çalıştırılır. (“Private Sub” ve “Public Sub” deyimlerinden daha önce bahsetmiştik. Bu form için “Private Sub”un “Sub” kullanmaktan farkı yok şimdilik.)

Şimdi bu düğme formu kapatma görevi üstlensin ve siz onun kod bölümüne şunu yazın:

DoCmd.Close

İlk kısımda modüllerle ilgili bahsettiğimiz yazım kuralları burada da geçerli. İsterseniz “Ortalama” işlevini bu düğmenin içinde kullanabilir, isterseniz işlevin kendisini bu formun içine alabilirsiniz. Bir veritabanı içindeki Genel (Global) yordamların aynı adı alamayacaklarını ve formlarda olduğu gibi raporlarda da aynı şekilde kod yazabileceğinizi söyleyeyim.

İPİN UCUNDAN AZICIK

Eğer sıkça kod yazmaya başlamışsanız Tools/Options/Form-Reports (Araçlar/Seçenekler/Form-Rapor) menüsündeki “Always use event procedures” (Her zaman olay yordamlarını kullan) seçeneğini aktif hale getirmeniz faydalı olacaktır. Bu, örneğin bir düğmenin tıklama (click) yordamına bir kod yazmak istediğinizde karşınıza herhangi bir soru çıkmadan, doğrudan modül sayfasında ilgili olay yordamının açılmasını sağlar.

LİSTEDE YOKSA EKLEYİVER

Açılan Kutu (ComboBox) nesnesini kullanırken yeni yazdığınız değer listede yoksa eklenmesini istiyorsanız aşağıdaki kodu deneyin. Önce nesnenin “Listede Yokken” (NotInList) olayının çalışması için kod sayfasını açın. Karşınıza alttaki satırlar gelecektir:

Option Compare Database

Option Explicit

Private Sub AcılanKutu1_NotInList(NewData As String, Response As Integer)

End Sub

“Private Sub..” ile “End Sub” satırları arasına Tablo1’deki kodu ekleyin.

Tablo1

```
If MsgBox("Yazılan değer listede yok. Eklensin mi?") = vbNo then exit sub
Dim db As Database
Set db = CurrentDb
db.Execute "insert into TABLO1(ALAN1) values (' & AcilanKutu1.Text & ')"
'db.Execute "insert into TABLO1(ALAN1) values (' & AcilanKutu1.Text & ')"
'Eger kayıt girilecek alan sayı değil de metin ise 2. satırı kullanın.
'Acilan kutudaki yeni metin ilgili tabloya eklenecek.
'TABLO1(ALAN1) yerine hangi tablonun hangi alanına ekleme yapılacaksa
'onları yazın. KISILER(AdiSoyadi) gibi. tablo ya da alan isimleri
'hata verirse başına ve sonuna köşeli parantez ekleyin.
'[kisi adları tablosu]([adi soyadı]) gibi.

Dim deger
deger = AcilanKutu1.Text
AcilanKutu1.Undo
AcilanKutu1.Requery
Response = acDataErrAdded
'yeni kaydı ben şimdi ekledim. hatayı yoksay.
AcilanKutu1.Value = deger
```

DEBUG WINDOW MU DESEM, HATA AYIKLAMA PENCERESİ Mİ DESEM...

Programı durdurup izlemek yapılan şeyleri gerektiğinde Locals Window ve Immediate Window'dan çokça faydalanabiliyoruz. Immediate Window programa ara verildiğinde (pause) o an çalıştırabileceğimiz kod satırları girmemizi sağlar. Tabii programın orijinal kodları değişmez, sadece o an için ayrı bir pencerede kod yazıp çalıştırma imkanımız oluyor.

LOCALS WINDOW İLE HERŞEYİ GÖRÜN

Şimdi herhangi bir veritabanı dosyamızı açıp içinde ne var ne yok, bakalım. Önce yeni bir modül sayfası açıp şu yordamı ekleyelim:

Sub Deneme()

Dim DB as Database

Set DB=CurrentDb()

End Sub 'bu satıra gelip F9 tuşuna basın.

Locals Window'un amacı yereldeki aktif değişkenlerin (bellekteki bize ait bilgilerin) görüntülenmesini sağlamak. Bu durumda biz ne kadar değişken tanımlarsak, onu Locals Window'dan izleme olanağımız var. Öyleyse bir CurrentDB tanımlamak gerekiyor.

Locals Window'u ancak program çalıştığı durumda izleyebiliriz. Bunun için en son satırın üzerinde iken F9 tuşu ile bir "breakpoint" oluşturalım. Bunu yapınca satırın rengi değişecektir. (İptal etmek için yine F9'a basılıyor.)

F5 ile kodu çalıştırılalım. F9 ile işaretlediğiniz satıra gelince program bekler. Breakpoint'lerin görevi bulunduğu satıra gelindiğinde programı durdurmaktır (pause). Şimdi View / Locals Window menüsünü açalım.

Benim örneğimde bellekte aktif olan iki nesne de pencerede görünüyor. Yanındaki (+) ile listeyi açarak FRM adıyla tanıttığım formun Caption değerini buluyorum. Orada "Form1" yazıyor.

Bu esnada yine View menüsünden Immediate Window'u açacağız. Oraya "? Frm.Caption" yazıp enter'a basın. Hemen altına Frm'nin başlığını yani "Form1" yazdı. Şimdi de (Frm.Caption="Forma yeni isim") yazın. Sonra yeniden "Frm.Caption"u yazdırmayı deneyin. Gördüğünüz gibi program durdurulmuş halde iken istediğimiz gibi müdahale yapabiliyoruz.

ADO GELDİ MERTLİK BOZULDU

Acces2000'de veritabanı işlemleri için ADO kütüphanesi kullanılıyor. Doğrudan bu seçili geldiği için biz babadan kalma DAO'muzu kullanacağız şimdilik. Referanslar kısmından ADO'yu kaldırıp DAO'yu seçin.

Örneğin dosyamızdaki tüm form ve raporları kodla listelemek istiyoruz. Şöyle bir yordam yazıp çalıştırın:

```
Sub Deneme()  
Dim db As Database  
Set db = CurrentDb()  
End Sub 'buraya gelip F9'a basın.
```

Şimdi Immediate Window'da şu satırları çalıştırın:

```
? db.containers("Forms").Documents.count  
? db.containers("Forms").Documents(0).Name  
Application.Assistant.Visible=True
```

Birincisinde dosyamda kaç adet form olduğunu, ikincisinde 1. formun adını öğrendim (0 numarası 1. formu işaret ediyor). Diğer nesnelerin de listesini bulmak istiyorsanız program bu halde dondurulmuş iken Locals Window'u kullanarak DB/Container/Documents öğelerini incelemelisiniz. Üçüncü satırı da siz tahmin edin artık.

Şimdi Immediate Window'da şu komutu çalıştırın:

MsgBox "F5 tuşuna basıp programı devam ettir."

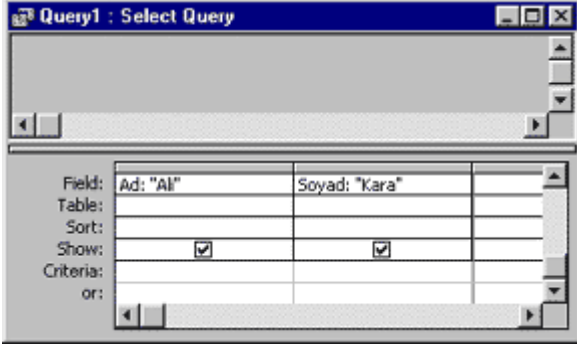
Bu arada (?) işaretinin "print" anlamında kullandığını söyleyeyim. İsterseniz doğrudan "print" yazabilirsiniz. Değer değiştirmek için de Immediate Window'a gerek yok her zaman. Locals Window'da ilgili değeri bulup seçerek değerini değiştirmeyi deneyin.

Bu konuyla ilgili bir de Watch Window var. Onun içine kendimiz değer tanımlı ekleyebiliyoruz. Örneğin biraz önceki örneğimizde bulunan DB değişkenini eklersek Locals Window'da olduğu gibi bunun tüm alt değerlerini görebiliyoruz. Yani bu, Locals Window'un bizim değiştirebildiğimiz hali sayılabilir. Ek olarak buna formül de eklenebilir.

HERKES İÇİN SORGULAR - TABLO OLMADAN SORGU YAPMAK

Tablo olmadan sorgu olur mu dersenez, olur. Hani daha önce her SQL cümlesinde mutlaka olması gereken iki bileşenden, “Select” ve “From” deyimlerinden bahsetmiştik. Sonra da SQL’in standart olanı ve standart olmayanı olduğunu söylemiştik. Bu da standart olmayan bir SQL kodu işte.

SELECT "Ali" AS Ad, "Kara" AS Soyad;



Tablo seçmeden de sorgu oluşturabilirsiniz.

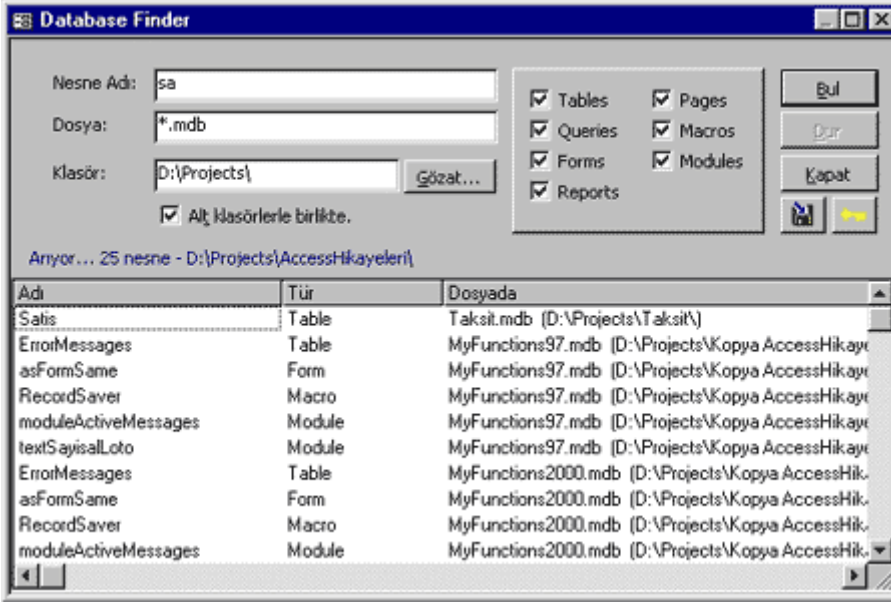
WINDOWS GEZGİNİ, GÖNDER KOMUTUNDA DOSYAYI TAŞIYOR

Windows Gezgininde SendTo (Gönder) komutunu biliyorsunuzdur. Hani sağ fare tuşu menüsünde olan. Kolaylık olsun diye oraya kendinize ait bir klasörün kısayolunu yerleştirdiniz ki tek komutta istediğiniz an bir dosyayı buraya gönderebilesiniz. Gelin görün ki kopyalamak istediğiniz dosya bu klasörünüzle aynı disk sürücüsü üzerindeyse dosyayı kopyalamak yerine taşıyor. O halde komutu seçerken CTRL tuşunu basılı tutmayı deneyin. Böylelikle seçilen dosyanın kendisi değil, kopyası gönderilir.

ACCESS HİKAYELERİ İLK ADD-IN'İMİZ

Access, add-inleri çalıştırmak için dosyada bir işlev arar ve onu çalıştırır. O işlevi biz hazırlarız ve içine programın başlaması için gereken komutları yazmalıyız. Tabii add-in'imizi de Access'e tanıtmak için ayarlar gerekiyor.

Önce “uSysRegInfo” adlı bir tablo oluşturulur. Haliyle add-in'in ilk yüklenmesi için Access'in bu tabloyu kullandığını söylemeliyim. Fakat Add-in'ler registry'de kayıtlı olduğu için bu zorunlu değil ama kolaylaştırıcı. Yani Access add-in yüklerken ya da kaldırırken registry'deki birkaç değeri düzeltmekten başka birşey yapmıyor.



Gelecek yazı Access'te Add-in oluşturma ile ilgili.

Biraz Keşif Biraz Kuşku

Şöyle elle tutulur ipuçları vermek bu ay vacip oldu. Access'in, sorgulama yeteneklerinin dışında, sorguları en umulmadık şekilde kullanabilme gibi yetenekleri de var. Gelgelelim işin ucunda biraz keşif ve biraz kuşku da var.

HAYAT AYRINTILARDA GİZLİ

Bazan en sıradan işlerde dikkat etmediğimiz bazı ayrıntılar birçok kolaylıklar getirebiliyor. Mesela tablo tasarımlarında pek dikkat etmediğimiz Lookup adlı bir özellik sayfası var. Bunu kullanarak tablolara kayıt girerken forma gerek kalmadan açılır listelerden seçim yaptırılabilir. Tablo görünümündeki süzme yetenekleri de cabası.

Tablo tasarımında iken bir alan tanımlayıp altta özellikler kısmındaki Lookup sayfasından ComboBox'u (Açılan Liste Kutusu) seçin. Sonra RowSource (Satır Kaynağı) kısmına gerekli SQL kodunu yazın. Yani seçim yaptırmak istediğiniz listenin kaynağını.

Field Name	Data Type	Description
ID	AutoNumber	Alt taraftaki özellikler kısmında NewValues(YeniDeğerler) özelliğine dikkat.
Ad	Text	Required(Gerekli), Indexed
Soyad	Text	Required(Gerekli), Indexed
Sehir	Text	Sehir tablosundaki şehirleri listeleyip secim yapılacak.

Field Properties	
General	Lookup
Display Control	Combo Box
Row Source Type	Table/Query
Row Source	SELECT Sehir FROM Sehirler;
Bound Column	1
Column Count	1
Column Heads	No
Column Widths	
List Rows	8
List Width	Auto
Limit To List	No

Tablolara kayıt girerken değerlerin listeden seçilmesini sağlayabilirsiniz.

Lookup sayfasındaki diğer özelliklere de dikkat etmekte fayda var. Örneğin Limit To List (Listeye Sınır) değerini YES olarak seçerseniz kayıt girişinde listedeki elemanlardan başka birşey yazılamaz. List Width (Liste Genişliği) içindeki Auto (Otomatik) değeri yerine bir sayı yazarsanız açılan listenin genişliğini belirlemiş oluyorsunuz.

Oradaki diğer özelliklerin fazlalığı dikkatinizden kaçmadıysa daha başka şeyler de anlatmamı bekliyor olmalısınız. ColumnCount (sütun sayısı) değerinin varlığından Lookup işleminde birden fazla sütun gösterilebileceğini anlıyoruz.

Field Name	Data Type	Description
ID	AutoNumber	Alt taraftaki özellikler kısmında NewValues(YeniDeğerler) özelliğine dikkat.
Ad	Text	Required(Gerekli), Indexed
Soyad	Text	Required(Gerekli), Indexed
SehirID	Number	Bu alan şehirlerin numarasını tutuyor.

Field Properties	
General	Lookup
Display Control	Combo Box
Row Source Type	Table/Query
Row Source	SELECT ID, Sehir FROM Sehirler;
Bound Column	1
Column Count	2
Column Heads	No
Column Widths	0cm;8cm
List Rows	8
List Width	Auto
Limit To List	Yes

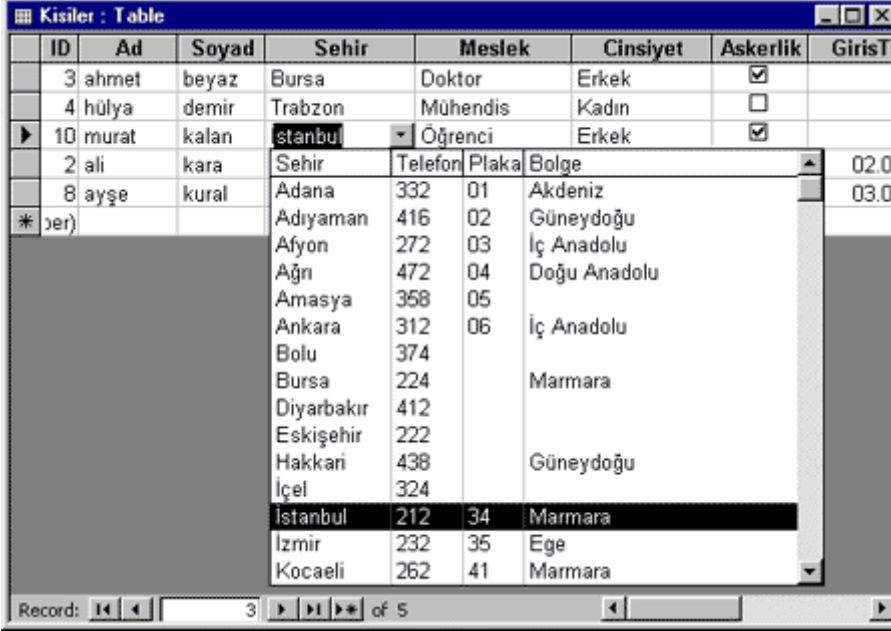
Seçim listesinde istediğiniz sayıda sütunu gösterebilirsiniz. Tabii ki bunlar SQL metni ile seçiliyor.

Resimde Column Widths (Sütun Genişlikleri) değerini “0cm;8cm” yapmamızın sebebi ilk sütunun gösterilmemesini istememiz. (İlk sütunun genişliği 0, ikincisinininki 8 cm.) Çünkü bir listeden herhangi bir şehir seçtiğimizde aslında onun numarası (yani görünmeyen kolondaki değer) kayıtlara girecektir.

BoundColumn değerine bakın. Onun 1 olması, SQL sorgusu ile gelen birinci sütunun kayda gireceğini gösteriyor. Eğer bu sütun, SQL sorgusu içinde ikinci sırada olsaydı BoundColumn değerini 2, sütun

genişliklerini “8cm;0cm” olarak değiştirecektik. Bu arada ListWidth (Liste Genişliği) değeri “Auto” olduğu için açılan liste kutusunun genişliği aktif sütunun genişliği kadar olur. O yüzden sütun genişliklerindeki diğer değerin fazla bir önemi yok şu durumda. ListRows (Listedeki Satır Sayısı) ile de açılan listenin satır sayısı belirlenebiliyor.

Burada Access’çilerin koyduğu birkaç kural var. Mesela açılan listenin BoundColumn sırasındaki alanı görünmez olursa LimitToList değerini YES yapamıyoruz. ColumnHeads (Sütun Başlıkları) değerinin ne olduğunu da tahmin ediyorsunuzdur artık.



ID	Ad	Soyad	Sehir	Meslek	Cinsiyet	Askerlik	GirisT
3	ahmet	beyaz	Bursa	Doktor	Erkek	<input checked="" type="checkbox"/>	
4	hülya	demir	Trabzon	Mühendis	Kadın	<input type="checkbox"/>	
10	murat	kalan	istanbul	Öğrenci	Erkek	<input checked="" type="checkbox"/>	
2	ali	kara	Sehir	Telefon	Plaka	Bolge	02.0
8	ayşe	kural	Adana	332	01	Akdeniz	03.0
			Adiyaman	416	02	Güneydoğu	
			Afyon	272	03	İç Anadolu	
			Agri	472	04	Doğu Anadolu	
			Amasya	358	05		
			Ankara	312	06	İç Anadolu	
			Bolu	374			
			Bursa	224		Marmara	
			Diyarbakir	412			
			Eskişehir	222			
			Hakkari	438		Güneydoğu	
			Icel	324			
			Istanbul	212	34	Marmara	
			Izmir	232	35	Ege	
			Kocaeli	262	41	Marmara	

TABLO KULLANMADAN DEĞER LİSTESİ OLUŞTURMAK

Hepsi bu kadar mı? Değil tabii. Access’çiler ayrıntıları artırıp bu yazının uzaması için ellerinden geleni yapmışlar. RowSourceType (Satır Kaynağı Tipi) özelliğine bakın şimdi.

Şimdiye kadar SQL sorgusu ile aldığımız verileri listeleyebiliyorduk. Şu halde RowSourceType değerini “ValueList” (Değer Listesi) yapıp hemen altındaki RowSource kısmına da “İstanbul;İzmir;Ankara” yazın. Böylelikle ille bir tablodan veri alıp seçtirmeye gerek kalmıyor. Sütun sayısı ve genişliklerini de buna kolayca uyarlayabilirsiniz. Yani iki sütun gösterip “34;İstanbul;35;İzmir;06;Ankara” şeklinde bir listeyi kullanabilirsiniz. O zaman genişliği “0;8” şeklinde ayarlamalısınız.

RowSourceType kısmındaki “FieldList” (Alan Listesi) seçeneğinin ne işe yaradığını anlatmayacağım. Kolayca tahmin ettiğiniz şeyi isterseniz deneyin. Ya da ne olduğunu zaten biliyorsanız boşverin.

KENDİ LİSTENİ KENDİN YAP

“Kendin pişir, kendin ye!” serimizin bu bölümünde Lookup ile ilgili ufak bir ipucunu vermeden geçmeyeceğim.

Excel kullananlar bilirler. Uzunca bir listeyi yazarken listenin yukarısında daha önce yazdıklarınız aktif hücrede önerilir. Belki aynı şeyi yazmak istiyorsanız kolaylık olsun diye. Biz de bunun Access’çesini yapalım diyorum.

Mesela “Kisiler” adlı tablomuzun “Meslek” adlı bir alanı olsun. Ama ayrıca bir “Meslekler” tablosu yapmak istemiyoruz. İstedığımız şey tabloya kayıt girerken o ana kadar girdiklerimizden seçim yapabileceğimiz bir liste oluşturmak.

Önce “Kisiler” tablosunu tasarım görünümünde açıp “Meslek” adlı alan için dizin (indeks) oluşturalım. Ki, kayıt girişinde hızlı çalışsın. Sonra alttaki Lookup sayfasına geçip DisplayControl değerini ComboBox (Açılan Liste Kutusu) yapın. İki altındaki RowSource satırına “SELECT Meslek FROM Kisiler GROUP BY Meslek HAVING (Meslek Is Not Null) ORDER BY Meslek;” yazın (tırnak işaretleri olmayacak şekilde). Başka bir şey yapmadan kaydedin ve tabloyu kayıt listesi görünümünde açın. Artık Meslek alanı üzerinde, eski yazdıklarınızı sıralı bir liste halinde görüp onlardan seçim yapabilirsiniz. Fakat yeni girilen kayıtlar listede yer almayıp tabloyu sonraki açışınızda görüntülenir.



ID	Ad	Soyad	Sehir	Meslek
2	ali	kara	İstanbul	Doktor
3	ahmet	beyaz	Adıyaman	Öğrenci
4	hulya	demir	Ağrı	Mühendis
8	tekin	kural	İzmir	Serbest
10	murat	kalan	İzmir	mühendis

Kendi listesini kendi kayıtlarından seçen bir veri alanı.

GEÇERSİZ VERİLERİ ANINDA KONTROL EDİN

Bütün bu özellikler form ve raporlarda da aynı şekilde kullanılabilir. Ek olarak Validation Rule ve Validation Text yani geçerlilik kısıtlamalarından bahsetmek gerek. Bir veri alanının alabileceği değerler bu özellikleri kullanarak kısıtlanabilir. Mesela doğum tarihi bilgisini içeren bir alana 1900 yılından önce doğmuş kimse kalmadığını düşündüğünüz için oraya girilen değer çok saçma olmamasını istiyorsunuz. Tarihin ille de 1900 yılından büyük olmasını sağlamak istediğinizde Validation Rule (Geçerlilik Kuralı) değerine “>#1.1.1900#” (tırnak işaretleri olmayacak şekilde) yazmalısınız. Uyarı mesajı olarak hemen alt satırdaki ValidationText (Geçerlilik Kuralı Metni) değerine doğru dürüst bir tarih değeri yazılmasını söyleyen bir uyarı metni ekliyorsunuz: “1900 tarihinden önce bir doğum tarihi girdiniz, buna katlanamam!”.

Bu ayarları tablonun tasarım görünümünde yaptığımızı tekrar hatırlatayım. Tüm alanlar için bu kuralı uygulayabilirsiniz. Ama benim asıl anlatmak istediğim bu değil. Bir adet de olsa tablonun kendisine geçerlilik kuralı tanımlama imkanımız var. Mesela kişinin cinsiyeti “Kadın” olarak seçildiğinde askerlik yapıldığına ilişkin bilgi giriliyorsa ikaz etmek gerekir.

Tablo tasarımında iken View/Properties komutu ile açılan Özellikler sayfasında bu değerler tanımlanabilir. Önce geçerlilik kuralını tam anlamıyla formülize etmemiz gerekiyor.

Bu işlemi kod ile yapmak istiyorsanız önce başvurular kısmında (Tools/References – Araçlar/Başvurular) Microsoft DAO 3.6 Object Library (DAO 3.6 ya da DAO 3.5) kütüphanesinin seçili olmasına dikkat edin. Herhangi bir modül sayfasında alttaki kodu çalıştırın.

Dim db As Database

Set db = CurrentDb

Dim td As TableDef

Set td = db.TableDefs("Kisiler")

td.ValidationRule = "Cinsiyet<>'K' or Askerlik=FALSE"

td.ValidationText = "Cinsiyet KADIN olarak seçildiğinde ASKERLİK bilgisi girilemez."

Tabloda birkaç alan arasında ortak geçerlilik kuralını bu şekilde kolayca tanımlayabiliriz. Alanların kendilerinde ise sadece kendileri için kural tanımlanabiliyor. Eğer tabloda yine kuralınız olacak ise onu da uygun bir kural metni oluşturup bu şekilde kaydedebilirsiniz. Mesela işe giriş ve işten çıkış tarihinin bir büyüklük sırası vardır. O halde bunun kuralını aynı tablomuza şu şekilde ekleyebiliriz:

td.ValidationRule = "(Cinsiyet<>'K' or Askerlik=FALSE) and (GirisTarihi<CikisTarihi)"

td.ValidationText = "Girilen değer kısıtlama kurallarına aykırı."

Ana koddaki son iki satırı bu şekilde değiştirmek yeterli. Sonrasında tabloya bu kurallara aykırı olabilecek kayıtlar girmeyi deneyin.

ÇEK BİR TABLO, FİLTRELİ OLSUN!

Tablo tasarımındaki güzellikler dışında süzme ve sıralama seçenekleri de yeterince cazip. Veri girişi görünümünde tabloyu istediğimiz gibi evirip çevirmemize izin veriyor Access.



Birazdan Ad alanı içinde 'ahm' olan kayıtlar görünecek.

Ad alanı üzerine gelip süzgeç kısmına “*ahm*” yazdığımızda içinde “ahm” olan isimlerin bulunduğu kayıtlar listelenecek. “*” işaretinin ve “Like” deyiminin nasıl kullanıldığını biliyorsunuzdur. Bu yazdığımız aslında ‘ Like “*ahm*” ‘ şeklinde kullanılıyor

Ek olarak süzgeç içinde birden fazla alan adını kullanabilirsiniz. Herhangi bir alanın üzerine gelin ve süzgeç kutusuna şunu yazın:

[cinsiyet] = 'e'

Ya da Cinsiyet alanı üzerine gelip şunu yazabilirsiniz:

(='e') and ([askerlik]=false)

Veyahut da CikisTarihi üzerine gelip şirkette bir yıldan uzun süre çalışmış kişileri şu şekilde kolayca bulabilirsiniz:

>=GirisTarihi + 365



ID	Ad	Soyad	Sehir	Meslek	Cinsiyet	Askerlik	GirisTarih
3	ahmet	beyaz	Adiyaman	Doktor	Erkek	<input type="checkbox"/>	
4	hulya	demir	Agn	Doktor	Kadin	<input type="checkbox"/>	
10	murat	kalan	izmir	Öğrenci	Erkek	<input checked="" type="checkbox"/>	
2	ali	kara	istanbul	Doktor	Erkek	<input type="checkbox"/>	02.04.20
8	ayşe	kural	izmir	Mühendis			

Birden fazla veri alanı için süzleme yapılabilir.

Süzgeç menüsündeki diğer komutları da incelemenizi tavsiye ederim. “Filter By Selection” (Seçime Göre Süz), “Filter Excluding Selection” (Seçim Dışında Tutarak Süz), “Remove Filter/Sort” (Süzgeç/Sıralama Kaldır) komutlarını Türkçe anlamlarını vermek dışında anlatmaya gerek yoktur herhalde. Aynı şekilde “Sort Ascending” (Artan Sıralama) ve “Sort Descending” (Azalan Sıralama) seçenekleri de sıralama yapıyor bildiğiniz gibi.

Bu işlemleri aynı anda birden fazla sütun için de yapmak mümkündür. Sütun başlığından tutup birkaç alanı seçin. Fareyi seçili bölge içinde kalan verilerin üzerine getirip fare menüsünü açın. Orada sıralama komutlarını kullanarak seçili alanlara göre ardışık sıralama yaptırabilirsiniz. Menüün aşağısındaki HideColumns, ColumnWidths komutlarını da seçili tüm alanlar için uygulayabilirsiniz.



ID	Ad	Soyad	Sehir	Meslek
3	ahmet	beyaz	Adiyaman	Doktor
4	hulya	demir	Agn	Doktor
10	murat	kalan	izmir	Öğrenci
2	ali	kara	istanbul	Doktor
8	ayşe	kural	izmir	Mühendis

Birkaç alanı birden seçerek ardışık sıralama yaptırılabilir.

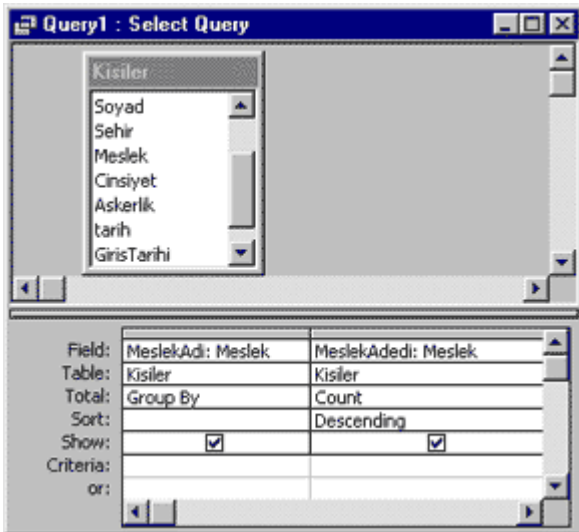
HideColumns (Sütunları Gizle) komutunu seçip de alanları gizleyip kaybedenler için Format/UnhideColumns (Biçim/Sütunları Göster) menüsünün kullanılabileceğini söyleyeyim.

Tablo tasarımlarında yapılan bu ayarlamaların bir faydası daha var. Zaten bütün bunlar form tasarımlarında da yapılabildiği için sihirbaz ile formunu hazırladığınız bir tabloda yaptığınız bu düzenlemeler forma aynı şekilde yansıtılıyor. Mesela Lookup ile listesini başka bir yerden alan veri alanları için aynı ayarların olduğu Combobox (Açılan Liste) yerleştiriliyor forma.

Hatırlarsanız geçen ay AutoNumber (Otomatik Sayı) hakkında bilgiler vermiştik. İşte orada bir noktaya değinmedik. Otomatik Sayı alanları için yeni kayıt anında vazgeçilmesi ile o numaranın bir daha kullanılmaması çok kullanıcı ortamlarında da gerekli. Yani aynı anda beş kişinin kayıt girdiği bir tabloda birinin kayıttan vazgeçmesi otomatik sayının o anki değerinin bir daha kullanılmamasını gerektirir. Aksi halde yönetmek daha masraflı olsa gerek.

HERKES İÇİN SORGULAR

Sıkça ihtiyaç duyulan bir sorgu türü, kayıtlar içinde belli bir değeri gruplayarak herbirinin adedini olduğunu bulmak. Örneğin Kisiler tablosundaki Meslek bilgisini gruplayıp her bir meslekten kaç adet olduğunu öğrenmek ve buna göre sıralamak istiyoruz. Önce yeni bir sorgu oluşturup Kisiler tablomuzu ekleyelim. Sonra View (Görünüm) menüsünden Totals (Toplamlar) komutunu seçelim. Bunu yapınca alttaki kısımda Totals (Toplamlar) satırı görüntülendi. Resimdeki gibi değerleri girip sorguyu çalıştırınca mesleklerin listesini ve herbirinin adedini sıralı bir şekilde görebiliriz.



GROUP BY ve COUNT seçenekleri ile her bir Meslek grubunun sayısını öğrenebiliyoruz.

KOD İLE FORM VE RAPOR TASARLAMAK

Aslında bu başlık kod ile tasarım yapma konusunu anlatmak için biraz yetersiz kalıyor. Zira kod ile sadece formlar ve raporlar değil, makrolar, kodlar değiştirilebiliyor ve hatta veritabanının bütünü başaşağı edilebiliyor.

Bir formu (yani nesneyi) değiştirip kaydedebilmek için normalden farklı olarak yapacağımız tek şey onu normal görüntüleme modunda değil de tasarım modunda açmak ve işimiz bitince kapamak.

Dim f As Form

'Tasarım modunda aç.
DoCmd.OpenForm "form1", acDesign
Set f = Forms("form1")

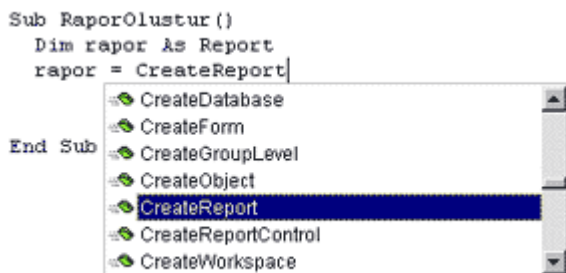
'Gerekli değişiklikleri yap.
f.Caption = "Kisilerle ilgili deneme formu"
f.RecordSource = "Kisiler"
f!text1.ControlSource = "ad"
f!text2.ControlSource = "soyad"
f!text1.Top = 100
f!command0.Left = 10
f!text2.Visible = False

'Değişiklikleri kaydederek kapat.
DoCmd.Close acForm, "form1", acSaveYes

MDE olarak derlenmiş dosyalarda bu şekilde tasarım yapılamıyor.

Dim Frm As Form, Buton As CommandButton
Set Frm = CreateForm
Set Buton = CreateControl(Frm.Name, acCommandButton, , , 200, 200, 2000, 500)
Buton.Name = "Buton1"
Buton.Caption = "Yeni Buton"
Frm.Caption = "Yeni Form"
DoCmd.Close acForm, Frm.Name, acSaveYes
'MsgBox Frm.Name & " adlı form oluşturuldu."
'Hatalı. Başına açıklama işareti koyduk.

Örnekteki gibi kolaylıkla yeni bir form da üretebiliyoruz. Bunu hem tasarım modunda hem de çalışma modunda yapma olanağımız var. Rapor yapmak için de CreateReport kullanılıyor.



Gelelim örnekteki hataya. Kodu çalıştırırken son satırda açık olmayan nesneye başvuru yapılamayacağına dair bir hata mesajı çıkıyor. Tanımlanmış olsalar da formlar kapatıldıkları anda kullanılamaz olurlar. Yani formu kapattığımız an Frm adlı değişken kaybolur. O halde son satırdaki "frm.Name" kısmını silmek gerekiyor.

İPİN UCUNDAN

Eğer bir formu ya da raporu gizli olarak açmak istiyorsanız DoCmd.OpenForm "Form1", acDesign, , , acHidden gibi altıncı parametrede acHidden değerini kullanın. Böylelikle kullanıcıya hiç göstermeden formu tasarım modunda açıp üzerinde istediğiniz değişiklikleri yaptıktan sonra kaydedip kapayabilirsiniz.

ASİSTAN ASİSTAN NERDESİN!

Office asistanından hoşlanıyorsanız onu programlamak da hoşunuza gidecektir. Öncelikle başvurular listesinden “Microsoft Office 9.0 Object Library” kütüphanesini seçmek gerekiyor. Ayrıca asistan da sistemde kurulu olmalı.

Application.Assistant.Visible = True

Application.Assistant.Animation = msoAnimationLookUpRight

Bu şekilde asistanı gösterip onu hareketlendirebilirsiniz. Asistanın Office’te yaptığı herşeyi kodla da yaptırabilirsiniz.

```
Sub Deneme()  
    With Application.Assistant  
        .Visible = True  
        .Animation = msoAnimationLookUpRight  
    End With  
  
End Sub
```



Asistana sağ yukarı bakmasını söyledim. Hep itaatkardır :-)

BANA PROSEDÜRÜNÜ SÖYLE...

Diyelim ki DENEME adlı bir yordamımız var. Bunu normalde Call Deneme(...) ile çalıştırabiliriz. Ama RUN komutu bakın bunu nasıl yapıyor:

Sub Deneme(deger1 As String)

MsgBox deger1

End Sub

Sub RunIleCalistir()

Dim x

x = Run("Deneme", "Run ile deneme")

'Run komutu max 30 adet parametre alır.

End Sub

GEZGİN C'Yİ AÇSIN

Windows Gezgini’ni açtığınızda Bilgisayarım’ın ya da My Computer’in değil de “C:”nin açılmasını istiyorsanız şu kısayolu kullanabilirsiniz:

explorer /e, c:

SİSTEM YAZITİPLERİNİ RESİM GİBİ KULLANIN

Çoğu zaman ne işe yaradığını pek düşünmediğimiz başka bir ayrıntı daha var. Fazlalıklmış gibi duran Symbol, Webdings ve Wingdings fonlarından faydalanmanın zamanı geldi artık.

Resimdeki güzel düğmenin başlığında “3456789.;<=!%)12OYXULzl” yazıyor. Yazıtipi Webdings olarak ayarlı. Access dışında da yani Windows’un tamamında bunu deneyebilirsiniz. Diğer yazıtiplerine de bir gözatin derim. Ufak bir resim arşivi gibi duruyor. Windows’taki pencerelerin sağ üst köşesindeki düğmelerde aslında Webdings yazıtipi ile 0,1,2,r yazıyor gibime geliyor...



Webdings yazıtipi ile yazılmış bir yazı!

HERKES İÇİN SQL

İyi sorgular yapabilmek için iyi hazırlanmış tablolarımızın olması gerekiyor. Dizinler, ilişkiler, geçerli değerler (indexes, relationships, default value) gibi yardımcı özelliklerle veritabanımızın sağlamlığını artırabiliriz.

PERFORMANS, AMA NASIL?

Daha önce karşılaştığım bazı ayrıntıları anlatmakla yetineceğim şimdilik. Sorgu yaparken Access’i yoran şeylerden bir tanesi tablolarda tanımlı DevaultValue (Varsayılan) değeridir. Ekranda bir sürü açılan liste kutusu (combobox) varsa ve herbiri kendi listesini başka bir tablodan araştırarak getiriyorsa formumuzun pek hızlı olmasını beklememeliyiz her zaman.

Örneğin veritabanında, beraber çalıştığımız müşteri şirketler ve bunlara ait personel listesi olsun. Kisiler tablosundaki SirketNo adlı alan Sirketler tablosundaki No alanı ile bağlantılı. Yani biz kişilere ait şirket eşleşmesini şirketin ismi yerine numarasını vererek sağlıyoruz. Liste kutularındaki seçeneklerle gerekli eşleşmeleri ayarlayıp kolayca listeleme yapıyoruz.

Öncelikle bu tür eşleştirmelerde mutlaka dizin (indeks) ve ilişki (relationship) tanımlanmalıdır performans için. Aynı şekilde arama yapılacak alanlarda da dizin oluşturulmalıdır. (Sirketler tablosundaki No alanı.)

Asıl söylemek istediğime gelirse, özellikle şöyle bir ipucu işimize yarayabilir: Eğer varsayılan değeri (default value) sıfır olarak seçilmiş bir alana ait açılan liste kutusunda başka bir tablodaki kayıtları listeliyorsak ve bağlı arama alanı değeri hiçbir zaman sıfır olmayacaksa (mesela autonumber – otomatik sayı alanları için) boş yere sıfır değeri için arama yaptırıp zaman kaybedeceğiz. Zira açılır liste kutusu kendisinde kayıtlı 0 değerini bağlı tablodan arayacak ama hiçbir zaman otomatik sayı türündeki bir alanda bunu bulamayacaktır. Sıfır olmayacak alanlara bağlı sayısal alanlardaki, geçerli olan sıfır değerini kaldırarak formlarımızdaki kayıt geçişlerini bir nebze de olsa hızlandırabiliriz.

Bu, bir süre önce rasladığım küçük ama çok önemli olduğumu düşündüğüm bir ayrıntıydı. Bu ay aslında sorgulardan ve SQL’den bahsetmek istiyorum.

MAZERETİM VAR, SQL BİLMİYORUM

Access ile SQL bilmemize gerek kalmadan hemen her türlü sorgulama işlemini yapabiliriz. Ama yine de Access kullanmadan SQL kodu yazabilmenin inceliklerini birazcık da olsa öğrenmek gerekiyor.

Geçen haftalarda şirketin diğer departmanlarındaki arkadaşlarla bir saat kadar SQL çalıştık. Hepsi de SELECT, FROM, WHERE, ORDER BY başlıkları ile beraber MAX, MIN, SUM, COUNT, TOP, ASC, DESC komutlarını da anlayarak toplantıdan ayrıldılar. Toplantının sebebi uzak erişimli sunucularımızdaki verilere kendi başlarına sadece SQL kodu yazarak ulaşabilmelerini sağlamaktı. Her grup kendi yetki alanı dahilindeki veritabanlarında istediği SQL kodunu çalıştırarak istediği kayıtları listeleyebilmeliydi. Burada da bu ay sadece SELECT komutu dahilinde, yani kayıt listelemek için çalışalım.

SQL’de onu kullanan sistemin yapısına göre yazımda bazı farklılıklar gösterse de yaygın veritabanı sistemlerinin hemen hepsinde kayıt arama ile birlikte, yeni kayıt ekleme, silme, değiştirme ya da veritabanını yönetme gibi gelişmiş özellikler vardır. Mesela Access dosyasında istediğiniz bir tablonun alanlarını ya da dizinlerini SQL kodu ile değiştirebilir veya yeni bir tablo oluşturup bunun içine istediğiniz kayıtları SQL kullanarak ekleyebilirsiniz.

GENELDE SQL

Access bize sorgularda, formlarda, raporlarda, makrolarda ve modüllerde SQL kullanabileceğimizi söylüyor. Kayıt bulup getiren tipik bir SQL cümlesi şu şekildedir örneğin:

SELECT Ad, Soyadi, Yas, Tarih FROM Kisiler WHERE Ad=’Ahmet’ ORDER BY Tarih;

Bu komut (FROM) Kisiler adlı tablodan, (WHERE) Adı “Ahmet” olan kayıtları, (ORDER BY) Tarih alanına göre artan sıralayacak ve (SELECT) Ad, Soyadi, Yas, Tarih alanlarını seçecek şekilde getirecektir.

ORDER BY sıralama yapılacak alanı belirtir. Genelde en sonda kullanılıyor. “ORDER BY Tarih” demekle sonucu Tarih alanına göre artan sıralıyoruz. “ORDER BY Tarih ASC” ile arasında fark yoktur. ASC (Ascending) artan sıralamayı belirtiyor. Yerine DESC (Descending) kullansaydık ters sıralama yapılacaktı. Sadece bununla bitmiyor. Birkaç alana göre de sıralama yapılabilir bununla. Mesela “.. ORDER BY Tarih DESC, Soyad ASC, Ad ASC” gibi. Yani Tarih’e göre ters sıraladıktan sonra önce Soyad’a ve ardından Ad’a göre düz sırala.

Aradığımız şeyi bulmak için WHERE komutunu kullanmamız gerekiyor.

.. WHERE ((Ad=’Ahmet’) AND (Soyad=’Kara’)) ..

şeklindeki bir kriter Ahmet Kara adlı kişinin kaydını arayacaktır. “WHERE Yas>=20” ile 20 yaş ve üzeri kişilerin kayıtlarını bulabiliriz. Ya da “Yas<>20” ile 20 yaşında olmayanlar bulunabilir. Eğer 20-40 yaş arası istiyorsanız “WHERE Yas BETWEEN 20 AND 40” demeniz yeterli.

İsterseniz içinde geçebilecek karakterlere göre de arama yaptırabilirsiniz.

.. WHERE (Ad Like “Ahm*”)..

“Ahm*”: Ahm ile başlayanlar,

“*Ahm”: Ahm ile bitenler,

“*ahm*”: Herhangi bir yerinde AHM olanlar (başında ve sonunda olanlar da dahil),

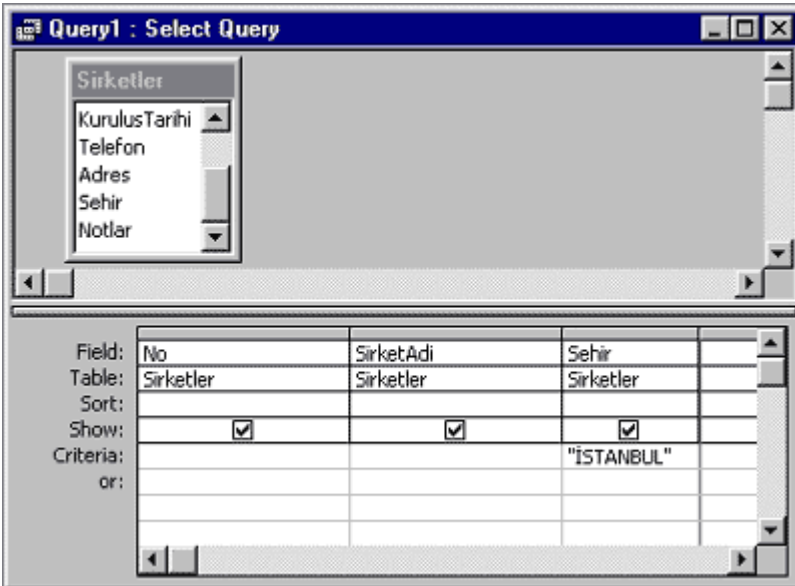
“*a?m*”: İçinde A olacak, sonraki bir adet harfin ne olduğu önemli değil ama onun yanında M harfi olmalı.

Bazı durumlarda çift tırnak yerine tek tırnak yazmanız gerekebileceğini hatırlamanız faydalı olacaktır. Bu örnekleri SQL cümlesi içindeki WHERE’in bulunduğu kısma yazarsanız çalışır. Tek başına bu kriterler çalışmaz.

FROM ile bildiğiniz gibi hangi tablodan kayıt alınacağı belirtiliyor. Şimdilik sadece tek tablo ile çalışıyoruz. SELECT ile bu tablo içindeki istediğimiz alanları görüntülüyoruz. “SELECT Ad, Soyad FROM ..” gibi bir cümle sadece Ad ve Soyad alanlarının getirileceğini belirtiyor. Bütün alanların gelmesini istiyorsanız “SELECT * FROM Kisiler ..” yazın, olsun bitsin.

Arama kriteri yoksa WHERE kullanmaya, sıralama yapılmıyacaksa da ORDER BY kullanmaya gerek yoktur. Yani tek başına çalışabilecek en küçük SQL cümlesi “SELECT * FROM Kisiler” şeklinde olacaktır. Bu, Kisiler tablosundaki tüm kayıtları ve tüm alanları getirir.

ACCESS’TE SQL



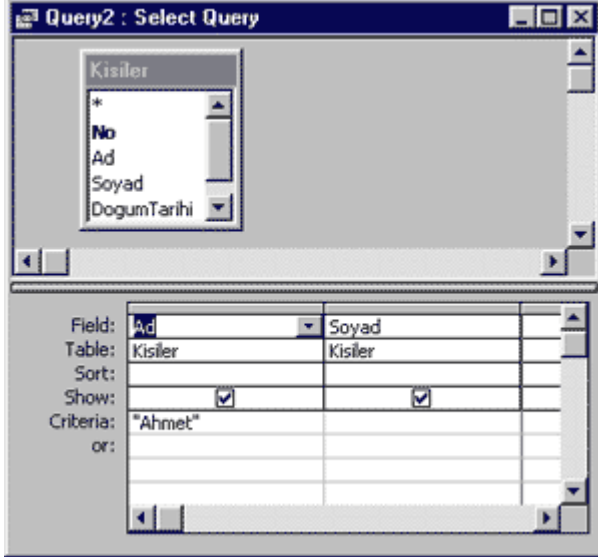
Veritabanı penceresinde Sorgu sayfasında yeni bir Sorgu oluşturalım. Sorguyu açtıktan sonra görüntülenen tablo seçme formundan sorgulamak istediğimiz tabloyu seçelim. (Yeni bir dosyada çalışıyorsanız, tabii ki önce bir tablo oluşturmalısınız.) Karşımıza, adının sonundaki sıfatı fazlasıyla hakeden Sorgu Sihirbazı çıktı.

Ben Kisiler tablosunu seçtim. Şimdi yukarıda öğrendiğimiz SQL kodlarını ilk olarak burada deneyebiliriz. Görünüm menüsünden SQL Görünümü’nü seçin.

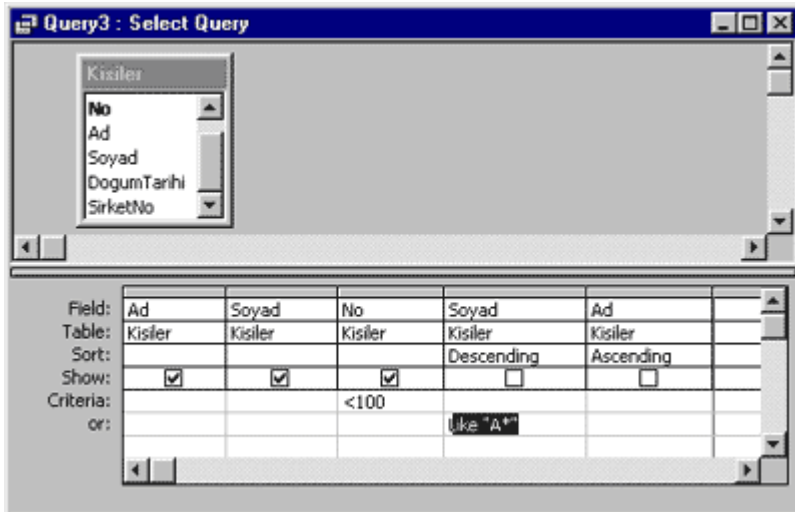
Bende “SELECT FROM Kisiler;” görünüyor. Bunu, öğrendiğimiz kurallara uyarak;

SELECT Ad, Soyad
FROM Kisiler
WHERE Ad="Ahmet"

olarak deęiřtiriyorum. Görünüm menüsünden Tasarım'a geçtięimizde bakın nasıl bir görüntü çıkıyor (Query2):



SELECT kısmında * işareti de kullanabilirdik. Şöyle bir ayrıntıyı hatırlatmalıyım burada ki, bazı durumlarda tasarım görünümünde bütün alanlar olmamasına rağmen sorguyu çalıştırınca bütün alanlar listelenebiliyor. Bu, sorguların bütün alanları gösterme özelliğinin aktif olmasından dolayıdır. Görünüm menüsündeki Özellikler komutundan çıkan pencerede "Tüm Alanları Göster" (Output All Fields) kısmındaki değeri NO yaparsanız sadece sizin seçtiğiniz alanlar listelenecektir.



Üçüncü örneğimizde (Query3 resmi) Ad, Soyad, No, soyad, Ad alanları seçilmiş.

SELECT Ad, Soyad, [No]
FROM Kisiler
WHERE (([No])<100) AND (Soyad) Like "A*")
ORDER BY Soyad DESC, Ad;

1. No alanının parantez içine alınmasının sebebi “NO”nun özel bir kelime olup yorumlayıcı tarafından “Hayır” anlamında anlaşılmamasıdır.

2. En sağdaki Soyad ve Ad alanlarının ikinci kez kullanılmasının amacı sıralamayı sütunların sıralanmasından farklı yapabilmek. Yani önce Ad’a sonra Soyad’ göre sıralamak isteseydik baştaki Ad ve Soyad alanlarında sıralama seçerdik. Ama biz önce Soyad’ı istediğimiz için son tarafa bunları koyup Göster (Show) işaretini kaldırdık. (Aynı şeyi 2. soyad alanını birincinin olduğu yere taşıyarak da yapabiliriz. Zira, hem ORDER BY hem de SELECT için seçili alanların sırası aynı olacak.)

3. Sadece Göster (Show) değeri işaretli olan alanlar görüntülenir. Bunlar SELECT kısmına dahil edilir. (Tüm alanları gösterme özelliği kapalıysa tabii.)

4. Dördüncü sütundaki LIKE cümlecini No alanının alt satırında kullanmamızın sebebi VEYA sorgusu elde edebilmek. Yani “ NO<100 OR SOYAD LIKE ‘A*’ ” gibi. İkisini aynı satırda kullanırsak VE sorgusu elde ederiz ki bu, cümlemizdeki “OR”un “AND”a dönüşeceği anlamına gelir.

Bu tasarım modunda ne kadar karışık ya da sade yaparsak yapalım Access evirip çevirip kendi bildiği gibi bir SQL kodu üretecektir.

```
SELECT Kisiler.Ad, Kisiler.Soyad, Kisiler.[No], *  
FROM Kisiler  
WHERE (((Kisiler.[No])<100)) OR (((Kisiler.Soyad) Like "A*"))  
ORDER BY Kisiler.Soyad DESC , Kisiler.Ad;
```

Beşinci madde olarak da şunu ekleyeyim o zaman: Alan isimlerinin başına tablonun ismi yazılabilir. Bu, birden çok sayıda kullanılan tabloların olduğu sorgularda daha işe yarar olur.

EXCEL’DEKİ GİBİ OTOMATİK SAYI SERİLERİ OLUŞTURMAK

Access’te Excel’deki gibi artan satırlarda otomatik sayı oluşturmak ne iyi olurdu diye düşündüğünüz olduysa ilk fırsatta deneyin derim. Kayıtlar arasında aşağı giderek seri olabilecek sayılar girin. (Yani 1, 3, 5, 7 .. gibi.) Oklarla aşağı doğru gittikçe yeni değerleri anlayıp sizin yerinize Access yazacaktır.

KENDİ FORMÜLÜNÜ KENDİN HESAPLA

Kullanıcının kendi formülünü hesaplayabilmesini sağlamak istediğiniz oldu mu hiç? EVAL işlevi bu iş için biçilmiş kaftan. Siz formül metnini veriyorsunuz, o da hesaplıyor:

```
MsgBox Eval("6^2 + 4*3 – Log(Exp(10))")
```

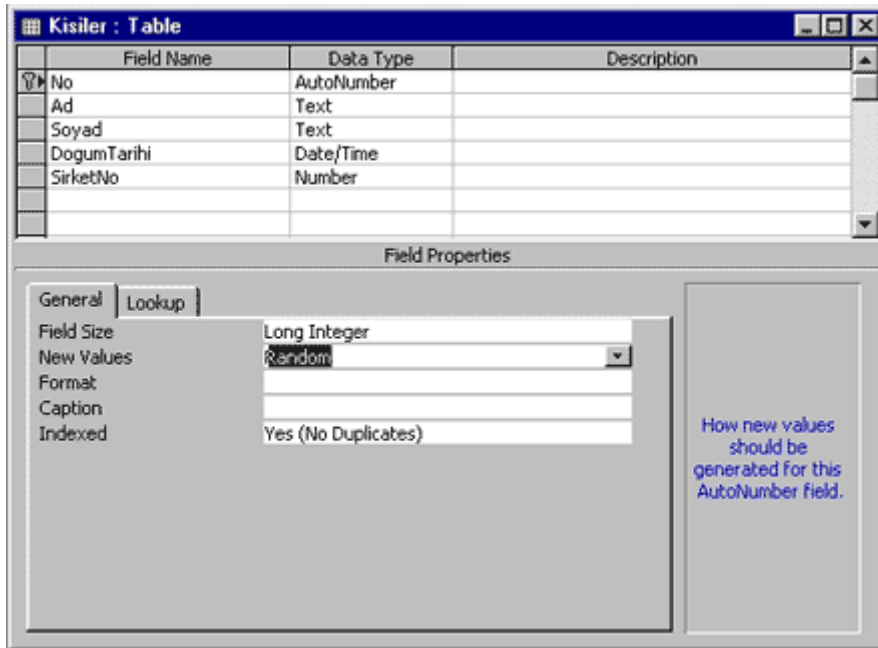
OTOMATİK SAYI

Bazan bu isim kendi kendine çalışan sayılar varmış gibi çağrışımlar uyandırıyor bende. Otomatik çamaşır makinası gibi. Gelgelelim bazı arkadaşlar bunun marifetlerinden pek muzdaripler. “Otomatik sayı, kayıt numarası neyse onu versin”, “Yeni kayda geçip vazgeçtiğimde numarayı alıyor, bir daha aynı numarayı kullanamıyoruz”, “Bütün kayıtları sildiğim halde 1’den başlamıyor” gibi yakınmalar oluyor.

Efendim, bu otomatik sayı öyle birşeydir ki görüntü açısından bir anlam taşıması için üretilmemiştir. Diğer bütün sistemlerde olduğu gibi en büyük amacı belli bir kayda özel bir kod vererek istendiğinde ona kolayca ulaşabilmeyi sağlamaktır. Mesela bütün alanların değerleri aynı olan iki kaydı birbirinden ancak Otomatik Sayı değerleri ile ayırabilirsiniz. Yani arkaplan işlemleri için kullanılır. Ama isterseniz ekranda da değerini gösterin.

Bunun artan olacak şekilde değerler almasını ayarlayabileceğiniz gibi rasgele değerler almasını da sağlayabilirsiniz. Tablo tasarım görünümünde NewValues (Yeni Değerler) özelliğini Random (Rasgele) yaparak.

Yine de programı kullanacak kişiler açısından bir daha tekrarlanmayacak bir fiş numrası, kod numarası, etiket numarası gibi bir değer elde edilmiş olur. Access'teki replikasyon işleminde de kullanılabilir AutoNumber alanlar.



ACCESS HİKAYELERİ

FARE MENÜSÜYLE MP3 LİSTESİ

Şimdi bunun Access'le ne ilgisi var? Yok tabii. Ama dolaylı olarak Access'le çalışırken MP3 dinlemek istiyor insan. Hele bu eski makinemde Access'le Winamp'ı aynı anda çalıştırmak istemeyen biri olarak Media Player tercihim. Media Player madem MP3 çalışıyor, M3U da çalışıyor. Hatta M3U liste dosyasının özel bir biçimde yazılmış olmasına gerek yok. Yeni bir metin dosyasının içine MP3 dosyalarının isimlerini klasör adlarıyla birlikte altalta yazın. Metin dosyasını kapatıp uzantısını M3U yapın ve Media Player'da açın.

Yeni bir metin belgesi daha oluşturup şunları aynı şekilde yazın:

```
@echo off
dir %1\*.mp3 /on /b /a-d >mp3ler.m3u
echo Liste oluşturuldu... OK...
echo Bu pencereyi kapatabilirsiniz...
echo.
```

Sonra bunu mp3liste.bat adıyla c:\ içine kaydedin. Şimdi de Windows Gezgin'i (Explorer) açıp Görünüm/Seçenekler menüsünü çalıştırın ve Dosya Türleri (FileTypes) sayfasına geçin. Listedeki Klasör'ü (Folder) seçin. Yeni (New) tuşuna basın. Üstteki yere "MP3 Listele", alttaki yere de "c:\mp3liste.bat %1" yazın ve Tamam (OK) tuşuna basın. Yine Tamam (OK) tuşlarıyla işlemleri onaylayın.

Böylece windows'ta herhangi bir klasörün üzerine geldiğinizde sağ fare menüsündeki MP3 Listele komutunu kullanarak o klasördeki MP3'leri aynı klasörün içinde "mp3ler.m3u" dosyasına kaydedebileceksiniz.

Eğer "dir %1*.mp3 /on /b /a-d /s >mp3ler.m3u" şeklinde "/s" eklerseniz tüm alt klasörleri arayarak klasör isimlerini de dosyaların başına ekler.

Fakat C: D: gibi bir kök dizinde bunu yapıyorsanız bu kod çalışmaz. Çünkü kök dizinlerde parametre olarak \ işareti gelirken klasör isimlerinde bu ek yoktur. O zaman "dir %1*.mp3..." yerine "dir %1*.mp3..." yazmalısınız.



Fikret Kızılok ve Fahir Atakoğlu favorim :-)

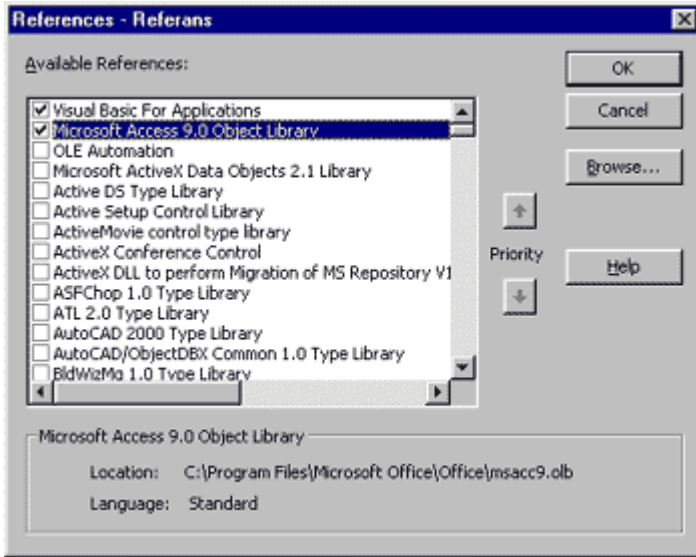
Access'te Başvuru Kütüphanesi (Reference Library) Yazmak

Biraz derinlere inip Access'teki References (Başvurular) listesine kendimizinkini ekleyeceğiz.

REFERENCE LIBRARY

Diğer Office uygulamalarında da olduğu gibi Access'te hem standart nesne kütüphaneleri hem de Activex kütüphaneleri kullanılabilir. Bu yazıyı yazdığım Word2000'de Visual Basic Editor'ü açtığımda Tools/ References menüsünden tüm başvuruları görebiliyorum. Aynısı Access için geçerli olup Office'te programlama anında tüm bu kütüphanelerden istediğimiz gibi faydalanabiliyoruz.

Access'te boş bir dosya açıp modül sayfasında yeni bir modül oluşturalım. Tools/ References (Araçlar/ Başvurular) menüsü ile başvuruların listelendiği pencereyi görelim.



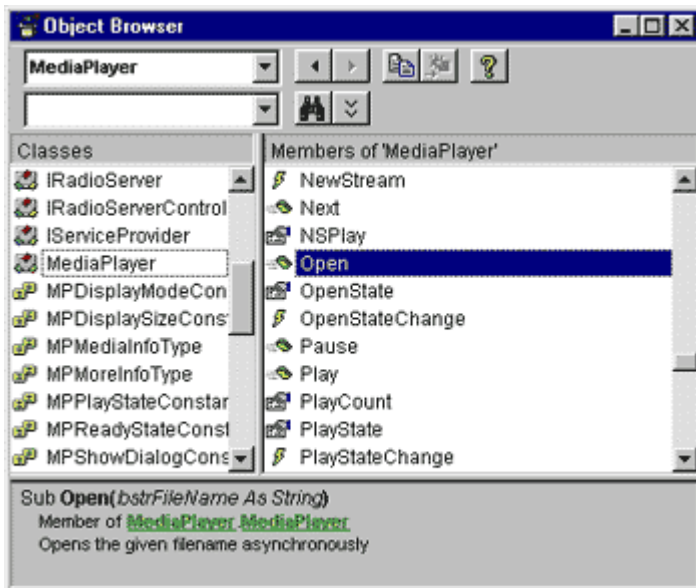
MSACC9.OLB dosyasının Microsoft Access 9.0 Object Library kütüphanesi olduğunu öğreniyoruz. Visual Basic For Applications ile beraber bu ikisi tüm Access uygulamalarında kullanılan dosyalar. Access bunları kaldırmamıza izin vermiyor.

Örneğin Access projenizde müzik çalmak istiyorsunuz ve elinizde bir Wav dosyası var. Bunun için eğer sisteminizde Microsoft Media Player yüklü ise büyük ihtimalle başvurular listesinde buna ait kütüphane görünecektir. O zaman çok kısa bir kodla müzik çalmayı başarabiliriz. Başvurular listesinden Windows Media Player'i seçip alttaki dört satırı çalıştırın.

On Error Resume Next

```
Dim muzik As New MediaPlayer.MediaPlayer  
muzik.Open "c:\windows\media\ding.wav"  
muzik.Play
```

Gördüğünüz gibi çok kolay. Ama listede onlarca kütüphanenin hangisinin nasıl kullanıldığını nereden öğreneceğiz? Üstteki müzik çalma örneğini vermeden önce başvurular listesinden hangisini örnek vereyim diye aradım. Hiçbirinin nasıl kullanıldığını zaten bilmiyor idiysem bana sadık dostumuz Object Browser (Nesne Gözetici) yardımcı olmuş demektir. F2'yle onu açtıktan sonra sol üstten Media Player'i seçerek içindekileri biraz incelemek gerekiyor.



İlk olarak müziği çalacak nesneyi bulmak lazım. Bunu sol alt listeden alıyoruz. İlk bakışta siz de aradığımız şeyin “Media Player” adında olduğunu anlayacaksınız. Bunu seçince sağdaki listeden de Open, Play gibi şeyler aramamız gerektiği de hemen aklımıza gelenlerden. Open’i seçtiğimizde alttaki kısımda nasıl kullanılacağı gösteriliyor. Bir dosya adını parametre olarak vermek gerekiyormuş. Sonra da Play’a bakacağız ve üstteki birkaç satırlık kodu çok kısa bir sürede keşfederek oluşturacağız. Eğer Media Player’iniz MP3 çalışıyorsa sizin de bu şekilde MP3 çalmanıza hiçbir engel yok. Hatta Access’te kendi müzik listenizi bile yapabiliyorsunuz bununla. Tabii VB’ciler tüm bunların VB’de de yapılabileceğini zaten biliyorlardır.

KENDİ BAŞVURU KÜTÜPHANENİ KENDİN YAP!

Geldik kendi başvuru dosyamızı yapmaya. Yeni bir Access dosyası açıp içine birer adet tablo, form ve rapor yerleştirelim. Örneğimizde üçü de “Kisiler” adında tablo, form ve rapor var. Sonra modül sayfasında yeni bir modül oluşturup aynen şunları yazıyorum:

Option Compare Database

Option Explicit

Public Function Topla(sayı1 As Long, sayi2 As Long) As Long

Topla = sayi1 + sayi2

End Function

Public Sub KisilerFormunuAc()

DoCmd.OpenForm "Kisiler"

End Sub

Public Sub KisilerRaporunuAc()

DoCmd.OpenReport "Kisiler"

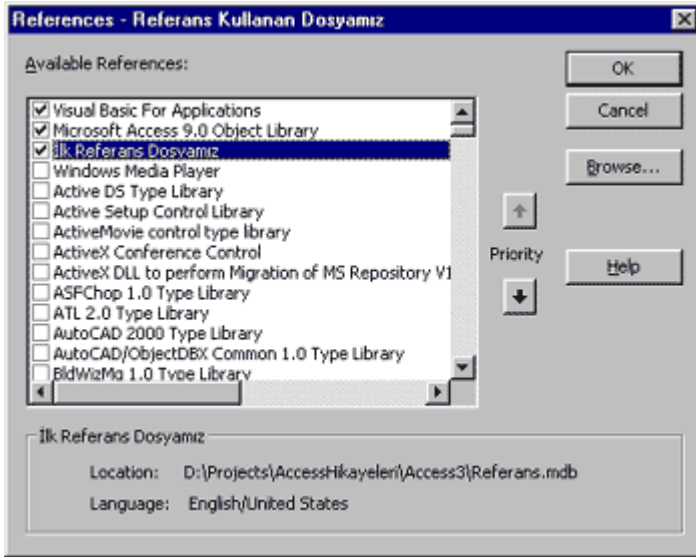
End Sub

Bir başvuru dosyasının içindeki bir formu, raporu veya diğer şeyleri açabilmek için aynı dosya içinde bu işleri yapan kodlarımızın olması gerekiyor. Örnekte rapor ve formu açabilmek için iki ayrı kod var. “Topla” adlı işlev ise genel bir yardımcı işlev görevi üstlenecek. Yani biz bu başvuru dosyasını nereye eklersek ekleyelim bu işlevi Access’in kendi işleviymiş gibi kullanıp, form ve raporu da açabileceğiz. Bu dosyamızın adı Referans.mdb olsun.

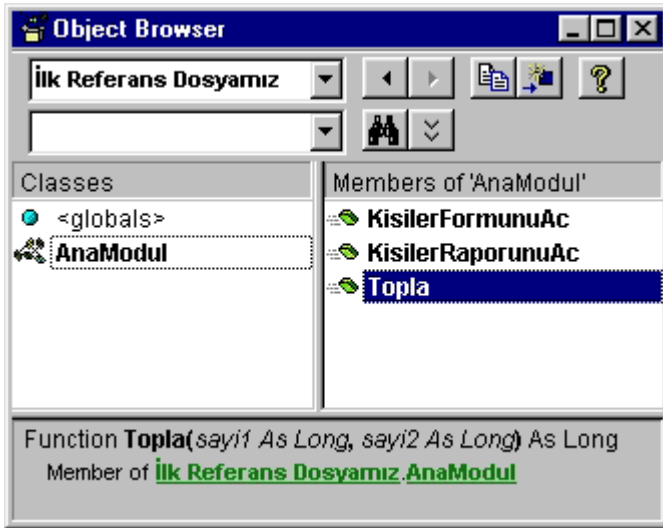
Zorunlu olmasa da bir ayar yapmamızda fayda var. Başvurular kısmında bu dosyanın görünen ismini belirlememiz gerekiyor. Yani projemizin adını. (Visual Basic Editor'de Tools/Properties/ProjectName.)

Application.SetOption "Project Name", "İlk Referans Dosyamız"

Bunu yaparak başvurular kısmında dosyamızın adının “İlk Referans Dosyamız” şeklinde görünmesini sağladık.



Şimdi bunu kullanmak için kapayıp yeni bir Access dosyası açıyoruz. Yeni bir modül sayfası açarak Başvurular bölümünde bu dosyayı bulup seçelim. Sonra Object Browser ile ilk başvuru kitaplığımızı inceleyelim.



İlk başvuru dosyamızdaki kullanılabilir işlevleri Object Browser ile görebiliyoruz.

Şimdi alttaki satırları çalıştırmayı deneyelim.

KisilerRaporunuAc
KisilerFormunuAc
MsgBox toplu(3, 5)

Böylelikle hem başka bir Access dosyası içindeki form ve raporu açabiliyoruz, hem de kendi ortak işlevimizi yazmış oluyoruz.

Tüm bunlardan sonra yapılması önerilen son bir işlem daha var. Başvuru dosyamızı derleyip kodlarını gizlemek ve kullanıma tam hazır hale getirmek. Şimdi Referans.MDB dosyasını tekrar açıp MDE dosyası oluşturalım. Tools/ Database Utilities/ Make MDE File (Araçlar/ Veritabanı Hizmet Programları/ MDE Dosyası Yap) seçeneği ile MDE dosyamızı oluşturduk. Bu şekliyle hem kimse kendi özel kodlarımızı ve tasarımlarımızı göremeyecek ve değiştiremeyecek hem de bu dosyamız normal bir EXE gibi hızlı çalışacak.

Dosyamız hazır. Kendisini belirtmek için uzantısını örneğin REF olarak değiştirelim. Şimdi asıl programımızdaki başvuru işaretini kaldırıp Referans.REF dosyasını yeniden yüklememiz gerekecek.

Bu ilk kütüphanemizi Access içinden bir DLL ya da OCX gibi kullanabiliriz artık.

Başvurularla ilgili genel olarak anlatacaklarım bu kadar. Sonrasında bununla ilgili birkaç ipucu vermekle yetineceğim. Bu özelliği kullanmaya karar vereceğiniz zaman CodeDB ve CurrentDB adlı iki nesneyi hatırlamanız faydalı olabilir. Bir başvuru dosyası içinde CurrentDB kullanırsanız bu, başvuruyu kullanan dosyayı işaret eder. Ama CodeDB kullanıyor iseniz bu da kodun bulunduğu dosyayı yani başvuru dosyasının kendisini işaret etmiş olur. Sıkça kullandığımız CurrentDB yerine CodeDB kullanmamızın sakıncası yoktur aslında. Halihazırda CodeDB zaten kodun yazılmış olduğu uygulamanın kendisini ifade ediyor.

İPUÇLARINA DEVAM - Bir Formu Veya Raporu Birden Çok Kere Açmak.

Access'te de VB ve diğer dillerde olduğu gibi bir formu ya da raporu aynı anda istediğimiz kadar açma şansımız var. İşte örnek:

Global Frm1 As Form_Form1

Global Frm2 As Form_Form1

Sub AyniFormu3kereAc()

DoCmd.OpenForm "Form1"

Forms("Form1").Caption = "Bu, Form1'in orijinali"

Set Frm1 = New Form_Form1

Set Frm2 = New Form_Form1

Frm1.Caption = "Bu, Form1'in ilk kopyası"

Frm2.Caption = "Bu da Form1'in ikinci kopyası"

Frm1.Visible = True

Frm2.Visible = True

MsgBox "FORM1'in 2 adet kopyası açıldı."

End Sub

Burada kullandığımız Form_Form1 tanımlaması Form1 formuna ait sınıfın adıdır. Geçen sayıda bahsettiğimiz sınıf modülleri konusunda bundan bahsetmek için acele etmedik. Normalde form ve raporlara ait kodların bulunduğu modüller aslında sınıftır ve bir sınıf ile yaptığımız herşeyi form ve raporlara ait modüllerle de yapabiliriz. Burada şunu unutmamakta yarar var. Eğer FRM1 ve FRM2'yi aynı yordam (SUB) içinde tanımlasaydık kod bitip bilgiler bellekten atılırken bu formlar kapanacaktı. Bunu önlemek için onları "Global" deyimi ile tanımladık.

Şimdi yeni bir form veya rapor oluşturup onun üzerinde deneme yapmaya çalışırken neden hata oluştuğunu sorabilirsiniz. Böyle bir durumda form veya raporunuza ait bir modülün varolmadığından şüphelenin. O zaman bunun kod kısmına bir harf de olsa birşeyler yazıp sınıf modülün oluşmasını sağlamalısınız. Çünkü kod olmadan sınıf modülü olmaz.

Gelecek sefer gelişmiş sorgu (query – sql) örneklerini incelemek istiyorum. Sadece Access'çilerin değil, SQL kullanan ve de kullanmak isteyenlerin bunu kaçırmamasını öneririm. Sonrasında Add-In yazma konusuna başlayabilmek için kod ile Access formlarının ve raporlarının nasıl düzenlendiğinin örneklerini inceleriz. Ama ondan önce ODE Tools'tan bahsetmeyi planlıyorum.

Güneşli güzel günler...

ACCESS HİKAYELERİ ODE TOOLS GÜNLERİM

Sevgili Günlük. Access97 için ODE Tools'u satın almış bulunuyoruz. Yanlışlıkla getirdikleri yükseltme sürümünden sonra temiz bir sürüm elde ettik. Office97 Developer Edition adlı bu kutu içinde hem Office97 Professional var, hem de Office Developer Tools (ODE Tools). En önemlilerinden biri de Office programlama için iki güzel kitap. Access'te ya da Office'te programlama yapmaya çalışırken neyi yapıp neyi yapamayacağımızı bu kitaplar gayet iyi anlatıyor.

O güne kadar MDB dosyalarını EXE'ye dönüştürdüğünü zannettiğim Access Setup Wizard'ının arayüzü de Access'te yapılmış. EXE'ye dönüştürme de zaten MDE dosyaları kısmen EXE'ler gibi çalıştığı için sorun yok. Artı ve eksiler şöyle:

- Setup Wizard ile güzel bir kurulum programı hazırlıyoruz ve programımızı Access'in kendisi olmadan müşterimize kurabiliyoruz. Paket gerekli dosyaları bizim için toparlıyor. İçine bizim programımızla birlikte Access Run-Time modülünü koyuyor.
- Bir bilgisayarda sadece bir adet Run-Time modülü olması yeterli. Sonrasında istediğimiz kadar MDB, MDE dosyasını kopyalarak bunları çalıştırabiliyoruz.
- Haliyle müşterimize Access kurmaya gerek kalmıyor.
- Türkçe Access'te hazırlanan bir kurulumu İngilizce Access olan makineye kurduğumuzda maalesef onun dilini Türkçe'ye çeviriyor. Terside geçerli. Bir bakıyoruz form tasarımlarındaki Properties penceresinde Özellikler yazmaya başlamış. Çok acıklı değil ama neyse...
- Kurulum hazırlama yeteri kadar ayrıntılı. VB'dekinde olduğu gibi birçok seçenek burada da var.
- Ayrıca kurulumda ODBC ve diğer veritabanı seçenekleri var. İstenirse grafik (Chart), Outlook ve Exchange modülleri de dahil edilebiliyor.
- OCX kullanmış isek onu listeye kendimiz ekliyoruz ve Wizard, o OCX için gerekli diğer dosyaları kendisi buluyor.

Gelgelelim bazan kafa karıştırıcı şeyler de oluyor. Bir tanesini denerken kurulum sırasında bazı hatalar oluştu. Neden olduğunu zar zor anlayabildik. Meğer ki ayarları yaparken klasör ismi sorulan yerlerde klasörün tam adı yazılacak, ama isminin sonunda bölü işaretini olmayacakmış. O kadar çok makine ve sistemde test ettim ki rasladığım problemlerin altından kalkmam epey vaktimi aldı. Sağolsun Microsoft'un destek sitesi böyle durumlarda ilaç gibi geliyor.

Epeydir, çözemediğim bir hata olduğunda en iyi yardımcılarımdan olan Altavista'ya o hata mesajının aynısını soruyorum. Akıllı AV şimdiye dek en az on adet büyük problemimin yarısından fazlasında, verdiği ilk cevapta çözüme ulaştırmıştı beni. Bunlar da MS'nin destek sitesi oluyor çoğunlukla. Sonradan Ask.com'u keşfettim. İlk sorduğum şey "where are you?" idi. Cevabı görünce gözlerim yaşarmıştı. ".. at AskJeeves!". Tabii bir de azar işittim. Soracak doğru dürüst birşeyim yok muymuş!

Kurulum paketi hazırlamaya devam ederken bir yandan da deneylere devam ediyorum. Zira paketi hazırlamak azıcık uzun sürüyor. Acaba "D" sürücüsüne kurulu yepyeni bir Windows'ta bu kurulumu çalıştırdığımda herşey yerli yerine oturacak mı? Cevap evet oldu. Son olarak merak ettiğim iki şey vardı. Kendi başvurularımızı eklediğimiz dosyalar sabit klasör adını arayacakları için başvuru problemleri ile karşılaşabilir miyiz? Bunun da cevabı evetti, maalesef ki. (Tabii o zaman bunun başka türlü çözebileceğini bilmiyordum.) Bir de Access 2000 yüklü makinelerde Access 97 için hazırlanmış kurulum sorun çıkarmadığı için testler dizisini sadece bir fireyle bitirmiş oldum.

Sonuç olarak Setup Wizard bize, dosyaları onarma (repair/compact) komutları dahil Programlar menüsündeki kısayolları da ayarlayan ve gerekirse Registry'ye kurulumla ilgili kayıtlar giren güzel bir kurulum paketi yapıyor.



EXE yapmasak da programlarımızı Access'e gerek kalmadan kurabileceğiz şimdi.

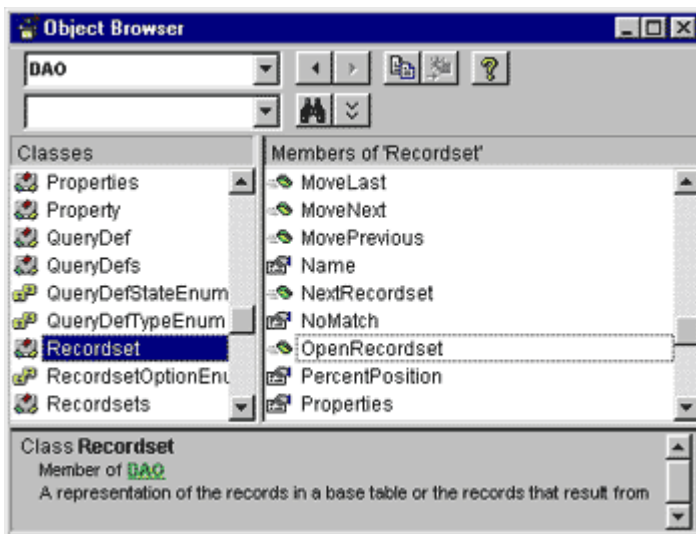
Hoşçakal Günlük.

DOSYA:

[AccessRef.zip](#)

Class modülleri (Sınıf Modülleri) ayrı nesneler olarak kullanır, onlara istediğimiz gibi işlemler yaptırabilir ve onları çağırıp tekrar kullanabiliriz. Sıkça kullanacağımız bir işlemi sınıf modülüyle kolayca yapabiliriz. Bu konunun aslında doğrudan Visual Basic (VBA) ile ilgili olmasından dolayı biz biraz daha Access'e yaklaştırmaya çalışacağız. O yüzden okuyucunun sınıf modüllerini temel olarak bildiğini, en azından bu yazıdan bunun kısmen anlaşılabilirliğini varsayıyorum.

Sıkça kullandığımız "Recordset" tanımını Sınıf yani bir Class'tır. Nesne Gözetici (Object Browser) öyle söylüyor:



Class Recordset

Member of DAO

A representation of the records in a base table or the records that result from

```
Dim RS As Recordset  
Set RS = CurrentDb.OpenRecordset("tablo1", dbOpenDynaset)  
RS.AddNew  
RS.Fields("isim") = "Hasan Baltacı"  
RS.Update  
RS.Close
```

Dim ile RS adlı değişkenin Recordset sınıfından olmasını sağladık. Addnew bu sınıfın metotlarından biri. Bu metot, RS'nin kayıt ekleme moduna geçmesini sağlıyor. Fields ise sınıfa ait bir özellik. Update ve Close ise diğer metotlardan.

LİSTE ÖZELLİKLİ BİR SINIF MODÜLÜ

Bizim sınıf modülümüz virgülle ayrılmış metinleri çözüp listeleyebiliyor ve sıralayabiliyor. Veritabanı penceresinde (Database Window) Modül (Module) sayfası açıkken Insert/ Class Module (Ekle/ Sınıf Modülü) menüsünü seçip ilk sınıf modülümüzü ekleyelim.

İlk bakışta herhangi bir modülden farkı yok.

Aşağıdaki kodu yeni bir sınıf modülü (Class Module) açıp yazın. Sonra bunu MyListBoxComma adı ile kaydedin. Unutmayın, modülü her zaman bu adla çağıracağız.

‘SINIF MODÜLÜ: MyListBoxComma

```
Option Compare Database  
Option Explicit
```

```
Private Dizi() As String  
Private Toplam As Integer  
Private Ayirac As String * 1
```

```
Private Sub Class_Initialize() ‘Sınıf yükleniyor.  
Toplam = 0  
Ayirac = ","  
End Sub
```

```
Public Sub AddItem(Strx As String) ‘Listeye eleman ekle.  
Toplam = toplam + 1  
ReDim Preserve dizi(1 To Toplam) As String  
Dizi(Toplam) = Strx  
End Sub
```

```
Public Property Get Seperator() As String ‘Ayıraç.  
Seperator = Ayirac  
End Property
```

```
Public Property Let Seperator(ByVal vNewValue As String)  
Ayirac = vNewValue  
End Property
```

Public Function Clear() ‘Listeyi temizle.

Erase Dizi

Toplam = 0

End Function

Public Property Get ListCount() As Integer ‘Listedeki eleman sayısı.

ListCount = toplam

End Property

Public Sub Sort() ‘Listeyi sırala.

...

...

End Sub

Private Function GetMin(sira As Integer) As Integer

‘Sıralamak için kullanılıyor. Sadece bu Class içinde

‘kullanılabilir. Çünkü Public olarak değil, Private olarak tanımlanmış.

...

...

End Function

Public Function SetComma(ByVal Strx As String)

‘CommaText olarak girilen metni listeye dönüştürür.

...

...

End Function

Public Function GetComma() As String

‘Tüm listeyi ayıraç kullanarak cümle haline getirir.

...

...

End Function

Bu modülden örneğin şu şekilde faydalanabiliriz:

Sınıfı MyListBoxComma adı ile kaydettiğimiz için bu adı kullanmalıyız.

‘New ekini de kendi yazdığımız sınıfları kullanabilmek için yazmak zorundayız.

Dim Liste As New MyListBoxComma

Liste.AddItem "ocak"

Liste.AddItem "subat"

Liste.AddItem "mart" ‘... Tüm ayları yazın.

MsgBox liste.ListCount & " adet eleman:" & Chr(13) & liste.GetComma

Liste.Sort

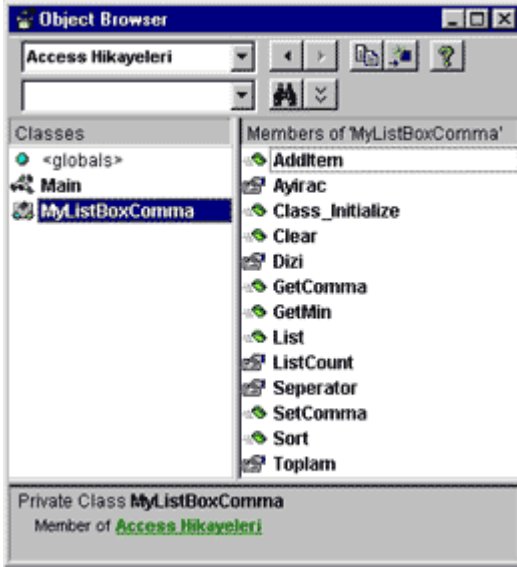
Liste.Seperator = Chr(13) 'Satirsonu işaretini(13) koyunca altalta listelenecek.

MsgBox "Listenin sirali hali:" & Chr(13) & liste.GetComma

Liste.Seperator = "-"

Liste.SetComma ("sonbahar-kis-ilkbahar-yaz")

MsgBox Liste.ListCount & " adet oldu simdi."



Modülü biraz inceleyince Private ve Public şeklinde iki tanım görüyoruz. Bildiğiniz gibi Private ve Public tanımlamaları yordamların dışardan kullanılıp kullanılamayacağını belirliyorlar. Bilmeyenler için Public kullanmadan dışarıdan okuma ya da değiştirme yapılamayacağını ve Private kullanırsak sadece modülün kendi içinden okuma ve yazma yapabileceğimizi de söyleyeyim.

Burada normal işlev (function) ve yordamlardan (sub) farklı olarak Property Get ve Property Let tanımları da kullanılmış. Örneğin Seperator (Ayırac) özelliği için hem Get hem de Let yöntemi var. ListCount için ise sadece Property Get var. Bu durumda ListCount özelliği sadece okunabilir, Seperator ise hem okunur hem de değiştirilebilir olmuştur.

Class_Initialize yordamı da anlayacağınız üzere bu modülün ilk çalışmaya başladığı yerdir. Fakat kullanılması zorunlu değildir. Aynı şekilde sınıflar bellekten atılırken varsa Class_Terminate adlı yordam çağrılır.

RAPOR YAZDIRIRKEN KOPYA SAYISINI SEÇMEK

“Rapor1” adlı raporu 3 defa yazdırmak için şu kodu deneyin:

DoCmd.OpenReport "Rapor1", acViewPreview

DoCmd.PrintOut acPrintAll, , , 3

Örneğin 6-8 arası sayfaları 3 kez yazdırmak için ikinci satırı şu şekilde değiştirebilirsiniz:

DoCmd.PrintOut acPrintRange, 6, 8, , 3

PrintOut komutu aktif olan rapor, form ya da tabloyu yazdırır. Kullanmadan hemen önce yazdırmak istediğiniz belgenin aktif olmasını sağlayın.

SINIRSIZ PARAMETRELİ İŞLEV (FONKSİYON) YAZMAK

Bazan öyle olur ki sınırsız parametre verebileceğimiz yordamlara ihtiyaç duyarız. İşte örneği:

Function FormatStr(ByVal MainStr As String, ParamArray StrList() As Variant) As String
‘ParamArray en son parametrenin değişken uzunluklu bir dizi olmasını sağlıyor.

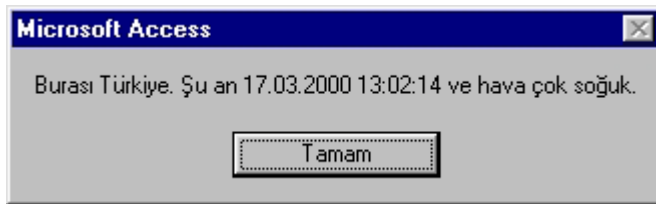
```

If UBound(StrList) < 0 Then Exit Function
Dim i As Integer
For i = 0 To UBound(StrList)
MainStr = Replace(MainStr, "%" & i + 1, StrList(i))
Next
FormatStr = MainStr
End Function

```

Bunu şu şekilde kullanın:

MsgBox FormatStr("Burası %1. Şu an %2 ve hava %3.", "Türkiye", Now(), "çok güzel.")
 'Resimdeki mesaja aldırmanın siz :)



ParamArray olarak tanımladığınız parametreyi en sona yerleştirin. Aksi halde hata oluşur.

DAHA GÜZEL KAYIT SİLME

Access'in kendi komut düğmesi sihirbazı ile gelen kayıt silme kodu ekrana uyarı iletisi çıkarıyor ve bu genelde iyi bir görüntü oluşturmuyor. Bunun yerine aşağıdaki kodu kullanabilirsiniz.

```

Public Sub KayitSil(Optional ByVal frm As Form, Optional ShowMsg As Boolean)
On Error Resume Next
If ShowMsg Then
If MsgBox("Kayit silinecek. Devam edilsin mi?", vbCritical + vbYesNo) = vbNo Then Exit Sub
End If
If frm Is Nothing Then
DoCmd.SetWarnings False 'Uyari mesajlarini gosterme.
DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
DoCmd.SetWarnings True 'Sonraki uyari mesajlarini gosterebilirsin.
Exit Sub
Else
If frm.NewRecord And (Not frm.Dirty) Then Exit Sub
End If

  'Yeni kayıt ya da kayıt degisikligi ise önce UNDO yap.
If frm.Dirty Then DoCmd.DoMenuItem acFormBar, acEditMenu, acUndo, , acMenuVer70

If Not frm.NewRecord Then 'Eski kayıt ise DELETE
With frm.RecordsetClone
.Bookmark = frm.Bookmark
.Delete
.MoveNext
If .EOF Then .MoveLast
frm.Bookmark = .Bookmark

```

End With

End If

End Sub

Kullanımı şöyle:

Call KayitSil (Me,True)

Eğer ikinci parametreyi vermezseniz ya da False olarak verirsiniz “Kayıt silinsin mi?” mesajı görünmez.

Call KayitSil (Me)

Aslında “Me”yi yani formun kendisini parametre olarak yollamanıza her zaman gerek yok.

Call KayitSil ()

ya da,

Call KayitSil (, True)

Bu şekilde isterseniz bir formdan ya da bir tablodan kendi mesajınızı göstererek kayıt silme işlemi yaptırabilirsiniz.

Gördüğünüz gibi bazı parametreleri vermemize her zaman gerek yok. Bunu yordamın başlığındaki “Optional” seçeneği ile sağlıyoruz. Optional belirttiğiniz ilk parametreden sonraki diğer parametrelerde de mutlaka aynısını kullanmanız gerekiyor.

Önceki sayıda SQL kodlarında tarih kullanımıyla ilgili verilen ipucunun Delphi’deki karşılığı soruluyor. VB’deki Format işlevi yerine FormatFloat bu işi görür:

Tarih:= #05.23.2000#;

// ya da Tarih:=EncodeDate(2000,5,23); //23.05.2000

ShowMessage(FormatFloat('#',Tarih)); //36669

Query1.SQL.Text:='SELECT Table1.* FROM Table1 WHERE Table1.Tarih=36669';

Geçen ayki ipucuna bakmayanlar için Access ya da VB’de Format(Now,“#”) komutunun kullanılabileceğini ve SQL kodlarında da bu şekilde belirtme yapılabileceğini söyleyeyim.

Tüm örnekleri ve Access’te yapılmış ufak bir Sınıf modüllü oyunu Ailenizin Web Sitesi’nde (<http://www.pcmagazine.com.tr>) Dergide Adı Geçen Dosyalar bölümünde bulabilirsiniz. Gelecek sefer ilk başvuru kütüphanemizi (Reference Library) yazacağız.

ACCESS HİKAYELERİ
HİKAYE BU YA...

Bazan Access'in son üç sürümünü aynı anda kullanmak gerekiyor. Office'in en sevdiğim özelliklerinden biri yeni kurulumlarda eski sürümlere acıyıp onlara kıymaması. Yapmamız gereken tek şey yeni kurulacak klasörü eskisinden farklı belirlemek.

Daha önce Access2'de yapılmış programlarımızı Access97'ye dönüştürmemiz gerekiyor. Ağdaki makinaların bir kısmında MS'nin binbir programlı özel paketinden çıkan Office97 Türkçe sürümü yüklü. Bir kısmı ise mecburen aldığımız İngilizce sürüm.

Access'le çalışan tüm tanıdıklarım malum Türkçe meselesinden yakınıyor. Bizimkilerin sorunu da nedir bir bilseniz! Form adlarında yumuşak g ve diğer Türkçe karakterler var diye hiçbir kod çalışmıyor. Neyse ki farklı dildeki bir Windows'ta bunu düzeltebildik.

Şimdi gelgelerim yüzlerce sorgu (query), form, kod içinde bu formun adı kullanıldığı için arkadaşlar başlarının kısmen belada olduğunu düşünmeye başladılar. Bazan bu işler imkansız gibi görünse de Access'teki gizli yardımcılarımız yine yardımımıza koşuyor.

Tools/Analyze/Documenter menüsünü çalıştırdık. Documenter, dosyamızın içinde ne var ne yok bir bir gösteriyor. Bilmem hangi formun hangi düğmesinin başlığı nedir, hangi sorgunun sql kodu nasıldır gibisinden herşey var. Böylelikle Access dosyamızın içerisinde Türkçe karakter olan tüm değerleri kolay yoldan bulacağız.

İşin sırrı nerede? File menüsündeki "Save As Table" komutunu çalıştırmak gerekiyor ki o kadar zamanımızı alan bu raporu heba etmeyelim. Gerisi "Object Definitions" adlı tabloyu incelememize kalmış.

Baştaki konuya dönersek MDB dosyalarının çift tıklatılmasıyla hangi sürümün açılması gerektiği önceden bilinmeli. Bu durumda dosyaların açılma seçeneklerine bunları yerleştirmek lazım. Bir de bizim kendi programlarımız için verdiğimiz uzantının da aynı şekilde ayarlanması gerekiyor.

DAX uzantılı dosyamız bir şekilde HKEY_CLASSES_ROOT\DAX_auto_file dizinine yerleşmiş. Windows Gezgini'nde sağ tuşla tıklatınca Aç komutu doğrudan Access2'yi açıyor. Şimdi biz buraya Access97 ve Access2000'i ekliyoruz.

HKEY_CLASSES_ROOT\DAX_auto_file\shell\open\command anahtarının değeri Access2'yi gösteriyor. Yeni değerler oluşturduk:

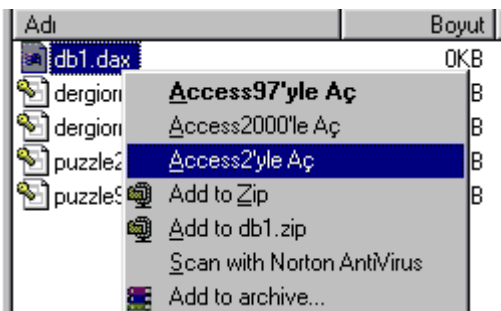
HKEY_CLASSES_ROOT\DAX_auto_file\shell\Access97 İle Aç\command

HKEY_CLASSES_ROOT\DAX_auto_file\shell\Access2000'le Aç\command

Her ikisinin Default değerine msaccess.exe dosyalarının yolunu yazdık.

Tabii bunun çalışması için HKEY_CLASSES_ROOT\dax anahtarında "DAX_Auto_File" yazıyor olmalı.

Eğer dosya Auto_File olarak tanımlı değilse daha kolay bir yolla aynı şeyleri yapmamıza izin veriyor Windows. Bildiğiniz gibi Windows Gezgini'nde Görünüm/Seçenekler/Dosya Türleri menüsünden yapılabiliyor bu.



Sonuçta resimdeki gibi bir sağ fare tuşu menüsü hiç fena değil bence.

DOSYA:

[AccessHaziran2000.zip](#)

Bu bölümde Access'le ilgili ipuçları, çözümler ve temel kurallarla ilgili bilgilerle beraber gelişmiş programlamaya daha çok ağırlık vermeye çalışacağız.

Başlıktaki 'hikaye' kelimesine gelince neden böyle bir başlık verdiğimi söyleyeyim. Bu yazı uzunca bir süredir Access ile çalışırken aldığım notların bir ürünü. Okuyuculardan gelen mesajlar doğrultusunda bu dizinin geri kalan kısmını şekillendireceğiz. Çok rasladığımız bazı sorunlar ve önemli ipuçları yazının devamında yer alıyor.

1. Otomasyon Hatası/ Derleme Hatası

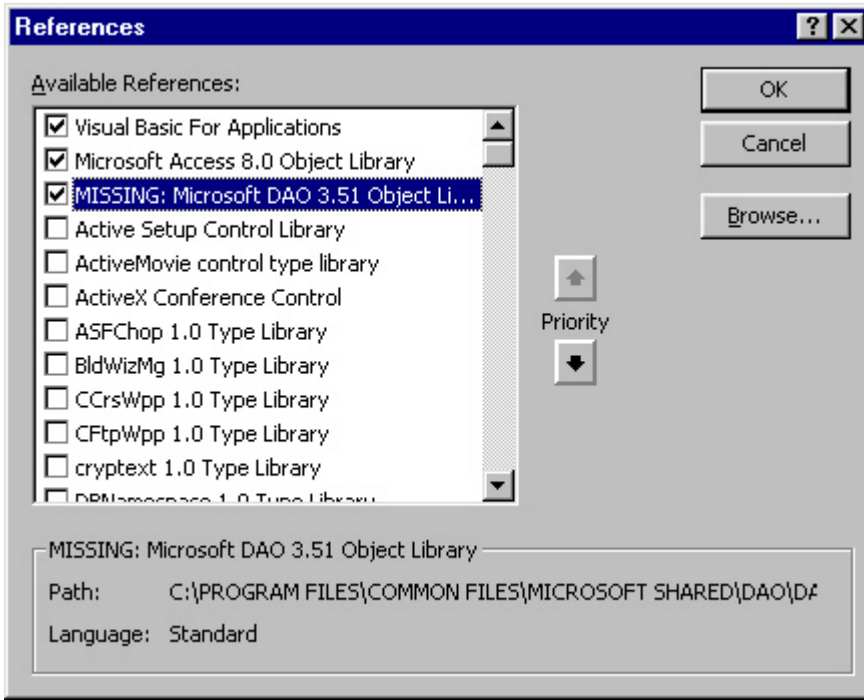
(Otomation Error/ Compile Error)

Bu çokça raslanan bir sorun. Otomasyon Hatası daha çok modüllerin derlenemediği durumlarda meydana çıkıyor. Bunun birçok sebebi olabilir. Bunları tam anlamıyla bilmesem de karşılaşp çözebildiklerimi anlatmayı yeğliyorum. Access'te kod yazmaya yeni başlamışsak bazı işlevlerin neden hata verdiğini anlamak için kendimizi epey yormamız gerekir. Mesela farklı dildeki (ya da konfigürasyondaki) iki Access'ten birinde Mid, InStr vs. komutlar çalışırken diğerinde hata oluşabiliyor. Sebebi çoğunlukla başvurulardaki bir eksikliktir.

References

(Başvurular)

Başvurular listesini herhangi bir modül açıkken "Tools/References" menüsünden görebiliyoruz. Burada "Missing:" (Kayıp) olarak belirtilmiş isimler arayın. Eğer varsa bunu kaldırıp yeniden yüklemeniz sorunu çözebilir. Altteki resimde başka bir bilgisayarda farklı klasör altında bulunan DAO350.DLL dosyası bulunamadığı için hata oluşmuş ve bu başvuru "MISSING:" olarak belirtilmiş. Daha sonra başvurulardan ayrıntılı olarak bahsedeceğiz.



DAO başvurusu bulunamadığı için dosya “MISSING:” (kayıp) olarak belirtilmiş.

2. Modül Penceresi Açılmıyor ve Kodlar Çalışmıyor

Bu problemin bazan doğrudan başvurularla ilgili olduğunu düşünebiliriz. Bazan da projemizdeki bir kod hatası buna sebebiyet verir. Herhangi bir modül sayfası açıkken “Debug/ Compile And Save All Modules” (Derle/ Tüm Modülleri Derle Ve Kaydet) menüsünü çalıştıralım. Bu işlem nerede hata olabileceğini birçok zaman gösterir. Eğer aynı adı taşıyan iki yordam varsa, ya da örneğin bir IF satırının bitirilmemiş olması veya “END SUB” kullanılmaması gibi hataları bu şekilde farkedebiliriz.

Bazı durumlarda derleme işlemi de sonuç vermeyebilir. O zaman bizim yapabildiğimiz son şey yeni bir dosya oluşturup hatalı olan dosyadan tüm modülleri, formları ve raporları kopyalamak. Bu esnada tüm modüller alınırken hatalı olanlarda ekrana ileti çıkıyor. Tabii ondan önce hatalı dosyamızı başka makinelerde test etmekte yarar var. Bazan başka bir yerde Compact (Veritabanı Düzenle) ve Repair (Veritabanı Onar) işlemleri sorunu giderebiliyor.

Bunun dışında bir de Türkçe karakter içeren nesne isimlerine dikkat etmek gerekli. Veritabanı içindeki nesne adlarında (Tablo, Sorgu, Form vs.) türkçe karakterler kullanmaktan olabildiğince kaçınmanızı öneririm.

3. Trim() ve Mid() İşlevleri Çalışmıyor

Bu hata çoğunlukla başvurularla ilgili. Trim işlevi VBA kütüphanesinin bir üyesi. Bunu Object Browser’dan (Nesne Yöneticisi) öğreniyoruz.

Bu dosyayı başvurular listesine tekrar yüklemek gerekebilir. Object Browser bileşeni ileride eklentiler yazarken bize epey yardımcı olacak. Gerekirse bunu daha sonra ayrıntılı olarak ele alabiliriz.

Object Browser (Nesne Yöneticisi)

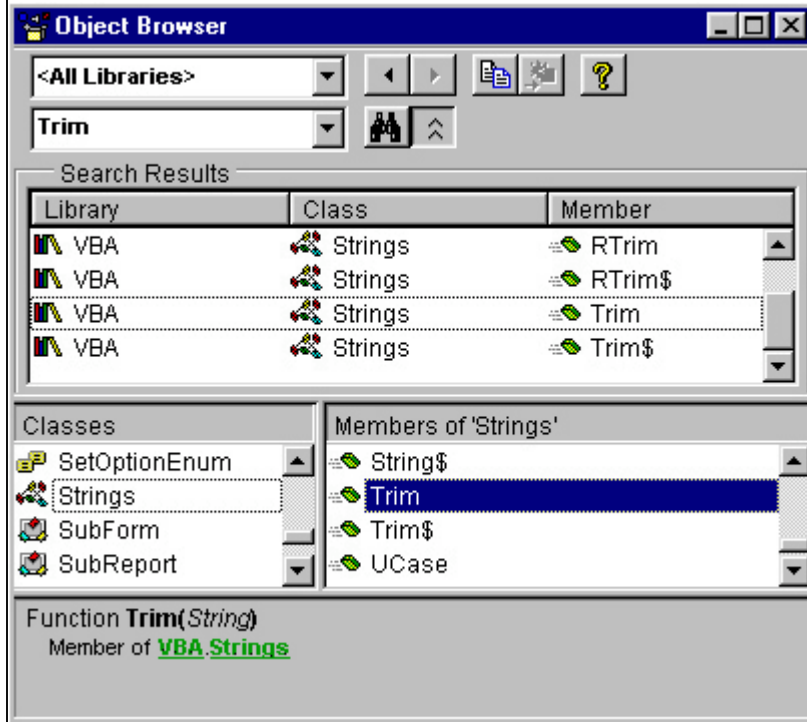
Trim komutunun bulunduğu kütüphaneyi öğrenmek için Object Browser’ı açıp Trim kelimesini aratıyoruz. Burada o an Access içinden kullanılabilecek tüm işlevlerin ve bileşenlerin listesini

bulabiliriz. Aşağıda yer alan VBA yazısını tıklattınca VBA kütüphanesinin içeriğini görüyoruz. Bu kütüphanenin dosyası VBA332.DLL adlı dosya.

Library VBA

C:\Program Files\Common files\Microsoft Shared\Vba\Vba332.dll

Visual Basic For Applications



4. Program Penceresi Kapanmıyor

Bunun sebebi açılan kayıt kaynaklarının kapatılmaması ve kısmen de düzensiz bellek kullanımı olur çoğunlukla. Mesela,

Sub RecordSetDeneme()

Dim RS as recordset

Set RS=CurrentDb.OpenRecordset("Table1", dbOpenDynaSet)

...

RS.Close

End Sub

Bu yordam devredışı kalmadan önce mutlaka RS.Close komutu ile açık olan kayıt kaynağı serbest bırakılmalıdır. Eger Database bileşeni tanımlanmış ise o da Nothing değerine eşitlenmeli.

Dim DB as Database

...

Set DB=Nothing

gibi. Aynı şekilde mantıksal değer ifade eden işlev parçacıklarının IF komutu içinde mutlaka TRUE ya da FALSE değerlerinden birine eşitlenmesi öneriliyor.

If DenemeFonksiyon() Then ... yerine

If DenemeFonksiyon()=True Then ... kullanılmalı.

Bu hatanın çözümüne The Access Web sitesinde rasladım. Her Accessçinin mutlaka bakması gereken bir site olduğunu söyleyeyim. (<http://mvps.org/access>)

5. “(AutoNumber)” / “(Otomatik Sayı)” Metnini Gizlemek

Veri türü otomatik değer olan bir veri alanını formdaki metin kutusuna eklediğimizde yeni kayıt anında “(AutoNumber)/(OtomatikSayı)” metni görünüyor. (Veri alanının adı “KayitNo” olsun.) Bunu engellemek için önce yeni bir form açıp üzerinde bir metin kutusu nesnesi oluşturalım. Adı “txKayitNo” olsun. Şimdi bunun içine veri alanının ismini “KayitNo” yerine şu şekilde yazıyoruz: “=[KayitNo]”.

Burada dikkat etmemiz gereken şey, formda “KayitNo” adlı başka herhangi bir nesnenin bulunmaması. Aksi halde kayıt verisi yerine aynı adlı nesnenin değeri görüntülenir.

6. Label/Metin Etiketi Nesnesinde Yeni Satır Kullanmak.

Label nesnelerinde başlığı yazarken yeni bir satıra başlamak için <Enter> yerine <Ctrl+Enter> tuşlarını kullanın. Aynı şekilde tablo görünümünde de bu tuş bileşimi yeni bir satıra başlamamızı sağlıyor. Bunu komut düğmeleri ve diğer nesneler için de deneyebilirsiniz.

Normalde Label ve CommandButton başlıklarında kısayol komutu için harflere altçizgi verirken "&" işaretini kullanıyoruz. Fakat bu işaretin kendisini nasıl göstereceğiz? Bunun için de “&&” yazmak yeterli.

7. Nesneyi Kendi Olay Yordamında Gizlemek.

Örneğin bir düğmenin "Click" yordamı içinden kendi görünür özelliğini kaldırmak istiyoruz (visible=false). Fakat odaklanılmış nesnelerde böyle bir işlemin geçersiz olduğuna dair hata mesajı çıkıyor. Bu satırdan önce başka bir nesneye odaklanmamız sorunu çözecektir. Yani;

Sub Button1_Click()

...

'Button1'in kendisini görünmez yapabilmek için Button2'ye odaklanalım.

Button2.SetFocus

Button1.Visible=false

...

End Sub

8. Formlara Eklenen Resimler Dosya Botunu Çok Büyütüyor

Form ya da raporlara özellikle büyük resimleri eklerken bunların BMP formatında olmasına dikkat edin. Ayrıca yüksek kalitede resim kullanmak, dosyanızın boyunun gereksiz yere artmasına neden olur. Daha önceden 300 KB'lık bir JPEG (16milyon renkli, 72 dpi) resmini dosyama eklemek istediğimde dosyanın boyutunun 2,5 MB arttığını gördüm. Aynı resmi aynı kalitedeki BMP dosyasına dönüştürüp aktardığımda dosyanın boyu sadece 350 kb. artmıştı.

9. SQL Kodlarında Tarih Kriterleri Sorun Çıkartıyor

Pek çoğumuz SQL kodlarında tarih arama kriterinin tanınmaması gibi bir sorunla karşılaşmışsınızdır.

“SELECT * FROM Table1 WHERE Tarih=#4.1.2000#”

gibi bir cümlede hata olduğu mesajı ile karşı karşıya kalırız. Sorun tarih ayırıcının “.” yerine “/” olması gerektiğinden kaynaklanıyor olabilir. Ya da veriler yanlış geliyorsa gün ile ay kısmının yer değiştirmesi gerektiğini düşünebilirsiniz. Ve bunlar her bilgisayarda farklı olabiliyor. Bu sorunlardan kurtulmak için aşağıdaki şekilde deneyebilirsiniz.

Format(Date(), "#") ‘=36617 1.4.2000 için

Eğer tarihi belirtmek gerekiyorsa gün ile ayın hangisinin daha önce yazılacağına içinde hiç şüphe kalmadan şu komutu tercih ederim:

Format(DateSerial(2000, 4, 1), "#") ‘=36617

Artık tarih değerinin açıkça yazılması yerine doğrudan aynı değere karşılık gelen 36617 sayısı SQL cümlesi içinde yer alacak ve bizi sıkıntıdan kurtaracaktır.

“SELECT * FROM Table1 WHERE Tarih=36617”

gibi. Bu, hemen hemen tüm SQL uygulamaları için geçerli. 31.12.1899 günü 1 olacak şekilde her tarihin mutlaka bir sayısal değeri vardır. Veritabanları da tarih verilerini bu şekilde saklıyorlar, sadece gösterimde biçim uyguluyorlar. Tarihin saat kısmı ise tahmin edeceğimiz üzere ondalık olarak kaydediliyor.

10. Query/Sorgu Yazımında Hata

Access’in Türkçe sürümünde sorgu yazarken “AND” ve “IS NULL” yerine “BETWEEN”, “REPEAT SECTION” gibi metinler otomatik olarak ekleniyor ve hatalı sorgu metinleri oluşuyor. Birçoğunuzun bildiği gibi bu hatalar servis güncellemeleri ile giderilebiliyor. En azından sorgu yazımında bu tür hataları önleyebiliyoruz. Sorgu yazarken Query Builder (Sorgu Sihirbazı) yerine SQL görünümünü kullanırsak bu tür kod değişiklikleri olmayacaktır. Fakat Query Builder ile yaptığımız ilk kayıta bu değişiklikler uygulanır.

Bu sayıda farklı birkaç konuyla ilgili bazı ipuçları vermeye çalıştık. Okuyuculardan gelen istek ve öneriler doğrultusunda da diğer bazı konulara değinmeye çalışacağız. İleriki aşamalarda Access’tе başvuru kütüphaneleri, içine-ekleler ve sihirbazlar yazmayı deneyeceğiz ve Access 2000’e geçmeyi unutmayacağız...

Otomatik Sayı (Autonumber) alanları ile ilgili birçok yerde hala aynı sorular soruluyor.

- Kayıtlar silindiğinde neden eski numaralar kullanılamaz oluyor?
- Bu alanı 1’den itibaren yeniden sıralayamaz mıyım?

Verilen cevaplar hep aynı. Herhalde bunu bilmeyen de kalmamıştır. Bu veri alanının temel özelliği her yeni kayıt anında otomatik artması. Örneğin Access içinde iken yeni kayıt moduna geçildiği anda bu değer otomatik olarak artırılır. ADO ile SQL Server’a bağlanırken konumlandırıcı tipine göre ekleme komutu gönderildikten sonra bu değerler değişik seçeneklere göre artırılabilirler. (*)

Numaraları 1'den itibaren sıralamak için veri alanı silinip yeniden oluşturulabilir. Ama eğer bu alan başka bir veri alanı ile bağlantılıysa veritabanında karmaşa oluşacaktır.

Yandaki resme bakıp aldanmayın. Normal şartlarda Otomatik Sayı alanı değiştirilmez. Benim yaptığım şey sadece veritabanı dosyasına dışarıdan müdahale etmektir.

ID	ad	departman
2	tekin	teknik
3	idil	idari
5	pelin	pazarlama
6	inci	insan kaynakları
9	serhat	servis
12	bilgen	bilgi işlem
15	murat	muhasebe
1	DENEME1	
2	DENEME2	
3	DENEME3	
4	DENEME4	
5	DENEME5	

Bu işlemin nasıl yapıldığını anlatıyorum fakat kullanmanızı tavsiye etmiyorum ve problem çıkmayacağını garantilemiyorum.

Deneme amaçlı bir veritabanı ve bu veritabanı dosyasında otomatik sayı alanı içeren bir tablo oluşturdum. Bu tabloya 15 kayıt ekledim ve bazı kayıtları sildim. Bu durumdayken dosyanın bir yedeğini aldım.

Normal şartlarda Otomatik Sayı değerinin veritabanı dosyası içinde bir yerlerde tutulması gerekir. Bu durumda otomatik sayı ile ilgili bir işlem sonrasında veritabanı dosyasında bir yere bu bilgi yazılmalıdır.

Bu teoriden yola çıkarak tabloda yeni kayıt satırına gelip birşeyler yazdım ama sonra ESC tuşu ile kayıttan vazgeçtim. Böylelikle son kullanılan otomatik sayı değeri dosyanın bir yerine yazılmış olur.

Dosya kapatıp orijinal ve yeni dosyaların arasındaki farkları program kodu yazarak araştırdım.

```
Sub FarklariBul()  
    Open "c:\autonumber1.mdb" For Binary As #1  
    Open "c:\autonumber2.mdb" For Binary As #2  
    Dim i As Long  
    Dim deger1 As Byte, deger2 As Byte  
    For i = 1 To FileLen("c:\autonumber1.mdb")  
        Get #1, i, deger1  
        Get #2, i, deger2  
        If deger1 <> deger2 Then Debug.Print i, deger1, deger2  
    Next  
    Close #1  
    Close #2  
End Sub
```

İlk denemelerimde birçok fark olduğunu gördüm. Bunun sebebi veritabanı penceresinde farklı sayfa ve nesnenin seçili olmasıydı. Bu durumda farkı daha iyi algılayabilmek için tablolar sayfasının açık olması ve de kontrol edeceğim tablonun üzerine konumlanmış olmam gerekiyordu. En son denememde sadece iki fark olduğunu gördüm.

Konum (bayt)	Birinci dosyadaki ASCII kod	İkinci dosyadaki ASCII kod
3.587	1	2
106.517	15	16

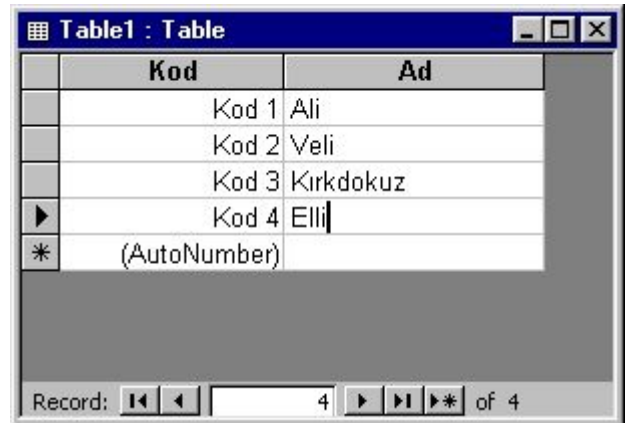
Her iki satır için ayrı ayrı değer aramaya başlamak gerekiyor. Otomatik sayı değerimiz en son 16 olduğuna göre bu değeri oluşturan baytları bulmak gerekiyordu.

106517. bayttan sonraki 4 bayt (LONG tipinde), 16 sayısını yani son kullanılan otomatik sayı değerini veriyordu. Bunun yerine bizim istediğimiz yeni değeri yazmamız yeterli oluyor.

```
Sub Getir()  
    Open "c:\autonumber1.mdb" For Binary As #1  
    Dim deger As Long  
    Get #2, 106517, deger  
    MsgBox deger  
    Close #2  
End Sub  
  
Sub YenisiniYaz()  
    Open "c:\autonumber2.mdb" For Binary As #2  
    Dim deger As Long  
    deger = 0  
    Put #2, 106517, deger  
    Close #2  
End Sub
```

Oraya ne yazılırsa otomatik sayı yeni değerini bu numaradan başlatıyordu. Burayı 0 olarak değiştirdikten sonra üst resimdeki gibi bir görüntü normaldi artık.

Bu işlemden sonra Veritabanı Düzenle (Compact) komutu kullanıldığında Otomatik Sayı normal hale dönecektir. Siz en iyisi bu değeri arkaplan işlemleri için kullanın ya da kullanıcıya herhangi bir kod değeriymiş gibi gösterin. Ben olsam ekran başındaki kişiyi resimde olduğu gibi kandırırdım. Nasıl mı oldu diyorsanız, tablonun tasarım görünümünde otomatik sayı alanının Format (Biçim) özelliğinin değerini tam olarak **"Kod" "#** şeklinde belirtmeniz yeterli.



Kod	Ad
Kod 1	Ali
Kod 2	Veli
Kod 3	Kırkdokuz
Kod 4	Elli
*	(AutoNumber)

Değiştirilmiş otomatik sayı değeri ile ilgili örnek dosyayı yükleyip inceleyebilirsiniz.

[autonumber_mdb.zip](#) (17 KB)