YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ÖĞRENCİ OTOMASYON SİSTEMİ

DİABY MOHAMMAD SALİM

F.B.E. Bilgisayar Bilimleri Anabilim Dalında
hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yard.Doç.Dr. Tülay TİNLİ

İSTANBUL, 1995

İÇİNDEKİLER

ŞEKİL VE TABLO LİSTESİ

TEŞEKKÜR

Proje çalışmam boyunca benden deneyim ve tecrübelerini esirgemeyen proje yöneticim Yrd.Doç.Dr. Tülay Tinli'ye teşekkürlerimi sunarım.

Ayrıca, Bilgisayar Bilimleri ve Mühendisliğ Bölüm Başkanı Prof.M.Yahya Karslıgil'e bana gösterdiği dikkat ve yakın ilgisinden dolayı teşekkür ederim.

Bunun yanı sıra, bilgisayar bilgime katkıda bulunan ve akademik başarımı sağlayan bütün hocalarıma teşekkür ederim.

Bilgisayar Bilimleri Mühendisliğ Bölümü Araştırma Görevlilerine, benimle ilgilenmeleri ve sorularımı cevapsız bırakmamaları nedeni ile ayrıca teşekkür ederim.

PROTEK(Proje Danışmanlık Mühendislik Bilgi İşlem Servisleri San. ve Tic. Ltd. Şti.) firmasındaki herkese tezimi hazırlarken bana her türlü olanakları sundukları için, yardımseverlik ve cömertliklerinden dolayı şükran duygularımı iletmek isterim.

Son olarak bana moral veren ve fikirleri ile katkıda bulunan bütün arkadaşlarıma teşekkürlerimi sunarım.

ÖZET

Günlük hayatımızda, bilgisayar biliminin en geniş yelpazesini veritabanı sistemleri oluşturmaktadır. Ve bu veritabanı sistemleri, bilgisayarların bütün tiplerinde uygun olarak çalışmaktadır. (Micro,Mini ve Mainframe)

Ögrenci otomasyonu projesinin amacı, çok sayıda kayıt olmasına rağmen kayıt ekleme, silme, degiştirme gibi işlemlerin çok basitce ve efektif olarak gercekleştirilmesidir.

Bu kitabın birinci bölümünde, Veritabanı yönetimine giriş güzel örneklerle anlatılmıştır. Verilerin toplanması ve yönlendirilmesi, fizibilite çalışmaları altında ikinci bölümde anlatılmaktadır.

Bu projenin uygulamalarında kullanılan veritabanı yazılım dili FoxPro dur. Yazılım dili olarak FOXPRO'nun seçilmesinin sebebi veritabanı yazılımlarının en büyüklerinden biri olmasıdır. FOXPRO'nun özellikleri, SQL, QUATTRO PRO ve PARADOX gibi diger yazılımlarla karşılaştırılması üçüncü bölümde anlatılmaktadır.

Bu projedeki veritabanı kullanımı temel bir FOXPRO bilgisi veya veri tabanı yönetimi bilgisini gerektirmemektedir. Bütün komutlar tanımlanmıştır ve dördüncü bölümde iyi bir şekilde açıklanmıştır.

Ek olarak, FOXPRO'da gerçekleştirilmiş programlar bu kitabın sonuna eklenmiştir.

BÖLÜM I

VERİ YÖNETİMİNE GENEL BAKIŞ

1.1 Örnekle Giriş

Veri tabanı sistemi, bilgisayarlaştırılmış kayıt tutma sisteminden başka birşey değildir. Veritabanını, elektronik veri toplama dolabı olarak kabul edebiliriz. Sistemi kullanacak olan kullanıcıya dosyalar üzerinde yapabileceği aşagıdaki gibi birçok kolaylık sağlanmıştır.

. Veri tabanına yeni(boş) dosyalar eklemek
. Mevcut dosyalara yeni kayıtlar eklemek
. Mevcut dosyadan bilgileri bulup getirme
. Mevcut dosyalar üzerinde güncelleme yapma
. Mevcut fiile' lardan kayıt silme
. Mevcut file'ları (boş veya herhangibir durumda) veritabanı içine taşıma

Şekil 1.1' de ÖGRETMEN adıyla olusturulmus küçük bir veritabanı bulunmaktadır.

İlk önce açık olarak bellidir ki ÖGRETMEN kütügü gibi bilgisayar kütükleri kütük diye adlandırılmayıp tablo adı verilmektedir ve bu kitapta sık sık bu terim kullanılacaktır. (Aslında bunlar ilişkisal tablolardır)

İkinci ve son olarak denilebilir ki ; Kütükte bulunan kayıtlarada 'satır' adı verilmektedir. (Bazen bunlar 'mantıksal kayıt' diye adlandırılabilirler) Bunun gibi ' sütun' terimi de bu mantıksal kayıtların alanları olarak mütala edilebilir.

Bu kitapta satır terimi ile kayıt terimi, sütun terimi ile de alan terimi dönüşümlü olarak kullanmaktayız.

1.2 VERİTABANI SİSTEMİ NEDİR ?

Bölüm 1.1 de de söylendigi gibi veritabanı sistemi bilgisayarlaştırılmış kayıt saklama sistemidir yani bir başka deyişle bu bilgisayarlaştırılmış sistemin genel amacı gelen bilgi isteklerine cevap verip bilgiyi hep hazırda tutmaktadır. (Bilgi ve veri terimleri bu bölümde BİLGİYE KARŞIN VERİ başlığı altında

BİR ÖĞRETMENİN DBF ÖRNEĞİ

OGRETKOD	UNVANINI	ADI	SOYADI	UNIVERSITE	FAKULTE	BOLUM	ANABILIM	EVTELFON	ISTELFON	E
01	Y.doç.dr.	Recep	Erdoğan	Yıldız Tek.Üni.	Elektrik E.Müh.	Bilgisayar	Bilgisayar	(O216)649 78 95	332	Ul
001	Y.doç.dr.	Abdürrezzak	Bozdoğan	Yıldız Tek.Üni.	Elektrik E.mÜh.	Bilgisayar	Bilgisayar	(0212)331 14 19	297	Ka
045	Y.doç.dr.	Can	Kakahasanoğlu	Yıldız Tek.Üni.	E.E.Müh..	Bilgisayar	Bilgisayar	(0212)444 23 11	356	Be
023	Y.doç.dr.	Aysun	Ünlü	Yıldız T. Üniversitesi	Elektrik	Elektrik	Elektrik Elektronik	(0216)256 85 23	345	59
012	Y.doç.dr.	Belkıs	Bilgin	Yıldız T. U.	Fen Bilimleri	Kimya	Fen Bilimleri	(0212)774 86 90	423	M
031	Doç. Dr.	Yavuz	Karayalçın			Bilgisayar	Elektrik Elekronik			
021	Prof.dr.	Mehmet	Karayalçın			Bilgisayar B. Ve Müh	Elektrik _ Elektronik			
044	Doç. Dr.	Zekiye	Hanım	Y. T. Ü.	E. E. F.	Bilgisayar B. MÜh.	Elek. Elektro. Fak.	(0216)888 23 65	275	67
207	Arş. Gör	Fatma	Yılmaz	Y. T. Ü.	Elektrik Elektronik	Bilgisayar	Elektronik	(0216)834 56 04	207	Ka
45	Prof. dr.	Hasan	Özdemir	Boğazici	Elektrik elrktronik	Bilgisayar	Bilgisayar	(0212)348 23 56	305	Be

Şekil 1.1 Öğretmen Dosya Örneği

incelenmektedir). Burada bilgi diye kastedilen sistemin bir kişi veya kuruluşa sunduğu anlamı olan her türlü şeydir. Şekil 1.2'de büyük ölçüde basitleştirilmis şekliyle bu veri tabanı sistemini görebiliriz.

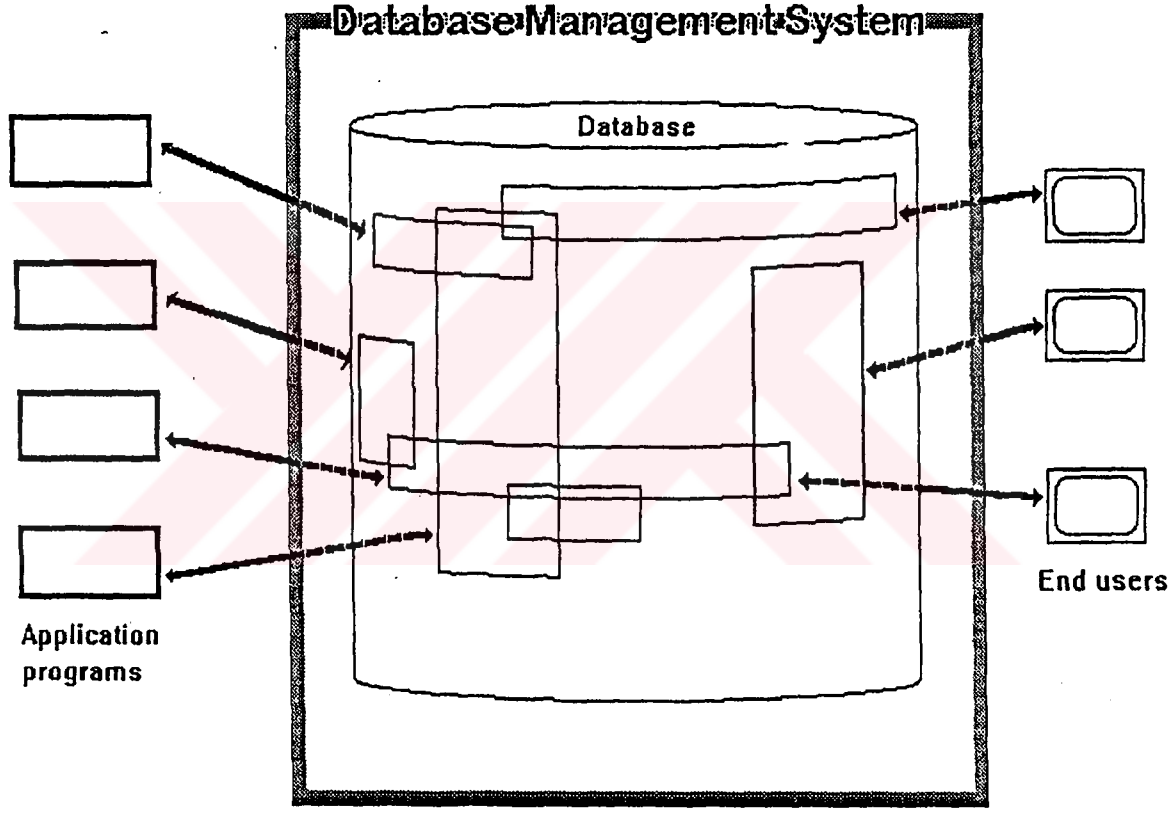
Şekil 1.2 dört esas unsurdan oluşan veri tabanı sistemini göstermektedir ki bu unsurlar VERİ, DONANIM;YAZILIM ve KULLANICILAR'dır. Bu dört unsur ileride açıkça belirtilecektir.

1.2.1 VERİ

Günümüzde küçük mikrobilgisayarlardan, büyük ana bilgisayarlara kadar her türlü bilgisayarda veritabanları kullanılmaktadır. Tabidir ki veritabanı kolaylıkları üstünde çalışan makinanın gücü ve kapasitesiyle dogru orantılıdır. Genel olarak büyük bilgisayarda veri tabanları çoklu kullanıma daha yatkın iken küçük bilgisayarlarda bu durum daha çok tekil kullanıma eğilim göstermektedir. Tekil kullanımda sisteme aynı anda sadece tek bir kullanıcı erişebilmektedir. Çoklu kullanımda ise aynı anda birçok kullanıcı sisteme ulaşabilmektedir. Bu kitapta Şekil 1.2 deki gibi çoklu kullanımı esas alacağız. Çoklu kullanımın bir hedefide her kullanıcının ayrı ayrı birbirini etkilememeksizin çalışabileceği bir ortamdır. Çoklu kullanımın kendine özgü problemleri de sistemin kendi iç sorunu olarak ortaya çıkmaktadır. Kullanıcılar tarafından herhangibir etkileşimi yoktur.

Böylelikle tarif edilmekte olan çoklu kullanıcılı veritabanı sistemi hem ortaklaşa kullanılabilmeli hem de entegre olmalıdır. Entegre olabilme özelliğinin büyük sistemlerde olduğu gibi küçük sistemlerde de bulunması gerekir. Bu iki özellik veri tabanı sistemlerinin önde gelen iki avantajı olarak karşımıza çıkmaktadır. (Tabii ki bunlardan başka birçok avantajlarından da söz edilebilir. Daha fazla detay bu kitabın incelediği konuların dışındadır.) Fakat önce entegre ve paylaşılabilirlik terimlerinin ne anlama geldiğini açıklayalım.

ENTEGRE terimi ile veri tabanının birkaç veri küğünü birleştirerek istenen bilgiyi ortaya getirmesini kastetebiliriz. Örneğin bilinen bir veritabanı isim,adres,bölüm,maaş, v.s. gibi alanlardan oluşan bir ÇALIŞANLAR kütüğünü , isim ve kurslar alanlarından oluşan bir KAYITLAR kütüğünü içersin. Eğitim kurslarındaki kayıt işlemleri için her kaydedilen çalışan bölümünün bilinmesi gerekir. Yani KAYITLAR kütüğüne bu bilgi için başvurmaya gerek olmadığı çok açıktır. Çünkü bu bilgi için başvurmağa gerek olmadığı cok açıktır. Çünkü bu bilgi sadece ÇALIŞANLAR kütüğünden elde edilebilir.



ŞEKİL 1.2 VERİ TABANI SİSTEMİNİN BASİTLEŞTİRİLMİŞ ŞEKLİ

EMPLOYEE

İSİM ADRES BÖLÜM MAAŞ ....

ENROLLMENT

İSİM DERS .....

Şekil 1.3 employee ve enrolment dosyaları

PAYLAŞILABİLİRLİK teriminden kasıt ise veri tabanındaki bölümlerin kullanıcılar tarafından ortaklaşa kullanılabilmesidir. Yani her kullanıcının aynı veriye erişebilmesidir. (Değişik kullanıcılar değişik amaçlar için aynı veriye erişebilirler). Daha önce de belirtildiği gibi değişik kullanıcalar aynı veriye aynı anda da erişebilirler buna da ("AYNI ANDA ERİŞİM") adı verilir. Bu tür erişim bir anlamda veri tabanının entegre olmasıyla da yakından ilgilidir. ÇALIŞANLAR ve KAYITLAR (eğitim kayıtları) kütükleri örneğine bakacak olursak ÇALIŞANLAR kütüğünün birçok bölüm tarafından kullanılabileceğini görebiliriz. Örneğin ÇALIŞANLAR kütüğünün eğitim depatmanı ve personel departmanı tarafından paylaşılması olayı dikkate alınabilir.

1.2.2 DONANIM

Sistemin Donanımı :

. Yardımcı bellek (secondary storage) verinin tutulduğu alan olmasıyla birlikte I/o cihazları, cihaz kontrolerları ve I/O kanallarıyla birlikte kullanılır.

. İşlemci (ler) ve ana hafıza da veri tabanı sistem yazılımına destek sağlar.

1.2.3 YAZILIM

Yazılım, fiziksel olarak düşündüğümüz veri tabanı ile sistemin kullanıcıları arasında yer alan bir katmandır, buna diğer bir adla veritabanı yönetimi veya daha genel olarak veritabanı yönetim sistemi (DBMS) adı verilir. Kullanıcılardan gelen tüm istekler DBMS tarafından cevaplandırılır. Bu bölümün başında değinilen kütük yaratma, kütük silme, kütükten kayıt okuma, güncelleştirme v.s. gibi kolaylıklar DBMS tarafından sunulan hizmetlerdir. DBMS'in sunduğu en genel fonksiyon ise veri tabanı kullanıcılarının donanım detaylarından izolasyonu olarak düşünülebilir. (Sistem programları, uygulama

programlarını donanım seviyesi detayları ile uğraşmamalarını sağlar). Birbaşka deyişlle, veritabanı yönetim sistemi DBMS donanım seviyesine inip veriyi kullanıcı katmanına aktarır ve gereken operasyonları icra eder. (Örneğin bu kitabın sonunda göreceğimiz FOXPRO operasyonları).

Kolaylıkla denilebilinir ki DBMS tüm sistemin içindeki en önemli yazılım parçasını teşkil etmektedir. Fakat bu yazılımdan başka yazılımlar da sistemde mevcuttur. Örneğin diger kolaylıklar, uygulama geliştirme ortamları, tasarım kolaylıkları, raporlayıcılar v.s.

1.2.4 KULLANICILAR

Üç çeşit kullanıcı grubundan bahsedebiliriz;

* İlk olarak UYGULAMA PROGRAMCISI grubunu ele alabiliriz. Bu grubun görevi veritabanını kullanan uygulama programları geliştirmektir. COBOL, PL/1 gibi eski kökenli programlama dilleri veya PASCAL, C gibi daha yeni diller kullanılabilinir. Bu programlama dilleri kullanılarak kayıt ekleme, silme, güncelleştirme veya okuma gibi bilinen klasik işlemler gerçekleştirilebilinir. Tüm bu operasyonlar tabii ki DBMS'e gerekli komutlar gönderilerek yapılmalıdır. Bu programlar BATCH programlar olabilir veya interaktif (etkileşimli çalışan) olarak terminal başında veritabına erişen uç kullanıcılara hizmet veren programlar olabilirki, modern programlar daha çok bu sınıfa girmektedir.

* İkinci grup uç kullanıcı grubudur ki bunlar bir terminal vasıtasıyla sistemle etkileşimli olarak çalışırlar. Bir uç kullanıcı on-line olarak veritabanına erişebilir. Veritabanı sistemlerinde en az bir tane olsun BUILT-IN uygulamaları vardır. DBMS'deki INTERACTIVE QUERY LANGUAGE PROCESSOR'ü söyleyebiliriz. Bu kitapda kullanılan FOXPRO dili, veritabanı sorgulama dilinin tipik bir örneği olarak bulunmaktadır.

* Üçüncü sınıf kullanıcı (Fig. 1.2 gösterilmemektedir)
  - . DATA ADMINISTRATION (Veri Yöneticisi) ve
  - . DATABASE ADMINISTRATION (Veritabanı Yöneticisi)

Veri yöneticisinin (Data Administration) işi, veritabanında saklanacak olan verinin yerini ve bu verinin güvenliğinin sağlanması için gerekli olan güvenliğe karar vermesidir.

Veri yöneticisi, yoneticidir, teknisyen değildir. Teknik kişi DATA ADMINISTRATOR (DB) (Veri yöneticisi) ın kararlarını yerine getirmekle sorumlu olan DATA-BASE ADMINISTRATOR (DBA) dır.

DBA, ver yöneticisine benzemez. DA profesyoneldir. DBA'nın işi, veritabanını yaratır, DA tarafından belirlenen güvenlik yöntemlerini geliştirip uygular. DBA, teknik asistan ve sistem programcısı seviyesindedir. Pratikta DBA'nın fonksiyonu birkaç kişiden olusan bir takım kurmak (kesinlikle bir kişi değil) ve DA'nın isteklerini yerine getirmektir.

BÖLÜM 2

VERİ TABANININ TEMELLERİ

2.1 VERİ TABANI (DATABASE) NEDİR ?

Bilgisayar çok kısa bir zamanda, büyük veri yığınları arasından istenen bilgiye hızla erişerek bunu ekranında görüntüleyebilmektedir. Bilgisayarin eriştiği bilgiye **VERİ (DATA)** , büyük veri yığınlarına ise **VERİ TABANI (DATABASE)** adı verilir.

Veri tabanları şu yada bu şekilde, farkli isimlerde yıllardır kullanılmaktadır. Telefon rehberi, randevu defteri günlük yaşantımızda sıkça karşılaştığımız veri tabanlarındandır. Bunlar rastgele erişilebilen bir takım bılgilerin, (ilerdeki bölümlerde veri (data) ve bilgi (information) arasındaki farka değinecektir.) belli bir mantık sırasında yerleştirilmesi sonucunda oluşmuşlardır.

Bilgisayarların günlük yaşantımıza yoğun bir şekilde girmesinden önce şirketler topladıkları bilgileri manual olarak işlemekte idi. Büyük evrak birimleri, nispeten hızlı bir şekilde veriye erişebilen personel ordusu ile çalışmakta idi. Manual olarak çok iyi idare edilen bir sistem dahi olsa, çalışanların tüm istekleri layıkı ile karşılaması mümkün olmamaktaydı. İş yönetim tekniklerinin günden güne daha karmaşık hale gelmesi ve artan rekabet yeni bilgi erişim sistemlerinin ortaya atılmasına neden oldu. Verinin yığınlar arasından çekilip alınmasının yanı sıra analiz edilmesi de gündeme geldi.

Bilgisayarların kullanımı, bilgi sistemleri konusunda bir devrim yaptı. Bilgisayarların yardımıyla çalışanların yaptıkları işler daha hızlı, daha düzgün ve daha az maliyet ile yapılmaya başlandı. Bilgisayarlarla veriye hızlı erişmenin yanı sıra bilgileri istenen biçimde sıralayabilme, saniyenin ondabiri sürede hesaplamalar yapabilme, çok çeşitli matematiksel işlemleri hatasız olarak gerçekleştirilebilmesi mümkün oldu.

Bilgisayarlı bilgi sistemleri iki temel parçadan oluşur; **DONANIM (HARDWARE)** ve **YAZILIM (SOFTWARE)**. Donanım, bilgisayarın elektronik devrelerden meydana gelen kısmına verilen addır. Yazılım, genellike disk üzerinde bulunan ve bilgisayara ne yapacağını söyleyen elektronik komutlar topluluğudur. Yazılım sayesinde bilgisayar bir uçağın kontrolünden, bir firmanın muhasebe işlerini kadar değişik işleri yerine getirebilmektedir.

Kişisel bilgisayarlar üzerinde çalışan yazılımlar genellikle şu üç temel gruba ayrılır;

* Kelime İşlem (Word Processing)
* Tablolama-Bordro (Spreadsheet)
* Veri Tabanı (Data Base)

FOXPRO, kişisel bilgisayarı elektronik bir evrak dolabına dönüştürebilen, kullanımı kolay ve etkin bir veri tabanı yazılımıdır.

2.2 VERİ (DATA) BİLGİ (INFORMATION) İLİŞKİSİ

Veri (data) ve **bilgi (information)** zaman zaman bir diğerinin yerine kullanılmaktadır. Bu bölümde bu iki terim arasındaki fark ortaya konacaktır. Veri, insanları, yeri, başka cisim veya kavramları ilgilendiren gerçeklerdir. Verinin bilgisayar tarafından erişilebilen hafiza ve disk türü ikincil hafiza birimlerinde tutulduğu kabul edilir. Veri, amaca uygun olarak işlenmediği sürece karar verecek kişi için bir anlam ifade etmez.

Bilgi (Information), karar vercek kişiye kolaylık sağlayacak şekilde işlenmiş, düzenlenmis veri'ye verilen addır. Örenegin ödeme sürelerini 60 gün yada daha fazla aşan müşterilerin isim, adres ve borç miktarlarını gösteren rapor, karar verecek kişi için bilgi niteliğini taşır.

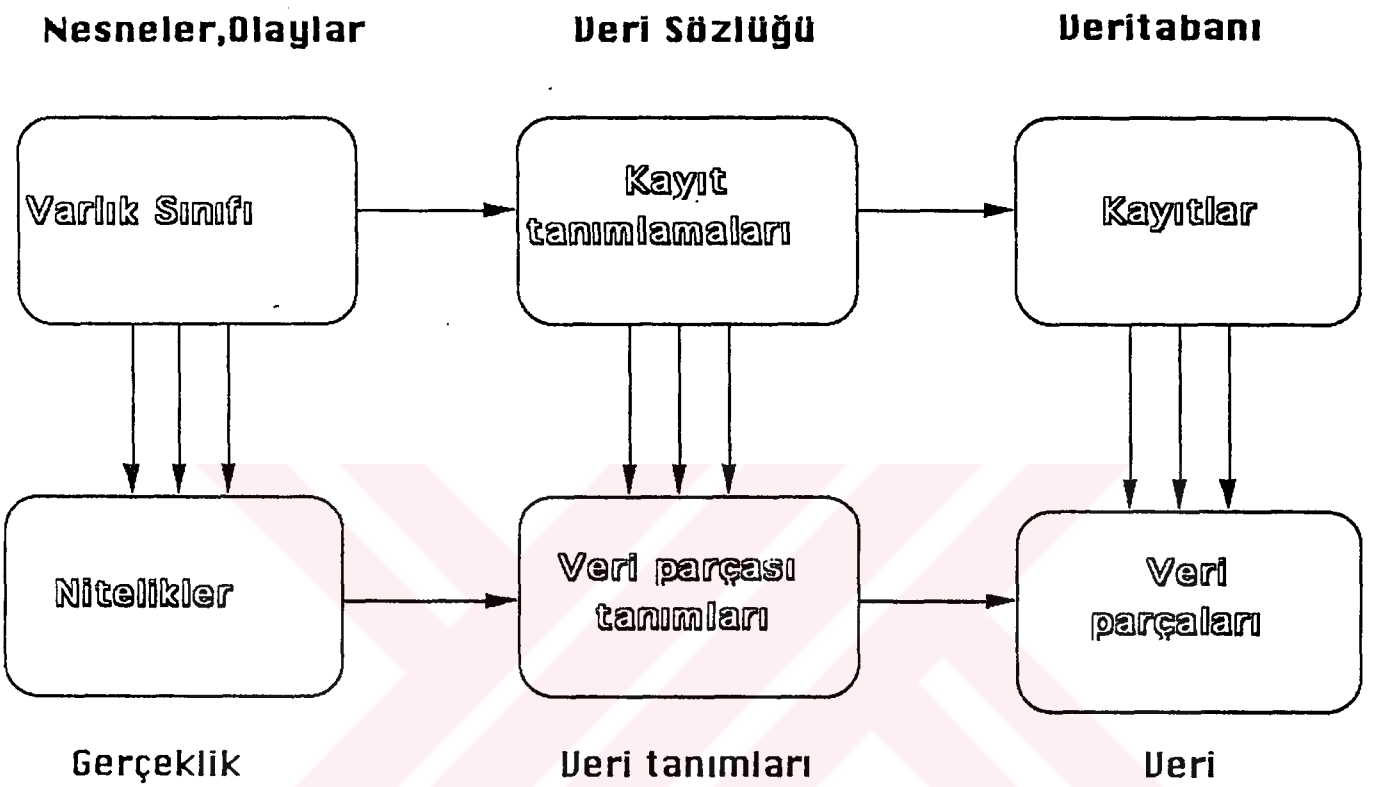
Uygulamada veri (data) ve bilgi (information) arasındaki farkı tespit etmek pek kolay olmamaktadır. Veri, herhangi bir problemin çözümünde yada bir kararın alınmasında kullanılmak için işlendiğinde bilgi halini alır. Sonuç olarak tarif verinin ne olduğuna bağlı olarak değişmekte bu nedenden ötürü veri ve bilgi terimleri biribirinin yerine kullanılabilmektedir.

2.3 VERİNİN YAPISI

Bu bölümde verinin temel tanımı üzerinde durulacaktır. Şekil 2.1 de görüldüğü gibi veriyi tanımlarken üç seviyeli soyutlama kullanılır.

Bunlar;

* Gerçeklik (Reality)
* Veri ile ilgili bilgi (Metadata-information abour data)
* Gerçek veri (Catual data)



şekil 2.1. Veri tanımlamasındaki üç temel kavram

Gerçeklik

Gerçeklik kurumun faaliyet gösterdiği alan, kuruma bağlı çeşitli birimler, kısacası kurumun kendisidir. Kurum, belli bir amaca ulaşmak için çalısan insanların, kullanılan cihazların ve nesnelerin topluluğudur. Her kurum bulunduğu çevreyi etkiler ve çeversinden etkilenir.

Bir varlık (entity), bir kurumun hakkında veri toplamak ve toplanan veriyi saklamak istediği nesne veya olay olarak tarif edilir. Varlık, kurum çalışanı, mamül yada bilgisayar gibi tutulur, hissedilir cinsten olabileceği gibi banka hesabı, hava yolculuğu gibi tutulamayan nesneler de olabilir. Kelebek Mobilya da kullanılan örnek varlıklar , müşteriler, mamüller, siparişler, üretim alanları ve çalışanlar olabilir.

Bir varlık sınıfı, benzer özellikler gösteren varlıkların bir topluluğudur. Varlık sınıfına örnek olarak müşteriler, öğrenciler ve hastalar gösterilebilir. Varlık sınıfları bazen varlık kümeleri yada varlık tipleri de denilmektedir.

Varlıklar, kolaylık olması açısından varlık sınıfları (entity class) olarak guruplanırlar. Genelde her varlık sadece ve sadece bir varlık sınıfına dahil olabilir. Buna rağmen varlık sınıflarının oluşturulması ve bu sınıfta yer alacak varlıkların tespit edilmesi keyfi olabilir. Örnek olarak çalışanların oluşturduğu bir varlık sınıfını ele alacak olursak acaba bunların hepsi tam-gün çalışan personel midir? Üretim merkezleri sınıfında hangi varlıklar vardır? Tek başına çalışabilen makinalar mi?, makina grupları mı? yada tüm çalışma bölümleri mi? Bu ve benzeri soruların kuruma ait veri yapısı modeli kurlurken cevaplanması gerekmektedir.

Varlık gruplarının sayısı, kurumun büyüklüğüne ve karmaşıklığına bağlı olarak değişmektedir. Örnek olarak orta ölçekte bir kurumun varlık sınıfları birkaç yüz tane olurken, Kelebek Mobilya fabrikası gibi küçük bir kuruluş muhtemelen yüz taneden az varlık grubu olacaktır.
Her varlık sınıfı için kurumun ilgi duyduğu bir takım özellikler vardır. Bir varlığı belirlemek için onu ifade eden özelliği kayıt edilir. Kelebek Mobilya örneğini ele alırsak, müşteriler ve mamüller iki ayrı varlık sınıfıdır. Bu varlık sınıflarının özellikleri aşağıda verilmistir;

MÜŞTERİLER	MAMÜLLER
Müşteri numarası	Mamül numarası
İsim	Mamül tarifi
Adres	Fiyat
Telefon	Ağırlık

Denge

Her varlık sınıfı için genel olarak on yada daha fazla özellik tanımlanmıştır. Buna göre birkaç yüz varlık sınıfına sahip bir kurumun birkaç bin özellik tanımlaması doğaldır.

Her varlık bulunduğu varlık sınıfında kendisini diğer varlıklardan ayırt edecek en az bir özelliğe (yada bunların bir birleşimine) sahip olmalıdır. Varlığı belirleyen yegane özelliğe **belirleyeci (identifier)** denir. Örneğin sosyal güvenlik numarası çalışan kişi için bir belirleyici (identifier), mamül numarası ise mamül için belirleyicidir. Bazı durumlarda, belirleyici bir yada birkaç özelliğin bir birleşimi olabilir. Örneğin fatura numarası ile mamül numarası gibi. Varlık belirleyicisinin tek olması gerekir, bir varlık sınıfı içerisinde aynı özelliğe birden fazla varlık sahip olamaz.

Özellikler, bağımsız varlıkların nitelikleridir. İlgilendiğimiz özelliklerden biriside **Birlik (Association)** dir. Birlik iki yada daha fazla varlık arasında bir tür ilişkidir. Birlik, aynı varlık sınıfı veya iki yada daha fazla varlık sınıfına dahil varlıklar arasında söz konusu olabilir. Bunu Kelebek Mobilya ile örneklemek gerekirse;

1-) Çalısanlara ait varlık sınıfında, bir kısım çalışanlar yönetici, kalanlar ise işçidir. **Yönetilen** isimli birlik bize bir yöneticiye bağlı olan işçileri ifade edecektir.

2-) **Müşteri** ve **sipariş** varlık sınıfları vardır. Buna göre bu iki varlık sınıfı arasında **Açık siparişler** isimli bir birlik tanımlanırsa, bu birlik herhangi bir müşteriye yollanması gereken siparişleri belirlememize yardımcı olacaktır.

Varlıklar arasındaki birlikler (bağlantılar) veri yapılarının alışıla gelen dosya yapıları ile arasındaki farklılığı ortaya koymaktadır. Varlılar, kurumun içinde bulunduğu ortam ve kurumun gerçek hayatta ilgi duydugu parçalardır. Ancak herhangi bir yönetici (yada işçiler için) mevcut varlıkların doğrudan izlemesi ile bir karar alması hiç kolay değildir. Herhangi bir stok hareketinde yöneticinin depoya gidip malları saydığını yada her siparişte müşterileri arayarak adreslerini öğrendiğini düşünün. Her varlığın tek tek izlenmesi yerine, varlıkların özelliklerini (nitelik/nicelik) tarif eden modeller kurum tarfından kullanılır. Verilerin özelliklerini tarif etme bizi veri ile ilgili bilgi yani **Metadata** kavramına götürür.

Veri ile ilgili bilgi - Metadata

Metadata kurum icinde kullanılan veri ile ilgili bilgidir. Metadata veri tabanı yöneticileri, kurumun varlıkları ile ilgili modeller geliştiren , varlıklar arasında bağlantılar kuran personel tarfından kullanılır. Metadata, kurum içinde tutulan veri rehberinde saklanır.

Veri (data), veri tabanında kullanıcı için anlam ifade eden en küçük parçadır. Veri kalemleri ne örnek olarak **Çalışanın ismi, Öğrenci numarası, Sipariş tarihi** verilebilir. Veri rehberinde her veri için kataloglanmış, veri parçasının ismini, uzunluğunu, tipini ve tanımını içeren bir bilgi bulunur.

Veri parçaları bazen **veri elemanları (data elements), alan (field)** veya **nitelik (attribute)** olarak da isimlendirilir. Buna rağmen biz veri terimini bir veri ünitesini anlatmak için kullanırız.
Alan (field) mantıksal bir terimden ziyade fiziksel bir yapıyı ifade etmekte ve **kayıt (record)** içerisinde verinin yerini (yani kaçıncı kolonda olduğunu) belirlemekte kullanılmaktadır. Daha önce nitelik (attribute), veriye yönelik terimden ziyade gerçek varlıklalra ait özellik olarak tarif edilmişti. Buna rağmen her nitelik için bir veri parçası olması nedeniyle bu kavramlar biri diğerinin yerine kullanılabilmektedir. Aslında bağlantılı (relation) modelde nitelik, veri parçasının yerine kullanılmaktadır. Veri elementi kavram , veri yapısı yerine nadiren kulalnılabilen bir terimdir.

Veri kümesi, bir bütün olarak isimlendirilen veri parçalarının bir bütünüdür. Örneğin **İsim** olarak adlandırılmış bir veri kümesi, **Soyad, Ad** isimli veri parçalarından oluşmuş olabilir. Benzer şekilde **Satiş** isimli veri kümesi, **ilk üç ay, ikinci üç ay, üçüncü üç ay** ve **dördüncü üç ay** olmak üzere dört adet veri parçasından oluşabilir. Veri kümeleri kullanılmadan önce mutlaka veri rehberinde tariflenmelidir. Her küme tipi için veri küme ismi, tanımı ve içerdiği veri parçalarının metadata olarak kayıt edilmesi gerekmektedir. COBOL programlama dilinde veri kümeleri genelde grup parçaları olarak ifade edilir.

Kayıt, veri parçaları ve/veya veri kümerlinin bir topluluğudur. Çoğu kurum her varlık sınıfı için bir kayıt tipi tarifler. Buna göre eğer **Sipariş** isimli bir varlık sınıfı var ise buan ait bir **sipariş kayıdı** tariflenmelidir. Her kayıt tipini tarif eden metadata'nın veri rehberinde yer alması gerekmektedir. Bu metadata, kayıt ismi, katıt tarifi, büyüklüğü (uzunluğu) verinin parçalarını , veri kümesini , **birincil** ve **ikincil anahtarlarının (primary / secondary key)** tanımını içermelidir.

Varlılları tanımlayan kayıtların yanı sıra varlıklar arası birlik/bağlantıları tanımlayan kayıtlar da vardır. Bu tip kayıtlar genelde bağlantılı varlıkları ilgilendiren durumlarda oluşturulur.

Anahtar (key), kayıdı belirleyen bir veri parçasıdır. Birincil (primary) ve ikincil (secondary) olmak üzere temel iki anahtar mevcuttur.

Birincil anahtar, kayıdı tek başına belirleyen bir veri parçasıdır. Herhangi bir kayıt için birincil anahtar o varlığı gerçek dünyada tanımlayan bir belirleyicidir. Örneğin **Öğrenci** için, **Öğrenci numarası** belirleyici birincil anahtardır. Belirleyici bilgilerin artması ile bir kayıdı belirleyecek pekçok birincil anahtar olması doğaldır. Bazen iki yada daha fazla veri parçası bir kayıdı belirlemekte kullanılabilir.

İkincil (secondary) anahtar normalde tek başına bir tek kayıdı berlirlemekten ziyade benzer özelliklere sahip bir grup kayıdı belirler. Örneğin, **Branş** isimli veri parçası **Öğrenciler** için ikincil bir ahantardır. Zira öğrenecilerin bir kısmı mühendislik branşında başka bir kısmı tıp branşında öğrenim görebilmektedir. Bundan dolayı ikincil anahtarlar veriler belli bir kategori altında toplanmak istendiğinde son derece yararlıdır.

Veri

Şekil 2.1 den de belli olduğu gibi son soyutlama veridir. Gerçek dünyadaki her varlık için, o varlığı ifade eden, tarifleyen veri parçalarından oluşan bir kayıt yapısı mevcuttur. Kelebek Mobilya örneğinde hareket edilirse; Çalışan isimli varlık sınıfında 50 çalışan mevcuttur. Buna göre veri tabanında bu 50 işçi içinde birer kayıt bulunmaktadır. Buna rağmen kayıt yapısının nasıl olduğuna dair tek metadata mevcuttur.
Dosya (File), belirli kaytı yapısında kayıtların toplandığı bir yapıdır. Öreneğin Kelebek Mobilyanın İşçi dosyasında orada çalışmakta olan 50 işçiye ait kayıtlar bulunmakatadır. Metadata (veri tarifi) ile veri arasındaki farkı vurgulamak gerekirse ; Metadata hiçbir zaman veri tabanında yer almaz , benzer olarak hiçbir kullanıcı verisi de veri rehberinde saklanmaz. (Bazı modern veri yönetim sistemleri, aynı sorgulama dilini kullanarak kullanıcıların veri rehberlerine erişmesine de müsade etmektedir.)

Dosya iki boyutlu bir dizi (matris) olarak da görülebilir. Şekil 2.2 de bunun bir örneği görülmektedir. Burada Kelebek Mobilya 'ya ait bilgiler bir tablo olarak gösterilmiştir. Bu tabloda yer alan veriler üretim varlık sınıfına dahildir. Her kolon bir veri parçasina ait bilgileri içermektedir. Tablonun üstünde bulunan veri isimleri, mamül niteliklerine karşı gelmektedir. Her satır, bir kayıdı ifade etmektedir.

Mamül kayıtları için birincil anahtar mamül numarasıdır. Özel bir mamül numarası (mesela 1250) sadece bir kayıtı ifade etmektedir. **Cinsi, Kaplama** ve **Depo** isimli veri parçaları ise ikincil anahtar olarak tariflenmiştir. Bu anahtarlardan herhangi birinin özel bir hali mevcut kayıtlardan bir alt kümeye

karşı gelebilecektir. Mesela **Cinsi** masa olan kayıtları incelediğimizde bu sınıflamaya 100 ve 350 mamül numaralı kayıtlar girmektedir.

ŞEKİL.2.2 **VERİ PARÇALARI**

MAMÜL NO	CINSI	KAPLAMA	DEPO	FIYAT(USD)
0100	MASA	MEŞE	DR	500
0350	MASA	AKÇAAĞAÇ	DR	625
0625	SANDALYE	MEŞE	DR	100
0975	BÜFE	ÇAM	FR	750
1000	DOLAP	KİRAZ	BR	800
1250	SANDALYE	AKÇAAĞAÇ	LR	400
1425	KİTAPLIK	ÇAM	LR	250
1600	KONSOL	HUŞAĞACI	BR	200
1775	DOLAP	ÇAM	BR	500
2000	BÜFE	MEŞE	LR	1200

BİRİNCİL ANAHTAR İKİNCİL ANAHTARLAR

2.4 VERİTABANI NELER YAPABİLİR, NELER YAPAMAZ ?

Günümüzde hepimizin hayatına şu veya bu şekilde bir veritabanı girmektedir. Örneğin, posta ile elimize geçen telefon veya elektrik faturalarının hazırlanmasında dahi işe bilgisayarlar tarafından yönetilen veritabanları girmektedir. Bu şekilde belki de şu anda aklımıza dahi gelmeyen günlük hayatımızdaki pek çok konuda veritabanları ile karşılaşmak artık doğal hale gelmiştir.

Veritabanlarından en çok beklenen, istenen zamanda istenen bilgiyi doğru olarak kullanıcıya sunmalarıdır, halbuki burada unutulan bilgisayarların aslında insanlar tarafından yazılan programların dışına çıkamayan ve bu programlarda belirtilen komutları hep aynı düzende ve işleyiş sırasında yerine getirdikleridir. Teknik olarak bilgisayarlar çok seyrek hata yaparlar, fakat bilgisayarlarda çalışan programları insanlar yazdıklarına ve bilgisayarlar da bu programı işlediklerine göre, programcının düşüncesindeki veya düşündüklerine programa aktarmasındaki herhangi bir hata bilgisayarın da yanlış çalışmasına sebebiyet verecektir. Veritabanları ile çalışırken problemler daha çok verilerin veritabanına aktarılması sırasında oluşmaktadır. FoxPro bu konuda programın imkanları dahilinde yapılabilecek testleri yaparak, mümkün olduğunca hata yapılmasını engellemeye çalışır. Örneğin, tarih bilgisi olarak,

"13/32/00" gibi bir değer girilmesine engel olacaktır, ama kullanıcı gerçekte "01/01/88" olan tarih için yanlışlık sonucu "01/01/87" girdiğinde FoxPro'nun burada yapacak birşeyi yoktur.

Elektronik bir veritabanı bilgiye çok hızlı erişilmesini ve bu bilginin yine büyük bir hızla değiştirilebilmesini mümkün kılmaktadır. Örnek olarak bir veritabanı aracılığıyla kısa bir zaman diliminde, ödeme tarihi gelmiş çeklerin sahiplerinin isimleri, adresleri tesbit edilebilir ve bir rapor halinde kağıda bastırılabilir. Bunun için bilgisayar, o günün tarihini, veritabanında kayıtlı olan bilgilerdeki tarihler ile tek tek karşılaştırarak, aradaki farkı hesaplar ve bu fark eğer belli bir değerden büyükse, o bilgiyi rapora aktarır.

Çeklerin kontrol edilmesi örneği, iş hayatında işlerin elektronik veritabanları kullanılarak kolaylaştırılmasına verilebilecek yüzlerce örnekten sadece bir tanesidir. Diğer bir güzel örnek de, bilgisayar aracılığı ile yürütülen bir sipariş sistemidir. Elle ile defterler tutarak bir sipariş sistemi yürütmektense, bütün ürünler bir veritabanına girilebilir ve böylece sipariş geldiğinde birim fiyat ile istenen miktar çarpılarak fatura bilgisi elde edilirken diğer bir taraftan da stok bilgisi o anda güncellenerek, stoğun durumu takip edilebilir ve gerekiyorsa adedi belli bir mikatrın altına inmiş mamüller firmalardan sipariş edilebelir. Veya talep miktarına göre hangi mamüllerden fazla hangilerinden az sipariş girilmesi gerektiğine karar verilebilir. Bilgisayar aracılığı ile veritabanı üzerinde yapılacak işlemler, elle yürütülen bir sisteme göre daha hızlı ve doğru olacaktır.

Oluşturulacak veritabanı sisteminin karmaşıklığı tamamen hangi uygulamada kullanılıcığına ve bu uygulamadan neler beklendiğine bağlı olarak değişmektedir. Çok basit uygulamalarda kullanıcı veritabanı ile doğrudan bağlantılı çalışabileceği gibi, daha geniş kapsamlı uygulamalarda kullanıcıya pek yapacak birşey düşmemekte ve sistem akaplı bir çevrim halinde çalışmaktadır. FoxPro her iki türde de uygulama geliştirilmesine imkan tanımaktadır.

2.5 PC, MINI ve MAINFRAME VERİTABANLARI

İş dünyasında ihtiyaç duyulan istekler, kişisel bilgisayar, mini bilgisayar veya mainframe üzerinde çalışabilen veritabanlarının mevcudiyetini gerekli kılmaktadır. Bu üç tip bilgisayar sistemi, kapladıkları alan, çalışma hızları ve fiyatları açısından birbirlerinden ayrılmaktadırlar. Doğru tipteki bilgisayar sisteminin seçilmesi için temel kriterler, uygulamaya konulacak sistemin kapsamının ve boyutlarının doğru olarak belirlenmesine dayanmaktadır. Örnek olarak, daha önce bahsedilen sipariş sistemini göz önüne alacak olursak, küçük bir firma için başlangıçta, PC üzerinde çalışabilecek bir veritabanı yeterli

olacaktır. Firma büyüdükçe eklenen departmanlar için yeni veri giriş elemanları ve bu elemanların girişlerini yapabilmeleri için kullanacakları PC'ler ve bu PC'lerin birbilerine bir bilgisayar ağı ile bağlı olmaları ihtiyacı ortaya çıkacaktır.

Belli bir süre sonra bu sayı 20'leri bulunca hatta aşınca, veritabanı bir PC'nin halledebileceği sınırları aşacağı için, mini bilgisayar sistemlerine yönelmek gerekli olacaktır. Fakat bu seçimle beraber, firmanın kendi içinde bilgiişlem ihtiyaçlarını giderebilmesi için veri giriş elemanların haricinde ayrı bir birim kurması ve gerekiyorsa eğitmesi ihtiyacı da oluşacaktır. Bu ilerlemenin bir adım ötesinde ise mainframe bilgisayarları gündeme gelecektir. Bu durumda da ayrı bir bilgiişlem personeli kaçınılmazdır.

Bir PC sistemi, fiyat - performans açısından bakıldığında, veritabanı üzerinde gerekli testlerin ve denemelerin yapılması için en uygun platformdur. Fakat PC'nin işlem kapasitesini aşacak uygulamalarda, son adım için planların mini veya gerektiğinde mainframe bilgisayarları için yapılması uygun olacaktır.

2.6 VERİTABANI NASIL PLANLANMALI ?

Manuel olarak yürüyen bir sistemin bilgisayar destekli hale getirilmesi kararı bu işlem için yapılması gerekli diğer işler de göz önüne alındığında çok küçük bir adım olmaktadır. Esas çalışma ve emek çalışan sistemi bilgisayar ile çalışır hale getirmek için gerekli adımların teker teker uygulanması ve arada çıkacak problemler ile başedilmesi ve ayrı ayrı parçaların bir arada uyum içinde bir bütün olarak çalışmasının sağlanması için verilmektedir.

Sadece isteklerin belirlenmesi ve bunların kağıda dökülmesi bile birkaç hafta sürecek bir işlemdir. Daha sonra bu isteklerin iyice netleştirilmesi, öncelik sırasına konması ve ne sıklıkta ihtiyaç duyulacağının belirlenmesi gereklidir. Mesela bir istek listesi hazırlamak için basitçe şu sorulara cevap aranabilir:

1. Mevcut durum şu anda nasıl değerlendirilmekte ?
2. Gerekli sonuçları elde etmek için ne gibi verilere ihtiyaç var ?
3. Veri, veritabanına aktarılabilecek şekilde mevcut mu ?
4. Ne gibi raporlara ihtiyaç duyuluyor ?
5. Manuel sisteme göre ne gibi gelişmeler sağlanabilir ?

2.7 VERİTABANI PLANLAMASINA KİMLER KATILMALIDIR ?

Bilgisayarlı bir veritabanının başarılı olması pek çok faktöre bağlıdır. Bu faktörlerden bir tanesi de hiç şüphesiz bu veritabanını kullanacak olan personeldir. Bilgisayarlı bir veritabanı sistemine geçilmesi isteği coğunlukla yönetim seviyesinden, özellikle genel müdürden gelir. Genel müdürün tüm işlem içindeki rolü küçümsenmemeli ama aynı zamanda sistemin çalışmasıyla hergün içiçe olan değişik departmanlardaki personele de konuya interaktif olarak katılma hakkı verilmelidir.

Kurulacak sistem hakkında, manuel sistemi hergün kullanan kişilere danışmanın iki temel faydası olacaktır. Birincisi bu kişiler manuel sistem ile hergün içiçe oldukları için sistemin zayıf ve kuvvetli taraflarını iyi bileceklerdir. İkincisi varolan sistemin geliştirilmesi için yönetimin aklına gelmeyecek fikirlere sahip olabilirler. Ayrıca bu kişilerin yeni sistemi beğenmeleri, yeni sistemin daha çabuk oturmasını sağlayacaktır.

Önemli bir nokta da sadece, sistemin içindeki insanlarla değil, aynı zamanda kurumla iş yapan, ihtiyaçlarını sağlayan veya ürettiklerini kullanan diğer kurumlar ile de görüşülmesi gerektiğidir. Kurum içindeki veri akışı da iyice incelenmeli, kullanılan formlara bakılmalı, bunlardaki gereksiz veya tekrarlanan alanlar indirgenmeli ve veri akışı olabildiğince elektronik hale getirilmeye çalışılmalıdır.

Eğer bu işlem çok büyük bir kurum içinde gerçekleştirilecekse, özellikle bazı işlerin bu kurumdaki departmanlar arası kordinasyon eksikliği sebebiyle tekrarlanmaması, ihtiyaç duyulan bilgiler zaten elektronik olarak mevcut veya bir departman tarafından üretiliyorsa, o bilgilerin kullanılacakları diğer departmanlara yine elektronik olarak aktarılmasının sağlanması da diğer önemli faktörlerdendir.

2.8 PROJE ZAMANLAMASI

Profesyonel bir sistemin oluşturulması belli bir zaman alacak bir iştir. Ne kadar süreceği özellikle projenin karmaşıklığı ile doğrudan ilişkilidir. Örneğin, sadece isim, adres ve buna bağlı verilerin girileceği ve daha sonra istendiğinde sıralı bir şekilde kağıda basılacağı bir veritabanını oluşturmak sadece birkaç saat alabilir. Halbuki manuel bir sistemin baştan aşağı bilgisayara dayalı bir veritabanı sistem ile değiştirilmesi işlemi ise çok uzun bir zaman dilimi alacak bir iştir. Sadece bu proje için gerekli araştırma ve inceleme çalışmalarının yapılması bile haftalar sürecektir. Bu şekilde oluşturulan sistem operasyona alındığı andan itibaren tekrar gözden geçirilmeye başlanmalı ve gerekli revizyonlar

yapılmalıdır. Bilgisayara dayalı bir veritabanının oluşturulması için gerekli aşamaları şöyle sıralayabiliriz:

1. Problemin belirlenmesi

Öncelikle problem açık bir şekilde tarif edilmeli, çözümü için nelerin yapılması gerektiği açıkça belirtilmelidir. Problemin oluşmasına sebebiyet veren temel faktörler iyice analiz edilmelidir.

2. Varolan kaynakların incelenmesi

Problemin tanımından sonra, çözülebilir bir problem olup olmadığı incelenmelidir. Eğer çözülebilir bir problem ise gerçekten çözümü için bilgisayara ihtiyaç duyulup duyulmadığına karar verilmelidir.

3. Bilgisayarla problemin nasıl çözüleceğinin belirlenmesi

Bilgisayarlı bir çözüm için karar verildikten sonra, detaylı bir plan ile adım adım çözümün nasıl olacağı belirlenmelidir.

4. Sistemin oluşturulması

Bu adımda sistemin gerçeklenmesine başlanabilir. Veritabanları oluşturulur ve programların yazılmasına başlanır.

5. Sistemin test edilmesi

Oluşturulan sistemin, gerçek ortama uygunluğu ve gerçek veriler ile nasıl çalıştığı çok dikkatli bir şekilde test edilmelidir. Buradaki önemli nokta testlerin kesinlikle kritik noktalarda da başarılı olduğunu görmeden varolan sistemin yenisi ile değiştirilmemesi gerektiğidir. Manuel ve bilgisyarlı sistemler bu testler bitene kadar paralel işletilebilir.

6. Sistemin devreye alınması

Test sonuçlarının tatminkar olduğuna karar verdikten sonra, bilgisayara dayalı sistem tamamen devreye alınmalı ve eski manuel sistem devre dışı bırakılmalıdır.

7. Sistemin izlenmesi ve gözden geçirilmesi

Sistemin operasyonel olmasında yaklaşık altı ay kadar sonra proje tekrar gözden geçirilmeli, zayıf ve kuvvetli olduğu noktalar tekrar incelenmeli gerekiyorsa değişiklikler yapılmalıdır.

2.9 VERİ-GİRİŞİ YÖNETİMİ

Daha önce de belirtildiği gibi bilgisayara dayalı bir veritabanı sistemi ancak veritabanına girilen verilerin doğruluğu kadar doğru ve düzgün çalışabilir. Genelde veri girilmesi işleminin personelden herhangi biri tarafından yapılabileceği hatasına düşülür ve bunun sonucunda da pekçok hata oluşabilir. Veri giriş işlemi iki adım olarak düşünülebilir:

* Verinin toplanması
* Toplanan verilerin bilgisayara girilmesi

Her iki adım da aynı ölçüde önemlidir. Verilerin bilgisayara girilecek hale getirilmesi dikkatli yapılması gereken bir işlemdir.

BİLGİLERİN HAZIRLANMASINDA İZLENECEK YOLLAR:

1. Bilginin toplanması için yeterli zaman verilmesi gereklidir.
2. Bilgiler girilmeden önce muhakkak gözden geçirilmelidir.
3. Bilgiler ile ilgili tüm sorular, girişten önce cevaplanmalıdır.
4. Bu işlemler sırasında iyi bir dökümantasyon yapılmalıdır.

Öncelikle ne kadar bilgi toplanacağı ve ne kadar detaya inileceği belirlenmelidir. Bu soruların cevabı bilgilerin toplanması için ne kadar zaman ayrılması gerektiğini belirleyecek faktörlerden biridir. Veri girişinin nasıl ve kimin tarafından yürütüleceği bu işlemin başarısını belirleyecektir. Bilgisayara bilgi girişini yapacak personelin manuel sistemde de bu bilgilerle uğraşan kişiler arasından seçilmesi ve eğitilmesi de yararlı olacaktır.

VERİ GİRİŞİ PERFORMANSINI ARTTIRACAK ÖNLEMLER:

1. Amaçlar ufak ve erişilebilir seçilmelidir.

Kişilerin cesaretini kırmamak için, veri girişi yapacak insanları bir gün içinde yüzlerce sayfa bilgi ile karşı karşıya bırakmak yerine girilecek veri daha ufak ve gün içinde bitirilebilir parçalara ayrılmaldır.

2. Girilen verilerin sık sık kontrol edilmesi.

Herkes hata yapabilir. Bu şartlar altında bir taraftan veri girişi yapılırken diğer taraftan da girilen verilerin ayrı bir personel tarafından kontrol edilmesi oluşabilecek yazım ve giriş hatalarını en aza indirecektir.

2.10 HATALARIN KOLAY BELİRLENME YOLLARI

Manuel sistemlerin en ekonomik ve uygulanabilir olduğu günlerden bu güne iş dünyasında pek çok değişiklik olmuştur. Hala daha pek çok şirket veri girişi için ayrı bir departman ve bununla birlikte ayrı personel çalıştırmasına rağmen, pek çoğu da, veriyi daha üretildiği noktada elektronik olarak girme ve böylece daha rahat kullanma yolunu seçmektedir.

Klavye aracılığıyla bilgilerin aktarılmasına alternatif olarak son zamanlarda özellikle sesli veya scanner'lı veri giriş sistemleri oldukça yaygın hale gelmişlerdir. Scanner'lar hem kullanım kolaylığı hem de hemen hemen her alanda rahatça uygulanabilir olmaları açısından önümüzdeki birkaç yıl kadar kısa bir süre içinde vazgeçilmez hale geleceklerdir.

Diğer bir nokta da elde edilen verinin nasıl saklanacağıdır. İki alternatif söz konusudur: resim veya yazı. Resim olarak saklama durumunda, scannerdan gelen bilgi üzerinde herhangi bir işlem yapılmaz ve olduğu gibi bilgisayara aktarılır, halbuki aynı bilgiyi yazı olarak saklamak için elde edilen resimin incelenmesi ve ondan anlamlı bir bilginin elde edilmesi gereklidir. Bu işlem çoğu zaman uzun ve zahmetli olduğu gibi, yüzde yüz doğru sonuç da vermeyeceği için yazı olarak saklamak daha yararlı olacaktır.

Manuel olarak veri girişi yapan şirketlerde aşağıdaki basit yöntemler kullanılırsa, basit giriş hataları en aza indirgenebilir:

1. Veri girişinden önce belli giriş şekillerinin tesbit edilmesi
2. Otomatik seçim tablolarının oluşturulması
3. Elektronik olarak girilen verilerin, kayıt öncesinde kontrol edilmesi.

BİR VERİTABANI SİSTEMİ NİÇİN KULLANILIR ?

AVANTAJLARI NELERDİR ?

Bu sorunun cevabı bir ölçüye göre sistemin çok kullanıcılı veya tek kullanıcılı olmasına göre değişir. Çok kullanıcılı sistemim tek kullanıcıya göre daha çok avantajları vardır. Bu yüzden önce tek kullanıcılı sistemleri inceleyelim.

Şekil 1.1 deki *Ö RETMEN* veritabanına geri dönelim. Bu veritabanında Bilgisayar bölümündeki öğretim üyelerinin bilgileri saklanmaktadır. Bu veritabanı tek kullanıcılı ve çok basit bir sistem olduğu için veritabanı sistemi kullanmanın getirdiği avantajlar kolayca göze çarpmaz. Şimdi de buna benzer bir *Ö RENCİ* veritabanını ele alalım. Bu veritabanında yaklaşık on bin öğrencinin yeraldığını düşünürsek herhangi bir öğrencinin bilgilerine erişilmesi işlemi çok sık tekrarlanacaktır. Bu veritabanı çok büyük olsa bile tek kullanıcılı bir durumdur. Bir veritabanı sisteminin geleneksel-kağıt üzerinde tutulan kayıt sistemine karşı getirdiği avantajlar bu örneklerde daha kolay göze çarpmaktadır. Bu avantajların bazıları aşağıda listelenmektedir.

- **Compactness**

  Büyük hacimli dosyalara gerek kalmaz.

- **Hız (Speed)**

  Bilgisayar veriye erişim ve değiştirme işlemini insaların yapabileceğinden daha hızlı yapar . Mesela 'Elimizdeki bal miktarı peynirden fazla mı ?' sorusu görsel veya elle herhangi bir araştırma yapmadan cevaplandırılabilir.

- **Daha az ağır ve sıkıcı iş(Less drudgery)**

  Dosyaların güncellenmesi için sürekli tekrar edilmesi gereken işlemler bilgisayar tarafından daha kısa zamanda yapılabilir.

- **Geçerlilik(Currency)**

Verinin güncellenmesi işlemi istenen herhangi bir zamanda ve doğru bir şekilde gerçekleştirilir.

Yukarıda açıklanan avantajlar, tek kullanıcılı veritabanlarından daha büyük ve karmaşık olan çok kullanıcılı veritabanları için daha çok geçerlidir. Ancak veritabanlarının çok kullanıcılı sistemler için getirdiği en büyük avantaj verilerin ***merkezi kontrolüdür(Centralized Control)***. Herhangi bir uygulamada her programın kendi dosyalarının disk veya teyp birimleri üzerinde dağınık olarak yer alması bu verinin kontrolünü zorlaştırmaktadır. Ancak bir veritabanı sistemi tüm dosyalar aynı veritabanına bağlı olarak değişik birimler üzerinde saklansa da kolayca kontrol imkanı verir.

Veritabanı Yaklaşımının Yararları

Veritabanının merkezi kontrol kavramının getirdiği avantajlar şunlardır.

Veriler arasında tekrarlar önlenir

Birçok işletmede her uygulamada kullanılan veriler ayrı dosyalarda saklanmaktadır. Ancak veritabaı sistemi kullanıldığı zaman bu verilerin ayrı dosyalarda aynen veya küçük değişikliklerle tutulmasından dolayı ortaya çıkan yer kaybı ve veri tekrarrlanması sorunu önlenmiş olur. Mesela bir işletmenin personel ve personel eğitimleri bilgilerini ele alalım. Bu uygulamaların gerektirdiği dosyalarda ortak kullanılan bilgiler yer almaktadır. Bölüm 1.2.1 de bahsedildiği gibi bu iki işlem birbirine entegre edilmiş dosyalar kullanılarak yapılırsa tekrarlar önlenmiş olur. Ancak bazı uygulamalarda veri tekrarlanması kaçınılmazdır. Bu durumda tekrarlanan verilerin dikkatli bir şekilde gözden kaçırılmadan güncellenmesi gerekir.

Veriler arası uyuşmazlık ortadan kalkar

Veritabanına aynı konu veya kişiyle ilgili kaydın iki kez tekrarlanarak girildiğini varsayalım. Ve veritabanı yönetim sisteminin bu tekrarları kontrol etmediğini düşünelim. Bu durumda bu kayıtla ilgili herhangi bir değişiklik yapılmak istendiğinde sadece biri üzerinde işlem yapılırsa yani aynı kaydın bir yerdeki değeri farklı diğer yerdeki değeri farklı ise veritabanı tutarsız durumdadır(Inconsistent State). Bu koşullarda kullanıcıya yanlış, eksik hatta karşıt bilgi verilebilir. Eğer herhangi bir konu veya kişiye ait tek kayıt varsa veya tekrarlar ortadan kalkmadığı halde bunlar üzerinde kontrol sağlanmışsa

veritabanı hiçbir zaman tutarsız durumda kalmaz. Kullanıcının yaptığı değişiklikler otomatik olarak güncellenir.

Bu işleme Yayılmış Güncelleme (Propagating Updates) işlemler adı verilir. Güncelleme işlemleri kayıt ekleme, silme ve değişiklik yapma gibi işlemlerin hepsini kapsar. Bununla birlikte az sayıda ticari sistemde, eğer tekrarlar kontrol edilmiyorsa yayılmış güncelleme işlemleri otomatik olarak gerçeklenir.

Veri değişik birimler tarafından paylaşılabilir

Aynı verinin farklı kısımlarının değişik birimler tarafından kullanılması veya aynı tür verilerin farklı işlemlerde kullanılması bilgisayarda saklanan verilerin paylaşılabilir olma özelliğinin getirdiği sonuçlardır. Ayrıca yeni uygulamalarda da önceden depolanmış veriler kullanılarak sürekli ayni dosyaların oluşması önlenir.

Veri üzerinde standartizasyon gerçekleştirilir

Veritabanı töneticisi (database administrator) verinin gösterimiyle ilgili çeşitli standartlar koyabilir. Uygulamada konulacak standartlar, toplu, yüklemeyle ilgili, bölümle ilgili, endüstriyel, ulusal ve uluslararası standartlar olarak gruplanadırılabilirler. Saklanan veri formatı üzerinde bir standart belirlemek özellikle verinin değişik sistemler arasında geçişinin yapılması ve veri değiş tokuşu için gereklidir. Aynı zamanda verinin isimlendirilmesinde ve dökümantasyonunda birtakım standartlar geliştirilmesi verinin değişik birimler arasında paylaşılması ve anlaşılması açısından önemlidir.

Veri üzerinde güvenlik önlemleri alınır

Veriye erişimin kontrol edilebilmesi, önemli verilere erişimde yetkilerin verilebilmesi ve bu önlemlerin şifrelerle desteklenebilmesi bir veritabanı sisteminin getirdiği özelliklerdir. Veritabanı yöneticisi sistem üzerinde tüm yetkilere sahip olan bir kişi olarak veritabanına erişimin uygun kanallar üzerinden yapılamsını sağlayabilir. Özel bilgilere erişimde belli kişilere, belli işlemlere, ve bazı veri parçaları üzerinde yetki verilerek her kullanıcının her yere erişmesi önlenmiş olur. Eğer bu gibi önlemler alınmazsa bilgisayarda kurulan veritabanı sistemi, geleneksel veritabanı sistemlerinden daha fazla güvenlik riski altında kalır. Veritabanı sisteminin merkezi yapısı iyi bir güvenlik sisteminin geliştirilmesini gerektirir.

Verinin doğruluğu ve bütünlüğü kontrol edilebilir

Bütünlük özelliğinin problemi veritabanına kayıtlı bilgilerin doğru olması gereğidir. İki ayrı kayıt arasında aynı bilgiler arasında tutarsızlık olması bütünlüğün bozulduğunu gösterir. Bu durum sadece dpolanmış veri üzerinde değil aynı zamanda tekrarlar olduğu zamanda ortaya çıkar. Hatta tekrarlar olmasa bile bazı hatalı veriler oluşabilir. Daha veri girişi sırasında verinin ve yapılan işlemlerin doğruluğu başka bir deyişle geçerliliği kontrol edilebilir. Bazı verilerin, boş-0 değer taşıması, karakter ve nümerik bilgiler arasındaki karışıklıklar, yanlış tarih girişleri, nümerik değerlerde basamak hataları kontrol edilerek geleneksel yöntemlerle kontrol edilemeyen kullanıcı hataları ortadan kalkar. Mesela bir personelin haftada 40 yerine 400 saat çalışması veya olmayan bir bölümde çalışması gibi yanlış bilgiler veritabanında yer alabilir. Bu bilgiler biraz önce belirtildiği gibi daha güncelleme işleminin başında kontrol edilirse bu sorun ortadan kalkmış olur. Veri bütünlüğü özellikle çok kullanıcılı sistemlerde çok önemlidir. Çünkü bu tür sistemlerde bir kullanıcının yaptığı hatalı güncelleme işlemi başka bir kullanıcın yaptığı işi bozacak yanlış veri üretebilir. Bir çok veritabanı yazılımı bu konuda yetersiz kalmakta bu yüzden veri bütünlüğü konusunda birçok çalışma yapılmaktadır.

Çakışan istekler dengelenebilir

Herhangi bir kuruluşun tüm ihtiyaçları (şahısların kişisel istek ve ihyiyaclarına karşın) veritabanı yöneticisi tarafından bilinir. Veritabanı yöneticisi tüm sistemi bu istekleri dengeleyecek şekilde dizayn eder. Mesela bir grup veriye hızlı erişim yapmak daha önemliyse bu yüzden başka uygulamalarda aynı veriye daha düşük performansla erişim göz önüne alınmayabilir.

Veri Bağımsızlığı

Bu konuda yukarıda bahsedilen avantajlardan biri olmasına rağmen bu kitapta detaylı olarak bahsedilmeyecektir. Ancak uygulama programı yazımında verinin tipinin değiştirilmesinin uygulamanın yapısını değiştirmemesi şeklinde tanımlanabilir.

3.BÖLÜM

FOXPRO ve DİĞER VERİTABANI YAZILIMLARI

Veritabanı dünyası bugün bir yol ayrımına gelmiştir. Hergün gelişmeler olmakta ve bu endüstri sınırları belirsiz bir yöne doğru gitmektedir. Bununla birlikte günümüzde bu konuda pekçok seçenek karşımıza çıkmaktadır. Ayrıca bu dünyada *4. Nesil Dilleri (Fourth Generation Languages)*, İlişkisel Veritabanı Yazılımları dikkati çekmektedir. Bu konuların bilgisayar donanımıyla ilişkili ve proje ihtiyaçlarına uygun olarak ele alınması gerekmektedir. Dördüncü Nesil Dillerinin düşündüğümüzden daha hızlı ilerlemesi her problemi üçüncü nesil dilleriyle çözmeye çalışan programcıların kendilerini yabancı hissetmelerine sebep olmaktadır.

Bu çalışmada tüm veri tabanı yazılımları listelenmeyecektir. Ancak Foxpro'nun diğer diller arasından niye seçildiğini anlatmak için bu veritabanı yazılımlarından üç tanesi örnek olarak alınacak ve birbirleriyle karşılaştırılacaklardır.

PARADOX

Paradox, kişisel bilgisayarların çok yönlülüğünü ortaya çıkartan kolay kullanımlı ve çok güçlü bir veritabanı yazılımıdır. Paradox ile yapmak istediğimiz işlemleri otomatikleştiren basit veya karmaşık sistemler geliştirilebilir. Paradox SCRIPT isimli gereci yardımıyla LOTUS 1-2-3 teki makrolara benzeyen (bir dizi komutu bir tuşa set ederek bunları tek tuş darbesiyle tekrarlayan) işlemler yapılmasına imkan verir.

Paradox'ta depolanan bilgi sorgulanabilir, üzerinde hesap yapılabilir ve değişik şekillerde düzenlenebilir. Ayrıca Paradox'un menülerle çalışılabilen ortamından veritabanındaki veriler kolayca sorgulanabilir. Aynı şekilde 'what you see is what you get' yaklaşımına bağlı olarak kolayca sistem geliştirilebilir. Mesela bir ekran formu düzenlenirken ekran sonuçta görmek istediğimiz şekilde kolayca hazırlanabilir. Bu arada tüm teknik detaylar Paradox tarafından arka planda idare edilir. Bu şekilde bir veritabanı yazılımı kolayca tasarlanarak birkaç dakika içinde bilgi girişi yapmaya ve sorgulamaya hazır hale gelir.

Paradox'ta kullanıcıya yardımcı olmak üzere geliştirilmiş bir uygulama programı dili ve özel uygulama yazılımı üreteci vardır. Büyük işletmelerde dBase, Lotus 1-2-3, Symphony, Pfs:file, Visicalc gibi popüler

yazılımları bilen elemanlar kolayca bulunurken asıl problem bu dillerin birbirleriyle entegre olarak çalışması sırasında ortaya çıkar. Paradox tüm bu ürünlerin formatlarını okuyabilir ve yazabilir.

Paradox yazılımı, 512 K belleğe, sabit diske, DOS 2.0 veya daha ileri bir işletim sistemine sahip herhangi bir IBM PC uyumlu bilgisayar üzerinde rahatça çalışabilir. 640K Belleğe sahip olamk ise çalışmaları rahatlatacaktır. Paradox belleği geçici bellek ve önbellek olarak iki parçaya ayırarak kullanır. Genelde %25 geçici bellek, %75 ise önbellek olarak kullanılır. Ancak bu oran Paradoxtaki EMS seçeneği kullanılarak değiştirilebilir. Paradox bir sayfada, seçime bağlı olarak 2-2000 satır arasında yazabilir. Yazıcıda otomotik olarak sayfa atlar ve yeni sayfadan yazılmaya başlanır. Bundan sonraki bölümde QUATRO PRO'nun güçlü özelliklerinden ve QUATRO PRO ile PARADOX'un birlikte çalışma şekillerinden bahsedilecektir.

QUATRO PRO

Son yıllarda Borland, programlama dilleri ve uygulama yazılımı üreticilerinin lideri olarak görülmektedir. Quatro Pro ise ürettikleri worksheet yazılımlarının en başarılı olanlarından biridir. Bu yazılım güçlü, değişik uygulamalara kolayca adapte edilebilen, kolay kullanımlı bir yazılımdır. Tuş özelliklerinin anlaşılması ve kullanılımı çok basittir. Bu yüzden sistemi yüklemek ve çalışır hale getirmek çok kısa zamanda gerçekleşebilir. Quatro Pro ile çok karmaşık uygulamlar kısa zamanda yapılabilir.

Tüm worksheet programlarında kullanılan tablo, Quatro Pro'da 255 sütun ve 8192 satırdan oluşur. Bu şekilde toplam 2.088.960 ayrı alana erişilebir. Bu alanlara metin, nümerik değerler veya formüller yazılabilir.

Quatro Pro, not defteri, referans tabloları ve hesap makinası özelliklerinin değişik kombinasyonlarda kullanılmasıyla genel amaçlı bir program gibi düşünülebilir. Bu yüzden finans modelleme, bütçe, tahmin ve analiz gibi birçok alanda kullanılır. Ayrıca istatistiksel ve bilimsel analizlerde de kullanılır. Quatro Pro'nun built-in fonksiyonları ise her çeşit matematik modelleme işlemine uygun kompleks işlemler gerçekleştirilebilir.

Quatro Pro'nun grafik özellikleri de bu alanlarda kullanılılan diğer elektronik çizelge programlarından daha gelişmiştir. Sadece hesaplamalara dayalı grafikler değil aynı zamanda kendi çizim programıyla çizilen resimler, rapor şeklindeki metinler yazının içine konabilir. Slayt gösterileri hazırlanabilir. Ayrıca

başka çizim programlarıyla oluşturulmuş resimler Quatro Pro içine entegre edilebilir. Recordlar üzerinde her türlü arama, sıralama, ekleme, silme işlemleri kolayca yapılabilir. Başka yazılımlarla depolanmış veriler Quatro Pro'nun içine aktarılabilir. Quatro Pro'nun kelime işlemcisi yardımıyla oluşturulan tablolardan ve yapılan hesaplamalardan özet raporlar hazırlanabilir. Makro kaydetme özelliği ile bir işlem grubu tek tuşa basılarak gerçekleştirilebilir. Ayrıca hesaplamalarda 15 basamak hassasiyetle işlem yapılmaktadır. Daha üst basamaklar ise yuvarlatılır.

QUATRO PRO ve PARADOX

Quatro Pro ve Paradox birbirleriyle ilişkili kullanılmak için dizayn edilmiştir. Eğer yeterli bellek varsa her iki program aynı anda kullanılabilir. Aynı daha herikisinde de kullanılır ve birinden diğerine geçiş sağlanabilir.

TİCARİ AMAÇLI İLİŞKİSEL DİLLER

SQL, QBE, QUL gibi diller ***'Sorgulama Dilleri (Query Language)'*** olarak bilinirler. Ancak bunlar veritabanını sorgulamaktan başka veri yapısının taif edilmesi, verinin değiştirilmesi veya veri üzerinde güvenlik önlemlerinin alınması gibi özelliklere de sahiptirler.

Bu kitapta bu dillerin bir örneği olarak SQL dilinden bahsedilecektir.

SQL DİLİNİN ÖZELLİKLERİ

- **Veri Tarif Etme Dili (Data Definition Language- DDL)**

  SQL DDL, tablolar arasında ilişkilerin tarif edilmesini, silinemsini kısaca ilişkilerin güncellenmesini sağlar.

- **İletişimli Veri İşleme Dili (Interactive Data Manipulation Language-DML)**

  SQL, hem relational algebra'ya hemde tupple relational calculus'a destek verir. Veritabanına kayıt ekleme, silme ve güncelleme özelliklerine sahiptir.

- **Katılmış Veri İşleme Dili(Embedded Data Manipulation Language)**

SQL, PL/I, FORTRAN, Pascal,COBOL gibi genel amaçlı dillerde yazılmış programlara entegre edilerek kullanılabilir.

- **Görüş Alanı Tanımlama**

SQL DDl yardımıyla kullanıcının verinin sadece belli bölümlerini görmesi sağlanabilir.

- **Yetkiler**

SQL'de bir veya bir grup kullanıcıya tablo, ilişki veya viewların kullanımıyla ilgili değişik erişim yetkileri verilebilir.

- **Bütünlük**

R Sequel'de karmaşık bütünlük önlemleri için gerekli komutlar yer alamkatadır. SQL'in yeni versiyonları daha çok bütünlük kontrolüne sahiptir.

- **İşlem Kontrolü**

SQL'de herhangi bir işlemin başlangıç ve bitişini seçmek için gerekli komutlarda yeralmaktadır. Birçok uygulama yazılımında IBM SAA-SQL de dahil paralelliğin kontrolü için veri üzerinde kilit mekanizması uygulanır.

SQL 'de TEMEL KOMUT YAPISI

Bir SQL komutunun yapısı aşağıdaki gibidir.

- **SELECT**

- **FROM**

- **WHERE**

SELECT bölümünde yapılan işleme uygun komut ve sorgunun sonunda görüntülenmek istenen alanlar yeralır. Yani relational algepradaki projection işlemidir.

FROM kısmında ise sorgulanacak yada işlem yapılacak tablo seçilir. Yani relational algebradaki cartesian product işlemidir.

WHERE komutu ise sorgulamanın koşulunun seçilmesini sağlar. Yani relatioanl algebradaki selection predicate'tir.

SQL'in GÜÇLÜ ÖZELLİKLERİ

SQL relational algebra'ya uygun işlemlerin kolayca gerçeklenmesini sağlar. Relational algebra'daki cartesien product, projection, algebra selection gibi kavramlar SQL'de From, Select ve Where komutlarıyla özdeştir. Her ikisinde de birleşim ve fark işlemleri yer alır. Sorgulama sonucu elde edilen değerler geçici olarak saklanabilir. Herhangi bir relational algebra komutu SQL'de kolayca ifade edilebilir. SQL bazı ek fonksiyonları ile Relational Algebra'dan daha güçlüdür. Ayrıca yüksek seviyeli FORTRAN, COBOL, PL/I, PASCAL ve C gibi dillerle birlikte kullanılması uygulama alanlarını genişletir.

FOX PRO'nun GÜÇLÜ ÖZELLİKLERİ

FoxPro genel anlamda yukarıda belirtilen veritabanı dillerinden fazla değişik değildir. Ancak bazı özellikleri FoxPro'nun güçlü bir veritabanı dili olamsını sağlar.

1.FoxPro 2.0(X) kullanımı mümkün olan tüm yayılmış belleği kullanır. 32 Bit'lik bir üründür. Foxpronun tüm standart fonksiyonları desteklenir. Ek olarak da aşağıdaki özellikler gelmiştir.

İndex sayısı sadece bellek ile sınırlıdır. Bu standart Foxpro için de geçerlidir.

Açık olan pencere sayısı, browse pencerelerinin sayısı ve bellek değişkenlerinin uzunluğu bellek ile sınırlıdır. Maksimum string uzunluğu 2 GigaByte'a kadar çıkabilir.

2.Bellek

Diğer veritabanı yazılımlarından farklı olarak bellek kullanımında büyük avantajlar getirir. Doğru şekilde düzenlenmiş ve konŞekile edilmiş belleğin tamamını kullanabilir.

FoxPro'nun yüksek hızla kullanılabilmesi için bilgisayarın belleğinin standart 640K limitini aşmış olması gerekir. PÇ sınıfındaki mikrobilgisayarlarda üç tür bellek vardır.

Conventional Memory	(C M)	(Standart Bellek)
Expanded Memory	(EXP.M)	(Genişletilmiş Bellek)
Extended Memory	(EXT.M)	Yayılmış Bellek)

a)Tüm mikrobilgisayarlar standart belleğe (640K'ya kadar) sahiptir. FoxPro'nun standart versiyonu en az 512K standart bellek kullanır. Belleğe yerleşik programlar yüklendikten sonra bunun 420K'ı serbest kalır.

b)Eğer yayılmış bellek kullanılırsa FoxPro iki kat daha hızlı çalışır.

3.SEGMENT LOADER

FoxPro'nun segment loader'ı özellikle network(ağ) ortamında uygulama programı hızını arttırır. Ayrıca kütüphane mekanizması tarafından kullanılan genel bağlantı mekanizmasına destek verir.

4.RUSHMORE TEKNOLOJİSİ

Bu teknoloji bir grup recorda çok hızlı ulaşılmasını sağlayacak(tek record indeksli erişimle kıyaslanacak bir hızla) bir veri erişim tekniğidir. FoxPrp 2.0(x) kişisel bilgisayarlardaki milyonlarca recorddan oluşan veritabanlarına, mainframe veritabanı sistemlerine yakın bir hızla ulaşılmasını sağlar.

5.FOXPRO ve SQL

FoxPro 2.0(x) de aşağıdaki SQL komutları kullanılabilir.

- **SELECT**
- **CREATE TABLE**
- **INSERT**

6. DİĞER ARAÇLAR

FoxPro'nun gücünü destekleyen diğer araçlarıda aşağıda görülmektedir.

- RQBE (Relational Query By Example)
- Screen Builder ve Menu Builder
- Project Manager
- Report Writer

7. GÖRÜŞ ALANI PENCERESİ

FoxPro veritabanları arasında birden çokluya ilişkiler kurabilir.

Bütün yukarıda belirtilenler SİSTEM KAPASİTESİ TABLO'sundan kolayca görülebilir. Sonuçta FoxPro maksimum performansla çalışan bir yazılımdır.

SİSTEM KAPASİTESİ

VERİTABANI VE İNDEKS DOSYALARI

SİSTEM KAPASİTESİ	FOX PRO	GENİŞLETİLMİŞ FOXPRO
Her bir veritabanı dosyasındaki maksimum kayıt sayısı	1 milyar	1 milyar
Her bir kayıttaki maksimum karakter sayısı	4000	4000
Bir keredeki maksimum alan sayısı	25	25
Her bir veritabanı alanındaki maksimum karakter sayısı	254	254
Her bir indeks anhtarındaki maksimum karakter sayısı (. IDX)	100	100
Her bir indeks anhtarındaki maksimum karakter sayısı (. CDX)	254	254
Her bir veritabanı dosyasındaki maksimum açık dosya sayısı	sınırsız	sınırsız
Bütün çalışma bölgelerindeki açık indekslerin maksimum sayısı	sınırsız	sınırsız

ALAN KARAKTERİSTİKLERİ

SİSTEM KAPASİTESİ	FOX PRO	GENİŞLETİLMİŞ FOX PRO
Karkter alanının maksimum büyüklüğü	254	254
Nümerik ve float alanların maksimum büyüklüğü	20	20
Alan adında yer alabilecek maksimum karakter sayısı	10	10

TABLO 3-1-A

HAFIZA DEĞİŞKENLERİ VE DİZİLER

SİSTEM KAPASİTESİ	FOX PRO	GENİŞLETİLMİŞ FOX PRO
Hafıza değişkenlerinin geçerli sayısı	256	256
Hafıza değişkenlerinin makksimum sayısı	3600	36.000
Maksimum dizi sayısı	3600	36.000

RAPOR YAZMA KAPASİTELERİ

SİSTEM KAPASİTESİ	FOX PRO	GENİŞLETİLMİŞ FOX PRO
Bir rapor tanımlamasındakı maksimum obe sayısı	sınırsız	sınırsız
Bir rapor tanımlamasındakı maksimum doğru sayısı	55	255
Gruplama seviyelerinin maksimum sayısı	20	20

PENCERE DESTEĞİ

SİSTEM KAPASİTESİ	FOX PRO	GENİŞLETİLMİŞ FOX PRO
Maksimum açık pencere (her tipte)	sınırsız	sınırsız
Açık izleme pencerelerinin maksimum sayısı	25	25

TABLO 3-1 B

BÖLÜM 4

ÖĞRENCİ OTOMASYON SİSTEMİNİN ÇALIŞMASI

Öğrenci otomasyon sistemi menüsu beş kısımda oluşur:

1_ ÖĞRENCİ ,

2_ DERS ,

3_ NOT ,

4_ BİLGİLER ,

5_ ÇIKIŞ .

MENÜ SİSTEMİ

I. ÖĞRENCİ
- **1. ILK KAYIT**
  - **1_ KAYIT BİLGİLERİ**
  - **2_ SOSYAL DURUM**
  - **3_ NÜFUS BİLGİLERİ**
- **2. YABANCI**
  - **1_ YABANCI BİLGİ**
  - **2_ EMNİYET BELGE**
- **3. ÖĞRENCİ BELGE**
- **4. KARTLAR**
  - **1_ KİMLİK**
  - **2_ PASO**
- **5. DERS**
- **6. DEFTER**
  - **1_ DEFTER GİRİŞ**
  - **2_ BİTİRME PROJESİ**
- **7. KARNE**
  - **1_ DERS TANIM**
  - **2_ KARNE BİLGİLER**
  - **3_ KARNE LİSTESİ**
- **8. DERS DİLEKCESİ**

II. NOT GİRİŞ
- **1_ VİZE**
- **2_ FİNAL**
- **3_ BÜTÜNLEME**

III. BİLGİLER
- **1. ÖĞRETMEN**
  - **1_ BİLGİ GİRİŞ**
  - **2_ LİSTE**
- **2. ASKER**
  - **1_ KİMLİK BİLGİ**
  - **2_ OKUL BİLGİ**
- **3. BÖLÜM**
  - **1_ BÖLÜM TANIM**
  - **2_ BÖLÜM LİSTE**

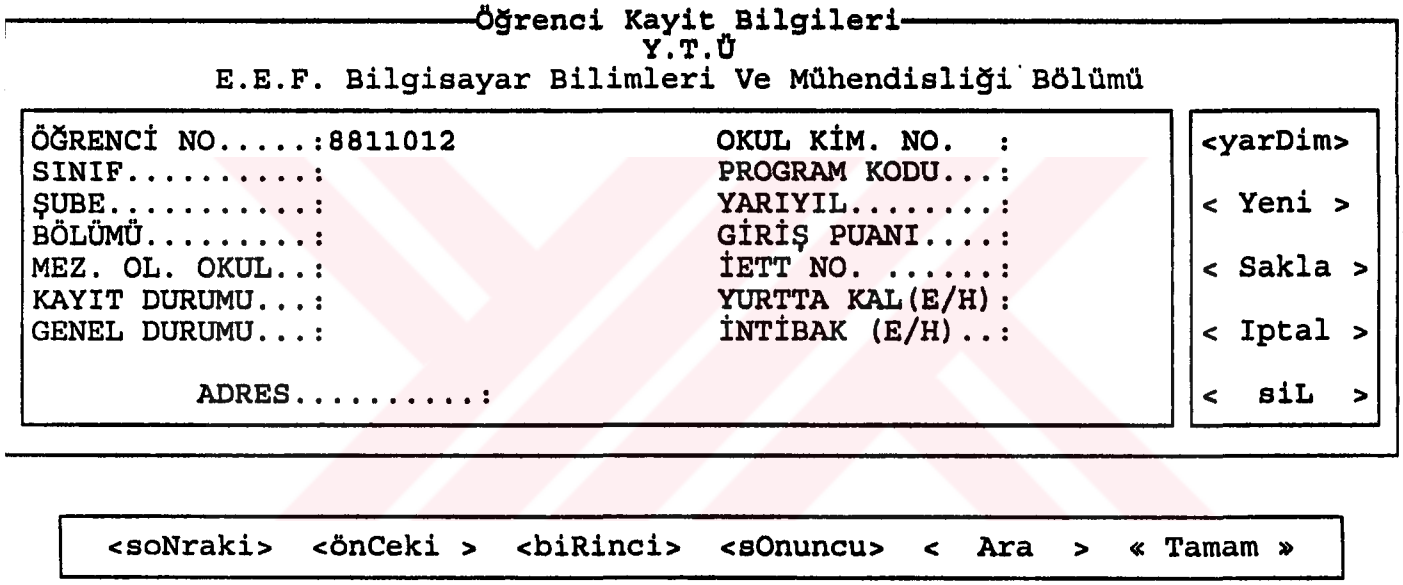
IV. SAAT

V. ÇIKIŞ
- **1_ YENİDEN İNDEKSLEME**
- **2_ SONUÇ**

PROGRAMLARDA KULLANILAN VERİTABANI DOSYALARI

VERİTABANI DOSYALARI	ALAN NO	BÜYÜKLÜK (BYTE)	VERİTABANINA BAĞLI PROGRAMLAR
ASKERLIK.DBF	23	320	ASKIMLIK, ASOKUL
BOLUM .DBF	09	226	BOLUMBIL, BLUMLIST
DERSLER .DBF	31	415	DISRAPOR, DEFTERI
DILEKCE .DBF	18	456	DILEKCE
KIMLIK .DBF	20	310	OGKIMLILK, PASO
NOTLAR .DBF	17	116	VIZE, FINAL, BUTUNLEM, SONUCA
OGRENCI .DBF	59	756	KAYIT, SOSYAL, CUZDAN, OGBELGE
OGRETMEN.DBF	13	444	HOCALAR
SONOT1 .DBF	35	282	KARNEGIR, KARNELST, KARNEKG
YABANCI .DBF	27	392	YABANCI, EMNBELGE

TOTAL	252	3717


Öğrenci Kayit Bilgileri
Y.T.Ü
E.E.F. Bilgisayar Bilimleri Ve Mühendisliği Bölümü
ÖĞRENCİ NO.....:8811012
SINIF.....:
ŞUBE.....:
BÖLÜMÜ.....:
MEZ. OL. OKUL..:
KAYIT DURUMU...:
GENEL DURUMU...:
ADRES.....:
OKUL KİM. NO. :
PROGRAM KODU...:
YARIYIL.....:
GİRİŞ PUANI....:
İETT NO. .....:
YURTTA KAL(E/H):
İNTİBAK (E/H)..:
<yarDim>
< Yeni >
< Sakla >
< Iptal >
< siL >
<soNraki> <önCeki > <biRinci> <sOnuncu> < Ara > « Tamam »

Menü şeridi her bir menü bloğu ile bağlantı kurulamasına olanak sağlar. Bu şerit popup menülerin açılmasına ve istenen menü seçeneğinin seçilerek gerekli işlemlerin yapılmasına yardımcı olur.

Menü şeridinin her her bir bölümü, menü blokları ve seçenekleri aşağıda detaylı olarak açıklanmıştır.

4.1 ÖĞRENCİ MENÜSÜ

ÖĞRENCİ menüsü seçeneklerden ve yeni bilgiler girilmesine olarak veren alt menülerden oluşur.

Dahası yabancı bir öğrenci dosyası üzerinde düzenleme yapılması veya bazı bilgilerin değiştirilmesi de mümkündür. Ayrıca bu menüde bir öğrencinin kimlik kartı basılmaya hazır hale getirilebilir.

4.1.1 İLK KAYIT

İLK KAYIT yeni öğrencilere ait bilgilerin girilmesine ve bu bilgilerin (ÖĞRENCİ.DBF) isimli çalışma alanına kaydedilmesine olanak verilir. ÖĞRENCİ menü bloğundan İLK KAYIT seçilirse üç seçenekten oluşan bir ait menü görünür:

* KAYIT BİLGİLERİ
* SOSYAL DURUM
* NÜFUS BİLGİLERİ

4.1.1.1 KAYIT BİLGİLERİ

KAYIT BİLGİLERİ bir kayıtın alanlarına bilgi girilmesine olanak verir. Bu ekran içinde öğrenci numarasi (Fakülte Numarası), sınıf (Sınıf) ve bölümü (Dalı) gibi bilgiler isteyen alanlar vardır.(Bu ekranın şekli önceki sayfada görülmektedir).

Son alandaki bilgiler girildikten sonra kursör doğrudan bu bilgilerin saklanmasına yarayan < SAKLA > butonuna gider. < ENTER > tuşuna basılarak veya mouse ile bu durumda kayıtta yer alan bilgiler (ÖĞRENCİ.DBF) adlı çalışma alanına kaydedilmiş olur, aynı anda ekranın sağ üst köşesinde YENİKAYIT SAKLANDI " mesajı görülür.

Bu ekranda iki tip kontrol söz konusudur, bir ekranın alt kısmında diğeri de ekranın sağında yer alır.

Birinci kontrol şu seçeneklerden oluşur:

< Sonraki >, < önCeki >, < Birinci >, < Sonraki >, < Ara >, < Tamam >.

Bu seçeneklerin yaptığı işler şöyledir ;

* < soNraki > ... bir sonraki kayıtta yer alan bilgilerin görülmesine olanak verir.

* < önCeki > ... bir önceki kayıtta yer alan bilgilerin görülmesini sağlar.

* < biRinci > ve < sOnucu > sırasıyla kayıtın başındaki ve sonundaki bilgilerin görülmesine olanak verir.

* < Ara > ... en öemli kontrol düğmelerinden biridir. istenen bir kaydin veya kayıtların bulunmasını sağlar. Aynı zamanda bütün mümkün indeks alanlarında indeksleme yapılması mümkündür. Dahası çıkışlar ekranda gösterilebilir veya printerden alınabilir.

Seçilen ekranla gerekli tüm çalışmalarınızı bitirdikten sonra bu aktif ekranı kapatmak için < Tamam > seçeneğini kullanabilirsiniz.

İkinci ve son kontrol tipi ise şu seçenekleri içerir:

< yarDim >, < Yeni >, < Sakla >, < iPtal >, < siL >

* < yarDim > ... Kullanıcının seçilen ekranı kullanmasına yardımcı olacak kuralları içerir.

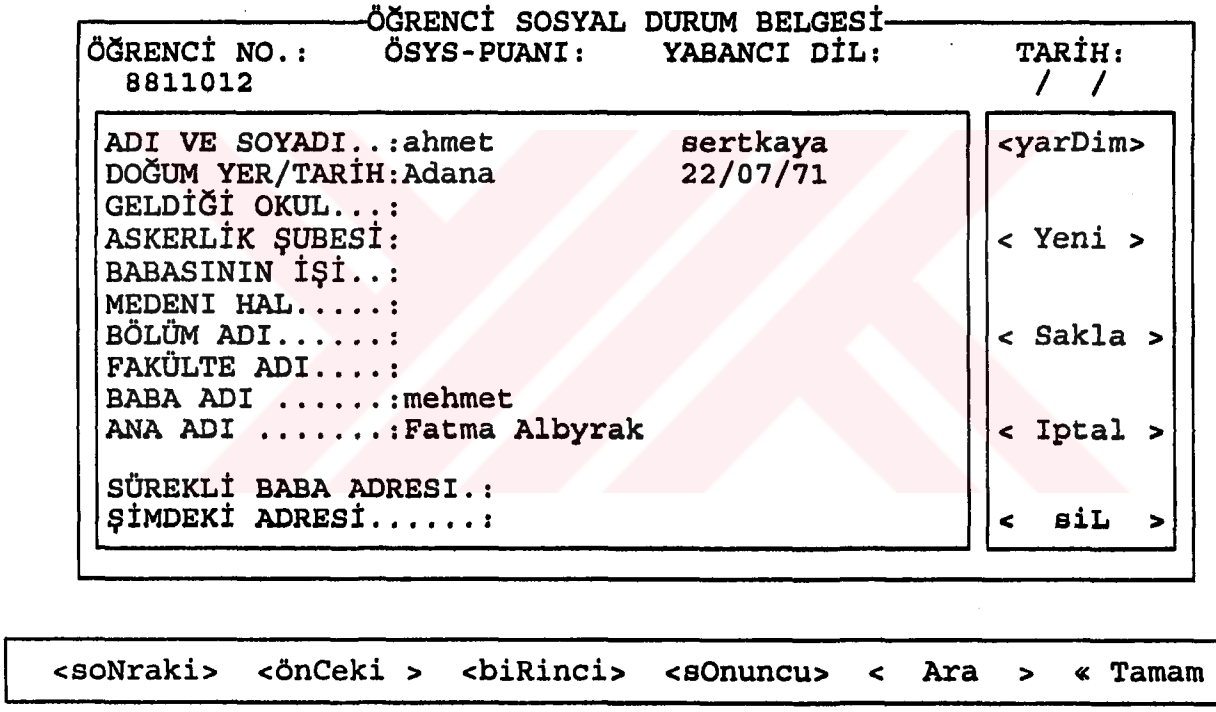
* < Yeni > ... Aktif ekranda yeni bir kayıda ait yeni bilgilerin girilmesine olanak sağlar. < Yeni > seçeneği ilk seçildiğinde bilgi alanlarına boşluklar yerleştirilir ve kursör ilk alana gelir. ilk alanin doldurulmasından sonra bir uyarı penceresi çıkar:

" LÜTFEN DEĞİŞİKLİKLERİ ONAY LAYINIZ " .

* < Sakla > seçeneği aktif ekrandaki yeni bilgileri veya yapılan değişiklikleri saklamaya yarar. Son saklama işleminden sonra herhangi bir değişiklik yapılmaışsa <Sakla> seçeneği kullanım dışıdır.

Bunlara ek olarak karşılaşılabilecek iki uyari penceresi daha vardır:

1_ Yeni bilgi girme işlemi bitirilerek < sakla > seçeneği seçildiğinde ise şu mesaj görülür:



<soNraki> <önCeki > <biRinci> <sOnuncu> < Ara > « Tamam »

" YENI KAYIT SAKLANDI "

2- Bazı değişiklikler yapılarak < Sakla > seçeneği seçildiğinde ise şu mesaj görülür:

" KAYITTA YAPILAN DEĞİŞİKLİKLER SAKLANDI "

* < iPtal > ... Bu seçenekle yeni girilen bilgiler veya yapılan değişiklikler kaydedilmemişse iptal edilebilir. Kayıt edildikten sonra bilgiler ancak < Sil > seçeneği kullanılarak silinebilir. Bu seçenek ancak yeni bilgi girldikten sonra veya bazı değişiklikler yapılmışsa kullanımdadır.

* < Sil > seçeneği aktif veritabanı veya veritabanlarından bir kayıdın çıkarılması için kullanılır.

Silmek işlemi sırasında hataları önlemek için silme işlemeni onaylamak için < Evet > seçildiğinde tekrar iki seçenekten oluşan bir uyarı penceresi görülür:
< Evet > , < Hayır >

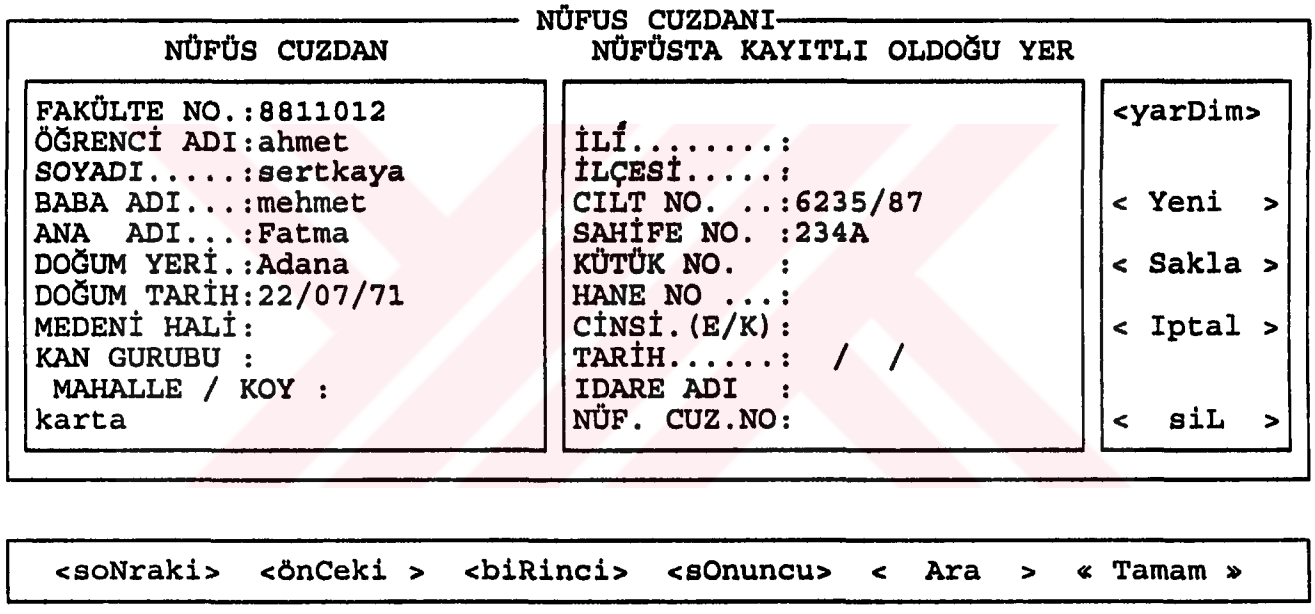
Eğer < Evet > seçeneği seçilirse seçilmiş olan kayıt aktif veritabanı dosyasından çıkarılacaktır. Ancak <Hayır > seçilirse veritabanı dosyasında bir değişiklik yer alamayacak sadece uyari peceresi kapanmış olacaktır.

NOT: Klavye yerine mouse kullanılması daha yaygındır. çünkü mouse komut seçiminin çok kolay ve hızlı yapılmasını sğlar. Oysa klavye işlemleri daha zevksiz kılar ve daha fazla zaman kaybına neden olur.

4.1.1.2 SOSYAL DURUM

İlk KAYIT menüsünden SOSYAL DURUMU seçeneği seçildiğinde saklanan eski bilgilerin bulunduğu bir giriş / çıkış ekranı görünür. Bu ekran yeni kayıt veya kayıtlar eklenmesine yada önceden kaydedilmiş bilgi ızerinde bazı değişiklikler yapılmasına olanak verir.

Bu kısım öğrencinin sosyal durum, doğum yeri ve yıl, babasının ismi, ve mesleği ve adres gibi öğrenciye ait sosyal bilgilerin saklandığı kısımdır. (Bu ekran bir önceki sayfada gösterilmiştir).



THE FULL SCREEN OF CUZDAN.

Bu ekramla kullanılan kontroller KAYIT BİLGİLERİ (4.1.1.1) için kullanılanlarla aynıdır ve aynı biçimde çalışır.

4.1.1.3.NÜFUS BİLGİLERİ

Giriş/Çıkış işlemlerinin yapıldığı ekranlardan birisi de **NÜFUS BİLGİLERİ** ekranıdır. Buradaki, **İLK KAYIT** menüsü, sosyal veri ekranında bulunan bilgilere benzer bilgilerin yanında,öğrenci için gerekli olan,baba adı, cinsiyet,kan grubu, doğum yeri gibi bilgileride içerir(Bir önceki sayfadaki şekile bakınız).

Bu bölümdekikontroller de daha önceki bölümlerdekikontrollerin aynısıdır. 4.1.1.1. bölümünde ayrıntılı olarak incelenen bu kontroller kısaca şöyle özetlenebilir :

<biRinci>, <soNraki>, <Sonuncu>, <önCeki> seçenekleri, mevcut kayıtlardan en baştaki,en sondaki,bir önceki ve bir sonraki kayıtları görmeyi mümkün kılar.

<Ara> seçeneği ile aktif veritabanı üzerinde istenen bir kayıtın bilgileri görülebilir veya printer'dan çıkış alınabilir.

<Tamam> seçeneği ise aktif window'un kapanmasını sağlar.

4.1.2 YABANCI

Yabancı öğrenciler ile ilgili işlemler **YABANCI** menüsü kullanılarak yapılabilir. Bu menükullanılarak yapılan girişler YABANCI.DBF file'ında saklanır.

YABANCI menüsündeki işlemler iki bölümde yapılmaktadır :

*** YABANCI BİLGİLER**

*** EMNİYET BELGE**

YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
DEKANLİĞİ

SAYI :B.30.2.YIL.0.49.00.04/ Tarih : / /
KONU :
İstanbul İli Emniyet Müdürlüğü
Emniyet Amirliği,
ŞEHİR
Fkültemiz Mühendisliği Bölümü sinif öğrenciler_
inden numarali / / eğitim-
öğretim yilinda almiş olduğu no.lu Kimlik kartini zayi ettiğini
belirten gazete duyurusu getirmiştir.
Adi geçen öğrencimize, Başbakanlik özlük ve Yazi İşleri Genel Mdürlüğü_
'nün 15.1.1981 gün ve 382-2300 sayili genelgesi uyarinca "kayda alindi"
belgesi verilmesini rica ederim.

Saygilarima,
DEKAN Prof.Basri Çetin

< Yeni > < Yazici > « Tamam »

YILDIZ TEKNIK ÜNİVERSİTESİ
ELEKTRİK - ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ BÖLÜMÜ

ARİH : / /
AYI :B.30.2.YIL.0.49.00.04/

oğlu/kizi 'nin / /
arihinden Üniversitemiz Elektrik-Elektronik Fakültesi Bilgisayar
ilimleri ve Mühendisliği Bölümüne numara ile kayit edildiğini
e halen bu Bölümün sinifinda BURSLU/BURSSUZ/KREDİLİ olarak kayitli
lduğunu bildirir bu belge adi geçen öğrencinin dilekçesi üzerine
üzenleşmiştir.

BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ
BÖLÜM BAŞKANI
Prof.Dr.M.YAHYA KARSLİĞİL

< Yeni > < Yazici > « Tamam »

4.1.2.1 YABANCI BİLGİLER

YABANCI BİLGİLER'de aşağıdaki iki işlemden birisi seçilebilir :

* **YABANCI ÖĞRENCİBİLGİ FORMU**
* **İKAMET TEZKERESİ BİLGİ**

1-YABANCI ÖĞRENCİ BİLGİ FORMU öğrencinin ad-soyad, baba adı, öğrenim durumu, Türkiye'deki ikamet adresi bilgilerinden oluşmaktadır.

2-İKAMET TEZKERESİ BİLGİ bölümü, pasaport gibi kimlik belgeleriiçin gerekli bilgilerin girişinde kullanılır.

4.1.2.2. EMNİYET BELGE

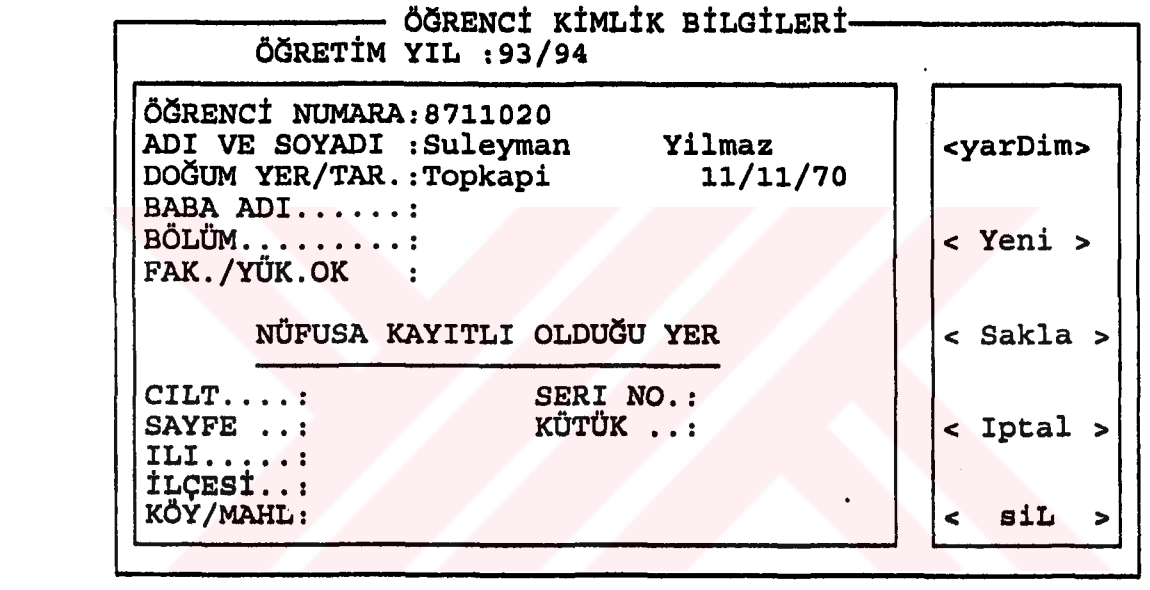
EMNİYET BELGE, vize işlemleri için göçmen bürolarından istenen formların hazırlanmasında kullanılır. **EMNİYET BELGE** seçildiğinde ilk olarak ekrana boş bir form gelir.Form gereklibilgilerle doldurulduktan sonra eğer çıkışalmak istniyorsa menü'den **<YAZI>** seçeneği seçilir. Daha sonra bilgi girişine devam etmek isteniyorsa **<YENİ>** seçeneği seçilir. Bütün işlemler bittiğinde, EMNİYET BELGE ekranından çıkış için, **<Tamam>** seçeneği seçilir.

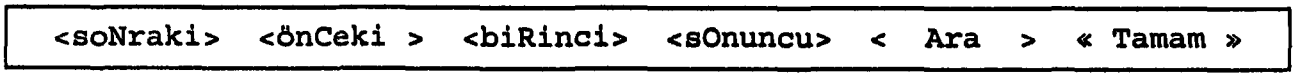
4.1.3 ÖĞRENCİ BELGE

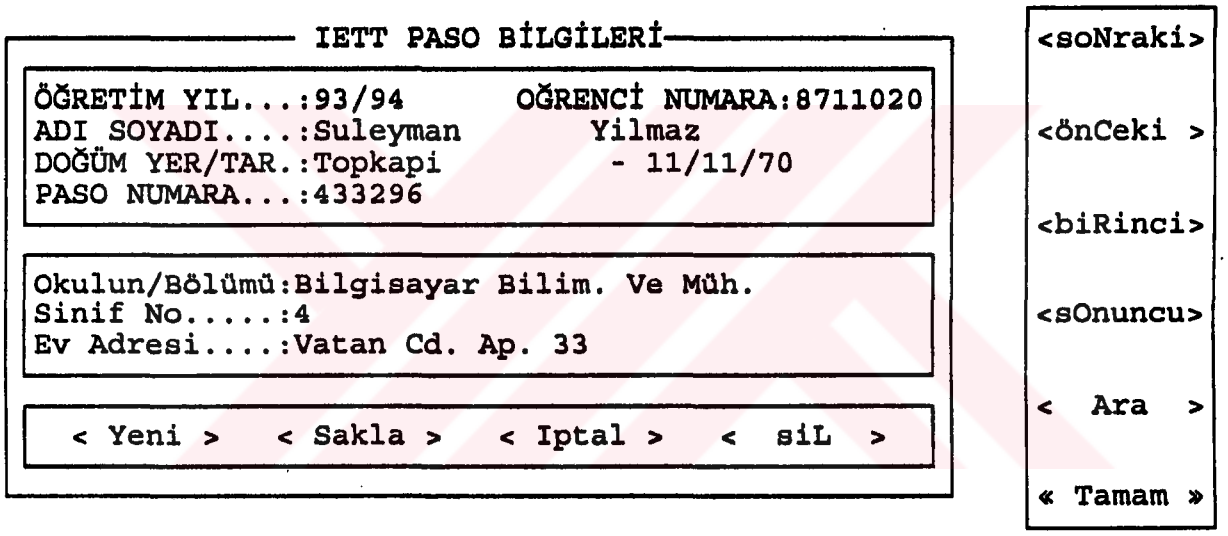
ÖĞRENCİ BELGE ekranı kullanılarak, öğrenci belgesi için gerekli bilgiler girilir ve printer çıktısı alınabilir.

4.1.4 KARTLAR

KARTLAR seçeneği, kimlik, paso gibi kimlik kartları bilgilerinin hazırlanması ve gerektiğinde bu bilgiler üzerinde değiştirme,silme gibi işlemler yapılması istendiğinde kullanılır. Bu bilgiler **KIMLIK.DBF** file'ında saklanır.


ÖĞRENCİ KİMLİK BİLGİLERİ
ÖĞRETİM YIL :93/94
ÖĞRENCİ NUMARA:8711020
ADI VE SOYADI :Suleyman Yilmaz
DOĞUM YER/TAR.:Topkapi 11/11/70
BABA ADI.....:
BÖLÜM.....:
FAK./YÜK.OK :
NÜFUSA KAYITLI OLDUĞU YER
CILT....:
SERI NO.:
SAYFE ..:
KÜTÜK ..:
ILI.....:
İLÇESİ..:
KÖY/MAHL:
<yarDim>
< Yeni >
< Sakla >
< Iptal >
< siL >


<soNraki> <önCeki > <biRinci> <sOnuncu> < Ara > « Tamam »


IETT PASO BİLGİLERİ
ÖĞRETİM YIL...:93/94 OĞRENCİ NUMARA:8711020
ADI SOYADI....:Suleyman Yilmaz
DOĞÜM YER/TAR.:Topkapi - 11/11/70
PASO NUMARA...:433296
Okulun/Bölümü:Bilgisayar Bilim. Ve Müh.
Sinif No.....:4
Ev Adresi....:Vatan Cd. Ap. 33
< Yeni > < Sakla > < Iptal > < siL >
<soNraki>
<önCeki >
<biRinci>
<sOnuncu>
< Ara >
« Tamam »

KARTLAR menüşü iki alt menüden oluşmaktadır:

* **KİMLİK**
* **PASO**

4.1.4.1.KİMLİK

Öğrenci kimlikleri KİMLİK ekranı kullanılırak hazırlanabilir. **KİMLİK** ekranı üzerinde **<Yeni>** seçeneği seçilerek bilgi girişi yapılabilir. Yeni kayıt girişi yapmak için, **(?)**'ne basılır. Bu durumda, **OGRENCLDBF** file'ından öğrencilerin numaralarını gösteren liste gelir. Bu liste üzerinden istenen öğrencinin numarası seçilerek, öğrencinin kimlik için gerekli bütün bilgilerinin **OGRENCLDBF** file'indan kopyalanması sağlanmış olur. **<Sakla>** seçeneği seçildiğinde, bu bilgiler **KIMLIK.DBF** file'ına kaydedilir.

4.1.4.2 PASO

Öğrenci pasoları **PASO** ekranı kullanılarak hazırlanabilir. **PASO** ekranı, öğrenci ad-soyad, numara, okul adı, bölüm adı, numarası, adres gibi alanlardan oluşmaktadır.

4.2. DERS MENÜ

DERS menüsü üzerinden, diploma kayıtları, ders listeleri, not listeleri, her öğrencinin aldığı derslerin listesi, gibi bilgilere erişmek, gerekiyorsa değişiklik yapmak mümkündür.

DERS menüsünün üç seçeneği vardır :

* **DEFTER**

* **KARNE**

* **DERS DİLEKÇESİ**

4.2.1 DEFTER

DEFTER seçeneği seçildiğinde, **OGRENCI.DBF** ve **DERSLER.DBF** file'ları açılır.

DEFTER iki alt menüden oluşmaktadır:

* **DEFTER GİRİŞ**

* **BİTİRME PROJESİ**

4.2.1.1. DEFTER GİRİŞ

DEFTER GİRİŞ seçildiğinde, öğrenci adı, numarası, ders adı, kredi, ders notu gibi, yeni bir kayıt girişi için gerekli olan alanlar ekrana gelir.

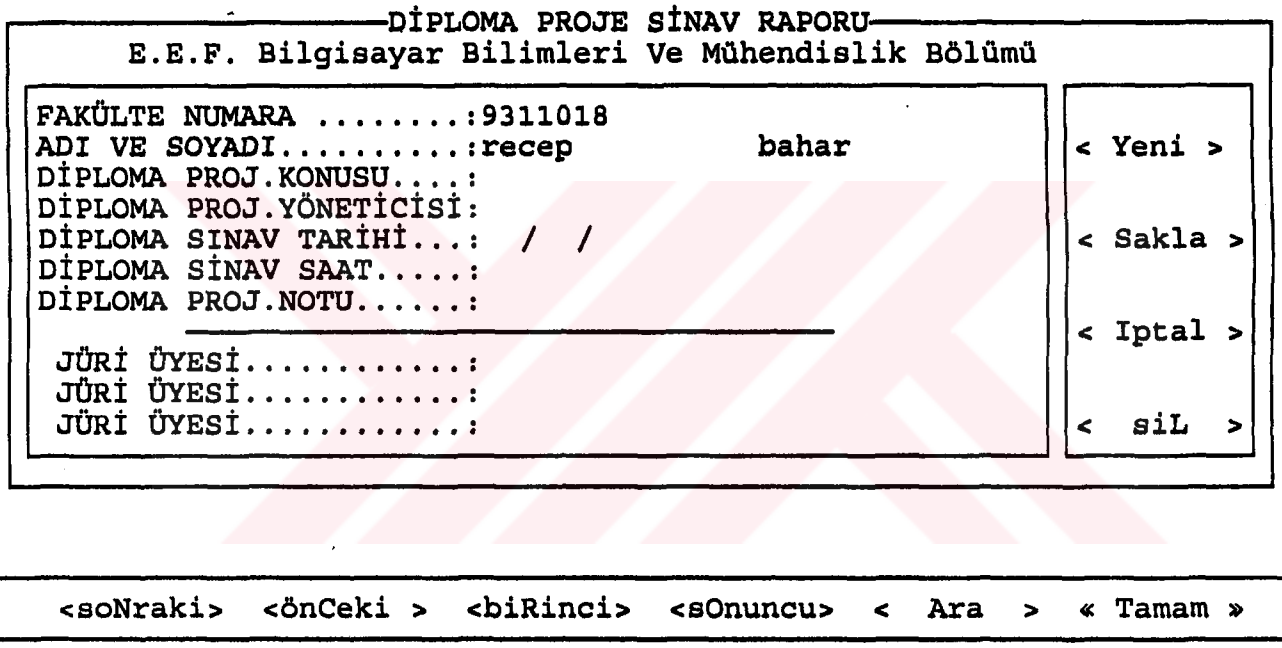
DEFTER GİRİŞ'de iki grup kontrol komutu vardır.

İlk grup kontrol komutları 6 seçenekten oluşmaktadır:
<soNraki>,<önCeki>,<biRinci>,<sOnuncu>,<Ara>,<Tamam>
Bu seçeneklerin görevleri daha önceki bölümlerde açıklanmıştır.
İkinci grup kontrol komutları ise 5 seçenekten oluşmaktadır:

1- **<yarDım>** seçeneği seçildiğinde, açık window'un nasıl doldurulması gerektiğini açıklayan bilgiler ekrana gelir.

2- Bilgi girişi için **<Yeni>** seçeneği seçilir. **<Yeni>** seçildikten sonra gelen giriş ekranında, cursor NUMARA alanı üzerine getirilerek soru işareti tuşuna(?) basıldığında öğrenci numaraları listesi ekrana gelir.Bu liste üzerinden,istenen öğrencinin numarası seçildiğinde, **OGRENCI.DBF** file'ından bu öğrencinin

 # Adı
 # Soyadı
 # Doğum Yer
 # Doğum Tarihi
 # Baba Adı


DİPLOMA PROJE SİNAV RAPORU
E.E.F. Bilgisayar Bilimleri Ve Mühendislik Bölümü
FAKÜLTE NUMARA .....:9311018
ADI VE SOYADI.....:recep bahar
DİPLOMA PROJ.KONUSU....:
DİPLOMA PROJ.YÖNETİCİSİ:
DİPLOMA SINAV TARİHİ...: / /
DİPLOMA SİNAV SAAT.....:
DİPLOMA PROJ.NOTU.....:
JÜRİ ÜYESİ.....:
JÜRİ ÜYESİ.....:
JÜRİ ÜYESİ.....:
< Yeni >
< Sakla >
< Iptal >
< siL >
<soNraki> <önCeki > <biRinci> <sOnuncu> < Ara > « Tamam »

bilgilerinin kopyalanması sağlanmış olur. Dersi kopyalamak için,yine (?) tuşuna basılabilir. Bu tuşa basıldığında, derslerin isimlerini gosteren DERSLIST listesi ekrana gelir. Bu liste üzerinden istenen ders seçilerek bu dersin adının aktif ekrandaki ders alanına kopyalanması sağlanmış olur. Daha sonra gerekli olan diğer bilgilerin de girişi yapılır.

Bu veritabanı file'ında ders kodu alanı, dersin okutulduğu sınıf ve sömestr bilgilerini gösteren iki bilgiden oluşmaktadır. Örneğin ikinci sınıf, birinci dönemde okutulan **FİZİK** dersi **"21 FİZİK"** olarak kodlanır.

4.2.1.2 BİTİRME PROJESİ

Bitirme Projesi sınav raporu işlemleri için, **BİTİRME PROJESİ** seçeneği seçilmelidir. Burada yapılan yeni kayıt girişleri ve mevcut kayıtlarda yapılan değişiklikler **DERSLER.DBF** file'ında saklanır.

4.2.2 KARNE

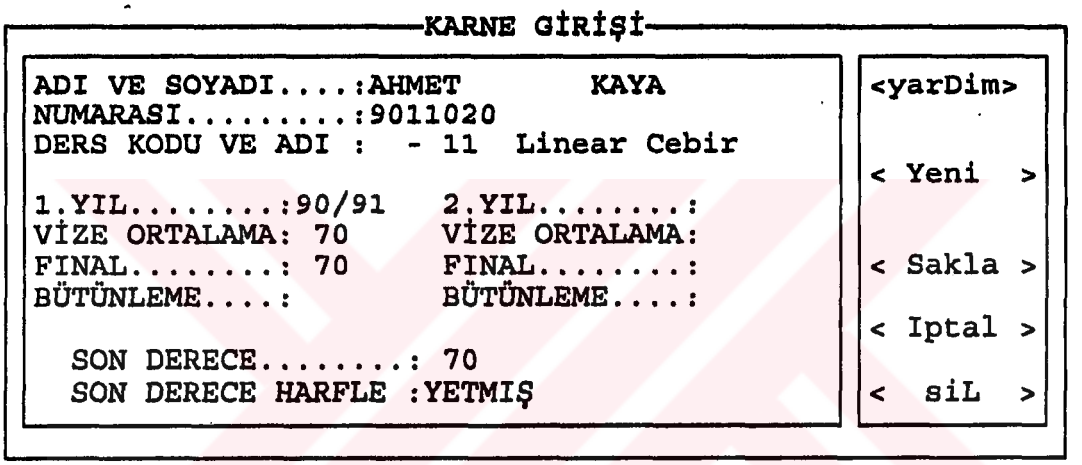
DERS menüsünden KARNE seçeneği seçildiğinde öğrencinin karne bilgilerinin yanısıra eğitim programı için de yeni bilgi girişi yapılabilir. Mevcut bilgiler üzerinde değişiklik yapılabilmesi de olanaklıdır. Ayrıca, herhangi bir öğrencinin karnesi ekranda görülebilir veya yazıcı çıkışı olarak alınabilir.

4.2.2.1 DERS TANIM

Giriş/Çıkış ekranının bu bölümünde yeni bir kayıt için veri alanlarına bilgi girişi yapılabilir veya eski bilgiler değiştirilebilir. DERS TANIM ekranı bir ders adının kredi sayısı ile birlikte girilebilmesine olanak tanır. Aynı zamanda sınav sonuçlarının da girilebilmesini sağlar (Vize, Final ve Bütünleme olarak).

Ders Adı Girişi Nasıl Yapılır?

Önce, yeni bir kayıt girebilmek için <YENİ> tuşuna basılarak boş veri alanları açılır. Bundan sonra, imleç Ders Adı alanında iken ? tuşuna basılarak dört yıldaki derslerin isimlerini içeren liste



<soNraki> <önCeki > <biRinci> <sOnuncu> < Ara > « Tamam »

görüntülenir. Bu durumda iken belirlenen ders adı mouse tuşu ile işaretlenir. Böylece bu isim aktif pencerenin ilgili veri alanına kopyalanmış olur.
(Bir önceki sayfada verilen tam ekran şekline bakınız.)

Bir dersin iki kereden fazla tekrarlanabilme olasılığı olduğundan ekran, sınav sonuçlarının girilmesi için iki farklı veri alanı gurubu içerir. Bu gruplar şunlardır:

(1) 1.YIL	(2) 2.YIL
VİZE	VİZE
FİNAL	FİNAL
BÜTÜNLEME	BÜTÜNLEME

Bir öğrenci bir derse ilk kez kayıt olduğunda yalnızca ilk gurup veri alanı doldurulur. Eğer öğrenci dersten geçerse (final veya bütünleme sınavında) imleç ikinci gurup alana gitmek yerine final sonucu alanına geçer. Bununla beraber eğer öğrenci dersten başarısız olursa imleç final sonucu alanına gitmek yerine ikinci gurup alanlara geçer ve veri alanları normal şekilde doldurulabilir.
Ayrıca, eğer bir öğrenci dersi ikinci kez de geçemez ise bu durumda verilerdeki değişiklikler yalnızca ikinci gurupta yapılabilir.

4.2.2.2 KARNE BİLGİLERİ

Bu bölümde, KARNE BİLGİLERİ seçeneği herhangi bir öğrenciye ait kişisel bilgilerin ve öğrenim süresi ve tarihi gibi öğrencinin öğrenim programı ile ilgili bilgilerin girilmesini sağlar.

Bir öğrencinin kişisel bilgileri (OGRENCI.DBF) dosyasından KARNE BİLGİLERİ alanlarına okunur ve (SONOT1.DBF) dosyasına kaydedilir.
İmleç öğrenci numarası alanına getirilerek ? tuşuna basılır ve ana ekranın ortasında öğrenci numaralarını içeren bir liste görüntülenir. İstenen numara seçilerek KARNE BİLGİLERİ veri alanına kopyalanır ve aşağıdaki alanlar doluncaya dek ENTER tuşuna basılır.

 # ADI
 # SOYADI
 # DOĞUM YERİ
 # DOĞUM TARİHİ
 # YABANCI DİLİ
 # GİRİŞ PUANI

Diğer veriler klavyeden girilir.

4.2.2.3 KARNE LİSTESİ

KARNE LİSTESİ üç bölümden oluşan bir ekrandır.
Bunlardan ilki yeni fakülte numarası girişini sağlayan boş "FAKÜLTE NUMARA" alanıdır. İkincisi üç seçenekten oluşan ve kontrol indeksi seçimini sağlayan bir listedir. Bu seçenekler şunlardır:

1- kayıt numarasına göre indeks
2- ders koduna göre indeks
3- ders adına göre indeks

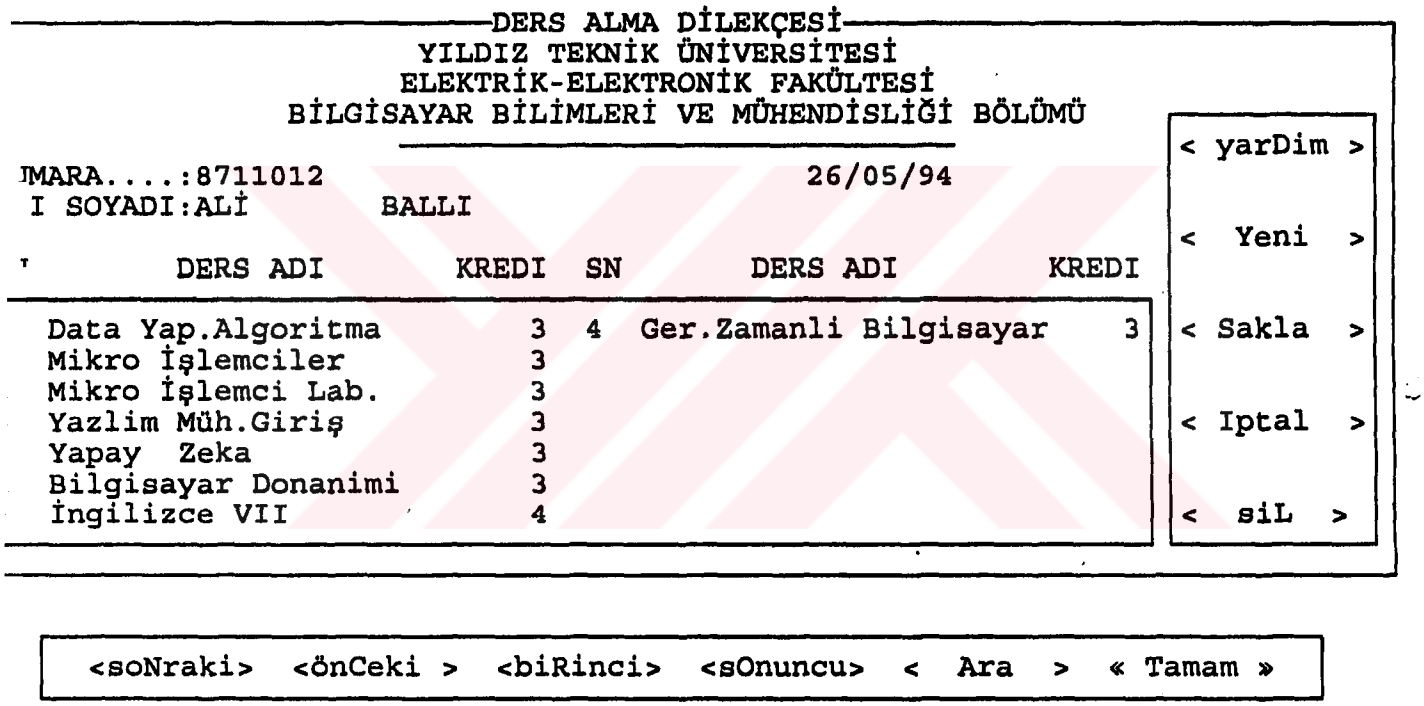
Son kısım ise denetim komutu seçenekleridir.

Bu bölüm belirlenen bir öğrencinin, numarasının girilmesi ile karnesinin ekranda görüntülenmesini sağlar.

<YARDIM> basma butonu mouse ile işaretlendiğinde bir komut penceresi açılır. Bu ekranın nasıl kullanılacağı bu pencere içinden görülebilir.

<YENİ> basma butonu fakülte numarası girişi için boş bir alan açar ve yeni bir fakülte numarasının girilebilmesini sağlar.

<TAMAM> basma butonu ise geçerli ekranın kapatılmasını sağlar.



DERS ALMA DİLEKÇESİ

YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ BÖLÜMÜ

İMARA....:8711012 26/05/94
I SOYADI:ALİ BALLI

DERS ADI	KREDI	SN	DERS ADI	KREDI
Data Yap.Algoritma	3	4	Ger.Zamanli Bilgisayar	3
Mikro İşlemciler	3			
Mikro İşlemci Lab.	3			
Yazlim Müh.Giriş	3			
Yapay Zeka	3			
Bilgisayar Donanimi	3			
İngilizce VII	4			

< yarDim >
< Yeni >
< Sakla >
< Iptal >
< siL >

<soNraki> <önCeki > <biRinci> <sOnuncu> < Ara > « Tamam »

4.2.3 DERS DİLEKÇESİ

DERS DİLEKÇESİ ilgili yarıyıla ait derslerin düzenlenmesini sağlayan ekranı görüntüler.

Yeni ders isimleri girişi için DERS DİLEKÇESİ ekranının kullanımı <YENİ> basma butonunun işaretlenmesi ile gerçekleştirilebilir. Bu durumda boş veri alanları açılır ve imleç ekranın ilk veri alanı olan (Tarih) alanına gider. ENTER tuşuna basıldığında tarih bilgisi sistemden okunarak bu alana yazılır.

İmleç FAKÜLTE NUMARASI alanında iken ? tuşuna basıldığında DERSLER.DBF dosyasından okunan ve öğrenci numaralarını içeren bir liste görüntülenir. İstenen numara belirlenerek mouse düğmesi ile işaretlenir.

Bu durumda belirlenen numara "FAKÜLTE NUMARASI" alanına kopyalanır ve ENTER tuşuna iki kez basılarak seçilen öğrencinin adı ve soyadı görüntülenir.

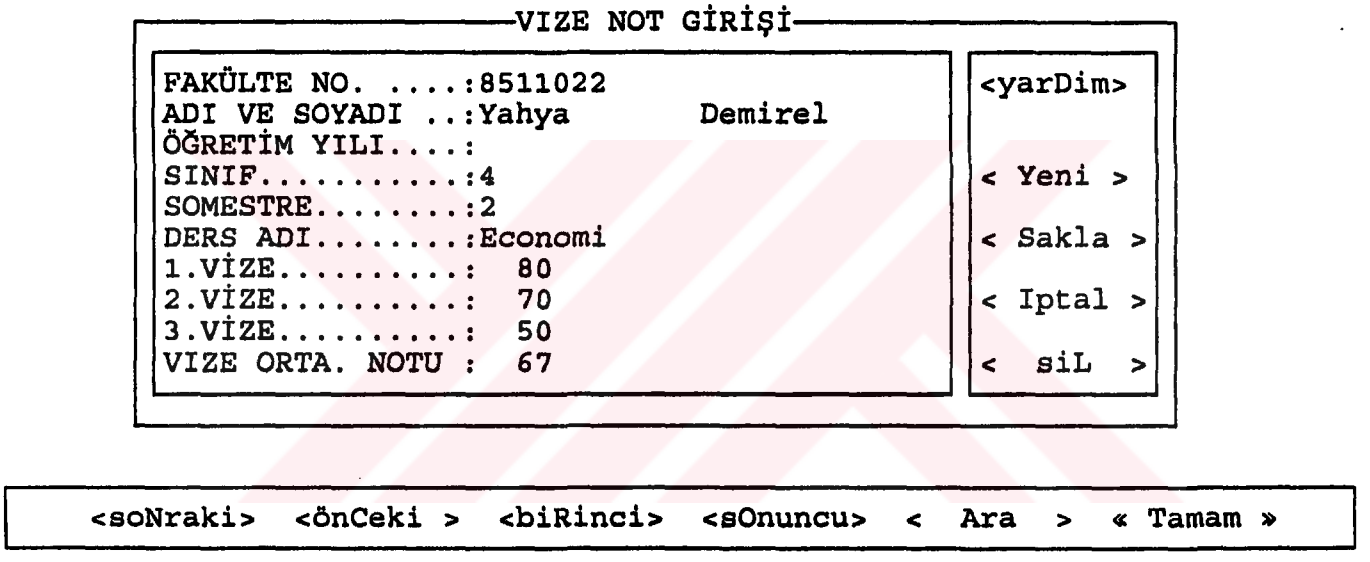
Eğer ders adı alanları doldurulmak istenirse imleç soldaki kutunun ilk alanına getirilir. Soru işareti (?) tuşuna basılarak dört yıla ait ders isimleri ders kodları ile birlikte bir liste halinde görüntülenir. İstenen ders adı mouse ile işaretlenir. Bu durumda ders adı seçilen ilk ders adı alanına kopyalanır. Bu işlem, gerekli tüm ders adlarının girişi tamamlanıncaya kadar tekrar edilir.

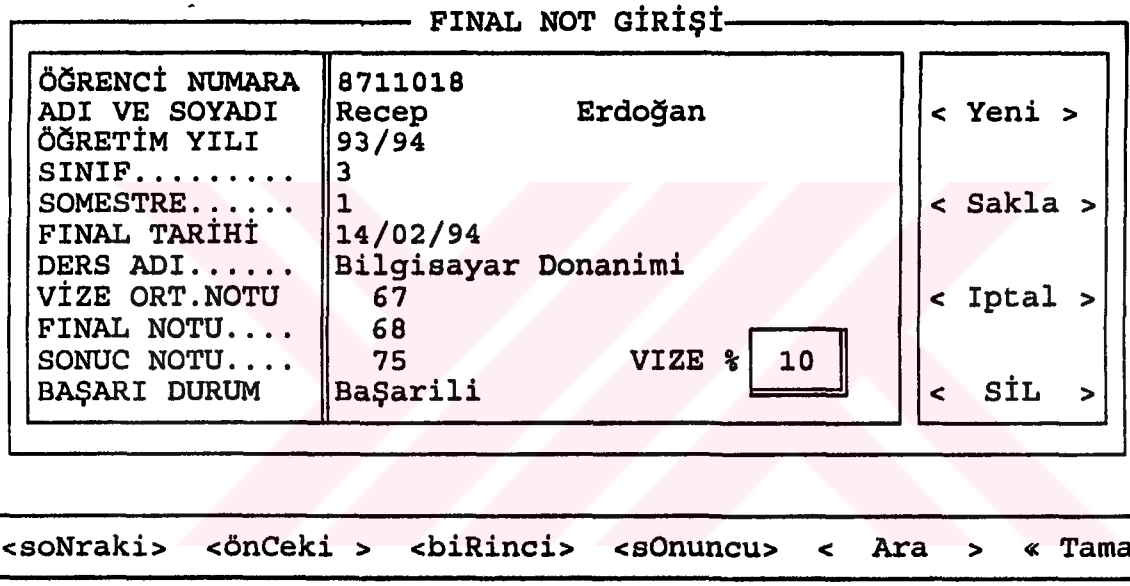
Kod iki basamaklı bir kod olduğundan ilk basamak sınıf numarasına karşılık düşer ve böylelikle herhangi bir sınıf için ilk ders adı seçimi, numarasına basmak suretiyle kolaylıkla yapılabilir.

NOT:
Ders adları girişi için 14 alan vardır. Bununla beraber, derslerin sayısını kredi toplamı sınırlar. Örneğin Bilgisayar bölümünde bir yarıyıl için kredi toplamı 33 dür. Bu hesaplama program tarafından değil kullanıcı tarafından yapılır.

4.3 NOT GİRİŞİ

Eğer sınav sonuçlarının bir veri dosyasına girişi istenirse, ana menuden NOT GİRİŞİ seçeneği seçilir. NOT GİRİŞİ menüsü şunları içerir:


VIZE NOT GİRİŞİ
FAKÜLTE NO. ....:8511022
ADI VE SOYADI ..:Yahya Demirel
ÖĞRETİM YILI....:
SINIF.....:4
SOMESTRE.....:2
DERS ADI.....:Economi
1.VİZE.....: 80
2.VİZE.....: 70
3.VİZE.....: 50
VIZE ORTA. NOTU : 67
<yarDim>
< Yeni >
< Sakla >
< Iptal >
< siL >
<soNraki> <önCeki > <biRinci> <sOnuncu> < Ara > « Tamam »


FINAL NOT GİRİŞİ
ÖĞRENCİ NUMARA
ADI VE SOYADI
ÖĞRETİM YILI
SINIF.....
SOMESTRE.....
FINAL TARİHİ
DERS ADI.....
VİZE ORT.NOTU
FINAL NOTU....
SONUC NOTU....
BAŞARI DURUM
8711018
Recep Erdoğan
93/94
3
1
14/02/94
Bilgisayar Donanimi
67
68
75
BaŞarili
VIZE % 10
< Yeni >
< Sakla >
< Iptal >
< SİL >
<soNraki> <önCeki > <biRinci> <sOnuncu> < Ara > « Tamam »

* **VİZE** (Yıliçi sınavı)
* **FİNAL**
* **BÜTÜNLEME**

4.3.1 VİZE

VİZE seçeneği yıliçi sınav sonuçları giriş ekranını görüntüler. Böylelikle, öğrenci numarası, ders adı, herbir ders için üç yıliçi sınav sonucu ile bunların ortalaması bilgilerinin girişi yapılabilir.

Bu kısım (DERSLER.DBF) dosyası ile bütünleştirilmiştir. Yeni bir kayıt girilmek istendiğinde, <YENİ> basma butonu işaretlenerek imleç, öğrenci numarası alanına götürülür. Soru işareti (?) tuşuna basılarak numara listesi alınır, istenen numara belirlenerek mouse ile işaretlenir ve imlecin bulunduğu alana kopyalanır. Diğer veri alanlarının doldurulması için ENTER tuşuna basmaya devam edilir:

* **ADI**
* **SOYADI**
* **ÖĞRETİM YILI**

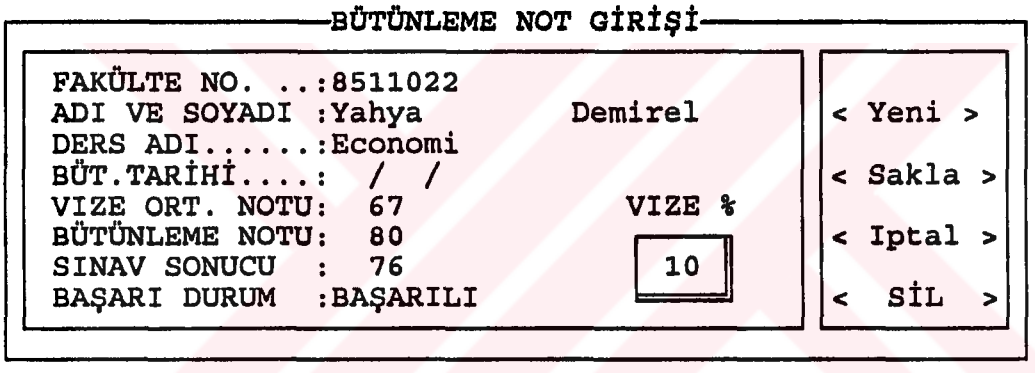
Yukardaki alanların doldurulmasından sonra kalan alanlar klavyeden girilerek doldurulur.

Kontroller: VİZE ekranındaki <yarDim>,<Yeni>,<Sakla>,<Iptal>,<siL> kontrollerinin tümü daha önceki bölümlerde geçenlerle aynı şekilde çalışır.

4.3.2 FİNAL

FINAL seçeneği seçildiğinde, final sınavı sonuçları için bir veri giriş ekranı görüntülenir. Bu bölümde istenen herhangi bir ders için sınav sonuçlarının girişi yapılabilir.

FINAL seçeneği mouse ile işaretlenerek veri giriş ekranı elde edilir. Bu ekran,yıliçi notlarının ortalaması gibi VIZE veri ekranından daha önce kayıt edilmiş bazı bilgileri de beraberinde getirir. FINAL veri ekranı sınav tarihlerini, ders isimlerini ve final sonuçlarını içerir.



<soNraki> <önCeki > <biRinci> <sOnuncu> < Ara > « Tamam »

Yıliçi notları ortalamasının bir değişken olarak göz önüne alınmasıyla, ilgili derslere ve bu dersleri veren kişilerin isteklerine uygun bir menü listesi değişen ihtiyaçları karşılayacak biçimde verilmiştir.

Veri alanlarını doldurmaya başlamadan önce yıliçi ortalama yüzdesi (% VIZE) seçilerek final sonucu, seçilen bu yüzdeye göre hesaplanabilir. Belirlenen yüzde, değiştirilinceye veya ekran kapatılıncaya kadar aktif olarak kalır.

Buna ek olarak, imleç "BAŞARI DURUMU" başlıklı alana götürülüp ENTER tuşuna basıldığında final sınavı sonucu, geçme durumunda "BAŞARILI", kalma durumunda ise "BAŞARISIZ" ifadesiyle yazılı olarak verilir. Kontroller bu bölümün başında açıklanmaktadır.

4.3.3 BÜTÜNLEME

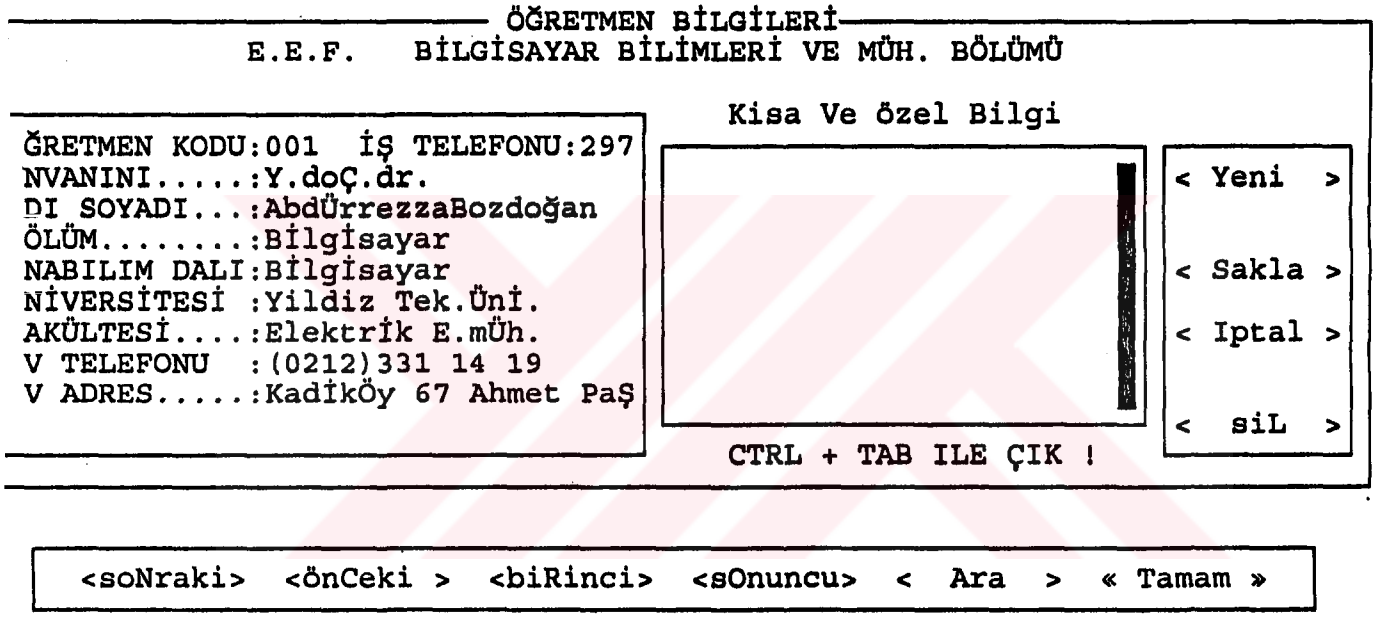
BUTUNLEME ekranı FINAL ekranı ile aynıdır. İkisi arasındaki tek fark FINAL ekranı final sınavı sonuçlarının girilmesini sağlarken BUTUNLEME ekranının bütünleme sınavı sonuçlarının girilmesini sağlamasıdır.

4.3.4 SONUCA

NOT GİRİŞ menüsünden SONUCA seçeneği seçildiğinde, ders adı ve indeks listesi alanlarından oluşan bir menü görüntülenir. Önce öğrenci numarası, adı ve soyadı bilgilerini içeren indeks listesinden istenen indeks seçimi yapılır. İndeks seçiminden sonra ders adı girişi yapılarak ENTER tuşuna basılır ve bu dersin sınavına giren öğrencilerin numara ve isimleri ile aldıkları notları içeren liste görüntülenir.

4.4 BİLGİLER

Bir bölümdeki herhangi bir öğretmen hakkında bilgi almak, bir öğrencinin askerlik durumunu öğrenmek veya bir bölüm hakkında bilgi edinmek için BILGILER menüsü seçilir. Ayrıca, çalışma esnasında zamandan haberdar olmak istenirse BILGILER menüsündeki SAAT seçeneği işaretlenir ve zaman bilgisi ekranın sağ üst köşesinde görüntülenir.


ÖĞRETMEN BİLGİLERİ
E.E.F. BİLGİSAYAR BİLİMLERİ VE MÜH. BÖLÜMÜ
ĞRETMEN KODU:001 İŞ TELEFONU:297
NVANINI.....:Y.doÇ.dr.
DI SOYADI...:AbdÜrrezzaBozdoğan
ÖLÜM.....:Bİlgİsayar
NABILIM DALI:Bİlgİsayar
NİVERSİTESİ :Yildiz Tek.Üni.
AKÜLTESİ....:Elektrİk E.mÜh.
V TELEFONU :(0212)331 14 19
V ADRES.....:KadİköY 67 Ahmet PaŞ
Kisa Ve özel Bilgi
< Yeni >
< Sakla >
< Iptal >
< siL >
CTRL + TAB ILE ÇIK !

<soNraki> <önCeki > <biRinci> <sOnuncu> < Ara > « Tamam »

4.4.1 OGRETMEN

OGRETMEN menüsü herhangi bir öğretmen ile ilgili yeni kayıt girişine izin verir. Aynı zamanda öğretmenler hakkında kısa ve özlü bilgi içeren bir liste de görüntülenir. OGRETMEN alt menüleri ise şunlardır:

* **BİLGİ GİRİŞ**
* **LİSTE**

4.4.1.1 BİLGİ GİRİŞ

BILGI GIRIS seçeneği bir veri alanı ekranı görüntüler ve öğretmenlerle ilgili bilgi girilmesini sağlar. Bu kısımda veri girişi soru işareti (?) sistemi kullanılmaksızın klavye üzerinden yapılır.

Buna ek olarak bu bölüm bir not alanı içerir. Bu özellik, bir öğretmen hakkında kısa bir kişisel bilgi kaydedilebilmesini sağlar. İstenen not yazıldıktan sonra CTRL ve TAB tuşları ile bu alandan çıkılabilir.

Burada da kontroller bu bölümün başında olduğu gibi aynı şekilde ve aynı amaçla kullanılır.

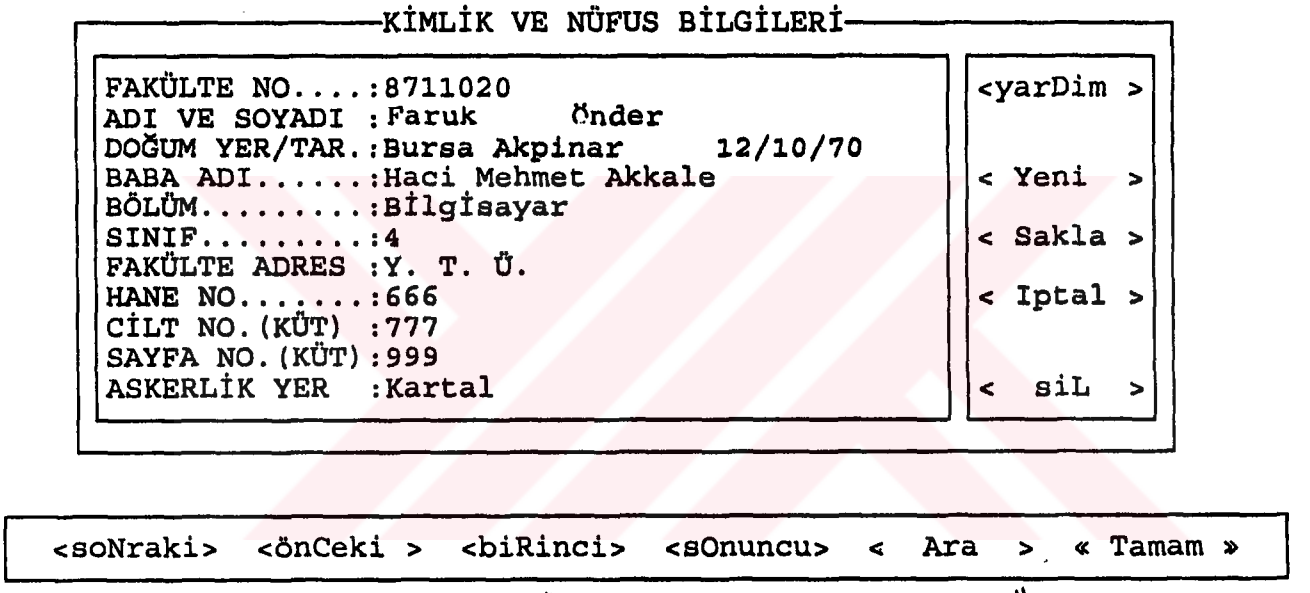
4.4.1.2 LİSTE

LİSTE seçeneği bilgi tarama olanağı sağlar. Bu sayede bir öğretmenin kodu veya adı girilerek GOSTER butonu işaretlendiğinde bir veri listesi görüntülenir. Bu liste penceresini kapatmak için pencerenin sol üst köşesine mouse ile tıklanır veya ESCAPE tuşuna basılır.

4.4.2 ASKER

ASKER menüsü seçildiğinde öğrencinin eğitim durumunun yanısıra askerlik durumu hakkında da bilgi edinilebilir.

ASKER iki seçenek içerir:



The full screen of "Kimlik ve Nüfus Bilgileri"

* **KİMLİK BİLGİ**
* **OKUL BİLGİ**

ASKER seçeneği ile kaydedilen bilgi (ASKERLIK.DBF) isimli dosyada saklanır.

4.4.2.1 KİMLİK BİLGİ

KİMLİK BİLGİ seçimi tamamlandığında veri girişini sağlayan bir ekran görüntülenir. Bu kısım iki veri tabanı arasındaki ilişkiden oluşur.

* **OGRENCI.DBF**
* **ASKERLIK.DBF**

Bu nedenle, yeni bir kayıt girilmek istendiğinde <Yeni> düğmesi işaretlenir. Bu durumda imleç ekranın ilk veri alanına gider. Soru işareti (?) tuşuna basılarak OGRENCI.DBF dosyasından okunan öğrenci numaraları listesi görüntülenir ve istenen numara işaretlenir. Bu durumda seçilen numara imlecin bulunduğu alana (ilk alana) kopyalanır.

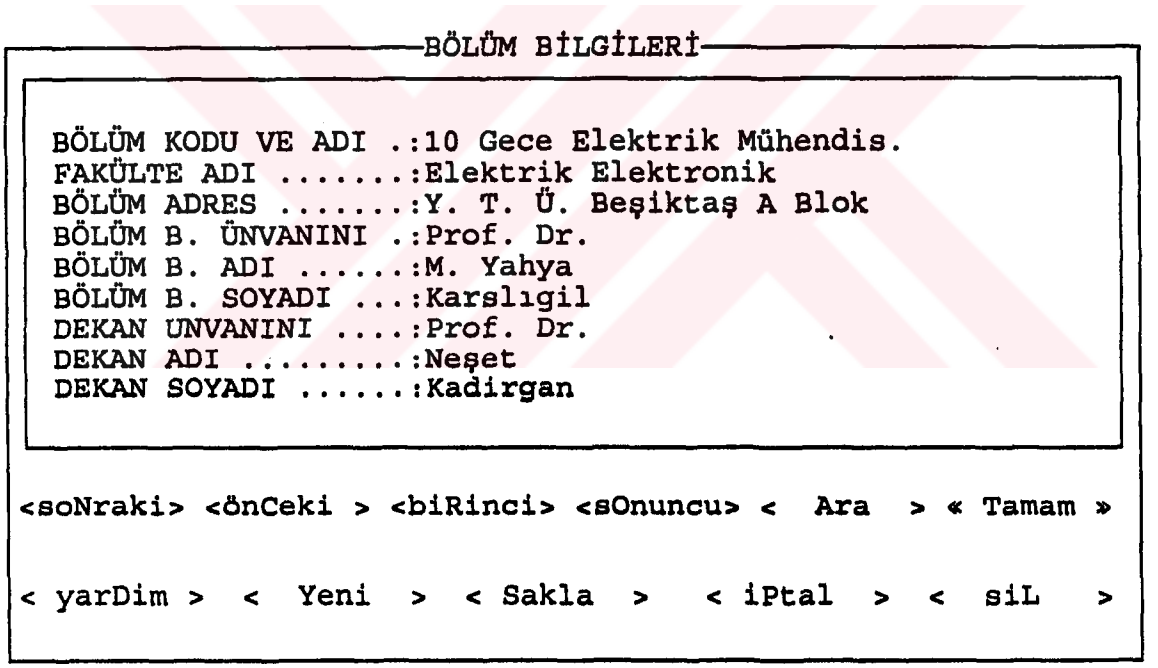
ENTER tuşu kullanılarak kalan alanlar da doldurulur. Son alandan sonra ENTER tuşuna basıldığında imleç <SAKLA> düğmesine gider ve tekrar <ENTER> tuşuna basılarak veri ASKERLIK.DBF dosyasına kaydedilir. (Bir önceki sayfadaki ekran şekline bakınız.)

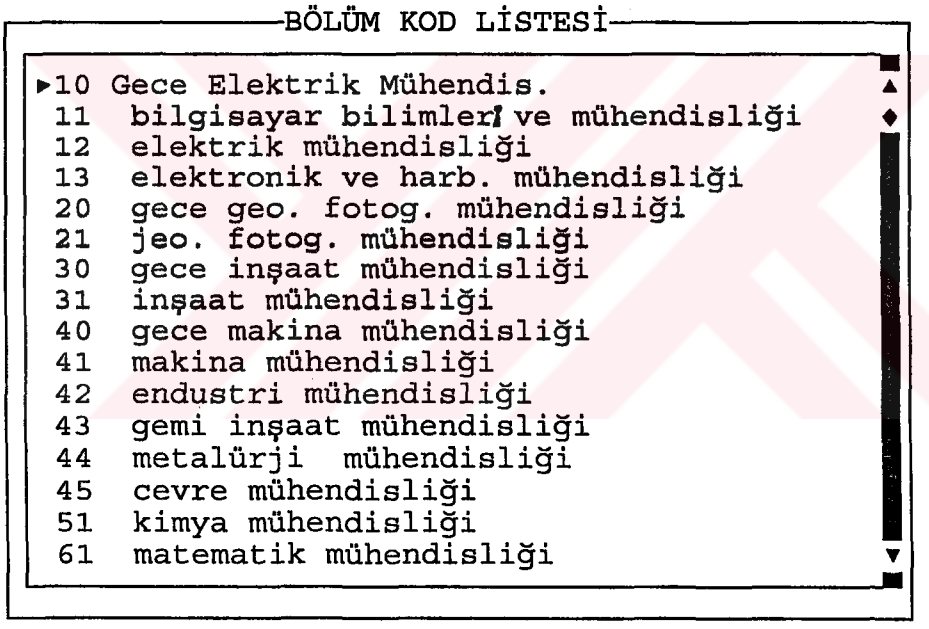
4.4.2.2 OKUL BİLGİ

OKUL BİLGİ seçeneği ile askerlik şubesinin bir öğrenci hakkında öğrenmek istediği bilgileri içeren bir ekran görüntülenir. Her ekran bir Giriş/Çıkış ekranı olarak göz önüne alınabilir ve eski veriler bu ekran yoluyla gösterilebilir.

Bu bölümde bilgi girişi yalnızca klavye üzerinden yapılabilir. Soru işareti (?) burada kullanılmaz.

Kontrol komutları, daha rahat bir çalışma şekli sağlamak amacıyla diğer ekranlarda olduğu gibi burada da kullanılabilir.


BÖLÜM BİLGİLERİ
BÖLÜM KODU VE ADI .:10 Gece Elektrik Mühendis.
FAKÜLTE ADI .....:Elektrik Elektronik
BÖLÜM ADRES .....:Y. T. Ü. Beşiktaş A Blok
BÖLÜM B. ÜNVANINI .:Prof. Dr.
BÖLÜM B. ADI .....:M. Yahya
BÖLÜM B. SOYADI ...:Karslıgil
DEKAN UNVANINI ....:Prof. Dr.
DEKAN ADI .....:Neşet
DEKAN SOYADI .....:Kadirgan
<soNraki> <önCeki > <biRinci> <sOnuncu> < Ara > « Tamam »
< yarDim > < Yeni > < Sakla > < iPtal > < siL >


BÖLÜM KOD LİSTESİ
▶10 Gece Elektrik Mühendis.
11 bilgisayar bilimleri ve mühendisliği
12 elektrik mühendisliği
13 elektronik ve harb. mühendisliği
20 gece geo. fotog. mühendisliği
21 jeo. fotog. mühendisliği
30 gece inşaat mühendisliği
31 inşaat mühendisliği
40 gece makina mühendisliği
41 makina mühendisliği
42 endustri mühendisliği
43 gemi inşaat mühendisliği
44 metalürji mühendisliği
45 cevre mühendisliği
51 kimya mühendisliği
61 matematik mühendisliği

4.4.3 BÖLÜM

BİLGİLER menüsünden seçilen BOLUM seçeneği iki alt menü görüntüler. Bu alt menüler şunlardır:

* **BÖLÜM TANIM**
* **BÖLÜM LİSTE**

BOLUM seçeneği veri tabanı üzerinde, bölümlerle ilgili aramalar ve değişiklikler yapma olanağını sağlar.

4.4.3.1 BOLUM TANIM

BOLUM TANIM seçeneği bölümler hakkında bilgi girişi yapılmasını sağlar. Bu ekran, bölümün kodu, adresi ve bölüm başkanının adı gibi bazı veri alanlarını içerir. (Bir öceki sayfadaki şekle bakınız)

4.4.3.2 BÖLÜM LİSTE

BÖLÜM LİSTE seçeneği, kayıtlı olan tüm bölümlerin kodlarıyla birlikte listelenmesini sağlar. Aynı zamanda, yeni bölümler için yeni veri girişi de yapılabilir. Eski veriler üzerinde herhangi bir zamanda değişiklikler yapılması da ayrıca olanaklıdır.

4.4.4 SAAT

Bilgisayar ortamında, en önemli konulardan biri zamandır. Bu nedenle SAAT seçeneği kullanıcının zamandan haberdar edilmesini sağlamak için projeye eklenmiştir. Zaman bilgisini elde etmek için, BİLGİLER menüsünden SAAT seçeneği seçilir ve mouse ile işaretlenir. Bu durumda sistemden okunan zaman bilgisi ana ekranın sağ üst köşesinde görüntülenir. Bu bilgi veri tabanı dosyaları kapatılıncaya dek aktif olarak kalır.

4.5 ÇIKIŞ

Ana menünün son seçeneği ÇIKIŞ dır. ÇIKIŞ seçeneğinde, ana menü kapatılmadan önce REINDEX seçeneği ile veri dosyalarının yeniden indekslenmesi önemlidir. SONU seçeneği ÇIKIŞ menüsünün ikinci seçeneği olup ana menüyü kapatarak DOS işletim sistemine dönülmesini sağlar.

SONU seçeneği seçildiğinde, seçimi onaylamak amacı ile bir mesaj penceresi görüntülenir. Bu mesaj penceresi "Program son!" uyarısı ile <Evet> ve <Hayir> düğmelerini içerir. Bunlardan <Evet> düğmesi seçildiğinde veri tabanının ana menüsü kapatılır, <Hayir> düğmesi işaretlendiğinde ise veri tabanı sistemi açık kalmaya devam eder.

SONUÇ

Bir bilgisayar sistemi yazılım ve donanım olmak üzere iki unsuru içeren bir sistemdir. Bununla birlikte, bu kitapta bilgisayar sisteminin donanım özellikleri ile ilgilenilmemiştir.

Bu tezin ana konusu olan sistemin yazılım kısmı, amacı bilgisayarın daha etkin kullanılmasını sağlamak olan bir dizi program ile ilgilidir.

Bilgisayarlar, büyük miktarlardaki bilgiyi işlemek için çok kullanışlı olan cihazlardır. Bu gerçeği, Öğrenci Otomasyon projesi sırasında yaşamış bulunmaktayım. Bir bölümün öğrencileri hakkındaki veriler bir kez bilgisayara girildikten sonra (örneğin 300 öğrenci için) kullanıcı bu bilgiyi el ile yapabileceğinden çok daha verimli bir biçimde elde edebilir veya yönetebilir.

Buna ek olarak bu projedeki çalışmam sırasında, veri tabanı sisteminin güvenilirliğini garantilemek için verilerin toplanması ve kullanıcıların görevleri konusuna çok özen gösterdim.

Veri tabanı dillerinin ne kadar hızlı geliştiğini kavramış oldum. Projeme FoxPro 2.0 MS DOS kullanımda iken başlamıştım. Programlarımın yazımında bu dili kullandım. Projenin sonuçlanmasından önce programın FoxPro 2.5 MS DOS ve 2.6 adıyla bilinen iki yeni sürümü yaygın olarak kullanıma girdi. Böyle bir ortamda, projede kullanılan veri tabanı dili geçerliliğini kaybedebilirdi. Ancak, proje veri tabanı sistemlerinin temel kavramlarını, fizibilite çalışmalarını ve bir veri tabanı yönetim sistemi tasarımının ilkelerini ortaya koyan bir çalışma olarak bu konuda bir referans olma özelliğini her zaman sürdürebilecektir.

FoxPro programlama dili tablo, etiket, rapor ve dosya yaratma gibi bir veri tabanı yönetim sisteminden beklenebilecek işlemleri daha az bellek kullanımı ile daha hızlı ve verimli bir şekilde yapması nedeniyle günümüzde veri tabanı sistemleri için kullanılan en uygun dildir.

Bununla birlikte, daha fazla bellek kullanımı sistem verimini artıracaktır.

Ağ yapısını destekleme özelliği, veri tabanı yönetim sisteminin özellikleri içinde en önemli olanlardan birisidir. Bu nedenle, FoxPro dili tarafından kuvvetle desteklenen NETWORK özelliği ve veri tabanı bilgisi birbirlerini tamamlayıcı olması açısından çok önemlidir.

Proje hakkındaki bu kısa açıklamayı sonlandırırken şunu söyleyebilirim ki: Bilgisayar Bilimlerindeki bu hızlı gelişmeye rağmen, insan doğasındaki giderek artan gelişme tutkusu nedeniyle hala geliştirilebilecek çok şey bulunmaktadır.

KAYNAKLAR

* FoxPro
Developer's Guide Includes putting It All Together
For Software

* FoxPro
Interface Guide For Software

* FoxPro
User's Gide For Software

* Language Reference
foxPro
Relational Database Management system
For MS - DOS and WINDOWS

* Step - By - Step
Quattro Pro 3.0
Peter K. & Mc Bride

* Paradox 3.0
The Complete Reference
James Keogh
Borland Osborne
Mc GRAW - HILL

* dBASE IV
Bidgoli & Hosseln
Information System Literacy Comcepts.

* Database system Comcepts
Ssecond Edition
Henry F. Korth & Abraham Silbersch
ATZ.

```
*        +-------------------------------------------------------+
*        |                                                       |
*        |  01/08/94              BELGE.SPR              14:26:01 |
*        |                                                       |
*        |-------------------------------------------------------|
*        |                                                       |
*        |  Author's Name                                        |
*        |                                                       |
*        |  Copyright (c) 1994 Company Name                      |
*        |  Address                                              |
*        |  City,     Zip                                        |
*        |                                                       |
*        |  Description:                                         |
*        |  This program was automatically generated by GENSCRN. |
*        |                                                       |
*        +-------------------------------------------------------+


#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
 SET TALK OFF
 m.talkstat = "ON"
ELSE
 m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.currarea = SELECT()


*        +-------------------------------------------------------+
*        |                                                       |
*        |                    Window definitions                 |
*        |                                                       |
*        +-------------------------------------------------------+
*

IF NOT WEXIST("ybelge")
 DEFINE WINDOW ybelge ;
  FROM INT((SROW()-21)/2),INT((SCOL()-77)/2) ;
  TO INT((SROW()-21)/2)+20,INT((SCOL()-77)/2)+76 ;
  NOFLOAT ;
  NOCLOSE ;
  SHADOW ;
  COLOR SCHEME 1
ENDIF
```

```
*       +--------------------------------------------------------+
*       ¦                                                        ¦
*       ¦              BELGE Setup Code - SECTION 2              ¦
*       ¦                                                        ¦
*       +--------------------------------------------------------+
*

#REGION 1
set echo off
set escape off
set deleted on
set date british
close databases

IF USED ("OGRENCI")
SELECT  OGRENCI
SET ORDER TO 1
ELSE
    SELECT 0
USE ( LOCFILE("\yildiz\ydbf\ogrenci.dbf","DBF","OGRENCI.DBF Nerede?"));
    AGAIN ALIAS OGRENCI ;
ORDER 1

ENDIF



*       +--------------------------------------------------------+
*       ¦                                                        ¦
*       ¦                   BELGE Screen Layout                  ¦
*       ¦                                                        ¦
*       +--------------------------------------------------------+
*

#REGION 1
IF WVISIBLE("ybelge")
 ACTIVATE WINDOW ybelge SAME
ELSE
 ACTIVATE WINDOW ybelge NOSHOW
ENDIF
@ 0,24 SAY "YILDIZ TEKNIK ÜNİVERSİTESİ"
@ 1,21 SAY "ELEKTRİK - ELEKTRONİK FAKÜLTESİ"
@ 2,16 SAY "BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ BÖLÜMÜ"
@ 3,27 TO 3,45
@ 4,2 SAY "TARİH  :"
@ 5,2 SAY "SAYI   :"
@ 5,10 SAY "B.30.2.YIL.0.49.00.04/"
@ 7,23 SAY "oğlu/kiz  "
@ 7,59 SAY "'nin"
@ 9,12 SAY "ve"
@ 9,15 SAY "Mühendisliği Bölümüne"
@ 9,45 SAY "numara ile kayit edildiğini"
@ 10,2 SAY "ve halen bu"
@ 10,15 SAY "Bölümün"
@ 10,27 SAY "sinifinda BURSLU/BURSSUZ/KREDİLİ"
@ 11,20 SAY "bu belge adi geçen öğrencinin"
```

```
@ 11,50 SAY "dilekçesi üzerine"
@ 12,2 SAY "düzenlefmiftir."
@ 13,38 SAY "BİLGİSAYAR BİLİMLERİ VE "
@ 13,62 SAY "MÜHENDİSLİĞİ"
@ 14,49 SAY "BÖLÜM BAPKANI"
@ 15,42 SAY "Prof.Dr.M.YAHYA KARSLİĞİL"
@ 16,0 TO 18,74
@ 8,2 SAY "tarihinden"
@ 9,2 SAY "Bilimleri"
@ 8,14 SAY "Üniversitemiz Elektrik-Elektronik Fakültesi Bilgisayar "
@ 10,60 SAY "olarak kayitli"
@ 11,2 SAY "olduğunu"
@ 11,11 SAY "bildirir"
@ 4,10 GET OBTAR ;
 SIZE 1,10 ;
 DEFAULT {  /  /  } ;
 PICTURE "@E" ;
 VALID _qs70uxycz()
@ 5,32 GET bsayi ;
 SIZE 1,4 ;
 DEFAULT " "
@ 7,2 GET baba ;
 SIZE 1,20 ;
 DEFAULT " "
@ 7,33 GET badi ;
 SIZE 1,25 ;
 DEFAULT " "
@ 7,64 GET bugr ;
 SIZE 1,10 ;
 DEFAULT {  /  /  } ;
 PICTURE "@E"
@ 9,37 GET bno ;
 SIZE 1,7 ;
 DEFAULT " "
@ 10,23 GET bsnf ;
 SIZE 1,3 ;
 DEFAULT " "
@ 17,32 GET m.yazici ;
 PICTURE "@*HN \<Yazici" ;
 SIZE 1,14,1 ;
 DEFAULT 1 ;

 VALID _qs70uy1vb()
@ 17,4 GET m.yeni ;
 PICTURE "@*HN \<Yeni" ;
 SIZE 1,15,1 ;
 DEFAULT 1 ;
 VALID _qs70uy4j3()
@ 17,56 GET m.tam ;
 PICTURE "@*HT \?\!\<Tamam" ;
 SIZE 1,13,1 ;
 DEFAULT 1

IF NOT WVISIBLE("ybelge")
 ACTIVATE WINDOW ybelge
ENDIF
```

```
READ CYCLE
RELEASE WINDOW ybelge
SELECT (m.currarea)
#REGION 0
IF m.talkstat = "ON"
 SET TALK ON
ENDIF
IF m.compstat = "ON"
 SET COMPATIBLE ON
ENDIF


*       +------------------------------------------------------+
*       |                                                      |
*       | _QS70UXYCZ            OBTAR VALID                    |
*       |                                                      |
*       | Function Origin:                                     |
*       |                                                      |
*       | From Screen:          BELGE,     Record Number:  31  |
*       | Variable:             OBTAR                          |
*       | Called By:            VALID Clause                   |
*       | Object Type:          Field                          |
*       | Snippet Number:       1                              |
*       |                                                      |
*       +------------------------------------------------------+
*

FUNCTION _qs70uxycz     &&  OBTAR VALID
#REGION 1
OBTAR=DATE()

*       +------------------------------------------------------+
*       |                                                      |
*       | _QS70UY1VB            m.yazici VALID                 |
*       |                                                      |
*       | Function Origin:                                     |
*       |                                                      |
*       | From Screen:          BELGE,     Record Number:  38  |
*       | Variable:             m.yazici                       |
*       | Called By:            VALID Clause                   |
*       | Object Type:          Push Button                    |
*       | Snippet Number:       2                              |
*       |                                                      |
*       +------------------------------------------------------+
*
FUNCTION _qs70uy1vb     &&  m.yazici VALID
#REGION 1
if eof()
   go bottom
endif
    REPORT FORMAT \YILDIZ\YRAPOR\YBELGER.FRX;
    NOEJECT NOCONSOLE TO PRINTER

show get m.yazici disable
```

```
*        +-------------------------------------------------------+
*        |                                                       |
*        | _QS70UY4J3            m.yeni VALID                    |
*        |                                                       |
*        | Function Origin:                                      |
*        |                                                       |
*        | From Screen:          BELGE,     Record Number:   39  |
*        | Variable:             m.yeni                          |
*        | Called By:            VALID Clause                    |
*        | Object Type:          Push Button                     |
*        | Snippet Number:       3                               |
*        |                                                       |
*        +-------------------------------------------------------+
*
FUNCTION _qs70uy4j3     &&  m.yeni VALID
#REGION 1
 STORE SPACE(10)  TO obtar
 STORE SPACE(4)   TO bsayi
 STORE SPACE(20)  TO baba
 STORE SPACE(25)  TO badi
 STORE SPACE(10)  TO bugr
 STORE SPACE(7)   TO bno
 STORE SPACE(2)   TO bsnf

SHOW GETS
_CUROBJ = OBJNUM(obtar)
show get m.yazici ENABLE
```

```
*        +-------------------------------------------------------+
*        |                                                       |
*        |  31/08/94              VIZEGIR.SPR              13:16:22  |
*        |                                                       |
*        |-------------------------------------------------------|
*        |                                                       |
*        |  Author's Name                                        |
*        |                                                       |
*        |  Copyright (c) 1994 Company Name                      |
*        |  Address                                              |
*        |  City,      Zip                                       |
*        |                                                       |
*        |  Description:                                         |
*        |  This program was automatically generated by GENSCRN. |
*        |                                                       |
*        +-------------------------------------------------------+


#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
 SET TALK OFF
 m.talkstat = "ON"
ELSE
 m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.currarea = SELECT()


*        +-------------------------------------------------------+
*        |                                                       |
*        |                     Window definitions                |
*        |                                                       |
*        +-------------------------------------------------------+
*

IF NOT WEXIST("controls")
 DEFINE WINDOW controls ;
  FROM 19, 5 ;
  TO 21,78 ;
  NOFLOAT ;
  NOCLOSE ;
  COLOR SCHEME 22
ENDIF

IF NOT WEXIST("vizegir")
 DEFINE WINDOW vizegir ;
  FROM 4, 12 ;
  TO 17,68 ;
  TITLE "VIZE NOT GÝRÝÞÝ" ;

  NOFLOAT ;
  NOCLOSE ;
  SHADOW ;
  COLOR SCHEME 2
ENDIF
```

```
*       +-------------------------------------------------------+
*       |                                                       |
*       |              VCONT Setup Code - SECTION 2             |
*       |                                                       |
*       +-------------------------------------------------------+
*

#REGION 1
REGIONAL m.cont, m.toprec, m.bottomrec, m.saverecno, m.quitting
m.quitting = .F.
m.cont = "Tamam"

IF EOF()
 GO BOTTOM
ENDIF

m.saverecno = RECNO()
GO TOP
m.toprec = RECNO()
GO BOTTOM
m.bottomrec = RECNO()
GO m.saverecno


*       +-------------------------------------------------------+
*       |                                                       |
*       |             VIZEGIR Setup Code - SECTION 2            |
*       |                                                       |
*       +-------------------------------------------------------+
#REGION 2
set echo off
set escape off
set deleted on
set exact on
set date british
close databases
IF USED("dersler")
 SELECT dersler
 SET ORDER TO 1
ELSE
 SELECT 0
 USE (LOCFILE("\yildiz\ydbf\dersler.dbf","DBF","DERSLER.DBF Nerede ?"));
  AGAIN ALIAS dersler ;
  ORDER 1
ENDIF

IF USED("notlar")
 SELECT notlar
 SET ORDER TO 1
ELSE
 SELECT 0
 USE (LOCFILE("\yildiz\ydbf\notlar.dbf","DBF","NOTLAR.DBF Nerede ?"));
  AGAIN ALIAS notlar ;
  ORDER 1
ENDIF

public m.fk_no,m.adi,m.soyadi,m.ogrt_yil

PRIVATE m.adding, m.editing,m.somest,m.ders,m.vize,m.vz1,m.vz2,m.vz3

m.adding = .F.
m.editing = .F.
```

```
m.yz=1
m.vz=0
m.fn=0
m.av=0
m.sum=0
SCATTER MEMVAR MEMO
*       +------------------------------------------------------+
*       |                                                      |
*       |                    VCONT Screen Layout               |
*       |                                                      |
*       +------------------------------------------------------+
#REGION 1
IF WVISIBLE("controls")
 ACTIVATE WINDOW controls SAME
ELSE
 ACTIVATE WINDOW controls NOSHOW
ENDIF
@ 0,4 GET m.cont ;
 PICTURE "@*HN so\<Nraki;ön\<Ceki;bi\<Rinci;s\<Onuncu;\<Ara;\?\!\<Tamam"
;
 SIZE 1,9,2 ;
 DEFAULT 1 ;
 VALID _qt10sgf2g()

*       +------------------------------------------------------+
*       |                                                      |
*       |                    VIZEGIR Screen Layout             |
*       |                                                      |
*       +------------------------------------------------------+
*

#REGION 2
IF WVISIBLE("vizegir")
 ACTIVATE WINDOW vizegir SAME
ELSE
 ACTIVATE WINDOW vizegir NOSHOW
ENDIF
@ 1,1 SAY "FAKÜLTE NO. ....:" ;

 COLOR W+/W
@ 2,1 SAY "ADI VE SOYADI ..:" ;
 COLOR W+/W
@ 4,1 SAY "SINIF...........:" ;
 COLOR W+/W
@ 5,1 SAY "SOMESTRE........:" ;
 COLOR W+/W
@ 6,1 SAY "DERS ADI........:" ;
 COLOR W+/W
@ 10,1 SAY "VIZE ORTA. NOTU :" ;
 COLOR W+/W
@ 0,44 TO 11,54 ;
 COLOR GR+/W
@ 7,1 SAY "1.VÝZE..........:" ;
 COLOR W+/W
@ 8,1 SAY "2.VÝZE..........:" ;
 COLOR W+/W
@ 9,1 SAY "3.VÝZE..........:" ;
 COLOR W+/W
@ 0,0 TO 11,43 ;
 COLOR GR+/W
@ 1,18 GET m.fk_no ;
 SIZE 1,7 ;
 DEFAULT " " ;
```

```
 WHEN _qt10sgkcl() ;
 VALID _qt10sglmv() ;
 COLOR SCHEME 5
@ 2,18 GET m.adi ;
 SIZE 1,12 ;
 DEFAULT " " ;
 VALID _qt10sgncv() ;
 COLOR SCHEME 5
@ 2,30 GET m.soyadi ;
 SIZE 1,13 ;
 DEFAULT " " ;
 VALID _qt10sgpu7() ;
 COLOR SCHEME 5
@ 3,18 GET m.ogrt_yil ;
 SIZE 1,8 ;
 DEFAULT " " ;
 VALID _qt10sgrcc() ;
 COLOR SCHEME 5
@ 4,18 GET m.sinif ;
 SIZE 1,4 ;
 DEFAULT " " ;
 VALID _qt10sgt4o() ;
 COLOR SCHEME 5
@ 5,18 GET m.somest ;
 SIZE 1,4 ;
 DEFAULT " " ;
 VALID _qt10sgugt() ;
 COLOR SCHEME 5
@ 6,18 GET m.ders ;

 SIZE 1,25 ;
 DEFAULT " " ;
 VALID _qt10sgvur() ;
 COLOR SCHEME 5
@ 7,18 GET m.vz1 ;
 SIZE 1,4 ;
 DEFAULT 0 ;
 PICTURE "@Z" ;
 VALID _qt10sgxbz() ;
 COLOR SCHEME 5
@ 8,18 GET m.vz2 ;
 SIZE 1,4 ;
 DEFAULT 0 ;
 PICTURE "@Z" ;
 VALID _qt10sgyke() ;
 COLOR SCHEME 5
@ 9,18 GET m.vz3 ;
 SIZE 1,4 ;
 DEFAULT 0 ;
 PICTURE "@Z" ;
 VALID _qt10sh0ca() ;
 COLOR SCHEME 5
@ 10,18 GET m.vize ;
 SIZE 1,4 ;
 DEFAULT 0 ;
 PICTURE "@Z" ;
 VALID _qt10sh23o() ;
 COLOR SCHEME 5
@ 4,45 GET m.newrecord ;
 PICTURE "@*VN \<Yeni" ;
 SIZE 1,8,4 ;
 DEFAULT 1 ;
 VALID _qt10sh3rc() ;
 COLOR SCHEME 1
```

```
@ 6,45 GET m.saverecord ;
 PICTURE "@*VN \<Sakla" ;
 SIZE 1,9,1 ;
 DEFAULT 1 ;
 WHEN _qt10sh5fj() ;
 VALID _qt10sh6n8() ;
 DISABLE ;
 COLOR SCHEME 1
@ 8,45 GET m.cancel ;
 PICTURE "@*VN \<Iptal" ;
 SIZE 1,9,1 ;
 DEFAULT 1 ;
 VALID _qt10sh7jn() ;
 DISABLE ;
 COLOR SCHEME 1
@ 1,45 GET m.yardim ;
 PICTURE "@*HN yar\<Dim" ;
 SIZE 1,8,1 ;
 DEFAULT 1 ;

 VALID _qt10sh8ah() ;
 COLOR SCHEME 1
@ 10,45 GET m.sil ;
 PICTURE "@*HN si\<L" ;
 SIZE 1,9,1 ;
 DEFAULT 1 ;
 VALID _qt10sh8yl() ;
 COLOR SCHEME 1
@ 3,1 SAY "Ö*RETÝM YILI....:" ;
 COLOR W+/W

IF NOT WVISIBLE("vizegir")
 ACTIVATE WINDOW vizegir
ENDIF
IF NOT WVISIBLE("controls")
 ACTIVATE WINDOW controls
ENDIF


READ CYCLE ;
 WHEN _qt10shagg() ;
 DEACTIVATE _qt10shagk() ;
 SHOW _qt10sghd8()

RELEASE WINDOW controls
RELEASE WINDOW vizegir
SELECT (m.currarea)


#REGION 0
IF m.talkstat = "ON"
 SET TALK ON
ENDIF
IF m.compstat = "ON"
 SET COMPATIBLE ON
ENDIF
```

```
*        +-------------------------------------------------------+
*        |                                                       |
*        |                  VIZEGIR Cleanup Code                 |
*        |                                                       |
*        +-------------------------------------------------------+

#REGION 2
IF m.quitting
 RELEASE WINDOW VIZEGIR
 RELEASE WINDOW controls
ENDIF

*
* SHOWSAVE
*
PROCEDURE showsave
IF NOT m.editing
 WAIT WINDOW "LÜTFEN DE‘ÝPÝKLÝKLERÝ ONAYLAYINIZ" NOWAIT
ENDIF

SHOW GET m.newrecord DISABLE
SHOW GET m.saverecord ENABLE
SHOW GET m.cancel ENABLE
SHOW GETS ONLY DISABLE WINDOW controls
m.editing = .T.

*        +-------------------------------------------------------+
*        |                                                       |
*        |  _QT10SGF2G            m.cont VALID                   |
*        |                                                       |
*        |  Function Origin:                                     |
*        |                                                       |
*        |  From Screen:          VCONT,     Record Number:   2  |
*        |  Variable:             m.cont                         |
*        |  Called By:            VALID Clause                   |
*        |  Object Type:          Push Button                    |
*        |  Snippet Number:       1                              |
*        |                                                       |
*        +-------------------------------------------------------+
*
FUNCTION _qt10sgf2g     &&  m.cont VALID
#REGION 1
DO CASE
CASE m.cont = "soNraki"
 SKIP 1
 IF RECNO() = m.bottomrec
  SHOW GET m.cont, 1 DISABLE
  SHOW GET m.cont, 4 DISABLE
 ELSE
  IF RECNO() > m.toprec
   SHOW GET m.cont, 2 ENABLE
   SHOW GET m.cont, 3 ENABLE
  ENDIF
 ENDIF
CASE m.cont = "önCeki"
 SKIP -1
 IF RECNO() = m.toprec
  SHOW GET m.cont, 2 DISABLE
  SHOW GET m.cont, 3 DISABLE
 ELSE
  IF RECNO() < m.bottomrec
   SHOW GET m.cont, 1 ENABLE
   SHOW GET m.cont, 4 ENABLE
  ENDIF
```

```
 ENDIF
CASE m.cont = "biRinci"
 GO TOP
 SHOW GET m.cont, 1 ENABLE
 SHOW GET m.cont, 2 DISABLE
 SHOW GET m.cont, 3 DISABLE
 SHOW GET m.cont, 4 ENABLE
CASE m.cont = "sOnuncu"
 GO BOTTOM
 SHOW GET m.cont, 1 DISABLE
 SHOW GET m.cont, 2 ENABLE
 SHOW GET m.cont, 3 ENABLE
 SHOW GET m.cont, 4 DISABLE
CASE m.cont = "Ara"
     do vbrowser.spr
CASE m.cont = "Tamam"
 m.idlequit = .T.
 m.quitting = .T.
 CLEAR READ
ENDCASE
SHOW GETS

*       +-------------------------------------------------------+
*       |                                                       |
*       | _QT10SGKCL            m.fk_no WHEN                    |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:          VIZEGIR,     Record Number:  13 |
*       | Variable:             m.fk_no                         |
*       | Called By:            WHEN Clause                     |
*       | Object Type:          Field                           |
*       | Snippet Number:       2                               |
*       |                                                       |
*       +-------------------------------------------------------+
*
FUNCTION _qt10sgkcl     &&  m.fk_no WHEN
#REGION 2
on key label ? do notliste.spr

*       +-------------------------------------------------------+
*       |                                                       |
*       | _QT10SGLMV            m.fk_no VALID                   |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:          VIZEGIR,     Record Number:  13 |
*       | Variable:             m.fk_no                         |
*       | Called By:            VALID Clause                    |
*       | Object Type:          Field                           |
*       | Snippet Number:       3                               |
*       |                                                       |
*       +-------------------------------------------------------+
*


FUNCTION _qt10sglmv     &&  m.fk_no VALID
#REGION 2
on key label ?
=showsave()
```

```
*       +-------------------------------------------------------+
*       |                                                       |
*       | _QT10SGNCV            m.adi VALID                     |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:          VIZEGIR,     Record Number:  14 |
*       | Variable:             m.adi                           |
*       | Called By:            VALID Clause                    |
*       | Object Type:          Field                           |
*       | Snippet Number:       4                               |
*       |                                                       |
*       +-------------------------------------------------------+
*
FUNCTION _qt10sgncv     &&  m.adi VALID
#REGION 2
if not empty(m.adi)
      gecicigoz =proper(alltrim(m.adi))
      gercekgoz =""
      uzunlugu  =len(gecicigoz)
      flag      =.t.
      for i=1 to uzunlugu
           if empty(substr(gecicigoz,i,1))
              if flag
                 gercekgoz=padr(gercekgoz,len(gercekgoz)+1," ")
                 flag=.f.
               endif
            else
gercekgoz=padr(gercekgoz,len(gercekgoz)+1,substr(gecicigoz,i,1))
              flag=.t.
            endif
       endfor
       gercekgoz=padr(gercekgoz,len(notlar.adi))
       m.adi=gercekgoz
       show get m.adi
endif
=showsave()

*       +-------------------------------------------------------+
*       |                                                       |
*       | _QT10SGPU7            m.soyadi VALID                  |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:          VIZEGIR,     Record Number:  15 |
*       | Variable:             m.soyadi                        |
*       | Called By:            VALID Clause                    |
*       | Object Type:          Field                           |
*       | Snippet Number:       5                               |
*       |                                                       |
*       +-------------------------------------------------------+
*
FUNCTION _qt10sgpu7     &&  m.soyadi VALID
#REGION 2
if not empty(m.soyadi)
      gecicigoz =proper(alltrim(m.soyadi))
      gercekgoz =""
      uzunlugu  =len(gecicigoz)
      flag      =.t.
      for i=1 to uzunlugu
           if empty(substr(gecicigoz,i,1))
              if flag
                 gercekgoz=padr(gercekgoz,len(gercekgoz)+1," ")
                 flag=.f.
```

```
                endif
             else
gercekgoz=padr(gercekgoz,len(gercekgoz)+1,substr(gecicigoz,i,1))
                flag=.t.
             endif
       endfor
       gercekgoz=padr(gercekgoz,len(notlar.soyadi))
       m.soyadi=gercekgoz
       show get m.soyadi
endif
=showsave()

*       +------------------------------------------------------+
*       |                                                      |
*       |  _QT10SGRCC            m.ogrt_yil VALID              |
*       |                                                      |
*       |  Function Origin:                                    |
*       |                                                      |
*       |  From Screen:          VIZEGIR,     Record Number:  16 |
*       |  Variable:             m.ogrt_yil                    |
*       |  Called By:            VALID Clause                  |
*       |  Object Type:          Field                         |
*       |  Snippet Number:       6                             |
*       |                                                      |
*       +------------------------------------------------------+
FUNCTION _qt10sgrcc     &&  m.ogrt_yil VALID
#REGION 2
=showsave()
*       +------------------------------------------------------+
*       |                                                      |
*       |  _QT10SGT4O            m.sinif VALID                 |
*       |                                                      |
*       |  Function Origin:                                    |
*       |                                                      |
*       |  From Screen:          VIZEGIR,     Record Number:  17 |
*       |  Variable:             m.sinif                       |
*       |  Called By:            VALID Clause                  |
*       |  Object Type:          Field                         |
*       |  Snippet Number:       7                             |
*       |                                                      |
*       +------------------------------------------------------+
FUNCTION _qt10sgt4o     &&  m.sinif VALID
#REGION 2
=showsave()
*       +------------------------------------------------------+
*       |                                                      |
*       |  _QT10SGUGT            m.somest VALID                |
*       |                                                      |
*       |  Function Origin:                                    |
*       |                                                      |
*       |  From Screen:          VIZEGIR,     Record Number:  18 |
*       |  Variable:             m.somest                      |
*       |  Called By:            VALID Clause                  |
*       |  Object Type:          Field                         |
*       |  Snippet Number:       8                             |
*       |                                                      |
*       +------------------------------------------------------+
FUNCTION _qt10sgugt     &&  m.somest VALID
#REGION 2
=showsave()
```

```
*       +-------------------------------------------------------+
*       |                                                       |
*       |  _QT10SGVUR            m.ders VALID                   |
*       |                                                       |
*       |  Function Origin:                                     |
*       |                                                       |
*       |  From Screen:          VIZEGIR,     Record Number:  19 |
*       |  Variable:             m.ders                         |
*       |  Called By:            VALID Clause                   |
*       |  Object Type:          Field                          |
*       |  Snippet Number:       9                              |
*       |                                                       |
*       +-------------------------------------------------------+
*
FUNCTION _qt10sgvur     &&  m.ders VALID
#REGION 2

if not empty(m.ders)
      gecicigoz =proper(alltrim(m.ders))
      gercekgoz =""
      uzunlugu  =len(gecicigoz)
      flag      =.t.
      for i=1 to uzunlugu
           if empty(substr(gecicigoz,i,1))
              if flag
                 gercekgoz=padr(gercekgoz,len(gercekgoz)+1," ")
                 flag=.f.
              endif
           else
gercekgoz=padr(gercekgoz,len(gercekgoz)+1,substr(gecicigoz,i,1))
              flag=.t.
           endif
       endfor
       gercekgoz=padr(gercekgoz,len(notlar.ders))
       m.ders=gercekgoz
       show get m.ders
endif
=showsave()

*       +-------------------------------------------------------+
*       |                                                       |
*       |  _QT10SGXBZ            m.vz1 VALID                    |
*       |                                                       |
*       |  Function Origin:                                     |
*       |                                                       |
*       |  From Screen:          VIZEGIR,     Record Number:  20 |
*       |  Variable:             m.vz1                          |
*       |  Called By:            VALID Clause                   |
*       |  Object Type:          Field                          |
*       |  Snippet Number:       10                             |
*       |                                                       |
*       +-------------------------------------------------------+
*
FUNCTION _qt10sgxbz     &&  m.vz1 VALID
#REGION 2
 IF m.vz1 > 0
m.av=m.av+1
m.sum=m.sum+m.vz1
ENDIF
SHOW GETS
=showsave()
```

```
*        +-------------------------------------------------------+
*        |                                                       |
*        |  _QT10SGYKE            m.vz2 VALID                    |
*        |                                                       |
*        |  Function Origin:                                     |
*        |                                                       |
*        |  From Screen:          VIZEGIR,     Record Number:  21 |
*        |  Variable:             m.vz2                          |
*        |  Called By:            VALID Clause                   |
*        |  Object Type:          Field                          |
*        |  Snippet Number:       11                             |
*        |                                                       |
*        +-------------------------------------------------------+
*
FUNCTION _qt10sgyke     &&  m.vz2 VALID
#REGION 2
 IF m.vz2 > 0
m.av=m.av+1
m.sum=m.sum+m.vz2
ENDIF
SHOW GETS
=showsave()

*        +-------------------------------------------------------+
*        |                                                       |
*        |  _QT10SH0CA            m.vz3 VALID                    |
*        |                                                       |
*        |  Function Origin:                                     |
*        |                                                       |
*        |  From Screen:          VIZEGIR,     Record Number:  22 |
*        |  Variable:             m.vz3                          |
*        |  Called By:            VALID Clause                   |
*        |  Object Type:          Field                          |
*        |  Snippet Number:       12                             |
*        |                                                       |
*        +-------------------------------------------------------+
*
FUNCTION _qt10sh0ca     &&  m.vz3 VALID
#REGION 2
 IF m.vz3 > 0
m.av=m.av+1
m.sum=m.sum+m.vz3
ENDIF
SHOW GETS
=showsave()

*        +-------------------------------------------------------+
```

```
*         |                                                          |
*         |  _QT10SH23O            m.vize VALID                      |
*         |                                                          |
*         |  Function Origin:                                        |
*         |                                                          |
*         |  From Screen:          VIZEGIR,     Record Number:   23  |
*         |  Variable:             m.vize                            |
*         |  Called By:            VALID Clause                      |
*         |  Object Type:          Field                             |
*         |  Snippet Number:       13                                |
*         |                                                          |
*         +----------------------------------------------------------+
*
FUNCTION _qt10sh23o     &&  m.vize VALID
#REGION 2
m.vize =ROUND((m.sum/m.av),0)
show get m.vize
SHOW GETS
=showsave()
m.av =0
m.sum=0
*         +----------------------------------------------------------+
*         |                                                          |
*         |  _QT10SH3RC            m.newrecord VALID                 |
*         |                                                          |
*         |  Function Origin:                                        |
*         |                                                          |
*         |  From Screen:          VIZEGIR,     Record Number:   24  |
*         |  Variable:             m.newrecord                       |
*         |  Called By:            VALID Clause                      |
*         |  Object Type:          Push Button                       |
*         |  Snippet Number:       14                                |
*         |                                                          |
*         +----------------------------------------------------------+
*
FUNCTION _qt10sh3rc     &&  m.newrecord VALID
#REGION 2
m.adding = .T.

SCATTER MEMVAR BLANK MEMO

SHOW GET m.newrecord DISABLE
SHOW GET m.saverecord ENABLE
SHOW GET m.cancel ENABLE

_CUROBJ = OBJNUM(m.fk_no)

SHOW GETS
SHOW GETS DISABLE WINDOW controls

*         +----------------------------------------------------------+
*         |                                                          |
*         |  _QT10SH5FJ            m.saverecord WHEN                 |
*         |                                                          |
*         |  Function Origin:                                        |
*         |                                                          |
*         |  From Screen:          VIZEGIR,     Record Number:   25  |
*         |  Variable:             m.saverecord                      |
*         |  Called By:            WHEN Clause                       |
*         |  Object Type:          Push Button                       |
*         |  Snippet Number:       15                                |
*         |                                                          |
*         +----------------------------------------------------------+
```

```
FUNCTION _qt10sh5fj     &&  m.saverecord WHEN
#REGION 2
IF EMPTY(m.FK_NO)
 ?? CHR(7)
 WAIT WINDOW "FAKÜLTE NUMARASINI GIRIN" NOWAIT
 _CUROBJ = OBJNUM(m.fk_no)
 RETURN .F.
ENDIF
*       +-------------------------------------------------------+
*       |                                                       |
*       |  _QT10SH6N8             m.saverecord VALID            |
*       |                                                       |
*       |  Function Origin:                                     |
*       |                                                       |
*       |  From Screen:           VIZEGIR,     Record Number:   25
*       |  Variable:              m.saverecord                  |
*       |  Called By:             VALID Clause                  |
*       |  Object Type:           Push Button                   |
*       |  Snippet Number:        16                            |
*       |                                                       |
*       +-------------------------------------------------------+
FUNCTION _qt10sh6n8     &&  m.saverecord VALID
#REGION 2
IF m.adding
 APPEND BLANK
 m.adding = .F.
ENDIF
m.editing = .F.
GATHER MEMVAR MEMO
SHOW GET m.newrecord ENABLE
SHOW GET m.saverecord DISABLE
SHOW GET m.cancel DISABLE
WAIT WINDOW "KAYIT SAKLANDI" NOWAIT
_CUROBJ = OBJNUM(m.cont, 1)
SHOW GET m.cont, 5 ENABLE
SHOW GET m.cont, 6 ENABLE
SHOW GETS
*       +-------------------------------------------------------+
*       |                                                       |
*       |  _QT10SH7JN             m.cancel VALID                |
*       |                                                       |
*       |  Function Origin:                                     |
*       |                                                       |
*       |  From Screen:           VIZEGIR,     Record Number:   26
*       |  Variable:              m.cancel                      |
*       |  Called By:             VALID Clause                  |
*       |  Object Type:           Push Button                   |
*       |  Snippet Number:        17                            |
*       |                                                       |
*       +-------------------------------------------------------+

FUNCTION _qt10sh7jn     &&  m.cancel VALID
#REGION 2
SCATTER MEMVAR MEMO
m.editing = .F.
m.adding = .F.
SHOW GET m.newrecord ENABLE
SHOW GET m.saverecord DISABLE
SHOW GET m.cancel DISABLE
_CUROBJ = OBJNUM(m.cont, 1)
SHOW GET m.cont, 5 ENABLE
SHOW GET m.cont, 6 ENABLE
SHOW GETS
```

```
*        +-------------------------------------------------------+
*        |                                                       |
*        |  _QT10SH8AH             m.yardim VALID                |
*        |                                                       |
*        |  Function Origin:                                     |
*        |                                                       |
*        |  From Screen:           VIZEGIR,     Record Number:   27
*        |  Variable:              m.yardim                      |
*        |  Called By:             VALID Clause                  |
*        |  Object Type:           Push Button                   |
*        |  Snippet Number:        18                            |
*        |                                                       |
*        +-------------------------------------------------------+
*
FUNCTION _qt10sh8ah      &&  m.yardim VALID
#REGION 2
DO VZYARDIM.SPR
*        +-------------------------------------------------------+
*        |                                                       |
*        |  _QT10SH8Y1             m.sil VALID                   |
*        |                                                       |
*        |  Function Origin:                                     |
*        |                                                       |
*        |  From Screen:           VIZEGIR,     Record Number:   28
*        |  Variable:              m.sil                         |
*        |  Called By:             VALID Clause                  |
*        |  Object Type:           Push Button                   |
*        |  Snippet Number:        19                            |
*        |                                                       |
*        +-------------------------------------------------------+
*
FUNCTION _qt10sh8y1      &&  m.sil VALID
#REGION 2
do noteha.spr

*        +-------------------------------------------------------+
*        |                                                       |
*        |  _QT10SHAGG             Read Level When               |
*        |                                                       |
*        |  Function Origin:                                     |
*        |                                                       |
*        |  From Screen:           Multiple Screens              |
*        |  Called By:             READ Statement                |
*        |  Snippet Number:        20                            |
*        |                                                       |
*        +-------------------------------------------------------+
*
FUNCTION _qt10shagg      && Read Level When


*
* When Code from screen: VIZEGIR
*
#REGION 2
IF VAL(SYS(1001)) < 223000
SET SKIP OF BAR 1 OF REPORT .T.
ENDIF
```

```
*        +-------------------------------------------------------+
*        |                                                       |
*        | _QT10SHAGK            Read Level Deactivate           |
*        |                                                       |
*        | Function Origin:                                      |
*        |                                                       |
*        | From Screen:          Multiple Screens                |
*        | Called By:            READ Statement                  |
*        | Snippet Number:       21                              |
*        |                                                       |
*        +-------------------------------------------------------+


FUNCTION _qt10shagk     && Read Level Deactivate
*
* Deactivate Code from screen: VIZEGIR
*
#REGION 2
IF m.editing
 ?? CHR(7)
 WAIT WINDOW "LÜTFEN DE*ÿPÿKLÿKLERÿ ONAYLAYINIZ" NOWAIT
 RETURN .F.
ENDIF
RETURN NOT WREAD()

*        +-------------------------------------------------------+
*        |                                                       |
*        | _QT10SGHD8            Read Level Show                 |
*        |                                                       |
*        | Function Origin:                                      |
*        |                                                       |
*        | From Screen:          Multiple Screens                |
*        | Called By:            READ Statement                  |
*        | Snippet Number:       22                              |
*        |                                                       |
*        +-------------------------------------------------------+
*
FUNCTION _qt10sghd8     && Read Level Show
PRIVATE currwind
STORE WOUTPUT() TO currwind
*
* Show Code from screen: VCONT
*
#REGION 1
IF EOF()
 GO BOTTOM
ENDIF
m.saverecno = RECNO()
GO TOP
m.toprec = RECNO()
GO BOTTOM
m.bottomrec = RECNO()
GO m.saverecno

IF RECNO() = m.bottomrec
 SHOW GET m.cont, 1 DISABLE
 SHOW GET m.cont, 2 ENABLE
 SHOW GET m.cont, 3 ENABLE
 SHOW GET m.cont, 4 DISABLE
ELSE
 IF RECNO() = m.toprec
  SHOW GET m.cont, 1 ENABLE
  SHOW GET m.cont, 2 DISABLE
```

```
  SHOW GET m.cont, 3 DISABLE
  SHOW GET m.cont, 4 ENABLE
 ELSE
  SHOW GET m.cont ENABLE
 ENDIF
ENDIF
*
* Show Code from screen: VIZEGIR
*
#REGION 2
IF NOT m.adding
SCATTER MEMVAR MEMO
ENDIF
IF NOT EMPTY(currwind)
 ACTIVATE WINDOW (currwind) SAME
ENDIF
```

```
*        +-------------------------------------------------------+
*        |                                                       |
*        | 31/08/94            FINALGIR.SPR              15:07:38 |
*        |                                                       |
*        |-------------------------------------------------------|
*        |                                                       |
*        | Author's Name                                         |
*        |                                                       |
*        | Copyright (c) 1994 Company Name                       |
*        | Address                                               |
*        | City,     Zip                                         |
*        |                                                       |
*        | Description:                                          |
*        | This program was automatically generated by GENSCRN.  |
*        |                                                       |
*        +-------------------------------------------------------+
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
 SET TALK OFF
 m.talkstat = "ON"
ELSE
 m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
m.currarea = SELECT()
*        +-------------------------------------------------------+
*        |                                                       |
*        |                    Window definitions                 |
*        |                                                       |
*        +-------------------------------------------------------+
*
IF NOT WEXIST("controls")
 DEFINE WINDOW controls ;
  FROM 20, 4 ;
  TO 22,77 ;
  NOFLOAT ;
  NOCLOSE ;
  COLOR SCHEME 22
ENDIF

IF NOT WEXIST("final")
 DEFINE WINDOW final ;
  FROM 4, 10 ;
  TO 18,70 ;
  TITLE " FINAL NOT GİRİŞİ" ;
  NOFLOAT ;
  NOCLOSE ;
  SHADOW ;
  COLOR SCHEME 2
ENDIF
```

```
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦              FCONT Setup Code - SECTION 2             ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
*

#REGION 1
REGIONAL m.cont, m.toprec, m.bottomrec, m.saverecno, m.quitting
m.quitting = .F.
m.cont = "Tamam"

IF EOF()
 GO BOTTOM
ENDIF

m.saverecno = RECNO()
GO TOP
m.toprec = RECNO()
GO BOTTOM
m.bottomrec = RECNO()
GO m.saverecno


*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦            FINALGIR Setup Code - SECTION 2            ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
*

#REGION 2
set echo off
set escape off
set deleted on
set exact on
set date british
close databases

USE  \yildiz\ydbf\notlar.dbf
SET ORDER TO 1

PRIVATE m.adding, m.editing,m.yz,m.vz,m.fn

m.adding = .F.
m.editing = .F.
m.yz=1
m.vz=0
m.fn=0

SCATTER MEMVAR MEMO
```

```
*        +-------------------------------------------------------+
*        ¦                                                       ¦
*        ¦                    FCONT Screen Layout                ¦
*        ¦                                                       ¦
*        +-------------------------------------------------------+
*

#REGION 1
IF WVISIBLE("controls")
 ACTIVATE WINDOW controls SAME
ELSE
 ACTIVATE WINDOW controls NOSHOW
ENDIF
@ 0,5 GET m.cont ;
 PICTURE "@*HN so\<Nraki;ön\<Ceki;bi\<Rinci;s\<Onuncu;\<Ara;\?\!\<Tamam"
;
 SIZE 1,9,2 ;
 DEFAULT 1 ;
 VALID _qt10wfid6()

*        +-------------------------------------------------------+
*        ¦                                                       ¦
*        ¦                  FINALGIR Screen Layout               ¦
*        ¦                                                       ¦
*        +-------------------------------------------------------+
*

#REGION 2
IF WVISIBLE("final")
 ACTIVATE WINDOW final SAME
ELSE
 ACTIVATE WINDOW final NOSHOW
ENDIF
@ 0,0 TO 0,47 ;
 COLOR GR+/W
@ 0,0 TO 12,0 ;
 COLOR GR+/W
@ 12,0 TO 12,47 ;
 COLOR GR+/W
@ 0,47 TO 12,47 ;
 COLOR GR+/W
@ 0,16 TO 12,16 DOUBLE ;
 COLOR GR+/W
@ 12,16 SAY "-" ;
 COLOR GR+/W
@ 0,16 SAY "-" ;
 COLOR GR+/W
@ 12,47 SAY "+" ;
 COLOR GR+/W
@ 0,47 SAY "+" ;
 COLOR GR+/W
@ 12,0 SAY "+" ;

 COLOR GR+/W
@ 0,0 SAY "+" ;
 COLOR GR+/W
```

```
@ 2,1 SAY "ADI VE SOYADI" ;
 COLOR W+/W
@ 4,1 SAY "SINIF........." ;
 COLOR W+/W
@ 5,1 SAY "SOMESTRE......" ;
 COLOR W+/W
@ 7,1 SAY "DERS ADI......" ;
 COLOR W+/W
@ 8,1 SAY "VİZE ORT.NOTU" ;
 COLOR W+/W
@ 9,1 SAY "FINAL NOTU...." ;
 COLOR W+/W
@ 10,1 SAY "SONUC NOTU...." ;
 COLOR W+/W
@ 10,33 SAY "VIZE %" ;
 COLOR W+/W
@ 6,1 SAY "FINAL TARİHİ" ;
 COLOR W+/W
@ 11,1 SAY "BAPARI DURUM" ;
 COLOR W+/W
@ 1,1 SAY "ÖĞRENCİ NUMARA" ;
 COLOR W+/W
@ 3,1 SAY "ÖĞRETİM YIL" ;
 COLOR W+/W
@ 0,48 TO 12,58 ;
 COLOR GR+/W
@ 3,12 SAY "I" ;
 COLOR W+/W
@ 1,17 GET m.fk_no ;
 SIZE 1,7 ;
 DEFAULT " " ;
 VALID _qt10wfn3g() ;
 COLOR SCHEME 5
@ 2,17 GET m.adi ;
 SIZE 1,13 ;
 DEFAULT " " ;
 VALID _qt10wfnwy() ;
 COLOR SCHEME 5
@ 2,30 GET m.soyadi ;
 SIZE 1,12 ;
 DEFAULT " " ;
 VALID _qt10wfob9() ;
 COLOR SCHEME 5
@ 3,17 GET m.ogrt_yil ;
 SIZE 1,8 ;
 DEFAULT " " ;
 VALID _qt10wfonv() ;
 COLOR SCHEME 5
@ 4,17 GET m.sinif ;
 SIZE 1,4 ;

 DEFAULT " " ;
 VALID _qt10wfpr7() ;
 COLOR SCHEME 5
@ 5,17 GET m.somest ;
 SIZE 1,4 ;
```

```
 DEFAULT " " ;
 VALID _qt10wfqeb() ;
 COLOR SCHEME 5
@ 6,17 GET m.final_tar ;
 SIZE 1,8 ;
 DEFAULT (  /  /  ) ;

 PICTURE "@E" ;
 VALID _qt10wfqmq() ;
 COLOR SCHEME 5
@ 7,17 GET m.ders ;
 SIZE 1,25 ;
 DEFAULT " " ;
 VALID _qt10wfqtm() ;
 COLOR SCHEME 5
@ 8,17 GET m.vize ;
 SIZE 1,4 ;
 DEFAULT 0 ;
 PICTURE "@Z" ;
 VALID _qt10wfr97() ;
 COLOR SCHEME 5
@ 9,17 GET m.fina ;
 SIZE 1,4 ;
 DEFAULT 0 ;
 PICTURE "@Z" ;
 VALID _qt10wfrg6() ;
 COLOR SCHEME 5
@ 10,17 GET m.sonuc ;
 SIZE 1,4 ;
 DEFAULT 0 ;
 PICTURE "@Z" ;
 VALID _qt10wfrr6() ;
 COLOR SCHEME 5
@ 11,17 GET m.basari ;
 SIZE 1,10 ;
 DEFAULT " " ;
 VALID _qt10wfsdl() ;
 COLOR SCHEME 5
@ 2,49 GET m.newrecord ;
 PICTURE "@*VN \<Yeni" ;
 SIZE 1,8,2 ;
 DEFAULT 1 ;
 VALID _qt10wfsvu() ;
 COLOR SCHEME 1
@ 5,49 GET m.saverecord ;
 PICTURE "@*VN \<Sakla" ;
 SIZE 1,9,2 ;
 DEFAULT 1 ;

 WHEN _qt10wft9t() ;
 VALID _qt10wftgx() ;
 DISABLE ;
 COLOR SCHEME 1
@ 8,49 GET m.cancel ;
 PICTURE "@*VN \<Iptal" ;
 SIZE 1,9,2 ;
```

```
 DEFAULT 1 ;
 VALID _qt10wftrx() ;
 DISABLE ;
 COLOR SCHEME 1
@ 11,49 GET m.sil ;
 PICTURE "@*HN Sí\<L" ;
 SIZE 1,9,1 ;
 DEFAULT 1 ;
 VALID _qt10wfumk() ;
 COLOR SCHEME 1
@ 9,39 GET m.yz ;

 PICTURE "@^ 10;20;30;40;50;60;70" ;
 SIZE 3,6 ;
 DEFAULT "10" ;
 VALID _qt10wfvcy() ;
 COLOR SCHEME 6, 6

IF NOT WVISIBLE("final")
 ACTIVATE WINDOW final
ENDIF
IF NOT WVISIBLE("controls")
 ACTIVATE WINDOW controls
ENDIF


READ CYCLE ;
 DEACTIVATE _qt10wfwyc() ;
 SHOW _qt10wfka2()

RELEASE WINDOW controls
RELEASE WINDOW final
SELECT (m.currarea)


#REGION 0
IF m.talkstat = "ON"
 SET TALK ON
ENDIF
IF m.compstat = "ON"
 SET COMPATIBLE ON
ENDIF

*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦                  FINALGIR Cleanup Code                ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
*

#REGION 2
IF m.quitting
 RELEASE WINDOW FINAL
 RELEASE WINDOW controls
ENDIF
```

```
*
* SHOWSAVE
*
PROCEDURE showsave
IF NOT m.editing
 WAIT WINDOW "LÜTFFEN DEĞİŞİKLİKLERİ ONAYLAYINIZ" NOWAIT
ENDIF

SHOW GET m.newrecord DISABLE
SHOW GET m.saverecord ENABLE
SHOW GET m.cancel ENABLE
SHOW GETS ONLY DISABLE WINDOW controls

m.editing = .T.


*        +--------------------------------------------------------+
*        ¦                                                        ¦
*        ¦ _QT10WFID6            m.cont VALID                     ¦
*        ¦                                                        ¦
*        ¦ Function Origin:                                       ¦
*        ¦                                                        ¦
*        ¦ From Screen:          FCONT,     Record Number:    2   ¦
*        ¦ Variable:             m.cont                           ¦
*        ¦ Called By:            VALID Clause                     ¦
*        ¦ Object Type:          Push Button                      ¦
*        ¦ Snippet Number:       1                                ¦
*        ¦                                                        ¦
*        +--------------------------------------------------------+
FUNCTION _qt10wfid6     &&  m.cont VALID
#REGION 1
DO CASE
CASE m.cont = "soNraki"
 SKIP 1
 IF RECNO() = m.bottomrec
  SHOW GET m.cont, 1 DISABLE
  SHOW GET m.cont, 4 DISABLE


 ELSE
  IF RECNO() > m.toprec
   SHOW GET m.cont, 2 ENABLE
   SHOW GET m.cont, 3 ENABLE
  ENDIF
 ENDIF
CASE m.cont = "önCeki"
 SKIP -1
 IF RECNO() = m.toprec
  SHOW GET m.cont, 2 DISABLE
  SHOW GET m.cont, 3 DISABLE
 ELSE
  IF RECNO() < m.bottomrec
   SHOW GET m.cont, 1 ENABLE
   SHOW GET m.cont, 4 ENABLE
  ENDIF
 ENDIF
```

```
CASE m.cont = "biRinci"
 GO TOP
 SHOW GET m.cont, 1 ENABLE
 SHOW GET m.cont, 2 DISABLE
 SHOW GET m.cont, 3 DISABLE
 SHOW GET m.cont, 4 ENABLE
CASE m.cont = "sOnuncu"
 GO BOTTOM
 SHOW GET m.cont, 1 DISABLE
 SHOW GET m.cont, 2 ENABLE
 SHOW GET m.cont, 3 ENABLE
 SHOW GET m.cont, 4 DISABLE
CASE m.cont = "Ara"
     do fbrowser.spr
CASE m.cont = "Tamam"
 m.idlequit = .T.
 m.quitting = .T.
 CLEAR READ
ENDCASE
SHOW GETS
```

```
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QT10WFN3G           m.fk_no VALID                    ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:         FINALGIR,     Record Number:  27 ¦
*       ¦ Variable:            m.fk_no                          ¦
*       ¦ Called By:           VALID Clause                     ¦
*       ¦ Object Type:         Field                            ¦
*       ¦ Snippet Number:      2                                ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
*
```

```
FUNCTION _qt10wfn3g     &&  m.fk_no VALID
#REGION 2
=showsave()
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QT10WFNWY           m.adi VALID                      ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:         FINALGIR,     Record Number:  28 ¦
*       ¦ Variable:            m.adi                            ¦
*       ¦ Called By:           VALID Clause                     ¦
*       ¦ Object Type:         Field                            ¦
*       ¦ Snippet Number:      3                                ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
FUNCTION _qt10wfnwy     &&  m.adi VALID
#REGION 2
```

```
if not empty(m.adi)
      gecicigoz =proper(alltrim(m.adi))
      gercekgoz =""
      uzunlugu  =len(gecicigoz)
      flag      =.t.
      for i=1 to uzunlugu
           if empty(substr(gecicigoz,i,1))
              if flag
                 gercekgoz=padr(gercekgoz,len(gercekgoz)+1," ")
                 flag=.f.
               endif
            else
gercekgoz=padr(gercekgoz,len(gercekgoz)+1,substr(gecicigoz,i,1))
               flag=.t.
            endif
       endfor
       gercekgoz=padr(gercekgoz,len(notlar.adi))
       m.adi=gercekgoz
       show get m.adi
endif
=showsave()
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QT10WFOB9           m.soyadi VALID                   ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:         FINALGIR,     Record Number:  29 ¦
*       ¦ Variable:            m.soyadi                         ¦
*       ¦ Called By:           VALID Clause                     ¦
*       ¦ Object Type:         Field                            ¦
*       ¦ Snippet Number:      4                                ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
FUNCTION _qt10wfob9     &&  m.soyadi VALID
#REGION 2
if not empty(m.soyadi)
      gecicigoz =proper(alltrim(m.soyadi))
      gercekgoz =""
      uzunlugu  =len(gecicigoz)
      flag      =.t.
      for i=1 to uzunlugu
           if empty(substr(gecicigoz,i,1))
              if flag
                 gercekgoz=padr(gercekgoz,len(gercekgoz)+1," ")
                 flag=.f.
               endif
            else
gercekgoz=padr(gercekgoz,len(gercekgoz)+1,substr(gecicigoz,i,1))
               flag=.t.
            endif
       endfor
       gercekgoz=padr(gercekgoz,len(notlar.soyadi))
       m.soyadi=gercekgoz
       show get m.soyadi
endif
```

```
=showsave()
*        +-------------------------------------------------------+
*        ¦                                                       ¦
*        ¦ _QT10WFONV            m.ogrt_yil VALID                ¦
*        ¦                                                       ¦
*        ¦ Function Origin:                                      ¦
*        ¦                                                       ¦
*        ¦ From Screen:          FINALGIR,      Record Number:   30
*        ¦ Variable:             m.ogrt_yil                      ¦
*        ¦ Called By:            VALID Clause                    ¦
*        ¦ Object Type:          Field                           ¦
*        ¦ Snippet Number:       5                               ¦
*        ¦                                                       ¦
*        +-------------------------------------------------------+
*
FUNCTION _qt10wfonv     &&  m.ogrt_yil VALID
#REGION 2
 =showsave()
*        +-------------------------------------------------------+
*        ¦                                                       ¦
*        ¦ _QT10WFPR7            m.sinif VALID                   ¦
*        ¦                                                       ¦
*        ¦ Function Origin:                                      ¦
*        ¦                                                       ¦
*        ¦ From Screen:          FINALGIR,      Record Number:   31
*        ¦ Variable:             m.sinif                         ¦
*        ¦ Called By:            VALID Clause                    ¦
*        ¦ Object Type:          Field                           ¦
*        ¦ Snippet Number:       6                               ¦
*        ¦                                                       ¦
*        +-------------------------------------------------------+
FUNCTION _qt10wfpr7     &&  m.sinif VALID
#REGION 2
=showsave()
*        +-------------------------------------------------------+
*        ¦                                                       ¦
*        ¦ _QT10WFQEB            m.somest VALID                  ¦
*        ¦                                                       ¦
*        ¦ Function Origin:                                      ¦
*        ¦                                                       ¦
*        ¦ From Screen:          FINALGIR,      Record Number:   32
*        ¦ Variable:             m.somest                        ¦
*        ¦ Called By:            VALID Clause                    ¦
*        ¦ Object Type:          Field                           ¦
*        ¦ Snippet Number:       7                               ¦
*        ¦                                                       ¦
*        +-------------------------------------------------------+
*
FUNCTION _qt10wfqeb     &&  m.somest VALID
#REGION 2
=showsave()
```

```
*       +-------------------------------------------------------+
*       |                                                       |
*       | _QT10WFQMQ            m.final_tar VALID               |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:          FINALGIR,     Record Number:  33|
*       | Variable:             m.final_tar                     |
*       | Called By:            VALID Clause                    |
*       | Object Type:          Field                           |
*       | Snippet Number:       8                               |
*       |                                                       |
*       +-------------------------------------------------------+
FUNCTION _qt10wfqmq     &&  m.final_tar VALID
#REGION 2
=showsave()
*       +-------------------------------------------------------+
*       |                                                       |
*       | _QT10WFQTM            m.ders VALID                    |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:          FINALGIR,     Record Number:  34|
*       | Variable:             m.ders                          |
*       | Called By:            VALID Clause                    |
*       | Object Type:          Field                           |
*       | Snippet Number:       9                               |
*       |                                                       |
*       +-------------------------------------------------------+
FUNCTION _qt10wfqtm     &&  m.ders VALID
#REGION 2
if not empty(m.ders)
      gecicigoz =proper(alltrim(m.ders))
      gercekgoz =""
      uzunlugu  =len(gecicigoz)
      flag      =.t.
      for i=1 to uzunlugu
           if empty(substr(gecicigoz,i,1))
              if flag
                 gercekgoz=padr(gercekgoz,len(gercekgoz)+1," ")
                 flag=.f.
               endif
            else
gercekgoz=padr(gercekgoz,len(gercekgoz)+1,substr(gecicigoz,i,1))
               flag=.t.
            endif
       endfor
       gercekgoz=padr(gercekgoz,len(notlar.ders))
       m.ders=gercekgoz
       show get m.ders
endif
=showsave()
```

```
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QT10WFR97            m.vize VALID                    ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:          FINALGIR,     Record Number:  35¦
*       ¦ Variable:             m.vize                          ¦
*       ¦ Called By:            VALID Clause                    ¦
*       ¦ Object Type:          Field                           ¦
*       ¦ Snippet Number:       10                              ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
FUNCTION _qt10wfr97     &&  m.vize VALID
#REGION 2
=showsave()
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QT10WFRG6            m.fina VALID                    ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:          FINALGIR,     Record Number:  36¦
*       ¦ Variable:             m.fina                          ¦
*       ¦ Called By:            VALID Clause                    ¦
*       ¦ Object Type:          Field                           ¦
*       ¦ Snippet Number:       11                              ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
*
FUNCTION _qt10wfrg6     &&  m.fina VALID
#REGION 2
m.tp1=(m.vize * m.vz)/100
m.tp2=(m.fina * m.fn)/100
m.sonuc=ROUND((m.tp1+m.tp2),0)
show get m.sonuc
IF m.sonuc >= 50
STORE "BAPARILI" TO m.basari
 show get m.basari
wait window "BUTUNLEME YOK";
    nowait
ELSE
STORE "BAPARISIZ" TO m.basari
 show get m.basari

wait window "BUTUNLEMEVAR";
nowait
  ENDIF
=showsave()
```

```
*       +-------------------------------------------------------+
*       |                                                       |
*       | _QT10WFRR6            m.sonuc VALID                   |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:          FINALGIR,     Record Number:  37|
*       | Variable:             m.sonuc                         |
*       | Called By:            VALID Clause                    |
*       | Object Type:          Field                           |
*       | Snippet Number:       12                              |
*       |                                                       |
*       +-------------------------------------------------------+
*

FUNCTION _qt10wfrr6     &&  m.sonuc VALID
#REGION 2
=showsave()

*       +-------------------------------------------------------+
*       |                                                       |
*       | _QT10WFSDL            m.basari VALID                  |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:          FINALGIR,     Record Number:  38|
*       | Variable:             m.basari                        |
*       | Called By:            VALID Clause                    |
*       | Object Type:          Field                           |
*       | Snippet Number:       13                              |
*       |                                                       |
*       +-------------------------------------------------------+
FUNCTION _qt10wfsdl     &&  m.basari VALID
#REGION 2
=showsave()
*       +-------------------------------------------------------+
*       |                                                       |
*       | _QT10WFSVU            m.newrecord VALID               |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:          FINALGIR,     Record Number:  39|
*       | Variable:             m.newrecord                     |
*       | Called By:            VALID Clause                    |
*       | Object Type:          Push Button                     |
*       | Snippet Number:       14                              |
*       |                                                       |
*       +-------------------------------------------------------+
FUNCTION _qt10wfsvu     &&  m.newrecord VALID
#REGION 2
m.adding = .T.
SCATTER MEMVAR BLANK MEMO
SHOW GET m.newrecord DISABLE
SHOW GET m.saverecord ENABLE
SHOW GET m.cancel ENABLE
_CUROBJ = OBJNUM(m.fk_no)
```

```
SHOW GETS
SHOW GETS DISABLE WINDOW controls
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QT10WFT9T            m.saverecord WHEN               ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:          FINALGIR,     Record Number:  40¦
*       ¦ Variable:             m.saverecord                    ¦
*       ¦ Called By:            WHEN Clause                     ¦
*       ¦ Object Type:          Push Button                     ¦
*       ¦ Snippet Number:       15                              ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+

FUNCTION _qt10wft9t     &&  m.saverecord WHEN
#REGION 2
IF EMPTY(m.FK_NO)
 ?? CHR(7)
 WAIT WINDOW "FAKÜLTE NUMARASINI GIRIN" NOWAIT
 _CUROBJ = OBJNUM(m.fk_no)
 RETURN .F.
ENDIF
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QT10WFTGX            m.saverecord VALID              ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:          FINALGIR,     Record Number:  40¦
*       ¦ Variable:             m.saverecord                    ¦
*       ¦ Called By:            VALID Clause                    ¦
*       ¦ Object Type:          Push Button                     ¦
*       ¦ Snippet Number:       16                              ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
FUNCTION _qt10wftgx     &&  m.saverecord VALID
#REGION 2
IF m.adding
 APPEND BLANK
 m.adding = .F.
ENDIF
m.editing = .F.
GATHER MEMVAR MEMO
SHOW GET m.newrecord ENABLE
SHOW GET m.saverecord DISABLE
SHOW GET m.cancel DISABLE
WAIT WINDOW "KAYIT SAKLANDI" NOWAIT
_CUROBJ = OBJNUM(m.cont, 1)
SHOW GET m.cont, 5 ENABLE
SHOW GET m.cont, 6 ENABLE
SHOW GETS
```

```
*       +-------------------------------------------------------+
*       |                                                       |
*       | _QT10WFTRX            m.cancel VALID                  |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:          FINALGIR,     Record Number:  41|
*       | Variable:             m.cancel                        |
*       | Called By:            VALID Clause                    |
*       | Object Type:          Push Button                     |
*       | Snippet Number:       17                              |
*       |                                                       |
*       +-------------------------------------------------------+


FUNCTION _qt10wftrx     &&  m.cancel VALID
#REGION 2
SCATTER MEMVAR MEMO
m.editing = .F.
m.adding = .F.
SHOW GET m.newrecord ENABLE
SHOW GET m.saverecord DISABLE
SHOW GET m.cancel DISABLE
_CUROBJ = OBJNUM(m.cont, 1)
SHOW GET m.cont, 5 ENABLE
SHOW GET m.cont, 6 ENABLE
SHOW GETS

*       +-------------------------------------------------------+
*       |                                                       |
*       | _QT10WFUMK            m.sil VALID                     |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:          FINALGIR,     Record Number:  42|
*       | Variable:             m.sil                           |
*       | Called By:            VALID Clause                    |
*       | Object Type:          Push Button                     |
*       | Snippet Number:       18                              |
*       |                                                       |
*       +-------------------------------------------------------+
*
FUNCTION _qt10wfumk     &&  m.sil VALID
#REGION 2
do noteha.spr
```

```
*        +-------------------------------------------------------+
*        |                                                       |
*        | _QT10WFVCY            m.yz VALID                      |
*        |                                                       |
*        | Function Origin:                                      |
*        |                                                       |
*        | From Screen:          FINALGIR,     Record Number:  43|
*        | Variable:             m.yz                            |
*        | Called By:            VALID Clause                    |
*        | Object Type:          Popup                           |
*        | Snippet Number:       19                              |
*        |                                                       |
*        +-------------------------------------------------------+
*
FUNCTION _qt10wfvcy     &&  m.yz VALID
#REGION 2

DO CASE
   CASE m.yz = 1
 STORE 10 TO m.vz
STORE (100 - m.vz) TO m.fn
show get m.vz
show get m.fn
GO TOP
   CASE m.yz = 2
 STORE 20 TO m.vz
STORE (100 - m.vz) TO m.fn
show get m.vz
show get m.fn
GO TOP
   CASE m.yz = 3
 STORE 30 TO m.vz
STORE (100 - m.vz) TO m.fn
show get m.vz
show get m.fn
 GO TOP
   CASE m.yz = 4
 STORE 40 TO m.vz
STORE (100 - m.vz) TO m.fn
show get m.vz
show get m.fn

 GO TOP
   CASE m.yz = 5
 STORE 50 TO m.vz
STORE (100 - m.vz) TO m.fn
show get m.vz
show get m.fn
 GO TOP
   CASE m.yz = 6
 STORE 60 TO m.vz
STORE (100 - m.vz) TO m.fn
show get m.vz
show get m.fn
 GO TOP
   CASE m.yz = 7
```

```
 STORE 70 TO m.vz
STORE (100 - m.vz) TO m.fn
show get m.vz
show get m.fn
 GO TOP
ENDCASE
SHOW GETS

*       +------------------------------------------------------+
*       ¦                                                      ¦
*       ¦ _QT10WFWYC           Read Level Deactivate           ¦
*       ¦                                                      ¦
*       ¦ Function Origin:                                     ¦
*       ¦                                                      ¦
*       ¦ From Screen:         Multiple Screens                ¦
*       ¦ Called By:           READ Statement                  ¦
*       ¦ Snippet Number:      20                              ¦
*       ¦                                                      ¦
*       +------------------------------------------------------+
FUNCTION _qt10wfwyc     && Read Level Deactivate
*
* Deactivate Code from screen: FINALGIR
*
#REGION 2
IF m.editing
 ?? CHR(7)
 WAIT WINDOW "LÜTFFEN DEĞİŞİKLİKLERİ ONAYLAYINIZ" NOWAIT
 RETURN .F.
ENDIF
RETURN NOT WREAD()

*       +------------------------------------------------------+
*       ¦                                                      ¦
*       ¦ _QT10WFKA2           Read Level Show                 ¦
*       ¦                                                      ¦
*       ¦ Function Origin:                                     ¦
*       ¦                                                      ¦
*       ¦ From Screen:         Multiple Screens                ¦
*       ¦ Called By:           READ Statement                  ¦
*       ¦ Snippet Number:      21                              ¦
*       ¦                                                      ¦
*       +------------------------------------------------------+
FUNCTION _qt10wfka2     && Read Level Show
PRIVATE currwind
STORE WOUTPUT() TO currwind

*
* Show Code from screen: FCONT
*
#REGION 1
IF EOF()
 GO BOTTOM
ENDIF
m.saverecno = RECNO()
GO TOP
m.toprec = RECNO()
```

```
GO BOTTOM
m.bottomrec = RECNO()
GO m.saverecno
IF RECNO() = m.bottomrec
 SHOW GET m.cont, 1 DISABLE
 SHOW GET m.cont, 2 ENABLE
 SHOW GET m.cont, 3 ENABLE
 SHOW GET m.cont, 4 DISABLE
ELSE
 IF RECNO() = m.toprec
  SHOW GET m.cont, 1 ENABLE
  SHOW GET m.cont, 2 DISABLE
  SHOW GET m.cont, 3 DISABLE
  SHOW GET m.cont, 4 ENABLE
 ELSE
  SHOW GET m.cont ENABLE
 ENDIF
ENDIF

*
* Show Code from screen: FINALGIR
*
#REGION 2
IF NOT m.adding
 SCATTER MEMVAR MEMO
ENDIF
IF NOT EMPTY(currwind)
 ACTIVATE WINDOW (currwind) SAME
ENDIF
```

```
*        +-------------------------------------------------------+
*        |                                                       |
*        | 12/09/94              DEFTERI.SPR               10:47:41 |
*        |                                                       |
*        |-------------------------------------------------------|
*        |                                                       |
*        | Author's Name                                         |
*        |                                                       |
*        | Copyright (c) 1994 Company Name                       |
*        | Address                                               |
*        | City,     Zip                                         |
*        |                                                       |
*        | Description:                                          |
*        | This program was automatically generated by GENSCRN.  |
*        |                                                       |
*        +-------------------------------------------------------+
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
 SET TALK OFF
 m.talkstat = "ON"
ELSE
 m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
m.currarea = SELECT()
*        +-------------------------------------------------------+
*        |                                                       |
*        |                    Window definitions                 |
*        |                                                       |
*        +-------------------------------------------------------+
*
IF NOT WEXIST("controls")
 DEFINE WINDOW controls ;
  FROM 20, 3 ;
  TO 22,76 ;
  NOFLOAT ;
  NOCLOSE ;
  COLOR SCHEME 22
ENDIF
IF NOT WEXIST("diplom")
 DEFINE WINDOW diplom ;
  FROM 1, 5 ;
  TO 18,73 ;
  TITLE " DIPLOMA DEFTERİ" ;
  FLOAT ;
  NOCLOSE ;
  SHADOW ;
  DOUBLE ;
  COLOR SCHEME 5
ENDIF
*        +-------------------------------------------------------+
*        |                                                       |
*        |              ECONT Setup Code - SECTION 2             |
*        |                                                       |
*        +-------------------------------------------------------+
*
#REGION 1
REGIONAL m.cont, m.toprec, m.bottomrec, m.saverecno, m.quitting
```

```
m.quitting = .F.
m.cont = "Tamam"
IF EOF()
 GO BOTTOM
ENDIF
m.saverecno = RECNO()
GO TOP
m.toprec = RECNO()
GO BOTTOM
m.bottomrec = RECNO()
GO m.saverecno

*       +--------------------------------------------------------+
*       ¦                                                        ¦
*       ¦              DEFTERI Setup Code - SECTION 2            ¦
*       ¦                                                        ¦
*       +--------------------------------------------------------+
#REGION 2
set echo off
set escape off
set deleted on
set exact on
set date british
close databases
IF USED("ogrenci")
 SELECT ogrenci
 SET ORDER TO 1
ELSE
 SELECT 0
 USE (LOCFILE("\yildiz\ydbf\ogrenci.dbf","DBF","OGRENCI.DBF Nerede ?"));
  AGAIN ALIAS ogrenci ;
  ORDER 1
ENDIF
IF USED("dersler")
 SELECT dersler
 SET ORDER TO 1
ELSE
 SELECT 0
 USE (LOCFILE("\yildiz\ydbf\dersler.dbf","DBF","DERSLER.DBF Nerede ?"));
  AGAIN ALIAS dersler ;
  ORDER 1
ENDIF
public dd,m.dersadi
PRIVATE m.adding, m.editing
m.adding = .F.
m.editing = .F.
SCATTER MEMVAR MEMO
*       +--------------------------------------------------------+
*       ¦                                                        ¦
*       ¦                    ECONT Screen Layout                 ¦
*       ¦                                                        ¦
*       +--------------------------------------------------------+
*
#REGION 1
IF WVISIBLE("controls")
 ACTIVATE WINDOW controls SAME
ELSE
 ACTIVATE WINDOW controls NOSHOW
ENDIF
@ 0,5 GET m.cont ;
```

```
PICTURE "@*HN so\<Nraki;ön\<Ceki;bi\<Rinci;s\<Onuncu;\<Ara;\?\!\<Tamam" ;
 SIZE 1,9,2 ;
 DEFAULT 1 ;
 VALID _qtd0n59j2()
*       +------------------------------------------------------------+
*       ¦                                                            ¦
*       ¦                    DEFTERI Screen Layout                   ¦
*       ¦                                                            ¦
*       +------------------------------------------------------------+
*
#REGION 2
IF WVISIBLE("diplom")
 ACTIVATE WINDOW diplom SAME
ELSE
 ACTIVATE WINDOW diplom NOSHOW
ENDIF
@ 3,1 SAY "FAKÜLTE NUMARA:"
@ 4,1 SAY "ADI VE SOYADI :"
@ 5,1 SAY "DO*UM YER/TAR :"
@ 6,1 SAY "BABA ADI......:"
@ 11,3 SAY "DIPLOMA PROJESİNİN KONUSU:"
@ 10,3 SAY "ARA PROJESI KONUSU.......:"
@ 1,0 TO 8,41
@ 9,2 TO 12,64
@ 1,42 TO 8,66
@ 13,2 TO 15,64 ;
 COLOR GR+/W
@ 2,1 SAY "DIPLOMA NO.VE TARİH.:"
@ 7,1 SAY "DERS ADI......:"
@ 2,23 GET m.dipno ;
 SIZE 1,7 ;
 DEFAULT " " ;
 VALID _qtd0n5ez9()
@ 2,33 GET m.diptarih ;
 SIZE 1,8 ;
 DEFAULT {  /  /  } ;
 PICTURE "@E" ;
 VALID _qtd0n5g0o()
@ 3,16 GET m.fk_no ;
 SIZE 1,7 ;
 DEFAULT " " ;
 WHEN _qtd0n5g8r() ;
 VALID _qtd0n5geg()
@ 4,16 GET m.adi ;
 SIZE 1,13 ;
 DEFAULT " " ;
 VALID _qtd0n5gl6()
@ 4,29 GET m.soyadi ;
 SIZE 1,12 ;
 DEFAULT " " ;
 VALID _qtd0n5hva()
@ 5,16 GET m.dyer ;
 SIZE 1,15 ;
 DEFAULT " " ;
 VALID _qtd0n5jci()
@ 5,33 GET m.dtarih ;
 SIZE 1,8 ;
 DEFAULT {  /  /  } ;
 PICTURE "@E" ;
 VALID _qtd0n5kft()
```

```
@ 6,16 GET m.babadi ;
 SIZE 1,25 ;
 DEFAULT " " ;
 VALID _qtd0n5lfs()
@ 7,16 GET m.dersadi ;
 SIZE 1,25 ;
 DEFAULT " " ;
 WHEN _qtd0n5lxy() ;
 VALID _qtd0n5m3u()
@ 2,58 GET m.dk ;
 SIZE 1,2 ;
 DEFAULT " " ;
 VALID _qtd0n5mak()
@ 3,58 GET m.kd ;
 SIZE 1,2 ;
 DEFAULT 0 ;
 PICTURE "@Z" ;
 VALID _qtd0n5mh3()
@ 4,58 GET m.notu ;
 SIZE 1,3 ;
 DEFAULT 0 ;
 PICTURE "@Z" ;
 VALID _qtd0n5mqz()
@ 5,58 GET m.kxn ;
 SIZE 1,4 ;
 DEFAULT 0 ;
 PICTURE "@Z" ;
 VALID _qtd0n5myt()
@ 6,58 GET m.ugiris_t ;
 SIZE 1,8 ;

 DEFAULT {  /  /  } ;
 PICTURE "@E" ;
 VALID _qtd0n5nl7()
@ 7,58 GET m.ub_ogy ;
 SIZE 1,8 ;
 DEFAULT " " ;
 VALID _qtd0n5o55()
@ 10,29 GET m.araprojadi ;
 SIZE 1,35 ;
 DEFAULT " " ;
 VALID _qtd0n5obj()
@ 11,29 GET m.diprojeadi ;
 SIZE 1,35 ;
 DEFAULT " " ;
 VALID _qtd0n5opi()
@ 14,19 GET m.newrecord ;
 PICTURE "@*HN \<Yeni" ;
 SIZE 1,8,1 ;
 DEFAULT 1 ;
 VALID _qtd0n5pk0() ;
 COLOR SCHEME 12
@ 14,31 GET m.saverecord ;
 PICTURE "@*HN \<Sakla" ;
 SIZE 1,9,1 ;
 DEFAULT 1 ;
 WHEN _qtd0n5qsf() ;
 VALID _qtd0n5rf6() ;
 DISABLE ;
 COLOR SCHEME 12
```

```
@ 14,43 GET m.cancel ;
 PICTURE "@*HN i\<Ptal" ;
 SIZE 1,9,1 ;
 DEFAULT 1 ;
 VALID _qtd0n5s0o() ;
 DISABLE ;
 COLOR SCHEME 12
@ 14,54 GET m.sil ;
 PICTURE "@*HN si\<L" ;
 SIZE 1,7,1 ;
 DEFAULT 1 ;
 VALID _qtd0n5sbc() ;
 COLOR SCHEME 12
@ 14,6 GET m.yardim ;
 PICTURE "@*HN yar\<Dim" ;
 SIZE 1,10,1 ;
 DEFAULT 1 ;
 VALID _qtd0n5shu() ;
 COLOR SCHEME 12
@ 5,31 SAY " "
@ 7,43 SAY "ÜNİ.BİTİR YIL :"
@ 6,43 SAY "ÜNİ.GİRİŞ TAR :"
@ 5,43 SAY "KR.X NOT......:"
@ 4,43 SAY "NOTU.........:"
@ 3,43 SAY "DERS KREDISI..:"
@ 2,43 SAY "DERS KODU ....:"
IF NOT WVISIBLE("diplom")
 ACTIVATE WINDOW diplom
ENDIF
IF NOT WVISIBLE("controls")
 ACTIVATE WINDOW controls
ENDIF
READ CYCLE ;
 DEACTIVATE _qtd0n5tuw() ;
 SHOW _qtd0n5br2()
RELEASE WINDOW controls
RELEASE WINDOW diplom
SELECT (m.currarea)
#REGION 0
IF m.talkstat = "ON"
 SET TALK ON
ENDIF
IF m.compstat = "ON"
 SET COMPATIBLE ON
ENDIF
*       +--------------------------------------------------------+
*       ¦                                                        ¦
*       ¦                  DEFTERI Cleanup Code                  ¦
*       ¦                                                        ¦
*       +--------------------------------------------------------+
*
#REGION 2
IF m.quitting
 RELEASE WINDOW diplom
 RELEASE WINDOW controls
ENDIF
* SHOWSAVE
PROCEDURE showsave
IF NOT m.editing
 WAIT WINDOW "LÜTFFEN DEĞİPİKLİKLERİ ONAYLAYINIZ" NOWAIT
```

```
ENDIF
SHOW GET m.newrecord DISABLE
SHOW GET m.saverecord ENABLE
SHOW GET m.cancel ENABLE
SHOW GETS ONLY DISABLE WINDOW controls
m.editing = .T.
*        +------------------------------------------------------------+
*        |                                                            |
*        | _QTD0N59J2            m.cont VALID                         |
*        |                                                            |
*        | Function Origin:                                           |
*        |                                                            |
*        | From Screen:          ECONT,     Record Number:    2       |
*        | Variable:             m.cont                               |
*        | Called By:            VALID Clause                         |
*        | Object Type:          Push Button                          |
*        | Snippet Number:       1                                    |
*        |                                                            |
*        +------------------------------------------------------------+
*
FUNCTION _qtd0n59j2     && m.cont VALID
#REGION 1
DO CASE
CASE m.cont = "soNraki"
 SKIP 1
 IF RECNO() = m.bottomrec
  SHOW GET m.cont, 1 DISABLE
  SHOW GET m.cont, 4 DISABLE
 ELSE
  IF RECNO() > m.toprec
   SHOW GET m.cont, 2 ENABLE
   SHOW GET m.cont, 3 ENABLE
  ENDIF
 ENDIF
CASE m.cont = "önCeki"
 SKIP -1
 IF RECNO() = m.toprec
  SHOW GET m.cont, 2 DISABLE
  SHOW GET m.cont, 3 DISABLE
 ELSE
  IF RECNO() < m.bottomrec
   SHOW GET m.cont, 1 ENABLE
   SHOW GET m.cont, 4 ENABLE
  ENDIF
 ENDIF
CASE m.cont = "biRinci"
 GO TOP
 SHOW GET m.cont, 1 ENABLE
 SHOW GET m.cont, 2 DISABLE
 SHOW GET m.cont, 3 DISABLE
 SHOW GET m.cont, 4 ENABLE
CASE m.cont = "sOnuncu"
 GO BOTTOM
 SHOW GET m.cont, 1 DISABLE
 SHOW GET m.cont, 2 ENABLE
 SHOW GET m.cont, 3 ENABLE
 SHOW GET m.cont, 4 DISABLE
CASE m.cont = "Ara"
     do ebrowser.spr
```

```
CASE m.cont = "Tamam"
 m.idlequit = .T.
 m.quitting = .T.
 CLEAR READ
ENDCASE
SHOW GETS
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QTD0N5EZ9           m.dipno VALID                    ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:         DEFTERI,     Record Number:  14  ¦
*       ¦ Variable:            m.dipno                          ¦
*       ¦ Called By:           VALID Clause                     ¦
*       ¦ Object Type:         Field                            ¦
*       ¦ Snippet Number:      2                                ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
FUNCTION _qtd0n5ez9     && m.dipno VALID
#REGION 2
=showsave()
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QTD0N5G0O           m.diptarih VALID                 ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:         DEFTERI,     Record Number:  15  ¦
*       ¦ Variable:            m.diptarih                       ¦
*       ¦ Called By:           VALID Clause                     ¦
*       ¦ Object Type:         Field                            ¦
*       ¦ Snippet Number:      3                                ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
FUNCTION _qtd0n5g0o     && m.diptarih VALID
#REGION 2
=showsave()
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QTD0N5G8R           m.fk_no WHEN                     ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:         DEFTERI,     Record Number:  16  ¦
*       ¦ Variable:            m.fk_no                          ¦
*       ¦ Called By:           WHEN Clause                      ¦
*       ¦ Object Type:         Field                            ¦
*       ¦ Snippet Number:      4                                ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
FUNCTION _qtd0n5g8r     && m.fk_no WHEN

#REGION 2
on key label ? do okulist.spr
```

```
*        +-------------------------------------------------------+
*        ¦                                                       ¦
*        ¦ _QTD0N5GEG           m.fk_no VALID                    ¦
*        ¦                                                       ¦
*        ¦ Function Origin:                                      ¦
*        ¦                                                       ¦
*        ¦ From Screen:         DEFTERI,     Record Number:  16  ¦
*        ¦ Variable:            m.fk_no                          ¦
*        ¦ Called By:           VALID Clause                     ¦
*        ¦ Object Type:         Field                            ¦
*        ¦ Snippet Number:      5                                ¦
*        ¦                                                       ¦
*        +-------------------------------------------------------+
*
FUNCTION _qtd0n5geg     &&  m.fk_no VALID
#REGION 2
on key label ?
=showsave()
*        +-------------------------------------------------------+
*        ¦                                                       ¦
*        ¦ _QTD0N5GL6           m.adi VALID                      ¦
*        ¦                                                       ¦
*        ¦ Function Origin:                                      ¦
*        ¦                                                       ¦
*        ¦ From Screen:         DEFTERI,     Record Number:  17  ¦
*        ¦ Variable:            m.adi                            ¦
*        ¦ Called By:           VALID Clause                     ¦
*        ¦ Object Type:         Field                            ¦
*        ¦ Snippet Number:      6                                ¦
*        ¦                                                       ¦
*        +-------------------------------------------------------+
*
FUNCTION _qtd0n5g16     &&  m.adi VALID
#REGION 2
if not empty(m.adi)
      gecicigoz =proper(alltrim(m.adi))
      gercekgoz =""
      uzunlugu  =len(gecicigoz)
      flag      =.t.
      for i=1 to uzunlugu
           if empty(substr(gecicigoz,i,1))
              if flag
                 gercekgoz=padr(gercekgoz,len(gercekgoz)+1," ")
                 flag=.f.
              endif
           else
gercekgoz=padr(gercekgoz,len(gercekgoz)+1,substr(gecicigoz,i,1))

              flag=.t.
           endif
       endfor
 gercekgoz=padr(gercekgoz,len(dersler.adi))
       m.adi=gercekgoz
       show get m.adi
endif
=showsave()
```

```
*       +-------------------------------------------------------+
*       |                                                       |
*       | _QTD0N5HVA           m.soyadi VALID                   |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:         DEFTERI,     Record Number:  18  |
*       | Variable:            m.soyadi                         |
*       | Called By:           VALID Clause                     |
*       | Object Type:         Field                            |
*       | Snippet Number:      7                                |
*       |                                                       |
*       +-------------------------------------------------------+
*
FUNCTION _qtd0n5hva     &&  m.soyadi VALID
#REGION 2
if not empty(m.soyadi)
      gecicigoz =proper(alltrim(m.soyadi))
      gercekgoz =""
      uzunlugu  =len(gecicigoz)
      flag      =.t.
      for i=1 to uzunlugu
          if empty(substr(gecicigoz,i,1))
             if flag
                gercekgoz=padr(gercekgoz,len(gercekgoz)+1," ")
                flag=.f.
             endif
          else
gercekgoz=padr(gercekgoz,len(gercekgoz)+1,substr(gecicigoz,i,1))
                flag=.t.
          endif
       endfor
       gercekgoz=padr(gercekgoz,len(dersler.soyadi))
       m.soyadi=gercekgoz
       show get m.soyadi
endif
=showsave()
*       +-------------------------------------------------------+
*       |                                                       |
*       | _QTD0N5JCI           m.dyer VALID                     |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:         DEFTERI,     Record Number:  19  |
*       | Variable:            m.dyer                           |
*       | Called By:           VALID Clause                     |
*       | Object Type:         Field                            |
*       | Snippet Number:      8                                |
*       |                                                       |
*       +-------------------------------------------------------+
FUNCTION _qtd0n5jci     &&  m.dyer VALID
#REGION 2
if not empty(m.dyer)
      gecicigoz =proper(alltrim(m.dyer))
      gercekgoz =""

      uzunlugu  =len(gecicigoz)
      flag      =.t.
      for i=1 to uzunlugu
```

```
            if empty(substr(gecicigoz,i,1))
               if flag
                  gercekgoz=padr(gercekgoz,len(gercekgoz)+1," ")
                  flag=.f.
               endif
            else
gercekgoz=padr(gercekgoz,len(gercekgoz)+1,substr(gecicigoz,i,1))
               flag=.t.
            endif
       endfor
       gercekgoz=padr(gercekgoz,len(dersler.dyer))
       m.dyer=gercekgoz
       show get m.dyer
endif
=showsave()
*        +------------------------------------------------------+
*        ¦                                                      ¦
*        ¦ _QTD0N5KFT           m.dtarih VALID                  ¦
*        ¦                                                      ¦
*        ¦ Function Origin:                                     ¦
*        ¦                                                      ¦
*        ¦ From Screen:         DEFTERI,     Record Number:  20 ¦
*        ¦ Variable:            m.dtarih                        ¦
*        ¦ Called By:           VALID Clause                    ¦
*        ¦ Object Type:         Field                           ¦
*        ¦ Snippet Number:      9                               ¦
*        ¦                                                      ¦
*        +------------------------------------------------------+
FUNCTION _qtd0n5kft     &&  m.dtarih VALID
#REGION 2
=showsave()
*        +------------------------------------------------------+
*        ¦                                                      ¦
*        ¦ _QTD0N5LFS           m.babadi VALID                  ¦
*        ¦                                                      ¦
*        ¦ Function Origin:                                     ¦
*        ¦                                                      ¦
*        ¦ From Screen:         DEFTERI,     Record Number:  21 ¦
*        ¦ Variable:            m.babadi                        ¦
*        ¦ Called By:           VALID Clause                    ¦
*        ¦ Object Type:         Field                           ¦
*        ¦ Snippet Number:      10                              ¦
*        ¦                                                      ¦
*        +------------------------------------------------------+
*
FUNCTION _qtd0n5lfs     &&  m.babadi VALID
#REGION 2
if not empty(m.babadi)
      gecicigoz =proper(alltrim(m.babadi))
      gercekgoz =""
      uzunlugu  =len(gecicigoz)
      flag      =.t.
      for i=1 to uzunlugu
           if empty(substr(gecicigoz,i,1))
              if flag
                 gercekgoz=padr(gercekgoz,len(gercekgoz)+1," ")
                 flag=.f.
              endif
           else
gercekgoz=padr(gercekgoz,len(gercekgoz)+1,substr(gecicigoz,i,1))
```

```
            flag=.t.
          endif
       endfor
       gercekgoz=padr(gercekgoz,len(dersler.babadi))
       m.babadi=gercekgoz
       show get m.babadi
endif
=showsave()

*       +-------------------------------------------------------+
*       |                                                       |
*       | _QTD0N5LXY           m.dersadi WHEN                   |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:         DEFTERI,     Record Number:  22  |
*       | Variable:            m.dersadi                        |
*       | Called By:           WHEN Clause                      |
*       | Object Type:         Field                            |
*       | Snippet Number:      11                               |
*       |                                                       |
*       +-------------------------------------------------------+
*
FUNCTION _qtd0n51xy     &&  m.dersadi WHEN
#REGION 2
on key label ? do derslist.spr
dd =1

*       +-------------------------------------------------------+
*       |                                                       |
*       | _QTD0N5M3U           m.dersadi VALID                  |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:         DEFTERI,     Record Number:  22  |
*       | Variable:            m.dersadi                        |
*       | Called By:           VALID Clause                     |
*       | Object Type:         Field                            |
*       | Snippet Number:      12                               |
*       |                                                       |
*       +-------------------------------------------------------+
*
FUNCTION _qtd0n5m3u     &&  m.dersadi VALID
#REGION 2
on key label ?
=showsave()
*       +-------------------------------------------------------+
*       |                                                       |
*       | _QTD0N5MAK           m.dk VALID                       |
*       |                                                       |
*       | Function Origin:                                      |
*       |                                                       |
*       | From Screen:         DEFTERI,     Record Number:  23  |
*       | Variable:            m.dk                             |
*       | Called By:           VALID Clause                     |
*       | Object Type:         Field                            |
*       | Snippet Number:      13                               |
*       |                                                       |
*       +-------------------------------------------------------+
FUNCTION _qtd0n5mak     &&  m.dk VALID
```

```
#REGION 2
=showsave()
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QTD0N5MH3            m.kd VALID                      ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:          DEFTERI,     Record Number:  24 ¦
*       ¦ Variable:             m.kd                            ¦
*       ¦ Called By:            VALID Clause                    ¦
*       ¦ Object Type:          Field                           ¦
*       ¦ Snippet Number:       14                              ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+

FUNCTION _qtd0n5mh3     && m.kd VALID
#REGION 2
m.topkd = m.topkd + m.kd
SHOW GETS
=showsave()
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QTD0N5MQZ            m.notu VALID                    ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:          DEFTERI,     Record Number:  25 ¦
*       ¦ Variable:             m.notu                          ¦
*       ¦ Called By:            VALID Clause                    ¦
*       ¦ Object Type:          Field                           ¦
*       ¦ Snippet Number:       15                              ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
*
FUNCTION _qtd0n5mqz     && m.notu VALID
#REGION 2
IF m.notu > 0
m.kxn = m.kd * m.notu
m.topkxn = m.topkxn + m.kxn
ENDIF
=showsave()
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QTD0N5MYT            m.kxn VALID                     ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:          DEFTERI,     Record Number:  26 ¦
*       ¦ Variable:             m.kxn                           ¦
*       ¦ Called By:            VALID Clause                    ¦
*       ¦ Object Type:          Field                           ¦
*       ¦ Snippet Number:       16                              ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+

FUNCTION _qtd0n5myt     && m.kxn VALID
#REGION 2
=showsave()
```

```
*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QTD0N5NL7           m.ugiris_t VALID                 ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:         DEFTERI,     Record Number:  27  ¦
*       ¦ Variable:            m.ugiris_t                       ¦
*       ¦ Called By:           VALID Clause                     ¦
*       ¦ Object Type:         Field                            ¦
*       ¦ Snippet Number:      17                               ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
*
FUNCTION _qtd0n5nl7     &&  m.ugiris_t VALID
#REGION 2
=showsave()

*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QTD0N5O55           m.ub_ogy VALID                   ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:         DEFTERI,     Record Number:  28  ¦
*       ¦ Variable:            m.ub_ogy                         ¦
*       ¦ Called By:           VALID Clause                     ¦
*       ¦ Object Type:         Field                            ¦
*       ¦ Snippet Number:      18                               ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
*
FUNCTION _qtd0n5o55     &&  m.ub_ogy VALID
#REGION 2
=showsave()

*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QTD0N5OBJ           m.araprojadi VALID               ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:         DEFTERI,     Record Number:  29  ¦
*       ¦ Variable:            m.araprojadi                     ¦
*       ¦ Called By:           VALID Clause                     ¦
*       ¦ Object Type:         Field                            ¦
*       ¦ Snippet Number:      19                               ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
*
FUNCTION _qtd0n5obj     &&  m.araprojadi VALID
#REGION 2
if not empty(m.araprojadi)
      gecicigoz =proper(alltrim(m.araprojadi))

      gercekgoz =""
      uzunlugu  =len(gecicigoz)
      flag      =.t.
      for i=1 to uzunlugu
```

```
            if empty(substr(gecicigoz,i,1))
               if flag
                  gercekgoz=padr(gercekgoz,len(gercekgoz)+1," ")
                  flag=.f.
               endif
            else
gercekgoz=padr(gercekgoz,len(gercekgoz)+1,substr(gecicigoz,i,1))
               flag=.t.
            endif
         endfor
         gercekgoz=padr(gercekgoz,len(dersler.araprojadi))
         m.araprojadi=gercekgoz
         show get m.araprojadi
endif
=showsave()

*        +------------------------------------------------------+
*        ¦                                                      ¦
*        ¦ _QTD0N5OPI           m.diprojeadi VALID              ¦
*        ¦                                                      ¦
*        ¦ Function Origin:                                     ¦
*        ¦                                                      ¦
*        ¦ From Screen:         DEFTERI,     Record Number:  30 ¦
*        ¦ Variable:            m.diprojeadi                    ¦
*        ¦ Called By:           VALID Clause                    ¦
*        ¦ Object Type:         Field                           ¦
*        ¦ Snippet Number:      20                              ¦
*        ¦                                                      ¦
*        +------------------------------------------------------+
*
FUNCTION _qtd0n5opi     &&  m.diprojeadi VALID
#REGION 2
if not empty(m.diprojeadi)
      gecicigoz =proper(alltrim(m.diprojeadi))
      gercekgoz =""

      uzunlugu  =len(gecicigoz)
      flag      =.t.
      for i=1 to uzunlugu
           if empty(substr(gecicigoz,i,1))
              if flag
                 gercekgoz=padr(gercekgoz,len(gercekgoz)+1," ")
                 flag=.f.
              endif
           else
gercekgoz=padr(gercekgoz,len(gercekgoz)+1,substr(gecicigoz,i,1))
              flag=.t.
           endif
       endfor

       gercekgoz=padr(gercekgoz,len(dersler.diprojeadi))
       m.diprojeadi=gercekgoz
       show get m.diprojeadi
endif
=showsave()
*        +------------------------------------------------------+
*        ¦                                                      ¦
*        ¦ _QTD0N5PK0           m.newrecord VALID               ¦
*        ¦                                                      ¦
*        ¦ Function Origin:                                     ¦
```

```
*       ¦                                                        ¦
*       ¦ From Screen:          DEFTERI,     Record Number:   31 ¦
*       ¦ Variable:             m.newrecord                      ¦
*       ¦ Called By:            VALID Clause                     ¦
*       ¦ Object Type:          Push Button                      ¦
*       ¦ Snippet Number:       21                               ¦
*       ¦                                                        ¦
*       +--------------------------------------------------------+
*
FUNCTION _qtd0n5pk0     && m.newrecord VALID
#REGION 2
m.adding = .T.
SCATTER MEMVAR BLANK MEMO
SHOW GET m.newrecord DISABLE
SHOW GET m.saverecord ENABLE
SHOW GET m.cancel ENABLE
_CUROBJ = OBJNUM(m.dipno)
SHOW GETS
SHOW GETS DISABLE ONLY WINDOW controls

*       +--------------------------------------------------------+
*       ¦                                                        ¦
*       ¦ _QTD0N5QSF            m.saverecord WHEN                ¦
*       ¦                                                        ¦
*       ¦ Function Origin:                                       ¦
*       ¦                                                        ¦
*       ¦ From Screen:          DEFTERI,     Record Number:   32 ¦
*       ¦ Variable:             m.saverecord                     ¦
*       ¦ Called By:            WHEN Clause                      ¦
*       ¦ Object Type:          Push Button                      ¦
*       ¦ Snippet Number:       22                               ¦
*       ¦                                                        ¦
*       +--------------------------------------------------------+
*
FUNCTION _qtd0n5qsf     && m.saverecord WHEN
#REGION 2
IF EMPTY(m.FK_NO)
 ?? CHR(7)
 WAIT WINDOW "Enter DIPLOMA NO. VE TARIH" NOWAIT
 _CUROBJ = OBJNUM(m.dip_no_tar)
 RETURN .F.
ENDIF
*       +--------------------------------------------------------+
*       ¦                                                        ¦
*       ¦ _QTD0N5RF6            m.saverecord VALID               ¦
*       ¦                                                        ¦
*       ¦ Function Origin:                                       ¦
*       ¦                                                        ¦
*       ¦ From Screen:          DEFTERI,     Record Number:   32 ¦
*       ¦ Variable:             m.saverecord                     ¦
*       ¦ Called By:            VALID Clause                     ¦
*       ¦ Object Type:          Push Button                      ¦
*       ¦ Snippet Number:       23                               ¦
*       ¦                                                        ¦
*       +--------------------------------------------------------+
*
FUNCTION _qtd0n5rf6     && m.saverecord VALID
#REGION 2
IF m.adding
 APPEND BLANK
```

```
 m.adding = .F.
ENDIF
m.editing = .F.
GATHER MEMVAR MEMO
SHOW GET m.newrecord ENABLE
SHOW GET m.saverecord DISABLE
SHOW GET m.cancel DISABLE
WAIT WINDOW "KAYIT SAKLANDI" NOWAIT
_CUROBJ = OBJNUM(m.cont, 1)
SHOW GET m.cont, 5 ENABLE
SHOW GET m.cont, 6 ENABLE
SHOW GETS

*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QTD0N5S0O           m.cancel VALID                   ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:         DEFTERI,     Record Number:  33  ¦
*       ¦ Variable:            m.cancel                         ¦
*       ¦ Called By:           VALID Clause                     ¦
*       ¦ Object Type:         Push Button                      ¦
*       ¦ Snippet Number:      24                               ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
*
FUNCTION _qtd0n5s0o     && m.cancel VALID
#REGION 2
SCATTER MEMVAR MEMO
m.editing = .F.
m.adding = .F.
SHOW GET m.newrecord ENABLE
SHOW GET m.saverecord DISABLE
SHOW GET m.cancel DISABLE
_CUROBJ = OBJNUM(m.cont, 1)
SHOW GET m.cont, 5 ENABLE
SHOW GET m.cont, 6 ENABLE
SHOW GETS

*       +-------------------------------------------------------+
*       ¦                                                       ¦
*       ¦ _QTD0N5SBC           m.sil VALID                      ¦
*       ¦                                                       ¦
*       ¦ Function Origin:                                      ¦
*       ¦                                                       ¦
*       ¦ From Screen:         DEFTERI,     Record Number:  34  ¦
*       ¦ Variable:            m.sil                            ¦
*       ¦ Called By:           VALID Clause                     ¦
*       ¦ Object Type:         Push Button                      ¦
*       ¦ Snippet Number:      25                               ¦
*       ¦                                                       ¦
*       +-------------------------------------------------------+
*
FUNCTION _qtd0n5sbc     && m.sil VALID
#REGION 2
DO DEHAYIR.SPR
```

```
*        +-------------------------------------------------------+
*        ¦                                                       ¦
*        ¦ _QTD0N5SHU            m.yardim VALID                  ¦
*        ¦                                                       ¦
*        ¦ Function Origin:                                      ¦
*        ¦                                                       ¦
*        ¦ From Screen:          DEFTERI,     Record Number:  35 ¦
*        ¦ Variable:             m.yardim                        ¦
*        ¦ Called By:            VALID Clause                    ¦
*        ¦ Object Type:          Push Button                     ¦
*        ¦ Snippet Number:       26                              ¦
*        ¦                                                       ¦
*        +-------------------------------------------------------+
*
FUNCTION _qtd0n5shu     &&  m.yardim VALID
#REGION 2
DO DEFYADIM.SPR
*        +-------------------------------------------------------+
*        ¦                                                       ¦
*        ¦ _QTD0N5TUW            Read Level Deactivate           ¦
*        ¦                                                       ¦
*        ¦ Function Origin:                                      ¦
*        ¦                                                       ¦
*        ¦ From Screen:          Multiple Screens                ¦
*        ¦ Called By:            READ Statement                  ¦
*        ¦ Snippet Number:       27                              ¦
*        ¦                                                       ¦
*        +-------------------------------------------------------+
*
FUNCTION _qtd0n5tuw     && Read Level Deactivate
*
* Deactivate Code from screen: DEFTERI
*
#REGION 2
IF m.editing
 ?? CHR(7)
 WAIT WINDOW "LÜTFFEN DE*ÿPÿKLÿKLERÿ ONAYLAYINIZ" NOWAIT
 RETURN .F.
ENDIF
RETURN NOT WREAD()

*        +-------------------------------------------------------+
*        ¦                                                       ¦
*        ¦ _QTD0N5BR2            Read Level Show                 ¦
*        ¦                                                       ¦
*        ¦ Function Origin:                                      ¦
*        ¦                                                       ¦
*        ¦ From Screen:          Multiple Screens                ¦
*        ¦ Called By:            READ Statement                  ¦
*        ¦ Snippet Number:       28                              ¦
*        ¦                                                       ¦
*        +-------------------------------------------------------+
*
FUNCTION _qtd0n5br2     && Read Level Show
PRIVATE currwind
STORE WOUTPUT() TO currwind
* Show Code from screen: ECONT
#REGION 1
IF EOF()
```

```
 GO BOTTOM
ENDIF
m.saverecno = RECNO()
GO TOP
m.toprec = RECNO()
GO BOTTOM
m.bottomrec = RECNO()
GO m.saverecno
IF RECNO() = m.bottomrec
 SHOW GET m.cont, 1 DISABLE
 SHOW GET m.cont, 2 ENABLE
 SHOW GET m.cont, 3 ENABLE
 SHOW GET m.cont, 4 DISABLE
ELSE
 IF RECNO() = m.toprec
  SHOW GET m.cont, 1 ENABLE
  SHOW GET m.cont, 2 DISABLE
  SHOW GET m.cont, 3 DISABLE
  SHOW GET m.cont, 4 ENABLE
 ELSE
  SHOW GET m.cont ENABLE
 ENDIF
ENDIF
* Show Code from screen: DEFTERI
#REGION 2
IF NOT m.adding
 SCATTER MEMVAR MEMO
ENDIF
IF NOT EMPTY(currwind)
 ACTIVATE WINDOW (currwind) SAME
ENDIF
```