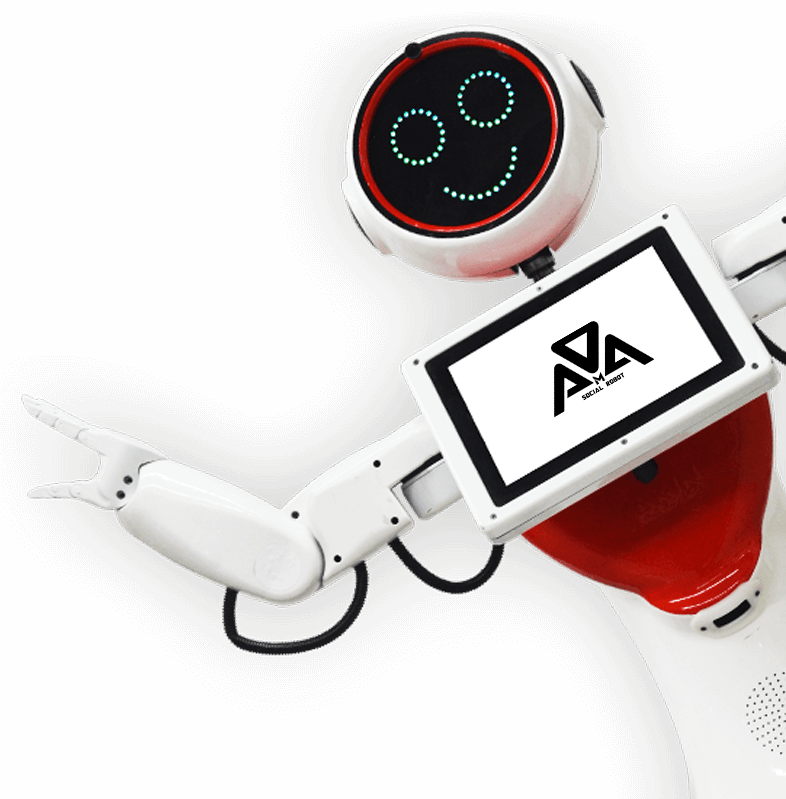
Akınsoft Bilgi Bankasındaki makaleler, Akınsoft Programları okundu, incelendi.

Mini Ada Neler Yapar?

Karşılama Hostesliği, misafirlerinizi karşılayarak bilgi verebilir, yüz tanıma özelliği ile karşılıklı diyalog sağlar.

Danışma Görevlisi, Mini Ada’nın görev alacağı bölgenin krokisi, bu alanlardaki kişi bilgileri robota tanıtılarak ekip arkadaşlarınıza ve ziyaretçilerinize yer yön tarifi, kat ve ofis bilgilendirmesi, eşlik etme ve ön ekranında kurulacak entegrasyon seçenekleri ile birçok işlemi hızlıca yapma şansı sağlar.

İnteraktif Yapı,internetten anlık olarak hava durumu, kur bilgisi, saat-tarih bilgisi vb. alıp, kullanıcıya sesli ya da yazılı olarak aktarabilir, robotumuzun ön ekranı aracılığı ile online görüşmeler sağlayarak, uzak mesafeli kontrol ve görüşmelerinizi yapabilirsiniz.

Eğitim Destekçisi, Eğitim sektöründe Mini Ada farkındalığını kullanarak, sesli diyaloglar kurabilir ve ön ekranında yer alan görseller ile bunu destekleyebilirsiniz. Sesli komutla projeksiyon ve sunumlarınızı Mini Ada’ya yaptırabilirsiniz.

Kontrol Görevlisi, Sahip olduğu barkod okuyucu ve yüz tanıma sistemi ile bilet, barkod, davetiye okuma işlemleri yapabilir, giriş çıkış kontrol ve veri kaydı işlemlerinde kullanabilirsiniz. Bağlı olduğunuz ERP sistem ile entegrasyon sağlayarak, stok kontrolü, sayım işlemleri, süreç takipleri gibi konularda kullanabilirsiniz.

Etkinlik Görevlisi, verilen koreografi doğrultusunda dans ve figürler yaparak; konser, organizasyon, fuar vb. etkinliklerde ilgi çekici şovlarla müşterilerinize veya misafirlerinize eğlenceli saatler yaşatabilirsiniz. Sahip olduğu selfie çekim özelliği ile özel gün ve organizasyonları müşterileriniz ve ziyaretçileriniz ile ölümsüzleştirebilir istenilen elektronik posta adresine yönlendirebilirsiniz.

Mini Ada, Ada GH6, ARAT, UV-C STERİLİZASYON, Servis Robotu tamamlanan satışa sunulan Robot Kol (Endüstriyel Robot Projesi), Akıncı (İnsansı Robot Projesi), PNCR (Tarım Robotu Prototipi), AROS devam eden projeleridir.

Bugün AKINSOFT WOLWOX ERP Programındaki;

* Stok Yönetimi

Depo İşlemleri

Seri No

* CRM (Müşteri İlişkileri Yönetimi)
* Satın Alma Yöntemleri

İrsaliye İşlemleri

Sipariş

* Üretim Yönetimi

MRP II

Basit Üretim

* Finans Yönetimi

Cari İşlemleri

Çek/Senet İşlemleri

Taksit İşlemleri

Banka İşlemleri

Döviz Takip

* Satış Yönetimi

Tüp/Su İşlemleri

Fatura İşlemleri

Teklif

* Servis Yönetimi

Servis İşlemleri

* Kasa

Bölümler incelendi.

**WOLVOX Programı Kurulum Talimatı**

FireBird tabanlı kurulumlar içindir.

1-Server Kurulumu

Kullanılmak istenen Wolvox programının kurulumu yapılır. Program kurulumu tamamlandıktan sonra Kontrol Paneli programı çalıştırılır.

Kontrol Paneli programı çalıştırıldığında öncelikle veri tabanı seçimi yapılmaktadır. Bu alandan Firebird veri tabanı seçilir ve Tamam butonuna basılır. Kullanıcı Girişi ekrana gelecektir. Kullanıcı adında Firebird tabanı kullanıcı adı olan SYSDBA yazılı gelir. Parola alanına program ilk defa kuruluyorsa masterkey yazılır.

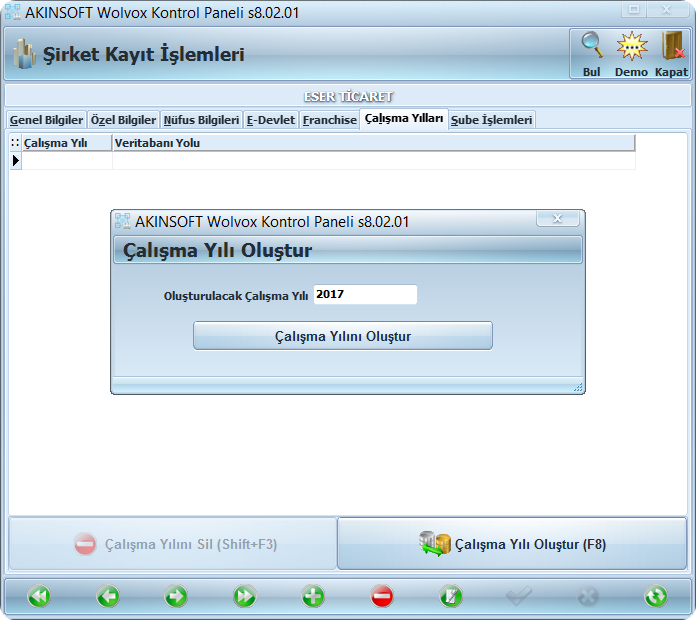


1-1-Şirket Kayıt İşlemleri

Kullanıcı girişi yapıldıktan sonra karşımıza demo şirket kullanıp kullanmak istemediğiniz sorusu gelecektir. Evet denmesi durumunda otomatik olarak DEMO\_WOLVOX kodlu bir şirket kaydı oluşturulacak.

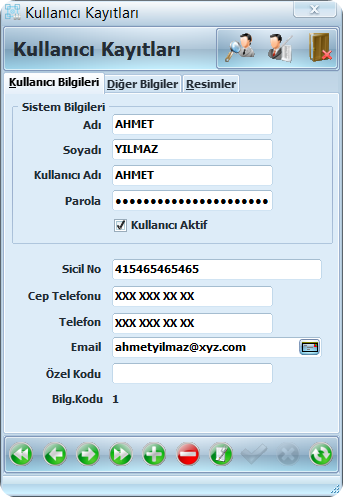
Programa kullanacağımız şirketleri kaydetmek için Yetkili menüsünden Şirket Kayıt İşlemleri alanına oluşturacağımız şirket bilgileri girildikten sonra kaydedilir. Daha sonra Çalışma Yılları sekmesinden Çalışma Yılı Oluşturulur(F8).

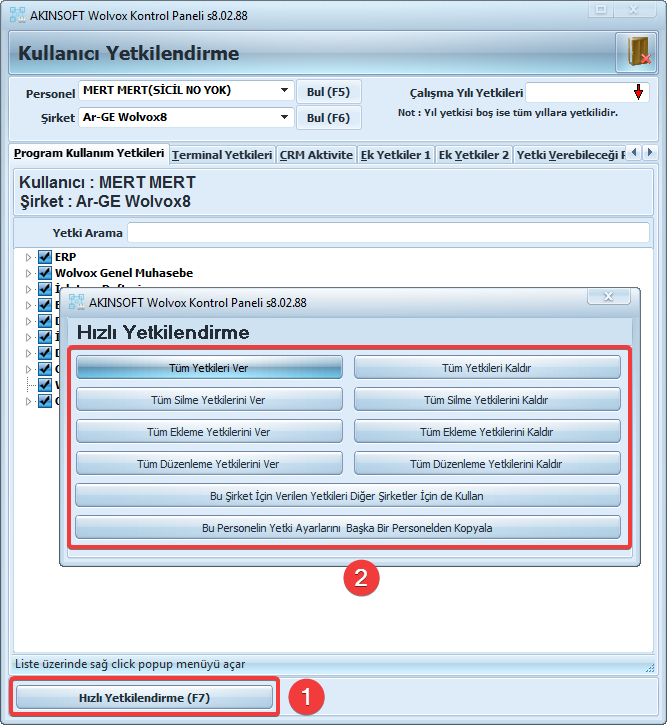




1-2-Kullanıcı Kaydı ve Kullanıcı Yetkilendirme

Programı kullanacak personelleri tanımlamak için ana menüden Kullanıcı Kayıt İşlemleri alanı seçilir. Kullanıcı Kayıt penceresinden programı kullanacak personel bilgileri kaydedildikten sonra Kullanıcı Yetkilendirme(F6) butonuna basılarak kullanıcıya verilecek yetkiler belirlenir.

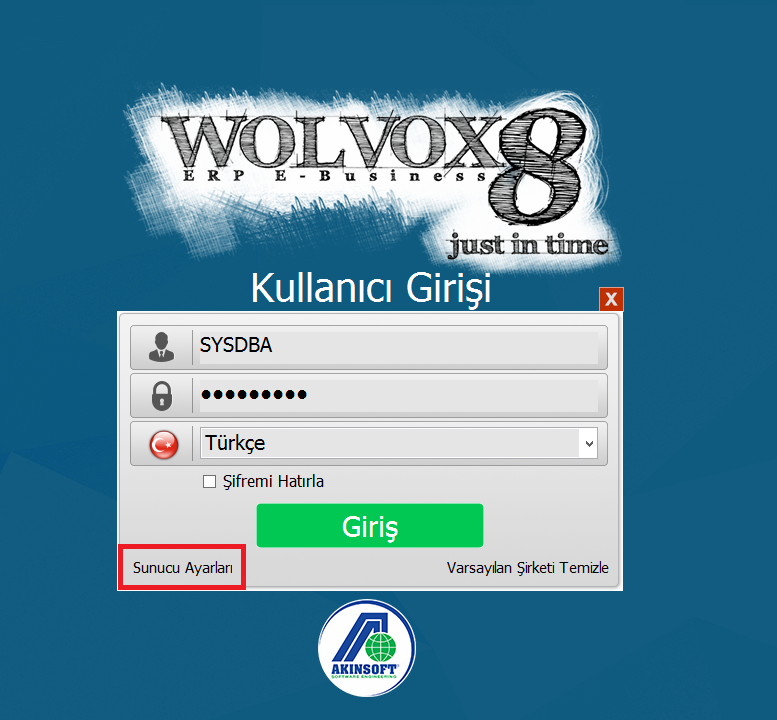




2-Client Kurulumu

2-1-Client Bağlantısı

Kullanıcı Girişi ekranında Sunucu Ayarları alanında Ana Makine İsmi alanına server bilgisayarın IP adresi veya bilgisayar adı yazılır. Kontrol Paneli Portu alanına server da çalışan Kontrol Paneli programının çalışma portu yazılır.





Bugün AKINSOFT WOLVOX ERP Programındaki kısa yollar öğrenildi.

Script Paket İşlemleri (Ctrl+F12)

Cari bul/Hızlı cari arama(F5)

Aktif stok kartı (Ctrl+F8)

Stok kart arama(F6)

Stok fiyatlarını görüntüleme (Ctrl+F7)

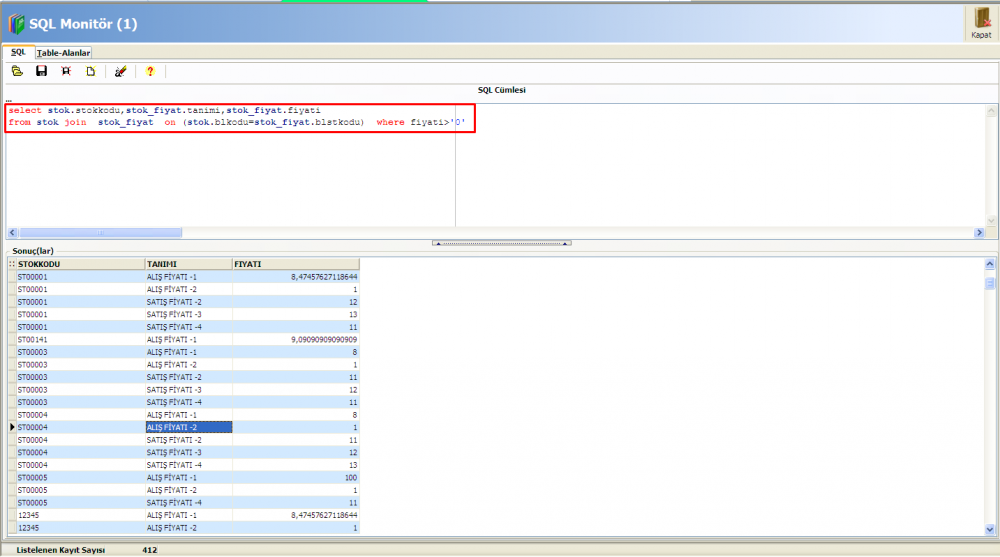
Son yapılan işlemi sil (Shift+F3)

Yeni satıra birden fazla stok ekle (Alt+F6)

Alan adı (Ctrl+Alt+F9)

SQL yapısal sorgulama dili hatırlandı.

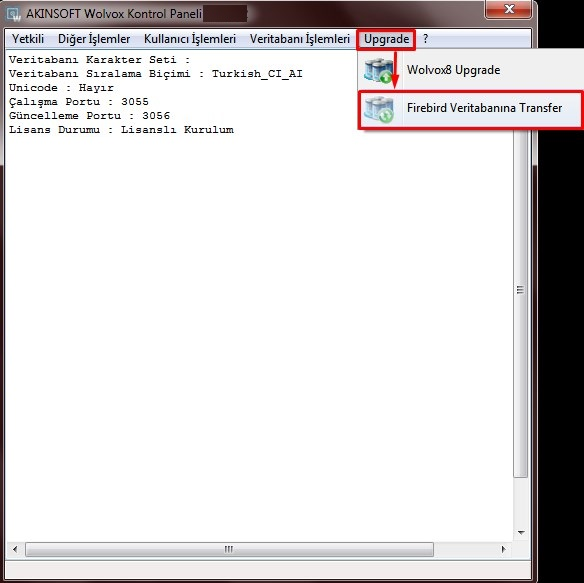
AKINSOFT WOLVOX ERP Programında Filtreleme Seçeneklerinde o sayfada yazılan Sql komutları bulunur. Program içinde bulunan SQL Monitöre kopyalandığında sayfadaki listelemeyi verir.



Bugün FireBird kuruldu.

Wolvox ERP FireBird Sql bağlantısı kuruldu.

AKINSOFT WOLVOX SQL veri tabanından FireBird veri tabanına transfer işlemi gerçekleştirildi.





**FireBird Şifresi Değiştirme**

FireBird veri tabanı ile kullanılan AKINSOFT yazılımları için kurulumda, varsayılan olarak gelen yetkili şifresi “masterkey”dir. Şifre kullanıcı tarafından değiştirilebilir fakat değiştirilmediği sürece masterkey olarak kullanılmaya devam edilir.

Eğer varsayılan şifre değiştirildikten sonra unutulduysa FireBird veri tabanının tamamen kaldırılarak yeniden kurulması gerekir.

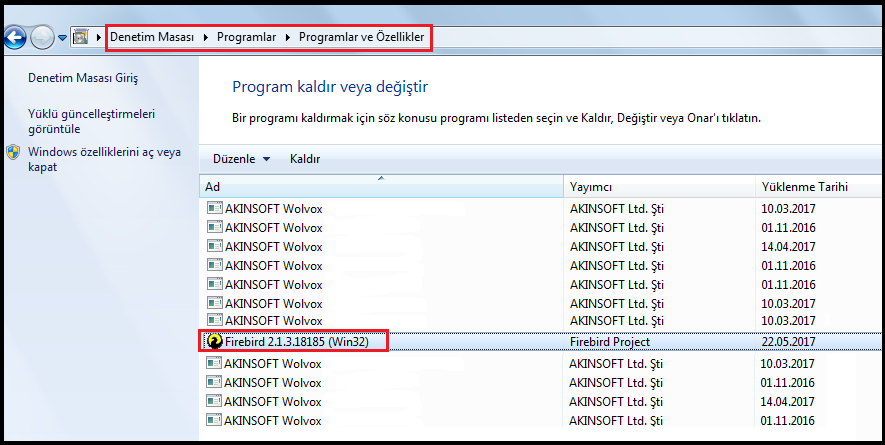
FireBird veri tabanı kaldırılmadan önce versiyonu kontrol edilmelidir. Hangi versiyon kullanılıyorsa aynısının kurulması gerekir.

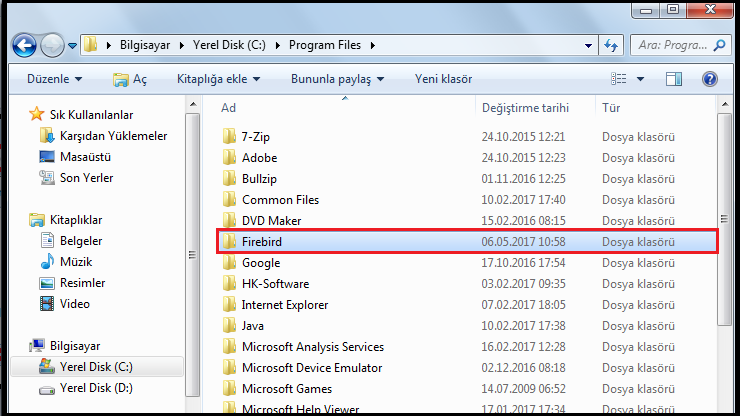
Kontrol paneli dahil olmak üzere tüm AKINSOFT yazılımları kapatılmalıdır.

Başlat🡪Denetim Masası🡪Programlar ve Özellikler menüsünden Firebird veri tabanı bilgisayardan kaldırılmalıdır.

Denetim masasından silme işleminden sonra C dizinindeki Program Files veya Program Dosyaları (x86)’nın içerisinde bulunan FireBird klasörü de silinmelidir.

Firebird veri tabanının tamamen silindiğinden emin olunduktan sonra yeniden kurulum yapılabilir.





Bugün AKINSOFT WOLWOV ERP Programında SQL Monitör üzerinde CARI, CARIHR tabloları birleştirildi VE tarihe göre sıralandı.

Lazarus kuruldu, içerisinde bulunan örnek projeler incelendi. 

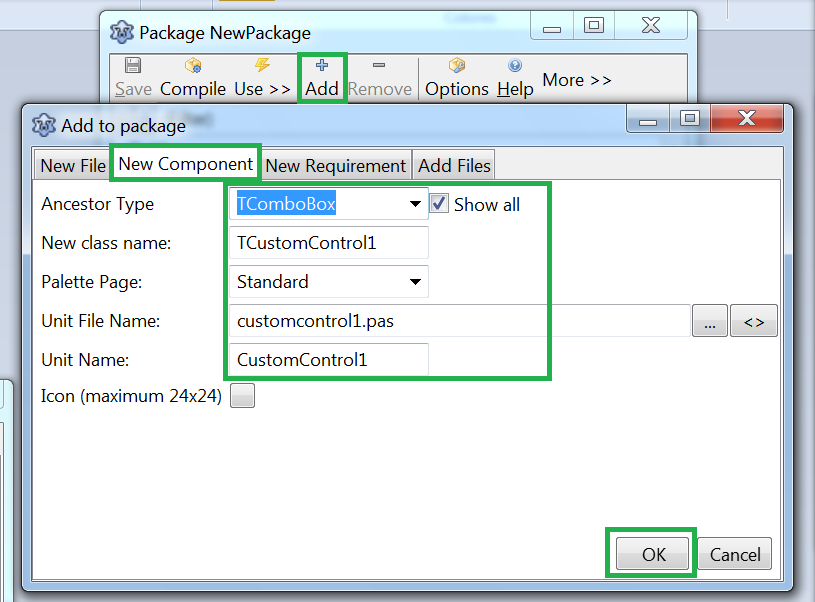
Lazarus açık kaynak kodlu Free Pascal derleyicisi tarafından geliştirilip, desteklenen çoklu platform görsel programlama tümleşik geliştirme ortamıdır. Pascal ve Object Pascal geliştiricileri için hızlı geliştirme ortamı, Delphi benzeri bir ortam sağlamayı amaçlar.

Open Office’te rapor oluşturuldu.

AKINSOFT WOLVOX ERP Programında FireBird Veri tabanında Kopma Sorunu:

Wolvox ERP programında FireBird veri tabanı kopma sorunu nedeniyle karşılaşılan client bağlantısı kopmaları Problemini çözümlemek için FireBird’in yayınladığı sürümlerin kurulumlarının gerçekleştirilmesi gerekmektedir. Kullanmakta olduğumuz FireBird versiyonunu indirip eski veri tabanını kaldırmamız gerekmektedir.

Bugün Lazarus’ta TToolBar,TMemo,SheelListView kullanılarak Not Defteri oluşturuldu.

TSaveDialog(Filter ,Metin Dosyası \*.txt, Tüm Dosyalar \*.\*, XML Dosyası .xml, CSV Dosyası .csv), TOpenDialog(Filter ,Metin Dosyası \*.txt, Tüm Dosyalar \*.\*) ,TImageList, TMainMenu eklendi.

Not defterine farklı kaydet ve düzenlenen(değiştirilen).txt dosyasını aynı yere kaydet.

Open button on click events:

if dlg\_open.Execute then editör.Lines.LoadFormFile(dlg\_open.FileName);

Save button on click events:

if dlg\_save.Execute then editör.Lines.SaveToFile(dlg\_save.FileName);

New button on click events:

Editor.Lines.Clear;

Bugün Lazarus Delphi araştırıldı.

Lazarus üzerinde TEdit, TButton kullanarak hesap makinesi yapıldı.

var

clear\_screen:boolean=false;

temp\_number:real=0;

1 on click events:

if(strtofloat(result\_screen.Text)=0) or (clear\_screen=true) then

begin

result\_screen.Text:=’1’;

clear\_screen.Text:=false;

end

else

begin

result\_screen.Text:=result\_screen.Text+’1’;

end;

end;

1-9 arasındaki tüm on click events yazmak yerine TButton(sender).Caption yazılır.

Addition, extraction, multiplication, division on click events:

Begin

if process=t process.addition then

result\_screen.Text:=floattostr(temp\_number+strtofloat(result\_screen.Text));

AKINSOFT OctoCloud API yapısı incelendi. Java Script -PHP.

Bugün OctoCloud API’den Talend API Tester yardımıyla öncelikle listeleme yapıldı.

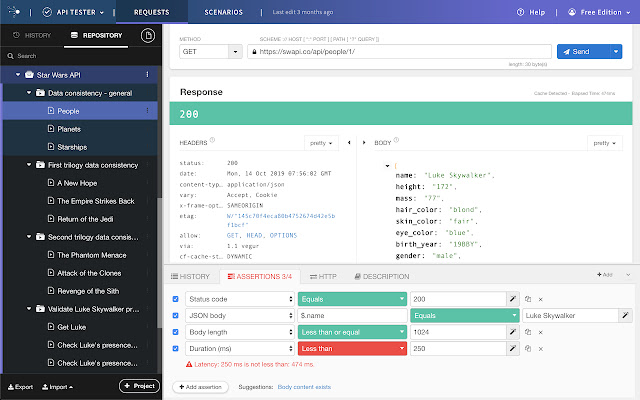
GET api/v1/products

Method GET <http://octocloudapp.akinsoft.com.tr/api/v1/products?page=1>

URL🡪Query Parameters

Headers🡪Authorization: Bearer …

OctoCloud API Dökümandan Kişi/Firma listesi linki alındı GET ile getirildi.



Bugün AKINSOFT WOLVOX ERP Programında Script Paket İşlemlerinden yeni paket oluşturuldu.

Paket Adı: OCTOAPI

Script Adı: RESTAPI

Tasarım üzerinde ilgili component’e sağ tıklanır Değişken atama kodu seçilir.

Atama kodu hafızaya alınınca Genel Script VAR’ın altına kopyalanan atama kodu yapıştırılır.

Bütün componentler için işlem tekrarlanır.

Tasarıma Memo,Button, IdHTTP, IdSSLIOHandlerSocketOpenSSLI eklenir.

Event Script, Button on click

Begin

AsForm Create(‘Tasarım Adı’).Show;

Memo1.Lines.add IdHTTP1.get(‘Token alma URL’)

End;

Programda bir hatayla karşılaşıldı.

Could not load SSL library.

Hata giderilmesi için SSL kütüphaneleri bilgisayarın D klasöründen WOLVOX ERP’ye yüklendi.

Lazarus IDE ,Çevrimiçi Paket yöneticisi REST DataWare seçildi.

WOLVOX ERP Programında T.C.M.B. Kur Download Görsel seçim açıldı.

Componentler incelendi.

Bugün veri tabanı bağlantısı için;

IdSSLIOHandlerSocketOpenSSL,

IdHTTP,

IdServerIOHandler,

IdIOHandlerSocket

Lazarusta forma eklendi.

Case When kullanımı araştırıldı.

AKINSOFT WOLVOX ERP Programından Cari Kodu, İşlem Tutarı, Borç Tutarı, Alacak Tutarı, Bilg. Cari Kodu alanları getirilecek.

Carilerin fatura, nakit, kalan alanlarının görüntüleneceği liste CASEWHEN kullanılarak oluşturuldu.

AKINSOFT WOLVOX ERP SQL Monitöründen (FireBird) CaseWhen kullanılarak Cari Hareketler Raporu yazıldı.

Bugün Stok fiyatlarının listesi özel rapor olarak hazırlandı. Listeden çift tıklayınca stok tanımının açılması sağlandı.

Özel Rapor Hazırlama

AKINSOFT WOLVOX ERP Özel Rapor İşlemlerinde Rapor Düzenleme kısmında yer alan SQL KODU içerisine istenilen, oluşturulan tablonun Sql kodu yazılır veya var olan tablonun özel raporu için filtreleme seçeneklerindeki Sql kodu yapıştırılır.

Script alanındaki Sanal Alan, Renklendirme, Çift Tıklama olayları içerisine istenilen sql kodları yazılır.

Renklendirme: if Field.DataSet.FieldByName(‘Sütun Adı’).As Integer>=Koşul then color:= cl(Renk)

Çift Tıklama:

Yazılacak Script Kodu şu formatta olmalıdır.

Begin if selectedField<> Nil then begin kod buraya yazılmalı end;  
end;

Kod buraya yazılmalı kısmına bu maddelerden biri veya birkaçı yazılabilir.

1.Faturayı Açar:

İf SelectedField.FieldNmae = ‘BLKODU’ Then OpenOldInvoice(DataSet.FieldByName(‘BLKODU’).AsInteger,”,0);

2.Cari Tanım Penceresini açar:

İf SelectedField.FieldName = ‘BLCRKODU’ Then OpenForm(‘YCariTanımlari1,DataSet.FieldByName (‘BLCRKODU’).AsInteger’);

3.Stok Tanım Penceresini Açar:

İf SelectedField.FieldName = ‘BLSTKODU’ Then OpenForm(‘YStokTanımlari1’,DataSet.FieldByName (‘BLSTKODU’).AsInteger);

4.İrsaliyeyi Açar:

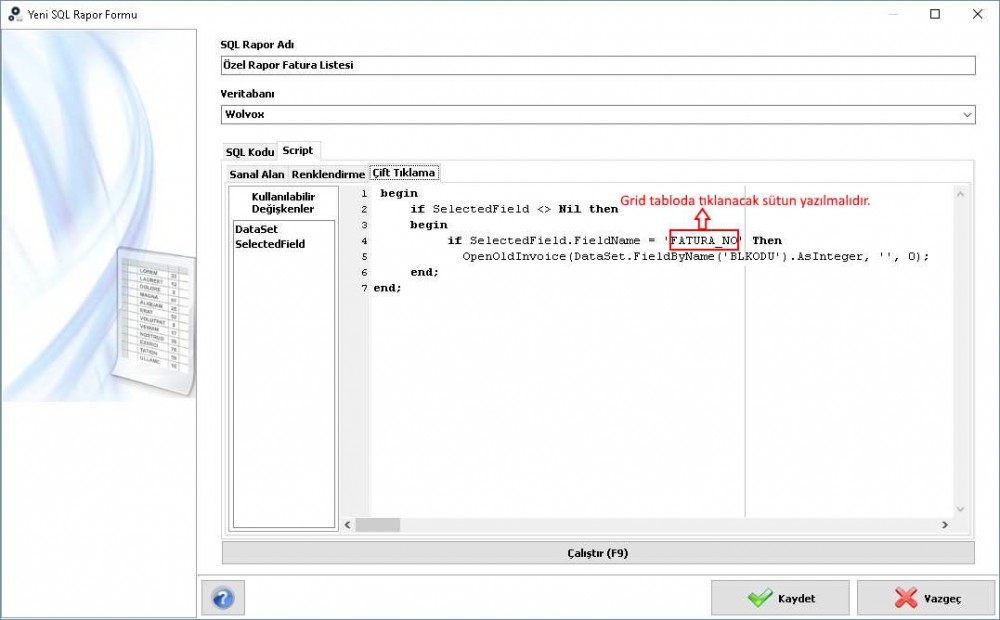
İf SelectedField.FieldName = ‘BLKODU’ Then OpenOldDispatch(DataSet.FieldByName(‘BLKODU’).AsInteger,”,0);

5.Teklifi Açar:

İf SelectedField.FieldName = ‘BLKODU’ Then OpenOldOffer(DataSet.FieldByName(‘BLKODU’).AsInteger,”,0);

6.Siparişi Açar:

İf SelectedField.FieldName = ‘BLKODU’ Then OpenOldOrder(DataSet.FieldByName(‘BLKODU’).AsInteger,”,0);





Bugün Cari Hareket Raporu ve Stok Fiyatları Raporları yazıldı.

STOK ve DEPO miktarlarını karşılaştıran Sql sorgusu yazıldı.

Alış-Satış Faturasından, Alış-Satış KDV tutarlarının farklarını ve alınan ürünün maliyetinde göreceğimiz Sql sorgusu yazıldı, faturası oluşturuldu.

Cari Kodu, Ticari Unvanı, Genel Toplam, Ara Toplam, KDV, Maliyet, Fark sütunlarından oluşan cari listesi oluşturuldu.

Ara Toplam + KDV=Genel Toplam,

Ara Tutar / Miktar=Maliyet

Stok Kalan ve Depo Kalan Unionall ile birleştirildi.

JOIN FATURAHR FH ON (FH.BLFTKODU = F.BLKODU)

Stok=Toplam Ürün adedi,

Kaç depo varsa toplam ürün adedi bölünmüş olabilir.

Örneğin stokta 20 adet bulunan modem 2 depoya 5,15 diye bölünmüş olabilir.

Bu yüzden depolar birleştirilir.

Özel Rapor ve Renklendirme yapıldı.

Bugün Müşteri odaklı şirket tanımı, Müşteri gibi düşünme, CRM (Müşteri İlişkileri Yönetimi) kavramları ve “müşteri her zaman haklıdır” sloganı araştırılıp Müşteri ve Ürün Genel İlişki ile ilgili makale hazırlandı.

**CRM-Müşteri İlişkileri Raporu**

****Cari kartlar sekmesinden kayıtlı cariler listelenebilir. Üzerinde bulunulan carinin aktivitelerini, dahil olduğu kampanyaları, verilen teknik destekleri servise getirdiği ürünleri, kaydedilen fatura, irsaliye, teklif ve sipariş formları görüntülenebilir.

**Müşteri Listesi Kayıt**

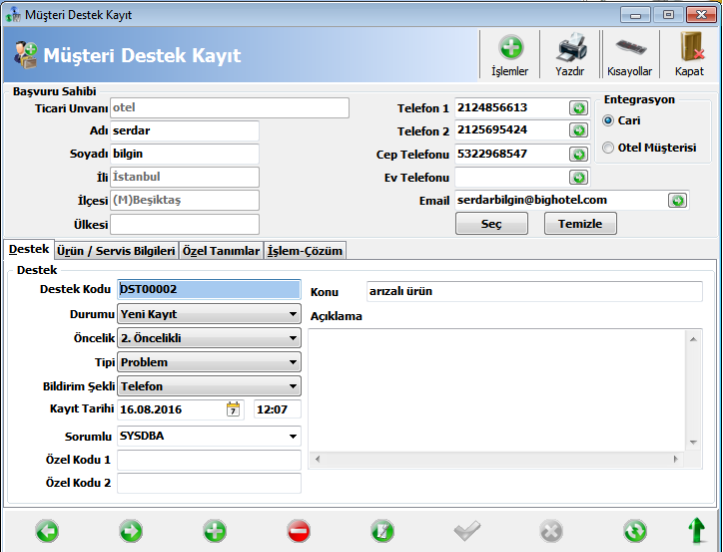
Kayıtlı olan cariler için cari listesi oluşturulabilir. Burada yapılan işlem carilerin kampanya veya satış fırsatı aktarımının kolaylıkla yapılmasına yardımcı olur.

Aynı zamanda müşteri listesi oluşturarak carileri listelere dahil edebiliriz. Liste Adı alanına oluşturmak istediğimiz liste adını yazıp kaydettikten sonra Cari Ekle butonuna basarak listeye dahil olan carilerin seçiminin yapılması gerekmektedir.

**Mevcut Müşteri Listeleri**

Oluşturulan müşteri listelerinin raporlanabileceği alandır. Mevcut müşteri listesine bastığımızda karşımıza gelecek olan listeden seçim yaparak üzerinde düzenleme yapılması gerekiyorsa yaparak kaydedebiliriz.

**Müşteri Destek Kayıt**



**COALESCE ()** fonksiyonu, içerisine yazdığımız değerlerdeki ilk NULL olmayan değeri döndürür ve **N** sayıda parametre alabilir.  
ISNULL () SQL Server a, NVL () Oracle’a özgü iken, COALESCE () fonksiyonu ANSI standart komutu olduğundan, hemen hemen tüm veri tabanlarında ortak kullanılabilir.

Kullanımı şu şekildedir:

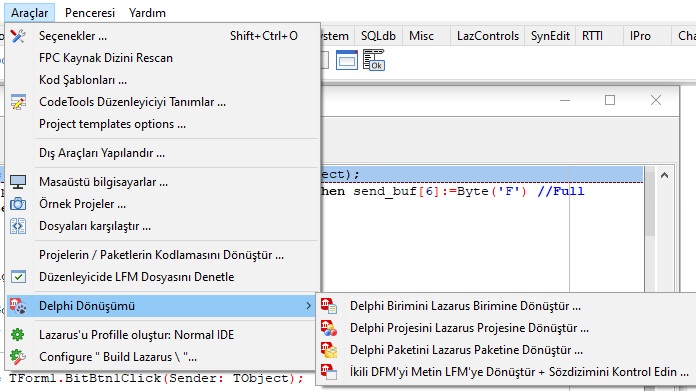
**COALESCE (ifade-1, ifade-2, ifade-3 … , ifade-n )**

Delphi-Rest-Client

Delphi-Lazarus Dönüşümü

Bir Delphi projesi, Delphi paketi ve tek bir birim dosyası Lazarus için dönüştürülebilir. Girişler Lazarus Tools menüsünde bulunabilir:

* Delphi birimini Lazarus birimine çevir.
* Delphi projesini Lazarus projesine çevir.
* Delphi paketini Lazarus paketine çevir.

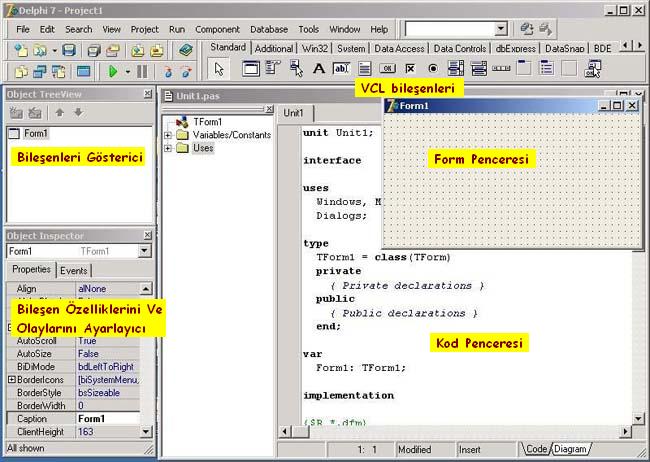


IdHTTP bulunamadı httpUtils hatası alındı.

Lazarus Indy10 SSL IdHttp

Could not Load SSL Library hatası alındı.

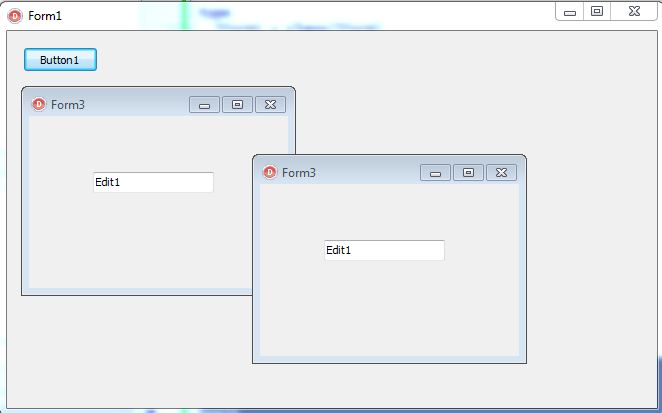
SSL,DLL dosyaları registry olmuyor bu yüzden hata çözülemedi.



Delphi’de birden fazla form create etme

File🡪New🡪Form

Bağlantılı olan formların tıklandığında görüntülenmesi için user kısmına Formadı.Show; yazılır. Unit tanımlanır.



Stok Hareket Girişi, Cari, Stok, Kasa Tanımı, Cari Hareket ve Kasa Hareketleri için Lazarus’ta her biri için yeni form oluşturuldu.

Cari formuna Adresi, İli, İlçesi, Vergi Numarası, Vergi Dairesi, Adı, Soyadı, Ticari Unvanı alanları, Stok formuna Stok Adı, Fiyatı, Birimi alanları eklendi.

Fiyatlar Stok formunun içinde yan sekmesinde Kasa ve Cari Harekette olduğu gibi yer alır.

Form1: PageControl, Cari, Stok, Kasa Button

Cari button double click

Form2: Adı, Soyadı, Ticari Unvanı, Adresi, İli, İlçesi, Vergi Dairesi, Vergi Numarası

DBEdit,DBMemo,DBCombobox.

Stok button double click

Form3: PageControl (Fiyat, Birimi), Stok Adı, DBGrid, DBEdit.

Kasa button double click

Form4: FormHeader(Kasa Tanımları), DBGrid, GroupBox

Form5: Kasa Bilgileri, Kasa Adı, Kayıt Tarihi, Muhasebe Kodu, DBEdit, DBDateEdit, DBCheckBox, DBButtonEdit, BitBtnAs.

Form6: Terminal Yetkileri, Kayıt-Yetkilendirme, terminaller, Panel, Listbox, Button.

Lazarus’ta Giriş, Cari Hareket, Cari Tanımları, Stok Tanımları, Kasa Tanımları, Stok Hareket, Kod Değiştirme (Cari), Stok ve İstatistik Fiyatları, Kasa Grup Tanımları, Hızlı Cari Arama, Stok Kart Arama, Kod Değiştirme (Kasa) 12 form oluşturuldu birbirine bağlandı.

Bağlanmak için; bağlanmak istenilen yere onclick

Formadı.Show; ve user kısmında form adı tanımları.

VeriTabanı bağlantısı için ZeosDB 0.2.2 yüklendi.

SQL Connector, Z Table, IB Connection, Data Source, Z Connection, FB Admin eklendi.

SQL Connector;

CharSet: WIN 1254

Connected: True

Connector Type: Firbird

Databasename: D:AKINSOFT WOLVOX 8 Database\_FB 001

Hostname: 127.0.0.1

Keep Connection: True

Login Prompt: True

Password: 1

USERNAME: SYSDBA

Z Table;

Connection: ZConnection

TableName: Adres

IB Connection;

CharSet: WIN 1254

Connected: True

Connector Type: Firbird

Databasename: D:AKINSOFT WOLVOX 8 Database\_FB 001

Hostname: 127.0.0.1

Keep Connection: True

Login Prompt: True

Password: 1

Port: 3050

FB Admin;

Host: 127.0.0.1

Password: 1

Port: 3050

Protocol: firebird-2.5

User: SYSDBA

Data Source;

Data Set: ZTable

Z Connection;

Protocol: firebird-2.5

SQL Hour Glass: True

User Meta Data: True

“C: \Windows \System32 \GDS32.DLL found but could not be loaded.” Hatası alındı.

C dizini altında System32’ye GDS32.DLL dosyası atıldı.

Sorun çözüldü, veri tabanı bağlandı.

Hata Ayıklayıcı istisna Bildirimi

“can not load default Firebird clients(“fbclient.dll” or “gds32.dll”)”

Proje sorunsuz derleniyor ancak çalışmıyor.

Z Connection;

ClientCodePage: UTF8

Connected: True

ControlsCodePage: cCP\_UTF8

Database: D:AKINSOFT WOLVOX 8 Database\_FB 001

DesignConnection: True

HostName: 127.0.0.1

LibraryLocation: C🡪İndirilenler🡪 Firebird64🡪 fbembed.dll

Password: 1

Protocol: firebird-2.5

User: SYSDBA

Tüm veri tabanı bağlantıları yapıldı.

DataSource ve ZTable eklendi. DataSource ZTable’a bağlanır.ZTable ZConnection’a bağlanır. TableName kullanılacak tablo adı seçilir.

Database bağlantısı başarılı ancak Cari Tanımlarında TDB Navigator çalışırken diğer Unitlerde çalışmıyor.

!STACK Overflow hatası.

Stack-overflow. Sorunları genellikle eksik, silinmiş veya LXFDVD118 yazılımının orijinal yüklenme konumundan yanlışlıkla taşınmış bir dosyadan kaynaklanır. Bu dosya sorunlarının büyük bir kısmı, AS dosyanızın en son sürümünün indirilmesi ve kurulmasıyla çözülebilir.

ZQuery ile (SQL Sorgu) tablolar birleştirildi.

Giriş🡪Cari Hareketler🡪Bul🡪Seçilen cari Cari Hareketlere işlendi.

DBNavigatorler (ekle, sil, güncelle, ileri, ilk, son) sorunsuz çalıştırıldı.

Cari Hareketlerinde yeni eklenen ya da güncellenen cari kaydı Kasa Raporu DBGrid’de görüntülendi.

Seçilen Kasa Raporu Hareketi aça tıklandığında Kasa Gelir Girişi formu açılır.

Hareket bilgileri ekranda doğru bir şekilde görüntülenir.

Kaydı aça tıklandığında Kasa Raporu, Cari Hareketlerde görüntülendi.

Sil, Temizle button on click events: Temizlenecek alanadı.Clear;

Filtrele için ActionList’ten Filtrele Action oluşturuldu.

Filtrele butonu için action eklendi.

Artık Projenin son aşamasına gelindi.

Giriş

Stok Tanımları

Kasa Tanımları

Cari Hareket

Stok Hareket

Kod Değiştirme -Cari-

Stok ve İstatistik Fiyatları

Kasa Grup Tanımları

Hızlı Cari Arama

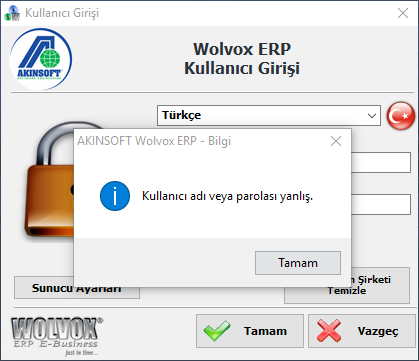
Stok Kart Arama

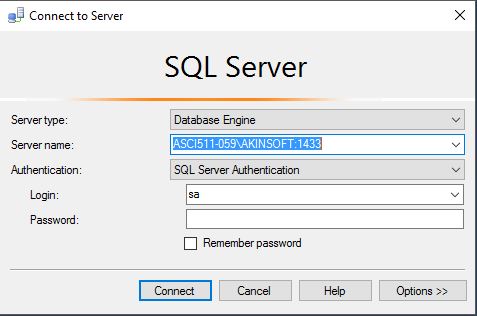
Kod Değiştirme -Kasa-

Kasa Raporu

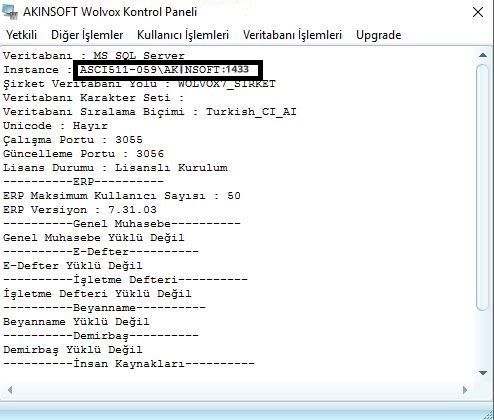
Kasa Gelir Girişi

Toplam 14 Unit oluşturuldu eksik unitler tamamlandı.

MSSQL Veri Tabanında Client olarak kullandığım Wolvox ERP Programında Kullanıcı adı veya parolası yanlış uyarısı alındı.

Sql Server tabanı çalışan Wolvox ERP programımızda client bilgisayarlarda programı açmaya çalıştığında uyarı ile karşılaşılıyorsa çözümü için MSSQL Server Management Studio da SERVER NAME alanına instance isminin sonuna fotoğraftaki gibi: port numarası yazarak çalıştırılmalı.

Daha sonra kontrol panelini yine aynı şekilde fotoğrafta görüldüğü gibi yine instance isminin sonuna Sql Management Studio da yaptığımız şekilde port numarasını yazarak çalıştırılması gerekmektedir.

Wolvox ERP Programını kapatıp açtığınız taktirde sorunsuz bir şekilde çalıştırılabilir.

Wolvox ERP Programında Stok Bazlı Servis ve Stok Bazlı Alınan Verilen özel teklif raporu yazıldı.

Stok Bazlı Alınan Teklif Raporu SQL Kodu

SELECT \* FROM ( SELECT THR.BLSTKODU, THR.STOKKODU, THR.BIRIMI, SUM(SAYIYUVARLA(MIKTARI,3,0)) TPL\_MIKTAR, SUM(SAYIYUVARLA(KPB\_TOPLAM\_TUTAR,2,0)) KPB\_TOPTUTAR, SUM(SAYIYUVARLA(KPB\_KDVLI\_TUTAR,2,0)) KPB\_KDVLITOPTUTAR FROM TEKLIF T JOIN TEKLIFHR THR ON (THR.BLTKKODU = T.BLKODU) LEFT JOIN STOK S ON (S.BLKODU = THR.BLSTKODU) WHERE T.BLKODU>0 [THR.BLSTKODU] [THR.STOKKODU] AND T.SILINDI = ’0’ AND T.TEKLIF\_TURU IN (2,4) GROUP BY THR.BLSTKODU, THR.BIRIMI, THR.STOKKODU ) SNC ORDER BY SNC.STOKKODU

Script🡪Renklendirme

BEGIN if Field.DataSet.FieldByName(‘BIRIMI’).AsString = ‘PORSİYON’ then begin Background:= clGREEN; AFont.Color:=clWHITE; end; if Field.DataSet.FieldByName(‘BIRIMI’).AsString = ‘GR’ then begin Background:= clBLUE; AFont.Color:=clWHITE; end; if Field.DataSet.FieldByName(‘KPB\_KDVLITOPTUTAR).AsInteger = 0 then begin Background:= clSilver; AFont.Color:=clWhite; end; END;

Stok Bazlı Servis Raporu

SELECT \* FROM ( SELECT SVHR.BLSTKODU, SVHR.STOKKODU, SVHR.BIRIMI, SUM(SAYIYUVARLA(MIKTARI,3,0)) TPL\_MIKTAR, SUM(SAYIYUVARLA(KPB\_TOPLAM\_TUTAR,2,0)) KPB\_TOPTUTAR, SUM(SAYIYUVARLA(KPB\_KDVLI\_TUTAR,2,0)) KPB\_KDVLITOPTUTAR, FROM SERVIS SV JOIN SERVISHR SVHR ON (SVHR.BLMASKODU = SV.BLKODU) LEFT JOIN STOK S ON (S.BLKODU = SVHR.BLSTKODU) WHERE SV.BLKODU>0 [SVHR.BLSTKODU] [SVHR.STOKKODU] AND SV.SILINDI= ’0’ GROUP BY SVHR.BLSTKODU, SVHR.BIRIMI, SVHR.STOKKODU ) SNC ORDER BY SNC.STOKKODU

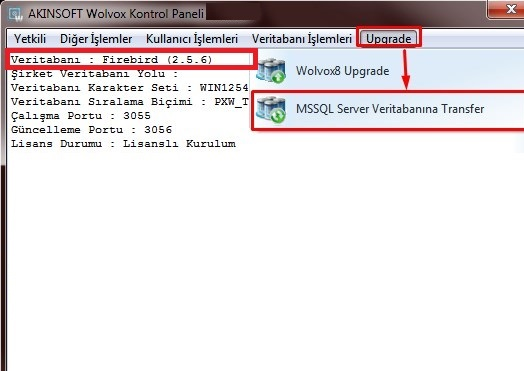
Script🡪Renklendirme

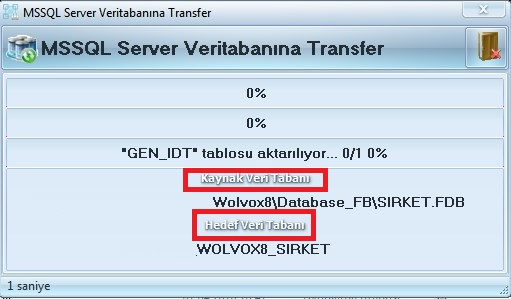
BEGIN if Field.DataSet.FieldByName(‘KPB\_KDVLITOPTUTAR).AsInteger <= 0 then begin Background:=TColor($0080FF); AFont.Color:=clWHITE; end; if Field.DataSet.FieldByName(‘KPB\_KDVLITOPTUTAR). AsInteger > 0 then begin Background:= clSilver; AFont.Color:=clWhite; end; END;

Microsoft SQL Server Management Studio 18 kuruldu.

Burada iki şirket oluşturulacak ve iki şirket arasında cari ve stok bağlantısı kurulacak. Örneğin birinci şirkete eklenen cari otomatik olarak ikinci şirkete de eklenecek.

Wolvox ERP veri tabanı SQL Server’a aktarıldı.





Eğer iki veri tabanı aynı sunucu üzerinde ise veri tabanı tam yolu değilse link oluşturulmalı.

Trigger,

Procedure,

Views,

External Resources,

Functions,

Database Trigger,

Indexes,

Graph Table,

File Table,

External Table araştırıldı,örnekleriyle birlikte bir makale hazırlandı.

SOL Server SQL Server

Database Database

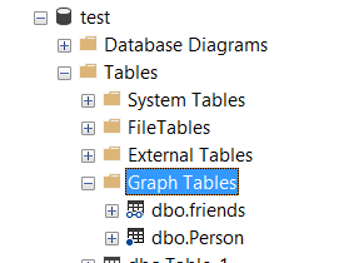
Tables External Resources

System Tables External

File Tables Data Sources

External Tables External

Graph Tables File Formats



SQL Server’a ikinci şirket eklendi.

2 farklı database;

1. WOLVOX8\_02\_2022\_WOLVOX
2. WOLVOX8\_1228\_2020\_WOLVOX

Cari tabloları birleştirildi.

WOLVOX8\_02\_2022\_WOLVOX üzerinde yapılan kayıtlar otomatik olarak 2 database üzerindeki CARI tablosuna eklendi.

UPDATE için çalışmalar yapıldı.

AKINSOFT WOLVOX ERP Programından 2. Şirket açıldı. Cari Tanımları üzerinden Cari kaydı yapıldı.

ERP’de kaydı yapılan cari bilgileri SQL Server’da 1 ve 2. Database üzerinde CARI isimli iki tabloya da otomatik olarak eklendi.

Create TRIGGER TrCariDB1 ON dbo.CARI AFTER INSERT AS BEGIN DECLARE @Cblkodu int, @Cckodu varchar(100), @Cticariunvani varchar(100), @Csil smallint, @Csubekodu varchar(30) SELECT @Cblkodu = INSERT.[BLKODU] , @Cckodu = INSERT.[CARIKODU] @Cticariunvani = INSERT.[TICARI\_UNVANI] , @Csil = INSERT.[SILINDI], @Csubekodu = INSERT.[SUBE\_KODU] FROM INSERTED

INSERT INTO [WOLVOX8\_1228\_2020\_WOLVOX ].[dbo].CARI([BLKODU], [CARIKODU],[TICARI\_UNVANI],[SILINDI],[SUBE\_KODU]) VALUES(@Cblkodu, @Cckodu, @Cticariunvani, @Csil, @Csubekodu) END

WOLVOX8\_1228\_2020\_WOLVOX (2. Database) CARI tablosunda kayıtlar olduğu için WOLVOX8\_02\_2022\_WOLVOX (1. Database CARI tablosu boş) BLKODU 69’a kadar cari kayıtlı bu yüzden WOLVOX ERP üzerinden 2.şirkette yapılan cari kayıtları -boş olduğu için- BLKODU 1’den başlar BLKODU elle değiştirilemez ve 2. Database’de tablo yeri dolu olduğundan kayıt yapılamaz.

Düzeltmek için;

Trigger içindeki INSERT INTO VALUES(@Cblkodu) yerine NEXT VALUE FOR CARI\_GEN yazılmalı. Bu işlem arka planda generatör tetikler.

SQL Server Managment

2.Şirket içinde insert, update triggerları yazıldı.

Wolvox ERP üzerinden insert ve update işlemi yapılan şirketten CARI kayıtları otomatik olarak diğer şirkette de insert,update olur.

Create TRIGGER TrGuncelle ON dbo.CARI AFTER INSERT AS BEGIN DECLARE @Cblkodu int, @Cckodu varchar(100), @Cticariunvani varchar(100), @Cadı varchar(100), @Csoyadı varchar(100), @Csil smallint, @Csubekodu varchar(30), @Caktif smallint, @Ctelno varchar(20), @Culke varchar(20) SELECT @Cblkodu = INSERT.[BLKODU] , @Cckodu = INSERT.[CARIKODU] @Cticariunvani = INSERT.[TICARI\_UNVANI] , @Cadı = INSERT.[ADI] @Csoyadı = INSERT.[SOYADI], @Csil = INSERT.[SILINDI], @Csubekodu = INSERT.[SUBE\_KODU], @Caktif = INSERT.[AKTIF], @Ctelno = INSERT.[TEL1], @Culke) = INSERT.[ULKESI\_1], FROM INSERTED

UPDATE [WOLVOX8\_1228\_2020\_WOLVOX ].[dbo].CARI SET [BLKODU] = @Cblkodu, [TICARI\_UNVANI] = @Cticariunvani, [SILINDI] = @Csil, [SUBE\_KODU] = @Csubekodu, [ADI] = @Cadı, [SOYADI] = @Csoyadı, [AKTIF] = @Caktif, [TEL1] = @Ctelno, [ULKESI\_1] = @Culke WHERE [CARIKODU] = @Cckodu END

Update işleminden sonra ERP’de Cari oluşturulurken BLKODU kaldığı yerden devam etmiyor.

BLKODU=1’e kaydoluyor.

Programda bir hatayla karşılaşıldı.

[AnyDAC] [Phys] [ODBC] [sqlncli11.dll] SQL\_NO\_DATA hatası veriyor ancak kayıtlı carileri üzerinde update sorunsuz çalışıyor.

SQL Server’da Cari eklenmeye çalışıldığında

Error Source.Net SqlClient Data Provider.

Error Message: Could not find stored procedure ‘InsertIntoCARI’. The statement has been terminated. Correct the errors and retry or pass ESC to cancel the changes(s).

Insertcari triggerı pasifleştirildi-silindi-.

BLKODU=1’den başlama hatası;

SQL Server’da WOLVOX8\_1228\_2020\_WOLVOX databese’in altında,

Programmability 🡪 Dbo.CARI\_GEN sağ tık 🡪 Properties

Curren Value, Restart sequence eşitle.

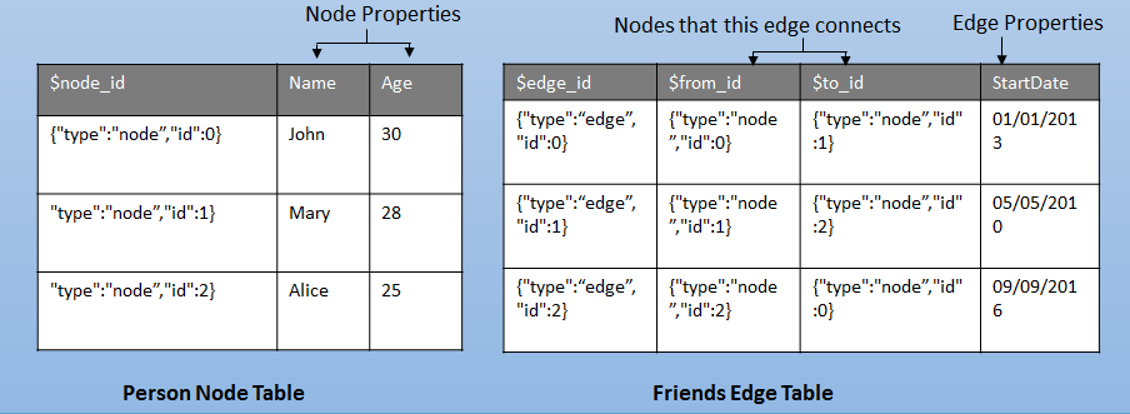
WOLVOX8\_02\_2022\_WOLVOX8 içinde aynısı yapılır.

Hata çözüldü.

Karşılıklı databaseler üzerindeki karşılıklı triggerlar silindi.

Create, Insert ve Update tek taraflı.

WOLVOX8\_02\_2022\_WOLVOX8 🡪 WOLVOX8\_1228\_2020\_WOLVOX8

Graph Table, tablolar arası ilişkilerin yoğun kullanıldığı veri tabanlarını analiz etmek istediğinizde ya da hiyerarşik veri ile uğraştığınızda Graph Database’i kullanmanız işimizi kolaylaştırır.

Node ve edge tabloları birleştirildi.

Node 🡪 STOKK ve DEPOO adında iki tablo oluşturuldu.

Tablolar ilişkilendirildi.

STOKK ve DEPOO düğümleri, edgetable(kenar) ile ortak sorgu yapıldı.

Cretae table STOKK( [BLKODU] int, [DEPO\_ADI] varchar(50), [STOKKODU] varchar(50) ) as node

Insert into STOKK values (1, ‘Kalem’, ‘ST00001’), (2, ‘Defter’, ‘ST00002’)

Create table DEPOO( [BLKODU] int, [DEPO\_ADI] varchar(50) ) as node

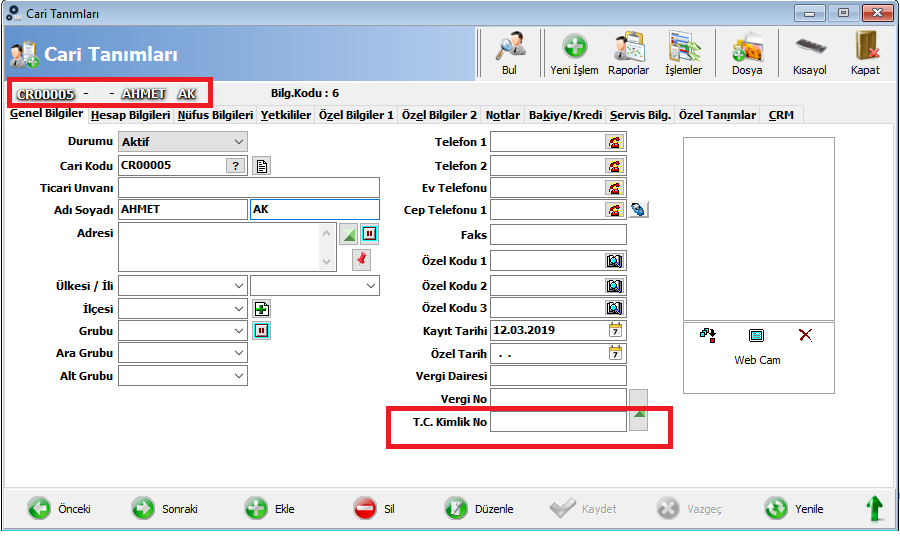
Insert into DEPOO values (1, ‘MERKEZ1’), (2, ‘ŞUBE-1’)

Create table edgetable( [BLKODU] int, [STOK\_ADI] varchar(50 [DEPO\_ADI] varchar(50) ) as edge

Create unique index ix\_edgetable\_from\_to on edgetable ($from\_id, $to\_id)

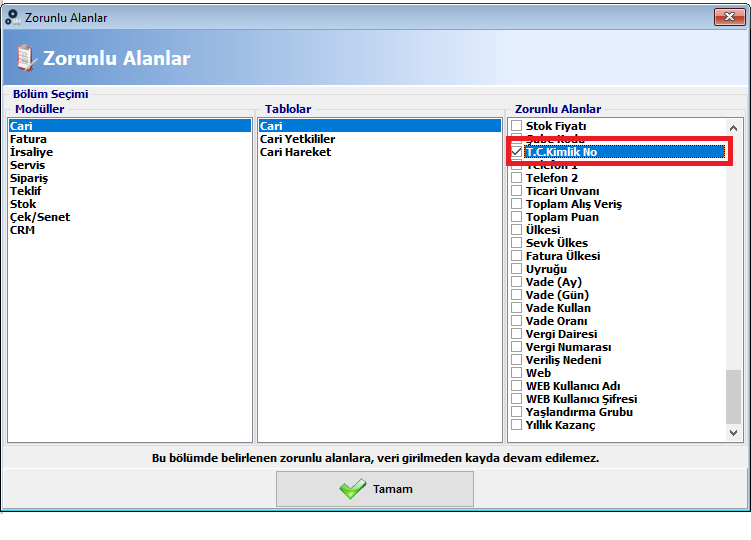
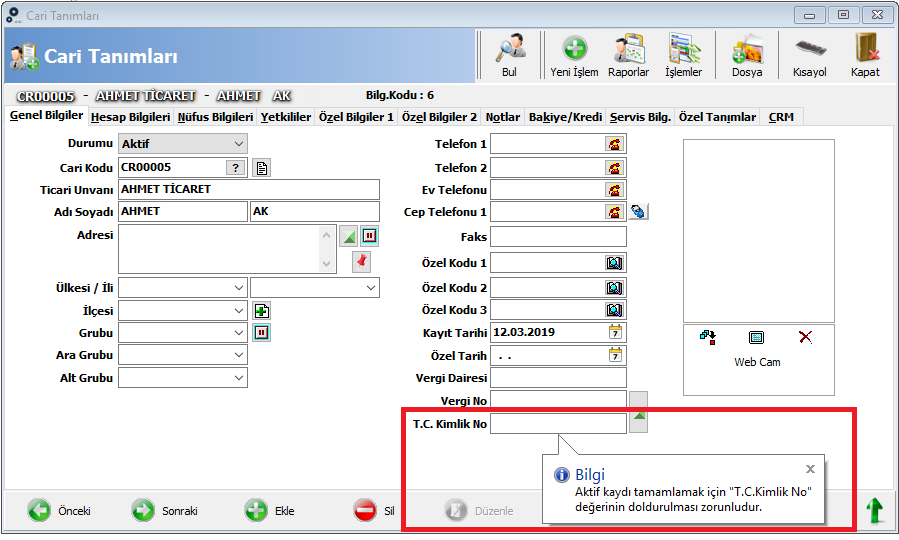
Select STOKK.BLKODU, STOKK.STOK\_ADI, DEPOO.DEPO\_ADI from DEPOO, edgetable, STOKK where match (DEPOO-(edgetable)- >STOKK)

**Cari Kaydederken Bir Alanın Zorunlu Olması**

Öncelikle bir cari kaydederken zorunlu alanlarda bölümünde T.C. kimlik numarası zorunlu alan olarak seçilmemiş ise cari tanımı yaparken T.C. kimlik numarasını yazmadan da kayıt işlemini gerçekleştirir.

Örneğin;

Eğer cari tanımı kaydederken T.C. kimlik numarası alanı doldurulmadan kayıt yapılmasını istersek yapılması gereken işlemler şunlardır;

* Yetkili🡪 Tanımlar🡪Zorunlu Alanlar kısmında Modüller bölümünde Cari seçeneği Tablo bölümünde Cari seçeneği Zorunlu Alanlar Kısmında T.C. Kimlik No seçeneği işaretlenmesi gerekiyor.
* Bu alana verilmiş olan zorunluluktan sonra yapılan cari tanımı işlemi esnasında T.C. Kimlik numarasının boş geçilmesine izin verilmez.

Bugün CARI update, insert işlemleri tamamlandı. Teslim edildi.

Makale (SQL) tamamlandı. Teslim edildi.

AKINSOFT WOLVOX8 ERP Programında

Yetkili 🡪Tanımlar🡪Script Paket İşlemleri

Yeni paket oluşturuldu.

Herhangi bir tabloda görsel seçim ile incelendi.

Bileşenler (events) kısmındaki tüm bileşenlerin özellikleri, kullanım yerleri, RunTime makalesi yazılacak.

Script Paket işlemleri açılır(ctrl+F12)

Run Time

Script 🡪 On loaded 🡪 Event Script 🡪 İşlem Öncesi

Var lb: TLabel; lb2:TLabel; begin lb := TLabel.Create(Self); lb.parent := FR\_CariFiltreleGenel1; lb.Left := 38 lb.Height := 25 lb.Top := 452 lb.Width := 124 lb.Caption :=’T.C. Kimlik No.’

lb2 := TLabel.Create(Self) lb2.parent := FR\_CariFiltreleGenel1; lb2.Left := 65 lb2.Height := 25 lb2.Top := 475 lb2.Width := 124 lb2.Caption := ’Vergi No.’

Script 🡪 OnStart 🡪 Event Script 🡪 İşlem Öncesi

Var ed:TAsFiltreEdit; ed1:TAsFiltreEdit; cb:TAsFiltreMultiEdit; cxb:TsCheckBox;

begin ed := TAsFiltreEdit.Create(Self); ed.parent := FR\_CariFiltreleGenel1; ed.Left := 116 ed.Height := 20 ed.Top := 448 ed.Width := 125 ed.f\_fieldname := ‘TC\_KIMLIK\_NO’; ed.f \_collate := ‘Turkish\_CI\_AI’ ed.f\_filtertype := ftLike1

ed1 := TAsFiltreEdit.Create(Self); ed1.parent := FR\_CariFiltreleGenel1; ed1.Left := 116 ed1.Height := 20 ed1.Top := 472 ed1.Width := 125 ed1.f\_fieldname := ‘VERGI\_NO’; ed1.f \_collate := ‘Turkish\_CI\_AI’ ed1.f\_filtertype := ftLike1

cb := TAsFiltreMultiEdit.Create(Self); cb.parent := FR\_CariFiltreleGenel1; cb.Left := 116 cb.Height := 20 cb.Top := 496 cb.Width := 125

cxb := TCheckBox.Create(Self); cxb.parent := FR\_CariFiltreleGenel1; cxb.Left := 95 cxb.Height := 20 cxb.Top := 496 cxb.Width := 125

end;

Script 🡪 OnTimer 🡪 Event Script 🡪 İşlem Öncesi

Var bt:TsBitBtnAS; bbt:TsBitBtnAS;

Begin bt := TsBitBtnAS.Create(Self); bt.parent := FR\_CariFiltreleGenel1; bt.Left := 4 bt.Height := 25 bt.Top := 520 bt.Width := 106 bt.Action := FiltreleA bt.Caption := ‘Filtrele1’ bt.Visible := True; bt.Images := ImageListNav16; bt.ImageIndex := 10

bbt := TsBitBtnAS.Create(Self); bbt.parent := FR\_CariFiltreleGenel1; bbt.Left := 115 bbt.Height := 25 bbt.Top := 520 bbt.Width := 126 bbt.Action := TemizleA bbt.Caption:=‘Temizle1’ bbt.Images := ImageListNav16; bbt.ImageIndex := 11

Bugün kalan combobox, checkbox sıralama doğru filtreleme yapıldı.

TAsFiltreRadioGroup🡪Items, Visiable, Caption,

TASRXGRID🡪f\_fieldorigin, datasource,

MenuItem,

Toolbar yapıldı.

Script 🡪 OnStart 🡪 Event Script 🡪 İşlem Öncesi

Var ed:TAsFiltreEdit; ed1:TAsFiltreEdit; cb:TAsFiltreMultiEdit; cxb:TsCheckBox; rdio:TAsFiltreRadioGroup; tbar:TToolBar; grid:TASRXDBGrid;

Begin rdio := TAsFiltreRadioGroup.Create(Self); rdio.parent := FR\_CariFiltreleGenel1; rdio.Left := 4 rdio.Height := 80 rdio.Top := 544 rdio.Width := 236 rdio.Visiable := True rdio.Items.Add (‘Evet’); rdio.Items.Add (‘Hayır’); rdio.Items.Add(‘Tümü’); rdio.Caption:=‘RiskliMüşteriler’ rdio.f\_sqlvalues.Add(‘RISKLI\_MUSTERI=1’); rdio.f\_sqlvalues.Add(‘(RISKLI\_MUSTERI=0 OR RISKLI\_MUSTERI IS NULL)’)

grid := TASRXDBGrid.Create(Self); grid.Parent := FR\_CariFiltreleGenel1; grid.Left := 13 grid.Height := 200 grid.Top := 200 grid.Width := 200 grid.f\_fieldorigin := ‘CARI.BLKODU’; grid.DataSource := Cari\_S grid.as\_sorter.FilterAction := FiltreleA; grid.as\_sorter.SortCombo := ComboSIRA5; grid.as\_sorter.DescCheck :=CheckTers5;

Cari Tanımları🡪Yetkililer🡪Girilen Yetkili Cari Yetkililer listesine kaydolur.

Stok Tanımlar🡪Birim/Barkod🡪Ek Barkodlar

Cari Yetkililer Listesi gibi Stok Ek Barkod Listesi yapılacak.

Grid ’de Stok kodu, Stok Adı, Barkod, Ek Barkod, Stok Tanımlar Listesinden Filtre1 filtreleme alanı yapılacak. Sıralama alanı olacak.

Cari Tanımlar Listesi, GNL Paket’e bakılabilir.

Script Component, Not defterine bakılabilir.

Oluşturulan tasarımda,

TAsAdQuery, TAsRXGride,

Select \* from STOK, select \* from STOK\_BARKOD bağlandı.

Ek barkod girilince Mouse’la gezildiğinde stok gridinde gösterildi.TAsFiltreEdit, TAsFiltreBtnEdit, TAsFiltreEdit, TWERPGrupFiltreCombo, TWERPAraFiltreCombo, TAsRadioGroup, TAsFiltreCheckBox, TsBitBtnAs, ImageList16, TsStatusBar, TsSpeedButton, TsLabel, TAsLongCombobox, TsCheckBox, TJvNetscopeSplitter, TAsFormHeader, sAlphaImageList1, TASRXDBGRID kullanarak Stok Ek Barkod Listesi oluşturuldu.

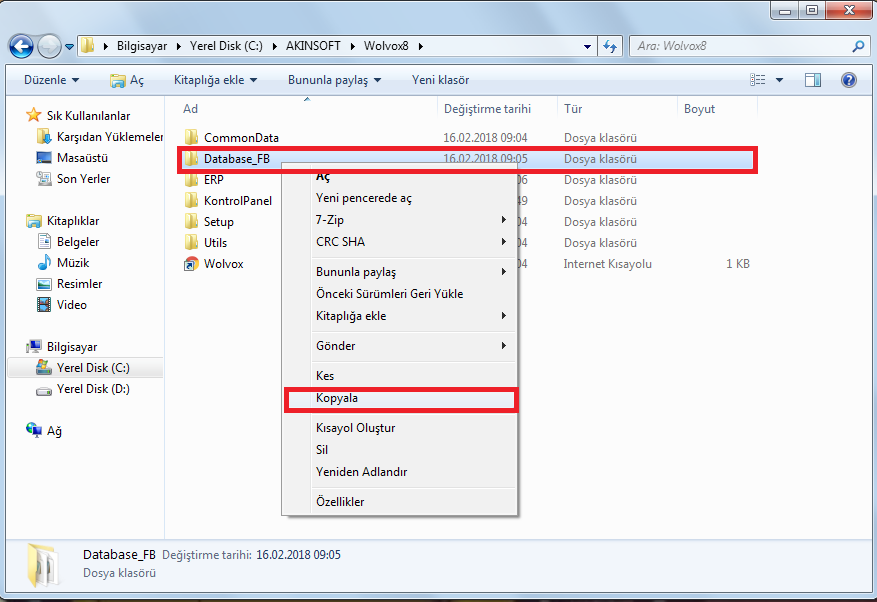
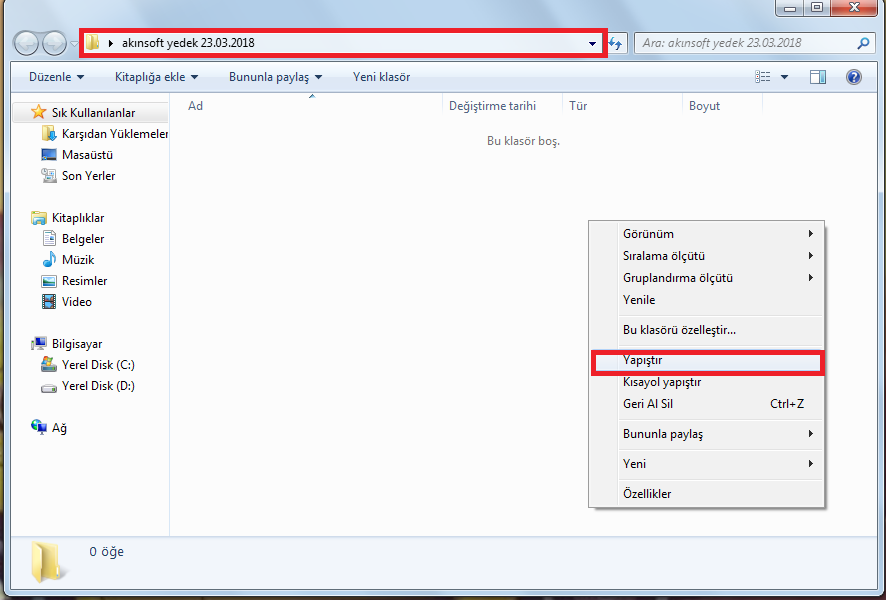
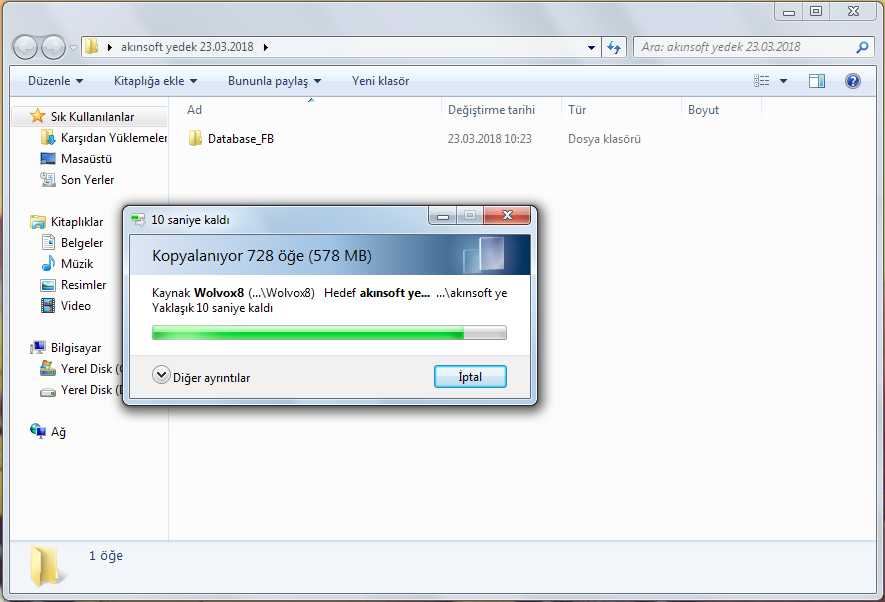
Aktif için TAsRadioGroup, Kalan Miktarı Göster TAsFiltreCheckBox, Filtrele, Temizle TsBitBtnAs, Listelenen Kayıt Sayısı TsStatusBar, Filtreleme seçenekleri TsSpeedButton, Sıralanacak Alanlar TsLabel, Aktar TsBitBtnAs.

**FireBird Veri Tabanlı AKINSOFT Programlarında Datadan Yedek Alma ve Geri Yükleme İşlemleri**

Akınsoft programında herhangi bir sebeple yedek almak için;

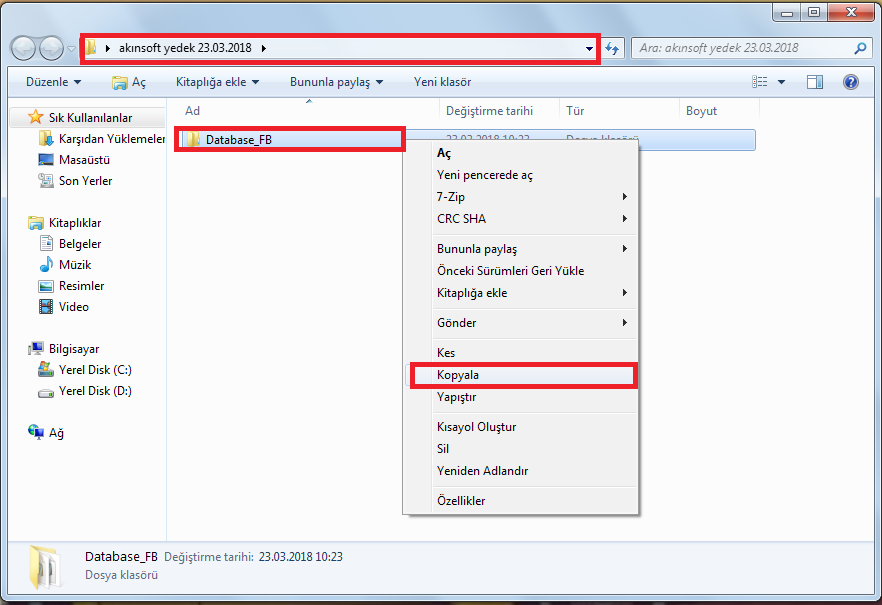
Öncelikle tüm AKINSOFT programlarının kapatılması ve Denetim masası/yönetimsel araçlar/hizmetler içerisinden FireBird Server ve Guardian’ı durdurulması gerekir. Aksi taktirde alınan yedek bozuk ya da boş olacaktır.

Tüm programlar kapatıldıktan sonra;

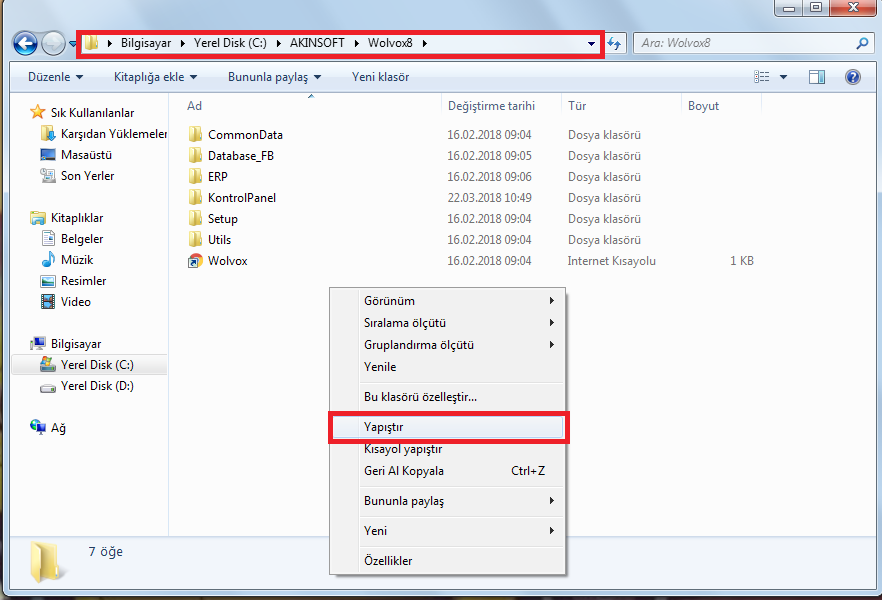
* Programın kurulu olduğu dizin içerisinden (AKINSOFT/WOLVOX7-8) içerisinde bulunan DATABASE\_FB klasörünü kopyalanır.
* Kopyalanan DATABASE\_FB klasörünü harici disk içerisine yedek için açılmış olduğunuz klasöre yapıştırılır. Kopyalama işlemi başlar.
* Kopyalama işlemi bittikten sonra yedek alınmış olur.

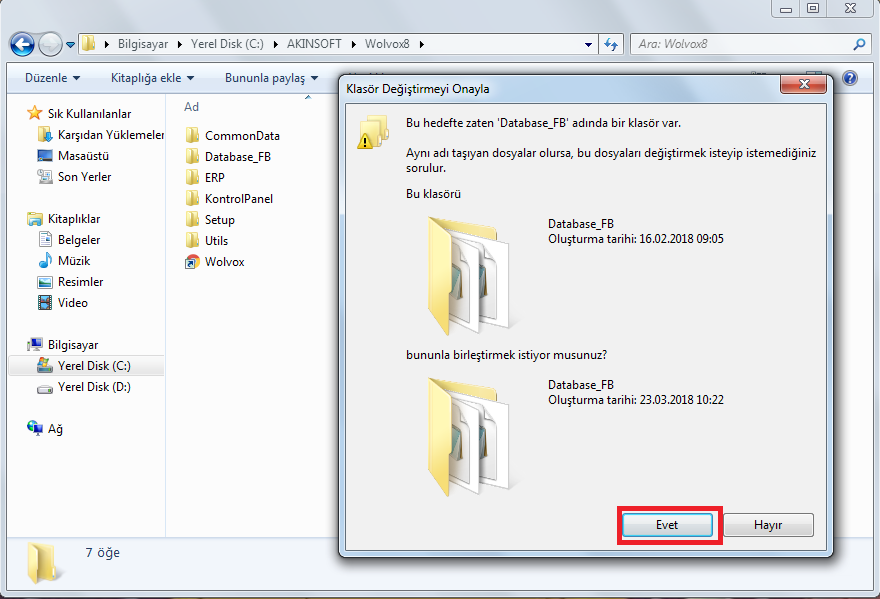
**FireBird Veri Tabanlı AKINSOFT Programımda Datadan Alınan Yedeği Geri Yükleme**

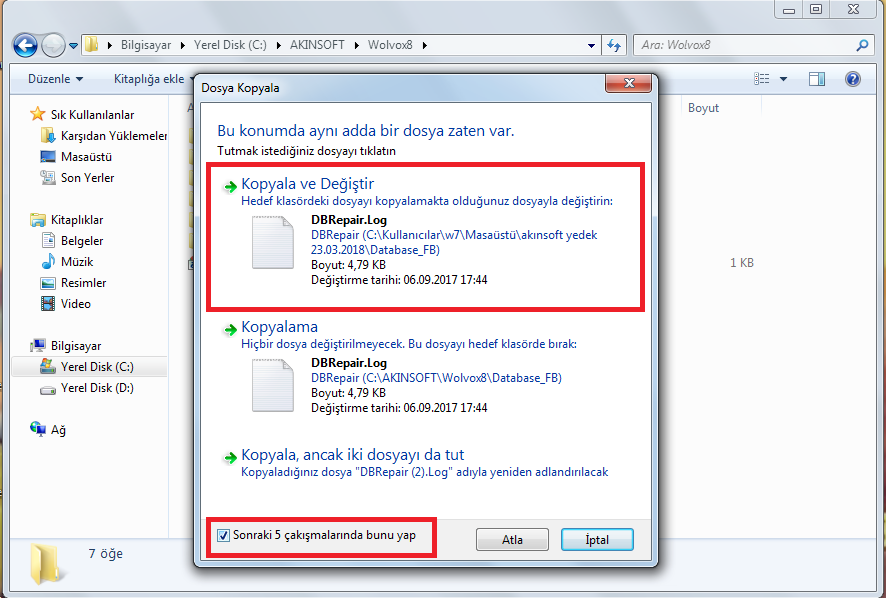
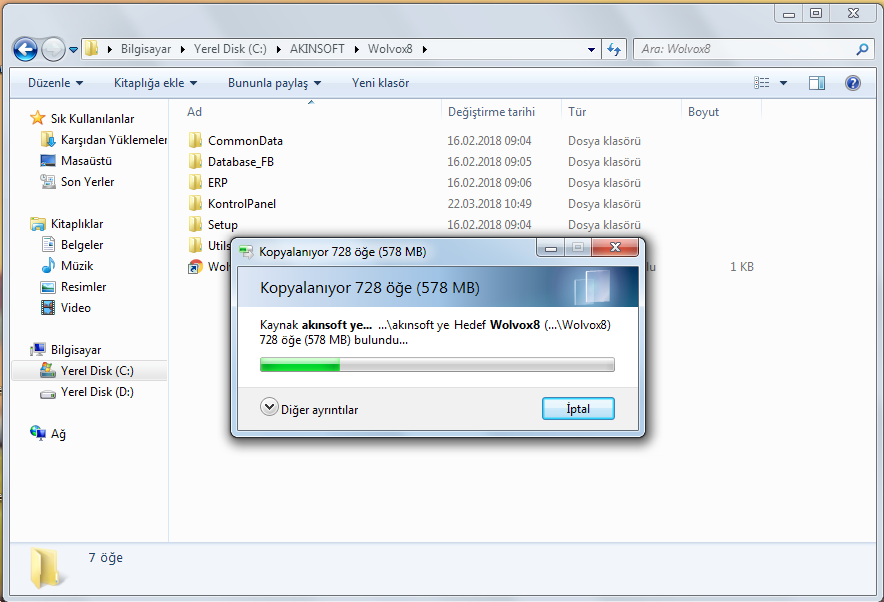
* AKINSOFT programının kurulumu yapılır.
* Mevcut kurulumu yapılan programlar kapatılır. Denetim Masası/Yönetimsel Araçlar/Hizmetler içerisinden FireBird Guardian’ın durulması gerekir. Kopyala yapıştır yöntemi ile harici diske yedeklenen DATABASE\_FB’yi geri yükleme yapılacak olan bilgisayara harici diskten alınır.
* Harici diskten bilgisayara alınmış olan yedek klasörünün içerisinden DATABASE\_FB kopyalanır.



* Programın kurulu olduğu dizin içerisine (AKINSOFT/WOLVOX7-8)DATABASE\_FB ‘nin olduğu sayfaya yapıştırılır.

Yeni bilgisayara ya da format atılmış bilgisayara geri yüklerken klasörde başka bir data yoksa yeni data direkt olarak atılabilir. Fakat datanın atılacağı klasörde bir başka data klasörü daha varsa bu klasörün adı değiştirilerek yeni data klasörüne atılmalıdır.

* Bu işlemden sonra ilk kurulumdaki DATABASE\_FB ile yedek alınan DATABASE\_FB yer değiştirdi ve tüm veriler geri yüklendi.



Tüm işlemler bittikten sonra Denetim Masası/Yönetimsel Araçlar/Hizmetler içerisinden FireBird Guardian başlatılır.

Staj süresince;

Değerli bilgilerini benimle paylaşan, kendisine ne zaman danışsam bana kıymetli zamanını ayırıp sabırla ve büyük bir ilgiyle bana faydalı olabilmek için elinden geleni sunan ve gelecekteki mesleki hayatımda da bana verdiği değerli bilgilerden faydalanacağımı düşündüğüm Etem KAYA’ya,

Şirket çalışanı olmayı davranışlarıyla öğrettiği için Ahmet EKEN’e ,

Kariyerim için büyük bir adım atmama yardımcı olan Yazılım sektöründe birçok projede ilklerine imza atan, başarısını yaptığı ve devam ettirdiği projelerle kanıtlayan AKINSOFT’a ve yanımda olan beni destekleyen çalışma arkadaşlarıma, AKINSOFT Ailesine teşekkür ediyorum.