YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

LİNUX İŞLETİM SİSTEMİ VE İNTERAKTİF DERS OTOMASYONU

Matematik Müh. Muharrem ÜNAL

FBE Matematik Mühendisliği Anabilim Dalında Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı :Yrd.Doç.Dr. İbrahim EMİROĞLU (YTÜ)

Y.Doç.Dr. Coşkun GÜLER

Doç.Dr. Ömer Gök

İSTANBUL,2003

İÇİNDEKİLER

Sayfa

KISALTMA LİSTESİ

BDE	Bilgisayar Destekli Eğitim
BTE	Bilgisayar Tabanlı Eğitim
CD-ROM	Compact Disc-Read Only Memory
CGI	Common Gateway Interface
FAQ	Frequently Asked Questions
FTP	File Transfer Protocol
HTML	Hyper Text Markup Language
SGML	Standard Genarilzed Markup Language
VRML	Virtual Reality Modelling Language
XML	Extendble Markup Language
IBM	International Business Machine
ISP	Internet Service Provider
ISS	Internet Service Server
PC	Personal Computer
PHP	Personel Home Page
ASP	Active Server Page
CFML	Cold Fusion
JSP	Java Server Page
PERL	Practical Extraction and Report Language
PPP	Point to Point Protocol
SQL	Structured Query Language
ODBC	Open-Database-Connectivity
TCP/IP	Transmission Control Protocol/ Internet Protocol
UUCP	Unix to Unix Copy Program
WTE	Web Tabanlı Eğitim
GPL	General Public Licence
SMP	Symmetric Multi Processor
WWW	World Wide WebSMP
RPM	RedHat Packet Managment
LILO	Linux Loader
PID	Process ID

ŞEKİL LİSTESİ

ÇİZELGE LİSTESİ

ÖNSÖZ

Hızla büyüyen, gelişen internet teknolojileri, bilgisayar sistemleri eğitim alanında da kendine yer bulmaktadır. Bu konuda çeşitli yöntemler kullanılmaktadır. Bu yöntemlerden bazıları Bilgisayar Tabanlı Eğitim (BTE), Video Konferans, Web Tabanlı Eğitim (WTE), Bilgisayar Destekli BDE vb...Bu yöntemlerin herbirinin diğerine göre avantajlı veya dezavantajlı olduğu noktalar vardır.

Bu tezin amacı Web Tabanlı Eğitim'de kullanılabilecek Linux işletim sisteminin, gerekli sunucu programlarının (Apache, ProFTP, PHP, MySQL) temin edilmesi ve kullanımı hakkında detaylı bilgiler sunmaktır.

Bu tez çalışmam sırasında bana yardımlarını esirgemeyen, beni her zaman destekleyen Sayın Hocam Yrd. Doç. Dr. İbrahim Emiroğlu'na teşekkürü bir borç bilirim.

ÖZET

Web tabanlı eğitim sistemleri incelenmiş. Bu teknolojinin teknik gereksinimleri ve tasarım aşamaları ile eğitmen, öğrenci ve kurum açısından avantaj ve dezavantajları açıklanmıştır.

İnteraktif ders otomasyonu uygulaması geliştirmek için kullanılabilecek Linux işletim sistemi, Apache Web Sunucu, ProFTP FTP sunucu, MySQL Veritabanı sunucu ve PHP sunucu programları tanıtılmış, bu programların temin edilmesi, kurulumu ve birbirleriyle olan etkileşimleri hakkında bilgiler verilmiştir. Ayrıca projenin kodlanması için ihtiyaç duyulan HTML, SQL ve PHP programlama dilinin öğeleri örnekler ile anlatılmıştır.

Anahtar Kelimeler: Uzaktan Eğitim, Web tabanlı eğitim, Linux işletim sistemi, Apache web sunucu, MySQL veritabanı sunucu, PHP ile Web programcılığı

ABSTRACT

In these thesis, web-based training systems are analysed, technical requirements, design phase, advantages and disadvantages of these tools are rewived from the point of instructors, learners and organizations.

Linux operating system, Apache Web Server, ProFTP FTP Server, MySQL Database Server and PHP Server to be used in developing interactive learning otomation system are presented and obtaining and installation of these applications are pointed out. Moreover, HTML, SQL and PHP used as programming languages are explained with examples.

Keywords: Distance learning, Web-based training, Linux operating system, Apache web server, MySQL database server, Web programming with PHP

1. GİRİŞ

Eğitim sistemlerinde insanların zamanla değişen, gelişen ihtiyaçlarını karşılamak, kaliteyi arttırmak ve karşılaşılan bazı sorunları aşmak amacıyla, gelişen bilgisayar ve web teknolojileri kullanılmaya başlanılmıştır. Bunun sonucununda öğrencilerin dersleri internet üzerinden hem görüntü hem de ses destekli olarak izleyebildiği, öğrenci kayıt işlemlerinin yapılabildiği, dersleri takip edebildikleri, ders notlarına, kılavuzlara ve alıştırmalara erişebildiği, sınav olabildikleri ve bu uygulama dahilinde başarı durumlarının ölçülebildiği ve buna göre not değerlendirmesinin yapılabildiği uzaktan eğitim sistemleri tasarlanmış. Böylece karşılaşılan, kapasiteleri hızla artan üniversitelerdeki yer, zaman ve eğitmen sıkıntılarına da alternatif çözümler oluşturulmuştur. Bilgisayar sundukları grafik ve seslerdeki sayısız teknolojik avantajları sayesinde eğitim kalitesine katkıda bulunmuşlardır. Metin içeriğine eklenmiş güzel görüntüler ve ses efektleri herkes için öğrenmeyi ve kavramayı arttırmaktadır. Bu araçlar, özellikle çoçukların pasif olmak yerine daha aktif olmasını sağlaması ile öğrencilerin daha fazla dikkatlerini vermesini ve öğrenilenlerin daha kolay akılda kalmasını sağlamaktadır.

Eğitimi verilecek konular hakkında görsel yönden çekiciliği olan yapılar kurmak ve eğitime katılacak kişilere bunu en iyi şekilde aktarmak temel amaçtır. Üniversiteler tarafından oluşturulan yapılar ise, Internet üzerinden belirli konulara destek veren anlatımları ve sertifika programlarını içermektedir. Bu konuda özellikle Amerika Birleşik Devletleri'nde bulunan bir çok üniversitede benzer uygulamalar ile karşılaşılmaktadır. Bu türde programlar genelde kayıt işlemleri, kursların takip edilmesi,belirtilen zamanlarda projelerin yapılması gibi uygulamaları içermektedir. Bu uygulamaların yanı sıra sınav uygulamaları ise genelde çoktan seçmeli yöntemine dayanmaktadır.

2. WEB TABANLI EĞİTİM

Web Tabanlı Eğitimde (WTE), eğitimin dağıtılması için temel yayın vasıtası olarak internet veya duruma göre intranet kullanılır. Her ne kadar web tabanlı olarak tanımlansa da bu tür eğitim teknolojisinde kullanılan bazı araçlar, örneğin dosya transfer protokolü (FTP), usenet, internet mail, telnet, listservs veya chat odaları web tabanlı değildir.

WTE bilgi dağıtım amaçlı olarakta kullanılır. Bu fonksiyonuda özellikle iyi tasarlanmış hipermetinler sayesinde sağlar. WTE, öğrencilerin kendi arasında ve eğitmen ile öğrenciler arasında elektronik posta (e-mail), bilgisayar konferansı yada listservs ve chat imkanlarıyla iletişime imkan verir. Bu etkileşim sunumu, WTE'yi iletişimin birinci derecede önemli olduğu uzaktan eğitim ve alıştırmalar için kullanılan bilgisayar tabanlı diğer eğitim teknolojilerinden daha uygun kılar. WTE'de malzemesi tasarımında dört yol önerilmektedir. Dağıtım, çeviri, katma değerli çevirim, yaratma ((McCormack ve Jones, 1997).

2.1 Genel Olarak WTE'de Tasarım

WTE malzemesi tasarımı günümüzde çoğu Internet'te bulunabilecek kolayca erişimi mümkün olan çok çeşitli araçlar sayesinde kolaylaşmıştır.

Web tabanlı ortamda sağlanması gereken bazı gereksinimler şunlardır:

1) Yardım sistemi: Yardım sistemi kolay ulaşılır ve anlaşılır olmalıdır.

2) Ara yüz: Ara yüz, basit ve standart olmalıdır. Kullanılan düğmeler (butonlar) anlaşılır olmalı, amaçlarının ne olduğu belli olmalı ve bütün ders sayfalarına aynı yerlere yerleştirilmelidir. Bütün web sayfaları okuması ve anlaşılması kolay olmalıdır.

3) Görüntü: Video ve diğer animasyon görüntülerinin içeriği boğmamasına dikkat edilmelidir. Görüntüler genellikle göze hoş gelir, ancak bilgiye bir değer kattığında kullanılmalı ve küçük tutulmalıdır. Ayrıca alternatif olarak mutlaka metinde bulunmalıdır.

4) Yapı: Web sayfalarının yapısı ders içerisinde kolayca gezinim ve dersin geliştirilebilmesine imkan sağlayacak şekilde tasarlanmalıdır. Anlaşılır bir yapı sitenin bakımı içinde yardımcı olur. Öğrencilerin kaybolmamaları için dersin haritasına bir bağlantıda standart düğme olarak yerleştirilmelidir.

5) Sesler: Web sayfalarındaki sesli bölümler öğrencilere önemli derecede ek bilgi sağlayabilir. Ayrıca farklı bir duyuya da hitap ettiği için öğrenim potansiyelini arttırabilir.

Fakat sesleri ders sayfalarına dahil ederken dikkatli olunması gerekir. Çünkü bir çok öğrencinin bilgisayarında ses dosyaları çalışmayabilir.

2.2 Gereksinimler

1) Donanım: Sunucular, istemciler ve ağ yapısı

a) Sunucular: BTE ve BDE dersleri ile kıyaslandığında WTE sadece kişisel bilgisayarlara değil aynı zamanda yüksek işletim gücüne erişim gerektirir. Web sayfalarının saklanması ve işlenmesi için yüksek kapasiteli sunuculara ihtiyaç duyulur. Çünkü hem bütün ders sayfaları için gerekli yeterli saklama kapasitesi hem de kabul edilebilir cevap süresi sağlanması için gerekli işlem gücüne ihtiyaç vardır.

b) İstemci: İstemci iş istasyonları sunucularda olduğu kadar yüksek işlem gücü gerektirmez. WTE sadece bir web tarayıcısı kullanımı gerekli kılar. Daha gelişmiş özellikler daha yüksek işlem ve derleme gücü olan kişisel bilgisayarları zorunlu yapar. Kullanılan kişisel bilgisayarlar gücü ne olursa olsun, ekranlar mümkün olduğunca yüksek çözünürlüklü olmalıdır.

c) Ağ yapısı: Gerçek anlamda Internet tabanlı olması için kurumun TCP/IP ağ yapısı bağlantısı sağlaması gerekir. Ayrıca bir çok WTE dersi uzun bağlantı süreleri gerektirdiğinden ağ yapısının bant genişliği yüksek seviyeli veri transferini halletmeye yeterli olmalıdır.

2) Yazılım: Web tarayıcıları, elektronik posta paketleri, ağ yapısı yazılım.

2.3 WTE nin avantajları

1) Çok kullanılan web tarayıcıları, Internet üzerinde bedava bulunmaktadır.

2) Daha önce yer sorunu ile ders ortamında bulunamayacak kişiler derse katılabilir.

3) Öğrenciler web üzerinde kendi hızlarında ve kendilerine uygun vakitlerde çalışma imkanı bulurlar.

4) Öğrencilerin çalışma ortamı üzerindeki kontrolü artar.

5) Günümüzde bir çok öğrenci web tasarımına aşina olduğundan WTE derslerinin kullanımı kolaydır.

6) Web ortamının dinamik yapısı itibariyle, eğitmenler gerektiğinde ders malzemesini

güncelleyebilir.

7) İletişim olanaklarından faydalanan web tabanlı ders alan bireyler eğitmenle diğer uzaktan eğitim teknolojilerinde olduğuna kıyasla daha fazla iletişimde bulunabilirler.

8) İletişim tabanlı WTE dersleri eşzamanlı veya eşzamansız olarak dersi alan diğer kişilerle bilgiyi paylaşma veya bir diğerinin probleminin çözümüne yardım edilmesine imkan sağlar.

9) WTE ortamlarında bulunan iletişim araçlarının çoğu ucuz kullanımı kolay ve/veya geliştirilmesi nispeten kolay araçlardır.

2.4 WTE'nin Dezavantajları

1) Derslere internetten katılan öğrencilerin servis sağlayıcısındaki herhangi bir sorun halinde derslere katılamaması söz konusudur.

2) Evden internete bağlanan bir çok kullanıcı düşük hızda ve kapasitede modemler kullanılmaktadırlar (14,4 Kbps den 33,6 Kbps). Ayrıca normal olarak kullanıcı uzun süre bağlantıda kalmak zorunda olacaktır.

3) Veri transferi esnasında virüs dağılımı daha kolay ve kontrolsüz olmaktadır.

4) Tüm kullanıcılarda en az bir web tarayıcısı ve minimum zorunlulukta yazılım gerekmektedir.

5) Evlerinden internete bağlanacek kullanıcıların ISS ücreti ödemsi gerekecektir.

6) Web bazlı malzemelerin en büyük sorunlarından biri sağlanan bilginin kalitesinin kontrolünün zorluğudur. Bu bağlamda Internet üzerinden sağlanan malzemenin ortak bir kalite standardına ulaşması oldukça zordur. Resmi diye bilinen pek çok sitede pek çok bilgi yanlışıyla karşılaşılabilir.

7) WTE'de sürekli ve büyük oranlarda bilgi transferi yapıldığı ve bu bilginin kişisel boyutunun olmamasından dolayı (güvenlik sistemi kullanılsa bile) öğrencilerin bu konuda bazı sorunlar yaşaması olasıdır.

8) Öğrenciler WTE imkanlarından faydalanmak için multimedya özellikleri fazla olan kaliteli bilgisayarlara ihtiyaç duyarlar. Ek olarak uzun süre Internette bağlı olarak da kalmak zorunda oldukları için ikinci bir telefon hattı ve ek bütçeye ihtiyaçları vardır.

9) Eğitmenlerin öğrencilerin dersi görüyor olduğu çevreyi kontrol etme şansları yoktur. Örneğin öğrencilerin dersi gördüğü ortamlarda gürültü, eğitmen tarafından kontrol edilemez.

10) Kullanıcılar Internet ortamında dolaşırken kötü dizayn edilmiş sayfalarda aradıklarını bulma zorluğu çekebilirler. Eğer ki iyi dizayn edilmiş ve kullanıcıya yol gösteren bir site sistemi oluşturulursa, bu sorun engellenebilir.

11) Elektronik iletişim gerek günlük hayatta gerek iş alanlarında pek çok doyurucu özelliğe sahiptir. Elektronik iletişim sonucunda ortaya çıkan medya faktörünün etkinliği ve sosyal mevcudiyet gibi pek çok konu, yüz yüze iletişim eksikliği tabanında artı ve eksileriyle incelenmiştir. Eğitmenler ve öğrenciler elektronik ekipmanların zayıflığından etkilenmektedirler. Elektronik posta da kullanılan bir diğer elektronik iletişim aracıdır. Fakat kişisel postalarla ders dokümanı niteliğindeki postaların karışması, posta hesabının aşırı doluluğu gibi sebeplerle bazen acil cevap bekleyen postalar için kullanışlı bir araç değildir. Ayrıca birisi yüz yüze etkileşimle karşılaştırıldığında daha verimsiz bir iletişim aracı olduğu gösterilmiştir (Daft ve Lengel, 1986).

12) WTE'de kullanılacak malzeme dijital formatta olmalıdır. Diagram, grafik, resim ve benzeri malzemelerde önceden taranmış ve dijital formata geçirilmiş olmalılardır.

13) Kendi malzemesini oluşturacak olan eğitmenlerin bazı sorunları giderme ve kendi malzemelerini rahatça oluşturmaları için en azından belli bir miktar HTML bilgisine sahip olmaları gerekmektedir.

14) Eğitmenler yaşadıkları sorunların giderilmesinde ve derslerin formatını geliştirme safhasında teknik elemanlara ve programcılara bağımlı olma riskini taşırlar.

15) Sanal bir ders ortamında öğrenciler ödevlerini elektronik olarak verirler. Ancak verilen ödevin gerçekten öğrenci tarafından mı yapılıp yapılmadığı bir soru işaretidir. Tabii olarak bu sorunla sadece değerlendirme safhasında karşılaşılmaktadır. Bununla beraber pek çok WTE derslerinde de değerlendirme söz konusu değildir. WTE'de eğitmen öğrencinin sınavı yada ödevi alıp almadığını kontrol edebilir ama verilen görevin gerçekten öğrenci tarafından yapılıp yapılmadığını kontrol edemez. Güvenilirlik için retina ve parmak izi testi gibi pek çok sistem kullanılmaktadır ancak uzaktan eğitim ortamında fiyat bazında oldukça pahalıdır.

3. LİNUX İŞLETİM SİSTEMİ

3.1 Giriş

3.1.1 Tarihçe

Linux işletim sistemi Finlandiyalı Linus Torvalds adlı bir öğrencinin hobi olarak yazmaya başladığı bir yazılımdır. Başlangıçta büyük ve profesyonel bir yazılım olarak düşünülmemiştir. Linus Torvalds'ın bir haber grubuna attığı linux hakkındaki bir mesaj büyük ilgi uyandırmış ve linux'un çekirdeğini geliştiren kişi sayısı artmıştır. Zamanla linux kullanıcıları katlanarak artmış ve ticari şirketler bu işletim sistemine destek vermeye başlamışlardır (Netscape, Intel, Informix, Oracle,...vs.). Bu da ticari anlamda linux'un gelişimini ivmelendirmiş ve çekirdek kodun bir milyon satıra ulaşmasını sağlamıştır. Binlerce çeşit ve tipte cihazlar ve kartlar desteklenmeye başlamıştır. (Çetin, 2000)

3.1.2 Yapısı

Linux işletim sistemi yapısı itibariyle Unix işletim sistemine benzemektedir. Pek çok işletim sistemi Unix'ten esinlenerek geliştirilmiştir. Unix dünya üzerindeki ilk ciddi işletim sistemlerinden biridir. Değişik versiyonları vardır. İlk çıktığında kaynak kodun ücretsiz dağıtılması nedeniyle üniversiteler ve çalışma grupları Unix'i daha iyi yapmak için çalışmışlar ve bu da kısa sürede Unix'in eksik yönlerinin tamamlanarak gelişimine hız katmıştır. Ancak geliştirilmesi ve kullanımı akademik ortamalarda olduğu için görsellik ikinci plana itilmiş gelişim daha çok hız ve güç yönünde oluşmuştur. Bu da herhangi bir kullanıcının altından kalkmasının zor olduğu kullanıcı dostu olmayan bir yazılımın ortaya çıkmasına sebep olmuştur. Pek çok şirket temel standartları destekleyen kendi Unix sistemlerini hazırlamıştır. (Çetin, 2000)

3.1.3 Ücretsiz Kaynak Kodu

Linux'un kaynak kodu ücretsizdir ve internet üzerinden çeşitli ftp arşivlerinden elde edilebilir. Kaynak kodu incelenebilir, ücretli veya ücretsiz olarak tekrar dağıtılabilir. Kaynak kodu yanında yüklü miktarda doküman ve kılavuz sayfası da karşılıksız olarak kullanıma sunulmuştur. Richard Stallman tarafından kurulan GNU projesi bu tür bir felsefenin doğmasına önayak olmuştur. GNU projesi ile birlikte GNU HURD adlı işletim sistemi yazılmaya başlamış, işletim sisteminin kaynak kodu ve işletim sistemiyle birlikte verilen yazılımaların kodları (derleyiciler, hata ayıklayıcılar, editörler vs.) internet üzerinden ücretsiz olarak dağıtılmıştır. GNU projesinin yarattığı bir başka yenilik de GNU GPL (General Public

Licence-Genel Kamu Lisansı). Buna göre programcının isim hakları gözetilerek program üzerinde değişiklik yapılmasına, değişiklikler özel olarak belirtildiği sürece izin verilir. Bu tarz lisanslama programların dağıtımını engellemek yerine insanlar tarafından daha geniş bir şekilde kullanılmasına izin verir. Bu lisanslamaya "COPYLEFT" adı verilir. (Çetin, 2000)

3.2 Temel Özellikleri

Dünyada milyonlarca olarak ifade edilen Linux kullanıcısı bu işletim sistemini çok kararlı, kolayca yapılandırılabilen ve ücretsiz bir işletim sistemi olduğu için tercih ediyorlar. Linux açık gelişim modeline göre geliştirilen güçlü bir işletim sistemidir. Internetin akıl almaz boyutu sayesinde bu işletim sistemine gönül verenlerin sayısı hızla arttırmıştır. Dolayısıyla bu platform üzerinde kullanılan uygulama yazılımları da aynı ivmeyle artmıştır. Tabakalı yapısı sayesinde bir işletim sisteminin kullanıcıya sağladığı tüm olanakları sunabilir. Ayrıca kullanıcının donanımdan bağımsız bir şekilde çalışmasını sağlar. (Çetin, 2000)

3.2.1 Çekirdek

Çekirdeğin adı "Linux"tur. Bütün dağıtımlar bu çekirdeği kullanır. Aralarındaki fark; dosya , dizin yapısı, dağıtıma giren belgeler, paketler, dağıtımın fiyatı ve hedeflediği kullanıcılardır. Çekirdek sistemin beynidir ve sistemin düzgün çalışmasından, kaynakların düzenlenmesinden, kullanıcıların görevlerinin sırayla yapılmasından, bellek denetiminden, yan birimlerin çalışmasından (CD-ROM, teyp, disket sürücü vs) sorumludur. Programlar çekirdek içindeki sistem çağrıları yardımıyla haberleşirler. Kısaca Linux'ta sistemin temel yönetimi çekirdek tarafından yapılır.

3.2.2 Birden Çok Kullanıcı Desteklenmesi

Baştan beri çok kullanılıcılı olarak tasarlanmış ve buna göre programlanmıştır. Bu yapının tercih sebebi kaynakların ortak kullanımı, zamandan ve maliyetten tasarruf sağlanmasıdır. Kullanıcılar çok yoğun çalışmadığı müddetçe sistemi sadece kendileri kullanıyormuş gibi hissederler. (Çetin, 2000)

3.2.3 Birden Çok Görevin Aynı Anda Yapılması

İşletim sistemi birden çok görevin aynı anda yapılmasına olanak veren yapıdadır. Bir kullanıcı bir editör programı aracılığı ile birşeyler yaparken başka bir kullanıcı da uzaktan bağlanarak başka bir programı çalıştırabilir, arka planda bir web sunucu veya ftp sunucu çalışabilir.

3.2.4 Kararlılık

Linux 1.0 sürümünün ortaya çıktığı zamanlardan itibaren beta testlerinden geçmiş ve kararlı bir yapıya bürünmüştür. Çekirdek hala hatalar içermektedir ve yeni kodlar eklendikçe de yeni hatalar ortaya çıkacaktır. Ancak açık gelişim modeli sayesinde devamlı kontrol altındadır. Çekirdek iki farklı sürüm halinde dağıtılır. Örnek olarak 1.2.13 kararlı bir çekirdek sürümdür, 1.3.56 ise halen geliştiriliyor ve test ediliyor anlamına gelir. Bu sürüm numarasının ortasındaki rakamdan anlaşılır. Eğer bu rakam tek ise çekirdek test aşamasındadır, eğer çift ise çekirdek kullanılabilirdir ve kararlı bir yapıdadır.

3.2.5 Çok İşlemci Desteği

SMP (Symmetric Multi Processor – Simetrik Çok İşlemci) desteği Linux'ta bulunur. Sürüm numarası 2 ile başlayan sürümler kuruluşta sistemde kaç işlemci olduğunu bulur ve buna göre kendisini ayarlar. 16 ya kadar işelmcili sistemler desteklenmektedir. NASA üzerinde Red Hat Linux sürümü bulunan nispeten düşük performanslı bilgisayarları kümelendirme adı verilen bir yapı altında hızlı ethernet kartlarıyla birleştirmişlerdir. Bu sayede yoğun işlemci gücü gerektiren programlar daha hızlı bir şekilde dağınık ortamda çözülebilmiştir.

3.2.6 Diğer İşletim Sistemleri

Linux işletim sisteminin önemli özelliklerinden biri de diğer işletim sistemleri için yazılan programları çalıştırabilme özelliğidir. Dosemu emulasyon programı yardımı ile DOS programları bir pencere içerisinde çalıştırılabilirler. Wine ve Wabi de windows emülasyonu yapabilen programlardır ancak hala eksikleri vardır. Executor ise kısıtlı sayıda Macintosh programlarını Linux altında çalıştırabilmektedir.

3.2.7 Linux ve Ağ Teknolojileri

Bilgisayarlar bilgiyi bir noktadan diğer bir noktaya ağlar vasıtası ile iletebilirler. Dünyada çok çeşitli ağ standartları vardır. Linux üç değişik yapıyla ağa bağlanabilir. Broadcast (örn. ethernet), Token Ring (Örn. FDDI) ve noktadan noktaya (Örn. PPP) bağlantı. Linux doğrudan doğruya internete bağlanabilmesi için TCP/IP desteği ile gelir. Bu sayede bu protokolü destekleyen bilgisayarlar ve sistemler ile rahatça iletişim kurabilir. Bunun yanında NetBEUI, Samba, Appletalk gibi protokollerde desteklenir. Linux daha az kullanılan bazı başka protokolleri de destekler.

3.2.8 Güvenlik

Linux 386 ve üstü işlemcilerde korumalı modda çalışır. Korumalı kip sayesinde çalışan bir

program özel şartlar oluşmadıkça diğer bir programın çalışmasını engelleyemez. Çok kullanıcılı sistemler için bu önemli bir özelliktir. Bu sayede herhangi bir kullanıcı sistemi arızaya uğratabilecek veya çalışmaz hale getirebilecek bir programı çalıştıramaz. Kullanıcıların dosya ve dizinleri basit bir koruma mekanizması ile korunmaktadır. Kullanıcı, dosyaları ile ilgili yetkileri değiştirerek bu dosyaları diğer kullanıcılardan saklayabilir veya onlara yetki vererek bu dosyalara erişimlerini sağlayabilir.

3.2.9 Grafik arabirimi

X Window adı verilen grafik arayüzü, sadece metin ekranda çalışmak istemeyen kullanıcılar için alternatif oluşturan bir yazılımdır. X Window altında yirmi kadar değişik pencere yöneticisi vardır. Kullanıcı istediği pencere yöneticisini internetten çekebilir ve kullanabilir. İstemci sunucu modeline göre çalışan X grafik arayüzü, uzaktan sanki makinanın başında oturuyormuşçasına programlarınızı çalıştırabilmenizi ve sisteminizi yönetebilmenizi sağlar.

3.2.10 Diğer Özellikler

Birden fazla programın çalışması halinde bellekte bunlardan sadece biri tutularak bellekten tasarruf sağlanır. Çalıştırılabilir program kullanılabilecek bellekten büyük olursa o zaman belleğe sadece o an için gerekli kısımlar yüklenerek programın çalıştırlıması sağlanır. Belleğin yetmediği durumlarda fazla veri disk ünitesine yazılarak saklanır. Bu bölüme takas alanı (swap space) denir. Her biri iki GB kadar 16 tane takas alanı tanımlanabilir. Paylaşılan kütüphaneler dinamik veya statik olabilir. Yani kütüphane aynı anda birden fazla program tarafından yüklenebilir. Linux'ta pekçok klavye tipi desteklenir. Aynı anda birden çok sanal ekranı destekler (En çok 64 tane) Minix, Fat, VFat, Fat32 gibi dosya sistemlerini destekler ve ayrıca kendine ait olan hızlı ext2 dosya sistemini kullanabilir.

3.2.11 Özet

Linux herkese göre bir işletim sistemi değildir. Unix tabanlıdır ve bundan dolayı kavranması ve kullanımı zordur. Windows ve türevi işletim sistemleri ise çok yaygındır ve üzerlerinde pek çok çeşit uygulama yazılımı vardır. Bu sistemlerle ilgili kaynak bulmak da gayet kolaydır. Ayrıca kullanılan donanım açısından da Linux işletim sisteminin dezavantajları vardır. Bazı donanım ürünleri hala Linux tarafından desteklenmemektedir. Bu nedenle sisteme Linux kurmadan önce varolan donanımların Linux sürücülerinin olup olmadığı araştırılmalıdır. Linux öğrenmesi ve kullanımı emek isteyen bir işletim sistemidir. Kullanıcı bu işletim sisteminde istediği performansı elde etmek için daha çok çaba göstermeli ve araştırmacı bir

kişiliğe sahip olmalıdır. Fakat kişisel kullanım alanındaki bu dezavantajları zamanla azalmaktadır. Birçok büyük yazılım firması işletim sistemine destek verme kararı almış ve kendi ürünlerinin bu platform üzerinde çalışan versiyonlarını da hazırlamaya başlamışlardır. Yeni donanım sürücüleri de hızla yazılmakta ve internette yayınlanmaktadır. Linux sisteminin avantajlı olduğu alan internettir. İnternetin temel yapısı ve organizasyonu Unix tabanlıdır. Bu Linux'a Unix benzeri bir sistem olduğundan internet üzerinde sunucu olarak gayet yüksek performans sağlamaktadır. Bu alanda düşük maliyetli donanım üzerinde bile yüksek performanslar elde edilebilmektedir. İnternetteki pekçok site Linux, Apache web sunucu ve MySQL veritabanını kullanmaktadır. Diğer önemli bir avantajı da ücretsiz olmasıdır. Benzer işletim sistemleri yüksek fiyatlarla satılmaktadır. Dağıtımının tek bir firmaya bağlı olmaması dünyanın dört yanından insanların geliştirilmesine katkı yapmasına olanak sağlamaktadır. (Petersen,2001; Çetin,2000)

3.3 Kurulum (Red Hat dağıtımı)

Linux kurulumu için en az 160 Mb boş alana ihtiyaç vardır. Kuruluş sırasında dağıtımın sağladığı tüm paketlerin kurulması istenmeyebilir. Bu durumda kullanıcı sadece ihtiyacı olan paketleri yükler. Daha sonra uygulamaları kaldırabilir veya kurabilir. Red Hat Linux'un RPM teknolojisi ile herhangi bir paket saniyeler içinde kurulabilir, silinebilir veya güncellenebilir. Red Hat Linux kaynak kodu ile birlikte verildiğinden her program yeniden derlenebilir. Bu sayede kaynak kod içinde istenilen değişiklikler rahatça yapılabilir. Linux sadece dağıtımlarda bulunan yazılımlarla sınırlı bir işletim sistemi değildir. Bunlara ek olarak çok çeşitli uygulamalar bulunmaktadır. Bu yazılımlar CD ortamında alınabilir veya internetteki FTP arşivlerinden elde edilebilir. Linux aynı sabit disk üzerinde başka işletim sistemleri ile birlikte çalışabilir. Ama bunun için sabit disk üzerinde yeterli boş alan bulunmalıdır. Daha sonra ayrılan bu bölümler (partitions) işletim sistemine en uygun şekilde formatlanmalıdır. Linux kendi disk formatı olan ext2 ile en iyi performansına ulaşır. Disk bölümlendirme için herhangi bir program kullanılabilir (Örn. fdisk). Kurulum sırasında Disk Druid programı da bölümlendirme işleminin yapılmasını sağlar. Burada önemli bir nokta swap bölümünün ayrılmasıdır. 10 ila 60 Mb lik bir alan bu ihtiyacınızı karşılayacaktır. Swap alanı ayırmak belleğin daha verimli kullanılmasını sağlayan ve performansı arttıran bir işlemdir. Red Hat dağıtımı kurulum sırasında birkaç seçenek sunar. İş istasyonu, Sunucu veya Özel kurulum seçeneklerinden biri seçilebilir. Daha sonra dil seçimi, fare tipi, klavye, ekran ayarları yapabilecek pencereler vasıtasıyla bu ayarlar da kurulum esnasında düzenlenir.

Sonra LILO (Linux Loader) kurulumu ile ilgili bir pencere gelir. LILO Bilgisayarın açılış sektörüne kendini kaydeden ve bilgisayar açıldığında ilk olarak çalıştırılan programdır. Sabit diskte birden fazla işletim sistemi varsa LILO istenilen işletim sisteminin seçilmesine olanak tanır. Bilgisayar açılırken gelen LILO satırında TAB tuşuna basıldığında secenekler görüntülenir. Kurulum daha sonra ağ ayarlarının, zaman seçiminin yapılacağı ekranlar ile devam eder. Bu ayarlar Linux kurulduktan sonra da yapılabilir veya değiştirilebilir. Kurulum sırasında veya daha sonra sisteme yeni kullanıcılar tanıtılabilir. Yalnız "root" kullanıcısı kurulum sırasında yaratılmalıdır. root Linux işletim sisteminde en yetkili kullanıcıdır. Son olarak dağıtımla gelen paketlerden yüklenmesi istenenler seçilir, X Window ayarları belirlenir.

Bütün kuruluş adımları tamamlandıktan sonra Linux dosyaları ve seçilen paketler ile ilgili dosyalar sisteme kopyalanır. Artık sistem kullanıma hazır hale gelir. Sistem tekrar başlatılınca eğer LILO kurulmuşsa:

boot :

şeklinde bir satır gelir, burada diskteki işletim sistemlerinden biri seçilebilir. Tanımlı olan işletim sistemlerini görmek için "tab" tuşuna basmak gerekmektedir.(Petersen,2001)

3.4 Sisteme Giriş

Kurulum sırasında grafik arayüzü seçilmediyse sistem başlatıldığında aşağıdaki gibi, bir mesaj gelir.

Red Hat Linux release 6.3 (Zoot)

Kernel 2.2.14 on an i686

login:

Burada login mesajının yanına kullanıcı adı yazılır ve enter tuşuna basılır. Sistem bu sefer de şifre giriş alanını açar. Şifre bilgisi de girildikten sonra kullanıcının tipine göre # veya $ komut satırı gelir. Buradan komutlar girilir ve işletilir. Örnek olarak sisteme yeni bir kullanıcı tanıtmak istenirse

#adduser Ahmet

şeklinde bir komut girilmelidir. Böylece Ahmet adında bir kullanıcı yaratılmış olur.

şifre girmek için ise

#passwd

Bu komuttan sonra gelen adımlarda iki kere şifre girilir .Böylece şifre tanıtılmış olur. Linux küçük, büyük harf duyarlı bir işletim sistemidir. "Ahmet" ve "ahmet" sistem için aynı bilgiler değildir.

Linux komutları hakkında sistemden yardım almakta mümündür. Bir komut hakkında sistemde bulunan bilgileri görüntülemek için "man" komutu kullanılır.

#man passwd

Kılavuz dosyaları çoğu dağıtımlarda /usr/man/ dizini altında bulunur. Sistemde aynı komut hakkında değişik tipteki kullanıcılar hitab eden yardım bilgileri bulunabilmektedir. İstediğimiz seviyedeki bilgiye ulaşmak için man komutunu

#man 2 mount

şeklinde kullanmalıyız. Bu mount komutu ile ilgili 2 seviyeli bilgileri gösterir.

3.5 Komut Yorumlayıcısı

Linux , Unix ve benzeri işletim sistemlerinde kullanıcının komut yazmasını sağlayan ve girilen komutlar için gerekli işlemleri yapan programlara kabuk (Shell) denir. Bu programlar kullanıcı ile sistem arasında etkileşimi sağlar. Komutlar belli kurallara göre tanımlanmıştır ve kullanıcı bu tanımların dışına çıkamaz. # veya $ işaretinin görüntülenmesi kullanıcının kabuk programına eriştiğini göstermektedir. Komut satırında komutlar tek başına veya gerektiğinde seçenekleriyle kullanılabilir.

Örneğin ls komutu bir dizin içerisinde yer alan dosyaların listesini görüntülemek amacıyla kullanılır. ls –l veya ls –a vs.. şeklinde değişik parametrelere göre dosyaları listelerken eleme yapar, özelliklerini gösterir veya göstermez.. Birçok komut ile birlikte bazı özel karakterler kullanılabilir. Bu karakterler arasında yer alan "*" ve "?" işaretleri özellikle önem taşımaktadır. "*" işareti dosya isimlerinin kullanımında esneklik sağlar. Bu karekter dosya isimlerinin herhangi bir yerinde kullanılmak suretiyle dosya isimlerindeki ortak bölgeleri tanımlar. "?" karekteri de benzer bir işlev taşır tek farkı her işaret için sadece bir karşılaştırma işlemi yapmasıdır. Mesela belirli bir dizindeki per ile başlayan dosyaların listelenmesi

istendiğinde kullanılacak komut ls dir.

$ ls per*

ls komutu bu secenkler kullanıldığında o dizindeki per1, per2 ,.... şeklindeki dosyalar görüntülenir. *per, * per *, şekillerinde de kullanılabilir. Linux ve Unix'te küçük, büyük harf ayırımı vardır ve sistem komutları genellikle küçük olarak yazılmıştır. Ayrıca bazı karekterler özel amaçlar için kullanıldığından dosya isimlerinde bulunamaz. (Örn. "/"). Komutlar tersi istenmedikçe girdilerini standart girdiden (klavye) alır, çıktılarını standart çıktıya (ekran) yazar. Bu özellik yönlendirme ve boru operatörüyle değiştirilebilir. Bu da birçok işlemin kolayca yapılabilmesini sağlar. (Özkan,1996)

3.5.1 Geri Planda Komut Çalıştırma

Normal olarak bir komut çalıştırıldığında, komut yorumlayıcısı bir sonraki komutun kullanımına ancak önceki bittiğinde izin verir. Fakat bir komutu geri planda çalıştırmak olasıdır. Böylece komut yorumlayıcısı komutun tamamlanmasını beklemez. Onu geri planda çalıştırarak kullanıcıya başka bir yorumlayıcı sunar. Komutları geri planda çalıştırmak için sonuna bir "&" işareti koymak yeterlidir. Bu olanak, komutların dışında kullanıcının hazırladığı programlar içinde geçerlidir. (Özkan,1996)

$ sort +1 –o per1 personel &

3.5.2 Giriş ve Çıkış Yönlendirme

Normalde çalıştırılan komutun yarattığı sonuçlar ekranda görüntülenir. Standart giriş, çıkış birimi ekran kabul edilmektedir. Kullanıcı isterse giriş çıkış yönünü değiştirebilir. Bunu sağlamak için ">", "<" işaretlerinden yararlanılır.

ls –l > liste

Burada listelenen dosya isimleri liste adlı dosyaya yazılır. liste adında bir dosya yoksa yaratılır varsa içi önce boşaltılır. Eğer dosyanın önceki içeriğinin silinmesini istemiyorsak o zaman komutu

ls –l >> liste

şeklinde yazmalıyız. Böylece yeni bilgiler dosyadaki bilgilerin sonuna eklenir.

3.5.3 Komut Borusunda Ara Dosyaların Saklanması

Komut borusu yardımıyla işlemlerde, istenirse bu süreç içinde yaratılan dosyalar saklanabilir. Bu bir komutun çıktısının başka bir komut için girdi olarak kullanılabilmesi olanağını verir. Bu işlem için "|" (pipe) karakteri kullanılır.

$ ls –al | lpr

Burada ls komutuyla seçilen dosya bilgileri lpr komutuyla yazıcıya gönderilmiştir.

3.5.4 Komutların Zamanla İlişkili Olarak Çalıştırılması

Bir komut belirli bir zaman sonra çalıştırılacaksa ve bu arada komut yorumlayıcısının başka bir iş yapması istenmiyorsa "sleep" komutu kullanılabilir.

$ sleep 60;date

şeklinde kullanıldığı zaman 60 saniye sonra gunün tarihi görüntülenir. Eğer komut belirli bir saatte çalıştırılmak isteniyorsa at komutu kullanılabilir. Ayrıca batch komutu da yapılması gereken bir işin kuyruğa atılmasını ve sistemin uygun gördüğü anda o işi yapmasını sağlar.

$ at zaman [tarih] [+ artma]

3.6 Dosya Yönetimi

3.6.1 Linux Dosyaları

Linux işletim sisteminde dosyaları üç başlık altında toplayabiliriz.

- Dizinler
- Sıradan dosyalar
- Özel dosyalar

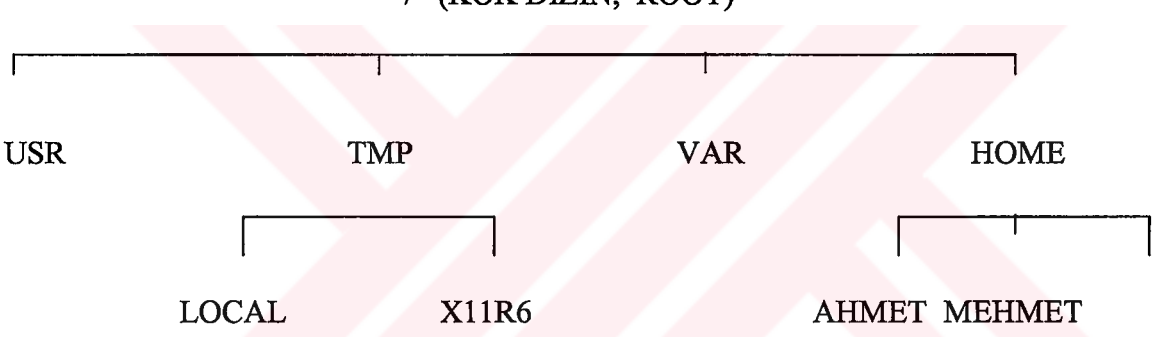
Dizin bir dosya olmasına karşılık içinde birçok dosya barındırabilmektedir. Başka dizinleri de içerebilir. Unix ve Linux hiyerarşik bir dizin yapısına sahiptir. Sıradan dosyalar bilgi dosyalarıdır. Bu dosyalar metin, kaynak kod veya programlardan oluşabilir. Programların kullandığı veri dosyaları da sıradan dosya olarak bilinir. Özel dosyalar ise sistem tarafından kullanılan dosyalardır. Donanımlarla ilgili birtakım bilgiler saklanır. (Akerdem ve Usta,2000)

3.6.2 Dosya İsimleri ve Dizin Yapısı

Linux işletim sisteminde her dosya belirli bir isme sahip olmalıdır. Dizin ve dosya isimlerinin başında nokta karekteri olursa bunlar gizli hale gelirler ve parametresiz ls komutuyla görüntülenmezler. Dosya ve dizin isimleri 255 karekteri aşamazlar.

Sisteme girince önceden tanımlanmış bir dizin altında bulunursunuz. Bu dizin normal kullanıcılar için genellikle /home ve ardından kullanıcı dizini ismidir. Bulunulan dizinin adını görmek için pwd (print working directory) komutu kullanılabilir. Linux ve Unix'te dizinler hiyerarşik yapıya sahiptirler. Bu yapının en üst noktasında kök dizin "root" bulunur, diğer dizinler bu dizinin alt dizinidir.

Çizelge 3.1 Linux dizin yapısı



Sisteme tanıtılmış her kullanıcının bir dizini vardır ve sisteme girilmesiyle birlikte kullanıcı kendisine ait dizine ulaşır. Kullanıcı dizinini iki türlü düzenleyebilir. Birincisi bütün programlarını ve dosyalarını hep bu dizin altında toplamak suretiyle tek düzeyli bir yapıda. Ya da çok düzeyli bir yapı altında alt dizinler oluşturmak suretiyle programlarını ve veri dosyalarını ayrı ayrı dizinlerde saklayabilir. Dizin yaratmak için kullanılan komut mkdir komutudur. Bu komutla dizin yaratırken ya açılacak dizinin bir üst seviyesinde olunmalı yada dizin yolu tam olarak verilmelidir. Dizin değiştirmek için ise MS-DOS da olduğu gibi cd komutu kullanılır. (Özkan,1996)

3.6.3 Dosyaların Listesi ve İçerikleri

Dosya ve dizinleri görüntülemek için ls komutu kullanılır. Bulunulan dizin hakkında bilgi almak için parametresiz olarak yazılır eğer "–a" parametresiyle kullanılırsa gizli dosyaları da görüntüler. "-l" parametresi dosyalarla ilgili tüm detayları listeler. Bu bilgiler dosyanın sahibi, ne zaman yaratıldığı ve grubudur. Ayrıca dosya erişim hakları da görüntülenir. ls –al şeklinde de kullanılabilir. Dosyaların içeriğini görüntülemek için less komutundan yararlanılabilir. Ok

tuşları ile dosya içinde hareket edilebilir ve q karekteri ile de dosyadan çıkarsınız. Tüm dosyayı ekrana yazdırmak için "cat dosyaadı" yazmak yeterlidir.

3.6.4 Dosyaların Kopyalanması ve Silinmesi

Dosyaların kopyalanması için cp ve bir yerden başka bir yere taşınması için mv komutları kullanılır. mv komutu aynı zamanda dosya isimlerini değiştirmede de kullanılabilir.

$ cp yazı.txt /tmp

$ mv yazı.txt /tmp/deneme

Kopyalama işlemi sadece dosyalar üzerinde değil dizinler üzerinde de yapılabilir. Farklı dosya sistemleri üzerinde olmamak kaydıyla bir dizin altındaki herşey başka dizinler altına kopyalanabilir. Dizin kopyalama işleminde –r parametresi kullanılır. Taşıma işleminde bu parametreye gerek yoktur.

Linux'ta bir dizin içinde yer alan dosyalar rm komutu yardımıyla silinmektedir. Bu komut dikkatli bir şekilde kullanılmalıdır. Hatalı kullanım nedeniyle istenmeyen sonuçlar oluşabilir. Kullanımı

rm [seçenekler] dosya, ...

$ rm *

yazarak bir dizinde içerisindeki tüm dosyalar silinebilir.

3.6.5 Erişim Hakları

Erişim hakları Linux dosya sistemini güvenliğinin belkemiğini oluşturur. Her dosyaya ayrı ayrı verilebilen erişim izinleri sayesinde çok daha rahat bir sistem yönetimi gerçekleştirilir. Üç çeşit erişim hakkı vardır.

Okuma izni: Dosyanın okuma izni varsa içeriği görüntülenebilir ve içerdiği dosyalar listelenebilir.

Yazma izni: Dosyanın yazma izni varsa o dosyaya yazılabilir, değiştirilebilir ve dosya silinebilir. Dizine de aynı şekilde yazma izni verildiyse o dizindeki dosyalar yazılabilir ve silinebilir.

Çalıştırma izni: Dosyayı çalıştırma hakkı verilir.

Bir dosya yaratıldığında sistem tarafından bazı default izinler verilir. Genellikle bu izinler okuma ve çalıştırmadır. Dosya yaratılırken verilen bu izinler umask komutuyla değiştirilebilir. Linux ve Unix'te yine bu erişim haklarını sınırlamak üzere üç tip kullanıcı tanımlanmıştır. Bunlar:

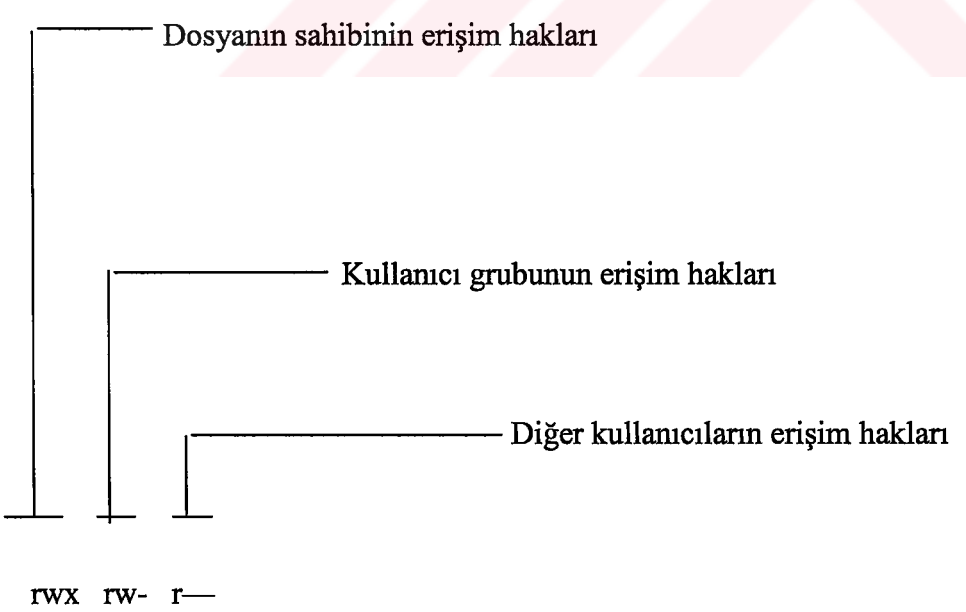
Dosyanın sahibi (owner)

Kullanıcının grubu (group)

diğer tüm kullanıcılar.

Dosyanın sahibi o dosyanın yaratıcısıdır ve sahibi olduğu dosyalar üzerinde her türlü erişim hakkına sahiptir ve istediği işlemleri yapabilir. Unix'te kullanıcılar belirli gruplar oluşturabilir. Bu sayede her bir kullanıcı grubuna dahil olan kişiler eğer grup için erişim yetkisi verilmiş ise grubun diğer üyelerinin dosyalarına erişebilirler. Hangi kullanıcının hangi gruba dahil olduğu /etc/group dosyasında yer almaktadır. Okuma izni r, yazma izni w, çalıştırma izni x ile gösterilir. Her dosya, kullanıcı ve erişimi sınırlayan 9 bitlik bir izin ifadesine sahiptir.

Çizelge 3.2 Linux dosya erişim bilgileri ve sınırlama bitleri



"r-- r-- ---" bu erişim hakkı incelenirse dosya sahibi sadece okuyabilir, gruptakiler sadece okuyabilir, diğer kullanıcılar erişemez.

3.6.6 Dosya Erişim Bilgilerinin Değiştirilmesi

Bir dosyanın sahibinin değiştirilmesi olası bir durumdur. chown komutuyla sağlanan bu olanak dosyanın sahibi veya sistem yöneticisi (root) tarafından kullanılabilir.

chown yeni sahibi dosya ismi

chown yeni sahibi dizin ismi

Bir dosyanın sahibini görüntülemek için ls –l komutunu ve seçeneğini kullanırız. Dosyanın grubunu değiştirmek için chgrp komutu kullanılır.

chgrp yeni grup adı dosya ismi

chgrp yeni grup adı dizin ismi

şeklinde kullanılır. Dosyaya olan erişim izinleri (okuma, yazma, çalıştırma) chmod komutuyla değiştirilir.

chmod izin modu dosya ismi

chmod izin modu dizin ismi

İzin modlarını iki türlü ifade etme olanağı vardır. Birincisi onaltılık moda göre tanımlama, İkincisi sembolik tanımlama.

$ chmod 600 dosya1

yazarak dosya1'in sahibine okuma ve yazma izni verilmiş olunur. Aynı sonuca sembolik kodlama ile ulaşılmak istenirse

$ chmod go-rwx prog1

Burada go –rwx tanımı ile, grup ve diğer kullanıcıların rwx yetkileri alınmaktadır. Böylece prog1 dosyasının izinleri rwx------ şeklinde olacaktır.

3.7 İşlem Yönetimi

3.7.1 İşlem Nedir ?

İşlemin ne olduğunun anlaşılması için işlem ve program arasındaki ilişkinin anlaşılması gereklidir. Program, bilgisayarın sabit diskinde saklanan ve ancak gerektiğinde çalıştırılan yazılımlardır. Program çalıştırıldığında doğal olarak bilgisayarın ana belleğine yerleşerek istenilen görevleri yerine getrimeye çalışır. Bir programın kopyası belleğe yüklendikten sonra

burada artık işlem adını alır. Bellek üzerindeki işlemlerin yönetimi yönetici bir program tarafından yapılır. Bu program bilgisayarın kaynaklarını kullanıcılar arasında uygun bir biçimde paylaştırır. Her bir işlem için belirli zaman aralığı ayrılır. Bu zaman aralıklarında işlemler sırayla çalıştırılır. Belleğin yetersiz kaldığı durumlarda yönetici program yeni işlemleri disk üzerine yerleştirerek yapmaya çalışır. (swapping) (Akerdem ve Usta 2000)

3.7.2 İşlemler Hakkında Bilgi Edinmek (ps – komutu)

İşlemlerin durumları hakkında bilgi edinmek üzere ps komutundan yararlanılır. Bu komut aktif işlemler hakkında çeşitli bilgileri görüntüler. Kullanımı:

ps [seçenekler]

Bazı seçenekler

-e Tüm işlemler ile ilgili bilgileri görüntüler.

-d Grup liderleri hariç, tüm işlemler

-a Grup liderleri ve terminallerle ilişki olanlar dışında kalan tüm işlemler

-f Tam listenin üretilmesine olanak tanır.

-l Ayrıntılı listenin görüntülenmesine olanak tanır.

ps komutu çalıştırıldığında ekranda işlemlerin durumları kolonlar halinde görüntülenir. Değişik kolonlardaki bilgiler işlemler hakkındaki değişik bilgileri gösterir. Bu kolonlardan işlemin bellekte mi, disk üzerinde mi çalıştırılıdığı, sistemin yaptığı bir işlem olup olmadığı, uygulamanın çalıştığını veya bekleme modunda olup olmadığı ve benzeri birçok bilgi anlaşılır. Ayrıca işlemin numarası (PID) bilgisi de görüntülenir.

3.7.3 Bir İşlemi Öldürmek (kill – komutu)

Çalışmakta olan bir işlemi kesmek üzere kill komutundan yararlanılır. Bu komut kill [-sinyal] PID ... şeklinde yazılır. Öldürülecek işlemlerin PID numaraları ps komutuyla elde edilir. Kill komutu ile birlikte birçok sinyal tanımlanabilir. Bunlardan –9 işlemi öldürür. Kullanıcı kill komutu ile ancak kendine ait işlemleri öldürebilir. Diğer işlemleri sistem kullanıcısı öldürebilir. Çalışmakta olan işlemlerin PID numaralarını almanın bir başka yolu da whodo komutudur. Bu komut yardımıyla o anda hangi kullanıcının hangi işlemleri çalıştırdığı işlem numaraları ile birlikte öğrenilebilir. (Akerdem ve Usta 2000)

```
$ ps
  PID TTY TIME COMMAND
 1140 015  0:03  sh
 2361 015  0:00  ps
$ kill –9 2361
```

3.8 Kabuklar Ve Kabuk Programları

Daha önce de bahsedildiği üzere kullanıcılar komutlarını kabuk programları (Komut Yorumlayıcısı) yardımı ile işletebilirler. Unix ve Linux üzerinde kullanılabilen birden fazla komut yorumlayıcısı vardır. Kullanıcı bunlardan istediğini kullanabilir. (Bourne, C , Bash ..)

3.8.1 Bash Kabuğu Özellikleri

Bash'in kullanıcıya zaman kazandıran en önemli özelliklerinden birisi dosya isimlerini tamamlamasıdır. Komut satırında tamamlanmamış bir komut veya dosya ismi yazdıktan sonra TAB tuşuna basılırsa satır tamamlanacaktır. Eğer komut satırındaki karakter kümesiyle başlayan birden fazla komut varsa bir sinyal sesi duyulacak ve yeteri kadar karakterin yazılması beklenecektir. (Petersen, 2001)

```
$ ls
postgres      mandel.doc     lilo-howto
$ vi post <TAB>
$ vi postgres
```

Komut satırındaki karakterler dosyayı veya komutu tanıtmaya yetmiyorsa, <TAB> tuşunun iki kez ard arda basılmasıyla ekrana mümkün olan tüm dosya isimleri getirilebilir.

```
$ ta <TAB> <TAB>
tac  tail  talk  tar
$ talk
```

Bash, komut satırında iken satırın kolayca değiştirilebilmesini sağlar. Böylece komut çalıştırılmadan önce birkaç tuş darbesiyle üzerinde değişiklik yapılabilir. Klavye üzerindeki

alt ve üst yön tuşları, daha önce yazılan komutların görülmesini ve arasında seçim yapılmasını sağlar. Sağ ve sol yön tuşları ile seçilen komutun üzerinde değişiklikler uygulanabilir. `alias` komutu ile bir komut veya komut kümesinin yerine bir isim tahsis edilebilir. İşleyişi bir makroya benzeyen bu komut yardımıyla uzun komutlar, daha kısa komutlarla tanımlanabilir. Bir `alias` komutu , anahtar kelimeyle başlar, ardından bir eşittir (=) işareti ve yerine kullanılacağı komut yazılır. Arada boşluk bırakılmaz.

```
$ alias dir='ls -al'

$ dir

total 668

-rw-r--r--   1 muharrem users 016 Dec  7 13:51 .profile

-rw-r--r--   1 ramazan  users 277 Nov 26 13:02 .signature
```

Daha karmaşık takma adlar da tanımlanabilir:

```
$ alias yedek="cd /var/log; tar -zcvf yedek.tgz cron debug lastlog; cd -"
```

3.8.2 Özel Kabuk Değişkenleri

Sisteme girildiği zaman Linux kullanıcıya bir kabuk tahsis eder ve kabuk üzerinde değişkenler tanımlanmasına izin verir. Kabuk içinde bir kabuk programı (script) çalıştırılınca sistem tarafından bir alt kabuk daha yaratılır. Bu andan itibaren iki kabuk çalışır, birisi sisteme girildiği anda tahsis edilen, diğeri de programın çalıştırılabilmesi için sistem tarafından çağırılan. Program bittiği anda alt kabuğun işlevi sona erer ve sistem tarafından öldürülür (Özkan 1996).

Bir kabuk altında tanımlanan değişken o kabuğa özgüdür. Bir kabuk altında erişilebilen değişkene diğer bir kabuk erişemez. Fakat bash kabuğu altında bir çevre değişkeni yaratılırsa, sistem tarafından her alt kabuğa bu çevre değişkeni kopyalanır. Böylece bir değişken yaratılır ama her alt kabuk tarafından kullanılabilir. Bash kabuğunda yaratılan çevre değişkenlerinin normal değişkenlerden tek farkı, bu değişkenin diğer alt kabuklar tarafından tanınmasıdır. Bir çevre değişkeni belirtmek için `export` komutu kullanılır. Aşağıdaki örnekte *dosyam* isimli çevre değişkeni tanımlanıyor.

```
$ export dosya1="deneme.txt"

$ echo $dosyam
```

internet.98.txt

Bundan sonra tüm alt kabuklar altında bu değişken tanımlı olacaktır.

`env` komutu yardımıyla sistemde öntanımı yapılan veya sonradan tanımlanan tüm değişkenler ekrana listelenir.

$ env

HOME=/home/gorkem

SHELL=/bin/bash

LS_OPTIONS=--8bit --color=tty -F -b -T 0

PS1=\h:\w\$

PS2=>

LOGNAME=gorkem

OSTYPE=Linux

Sisteme girildiği anda tanımlanan bazı komutlar ve açıklamaları aşağıda verilmiştir:

HISTFILE : Tüm yazılan komutlar `.bash_history` adlı bir dosya içinde tutulur. Her kullanıcının kendi ev dizinleri içinde kullanıcıya özgü bu dosyadan vardır.

PATH : PATH değişkeninde bir komut yazıldığı anda sistem tarafından aranacak olan patika listesi görüntülenir. Örnek olarak `who` komutu `/usr/bin/` dizini altındadır ve bu bilgi PATH değişkeninde saklı tutulur. Kullanıcının yaptığı programları çalıştırabilmesi için PATH değişkeninde bulunduğu dizini de tanımlaması gerekir.

PATH değişkeninde her dizin ':' işareti ile birbirinden ayrılır. Örnek olarak `/usr/local/bin:/usr/bin` dizisi sırasıyla `/usr/local/bin` ve `/usr/bin` patikalarına karşılık gelir. Kullanıcı PATH değişkenine yeni girdiler ekleyebilir. Aşağıdaki örnekte bunun nasıl uygulandığı görülüyor. Kullanıcı kısaca PATH değişkeninin sonuna kendisinin istediği patikayı veya patikaları ekler.

$ echo $PATH # PATH degiskenini ekrana bas

$ export PATH=$PATH:~/net # buna /usr/net patikasini ekle

SHELL : O an kullanılan kabuğun patika ismini verir. Kabuk programları genellikle `/bin` dizini altında tutulurlar. Her kabuğun patika ismi sistemdeki `/etc/shells` dosyasında bulunmalıdır.

$ echo $SHELL

/bin/bash

HOME : Kullanıcının ev dizinini gösteren patikayı ekrana basar. Her kullanıcının ev dizini, sistemde hesap açılırken sistem görevlisi tarafından belirlenir. Linux Slackware dağıtımı altında bu dizin öntanımlı olarak `/home` 'dur. Aşağıdaki örnek kullanıcının ev dizinine ait patikayı ekrana basıyor.

$ echo $HOME

/home/muharrem

LOGNAME : Sistemdeki kullanıcı hesabınının ismini tutar. Her kullanıcı için farklı bir değerde olur.

$ echo $LOGNAME

root

TERM : Kullanıcının halen üzerinde çalıştığı terminal tipini görüntüler. `vi` ve `pico` gibi editörler çalıştırıldıkları anda TERM çevresel değişkenine bakarlar. Bu sayede ekran ile ilgili bilgileri önceden düzenleyebilirler. Aşağıdaki komut önce halihazırdaki TERM değişkenini görüntülüyor, ardından değişkene başka bir değer atıyor.

$ echo $TERM

linux

$ export TERM=vt100

$ echo $TERM

vt100

3.8.3 Sisteme Giriş Dosyaları

Sisteme girerken her giriş anında çalıştırılan birtakım dosyalar bulunur. Bash kabuğu ile doğrudan ilgili olan giriş dosyaları arasında

/etc/profile

~/.bash_profile

~/.bash_login

~/.profile

~/.bashrc

sayılabilir. Kullanıcı bu dosyaların kendine ait olanlarını dilediği gibi değiştirebilir. Sisteme girerken tanımlamak istediği değişkenleri tanımlar, çalıştıracağı programları yazar. `/etc/profile` dosyasını ise tüm kullanıcılar ortak kullanır. Her kullanıcı sisteme girdiğinde sistem tarafından bu dosya çalıştırılır. Bu dosyada kullanıcıların ihtiyacına yönelik özel sistem değişkenleri bulunur, ek olarak birtakım kontrol ve kayıt işlemleri yapılır.

Örnek bir `/etc/profile` dosyası şekildeki gibidir.

```
if [ "$TERM" = "" -o "$TERM" = "unknown" ]; then
 TERM=linux
fi
#PS1='`hostname`:`pwd`# '
if [ "$SHELL" = "/bin/pdksh" -o "$SHELL" = "/bin/ksh" ]; then
 PS1="! $ "
elif [ "$SHELL" = "/bin/zsh" ]; then
 PS1="%m:%~%# "
elif [ "$SHELL" = "/bin/ash" ]; then
 PS1="$ "
else
 PS1='\h:\w\$ '
fi
PS2='> '
```

```
ignoreeof=2

export PATH DISPLAY LESS TERM PS1 PS2 ignoreeof

umask 022

# set up the color-ls environment variables:

if [ "$SHELL" = "/bin/zsh" ]; then

  eval `dircolors -z`

elif [ "$SHELL" = "/bin/ash" ]; then

  eval `dircolors -s`

else

  eval `dircolors -b`

fi
```

`.bashrc` dosyası, her bash kabuğu veya alt kabuk çalıştırıldığı anda okunur. Her kabuk programı çalıştırılınca bir alt kabuğa ihtiyaç olduğundan bu durumda `.bashrc` dosyası da okunacak ve içerdiği değişkenler proram çalışmadan önce tanımlanacaktır.

Sistemden çıkarken varsa çalıştırılan dosyanın ismi `.bash_logout`'tur. Bu dosya ile kullanıcı sistemi terkederken fazladan işlemler yapabilir. Aşağıdaki örnekteki `.bash_logout` dosyasında kullanıcı sistemden her çıktığında ekran temizlenecek ve en üste 'Hoscakal' yazacaktır.

```
clear

echo "Hoscakal!"
```

3.8.4 Kabuk Programlarına Giriş

Linux işletim sisteminde Unix'te olduğu gibi komutlar tek tek çalıştırılabileceği gibi topluca da çalıştırılabilir. Komutlardan oluşan yordamlar hazırlanarak tıpkı yüksek düzeyli bir programlama dili işlevlerini yerine getirmek olasıdır. Bu sayede kabuk programları yazılabilir. Kabuk programları içerisinde değişkenler, parametreler, özel karekterler, kabuk deyimleri, komutlar ve açıklama satırları yer alabilir. Üstelik kontrol deyimleri tanımlanabilir, dosyalardan okuma yazma işlemleri yerine getirilebilir. Her kabuğun kendine özgü

programlama dili yapısı vardır. Bash kabuğu ise güçlü programlama özellikleriyle karmaşık programların rahatça yazılmasına izin verir. Genellikle, bir programı oluşturacak olan komutlar bir dosyaya yazılırlar ve ardından bu dosya çalıştırılır. Herhangi bir editör yardımıyla yazılan program, daha sonra kabuk altında çalıştırılır. Bir kabuk programı diğerlerini çalıştırabilir. Bu düzende kabuk programlarını daha karmaşık komutların biraraya gelmiş ve yapısallaşmış haline benzetebiliriz. Bash'in en büyük dezavantajı, gayet yavaş, hantal olması ve sistem kaynaklarını kolayca tüketmesidir. Bu dosya yaratıldıktan sonra doğrudan dosyanın ismi girilerek veya dosya isminden önce '.' karakteri getirerek çalıştırılabilir. Bir kabuk programı, çalıştırma bitini 1 yapmak suretiyle "çalıştırılabilir" hale getirilir. `chmod` komutu yardımıyla bir programı çalıştırılabilir yapmak için ,

$ chmod +x komut-ismi

yazılabilir. Bundan sonra programın ismi yazılıp enter tuşuna basıldığı zaman bir program Linux komutuymuş gibi çalışacaktır.

$ cat calistir

echo -n "Tarih : "

date

$ chmod +x calistir

$ calistir

Tarih : Sun Dec 8 07:11:51 EET 1996

Yukarıdaki örnekte "calistir" isimli iki satırlık bir kabuk programının önce içeriği ekrana yazıldı, ardından çalıştırılacak duruma getirildi ve çalıştırıldı. Kabuk programları yazarken dosyanın işlevini ve her satırdaki komutun veya komut kümesinin ne amaçla kullanıldığını gösteren açıklama satırları kullanmak işe yarar. Bir açıklama eklemek için satır başına (veya boş satıra) # işareti eklenir ve ardından istenilen cümle girilir. # işaretinden sonraki tüm satır kabuk tarafından gözardı edilir. Aşağıdaki programda komut öncesinde yeralan açıklama satırı, komut hakkında bilgi veriyor.

gunzip komutu dosya acmak icin kullanilir.

gunzip sistem.gz

Yorum satırı, komutun sonuna da eklenebilir.

ps -aux # sistem süreçleri hakkında ayrıntılı bilgi..

Bir kabuk altında çalışırken başka bir kabuk için yazılmış bir programı çalıştırmak mümkündür. Örneğin tcsh altayken ve daha evvel bash kullanarak yazılan bir program çalıştırılmak isteniyorsa, önce bash yazarak kabuk değiştirilmeli, ardından program çalıştırılmalı ve tekrar tcsh'a dönülmelidir. Tüm bunlaı otomatik olarak yapılabilir. Programın en başına `#!` karakteri, ardından programın çalışacağı kabuğun patikası yazılır. Örneğin `#!/bin/bash` komutu programın en üstüne eklenirse bu program bash kabuğu altında çalışacaktır (Özkan,1996; Akerdem ve Usta 2000, Petersen 2001)

3.8.5 Değişkenlerin Kullanımı

Bir değişkene değer atandığı anda sistem tarafından tanınır. Değişkenler alfabetik veya nümerik karakterlerden oluşabilirler fakat bir değişken sayısal bir değer ile başlayamaz. Bunların dışında değişken isminin içinde "_" karakteri de bulunabilir. Bir değişkene değer ataması "=" işareti yardımıyla yapılır.

$ mesaj="akşama yemeğe geliyorum"

İçeriği olan bir değişkene başına "$" işareti konularak ulaşılır. Aşağıda, `echo` komutu yardımıyla bir değişkenin içeriği ekrana basılıyor.

$ echo $mesaj

akşama yemeğe geliyorum

$ echo yarın $mesaj

yarın akşama yemeğe geliyorum

3.8.6 Giriş/Çıkış İşlemleri

Bir kabuk programı çalışırken kullanıcıdan klavye yardımıyla bilgi girmesi sağlanabilir. Bu tür işlemler için tanımlanan `read` komutu klavyeyi okur ve aldığı bilgiyi bir değişkene atar. Aşağıdaki komutları içeren program yardımıyla klavyeden okunan değer ekrana yazılıyor. `echo` komutundan sonra birden fazla değişken grubu veya hem değişken, hem de dizi kullanılabilir.

echo Bir sayi giriniz..

read sayi

echo Girilen sayi : $sayi

Bazı durumlarda girilen değer özel karakterleri içerebilir. Bu durumda istenmeyen bazı sonuçların doğması kaçınılmaz olur.

3.8.7 Aritmetik İşlemler

Bash kabuğunda matematiksel işlemlere büyük sınırlamalar getirilmiştir. Tamsayı değişkeni dışında matematiksel değişken kullanmak için bu işlemler için geliştirilmiş ve kolaylıklar sağlayan `awk` veya `bc` kullanabilir. Aritmetik işlemler için `eval` komutunu veya bash kabuğu altında yerleşik (builtin) komut olan `let` komutunu kullanabilirsiniz. Aşağıda let komutunun kullanımı görülüyor.

$ let "degisken=aritmetik islem"

Bu örnekte iki sayı çarpılıp çıkan sonuç başka bir değişkene yazılıyor.

$ let "carpim=2*7"

$ echo $carpim

Aritmetik değişken tanımlamanın diğer bir yolu da `typeset` komutu kullanmaktır.

$ typeset -i sonuc (sonuc değiskeni bir doğal sayi içerecek)

$ a=100 ; b=56 (iki komutu ayırmak için ; kullanılabilir)

$ sonuc=a*b

$ echo $sonuc

5600

3.8.8 if-else Kalıbı ve Kontrol İşlemleri

Hemen her programlama dilinde olan `if` kalıbı bir Linux komutunun çalışmasını kontrol (`test`) eder. `if` komutu yerleşik bir komuttur. `if` komutunun ardından gelen Linux komutu çalıştırılır ve komutun çıkış durumu (exit status) gözönüne alınarak ardından gelen `then` deyimiyle birlikte devamı işletilir. Genellikle komutun iki türlü çıkış durumu olacağından `else` komutunun ardından gelen komut zinciri, diğer çıkış durumunda çalıştırılır. Her `if`, bir `fi` komutuyla bitmelidir. Bir if komutu içerisinde başka bir if komutu da kullanmak olasıdır. Aşağıda if-then-else komutunun örnek sözdizimi görülüyor.

```
if linux komutu
  then
    komut1
    komut2
    ...
  else
    komut1
    komut2
    ...
fi
```

`if` komutu genellikle kendine `test` komutu ile birlikte kullanım bulur. Bu komut yardımıyla mantıksal işlemler yapılabilir, sayılar ve hatta diziler karşılaştırılabilir. Anahtar sözcük olan `test`'ten sonra opsiyonlar ve/veya karşılaştırılacak olan değerler yazılır. Her opsiyon bir mantıksal işleme karşılık gelir. Örneğin `-lt` opsiyonu ilk girilen aritmetik değişkenin ikinci değerden küçük olup olmadığını denetler. Benzer şekilde `=` opsiyonu da iki karakter kümesinin eşitliğini kontrol eder. Aşağıda `test` komutunun örnek kullanımı yeralıyor.

```
$ test 5 -eq 3
$ a="linux"
$ test $a="linux"
```

komutun işletilmesinin ardından kabuğa bir değer döndürülür. Bu değer komut başarılı olarak işletilmişse 0, değilse 1'dir. Son çalıştırılan tüm Linux komutlarının çıkış değeri $? değişkeninde tutulur. test komutunun çıkış değeri de bu yolla öğrenilebilir.

```
$ sayi=4
$ test $sayi -eq 4
$ echo $?
```

0

$ test $sayi -lt 2

$ echo $?

1

`test` komutu yerine parantezler de kullanılabilir. Yukarıdaki iki örnek, parantez kullanılarak şu şekilde yazılabilir:

$ [$sayi -eq 4]

$ [$sayi -lt 12]

Dikkat edilmesi gereken bir nokta, köşeli parantez kullanırken araya boşlukların eklenmesidir. Parantezler başlı başına bir komut olarak görüldüklerinden sağında ve solunda en az bir boşluk bırakılmalıdır. `test` komutunda sıkça kullanılan diğer seçenekler şunlardır:

Çizelge 3.3 Linux test komutu karşılaştırma parametreleri

Aritmetik karşılaştırma	
-gt	büyük
-lt	küçük
-ge	büyük eşit
-le	küçük eşit
-eq	eşit
-ne	eşit değil
Dizisel karşılaştırma	
-z	boş dizi
-n	tanımlı dizi
=	eşit diziler
!=	farklı diziler

Dosya karşılaştırması	
-f	dosya var
-s	dosya boş değil
-r	dosya okunabilir
-w	dosyaya yazılabilir
-x	çalıştırılabilir dosya
-h	sembolik bağlantı
-c	karakter aygıt
-b	blok aygıt
Mantıksal karşılaştırma	
-a	VE
-o	VEYA
!	DEĞİL

3.8.9 Case Kalıbı

Birkaç alternatif arasından seçim yapmak için kullanılan bir komut olan `case`, bir eşleştirme gördüğü anda belirli bir komut kümesini işleme sokar. case yapısı `case` komutu ile başlar, eşleştirilecek olan anahtar sözcük yazılır ve seçenekler alt alta, her seçeneğe ait olan komutlarla birlikte belirtilir. Tüm yapı `esac` komutu ile son bulur.

```
case anahtar-sözcük in
    secenek1)
        komutlar
        ;;
    secenek2)
```

```
        komutlar
        ;;
    *)
        komutlar
        ;;
esac
```

Her seçeneğin son komutu ardından ;; işareti gelmelidir. Seçenekler arasında özel karakterler *(*, [,], ? gibi)* kullanılabilir. Hiçbir eşleme yapılmadığı zaman *) seçeneği değerlendirilecek ve buna bağlı olan komutlar işletilecektir. * kullanımı isteğe bağlıdır. Aşağıda `case` komutuna ilişkin kısa bir örnek veriliyor.

```
#!/bin/bash
clear
echo   "1. ekranı temizle"
echo   "2. sistemdekileri görüntüle"
echo   "3. dizindeki dosyaları göster"
echo -n "Seçeneği giriniz : "
read secenek
case $secenek in
    1)
     clear
     ;;
    2)
     w
     ;;
    3)
```

```
    ls -al
    ;;
  *)
    echo Hatali seçenek
esac
```

3.8.10 Döngüler

Diğer tüm programlama dillerinin en büyük gücü olan döngü işlemlerine kabuk altında da izin veriliyor. `while` ve `for`. `while` komutu her döngüde bir denetleme mekanizmasını harekete geçirirken `for` döngüsü bir listenin elemanlarını sırayla seçer.

3.8.10.1 While-Do Döngüsü

Döngü bloğu `while` anahtar kelimesiyle başlar, ardından gelen koşul sağlandığı sürece döngü işletilir. Önce koşulun sağlanıp sağlanmadığına bakılır. Döngüden çıkabilmek için mutlaka döngü içindeki koşul ifadesinin değerini yanlış yapacak bir durum oluşmalıdır, aksi halde sonsuz döngü oluşur.

```
while kosul ifadesi
  do
    komutlar
  done
```

`if` komutuyla birlikte kullanılan `test` komutu, `while` döngüsünde koşul ifadesi olarak da yer alabilir.

Aşağıda 1'den 100'e kadar sayan ve ekrana basan bir döngü görülüyor.

```
#!/bin/bash
deger=0
while [ $deger -lt 100 ]
  do
    deger=$((deger+1))
```

```
    echo $deger
done
```

Yukarıda kullanılan ((ve)) karakterleri arasına matematiksel bir işlem getirilebilir. Bu özellik bash kabuğuna özgüdür.

3.8.10.2 For-Do Döngüsü

Bir liste dahilindeki tüm değerlere sırayla erişimi sağlar. for komutundan sonra yeralan liste sırayla kullanılır ve herbirisi için döngü çalıştırılır. Listenin sonuna gelindiğinde ise döngüden çıkılır.

```
for degisken1 in deger1 deger2 ... degerX
  do
    komutlar
  done
```

Aşağıdaki örnek bu döngüyü kullanarak ekrana bir dizi kelime yazıyor. Döngü boyunca akasya, elma ve visne kelimeleri "agac" değişkenine kopyalanıyor ve her döngüde bu değişkenin içerdiği bilgiler ekrana yazılıyor.

```
for agac in akasya elma visne
  do
    echo $agac
  done
```

for-do döngüsü, dosya isimleri üzerinde yapılan işlemlerde de büyük kolaylıklar sağlar. Bunun için özel karakterlerden yararlanmak da olasıdır. Örnek olarak * karakteri o anki çalışma dizini içindeki tüm dosyaları seçer.

```
for a in * ; do
    file $a
    done
```

3.8.10.3 Until Döngüsü

Döngüye son verme işlemi bir koşula bağlanmak isteniyorsa until komutu tercih edilir. Bu komut,

until komut

do

komutlar

done

3.8.10.4 Örnek Kabuk Programları

Kabuk programları yardımıyla kullanıcı menüleri elde edilebilir. Kullanıcı bu programlar vasıtası ile kabuğa ulaşmadan bir çok işlemi yerine getirebilir.

Ana Program

#menu.sh

kullanıcılar için hazirlanmiş menü programi

trap ":" 2

temizle='tput clear'

while true

do

echo "'$cls"

echo "1) Sistemin Durumu"

echo "2) Mesaj İletimi"

echo "3) İşlem Kontrolü"

echo "4) Diğer Sistemlere Bağlanmak"

echo "5) Çıkış"

echo

echo "Seçiminiz :\c"

```
read secim
case $secim
in
durum.sh
        ;;
mesaj.sh
        ;;
islem.sh
        ;;
bağlan.sh
        ;;
exit 1
        ;;
*) echo "Seçiminiz yanlış.."
bekle.sh
esac
done
```

Yukarıdaki menü programi kullanıcının seçimine göre başka kabuk programlarını çağırmaktadır.

bekle.sh programi : ekrandaki görüntüyü return tuşuna basılana kadar bekletir.

```
# bekle.sh
# return tuşuna basılmasını bekler.
echo "Devam etmek için <return> tuşuna basınız m\c"
stty –echo
```

```
read dur

atty echo

echo
```

durum.sh : Bu program sistemin çok kullanılıcılı durumda olup olmadığını kontrol ederek sonucu ekranda görüntüler.

```
# Sistemin durumunu kontrol eder.

set 'who -r'

if [$7 = 2]

then

echo

echo "Sistem çok kullanıcılı durumda"

echo

who

echo

echo 'who |wc –1' kullanıcı sistemde çalışıyor.

echo

bekle.sh

else

echo "Sistem tek kullanıcılı durumda"

bekle.sh

fi
```

Bu örnekleri çoğaltmak mümkündür. Görüldüğü gibi kabuk programlari, yalnızca Linux komutlarını çalıştırmak ile kalmaz kendine ait komutlar, deyimler ve değişkenler de içerirler.

3.9 X Window Grafik Arabirimi

3.9.1 X Window Nedir?

Açık sistemlerin kullanıcıya sunduğu en büyük özelliklerden biri olan X Window, Linux'un doğduğu andan beri destek görmeye başladı ve internet üzerinde bedava dağıltılmasıyla birlikte Linux dağıtımları içinde bir standart olarak kendine yer edindi. X Window'da kullanılan bir çok pencere yöneticisi vardır. Pencere denetleyicilerin herbirinin farklı özellikileri vardır. X Window istemci sunucu modeline göre çalışır. Ana makine üzerinde çalışan X sunucusu grafik donanımı üzerindeki tüm giriş çıkış yetkilere sahiptir. Bir X istemcisi sunucuya bağlanarak istediği işlemleri sunucuya yaptırır. İstemcinin görevi emir vermek, sunucunun ise bu emri uygulamaktır. Burada bilinmesi gereken en önemli kavram X Window'un ağ tabanlı bir yapıya sahip olmasıdır. Bir istemci ve sunucunun olduğu her yerde ağdan da sözedebiliriz. Kısaca X Window ekranda ki pencerelerin denetiminde söz sahibi olan tek programdır, her pencerenin nasıl görüneceğini, ne kadar büyük olduğunu bilir ve kullanıcı tarafından verilen pencere kapatma , küçültme gibi komutları yerine getirir.

X Window ile rahat çalışmak için en az 486 tabanlı bir bilgisayar, 32 mb RAM ve pentium 200 bilgisayar gereklidir.

3.9.2 X Window Yapılandırması

X Window'un kurulumu ve yapılandırması basittir. Red Hat dağıtımı ile gelen XConfigurator kolayca yönetilen bir arayüze sahiptir. Eğer ekran kartı otomatik olarak tanıtılamazsa SuperProbe komutu ile gerekli sürücü bilgileri alınarak bu bilgiler ışığında ekran kartı tanıtabilr ve görüntü ayarları düzenlenebilir. Eğer kurulum sırasında X Window'un açılışta çalışıp çalışmayacağına ilişkin soru evet ile geçilmişse zaten her açılışta default olarak grafik ekran ile sistem açılır. Aksi takdirde komut satırından startx komut ile de başlatılabilir. X Window altındaki farklı pencere yöneticileri arasında geçiş yapmak için komut satırından switchdesk komutu girilir ve gelen listeden istenilen arayüz seçilir.

3.10 İnternet Sunucuları

Özellikle Internet hizmetleri (Web, FTP, Gopher gibi) konusunda oldukça iddalı olan Linux, bu yönüyle Unix ile Internet'in gelişimi arasındaki yakın ilişkiyi yansıtır. Kullanıcılar, linux ile kendi Web sitelerini oluşturabilir ve diğer kullanıcıların oluşturulan bu sitedeki dosyalara erişimini sağlayabilir. Bu şekilde iş gören bir sisteme "sunucu" adı verilir ve verdiği hizmete göre tanımlanır. Sistem bir Web sunucusu veya FTP sunucusu olarak yapılandırılabilir. Tek

bir Linux sistemi birkaç değişik hizmeti sağlayabilir. Sistem aynı anda Web, FTP, Gopher ve WAIS sunucusu da olabilir. Kullanıcılardan biri sistemden dosya indirirken bir diğeri Web sayfalarında gezinti yapabilir. Tek yapılması gereken Linux sistemine her hizmet için gerekli yazılımın kurulması ve çalıştırılmasıdır. Bu sunucular her biri sunduğu kendi özel hizmetleri için uzak kullanıcılardan talepler bekleyen, sürekli çalışan kanallar gibi iş görür.

Red Hat ve OpenLinux sistemleri Web ve FTP sunucularını yükler ve çalıştırır. Bu ürünler internet sunucuları akılda tutularak tasarlanmıştır. Bu Linux dağıtımlarından biri yüklendiğinde bu hizmetlere ait yazılımlarda otomatik olarak yüklenirler ve yapılandırılırlar. Tek yapılması gereken sistem için özelleştirilecek ayarların değiştirilmesidir. Sistem her başlatıldığında bu sunucular otomatik olarak başlatılırlar.

3.10.1 Sunucuları Başlatmak

Bir sunucu, diğer programlarla birlikte çalışan ve sürekli olarak sunduğu hizmetler için sistemdeki diğer kullanıcılardan veya sisteme bir ağ üzerinden bağlanan uzaktaki kullanıcılardan gelen talepleri izleyen bir kanaldır. Bir kullanıcıdan bir talep aldığında, hizmetlerini sunmak için bir oturum başlatır. Örneğin, eğer kullanıcılar sistemden bir dosya indirmek istiyorlarsa, FTP sunucusundan bir oturum açmasını talep etmek için kendi FTP istemcilerini kullanırlar. Oturumda, sisteme erişebilir ve istedikleri dosyayı indirebilirler.

Bir sunucuyu başlatmanın birkaç yöntemi vardır. Bunlardan biri komut satırına sunucu programının ismini ve argümanlarını yazarak sunucuyu manuel başlatmaktır. Tekrar komut satırı görüntülenmesine rağmen sunucu çalışmaya başlamış olacaktır. Sunucunun çalışıp çalışmadığını görmek amacıyla, sistemde çalışan işlemleri gösteren aşağıdaki komutlar kullanılabilir. (Çetin ve Çelik,2000)

ps –aux

ps – aux | grep 'httpd'

Sunucuyu manuel başlatmak ve durdurmak için özel başlangıç programcıkları da kullanılabilir. Örneğin start parametresi ile birlikte kullanılan /etc/rc.d/init.d/httpd programcığı gibi. stop parametresiyle bu program sunucuyu durdurabilir.

Sistem her başlatıldığında bütün sunucuları manuel başlatmak yerine otomatik olarak da başlatılabilir. Sunucunun nasıl kullanılmak istendiğine bağlı olarak bunu yapmanın iki yolu vardır. Birinci yöntem sunucu programın sistem açılır açılmaz devreye girmesi ve kapanana kadar çalıştırılmasıdır. İkinci yöntem ise sunucunun bir kullanıcıdan talep geldiğinde

çalışmasıdır. Burada seçilecek yöntem sunucunun ne sıklıkla kullanıldığına bağlıdır. Çok kullanılıyorsa birinci yöntem, az kullanılıyorsa ikinci yöntem kullanılmalıdır. Otomatik olarak başlatılan ve sürekli olarak çalıştırılan bir sunucu yalnız sunucu olarak isimlendirilir. Çoğu dağıtım, sistem her başlatıldığında sunucuları otomatik olarak başlatır. Bu prosedür /etc/rc.d/init.d dizininde yer alan sunucular için özel başlangıç programcıklarını kullanır. Alternatif olarak, rc.local gibi sistem başlangıç programcıklarındaki bir sunucu doğrudan da çağrılabilir.

Bir sunucunun sadece bir hizmet talebi geldiğinde başlatılması için söz konusu sunucu "Internet Süper Sunucusu" olarak bilinen inetd kanalını kullanarak yapılandırılmalıdır. Bu kanal sunucu taleplerine bakar ve bir talep geldiğinde sunucuyu başlatır. Yeni arayüzler kanalların kolaylıkla yönetilmesine imkan veren GUI arabirimine sahiptir.

3.10.2 FTP Sunucusu

FTP (File Transfer Protocol-Dosya Transfer Protokolü) bir makineden diğerine yapılan dosya transferinde en çok kullanılan yollardan biridir. Internet desteği olan işletim sistemlerinin hepsinde FTP standart program olarak kullanıcıların hizmetine sunulmuş durumdadır. Linux altında da FTP desteği vardır. Temel FTP komutları:

# ftp gelecek.com.tr	(Bir ftp sitesine bağlanma)
ftp>help	(Kullanılabilecek komut listesi)
ftp>dir	(Dizindeki dosya listesi)
ftp>cd pub	(Dizin değiştirme)
ftp>lcd /root/gimp	(Yerel dizini belirtme)
ftp>get dosya	(Dosyaları alma)
ftp>put dosya	(Dosyaları gönderme)
ftp>quit	(Çıkış)

Linux sistemlerinde kullanılabilecek birkaç FTP Daemon'u mevcuttur. Linux dağıtımlarının büyük bir kısmı, wu-ftp olarak isimlendirilen FTP sunucusu ile birlikte gelir. Ayrıca Pro FTPD de Linux'ta kullanılabilen ve Apache web sunucusu tasarımını esas alan popüler bir FTP sunucusudur. (Çetin ve Çelik,2000)

3.10.2.1 Proftpd FTP Server (FTP Sunucu)

Proftpd yazılımı www.proftpd.net adresinden indirilebilir. Bu yazılım hem tar.gz, hem RPM, hem de deb formatında indirilebilir. Yapılması gereken işlemler oldukça basittir. Bu adresden RPM paketleri indirilir. Alınması gereken iki RPM paketi vardır. Bir tanesi, ana proftpd dosyalarının bulunduğu "core" paketi diğeri de isteğe bağlı olarak alınabilecek "standalone" ya da "inetd" paketleri.

Proftp programı iki şekilde çalışabilir, birincisi doğrudan doğruya inetd tarafından kontrol edilebilir şekilde, diğeri de "standalone" adı verilen, ve proftp'nin kendi kendine çalıştığı bir program olacak şekilde. İlk seçenekte sistemde proftp ile ilgili bir program sistemde çalışmaz. inetd kullanıcı taleplerini izler ve bir talep geldiği zaman proftp programını çalıştırır. İkinci seçenekte ise proftp programı sürekli arka planda çalışır ve ftp portuna gelen istekleri değerlendirir.

Proftp programının kurulumuna başlamadan önce dağıtımlarla birlikte gelen wu-ftpd programının devre dışı bırakılması gerekir. Çünkü farklı iki servis aynı portu kontrol edemez. wu-ftp'yi kapatmak için /etc/inetd.conf dosyasındaki ftp ile başlayan satırın başına # karakteri konmalıdır. Daha sonra inet servisi "/etc/rc.d/inet.d/inet restart" komutuyla tekrar çalıştırılmalıdır. Alınan RPM paketleri proftp programının sürekli çalışması şeklinde ayarlanmıştır. Bu RPM paketleri açıldıktan sonra hiçbir ayar değişikliğine gitmeden ftp servisi çalışmaya başlayacaktır. (Çetin ve Çelik,2000)

3.10.3 Apache Web Server (Web Sunucu)

Apache dünyada en çok kullanılan web sunucuya verilen isim, modüler bir yapıya sahip. Açılış sırasında istenilen özellikler eklenebilir veya istenmeyen özellikler etkisiz hale getirilebilir. Apache kısaca 80 nolu porta gelen web isteklerine karşılık vererek sunucuya daha önceden yerleştirilmiş HTML dosyalarını istemciye gönderen yazılımdır (Petersen,2001). Red Hat ve OpenLinux dahil olmak üzere birçok Linux dağıtımı gereken bütün dizinler ve konfigürasyon dosyaları ile birlikte Apache Web Sunucusunu sisteme otomatik yükler. Sistem her başlatıldığında Web sunucusu da devreye girer ve çalışır. Web sitesi veri dosyaları için ayrılan dizin /home/httpd/html dizinidir. Yapılması gereken Web dosyalarının bu dizine veya bu dizinin alt dizinlerine yerleştirilmesidir. Daha sonra gereken ağ sunucu konfigürasyonlarını düzenlemek ve uzaktaki kullanıcılara açık olan dosyaları ve dizinleri belirlemektir. Apache'nin sistemde kurulu olup olmadığını anlamak için $ rpm –qi apache komutu kullanılabilir. Bu komut ile sistemede bu yazılım varsa bu yazılım hakkındaki

ad,versiyon, sürüm,..vs bilgiler alınır. Sistemde yoksa orijinal CD üzerinden veya internetten bulunabilir. Apache paketleri sisteme kurulduktan sonra birtakım dizinlere kendi yapılandırma dosyalarını,httpd programını ve kılavuz sayfalarını yerleştirir.

/etc/httpd: Bu dizin normalde sadece yapılandırma dosyalarını içerir. Eski sürümlerde üç farklı yapılandırma dosyası varken yeni sürümlerde bire indirilmiştir. Böylece sadece httpd.conf üzerinde yapılacak değişikliklerle düzgün ve yüksek performans elde edilebilmektedir.

/home/httpd: Bu dizin cgi-bin,html ve icons dizinini içerir. cgi-bin dizini sunucu tarafında çalışacak programcıklar içindir. HTML dizini kullanıcıdan gelen istekleri karşılayacak dosyaların bulunduğu dizindir. Sunucuya ilk bağlanıldığında bakılacak olan dosya /home/httpd/html/index.html olacaktır. Bundan dolayı sitenin ana sayfası bu isimde olmalıdır. icons dizininde ise Apache'nin kullandığı birkaç simge dosyası yer alır.

/home/httpd/html/manuel: Bu dizinde Apache'nin kullanımıyla ilgili referanslar yer alır.

/var/log/httpd: Dosya sistem hiyerarşi standardına göre tüm kayıt dosyaları Linuxta /var/log altına yerleştirilir.

/etc/dc.d/init.d/httpd: Apache'nin çalıştırılması için kullanılan programdır. Ayarlar değiştirildiği zaman Apache'nin bu program vasıtası ile tekrar çalıştırılması gerekmektedir.

Apache'nin çalıştırılması	/etc/rc.d/init.d/httpd start
Apache'nin durdurulması	/etc/rc.d/init.d/httpd stop
Apache'nin durumunun öğrenilmesi	/etc/rc.d/init.d/httpd status
Apache'nin durdurulup yeniden çalıştırılması	/etc/rc.d/init.d/httpd restart
Apache'nin yapılandırma değişikliği	/etc/rc.d/init.d/httpd reload

4. HTML İLE WEB DÖKÜMANLARI HAZIRLAMA

4.1 Anlamlandırma Dilleri

Basılı yayının gelişmesiyle yayıncıların yazılı metinlerin baskı makinelerinde farklı bir şekilde/biçimde yayınlanması için hazırladıkları notlar ve özel semboller 'anlamlandırma' (markup) olarak ifade edilmekteydi. Bu durum, metnin belirli kısımlarının özel bir anlam kazandırmak üzere işaretlenmesi işlemidir. Bu amaçla kullanılan işaretler, kurallar ve gramer kümesi 'anlamlandırma dili' (markup language) şeklinde tanımlanır. Bu tip işaretleme yapılarını birçok ortamda görebiliriz. Örneğin kelime işlem programları bir metnin içerdiği kısımları, yazı tiplerini, font ve stillerini ayırmak için metin içine gömülü bir çok işaret yerleştirir. Metin ve belgelerin kolay bir şekilde taşınabilmesi, paylaşılabilmesi ve işlenebilmesi için ilk işaretleme dili olan GML-Genelleştirilmiş Anlamlandırma dili, 1960 sonlarında IBM'de yapılan araştırmalar sonunda ortaya çıktı. GML daha sonra ANSI'de (American National Standard Institute) oluşturulan bir grup tarafından geliştirilerek SGML adı altında 1986 yılında uluslararası bir standart olarak kabul edildi.

Dökümanların yapısı büyük ölçüde dökümanların görünüşünü hazırlamak için kullanılan dile bağlıdır. Bazı ileri diller dökümanın görünümü ile ilgili oldukça zengin kontrol imkanları sunmaktadır. Diğer diller temel özellikleri sağlarlar ve ileri özellikler yerine kullanım kolaylığı ve sempati sunarlar. Aşağıda yaygın olarak kullanılan diller yer almaktadır. (Yurt vd., 2000; Bahar ve Türkmen 1998)

- Standart Genelleştirilmiş Anlamlandırma Dili (SGML)
- Sanal Gerçeklik Modelleme Dili (VRML)
- Sayfa Tanımlama Dilleri
- Hipermetin Anlamlandırma Dili (HTML)
- Genişleyebilir İşaretleme Dili (XML)

4.1.1 SGML (Standart Genelleştirilmiş Anlamlandırma Dili)

Web dökümanlarının çoğunun yapısı Standart Genelleştirilmiş Anlamlandırma Dili'ne dayanan bir dili kullanmaktadır. SGML standart metinlerle açıklanan ve genelleştirilmiş anlamlandırma kullanan karmaşık dökümanları paylaşmak için yöntemler tanımlar. Karmaşık yapıları düz metinlerle tanımlamak dökümanın her tipte bilgisayara dağıtılabilmesini sağlar ve

formatı anlamlandırma ismi verilen ve insanlar tarafından okunabilen bir yapıda sunar. Anlamlandırmada standart karakterler kullanıldığı için herhangi bir kişi ayrıca bir yazılım kullanmaya gerek kalmadan dökümanları anlamlandırma dilinde hazırlayabilir.

SGML birkaç sınırlaması olan ileri bir dildir. SGML'de metin ve görüntülerin yerleştirilmesi konusunda tam kontrol yetkisi sunulmaktadır. Yani kullanıcının inceleyicisi, görüntüleri ve metinleri tam olarak tasarladığınız yerde çıkarırlar. SGML güçlü bir anlamlandırma dili olmasına rağmen , Web'de yaygın olarak kullanılmaz. Bununla birlikte, daha çok sayıda yayımcı SGML'in yeteneklerini keşfettikçe bu gerçek değişmektedir. SGML'in temelini oluşturduğu anlamlandırma dillerinden en önemli iki tanesi Sanal Gerçeklik Modelleme Dili (VRML) ve Hipermetin Anlamlandırma Dili (HTML) dir.

4.1.2 VRML (Sanal Gerçeklik Modelleme Dili)

Web üstündeki teknoloji büyük bir hızla gelişmektedir. Son gelişmelerden biride VRML'dir. VRML, standartlaştırılmış anlamlandırma dilini kullanarak karmaşık modeller ve çok boyutlu dökümanlar hazırlanmasına olanak sağlar. Web yayımcıları için sanal gerçekliğin göstergeleri sınırları aşmaktadır. VRML kullanarak MB'lik disk alanını kullanacak olan hesaplar ve veriler birkaç yüz satırlık anlamlandırma koduna dönüştürülebilir. Böylelikle VRML dosyalarının bilgisayara alınma süresi azaltılmış ve ağın bant genişliği etkin kullanılmış olur. Ayrıca karmaşık bir mod anlaşılabilir ve okunabilir bir formata dönüşmüş olur. VRML' de internette yaygın olarak kullanılmamaktadır.

4.1.3 Sayfa Tanımlama Dilleri

Bazı Web dökümanları anlamlandırma dilleri yerine sayfa tanımlama dilleri kullanılarak formatlanırlar. Sayfa tanımlama dilleri çoğunlukla Adobe Acrobat veya Corel VENTURA gibi ticari sayfaların görünüm uygulamalarına özel formatları kullanırlar. Sayfa görünümü uygulamaları sempatik grafik arabirimleri ve sayfa görünümü üzerinde çok sayıda kontrol olanağı sağladıkları için popülerdir. Bu uygulamaların kullandığı formatlar tescilli olmakla birlikte bu formatlar SGML tarafından açıklanan standartları temel almaktadır.

4.1.4 HTML (Hipermetin Anlamlandırma Dili)

HTML en yaygın olarak kullanılan anlamlandırma dilidir. HTML popülerliği, kullanım kolaylığı ve sempatikliği sayesinde daha da kök salmaktadır. HTML ile çabucak ve kolaylıkla Web dökümanları oluşturulabilir ve bu dökümanlar geniş bir okuyucu kitlesine sunulabilir. HTML, metin ve görüntülerin görsel nitelikleri ve hatta metinlerin stilleri –koyu, alt çizgili,

eğik- olarak belirlenebilir. HTML'in font tipini belirleyebileceğiniz eklentileri de vardır ancak standart HTML özellikleri bu yeteneklere sahip değildir.

HMTL'de ileri görünüm kontrollerinin çoğu kullanılmadığı halde Web merkezlerinin büyük bir kısmında tercih edilen yayım dili HTML'dir. Sınırlamalar HTML'in karmaşılığını önemli ölçüde azaltmak için bir yoldur. HTML'in değişik sürümleri bulunmaktadır. Her yeni sürüm daha fazla yetenek ve işlev sunmaktadır. Bu sürümlere ek olarak internet geliştiricilerinin bir kısmı HTML için eklentiler oluşturmuşlardır. Eklentiler standart olmadığı halde çoğu Web yayımcıları tarafından kabul görmüştür. Örneğin Netscape'in bazı eklentileri o kadar popülerdir ki sanki HTML standart özellikleri gibidir.

4.1.5 XML (Genişleyebilir Anlamlandırma Dili)

XML de HTML gibi işaretleme etiketlerini kullanan bir dildir. HTML ve XML arasındaki temel fark XML işaretleme etiketlerinin bilginin içeriğini tanımlamak amacıyla kullanılmasıdır. XML bir meta dildir. Diğer bir deyişle diğer yeni işaretleme dillerini tanımlamak için kullanılan bir dildir. XML ile herhangi bir uygulama için XML belgesinin içinde bulunacak verinin içeriği ve içerdiği veri tiplerini tanımlayacak uygulamaya özel bir işaretleme dili tanımlanabilir. Meta Veri, veri hakkındaki bilgidir. XML etiketleri veri hakkındaki meta bilgiyi tanımlamaktadır.

<CONTACT>

<NAME> Frank Rizzo </NAME>

<ADDRESS>1212 W 304TH Street</ADDRESS>

<CITY>New York</CITY>

<STATE>New York</STATE>

<ZIP>10011</ZIP>

<PHONE>

<VOICE>212-555-1212</VOICE>

</PHONE>

</CONTACT>

Örnekte görüldüğü üzere XML belgesi, içindeki verinin içeriğini tanımlayan etiketler

içermektedir. Belgenin web tarayıcılarında nasıl formatlanacağı konusunda bilgi içermez. Belgenin formatlanması CSS veya XSL teknolojileri ile yapılır. Tabi bir belge içerisinde herhangi bir etiket ismi verip bunu kullanmak mümkün değildir. Bir uygulamadaki XML belgesinin yapısı ve içereceği etiketler o uygulama için geliştirilmiş olan özel işaretleme dili ile tanımlanır. Bu işaretleme dilinin yapısı ise XML DTD (Doküman Tip Tanımlama) veya XML Schema olarak adlandırılan belge tanımlama dosyasında belirtilir.

<!ELEMENT contact (name, adress+,city,state,zip,phone)

<!ELEMENT name (#PCDATA)>

.....

<!ELEMENT phone (voice)>

<!ELEMENT voice (#PCDATA)>

XML dili, XML sözdizim kuralları, XML DTD,XML Schema, XML Namespaces, Xpath, Xpointer, XSL, XSLT, XSLF, SAX ve DOM gibi birçok belirtim ve teknolojinin birleşiminden oluşmaktadır. XML hızla gelişen ve üzerinde bir çok araştırma, geliştirme çalışmaları yürütülen bir dildir. Bir çok araştırmacı ve yazılım geliştirici XML'in yazılım endüstrisinde radikal değişikliklere sebep olacağına inanmakta ve XML'i elektronik veri değişiminin yeni ASCII standardı olarak kabul etmektedir. Bir çok lider firma, XML ve XML uygulama standartlarını desteklemekte ve XML tabanlı yeni ürünlerini bilgi teknolojisi uygulamalarının hizmetine sokmaktadır.

4.2 HTML Dökümanı Oluşturma

HTML Dökümanlarının biçinmlendirilmesi, im (tag) adı verilen anlamlandırma kodlarına dayanır. İmler dökümanın yapısını tanımlar ve parantez içinde bulunan bir eleman adı içerirler; birinci seviye başlığın başlangıcını gösteren <H1> gibi. HTML büyük küçük harfe duyarlı değildir; <h1> ve <H1> aynı anlama gelir. İmlerin çoğu çiftler halinde kullanılır. Başlangıç imi olarak adlandırılan im, inceleyiciye bir doküman elemanının başladığını bildirir. Bitiş imi olarak adlandırılan başka bir im, inceleyiciye bir elemanın sonlandığını bildirir. Başlangıç ve bitiş imi arasındaki tek fark, bitiş iminde eleman adından önce sağa eğik bir çizginin bulunmasıdır. Örneğin başlangıç imi <H1> ile bitiş imi </H1> bir çift oluşturur. HTML'de gösterilmek üzere özel bir karakter tanımlayabilirsiniz. Özel karakterler, önünde ampersand ve sonunda noktalı virgül bulunan bir eleman adı ile tanımlanır. Örneğin

& ampersand sembolünü tanımlar.

4.2.1 Doküman Yapısını Tanımlama

Her HTML dökümanı, <HTML> anlamlandırma imi ile başlamalı ve </HTML> anlamlandırma imi ile sonlanmalıdır. Başlangıç imi <HTML>, inceleyiciye dökümanın HTML formatlı bir doküman olduğunu belirtir ve dökümanın başlangıcına işaret eder. Bitiş imi </HTML>, dökümanın sonuna işaret eder ve herhangi bir HTML dökümanında daima son öğedir. Bu imleri taşımayan HTML dökümanları olabilir fakat bu imleri kullanmamak başarısız bir tasarımdır. Çünkü HTML tek anlamlandırma dili değildir, doküman HTML olarak tanımlanmadığında okuyucunun inceleyicisi şaşırabilir.

Her HTML dökümanın bir başlık ve bir gövdesi olmalıdır. Başlık ilk <HTML> imini izler. HTML başlığı, dökümanın başlığı gibi anahtar görüntüleri belirler. Başlığın başlangıcı <HEAD> imi ile belirtilir. Başlığın sonu </HEAD> imi ile belirtilir. Dökümanın başlığı takip eden ana bölümüne gövde denir. Gövde, okuyucunun inceleyecisinin görüntülenmesini istediğiniz metin ve nesneleri içerir. Başlangıç imi <BODY> ve bitiş imi </BODY> dir. Başlık bölümü öncelikle hizmet birimlerine doküman hakkındaki bilgileri sağlamak için kullanılır. Başlık bölümündeki her şey başlangıç ve bitiş başlık etiketi arasında yer alır. Aşağıdaki HTML etiketleri başlık için ayrılmıştır.

TITLE : En çok kullanılan başlık imi, <TITLE> imidir. Doküman başlığının başlangıcını gösterir. Bitiş doküman başlığı imi </TITLE> dır. Her dökümanın sadece bir başlığı olabilir. Kısıtlı sayıda karakter gösterebildiğinden başlıklar kısa ama açıklayıcı olmalıdır, dökümanın konusuna ve içeriğine ait yeterli bilgi sağlamalıdır.

BASE : Normalde, bağıl (relatif) dosya yolu kullanarak yerel bir web hizmet birimindeki dosyalara erişilebilir. Bir dosya yerleştirilirken bağıl dosya yolu kullanıldıysa, bu dosya o anda açık olan dosyaya bağıl olarak yerleştirmiş olur. Bu bağıl yolları kullanmanın normal şekli olmakla birlikte tüm bağıl bağlantılar için <BASE> imi kullanılarak temel bir yol tanımlanabilir. <BASE> imi sadece <HEAD> elemanın içinde yer alabilir ve tek geçerli özniteliği (attribute) HREF'dir. Temel yol şu şekilde tanımlanır.

<BASE HREF=http://tvp.com/> Bu tanımlama inceleyiciye dosyadaki tüm bağıl yolların başına bu adresi eklemesini bildirir. Bu temel yol resimlerin yolları da dahil tüm bağıl yollara eklenir.

ISINDEX : Bu im, dökümanın etkileşimli olarak aranabilmesi için kullanılır. Bir ISINDEX

sorgulamasının başlatılması için kullanıcının <ISINDEX> imi içeren bir HTML dökümanı yaratan geçit bilgisayarı programcığına erişmesi gerekmektedir. Kullanıcı, sorgulama tarafından istenen bilgiyi girdiğinde, orjinal programcığın yolunu ve kullanıcının girdiği bilgiyi içeren özel bir URL, işleme girmesi için geçit bilgisayarı programcığına gönderilir. Geçit bilgisayarlarının hizmet birimlerine bilgiyi nasıl ilettiklerini tanımlayan yazılıma Ortak Geçit Bilgisayarı Arabirimi (CGI-Common Gateway Interface) adı verilir. CGI , HTML dökümanlarının geçit bilgisayarı programcıkları adı verilen dış programları çağırabilmesine imkan tanır. CGI, etkileşimli formlar ve görüntü haritaları yaratmak için gerekli temeli sağlar.

LINK : <HEAD > elemanı içinde, doküman ve diğer nesneler arasında ilişkileri belirleyen özel bağlantılar tanımlama imkanı verir. Döküman birden fazla <LINK> elemanı içerebilir. HREF özniteliği <LINK> imi ile kullanılması zorunlu olan tek özniteliktir. HREF özniteliği kullanılarak, bir hipermetin referansı tanımlanabilir.

<LINK HREF="wphp.html">

META : <META> imi kullanılarak dökümanın <HEAD> elemanı içerisinde özel veya fazladan bilgi gönderebilirsiniz. Dökümanı alan hizmet birimi, istemcinin kullanması için cevap başlığına bu bilgiyi katar. Content, HTTP-EQUIV, Name adında üç özniteliği vardır.

<META HTTP-EQUIV="Son tarih" CONTENT="31 Aralık 1998">

web hizmet birimi bunu cevap başlığına şu şekilde ekler.

Son tarih:31 Aralık 1998

Doküman birden fazla <META> bilgi içerebilir.

NEXTID : Bu im sadece tanımlayıcılar yaratan HTML editörleri tarafından kullanılır. Editör <NEXTID> imini kullanarak bir elemana atanacak olan bir sonraki ID değerini izler. N şeklinde tek özniteliği vardır. İnceleyiciler bu imi kullanmazlar.

4.2.2 HTML Gövdesinin Tasarlanması

Bir web dökümanının ana bölümü gövdedir. Gövdedeki herşey gövde başlangıç ve bitiş imleri arasında yer alır. Gövde içinde düzinelerce HTML imi yer alabilir. Döküman tanımlarken, gövde bölümüne dahil edilecek elemanlara dikkat edilmelidir. İyi tasarlanmış dökümanlar zorlanmadan okunabilir ve etkileri tasarımlarının basitliğine dayanır. Bir dokümanın yapısını güçlü fakat karmaşık olmayan şekilde tasarlamak için birçok teknik kullanılabilir. Sayfadaki boşluk, dökümanın daha kolay okunmasını ve okuyucunun ilgisinin fikirlere odaklanmasını

sağlayabilir. Vurguyu yaratan ve okuyucunun ilgisini çeken malzemenin dağılımıdır. Beyaz bir sayfa yaratılmasına yardımcı olacak iki anahtar öğe paragraflar ve başlıklardır. Renkler dökümana eklenebilecek bir diğer anahtar özelliktir. Çoğu zaman bir sayfayı renklendirmenin en iyi yolu grafik görüntü kullanmaktır. Stratejik noktalara yerleştirilmiş birkaç resim, sayfanın etkisini önemli ölçüde arttırabilir. Temel olarak HTML dökümanın en önemi öğeleri, başlıklar ve paragraflardır.

Başlıklar : Başlıklar kullanılarak fikirler daha iyi düzenlenebilir. Genellikle başlıklar normal yazı karakterinden daha büyük ve farklı yazı tipinde olurlar. HTML,<H1>'den,<H6>'ya kadar altı düzeyde başlık yaratma imkanı verir. HTML karakter büyüklükleri üzerinde doğrudan bir kontrol yoktur. Karakterin büyüklüğü, dökümanı inceleyeci yazılımın başlangıç konfigürasyonuna bağlıdır. Başlık imleri <H1>..</H1> şeklinde çiftler halinde kullanılır.

Paragraflar : Dökümanı paragraflara ayırmak için paragraf imi <P> kullanılır. </P> şeklinde paragraf bitiş imi vardır ama kulanımı zorunlu değildir.

Açıklamalar : Programcılar genelde yazdıkları program kodları arasına kodla ilgili açıklamalar yapmaktadır. Bu kodu görecek kişilere veya programcının kendisine kodu anlamakta kolaylık sağlamak amacıyla yazılmaktadır. HTML dökümanlarınada benzer şekilde değişiklikler veya eklemeler yapılması gerektiğinde, zaman kazanmak ve hata payını azaltmak amcıyla açıklamalar yerleştirilebilmektedir. Açıklamalar sadece kaynak görüntülendiğinde okunabilir, metinler ve resimlerle gösterilmez.

Sadece paragraflar ve başlıklar içeren bir web dökümanı sıkıcı olacaktır. Genellikle dökümanın ana bölümleri vurgulanmak istenir. Bunu gerçekleştirmek için, karakter stil imleri adı verilen bir dizi HTML imi kullanılır. Karakter stil imleri sayesinde, metinde koyu ve yatay karekterler kullanarak istenilen noktalar vurgulanabilir. HTML'de karekter stil imleri iki bölümde incelenebilir: Fiziksel stiller ve mantıksal stiller.

Fiziksel stiller inceleyiciye göstereceği formatı anlatır.

<B> Koyu font </B>

<I> Italik font </I>

<U> Altı çizgili font </U>

<TT> Daktilo fontu veya eş genişlikli font </TT>

Mantıksal stil imleri kesin bir format belirtmez. Sadece inceleyiciye metnin nasıl

kullanılacağını belirtir ve inceleyiciye özgü konfigürasyona göre metnin gösterilmesini sağlar.

<EM> Vurgu imi </EM>

<STRONG>Güçlü vurgu imi</STRONG>

<CITE>Başka bir metinden aktarma imi</CITE>

<CODE>Bilgisayar kodu veya program örneği imi</CODE>

<KBD>Klavyeden girilen karakter stili imi</KBD>

<SAMP>Gerçeğe uygun karakterlerden oluşan örnek imi</SAMP>

<VAR>Değişken isimleri imi</VAR>

4.2.3 Bağlantılar

Bulunulan HTML sayfasından bir başka sayfaya bağlantı kurabilmek amacıyla <A> etiketi kullanılır.

"HREF" parametresiyle birlikte başka bir dokümana veya sabitlemeye bağlantı yaratır.

"NAME" parametresiyle bağlanabilecek bir sabitleme yaratır.

HREF ="..." Bu dokümana bağlanabilecek doküman URL'si.

NAME ="...." Sabitlemenin adı.

<A HREF = "http://www.yildiz.edu.tr">Buraya tıklayınız!</A>

4.2.4 Metin Hizalama

Bir metin içerisindeki satırları, paragrafları ve blokları listelemek veya belli bir düzene sokmak amacıyla <DIV> etiketi ile birlikte "align" parametresi kullanılır. <DIV> etiketi üzerinde işlem yapılacak metin parçasını diğer bölümlerden ayırır. Hizalamayı dokümandaki bütün paragraflara uygulamak için BODY elemanının sonlanmasından önce ayırma bitiş etiketi </DIV>'i yerleştirmek gerekir. "align" parametresi ile birlikte kullanılan "right, left, center, justify" seçenekleri ise ayrılan metnin hangi biçimde görüntüleneceğini belirler. Aşağıda bu özellikleri içeren bir HTML program parçası görülmektedir.

<DIV ALIGN= "center">Yıldız Teknik Üniversitesi</DIV>

HTML'de listeleme üç şekilde yapılmaktadır. Bunlardan birincisi numaralandırma yaparak

oluşturulan listelerdir. Numaralı listeler aynı zamanda sıralanmış listeler olarak da adlandırılırlar. Bu liste için, ayrılmış HTML etiketi, Ordered List (sıralanmış liste)'nin kısaltılması olan <OL>'dir. Tanım listesi etiketi çift olarak kullanılırlar, başlangıç etiketi <OL>, bitiş etiketi </OL>'dir. Sıralanmış listedeki her elemandan önce <LI> etiketi kullanılır. Sıralanmış listedeki her eleman, numaralandırılmış veya harflendirilmiştir. Harfler sadece iç içe listelerde kullanılır. Bir inceleyici, liste başlangıç etiketi <LI>'yi gördüğünde; yeni bir satıra atlar, liste elemanının metnini içerden başlatır, liste elemanının önüne uygun harfi veya numarayı koyar.

İkinci yöntem noktalı listelerdir. Noktalı listeler, belli bir sırası olmayan amaçları, konuları ve fikirleri listelemek için kullanılırlar. Bu tip listeye ait HTML etiketi <UL> etiketidir, UL Unordered List (düzenlenmemiş liste) kelimelerinin baş harflerini temsil eder. Noktalı liste etiketleri çiftler halinde kullanılırlar. Noktalı liste başlangıç etiketi <UL>; bitiş etiketi </UL>'dir. Listedeki maddeler için <LI> etiketi kullanılır. <LI> etiketi başlangıç ve bitiş etiketi ile kullanılabileceği halde, bitiş etiketi </LI> genelde gerekli olmamaktadır.

Son yöntem ise tanım listeleridir. Küçük sözcükler, tanım listeleri olarak da adlandırılırlar. Bu liste için ayrılmış HTML etiketi, Definition List (tanım listesi)'nin kısaltılması olan <DL>'dir. Tanım listesi etiketleri çift olarak kullanılırlar. Başlangıç etiketi <DL>, bitiş etiketi </DL>'dir. Tanım listesindeki her elemanın iki ögesi vardır:

1) Tanım başlığı adı verilen bir anahtar kelime

2) Tanım verisi adı verilen bir anahtar kelime

Tanım başlığı etiketi <DT>, küçük sözcük terimini veya kullanıcının tanımladığı anahtar kelimeyi belirler. Tanım verisi etiketi <DD>, küçük sözcük terimiyle veya anahtar kelimeyle ilgili tanımı belirler. Eğer terim birden fazla tanıma sahipse birden fazla tanım verisi etiketi kullanılabilir.

4.2.5 Metin Biçimlendirme ve Karakter Türü Denetimi

Görüntülenecek metnin biçimi, kullanılacak karakterlerin büyüklüğü, çeşidi ve rengi HTML etiketlerinin yardımıyla denetlenebilir. Bir dokümanı yaratırken, bazı metin bölümlerinin onu çevreleyen metinden daha büyük olması istenirse <BIG> etiketi kullanılır. Metin normalden küçük yazılmak istenirse <SMALL> etiketi kullanılır. <SMALL> etiketi ile vurgulanan metin, normal paragraf metninden daha küçük görüntülenecektir. <BIG> ve <SMALL> etiketleri çiftler halinde kullanılır. Üstü çizili metinler oluşturulmak isteniyorsa <S> etiketi

kullanılır. <S> etiketi de çiftler halinde kullanılır. Üstü çizili metinler, inceleyiciler tarafından yazı boyunca üzerine bir çizgi çekilmiş olarak gösterilirler. Üstü çizili metinler aşağıdaki şekilde tanımlanabilir:

<S>Üstü çizili metin.</S>

Fiziksel stiller, inceleyiciye göstereceği formatı anlatır. HTML'de koyu, yatay, altı çizili ve eş genişlikli olmak üzere toplam dört fiziksel stil vardır. Her fiziksel stilin bir başlangıç, bir bitiş etiketi vardır. Kullanılacak karakterler <B> etiketi ile koyu renkte, <I> etiketi ile italik biçiminde <U> etiketi ile altı çizilmiş olarak basılabilir.

<TT> etiketi ile eş genişlikli olarak basılır.

<B>Koyu font</B>

<I>İtalik font</I>

<U>Altı çizili font</U>

<TT>Daktilo fontu veya eş genişlikli font</TT>

HTML dosyasındaki yazım şeklini bırakılan boşluklarda göz önüne alınarak metnin aynen görüntülenmesi amacıyla <PRE> etiketi kullanılır. Belirtilen bu etiketlerin başlangıç ve bitiş noktaları arasında belirtilen karakter veya karakter toplulukları etiketin ne olduğuna bağlı olarak şekil alırlar.

Ayrıca kullanılan karakterlerin büyüklükleri ve renkleri de <FONT> etiketi ile belirlenir. Font büyüklüğünü 1 ve 7 arasındaki değerleri kullanarak doğrudan belirlemek mümkündür.

<FONT SIZE=6 COLOR="blue">Yıldız Teknik Üniversitesi</FONT>

Yukarıda verilen örnekte <FONT> etiketi ile belirlenen bölüm bu etiketin içinde verilen karakter büyüklüğü ve rengine bağlı kalarak web sayfasında görüntülenecektir. Karakterin rengi saptanırken renk isimlerinin yerine bu renklere karşılık gelen iki basamaklı ondalık sayı düzenindeki değerleri "rgb" (kırmızı, yeşil, mavi) renk standardında kullanılabilir.

4.2.6 HTML Formları

Bu bölüme kadar HTML hakkında sözü edilen temel konular web sayfalarında kullanıcılara bilgi vermeye yönelik uygulamalardı. HTML aynı zamanda bunun tersi olan kullanıcıdan bilgi almaya yönelik özellikler taşıyan uygulamalara da sahiptir. HTML formları,

kullanıcılardan bilgi almaya yönelik özellikler taşıyan uygulamalara da sahiptir. HTML formları, kullanıcılardan bilgi almaya yönelik alanları içeren Web sayfalarının bir parçasıdır.

Bir HTML formunda kullanıcıdan bilgi alınan alanlar için <INPUT> etiketi kullanılır. Bu etiket içerisinde kullanılan "type", "name" ve "value" önemli parametrelerdir. "Type" parametresi kullanıcıdan alınan bilgilerin türünü belirler. "Name" parametresi etkileşimli programlar açısından oldukça önemli bir yere sahiptir. Bu parametrenin karşılığında verilen isim aslında kullanıcıdan form aracılığıyla alınan bilginin atandığı değişkendir ve bu değişkeni HTML'in yanı sıra PHP'de kullanılabilmektedir. Çoktan seçmeli durumlar için verilen bilgilerin aynı değişkene atanabilmesi amacıyla "value" parametresi kullanılır.

4.2.7 Resim ve Renk Özellikleri

Web sayfalarının hazırlanması sırasında göze hitap edecek özelliklerin varlığı HTML programları içerisinde "jpg" ve "gif" gibi biçimlere sahip dosyaların kullanılmasına bağlıdır. Bu biçimler görsel özellikler taşıdıklarından tasarım konusunda çok fazla kullanılırlar. Bu dosyaları HTML sayfalarında görüntülemenin kullanış amacına göre iki yolu vardır. Web sayfalarında yalnızca görüntü amaçlı basmak için <IMG> etiketi ile diğer HTML sayfaları ile bağlantı kurmak için <A HREF> etiketi bulunmaktadır. <IMG SRC>etiketi, dokümanlarda metinle birlikte görüntü de kullanılabilmeye olarak sağlar. <IMG> etiketi üç parametreye sahiptir ve sonlanma elemanı yoktur. <IMG> etiketinin kullanılması gereken tek parametresi "SRC" dir. Bu şekilde sözü edilen resime ait, kaynak, yol ve isim belirlenir.

<IMG SRC ="yildiz.gif">

Resmin ayrıca bağlantı aracı olarak bu tür biçimler kullanılmak istendiği zaman aşağıdaki gibi bir satırın program içerisine yazılması gerekmektedir.

<A HREF ="sinav.html"><IMG SRC="yildiz.gif"></A>

Sayfaları hazırlanan resimler konulacağı gibi metnin veya arka planın renkleri üzerinde ayarlamalar da yapılarak tasarıma ek özellikler katabilir. Bunlara ek olarak sayfaların arka planları olarak resim dosyaları da yüklenebilir.

<BODY BGCOLOR ="aqua" TEXT ="blue">

<BODY BGGROUND ="yildiz.gif">

4.2.8 Tablolar

Web sayfalarının tasarımında kullanılan en etkin araçlardan birisi tablolardır. Tablolar, metin ve/veya resimleri belirli bir satır ve sütun düzeni içerisine koyar. Tabloların etkin hale getirilmesini <TABLE> etiketi sağlar. <TABLE> etiketinin bir çok parametresi kullanılarak, sayfada tabloların nerede ve nasıl görüntüleneceği tanımlanabilir. <TABLE> etiketi başlangıç ve bitiş etiketi ile birlikte kullanılır. Başlangıç tablo etiketi <TABLE>'ın geçerli bazı parametreleri aşağıdaki gibidir:

"CELLPADDING" ve "CELLSPACING" parametreleri: Tablonun okunabilirliğini arttırmak için "CELLPADDING" ve "CELLSPACING" parametreleri kullanılır. Veri hücreleri içine boşluk yerleştirmek için "CELLPADDING" parametresi kullanılır. Hücre içindeki ve arasındaki boşluklar yürürlükteki birim cinsinden genellikle piksel cinsinden belirlenir.

"WIDTH" parametresi: "WIDTH" parametresi, bir tablo için bağıl veya sabit genişlik belirlemek amacıyla kullanılır. Önceden belirlenmiş "WIDTH" değeri , yürürlükteki pencere genişliğidir. Genişlik, birimler cinsinden veya yürürlükteki pencere genişliğinin bir yüzdesi cinsinden tanımlanmış olur. Bu sayısal değerden sonra bir yüzde işareti yoksa bu sayı yürürlükteki birim, genellikle piksel cinsinden yorumlanır.

"Cols" parametresi: "COLS" parametresi, bir tablodaki sütun sayısını belirlemek için kullanılır. Dört sütunlu bir tabloyu aşağıdaki gibi belirleyabiliriz:

<TABLE COLS =4>

"BORDER" ve "FRAME" parametreleri: HTML'de tablo yaratmak, genellikle iki adımdan oluşan bir işlemdir. Tablonun çevresindeki, çerçevenin genişliğini, "BORDER" parametresi kullanılarak belirlemek mümkündür. Tablo çevresine bir çerçeve yerleştirmek için "FRAME" parametresi kullanılabilir. Tablolar önceden belirlenmiş olarak çerçeveye sahip değildirler. Tablonun çerçevelendirme tipi aşağıdaki değerler kullanılarak belirlenebilir.

FRAME = NONE Tablonu çevresinde çerçeve yoktur (default)

FRAME = TOP Tablonun sadece üst kısmına çerçeve konur.

FRAME = BOTTOM Tablonun sadece alt kısmına çerçeve konur.

FRAME = TOPBOT Tablonun alt ve üst kısmına çerçeve konur.

FRAME = SIDES Tablonun sağ ve sol kenarlarına çerçeve konur.

FRAME = ALL Tablonun dört kenarına da çerçeve konur.

FRAME = BORDER Tablonun dört kenarına da çerçeve konur.

(ALL ile aynı görevi görmektedir.)

Bütün tablolar bir veya daha fazla veri satırı içermek zorundadır. Satırlar bir tablonun başlık, gövde veya alt bilgi bölümünde tanımlanır. <TR> etiketi kullanılarak, tablonun içerdiği her bölüm için bir satır tanımlanabilir. Bu yüzden bir başlık, iki gövde ve bir alt bilgi bölümünden oluşan bir tablo en az dört satıra sahip olacaktır. Tablolarda sadece başlangıç <TR> etiketi kullanılır.

Hücre düzeyinde yapılan atamalar, bağımsız veri kümeleri veya başlıklar içindir. Veri hücreleri, tabloda görüntülenecek sayıları ve ifadeleri içerirler. <TD> etiketi kullanılarak veri hücrelerini tanımlamak mümkündür. Başlık hücreleri, bölüm, sütun ve satır başlıkları içerir. <TH> etiketi kullanılarak başlık hücreleri tanımlanabilir.

```
<TABLE BORDER=0 WIDTH =”100%” CELLPADDING=0>
<TR> <TD WIDTH =”50%”>ÜLKE</TD>
<TD WIDTH =50%”>ŞEHİR</TD></TR>
<TR><TD WIDTH =”50%”>Türkiye</TD>
<TD WIDTH =”50%”>İstanbul</TD></TR><TABLE>
```

Bu örnekte toplam iki satır ve bu satırların ikişer hücreye bölünmesinden ortaya çıkan iki sütundan oluşan bir tablo bulunmaktadır. Bu biçimlendirme sayesinde sayfa üzerinde yan yana veya alt alta gelmesi gereken satır ve sütunlar düzenlenebilmektedir. Ayrıca her bir sütun için tablonun kaçta kaçının ayrılacağı , tablonun sınırları gibi özellikler “BORDER” ve “WIDTH” parametreleri ile belirlenebilmektedir.

5. MySQL VERİTABANI VE SQL SORGULAMA DİLİ

5.1 MySQL Veritabanı

Bilindigi gibi bilgisayar dünyasında kullanılan birçok veritabanı programı ve sunucusu mevcuttur. Bunların çoğu kullanışlı olmasına rağmen çok pahalı paket programlardır. MySQL' in en büyük özelligi ücretsiz olmasıdır fakat ticari amaçla kullanıldığı takdirde küçük bir ücret ödemek gerekir. MySQL' in diğer en büyük özelligi ise veritabani pazarindaki en büyük rakiplerinden daha iyi, hızlı ve kullanışlı olmasıdır. MySQL halen daha geliştirilmekte olmasına rağmen mevcut haliyle zengin ve çok kullanılan fonksiyonlar sunmaktadır. MySQL' in yazılış nedeni, yaratıldığı yerde çok büyük bir veritabanını işleyebilecek bir SQL sunucusuna olan ihtiyaçtı. (Şamlı, 2002)

5.1.1 MySQL' in Özellikleri

- Sistemde birden fazla CPU var ise bunları kullanma.
- Değişik işletim sistemlerinde çalışması.
- Değişik sütun tipleri. İşaretli/İşaretsiz 1, 2, 3, 4 ve 8 byte uzunluğunda tamsayılar.
- Aynı sorgulama içinde değişik veritabanlarındaki tabloları birleştirme.
- Windows95 için ODBC (Open-DataBase-Connectivity). Microsoft Access'i kullanarak MySQL server'a bağlanabilme.
- Büyük veritabanlarını işleyebilme özelliği.
- C ve C++ dillerinde yazılmıs olması.
- Bütün verilerin ISO-8859-1 Latin 1 formatında kayıt ediliyor olması.

5.1.2 MySQL' in Elde Edilmesi

MySQL değişik işletim sistemleri için mevcut olup http://www.tcx.se adresine bağlanılıp elde edilebilecek siteler sırasıyla ülkelerine göre listelenmistir. Ayrıca MySQL' in resmi web sitesi olan www.mysql.com adresinden de indirilebilir. Bu adresten yazılımın son versiyonun zip uzantılı olarak da indirilmesi mümkündür. Aynı sayfadan myodbc yazılımı da mevcuttur.

5.1.3 MySQL' in Kurulumu

MySQL in kurulumu iki ceşit olabilir. Birinci seçenek, kaynak kodların elde edilip bunların

derlenmesi ve MySQL kurulması. İkinci seçenek ise binary yani derlenmiş MySQL yazılımı kullanarak kurulması. Eğer sisteme uygun binary dağıtım seti var ise binary setini veya source setini kullanmak arasında hiçbir fark yoktur. Aşağıda sırası ile binary dagıtım seti kurulduğunda belirecek olan dizinlerin ve bunlarin hangi dosyalari içerdiği gosterilmektedir.

bin	İstemci programlari ve mysqld sunucusu
data	Log dosyalari ve veritabanlari
scripts	mysql_install_db
share	Hata mesaj dosyalari
sql-bench	Test dosyalari

MySQL' in binary dagıtım seti Linux ortamında sıkıştırılmış bir şekilde sunulmaktadır. Bu sıkıştırılmış halini açabilmek için sistemde GNU gunzip ve tar gibi programların mevcut olması gerekmektedir. Bunlardan gunzip programı uncompress edip tar ise unpack etmektedir. Eğer sistemde bunlar mevcut ise aşağıdaki adımlar uygulanarak kurulum yapılabilir.

1) İlk önce sistemde MySQL'in kurulacağı dizin belirlenmeli. Linux üzerinde genelde programlar /usr/local dizini altında bulunmaktadır. Standartlara uyarak binary dagıtım seti /usr/local dizini altına kopyalanmalı.

2) /usr/local dizini altına kopyaladıktan sonra aşağıda belirtildigi gibi yukarıda bahsedilen gunzip ve tar programları kullanılarak binary dağıtım seti açılmalı

prompt> gunzip < mysql-SÜRÜM-İŞLETİMSİSTEMİ.tar.gz | tar xvf -

Bu komut girildikten sonra "mysql- SÜRÜM-İŞLETİMSİSTEMİ " şeklinde /usr/local altında bir dizin oluşacaktır.

3) Dizin oluşturduktan sonra bu dizin için bir sembolik bağlantı yaratılabilir. Sembolik bağlantı mysql olarak seçilirse eğer, /usr/local altında binary dağıtım setinin açılmış halinin bulundugu dizine girmek için prompt>cd mysql- SÜRÜM-İŞLETİMSİSTEMİ yazmak yerine prompt>cd mysql yeterli olur. Sembolik bağlantı yaratmak için kullanılacak komut:

prompt>ln -s mysql- SÜRÜM-İŞLETİMSİSTEMİ mysql

4) Sembolik bağlantğ yaratıldıktan sonra prompt>cd mysql komutu girilerek binary dağıtım

setinin açılmış halinin bulundugu dizine girilir. Burada yukarıda belirtilen dizinler mevcuttur. En önemli olan dizinler bin ve scripts dizinleridir. Sistemin herhangi bir yerinden bin dizini altindaki programlara ulaşmak için sistemde PATH kısmına bu dizin eklenmelidir. Scripts dizini altında bulunan mysql_install_db programı sunucuya erişim haklarının başlatılması için kullanılmaktadır.

Bu işlemden sonra sisteme mysql kurulmuştur. Kurulumun doğru çalışıp çalışmadığını kontrol etmek için yapılması gerekenler sırası ile;

1) mysqld sunucu programı çalıştırılarak başlangıç erişim tablosu (kullanıcıların ne haklarla veritabanına erişebileceklerini gösteren tablo) kurulur. Bu scripts dizini altında bulunan mysql_install_db programı ile elde edilebilir. prompt>scripts/mysql_install_db Eğer bu kurulmazsa, mysqld: Can' t find file: 'host.frm' gibi bir hata üretir. Bir önemli nokta da bu program calıştırılırken, root kullanıcısı olma gereğidir.

2) Sunucunun çalışıp çalışmadığı mysqladmin programı ile kontrol edilebilir.

prompt>bin/mysqladmin version

Bu komutun sonucunda sistemden sisteme göre değişik sonuçlar çıkabilir ama genel olarak aynıdır. Bu komuttan sonra sistemde kurulu mysql yazılımı ile ilgili bilgiler görüntlenir.

mysqladmin Ver 6.3 Distrib 3.22.9-beta, for pc-linux-gnu on i686

.....

3) Sunucu'nun kapatılıp kapatılmadığına bakılması.

prompt>bin/mysqladmin -u root shutdown

4) Sunucu'yu tekrardan çalıştırma

prompt>bin/mysqld &

Sunucu'da hiçbir problem yok ise yapılması gereken MySQL sunucusunda tanımlı olan root kullanıcısına şifre vermektir. MySQL sunucusunda tanımlı olan root kullanıcısı sunucudaki en yetkili kullanıcıdır. Yalnız bu kullanıcı Linux sistemindeki root kullanıcısı değildir. MySQL sunucusunun kendine özgü kullanıcı ve şifre listesi vardir. root kullanıcısı MySQL sunucusunda herseyi yapmaya yetkilidir. root kullanıcısına şifre vermek için

prompt>mysql -u root mysql

mysql>UPDATE user SET Password=PASSWORD('yeni_sifre') WHERE user='root';

Sunucunun yeni degerleri okumasi icin tekrar yuklenmesi gereklidir.

prompt>mysqladmin -u root reload

root kullanıcısı MySQL sunucusunda tanımlı olan en yetkili kullanıcıdır. Fakat veritabanını root kullanıcısı haricinde kullanacak kisiler olacaktir ve bunlarin veritabanlarına veya veritabanındakı tablolara erişim haklari düzenlenmelidir. Aşağıdaki örnekte MySQL sunucusunda bir kullanıcının nasıl yaratıldığı kısaca açıklanmıştır.

1) Kullanıcıyı yaratmak için MySQL sunucusuna root olarak bağlanılmalıdır.

prompt> mysql --user=root mysql

2) Bağlantı başarılı ise, MySQL' de şifresi MySqL98, kullanıcı adı "personel" ve tüm haklara sahip olan bir kullanıcı yaratılması

mysql> INSERT INTO user VALUES('%','personel', PASSWORD ('MySqL98'), 'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y');

3) MySQL' in değerleri okuyabilmesi için

mysql> quit

prompt> mysqladmin --user=root reload

Sistemde şimdi root kullanıcısının haricinde bir personel kullanıcısı oluşmuştur ve bu kullanıcı root kullanıcısı ile benzer haklara sahiptir.

5.1.4 MySQL Veri Türleri

MySQL'de birçok veri türü oluşturulabilir. Ancak Web programları açısından önemli olan birkaçı ve özellikleri şöyle sıralanabilir:

Çizelge 5.1 Web uygulamalarında sık kullanılan MySQL veri türleri ve özellikleri

Tür adı	**Özellikleri**
INT	Tamsayı: -2147483648'den 2147483647 kadar değişen diziye "signed" (işaretli), 0'dan 4294967295'e kadar değişenine "unsigned" (işaretsiz) denir.

VARCHAR(n)	n sayısını geçmemek şartıyla değişen boyutta karakter olabilir.
CHAR(n)	Kesinlikle n sayısı kadar karakter olabilir.
TEXT	En fazla 65535(2^16-1) karakter alabilen metin alanı.
MEDIUMTEXT	En fazla 16777215(2^24-1) karakter alabilen metin alanı.
DATE	1000-01-01'den 9999-12-31'e kadar değişebilen tarih alanı.
TIMESTAMP	1 Ocak 1970'den 18 Ocak 2038'e kadar olan ve Yıl+Ay+Gün+Saat+Dakika+Saniye biçimindeki zaman bilgisi.

5.2 SQL Sorgulama Dili

SQL, veri tabanı uygulamalarında, veri tanımlama, veri tabanı bütünlüğünün kontrolü, veri tabanlarına erişimin kontrolü ve veri tabanlarının sorgulanması (çeşitli kriterlere göre bilgi dökümü ve rapor elde edilmesi) ve güncellenmesi amaçları için gerekli komutlara sahip olan bir alt dildir (sub language). Alt dil denilmesinin nedeni, bir bilgisayar dilinin sahip olması gereken tüm komutlara sahip olmayışıdır.

SQL' in sahip olmadığı komutlar, çevrim (döngü-loop) oluşturan komutlarla, IF, THEN ELSE ya da GOTO gibi kontrol ve dallanma komutlarıdır.

Fakat SQL' in diğer dillerle birlikte kullanılması mümkün olduğundan, gerekiyorsa, SQL komutları diğer dillerin çevrim ya da kontrol komutları içinde kullanılabilir. (Uysal,1994)

5.2.1 Bir Veri Tabanının Oluşturulması

SQL ile gerçekleştirilecek işlemlerde, üzerinde işlem yapılacak olan veri tabanı kütükleri ya da tablolar, bir veri tabanı oluşturulur. Bu veri tabanını oluşturmak için,

CREATE DATABASE isim;

şeklindeki SQL komutunu kullanmak gerekir. Bu komut belirtilen isim'deki veri tabanını oluşturur. (Burada veri tabanı, içinde çok sayıda veri tabanı kütüğü ya da SQL terimleri ile tablo bulunan bir çevre bellek bölgesidir ya da bir alt dizin (subdirectory) adıdır).

5.2.1.1 Tabloların Yaratılması

SQL ile tablo oluşturmak için,

CREATE TABLE isim;

şeklindeki komut kullanılmaktadır. Ancak bu komutu uygun şekli ile kullanmak gerekir. Aşağıda personel bilgilerinin (Sicil No., Sosyal Güv.No., Ad, Soyad, Doğum Tarihi, Adres, Cinsiyet, Brüt Maaş, Bölüm No, Yönetici Sosyal Güv.No.) bulunduğu, Personel isimli tablonun oluşturulması için gerekli SQL komutları verilmiştir:

```
CREATE TABLE personel
( sicil INTEGER NOT NULL,
sosy_g_no CHAR ( 8 ) NOT NULL,
ad CHAR ( 10 ) NOT NULL,
soyad CHAR ( 10 ) NOT NULL,
dog_tar DATE,
adres CHAR ( 50 ),
cins LOGICAL,
Brut NUMERIC ( 13 , 2 )
bol_no SMALLINT,
yon_s_g_n CHAR ( 8 ) );
```

5.2.1.2 SQL' de Veri Tipleri

Çizelge 5.2 SQL dili veri tipleri ve özellikleri

Veri Tipi	SQL Komutu	Özellliği
1-) Sabit uzunluklu karakter	CHAR (uzunluk)	Sayısal işleme sokulmayacak verilerdir. Adres, isim, açıklama v.b. Uzunluk sabit olarak belirlenir. CHAR (15) gibi. Maksimum uzunluk 254 karakterdir.

2-) Değişken uzunluklu karakter	VARCHAR (uzunluk) Buradaki uzunluk maksimum uzunluktur. Veri daha kısa ise, uzunluğun gerektiği kadarı kullanılır.	Karakter türündeki veriler gibidir. Tek fark, uzunluk değişkendir, yani verinin gerektirdiği kadar uzunluk kullanılır. Bu veri tipi standart değildir. Her derleyicide bulunmaz
3-) Nümerik Tamsayı	INTEGER	-2147483648 ile 2147483647 arasındaki tamsayılar için kullanılır. Ondalık nokta kullanılamaz. Bellekte 4 byte'lık yer kaplar.
4-) Nümerik Kısa Tamsayı	SMALLINT	-32768 ile 32767 arasındaki tamsayılardır. Bu veri tipi standart değildir. Bazı derleyicilerde tanımlanmamıştır. Bu durumda veri INTEGER' a çevirilir.
5-) Ondalık Sayı	DECIMAL (x , y) REAL (x , y) ya da NUMERIC (x , y) şeklinde tanımlanabilir.	x, sayının maksimum hane (digit) sayısı, y ise ondalık noktadan sonraki hane sayısıdır. x en fazla 20 olabilir. y, 0 - 18 arasındadır.
6-) Üstel Sayı	FLOAT (x , y) Örnek:	x sayısının toplam hane (digit) miktarıdır. Maksimum değer 20' dir. y, ondalık noktadan sonraki hane sayısıdır. 0-18 arasındadır. 0.1 E – 307 ile 0.9 E + 308 arasındaki değerleri alabilir.

	3.1E + 17 gibi (3.1 * 10^17 anlamındadır)	Çok küçük ve çok büyük sayılar için uygun olan veri tipidir.
7-) Tarih türü veriler	DATE	Tarih türü bilgiler üzerinde işlem yapma imkanı sağlar. SQL için standart bir veri tipi değildir.
8-) Mantıksal Veri	LOGICAL	Doğru (true, .T.) ya da yanlış (false, .F.) şeklinde değer alabilen veriler için kullanılır.
9-) Zaman Türü Veriler	TIME	ssddsn şeklindeki (saat, dakika, saniye) verileri için kullanılır. Standart değildir.
10-) Tarih ve zaman türü veriler	TIMESTAMP	Tarih ve zaman türü verilerin bir karışımı şeklinde kullanılan veriler içindir. Standart değildir.
11-)Grafik türü veriler	GRAPHIC (n)	n adet 16-bit' lik karakterden oluşan bir sabit uzunluklu bilgi tanımlamak için kullanılır. (Karakter türü veri, 8 bit' lik karakterlerden oluşmaktadır). Standart değildir.
12-) Değişken uzunluklu Grafik türü veri	VARGRAPHIC (n)	n adet 16 – bit' lik karakterden oluşan değişken uzunluklu (n maksimum uzunluktur) bir bilgi tanımlamak için kullanılır. Standart değildir.

Çizelge 5.3 Tarih formatları

Standart Adı	Tarih Formatı	Örnek
Uluslararası Standartlar Organizasyonu (ISO)	yyyy-aa-gg y – yıl a – ay g – gün	1992-11-29
IBM ABD Standardı USA	aa/gg/yyyy	11/29/1992
IBM Avrupa Standardı EUR	gg.aa.yyyy	29.11.1992
Japon Endüstri Standardı JIS	yyyy-aa-gg	1992-11-29

Çizelge 5.4 Zaman formatları

Standart Adı	Zaman Formatı	Örnek
Uluslararası Standartlar Organizasyonu (ISO)	SS.dd.ss S – saat d – dakika s – saniye	16.25.07
IBM ABD Standardı USA	SS:dd AM veya PM	16:25 PM
IBM Avrupa Standardı EUR	SS.dd.ss	16.25.07
Japon Endüstri Standardı JIS	SS:dd:ss	16:25:07

Tablolar ile ilişkili veri tiplerinden başka açıklanması gereken diğer bir hususta NOT NULL ifadesidir.

NOT NULL ifadesi, sözkonusu alan ile ilişkili olarak mutlaka veri yüklemesi gerektiğini, ilgili alanın boş bırakılamayacağını anlatmaktadır. Genellikle sıralama anahtarı olarak ta kullanılabilen alanlar bu şekilde tanımlanmaktadır.

NOT NULL ifadesi yoksa, o alan NULL anlamındadır; yani o alan ile ilşkili olarak tabloya veri yüklenmemesi durumuna da müsaade edilmektedir.

5.2.1.3 Tablolara Veri Yüklenmesi

Bir tabloya veri girişi (ya da veri yüklenmesi) işlemi için, SQL'de mevcut olan komut

INSERT INTO / VALUES komutudur.

Bu komut yardımı ile personel adlı tabloya, ilk satır bilgileri aşağıdaki gibi düzenlenmiş bir INSERT komutu ile girilebilir:

INSERT INTO personel

VALUES (1, '27345627', 'Ahmet' , ' Okan' ,{01/05/62}, 'Cumh. Cad. 47 / 2 Taksim-İstanbul' , .T. , 17500000.00 , 1 , '23112244');

Komut içindeki değerler incelendiğinde, sayısal değerler olduğu gibi yazılmakta, karakter türü veriler ' ' sembolleri içine alınmakta, lojik türdeki veriler .T. ya da .F. şeklinde belirlenmekte ve tarih türü bilgi ise { } sembolleri ile ayırdedilmektedir.

INSERT INTO komutu ile , belli bir anda, tabloya, bir satır yüklenmektedir. Tabloya çok sayıda satır girilmek istenirse, satır sayısı kadar INSERT komutu kullanılmalıdır.

SQL dilinin tüm ekran, üzerinde bilgi girişi için kolaylık sağlayan ve etkileşimli olarak çalışmayı sağlayacak komutları yoktur. Bu nedenle, SQL tabloları içine bilgi girişi genellikle, SQL' in içinde kullanıldığı üst dilin komutları kullanılarak gerçekleştirilir.

5.2.1.4 Tablolardaki Sütun İsimleri ve Tablo İsimleri ile İlişkili Kurallar

SQL dilinde bir tabloya ya da tablo içindeki bir sütuna isim vermek için gerekli kurallar, bir SQL uygulamasından ötekine değişebilmektedir. Fakat genellikle, geçerli olan kurallar aşağıda verilmiştir:

- İsim uzunlukları 18 karaktere kadar olabilir (Bazı SQL uygulamalarında 8 karaktere kadar)
- İlk karakter bir harf olmalıdır. Onu izleyen karakterler, harf, rakam ya da alt çizgi (_)

sembolü olabilir.

5.2.2 Temel SQL Sorgulamaları

5.2.2.1 SELECT Komutu

Tek tablodan gerekli bilgileri elde etmek için sorgulama yapabilecek SQL komutu olan SELECT' in en basit şekli aşağıdaki gibidir:

SELECT * FROM personel;

Bu komut, personel adlı tablo içindeki bütün bilgileri koşulsuz olarak listeleyecektir. SELECT sözcüğünü izleyen kısımda * sembolünün bulunması, ilgili tablodaki bütün sütun isimlerinin ve ilgili bilgilerin listelenmesini sağlayacaktır.

SELECT sözcüğünü izleyen kısımda sütun adları, FROM sözcüğünü izleyen kısımda ise tablo ismi belirtilmektedir. Komutun ,

SELECT sicil, ad, soyad, brut FROM personel ;

şeklinde kullanılması ise, sadece belirtilen sütun başlıkları ile ilgili bilgilerin listelenmesini sağlayacaktır.

5.2.2.2 Tekrarlı Satırların Ortadan Kaldırılması

SQL' de , ilişkisel veri tabanından farklı olarak, birbirinin aynı data içeren satırlara müsaade edilir. Birbirinin aynı olan satırların, listeleme esnasında, bir kez yazılması için, SELECT komutuna DISTINCT sözcüğü eklenir.

SELECT DISTINCT sat_no

FROM Par_sat;

Bu komut ile Par_Sat isimli tablodan satıcı no'ları tekrarsız olarak listelenecektir.

5.2.2.3 Tablo Bilgilerinin Sıralanmış Olarak Listelenmesi

Tablodan listelenecek bilgilerin, belirli bir sütun adına göre sıralanmış olarak görüntülenmesi için , SELECT komutuna ORDER BY sözcüğü ilave edilir.

SELECT sicil, ad, soyad, brut

FROM personel

ORDER BY brut ASC;

Bu komut, personel tablosundaki bilgilerden sicil, ad, soyad ve brüt sütunlarını maaş'a göre artan sırada (küçükten büyüğe doğru) sıralı olarak listeleyecektir.

ASC sözcüğü ascending (artan) anlamındadır. Veriler azalan sırada (büyükten küçüğe ya da alfabetik olarak Z' den A' ya doğru) sıralanmak istense idi, bu sözcük yerine DESC (descending) sözcüğünü kullanmak gerekecekti.

SELECT sicil, ad, soyad, brut

FROM personel

ORDER BY brut DESC;

Bir tablo içindeki veriler, aynı anda birden çok sütuna göre de sıralanabilir.

SELECT sicil, ad, soyad, brut

FROM personel

ORDER BY ad ASC, brut DESC;

Bu komut sonucunda tablo öncelikle ad'a göre artan sırada (A'dan Z' ye doğru) sıralanacak, sadece aynı ad' a sahip olanlar da kendi aralarında brüt'e göre azalan sırada (yüksek maaştan düşük maaşa doğru) sıralanacaktır.

5.2.2.4 Koşula Bağlı Olarak Listeleme

SELECT komutu ile bir tablonun satırları içinden sadece verilen bir koşulu sağlayan satırlar listenebilir. Örneğin, brüt maaşı 500.000.000 TL'den fazla olan personel listelenmek istenirse, SELECT komutu aşağıdaki gibi yazılmalıdır:

SELECT *

FROM personel

WHERE brut >500000000;

Burada WHERE sözcüğünü izleyen kısımda koşul belirtilmektedir. Koşul belirtilirken iki veri birbiri ile karşılaştırılmaktadır. Karşılaştırma ifadesinde, karşılaştırılan verilerin türü aynı olmalıdır.

Çizelge 5.5 SQL Karşılaştırma operatörleri

Operatör	Anlamı
<	...den daha küçük
>	...den daha büyük
=	eşit
<=	küçük veya eşit
>=	büyük veya eşit
<>	eşit değil
!=	eşit değil
* Bazı SQL gerçekleştirimlerinde !< (den küçük değil) ve !> (den büyük değil) operatörleri mevcuttur.	

5.2.2.5 Çeşitli Veri Tipleri İçin Basit Sorgulamalar

Nümerik Veri Tipi

Nümerik (sayısal) veri tipi, SMALLINT, INTEGER, DECIMAL, NUMERIC ya da FLOAT tip bildiri sözcüklerinden biri ile tanımlanan, matematiksel işlemlere sokulabilen, özel semboller içine alınmayan verileri kapsar.

Örnek: Brüt Maaşı 800.000.000'dan fazla olmayan personeller

```
SELECT *

FROM personel

WHERE brut <= 800000000;
```

Karakter (CHAR) Veri Tipi

Karakter türündeki veriler, çift tırnak (" ") veya tek tırnak (' ') sembolleri içine yazılırlar. Bu tip veriler, rakamlardan oluşsa bile, matematiksel işlemler içinde kullanılamazlar.

Örnek: Adı Ali olmayan personeller

```
SELECT *
FROM personel
WHERE ad <> "Ali";
```

Tarih Veri Tipi

Tarih tipi veriler, { } sembolleri içine yazılmalıdır.

Örnek: Doğum Tarihi 1960 yılından önce olan personeller

```
SELECT *
FROM personel
WHERE dog_tar <= {12/31/59};
```

Mantıksal (Lojik) Veri Tipi

Mantıksal veriler için mümkün olabilen sadece iki değer sözkonusudur : Doğru (True , .T.), yanlış (False, .F.). Personel tablosunda cinsiyeti erkek olanlar True, kadın olanlar False olarak kodlandığı kabul edilirse, personel içinden erkek olanların listelenmesi için, aşağıdaki komut yazılmalıdır:

```
SELECT *
FROM personel
WHERE cins=.T.;
```

ya da

```
SELECT *
FROM personel
WHERE cins;
```

5.2.2.6 Birden Çok Koşula Dayalı Sorgulamalar NOT, AND, OR

NOT, AND ve OR mantıksal operatörleri yardımı ile birden çok koşulun gerçekleşmesine bağlı olarak ifade edilebilecek karmaşık ya da bileşik koşulu listelemeleri gerçekleştirmek mümkün olmaktadır.

Örnek: Maaşı 500.000.000 TL'dan fazla olan ve cinsiyeti erkek olan personeller

```
SELECT *

FROM personel

WHERE brut>500000000 AND cins=.T.;
```

NOT, AND ve OR operatörlerinin etkileri aşağıdaki tablolarda belirtilmiştir.

(T-True/ F-False)

Çizelge 5.6 SQL NOT Operatörü değerleri

Koşul	**NOT koşul**
.T.	.F.
.F.	.T.

Çizelge 5.7 SQL AND Operatörü değerleri

1.Koşul	**2.Koşul**	**1.Koşul AND 2.Koşul**
.T.	.T.	.T.
.T.	.F.	.F.
.F.	.T.	.F.
.F.	.F.	.F.

Çizelge 5.8 SQL OR Operatörü değerleri

1.Koşul	**2.Koşul**	**1.Koşul AND 2.Koşul**
.T.	.T.	.T.
.T.	.F.	.T.
.F.	.T.	.T.
.F.	.F.	.F.

5.2.2.7 Bir Veri Kümesi İçinde Arama – IN Operatörü

```
SELECT *

FROM personel

WHERE bol_no=1 OR bol_no=2 OR bol_no=3;
```

Personel tablosundan bölüm No.'su 1, 2 ya da 3 olan kayıtları listeleyen yukarıdaki SQL komutu yerine IN operatörü kullanarak ta aynı sorgulama daha kısa bir ifade ile yapılabilmektedir.

```
SELECT *

FROM personel

WHERE bol_no IN (1,2,3 );
```

IN Operatörü NOT ile de kullanılabilir.

```
SELECT *

FROM personel

WHERE bol_no NOT IN (1,2,3 );
```

5.2.2.8 Aralık Sorgulaması – BETWEEN Sözcüğü

```
SELECT *

FROM personel

WHERE brut > = 500000000 AND brut < = 1000000000 ;
```

Personel tablosundan maaşı 500.000.000 TL ile 1.000.000.000 TL arasında olan kayıtları listeleyen yukarıdaki SQL komutu yerine BETWEEN sözcüğü kullanarak ta aynı sorgulama daha kısa ve etkin bir şekilde yapılabilmektedir.

```
SELECT *

FROM personel

WHERE brut  BETWEEN 500000000 AND 1000000000 ;
```

5.2.2.9 Karakter Türü Bilgi İçinde Arama Yapma – LIKE Sözcüğü

Karakter türü bilgi içeren bir kolonda, bir veya birden fazla karakteri aramaya yarayan bu komut aşağıdaki şekilde kullanılmaktadır:

Örnek: Taksim semtinde ikamet eden personeller

```
SELECT *

FROM personel

WHERE adres LIKE '%Taksim%';
```

Bu komut, adres alanında 'Taksim' ifadesi geçen kayıtları listeleyecektir.

% sembolü, 'Taksim' sözcüğünün öncesi ve sonrasındaki karakterler ne olursa olsun anlamındadır. Yukarıdaki LIKE 'Taksim' in en başta veya sonda olduğu adresleri listeleyecektir.

LIKE sözcüğü alt çizgi (_) sembolü ile birlikte de kullanılabilmektedir.

```
SELECT *

FROM personel

WHERE ad LIKE 'Mehmet____';
```

şeklindeki komut ile ad alanı "Mehmet" ile başlayan ve ad alanı uzunluğu 10 karakter olan isimlere sahip personeli listeyecektir.(_) sembolü, tek karakterlik bir bilgiyi temsil etmektedir.

5.2.3 SQL' de Aritmetiksel İfadeler ve Fonksiyonlar

5.2.3.1 Aritmetiksel İfadeler

SELECT komutu ile, veri tabanında mevcut tablolardan listeleme yaparken, tabloda ayrı bir sütun olarak yer almamış ve ancak bir hesaplama sonucunda üretilebilecek bilgileri de liste içine katmak mümkündür.

Aşağıdaki SELECT komutu ile, personelin şu anda geçerli maaşı ile, bu maaşın %32 zamlı şekli listelenmektedir:

```
SELECT  ad, soyad, maaş , maaş * 1.32  FROM personel
```

Çizelge 5.9 SQL Kullanılan aritmetiksel semboller

Operatör	İşlevi
** veya ^	Üs alma
*	Çarpma
/	Bölme
+	Toplama
-	Çıkarma

Öncelik sırası, matematikte ve diğer bilgisayar dillerinde olduğu gibi, Üs alma, Çarpma-Bölme (aynı önceliğe sahip), Toplama-Çıkarma (aynı önceliğe sahip)' dır. Parantez kullanılarak, öncelik sıraları değiştirilebilir.

5.2.3.2 Kümeleme Fonksiyonları

SQL, tablo içinden, çeşitli matematiksel işlemlerin sonucunu otomatik olarak üretmeyi sağlayan, fonksiyonlara sahiptir.

SUM Fonksiyonu

Fonksiyonun argümanı olarak belirtilen sütun ile ilişkili olarak toplam işlemini gerçekleştirir.

Örnek: İşletmedeki personelin brüt maaşları toplamı ne kadardır?

SELECT SUM (brut)

FROM personel;

AVG Fonsiyonu

Aritmetiksel ortalama (average) hesaplamak için kullanılır.

Örnek: Bilgi İşlem Bölümü'nün (bol_no = 5) maaş ortalaması nedir?

SELECT AVG (brut) FROM personel WHERE bol_no=5;

MAX Fonksiyonu

Tablo içinde, belirlenen sütun içindeki en büyük değeri bulur.

Örnek: İşletmedeki en yüksek maaş ne kadardır?

```
SELECT MAX ( brut )

FROM personel;
```

MIN Fonksiyonu

Tablo içinde, belirlenen sütun içindeki en küçük değeri bulur.

Örnek:İşletme içinde 4 Mayıs 1970' den önce doğanlar için, asgari ücret nedir?

```
SELECT MIN ( brut )

FROM personel

WHERE dog_tar < {05/04/70};
```

COUNT Fonksiyonu

Tablo içerisinde herhangi bir sayma işlemi gerçekleştirmek için kullanılır.

Örnek: Ücreti 600.000.000 TL'den fazla olan personel sayısı nedir?

```
SELECT COUNT ( * )

FROM personel

WHERE brut > 600000000;
```

COUNT komutunda, * argümanının kullanılması, bütün sütunların işleme sokulmasını, alan adının belirtilmesi ise (COUNT (bol_no) gibi), sadece belirtilen sütunun işleme sokulmasını sağlar.

5.2.3.3 Gruplandırarak İşlem Yapma

Kümeleme Fonksiyonlarını, tablodaki bilgileri bazı özelliklere göre gruplandırarak bu gruplandırılmış veri üzerinde uygulamak mümkündür. Bu işlem, GROUP BY sözcüğü kullanılarak gerçekleştirilir.

Örnek: İşletmedeki her bölümdeki ortalama maaş nedir?

```
SELECT bol_no, AVG ( brut )

FROM personel
```

GROUP BY bol_no;

Gruplandırarak, kümeleme fonksiyonları uygulanırken, koşul da verilebilir. Bu durumda, grup üzerindeki hesaplamalarla ilişkili koşul belirtilirken HAVING sözcüğü kullanılmalıdır.

Örnek: Ortalama maaşın, 900.000.000 TL'den fazla olduğu bölümlerdeki personele ait ortalama maaşların listelenmesi

SELECT bol_no,AVG (brut)

FROM personel

GROUP BY bol_no

HAVING AVG (brut) > 900000000;

HAVING sözcüğü, SELECT komutunda GROUP BY sözcüğü bulunmadığı zaman geçersizdir. HAVING sözcüğünü izleyen ifade içinde, kümeleme fonksiyonlarından en az biri bulunmalıdır.

WHERE sözcüğü bir tablonun tek tek satırları üzerinde işlem yapan koşullar için geçerli iken, HAVING sözcüğü sadece, gruplandırılmış veriler üzerindeki işlemlerde geçerlidir. Bazı durumlarda HAVING ve WHERE sözcükleri birlikte, SELECT komutu içinde kullanılabilir

Örnek: Personel tablosu içinde, her bölümde, erkek personele ait maaşlar için, ortalamanın 900.000.000 TL'den fazla olduğu bölümleri listeleyiniz.

SELECT bol_no, AVG (brut)

FROM personel

WHERE Cins = .T.

GROUP BY bol_no

HAVING AVG (brut) > 900000000;

5.2.4 Birden Fazla Tabloyu İlişkilendirerek Sorgulama

SQL' in sorgulama gücü tek tablo üzerinde sorgulama yapmaktan daha ziyade, daha sık karşılaşılan ve güç olan, birden çok tablonun birbiri ile ilişkilendirilmesini gerektiren sorgulamalarda ortaya çıkmaktadır.

5.2.4.1 Birleştirme (JOIN) işlemi

sicil	sosy_g_no	ad	soyad	dog_tar	adres	cins	brut	bol_no	yon_s_g_n
112	27641	ali	can	01/05/60	fatih	.T.	800000000	1	037165
175	3777654	ayşe	şen	04/07/65	kadıköy	.F.	700000000	1	037165

Şekil 5.1 Örnek veritabanı tablosu (Personel bilgileri)

bolum_ad	bolum_no	y_sosy_g_n	y_is_b_tar
satış	1	037165	01/07/89
bilgi işlem	5	288111	05/02/92

Şekil 5.2 Örnek veritabanı tablosu (Bölüm bilgileri)

Belirli bir personel ile ilişkili bilgiler , personel tablosunun o personele ait satırında mevcuttur. Ancak, personelin yöneticisi ile ilişkili bilgilerin bir kısmına ise bölüm tablosundan erişilebilir. Çalışan her personel ve bu personelin yöneticisi ile ilişkili bilgilere ulaşılması durumu zorunlu olarak personel ile bölüm tabloları arasında ilişki kurulmasını gerektirir. Bu ilişki, ancak, müşterek bir alan yardımı ile kurulabilir. Burada müşterek alan bölüm numarasıdır ve personel tablosunda bol_no, bölüm tablosunda ise bolum_no adı ile yer almaktadır.

Müşterek alana göre, personel ve bölüm tablolarının birleştirilmesi (JOIN) demek, her iki tablodaki tüm alanları içeren yeni bir tablo oluşturmak demektir. Yalnız bu tabloda sadece, her iki tabloda da mevcut olan bölüm numaraları ile ilişkili satırlar yer alacaktır.

Birleştirme işlemi ile listeleme aşağıdaki SQL komutu ile gerçekleştirilebilir:

```
SELECT *

FROM personel, bölüm

WHERE personel.bol_no = bölüm.bolum_no ;
```

İlişkilendirme "Tablo adı . kolon adı" ifadesi ile sağlanmaktadır.

Personel ve bölüm tablolarının, müşterek alan olan bölüm numarası üzerinde JOIN işlemine tabi tutulması sonucunda elde edilen bilgi

sicil	sosy_g_no	Ad	Soyad	Dog_tar	Adres	cins	brut	bol_no	yon_s_g_n	bolum_ad	bolum_no	Y_sos_g_no	y_is_b_tar
112	27641	ali	can	01/05/60	fatih	T	800000000	1	037165	satış	1	037165	01/07/89
175	37654	ayşe	şen	04/07/65	kadıköy	F	700000000	1	037165	satış	1	037165	01/07/89

Şekil 5.3 Örnek sorgu sonucu (JOIN –Birleştirme işlemi)

Tablodan görüleceği üzere, personel ve bölüm tablolarında sadece her iki tabloda da aynı olan bölüm numaralarına ait satırlar alınarak birleştirilmiştir.

Listelenen birleştirilmiş tabloda, her iki tablodan da alındığı için tekrar eden bölüm numarası ve yönetici sosyal güvenlik numarası gibi sütun başlıklarının bir kez yazılması isteniyorsa, bu taktirde SELECT komutunda * sembolü yerine sadece, sonuçta yazması istenen sütun başlıkları kullanılmalıdır.

5.2.4.2 Bir Tablonun Farklı İsimlerdeki Eşdeğerleri ile Sorgulama

Daha önceden tanımlanmış bir tablonun, farklı isimli bir eşdeğerini oluşturarak sorgulamalarda kullanmak mümkündür.

Örnek: Her personel için, personel sicil numarası, ad ve soyadı ile bu personelin yöneticisinin ad, soyad ve doğum tarihini listeleyiniz.

```
SELECT A.sicil, A.ad, A.soyad, B.ad, B.soyad, B.dog_tar

FROM personel A B

WHERE A.yon_s_g_n = B.sosy_g_no;
```

Bu SELECT komutu ile, personel tablosunun A ve B isimli birer kopyası oluşturulur. Bu kopyalara personel kütüğünün eşdeğerleri ya da takma adlıları (aliases) adı verilir. Buradaki yöntem ile bir tablonun kendisi ile birleştirilmesi işlemi kendi üzerinde birleştirme (self-join) adını almaktadır.

5.2.4.3 İçiçe SELECT Komutları (NESTED SELECTS)

Bazı sorgulamalar, özelliği itibari ile içiçe SELECT komutlarının kullanılmasını

gerektirebilir. İçteki SELECT komutunun bulduğu sonuç, dıştaki SELECT komutunun işlevini yerine getirmesi için kullanılır.

Örnek: Parça numarası 24 olan parçayı kullanan projelerde çalışan personelin listelenmesi

par_no	par_ad	pr_no	fiyat	agırlık
24	Vida	2	2000	500
24	Vida	4	2000	500
37	Cıvata	2	6000	800
87	conta	2	7000	5000
112	pim	5	6000	70

Şekil 5.4 Örnek veritabanı tablosu (Parça bilgileri)

proje_ad	proj_no	yer	bl_no
1	1	İstanbul	4
2	2	İstanbul	4
3	3	Ankara	5
4	4	Ankara	5
5	5	İzmir	4

Şekil 5.5 Örnek veritabanı tablosu (Proje bilgileri)

sicil	sosy_g_no	ad	soyad	dog_tar	bol_no	adres
117	274251	ali	can	05/01/60	4	fatih
247	527241	hasan	okan	04/07/62	4	pendik
1148	52625	mert	caner	04/08/70	5	beşiktaş

Şekil 5.6 Örnek veritabanı tablosu (Personel bilgileri)

per_s_g_no	proje_no	saat
274251	1	250
527241	2	350
527672	3	400
443211	5	300
527625	4	250

Şekil 5.7 Örnek veritabanı tablosu (Çalışma bilgileri)

Bu tablolardan yararlanılarak aşağıdaki içiçe SELECT komutları ile istenen işlem gerçekleştirilebilir:

```
SELECT * FROM personel
WHERE sosy_g_no
IN ( SELECT per_s_g_no
    FROM parca, proje, çalışma
    WHERE pr_no = proj_no AND
        proj_no = proje_no AND
        par_no = 24 );
```

Buradaki içteki SELECT komutu, parça, proje ve çalışma tablolarını proje numaraları üzerinde birleştirerek elde edilen genişletilmiş tablodan sadece parça no' su 24 olan satırdaki personel sosyal güvenlik numaralarını çıkarmakta, dış SELECT komutu ise, personel tablosundan bu sosyal güvenlik numaralarına sahip personelleri listelemektedir.

sicil	sosy_g_no	ad	soyad	dog_tar	bol_no
247	527241	hasan	okan	04/07/62	4
1148	527625	mert	caner	04/08/70	5

Şekil 5.8 Örnek sorgulama sonuc bilgileri (NESTED-SELECTS İç içe select)

5.2.4.4 UNION sözcüğü

UNION sözcüğü küme birleşimi işlemi görür. İki ayrı SELECT komutunun sonucunda elde edilen tabloların birleşimi işlemini gerçekleştirir.

Örnek: Adı Ahmet ve Soyadı Caner olan kişi ya da kişileri, işletmenin yürüttüğü projelerde çalışan bir kişi (sıradan bir personel ya da bölüm yöneticisi) olarak bulunduran projelerin isimlerini ve projelerin yürütüldüğü yerleri listeleyiniz.

```
( SELECT proj_ad, yer
FROM proje, bölüm, personel
WHERE bl_no = bolum_no AND
        y_sos_g_no = sosy_g_no AND
        ad = "Ahmet" AND
        soyad = "Caner")
UNION ( SELECT proj_ad, yer
         FROM proje, çalışma, personel
         WHERE proj_no = proje_no AND
         per_s_g_no = sosy_g_no AND
         ad = "Ahmet" AND
         soyad = " Caner")
```

UNION sözcüğü ile, iki ya da daha çok SELECT' in sonucu olan tabloların küme birleşimi işlemine tabi tutulması için iki koşul gereklidir:

1-) SELECT komutları sonucunda elde edilecek tablolar aynı sayıda kolon içermelidirler.

2-) Sonuç tabloların karşılıklı olarak kolonları aynı veri tipi ve aynı genişlikte olmalıdır.

5.2.4.5 ANY Sözcüğü

ANY sözcüğünün SELECT komutu içindeki etkisi aşağıdaki örnek ile açıklanmaktadır:

Örnek: Satış bölümünde çalışan personelin herhangi birinden daha düşük maaş alan ve mühendislik bölümünde çalışan kişilerin listelenmesi

```
SELECT *
FROM personel
WHERE brut < ANY
( SELECT brut
  FROM personel
  WHERE bol_no = 2 ) AND
  bol_no = 1;
```

Bu çözümün eşdeğeri olan ifade ise şöyledir:

```
SELECT *
FROM personel
WHERE brut < ( SELECT MAX ( brut )
                FROM personel
                WHERE bol_no = 2 ) AND bol_no = 1 ;
```

Buradaki düşünce tarzı şöyledir:

Mühendislik bölümünde çalışan ve satış bölümündeki en yüksek maaştan düşük maaş alan bir kişi " Satış bölümündeki herhangi bir maaştan düşük olma" koşulunu sağlayacaktır.

ANY (herhangi bir) sözcüğü yerine, tamamen eşdeğeri olan SOME sözcüğü de kullanılabilir.

5.2.4.6 ALL Sözcüğü

"Hepsi, tamamı" anlamındaki bu sözcük, SELECT komutu içerisinde belirli bir koşulu sağlayan bir grup datanın tamamınca sağlanan koşullarla ilişkili olarak kullanılır.

Örnek: Satış bölümünde çalışan ve mühendislik bölümündeki personelin hepsinden daha fazla maaş alan personelin listelenmesi

```
SELECT *
FROM personel
```

```
WHERE brut > ALL ( SELECT brut
                   FROM personel
                   WHERE bol_no = 1 )
                   AND bol_no = 2 ;
```

ya da

```
SELECT *
FROM personel
WHERE brut > ( SELECT MAX ( brut )
               FROM personel
               WHERE bol_no = 1 )
               AND bol_no = 2 ;
```

5.2.4.7 EXISTS Operatörü

" Var, mevcuttur" anlamındaki bu sözcük, SQL' de bir Boolean (Lojik, mantıksal) operatördür. İçteki SELECT komutunun sorgulaması sonucunda, en az bir tablo satırı üretilmişse EXISTS operatörü true (doğru) değerini, hiçbir tablo satırı üretilmemişse, EXISTS operatörü false (yanlış) değerini üretir.

EXISTS operatörü, AND, OR ve NOT gibi diğer mantıksal operatörlerle birlikte de kullanılabilir.

Örnek: Numarası 27 olan parçayı satan satıcılarla ilişkili tüm bilgilerin listelenmesi

```
SELECT *
FROM Satıcı
WHERE EXISTS ( SELECT *
               FROM par_sat
               WHERE sat_no = satıcı_n AND parca_n = 27 );
```

İç SELECT' te WHERE' i izleyen koşulu sağlayan satırlar, par_sat içinde bulundukça (

mevcut iseler) EXISTS operatörü true değerini verecek bu durumda dış SELECT' in WHERE koşul kısmı doğru olacağı için, o satıcı ile ilişkili bilgiler, satıcı tablosundan listelenecektir.

5.2.5 View (Bakış) Oluşturmak

5.2.5.1 Temel Tablo (BASE TABLE) ve Bakış (VIEW) Kavramı

Bu bölüme kadar oluşturulan ve kullanılan tablolar, veri tabanı ya da SQL açısından temel tablolar (base tables) adını alırlar. Bu tablolar fiziksel olarak veritabanı ortamında mevcut olan tablolardır.

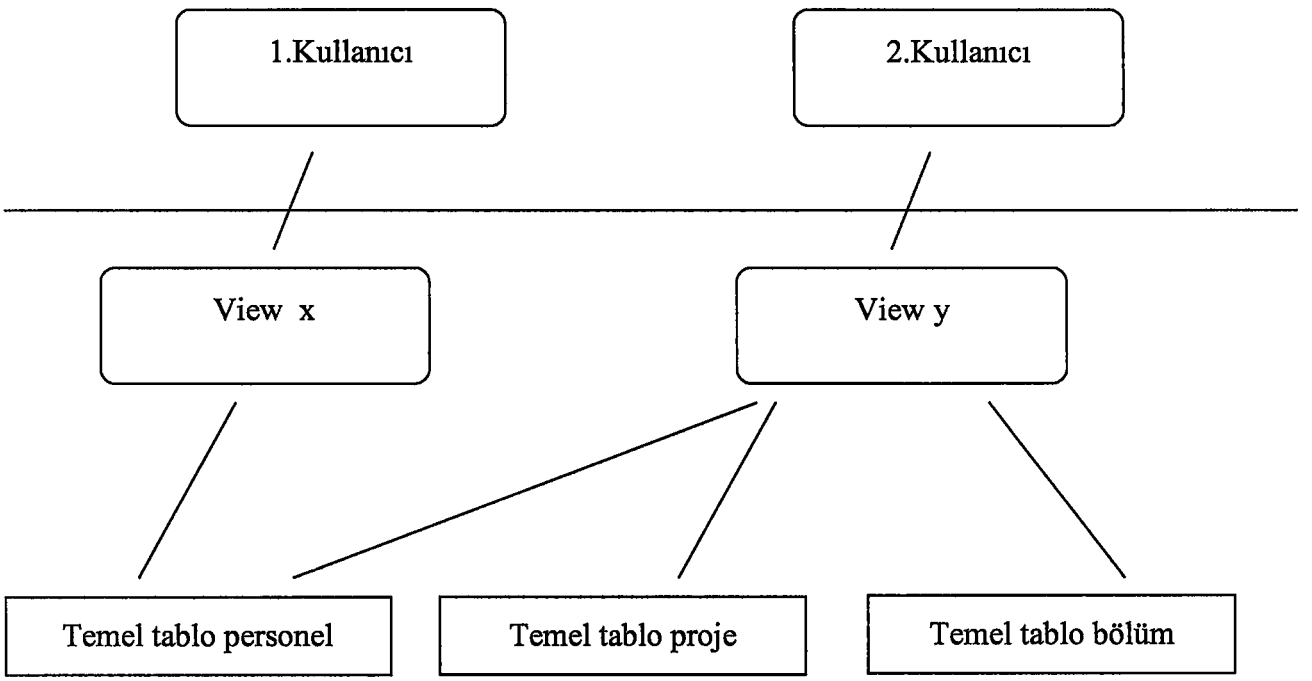
Veri tabanı kavramı içinde teorik olarak mevcut olan view (bakış) terimi farklı kullanıcıların veri tabanına nasıl baktıklarını ya da bakış açılarını anlatan bir terimdir. Bu anlamda bir kullanıcı, fiziksel olarak mevcut olan tabloların sadece bir kısmı ile ilgilenebilir.

Temel tabloların tamamı ya da bir kısmı kullanılarak elde edilen ve fiziksel olarak veri tabanında mevcut olmayan sadece nasıl elde edilebileceğine dair bilginin saklandığı tablolara bakış (view) adı verilir. Bunlar için virtüel tablo terimi de kullanılır.

SQL bu tip tabloların oluşturulma ve kullanılmasına da olanak sağlayan komutlara sahiptir.

Çizelge 5.10 da kavramsal olarak bir view' in nasıl oluşturulacağı gösterilmektedir.

Çizelge 5.10 Kavramsal olarak SQL view yapıları



Örneğin, 1. kullanıcı, personel adlı tablonun sadece bayan personeli ile ilgilenmektedir. Bu nedenle, x adlı, sadece bayan personeli içeren bir view tablosu, personel adlı temel tablodan elde edilebilir.

2.kullanıcı ise, örneğin 3.bölümün yönettiği projelerde çalışan personel listesi ile ilgilenmektedir. O nedenle personel, proje ve bölüm adlı temel tabloları kullanarak Y adlı bir view oluşturulabilir.

5.2.5.2 VIEW Oluşturmanın Yararları

Veri Güvenliği

Veri tabanı içinde bulunan tablolardaki bazı sütunlarda bulunan bilgilerin herkes tarafından görülmesi istenmeyebilir. Örneğin personelin brüt maaşlarının herkes tarafından listelenebilir olması mahsurlu olabilir. Bu durumda, personel adlı temel (base) tablodan, persview adlı bir view oluşturulabilir:

```
CREATE VIEW persview
AS SELECT sicil, sosy_g_no, ad, soyad, dog_tar, adres,  cins, bol_no, yon_s_g_n
FROM personel;
```

persview adlı view, herkesin kullanımına açık, personel adlı base tablo ise, yetkili kişiler dışındakilere, erişilemez hale getirilirse, maaşların herkes tarafından erişilebilir bilgi olması önlenmiş olur.

Bir View'den bilgi listelenmesi temel tablolardan bilgi listelenmesinden farklı değildir:

```
SELECT *
FROM persview;
```

persview' den maaşlar hariç, tüm personel bilgileri listelenecektir.

Bir temel tablodan bir view oluşturulurken, temel tablodaki aynı sütun isimlerini kullanmak zorunlu değildir.

Sorgulamanın Daha Basit Hale Gelmesi

Karmaşık sorgulamalarda, bazı SELECT komutlarının sonuçları diğer SELECT komutlarınca

kullanıldığında, sorgulamanın düzenlenmesinde yanlışlıklar yapma olasılığı artar.

Karmaşık sorgulamalar, View özelliği kullanılarak daha basit hale getirilebilir. Buradaki temel fikir şudur : View, bir sorgulama sonucu elde edilen bilgiyi isimlendirerek elde edilen bir virtüel tablodur; o halde karmaşık SELECT komutu içinde, sonucu kullanılacak başka bir SELECT komutu kullanmak yerine, bu sonucu bir view olarak isimlendirerek, view adını kullanmak.

Örnek : Satış bölümünde çalışan personelin herhangi birinden daha düşük maaş alan ve mühendislik bölümünde çalışan kişileri listeleme

```
SELECT *
FROM personel
WHERE brut < ANY ( SELECT *
                   FROM personel
                   WHERE bol_no = 2 ) AND
                   bol_no = 1;
```

(Satış bölümü kodu 2 ve mühendislik bölümü kodu ise 1 kabul ediliyor)

Bu sorunun cevabı olan tablo bir view olarak saklanırsa :

```
CREATE VIEW S1view
AS SELECT *
FROM personel
WHERE brut < ANY ( SELECT *
                   FROM personel
                   WHERE bol_no = 2 ) AND
                   bol_no = 1;
```

Bundan sonra aynı tip sorgulama için sadece

```
SELECT *
```

FROM S1view;

yazmak yeterli olacaktır.

Sadece VIEW Kullanılarak Gerçekleşebilen Sorgulamalar

Bir tablodan elde edilecek bilgiler için, iki kademeli işlem gerektiren sorgulamalarda, ilk adımda bir view oluşturup ikinci adımda esas sorgulamayı bu view yardımı ile gerçekleştirmek, çoğu kez kaçınılmaz bir durumdur.

Örnek: Her bölümde, o bölümdeki ortalama maaştan daha yüksek maaş alanları listeleme

Bu sorunun cevaplandırılması için önce her bölümdeki ortalama maaşların bulunması gereklidir.

CREATE VIEW BOL_OR_VIEW (bol_no, ort_brut)

AS SELECT bol_no, AVG (brut)

FROM personel

GROUP BY bol_no;

Daha sonra, yaratılan BOL_OR_VIEW yardımı ile (bu view, bölüm no' ları ve bölüm ortalama maaşlarını saklamaktadır) sorulan sorunun cevabı elde edilebilir.

SELECT * FROM personel

WHERE bol_no = BOL_OR_VIEW.bol_no AND brut > BOL_OR_VIEW.ort_brut;

Veri Bütünlüğünün Sağlanması

View oluşturma esnasında CHECK sözcüğünün kullanılması ile, o view' i oluştururken sağlanması gereken koşulların, daha sonra view içine veri ekleme ya da değişiklik işlemlerinde de ihlal edilmesi engellenmiş olur.

Örneğin aşağıdaki gibi bir view oluşturulsun:

CREATE VIEW UST_PER_VIEW

AS SELECT FROM personel

WHERE brut > 250000000

WITH CHECK OPTION;

Burada, brüt maaşı 250.000.000 TL' nın üstünde olan personelden oluşan bir UST_PER_VIEW adlı view oluşturulmuştur. Daha sonra bu view içine

INSERT INTO UST_PER_VIEW

VALUES (27251 , '27865427' , 'ayşe' , 'okan' , {01/05/1962} , 'Cumh.Cad.46-Taksim' ,

.F. , 130000000 , 1 , '27651112') ;

komutu ile brüt maaşı 130.000.000 TL olan bir personel eklenmek istendiği zaman şu hata mesajı alınacaktır.

Error : not enough non-null values

Eğer CHECK opsiyonu kullanılmasaydı hata mesajı alınmadan bu veri view içine yüklenecekti.

5.2.6 Tablolarda Değişiklik Yapma

5.2.6.1 Tabloya Veri Ekleme

SQL' de, mevcut bir tabloya veri eklemek için kullanılacak olan komut INSERT komutudur.

5.2.6.2 Tablo Satırlarını Silme

Bir tablonun satırlarını silmek için gerekli komut DELETE komutudur. Satır silme koşullu ya da koşulsuz gerçekleştirilebilir.

DELETE FROM personel;

Bu komut ile personel tablosundaki tüm satırlar silinecektir. Koşula bağlı satır silmek için DELETE komutuna WHERE sözcüğü eklenmeli ve bunu izleyen ifade koşulu göstermelidir.

DELETE FROM personel WHERE bol_no=2;

Bu komut ile, 2 numaralı bölümdeki personelin tümü tablodan silinecektir.

5.2.6.3 Tablo Satırlarındaki Verilerde Değişiklik Yapma – Güncelleme (UPDATE) İşlemi

Tablo satırlarında güncelleme yapmak için SQL' de UPDATE komutu kullanılır. UPDATE komutu da koşullu ya da koşulsuz olarak gerçekleştirilebilir.

Örnek: Tüm Personelin brüt maaşlarına %12 zam yapma işlemi

UPDATE personel SET brut = brut * 1.12;

5.2.6.4 Tablonun Yapısında Değişiklik Yapma

ALTER TABLE komutu ile bir tablonun yapısında değişiklik yapmak mümkündür. Standart SQL' de bu değişiklikler, tabloya yeni bir kolon ekleme (ADD sözcüğü yardımı ile) ve mevcut bir kolonun özelliklerini değiştirme (MODIFY komutu ile kolon genişliğini değiştirme ya da kolondaki verinin NULL ya da NOT NULL özelliğini değiştirme) şeklindedir.

Standart dışına çıkan birçok SQL gerçekleştirimlerinde ise ayrıca tablodan bir kolon silme (DROP), mevcut bir kolonun adını değiştirme (RENAME) ya da tablonun adını değiştirme (RENAME TABLE) özellikleri de, ALTER TABLE komutu içinde mevcuttur.

Örnek: Personel tablosuna, işe başlama tarihini belirten yeni bir kolon eklenmesi

ALTER TABLE personel

ADD is_bas_tar DATE; ya da

ALTER TABLE personel

ADD is_bas_tar DATE NOT NULL;

ADD sözcüğü ile aynı anda birden fazla kolon eklenebilir.

Örnek: Daha önce proje adlı tabloda VARCHAR (15) olarak tanımlanmış olan yer isimli alanı, 25 olarak genişletme

ALTER TABLE proje

MODIFY yer VARCHAR (25);

Mevcut bir kolon üzerinde değişiklik yapma, değişken uzunluklu bir veri tipine sahip olan kolonun genişliğini arttırma ile sınırlıdır. Bu anlamda,kolon genişliğini azaltma ya da veri tipini değiştirme mümkün değildir. MODIFY sözcüğü ile aynı anda birden fazla kolon üzerinde değişiklik yapılabilir.

Örnek: Personel tablosundan is_bas_tar kolonunu silmek

ALTER TABLE personel

DROP is_bas_tar;

Aynı anda birden fazla kolon silinebilir. Bu durumda DROP komutu içinde bunları virgülle ayırmak gerekir.

Örnek:Personel tablosunun adını elemanlar olarak değiştirme

ALTER TABLE personel

RENAME TABLE elemanlar;

Örnek: Personel tablosunda brut alanını, br_maaş olarak değiştirme

ALTER TABLE personel

RENAME brut br_maaş;

Örnek: Proje adlı tablonun tümüyle silinmesi

DROP TABLE proje ;

Veri tabanından bir tablo, DROP TABLE komutu ile silindiği taktirde, bu tablodan üretilmiş bütün VIEW' ler, bu tablodan üretilmiş eş tablolar, tablo üzerindeki indeksler ve tablo için konulmuş bütün öncelikler de sistemden silinir.

VIEW' ler üzerindeki ekleme, silme ve değişiklik işlemleri esas itibari ile tablolar üzerinde yapılan benzer işlemlerden çok farklı değildir. Fakat View' ler üzerinde bu tip işlemlerin gerçekleştirilmesinde bazı kısıtlamalar da mevcuttur. Aşağıdaki hususlara dikkat edilmesi gerekmektedir:

1-) Bir view' in güncellenebilir nitelikte olması için, bir birleştirme (join) işlemi sonucunda üretilmemiş olması gerekir. Başka bir deyişle, CREATE VIEW komutunda FROM sözcüğünü izleyen kısımda sadece bir tablo adı bulunmalıdır.

2-) View içindeki hiçbir kolon bileşik (aggregate) fonksiyonlarca üretilmiş olmamalıdır. (MAX, SUM vb. fonksiyonlar)

3-) View' in üretildiği SELECT komutunda, DISTINCT, GROUP BY ya da HAVING sözcüklerini içeren parçaların işlevleri yerine getirilmiş olmamalıdır.

Bu koşulları sağlamayan view' ler sadece okunabilir (read-only) özellikteki view' lerdir ve üzerlerinde herhangi bir değişiklik yapılamaz.

6. PHP İLE WEB PROGRAMCILIĞI

6.1 Giriş

Web' in dili olan HTML ilk çıkışından sonra günümüzün ihtiyaçlarını karşılayamaz hale gelmiştir. Bu arada Web sayfalarını bilhassa görünüm açısından etkileşimli kılmak için JavaScript dili geliştirilmiştir. Geliştirilen bu dil, inceleyicide çalışan bir script diliydi ve HTML' nin durağan kalıbı yerine etkileşimli sayfalar yaratılmasını sağlamıştır. Sonraları kullanıcıların kayıtlarını tutabilmek ve gerekli bazı bilgileri Web' e aktarabilmek için yeni programlama dilleri tasarlanmıştır. (Şamlı,2002; Öcal 2000) Örneğin;

- ASP "Active Server Page"
- JSP "Java Server Page"
- CFML "ColdFusion"
- PHP "Personal Home Page"

Geliştirilecek herhangi bir projenin bu dillerden biri ile oluşturulması mümkündür. Yalnız bu programlama dillerinin, birbirlerine göre bazı avantaj ve dezavantajları vardır.

ASP Microsoft firmasının geliştirdiği bir script dilidir ve ücretsiz olarak dağıtılmaktadır. Bazı kütüphaneleri ise ücretlidir. Aynı zamanda SQL programı olarak yine aynı şirketin MsSQL programı lisanslı olarak temin edilebilmektedir.

JSP ise sun firmasının ürettiği açık kaynak koduna dayanan yeni bir teknoloji olarak nitelendirilebilir. Henüz gelişim süreci tamamlanmamıştır.

CFML ise Allaire firmasının çıkardığı bir script dilidir. Macromedia firmasının Allaire firmasını satın almasından sonra yeni versiyonu piyasaya sürülmüştür. Ancak bu program yüksek lisans ücretiyle pazarlanmaktadır ve SQL desteği yoktur.

PHP programı ücretsiz olarak temin edilebilen bir yazılımdır. Hız açısından en büyük rakibi olarak gösterilen ASP' ye %400 lere varan farklar yaratabilmektedir. Genelde MySQL veya PostgreSQL programları ile kullanılmaktadır. Bu programlar da ücretsizdir. Bir artı özelliği de farklı işletim sistemlerinde çalışabilmesidir. Linux, Windows, Solaris ve HP işletim sistemlerinde çalışır ve birçok sunucu yazılımını destekler.

6.2 Gerekli Programların Temin Edilmesi

PHP' nin çalışması için bir Web sunucu programı ve PHP yorumlayıcısının olması yeterlidir. Ama diğer yardımcı programların da temin edilmesi yararlıdır. Gerekli olan programlar

- PHP 4
- Apache Web Server
- MySQL
- PHPMyAdmin
- PHPed
- UpMail Server

PHP programının son olarak 4.1.2 versiyonu kullanıma sunulmuştur. Bu versiyon, PHP' nin resmi sitesi olan www.php.net' den indirilebilir. Ayrıca PHP' yi hızlandırmak için çalışan Zend firmasının www.zend.com sayfasından da indirmek mümkündür. Bu sayfalarda zip uzantılı Windows versiyonu ayrıca exe uzantılı olarak da bulunmaktadır. Exe uzantılı dosya, hiç ayar gerektirmeden kurulumu gerçekleştirmektedir. Ancak bu dosya ile kurulduğunda PHP' nin özelliklerinden tam olarak yararlanılamamaktadır. Bazı kütüphaneler bulunmamaktadır.

Apache programı çoğu Linux dağıtımlarının içinde olan ve otomatik olarak sisteme yüklenen bir Web sunucu programıdır. Resmi sitesi olan www.apache.org.tr adresinden indirilebilir. Kurulumu ve ayar dosyaları daha önceki bölümlerde anlatılmıştır.

MySQL programı da resmi Web sitesi www.mysql.com adresinden temin edilebilir. Kurulumu ve ayar dosyaları daha önceki bölümlerde anlatılmıştır. Sitede zip uzantılı Windows versiyonu da bulunmaktadır. Ayrıca myodbc yazılımının da temin edilmesi mümkündür. Myodbc yazılımının kullanılabilmesi için ODBC yazılımının da güncellenmesi gerekmektedir. Güncelleme için Microsoft firmasının sayfası kullanılabilir.

PHPMyAdmin, MySQL' i Windows ortamında bir inceleyici sayfasında yönetmeye yarayan bir programdır. Herhangi bir tabloya veri girişi veya silme işi MS-DOS ortamında yapılmaktadır. PHPMyAdmin bu zorunluluğu ortadan kaldırmaktadır. Programın Türkçe de dahil çoklu dil desteği mevcuttur. Bu program www.phpwizard.net adresinden indirilebilir.

PHPed, php programlarının yazılmasında kullanılacak bir metin editörüdür. Bu metin editörü,

PHP dosyalarını tanıması, dosyalar yazılırken yardımcı olması ve ön izleme yeteneği ile kullanışlı bir programdır. www.soysal.com adresinden indirilebilir.

UpMail Server, bilgisayarda yerel bir POP3/SMTP programı kurmaktadır. Bu program aracılığı ile internete bağlı olmaksızın bilgisayara mail adresleri eklenebiliyor ve mail gönderme ve alma işlemler yapılabiliyor. PHP' nin mail işlemlerini gerçekleştirebilmek için bu programa ihtiyaç duyulmaktadır. www.upland.co.uk adresinden elde edilebilir.

ZEND Optimizer, bir dll dosyadan oluşur. PHP programlarını optimize ederek daha etkin çalışmasını sağlar. Zend firmasının sayfasından elde edilebilir (Şamlı, 2002).

6.3 PHP' nin Kurulumu

Linux işletim sisteminde PHP, dinamik yöntem ve statik yöntem olmak üzere iki değişik şekilde kurulabilir. Genelde statik yöntem kullanılmaktadır. Apache sunucu ve PHP programı ayrı ayrı kurulur. Hız açısından fazla tercih edilmez ancak PHP için bir güncelleme gerektiğinde Apache sunucusunun yeniden derlenmesi gerekmez. Sadece PHP programı derlenir.

www.php.net adresinden sıkıştırılmış olarak bulunan PHP dosyası indirilip, /tmp dizinine kopyalanır, bu dosya açılarak oluşan klasörün içine girilir.

```
gunzip -c php-4.1.2.tar.gz | tar xvf -

cd php-4.1.2
```

Daha sonra configure dosyası çalıştırılır. Bu komutun bir çok seçeneği bulunmaktadır. Bunların tümünü görmek için configure -help komutu kullanılabilir. Burada sadece ihtiyaç duyulan seçenekler eklenmiştir

```
# ./configure \

--with-mysql \

--with-pgsql=/usr/local/pgsql \

--with-apxs=/wwwr/bin/pgsql \

--with-xml \

--with-ftp \
```

Bu komut satırları çalıştırıldıktan sonra derlemeye ön hazırlık yapılır. PHP' yi derlemek için make, make install komutları ardarda çalıştırılır. Herhangi bir hata ile karşılaşılmamış ise ayar dosyasını kopyalamak için

cp php.ini-dist /usr/local/lib/php.ini

komutu çalıştırılır. Bu ayar dosyasında gerekli düzenlemeler yapıldıktan sonra, yapılması gereken işlem sistemde kurulu olan Apache sunucu programını PHP' den anlar hale getirmektir. Bunun için Apache sunucu programının bulunduğu dizindeki httpd.conf dosyasını herhangi bir metin editöründe açarak alttaki satırların başındaki # işaretini kaldırmak yeterlidir.

#AddType aplication/x-httpd-php .php

#AddType aplication/x-httpd-php-source .phps

Bu satırların başındaki diyez işaretleri kaldırıldıktan sonra bir metin editörü vasıtası ile info.php adında yeni bir dosya oluşturup içine

<?php

phpinfo();

?>

yazmak gereklidir. Daha sonra yapılması gereken Apache programının yeniden başlatılmasıdır. Bu aşamalardan sonra inceleyici açılıp adres kısmına "http://localhost/info.php" yazıldığında, sistemde kurulan PHP hakkında bilgi veren sayfa görüntüleniyorsa kurulum işlemleri hatasız olarak tamamlanmış demektir.

Dinamik yöntem hız açısından daha makul sonuçlar vermektedir. Fakat bazı dezavantajları vardır. PHP programı derlendiğinde sunucu da derlenmelidir. Ayrıca hatalı yazılmış PHP kodları PHP programını çalışmaz hale getirdiğinde web sunucu programı da çalışmaz hale gelir.

6.4 PHP' nin Ayar Dosyası (php.ini)

PHP' de yapılacak tüm ayarlamalar php.ini dosyasından yapılmaktadır. Ayarların metin dosyasından yapılması kullanıcı için zor bir işlemdir. PHP, Linux tabanlı bir programdır ve

Linux güçlü ve kararlı bir işletim sistemi olmasına rağmen kullanıcıya Windows kadar kolaylık sağlamamaktadır. Bu yüzden Windows tabanlı programlarda ayarlamalar daha kolaydır.

Php.ini ayar dosyasında yorum satırı yapan sembol noktalı virgüldür (;). Eğer bir komut etkisizleştirilmek isteniyorsa başına noktalı virgül konmalıdır.

Kurulumdaki ayarların kullanılması PHP dosyalarının yazılmasında bir sıkıntıya yol açmaz. Yine de dosyadaki ayarların önemli olanlarını bilmek gerekir.

short_open_tag: PHP kodlar <?..?> takıları arasına yazılır. Bu takılar HTML ile PHP kodlarını birbirinden ayırt eder. Bu komutun karşılığı on olmalıdır.

asp_tag: ASP kodları <%..%> takıları arasına yazılır. PHP' de bu takıların arasına yazmak isteniyorsa bu değer on olmalıdır. Varsayılan değeri off' tur.

max_execution_time: Bu parametre PHP dosyalarının maksimum çalışma süresini sınırlandırılır. Varsayılan değeri 30' dur. Böylece döngüye girmiş bir PHP kodunun sistemi çökertmesi engellenmiş olur.

memory_limit: Bu parametre PHP dosyaları çalışırken hafızada tutulabilecek maksimum büyüklüğü belirler. Varsayılan değeri 8Mb' tır.

post_max_size: Bir form aracılığı ile ziyaretçiden POST metoduyla alınabilecek maksimum bilgi miktarını gösterir. Varsayılan değeri 8Mb' tır.

extension_dir: PHP ile birlikte gelen gerekli kütüphanelerin bulunduğu dizini gösterir.

6.5 PHP Nasıl Çalışır

Ziyaretçi inceleyicisinin adres kısmında http://localhost/bak.php yazdığında, sunucu istenilen dosyanın php uzantıya sahip olduğunu görünce, bu dosyayı önce PHP.exe programına göndermektedir. PHP programıda gelen bu dosya içerisindeki <?..?> takıları arasına yazılmış kodu icra edip saf bir HTML dosyası oluşturmakta ve bu dosyayı sunucuya göndermektedir. Sunucu da bu dosyayı ziyaretciye HTML olarak göndermektedir. HTML ve PHP dosyaları arasındaki temel fark budur. HTML dosyaları talep geldiğinde direkt ziyaretçiye gönderilmektedir. PHP dosyaları ise, önce yorumlayıcıya gönderilmekte ve bunun sonucunda oluşan HTML dosyası ziyaretçiye gönderilmektedir. Bu mantık ASP,CFML,JSP ve PERL gibi sunucu tabanlı dosyalar için geçerlidir.

PHP komutları dosyada <?..?> ,<? php ...?>, ve <script language="PHP">..</script> etiketleri arasına yazılır. Gerekli ayarlamalar yapılarak <%..%> etiketi de kullanılabilir. Birinci yöntem kısa kodlarda kullanılır. İkinci yazım tarzı ise uzun kodlarda kullanılır.

6.6 PHP' nin Yazım Kuralları

PHP bir script dildir. Yani oluşturulacak dosyalar düz metin dosyalarıdır. C diline çok benzer ve çoğu komutunu da C dilinden almıştır. PHPed vasıtası ile bir dosya oluşturalım. PHP'de ekrana bir yazıyı yazdırmak için üç komut vardır. Bunlar echo, print, printf komutlarıdır. Genelde ilk iki komut kullanılır. Alttaki kod yazılıp deneme1.php olarak kaydedilip, ön izlemede veya inceleyicide görüntülenirse ekrana Merhaba Dünya yazısı çıkacaktır.

```
<?php

echo "Merhaba Dünya";

?>
```

Komutların bitiminde noktalı virgül işareti kullanılmalıdır. Bu işaret komut sonlandırmaktadır. Ayrıca komut içerisinde tırnak işaretleri de kullanılmaktadır. Aşağıdaki örnekte PHP kodu HTML kodunun içerisine yerleştirilmektedir.

```
<HTML>

<HEAD>

<TITLE>PHP içerisinde HTML </TITLE>

<META http-equiv="Content-Type" content="text/html; char-set=windows-1254" >

</HEAD>

<BODY>

<?php

print "<H1 align=center>Merhaba Dünya</H1>";

</BODY>

</HTML>
```

Yukarıdaki örnekte, print komutu ile HTML komutları yazdırılmaktadır. Fakat bu yapılırken

HTML komutunda kullanılması gereken çift tırnak işaretlemesi kullanılmamaktadır. Aslında <H1 align="center"> şeklinde kullanılmalıydı. Bunun sebebi print komutunda kullanılan çift tırnaktır. Eğer HTML komutu doğru olarak çift tırnaklarla birlikte kullanılsaydı PHP yorumcusu ikinci olarak yakaladığı çift tırnak işaretini komut sonu olarak görüp noktalı virgül isteyecek ve aksi takdirde hata verecekti. Bunu önlemenin bir yolu print komutunda tek tırnak kullanmak diğer yolu da anlamsızlaştırma işareti olan "\" işaretini kullanmaktır.

print "<H1 align=\"center\"> Merhaba Dünya</H1>

yazımında ortadaki çift tırnaklar, PHP yorumlayıcısı tarafından dikkate alınmaz. HTML' de yorum satırı <!--yorum satırı-- > şeklinde kullanılır. PHP kodlarına yorum eklemek için üç yol vardır. Bunlar

<?php

// Birinci yorum metodu

İkinci yorum metodu

/* Üçüncü yorum

metodu */

?>

Yorum satırları PHP tarafından dikkate alınmaz. Bunların amacı yazılan programların okunabilirliğini arttırmak ve bir müdahale gerektiğinde bunu kolaylaştırmaktır. Açıklamalar gereksiz yere bütün kodların incelenmesini engeller.

6.7 PHP' de Değişken Tanımlaması

Değişkenler programcılığın yapı taşlarından biridir. PHP' de değişken tanımlamak için dolar simgesinden yararlanılır. Değişken isimleri rakamlarla başlamamalı ve Türkçe karakterler içermemelidir. Örneğin ;

$sair;

$rakamlar;

gibi. Değişkene değer aktarmak istendiğinde,

$sair="Mehmet Akif Ersoy";

```
$rakamlar="0123456789";
```

```
<?php
$sehir="Rize";
$deger="2480";
print "Türkiye'nin $deger metrekare yağış alan şehri $sehir'dir.";
?>
```

Bu dosya çalıştırıldığında ekrana çıkacak yazı;

Türkiye'nin 2480 metrekare yağış alan şehri Rize'dir.

Örnek:

```
$isim="İlker";
$deger="isim";
print $$deger;
```

Burada ekrana "İlker" yazısı çıkacaktır. Bunun sebebi değer değişkeni yerine konduğunda başındaki fazladan $ işareti nedeniyle yeni bir değişken oluşmasıdır. PHP yorumlayıcısı yeni oluşan $isim bilgisini de tekrar değişken olarak algılar ve dosyada bu değeri arar.

Dikkat edilmesi gereken bir nokta da bir değişken içindeki bilgi değiştirilmek istendiğinde öncelikle bu değişkenin içerisindeki bilginin silinmesi gerektiğidir.

```
$il="RİZE";
print $il;
$il=NULL;
$il="ISTANBUL";
print $il;
```

NULL parametresi PHP'ye 4 versiyondan sonra eklenmiştir. Daha önceki versiyonlar tarafından desteklenmez. Değişkenlerin kullanımı ile ilgili olarak dikkat edilecek bir hususta tek tırnak ve çift tırnak ayırımıdır. Çift tırnak içerisinde kullanılan $ sembolü değişken olarak algılanır fakat tek tırnak içerisinde kullanılırsa değişken olarak algılanmaz.

6.8 PHP' de Matematiksel İşlemler

PHP' de matematiksel işlemlere tam destek verilmektedir. Sinüs, kosinüsten mutlak değer almaya kadar birçok matematiksel işlem desteklenmektedir.

6.8.1 Operatörler

Çizelge 6.1 PHP' de operatörler

Operatör	Tanım
%	Modul alma işlemi için kullanılır.
*	Çarpma işlemi için kullanılır.
+	Toplama işlemi için kullanılır.
-	Çıkarma işlemi için kullanılır.
/	Bölme işlemi için kullanılır.

Dört işlem operatörleri matematikte kullanılan simgelerle gösterilmektedir. PHP' de matematiksel işlemleri yaparken dikkat edilecek husus ifadenin parantezler içerisinde yer almasıdır. Aksi takdirde yazı metin gibi algılanır ve yazı olarak ekrana yazılır.

```
print (15+15); // 30

print (5 * 7);  // 35  vs...
```

% işareti bölme işleminde kalan değeri verir.

```
print (15%4); // 3
```

Aşağıda örnek işlemler ve çıktılar yer almaktadır.

```
$a1=10;

$a2=15;

$a3=20;

print "$a1 + $a3";                    // 10 + 20

print ($a1 + $a2 * $a3);              // 310
```

6.8.2 Karşılaştırma Operatörleri

Çizelge 6.2 PHP' de Karşılaştırma Operatörleri

Operatör	Tanım

==	Eşitse
!=	Eşit değilse
===	Aynı ise
>	Büyükse
<	Küçükse
<=	Küçük eşitse
>=	Büyük eşitse
&&	Ve
||	Veya

Burada bulunan karşılaştırma operatörleri yalnızca matematiksel ifadelerde değil aynı zamanda program denetimi sırasında da kullanılır. Örneğin

```
<?php
if($ortalama) {
    $toplam=($s1+$s2+$s3)/3;
    if($toplam>=50) }
        print "Ortalamanız <b>$toplam</b>, bu dersten geçtiniz.";
    } else {
        print "Ortalamanız <b>$toplam</b>,  dersten kaldınız.";
    }
} else {
print '
<form action="ortalama.php" method="GET">
1.Sınav: <input type="text" name ="s1"><br>
2.Sınav: <input type="text" name ="s2"><br>
3.Sınav: <input type="text" name ="s3"><br>
<input type="submit" name="ortalama" values="ORTALAMA" >';
} ?>
```

Bu kod önce kullanıcıdan üç sınav notu alacak ve daha sonra bu notların aritmetik ortalaması 50 veya üstünde ise geçtiniz, altında ise kaldınız şeklinde mesaj verecektir. Burada if

deyiminde büyük eşitse karşılaştırma operatörü kullanılmaktadır.

6.8.3 PHP' de Matematik Fonksiyonları

Çizelge 6.3 PHP' nin Matematik Fonksiyonları Listesi

Fonksiyon	İşlevi
abs()	Mutlak değer
acos()	Yay kosinüsü
asin()	Yay sinüsü
atan()	Yay tanjantı
atan2()	İki değişkenin yay teğeti
base_convert()	Keyfi esaslar arasında sayıyı çevir.
bindec()	Ondalığa ikilik
ceil()	Yukarıya yuvarla
cos()	Kosinüs
decbin()	İkiliğe ondalık
dechex()	Ondalığa onaltılık
decoct()	Ondalığa sekizlik
deg2rad()	Dereceyi radyana çevir.
floor()	Aşağıya yuvarla
getrandmax()	En geniş mümkün rastgele değeri göster
hexdec()	Onaltılığa ondalık
lcg_value()	Doğrusal congruential jeneratörünü birleştir.
log()	Doğal logaritma
log10()	On tabanına göre logaritma
max()	En yüksek değeri bul
min()	En düşük değeri bul
pi()	Pi sayısı
pow	Üssel ifade
rad2deg	Radyanı dereceye çevir.
rand()	Rastgele sayı üret
round()	Yuvarla
sin	Sinüs
sqrt()	Karakök
tan()	Teğet

PHP' de birçok matematik fonksiyonu yer almaktadır. Kullanımı print abs(-30) şeklindedir. Bu ifade ekrana 30 yazdıracaktır. Matematiksel işlemlerde dikkat edilecek nokta ondalık gösterimi için nokta işaretinin kullanılmasıdır. print floor(99 * 0.5) burada da ekrana 49 yazılmaktadır (Şamlı 2002).

6.9 PHP' de Mantıksal Program Denetimi

Programcılığın en önemli konusu mantıksal denetimlerdir. Mantıksal denetleme; programa bir

şart verilerek, bu şartın gerçekleşmesi veya gerçekleşmemesine göre programın yönlendirilmesi ve buna göre farklı işlemlerin yerine getirilmesidir. Belli bir şartın denetlenmesi için değişik komutlar kullanılabilir.

6.9.1 If Deyimi

PHP dilinde en çok kullanılan mantıksal denetleme deyimidir. Bu deyimin yazılışı

if (koşul) { koşul doğru ise işlenecek komutlar) }else{ koşul doğru değil ise işlenecek komutlar}

Örneğin

```
if (15 = = 12) { print "Bu iki sayı eşittir.";
} else { print "Bu iki sayı eşit değildir.";
}
```

Burada koşul doğru olmadığı için ekrana "Bu iki sayı eşit değildir." yazılır. Deymin koşul kısmında aynı anda birden fazla koşulda sorgulanabilir. if (5 > 4 && 4 > 2) şeklinde. Koşullar && ile birleştirildiğinden her iki koşul aynı anda doğru olursa ilk komutlar aksi takdirde else ile verilen komutlar yürütülür. PHP dilinde else yerine elseif deyimi de kullanılabilir. Böylece şart sağlanmadığı zamanlarda else komutları yürütülmeden başka koşullar kontrol edilebilir.

```
if (koşul veya koşullar) {doğru ise yürütülecek komutlar}
elseif (başka koşul veya koşullar) { bu koşul doğru olduğunda yürütülecek komutlar}
else {koşullardan hiçbiri gerçekleşmezse yürütülecek komutlar}
```

İf deyimin bir başka yazım şeklide

```
if ( 45 > 44 ) :
print "Bu koşul doğrudur";
else:
print "Bu koşul doğru değildir";
endif;
```

if deyimi ile birlikte kullanılabilecek birkaç deyim vardır.

isset() : Bir değişkenin içerisinde değer olup olmadığını kontrol eder. Değer varsa doğru (TRUE), değer yoksa yanlış (FALSE) sonucunu verir. if (isset($degisken)): şeklinde kullanılır.

empty() : isset() komutunun tersi olarak çalışır. Değer varsa yanlış (FALSE), değer yoksa doğru (TRUE) sonucunu verir. if (empty($degisken)): şeklinde kullanılır.

intval() : Belirtilen sayıyı tam sayı olarak alır.

is_integer: Değişkenin tam sayı olup olmadığını sınar.

6.9.2 Switch deyimi

PHP dilinde program denetiminde kullanılan bir deyim de switch deyimidir. Farklı koşullara göre farklı komutların yürütüleceği durumlar if deyimleri ile yapılabilse de switch deyimi daha okunabilir ve hızlıdır. Yazımı;

```
switch (değişken) {
case "birinci koşul";
        koşul doğru ise işlenecek komutlar;
        break;
case "ikinci koşul";
        koşul doğru ise işlenecek komutlar;
        break;
.....
default;
hiçbir koşul sağlanmadığında işlenecek komutlar;
}
```

6.9.3 include()

Bu komut harici dosyaları kendi içerisinde işleyip istenilen dosyaya monte eder. Böylece çeşitli dosyalarda kullanılacak aynı ifadeler tek bir dosyada tutularak dosyaların

yönetilmesinde kolaylık sağlar. Harici dosya nasıl hazırlandıysa monte edilen dosyaya aynen yansır. Eğer dosyaya PHP komutları konulacaksa, PHP ayraçlarının da kullanılması gereklidir. Bu durumda PHP yorumlayıcısı, harici dosyayı da PHP dosyası olarak işleyecektir. Yazımı <? include ("harici.inc"); ?> şeklindedir.

6.10 Döngüler

Döngüler programlama dillerinin vazgeçilmez araçlarıdır. Döngüler bir koşula bağlı olarak ya da koşula bağlı olmaksızın belirli bir sayıda istenildiği kadar tekrarlanan süreçlerdir.

6.10.1 While Döngüsü

En çok kullanılan döngüdür. Bir şartı karşılaması veya karşılamaması durumunda döngü işler. Döngüler kurulurken sonsuz döngü oluşturmamaya dikkat etmek gerekir. Böyle bir durum oluşursa program kilitlenir. While komutunun yazımı iki şekilde olabilir:

```
while ( koşul ) {
 koşulun doğru olması durumunda işlenecek komutlar
}
```

alternatif yazımı ise;

```
while ( koşul ) :
        koşulun doğru olması durumunda işlenecek komutlar
endwhile;
```

```
<?php
print "Matematikteki rakamlar:<br>\n";
$i=0;
while ( $i <= 9 ) {
      print "<b>$i</b><br>\n";
      $i++;
}
```

?>

Bu kod ekrana rakamları yazar. i değişkeni döngünün her tekrarlanışında bir artırılıyor ve koşulda kontrol ediliyor. Böylece i değişkeni 10 olduğunda koşul artık sağlanmıyor ve ekrana rakamlar yazıldıktan sonra döngü sonlanıyor.

6.10.2 Do .. While

While döngüsüne benzer yapıdadır. Bir şartı karşılaması veya karşılamaması durumunda döngü işler. While döngüsünden farkı çalışma tarzıdır. While döngüsünde önce koşul kontrol edilir ve sağlanıyorsa komutlar yürütülür. Bu döngüde ise önce komutlar yürütülür sonra koşul kontrol edilir. Bu da koşul sağlanmazsa bile döngü içindeki komutların en az bir kez çalıştırılacağı anlamına gelir. Yazımı;

```
do {
    koşul doğru ise yapılacak işler
}
while (koşul)
```

6.10.3 For Döngüsü

for döngüsü de while döngüsü gibidir. Farklı olarak burada sayaç değişkeni kullanılır. Yazımı

```
for (değişken; koşul; artma miktarı) {
    koşul doğru ise işlenecek komutlar
}
alternatif yazım
for (değişken; koşul; artma miktarı) :
    koşul doğru ise işlenecek komutlar
endfor;
```

while örneğimizi bu döngü komutu ile yazarsak;

```
<?php
print "Matematik'te ki rakamlar:";
```

```
for ($sayac=0; $sayac <= 9; $sayac++){
    print $sayac;
}
?>
```

herhangi bir döngüden çıkış için "break" komutu kullanılır. Döngünün belirli bir kısmının yürütülmesini engellemek ama döngüden çıkmamak için continue komutu kullanılır. PHP yorumlayıcısı bu komuta rastladığında bundan sonraki komutları işlemez ve döngününün başına döner.

6.11 Fonksiyonlar

Fonksiyonlar bir kere tanımlanır ve bir çok kereler, ihitiyaç olan noktalarda hizmete çağrılır. Bir programlama dilinin önemli parçalarıdır. Genellikle bir iş veya hesaplama yapar ve bunun sonucunda ortaya yeni bir değer çıkartırlar. Bu değere fonksiyonun geri dönüş değeri denir. PHP' de kullanıma hazır bir çok fonksiyon hazır gelir. echo() ve print() komutları aslında bu tür fonksiyonlardır. PHP kullanıcının kendi fonksiyonlarını tanımlamasına imkan sağlar. Tanımlanacak fonksiyonun kendini çağıran komuttan alması gereken değerler var ise bu değerlere argüman denir. Fonksiyon değer almayacaksa bile içi boş parantezler kullanılması gerekir. PHP'de fonksiyon yazımı;

```
function fonksiyon_adı( argüman1,argüman2,..,argümanN) {
        yürütülecek komutlar
}
<?php
function yazdirH2($metin) {
    print("<H2>$metin</H2>\n");
}
.....
yazdirH2("Bu H2 başlık")
?>
```

Burada yazdirH2 fonksiyonu kendisine argüman olarak gönderilen metni H2 başlık olarak ekrana yazdırır. Geleneksel olarak kullanılacak fonksiyonlar dosyasının başında yer alır. Daha sonra nereden isteniyorsa çağırılabilir. Fonksiyonlar da başka fonksiyonları çağırabilirler. yukarıdaki fonksiyona metnin direkt kendisi yazılarak bilgi gönderilebileceği gibi bir değişken vasıtası ile de değer geçirilebilir. Burada tanımlanan fonksiyon kendisi işlem yapan bir fonksiyondur ve geriye değer çevirmez. Alttaki fonksiyon örneğinde ise gönderilen argümanlarla işlem yapılıp geriye bir değer gönderilmektedir.

```
function topla($deg1, $deg2) {
        $sonuc =  $deg1 + $deg2;
        return $sonuc;
}
```

Bu fonksiyon kendisine argüman olarak gelen iki değişkeni topluyor ve sonucu return komutu vasıtası ile çağrıldıgı komuta gönderiyor. Bu fonksiyon şu şekilde kullanılabilir:

```
print topla($sayi1,$sayi2);
```

fonksiyon argümanlarına varsayılan değer atanabilir.

```
function topla($deg1, $deg2=5) {
        $sonuc =  $deg1 + $deg2;
        return $sonuc;
}
print topla($sayi1);
```

bu tanımlama ile ikinci değere herhangi bir atama yapılmadığında bu değerin 5 olacağını anlatır.

6.11.1 Değişkenlerin Kapsamı

Fonksiyonlarla ilgili önemli bir nokta da fonksiyonlarda kullanılan değişkenlerin kapsama alanıdır. Bir fonksiyonun değişkenleri sadece o fonksiyon için geçerlidir. Başka hiçbir noktadan bu değişkene erişilemez veya bu değişkenle ilgili işlem yapılamaz.

```
<?php
```

```
$metin = "Başkalarına yararlı olmanın sınırı yoktur."

function yazdir () {

      print("<H1>İşte metin:$metin</H1>";

}

?>
```

Bu fonksiyonun ekrana $metin değişkenin değerini yazması beklenebilir. Fakat üstte tanımlanan ve içine bilgi konulan $metin değişkeni ile fonksiyon içerisindeki $metin değişkeni farklıdır. Fonksiyon kendi dışında tanımlanmış olan değişkene erişemez dolayısı ile ekrana bu değişkenin içeriği yazılamaz. Bir fonksiyonun dışında tanımlanmış değişkenlere erişebilmesini sağlamak için global deyimini kullanırız.

```
function yazdir() {

    global $metin;

   print("<H1>İşte metin:$metin</H1>");

}
```

Bu koddaki global deyimi fonksiyonun kendi dışında tanımlananan $metin değişkenine erişebilmesini dolayısı ile ekrana içeriğinin yazabilmesini sağlar. Dikkat edilecek bir husus ta fonksiyonun artık bu değeri değiştirebileceğidir. Değişkenlerle ilgili bir başka önemli deyim de static deyimidir. Fonksiyonlardaki değişkenler fonksiyon sona erdiğinde yok olur. Bunların yok olması istenmiyorsa static deyimi kullanılabilir. Kullanımı

```
function saydir () {

static $sayi=0 ;

.....

}
```

şeklindedir. Bu şekilde bu fonksiyon her çağrıldığında $sayi değişkeni yeniden tanımlanmaz ve içindeki değer sıfırlanmaz.

6.12 Dizi-Değişkenler, Nesneler

PHP' de, diğer programlama dillerinde olduğu gibi, kullanılan veriler değişkenlerde tutulur. Aynı tipte verileri tutmak ve bunlar ile benzer işlemler yürütmek gerektiğinde bu değişken tanımlaması yetersiz kalır. İşte bu tür grup bilgilerini topluca tutmada kullanılan değişkenlere dizi-değişken (array) denir.

PHP ayrıca nesne-yönelimli diller kapsamına girer. Bunun nedeni kendi değişkenleri ve kendi fonksiyonları olan nesne (object) oluşturma imkanı sunmasıdır.

6.12.1 Dizi-Değişkenler

PHP'nin bu amaçla sağladığı araç, çok boyutlu dizi değişkeni oluşturma aracıdır. Burada endeks adları bir kelimeden fazla ise tırnak içine almak gerekir.

```
<?php
$öğrenciler=array(
array( adi => "Özbay ", soyadi=>"Altun"),
array( adi => "Hasan ", soyadi=>"Civelek"),
.....
array( adi => "Özbay ", soyadi=>"Tabak"),
);
?>
```

Kullanımı;

```
print $ogrenciler[0][adi];
```

Ayrıca dizi-değişken tanımlamanın bir yolu da

```
$deneme[ ] = "Ali ";
$deneme[ ] = "Mehmet ";
$deneme[ ] = "Ahmet";
```

dizi değişkeni kullanırken elemanların indeksi 1 den değil 0 dan başlar. Yani dizinin 1.elemanı 0 nolu indekse sahiptir. Bu durumda 3. elemanın indeksi de 2 olur.

Döngüler konusunda atlanan bir döngü komutu daha vardır. Bu komut dizi değişkenler ile kullanılan foreach komutudur. kullanımı

```
$deneme = array ('a','b','c','ç','d','e','f','g');

foreach ( $deneme as $deger ) {

        echo "$deger <br>";

}
```

6.12.2 Nesneler

Nesne, kendi değişkenleri olan ve icra edeceği komutlardan oluşan fonksiyonları ile bir bütündür. Nesne bir kere tanımlandıktan sonra istenildiği kadar örneği oluşturulabilir. Bir nesne oluşturmak için onu tanımlamak gerekir. Nesne tanımlama PHP'de class deyimi ile yapılabilir.

```
<?php

class ogrenci {

// özellikleri tanımlama

  var $adi;

  var $soyadi;

  var $sinav1;

  var $sinav2;

  var $not;

// metodların belirlenmesi

funciton adi_belirle($n) {

        $this->adi = $n;

}

funciton soyadi_belirle($n) {

        $this->soyadi = $n;
```

```
}

.....

}

// Başka program kodları

$ogr1 = new ogrenci();

$ogr->adi_belirle ("Şahika");

$ogr->soyadi_belirle ("Tabak");

?>
```

6.13 Formlar

Internet' te Form, bir Web sayfasının ziyaretçiden veri alabildiği ve bunları Web sunucusuna ulaştırabildiği başlıca araçtır. Form, ziyaretçiden istenilen bilgilerin yanısıra ziyaretçinin bilgisayarından Web sunucusu bilgisayara, daha birçok bilgiyi de beraberinde getirir. Web programcısının ve Web tasarımcısının bilgileri bilmesi gereklidir. Söz gelimi, ziyaretçinin inceleyici türü ve sürümü belirlenerek uygun sayfaya yönlendirme yapılabilir. Ziyaretçiden istenilen bilgilerin sunucuya ulaştığında nerede ve hangi değişkende tutulduğuna kadar, gerekli birçok bilgi sunucu çevre değişkenleri ve sunucu değişkenleri denilen dizilerde bulunur. Web Server' ı ve programcıya verdiği bilgileri daha yakından tanımak gereklidir.

Alttaki örnek program çalıştırıldığında

```
<HTML>

<HEAD>

<TITLE>PHP'de Nesneler</TITLE>

<meta http-equiv="content-type" content="text/html; charset=ISO-8859-9">

<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">

</HEAD>

<BODY>

<?php
```

```
foreach ($GLOBALS as $anahtar=>$deger ) {
        print ($anahtar . " = " . $deger . "<br>");
}
?>
</BODY>
</HTML>
```

PHP' nin daima varolan $GLOBALS dizisinin üyeleri görüntülenir. Program çalıştırıldığında sisteme ve Web sunucu programına bağlı olmak üzere, ekranda birçok değişken görüntülenir. Bunlar arasında bütün HTTP Server programları için ortak ve Web programcısı için önemli değişkenler şunlardır:

HTTP_ENV_VARSHTTP: Sunucu programın çalışmakta olan PHP dosyası için oluşturduğu çevre değişkenlerinin yazılı olduğu dizi değişken. Bu değişkenin içinde şu unsurlar bulunur:

- HOSTNAME: Sunucu' nun IP adresi
- SHELL: Unix sisteminde kullanılan Shell programı
- HOSTTYPE: Sunucu' nun adı ve sürümü
- OSTYPE: Sunucu' nun işletim sistemi
- HOME: Çalışan programın kök dizini
- PATH: Çalışan programın Sunucu' daki yolu

HTTP_SERVER_VARS: Sunucu programın çalışmakta olan PHP dosyasına sunduğu bazı bilgilerin bulunduğu dizi değişken. Bu değişkenin içinde şu unsurlar bulunur:

- PHP_SELF: Çalışan PHP programının bulunduğu dizin ve adı
- PATH_TRANSLATED: Çalışan PHP programının fiziksel yolu

HTTP_GET_VARS: Bir Form' dan GET metoduyla alınan bilgilerin anahtar = değer çiftleri olarak kaydedildiği dizi değişken

HTTP_POST_VARS:Bir Form' dan POST metoduyla alınan bilgilerin anahtar = değer çiftleri olarak kaydedildiği dizi değişken

HTTP_USER_AGENT: Ziyaretçinin bilgisayarında kurulu Internet inceleyici programı

QUERY_STRING: Form ile bilgi alırken GET metodu kullanıldığı takdirde, inceleyicinin göndereceği bilgilerin tutulduğu değişken

REMOTE_ADDR: Ziyaretçinin bilgisayarına ISS tarafından atanmış IP adresi

REQUEST_METHOD: Form ile gelen bilgilerin gönderildiği metot: GET veya POST

REQUEST_URI: O anda çalışmakta olan PHP dosyasının adı ve varsa bu ada eklenmiş Query_String

SCRIPT_FILENAME: O anda çalışmakta olan PHP programının dosya adı

SCRIPT_URI: O anda çalışmakta olan PHP programının tam URL adresi

SERVER_ADDR: Sunucunun IP adresi

SERVER_PROTOCOL: Sunucunun HTTP protokolünün sürümü

6.13.1 Form' dan GET Metoduyla Gelen Bilgiler

Ziyaretçilerin ne tür inceleyici kullandıklarını HTTP_USER_AGENT değişkeninin değerini alarak ve bu değerin içinde belirli anahtar kelimeler aratılarak bulunabilir. Form ile gelen bilgiler, GET metodu ile alınıyorsa, hem QUERY_STRING, hem de HTTP_GET_VARS dizisine kaydedilir. POST metoduyla alındığında bilgiler HTTP_POST_VARS değişkenin değerleri arasında bulunur.

Basit bir HTML Form' u ile bilgilerin alınması;

```
<HTML>

<HEAD>

<TITLE>PHP'de Formlar</TITLE>

<meta http-equiv="content-type" content="text/html; charset=ISO-8859-9">

<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">

</HEAD>

<BODY>

<FORM ACTION="formlar02_isle.php" METHOD="GET">
```

Adınız, Soyadınız: <INPUT TYPE="TEXT" NAME="adi">

<INPUT TYPE="SUBMIT" VALUE="Gönder">

<INPUT TYPE="RESET" VALUE="Vazgeç">

</FORM> </BODY>

</HTML>

Ziyaretçinin "Gönder" düğmesini tıklamasıyla birlikte Form' un içerdiği bilgilerin METHOD parametresinde yazılı olan GET yöntemiyle, ACTION parametresindeki programa gönderilir. "Gönder" düğmesi tıklandığında URL adres kutusunda aşağıdaki ifade yer alır.

http://server/formlar02_isle.php?adi=Muharrem+Ta%E7

Bu, HTTP protokolüne göre GET yoluyla bilgi göndermekte kullanılan yöntemin tam bir örneğidir: İnceleyiciler, GET yoluyla bilgi göndereceği zaman, Form' daki bütün bilgileri URL-Encoding denen sistemle kodlar. Web sunucu, bu bilgileri alınca, önce kendi oluşturduğu bazı değişkenlere (hem QUERY_STRING, hem de HTTP_GET_VARS dizisine) yazar ve sonra URL hanesinde adı yazılı olan programa (sayfaya) verir.

formlar02_isle.php sayfası

<HTML>

<HEAD>

<TITLE>PHP'de Formlar</TITLE>

<meta http-equiv="content-type" content="text/html; charset=ISO-8859-9">

<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">

</HEAD>

<BODY>

<?php

print ("Sayın $adi\n\n");

?>

</BODY>

```
</HTML>
```

Yukarıdaki örnekte PHP programının, alan adlarını değişken adı, bunların karşısında yazılı olan verileri de bu değişkenin değeri sayması kullanılıyor. Aynı işlem diziler vasıtası ile yapılırsa

```
<HTML> <HEAD>
<TITLE>PHP'de Formlar</TITLE>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-9">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
</HEAD>
<BODY>
<?php
foreach ($HTTP_GET_VARS as $anahtar=>$deger ) {
        print ("<b>$anahtar = $deger <br>\n");
}
?>
</BODY>
</HTML>
```

6.13.2 Form' dan POST Metoduyla Gelen Bilgiler

HTML Form etiketinin METHOD parametresinin değeri GET olabildiği gibi POST da olabilir. HTTP sunucusu bu yöntemle gelen bilgileri $HTTP_POST_VARS dizi-değişkeninde tutar

```
<FORM ACTION="formlar02a_isle.php" METHOD="POST">
```

Aynı şekilde son Form işleme programında bu değişiklik yapılırsa

```
foreach ($HTTP_POST_VARS as $anahtar=>$deger ) {
```

ikinci dosya aynı şekilde çalışır.

HTTP açısından GET ile POST' un tek farkı gelen değerlerin nerede nasıl tutulduğundan ibaret değildir. GET yönteminde, bir inceleyicinin sunucuya gönderebileceği verinin uzunluğu, Sunucunun ayarlarına bağlı olmak üzere, sınırlıdır. Oysa POST ile alınacak veri miktarı, sadece sunucunun bulunduğu bilgisayarın sabit disk alanıyla sınırlıdır. Bir başka fark, inceleyicinin GET yoluyla gönderdiği verilerin (ve bu arada ziyaretçinin parola olarak yazdıklarında ekrana yıldız olarak çıkan metinler dahil) tümü, sunucuya, URL-kodlanmış metin olarak, inceleyici'nin URL adres hanesine de yazılmasıdır. Birçok kullanıcı için bu bir güvensizlik belirtisi sayılır. Bu iki unsur Formlarda metod olarak GET yerine POST kullanmanın daha yerinde olduğunu gösterir.

6.14 Dosya İşlemleri

PHP' de dosya okutma ve yazdırma işlemleri dolayısıyla bir dosyanın varlığı veya yokluğu, ya da bir dosyaya ait sanılan ismin bir klasöre ait olması, programın sağlıklı işleyebilmesi açısından büyük önem taşır. PHP bu amaçla bize birkaç kullanıma hazır fonksiyon sağlamaktadır.

6.14.1 Dosyanın Varlığının Kontrolü (file_exits())

Bir dosyanın var olup olmadığını denetleyen bu fonksiyon, dosya varsa true/doğru, yoksa false/yanlış sonucunu verir. Örnek:

```
if ( file_exits ( "bir_dosya.txt" ) )

print ("Dosya var!");
```

Dosya yoksa, program "Dosya var!" yazmadan yoluna devam edecektir.

6.14.2 Dosya / Dizin Kontrolü (is_file() / is_dir())

Kimi zaman klasörler de tıpkı dosyalar gibi adlandırılabilir. Bir dizinde görülen ismin gerçekten bir dosyaya ait olup olmadığı bu fonksiyonla sınanır. Sınama doğru, yani isim bir dosyaya ait ise fonksiyon true/doğru, değilse false/yanlış sonuç verir. Örnek:

```
if ( is_file ( "bir_dosya.txt" ) )

print ("Bu bir dosyadır!");
```

İsim bir dosyaya ait değilse program "Bu bir dosyadır!" yazmadan yoluna devam edecektir. Sınama ismin bir klasöre ait olup olmadığına bakılarak ta yapılabilir. Bu durumda is_dir() fonksiyonunu kullanırız. İsim bir dizine aitse fonksiyon true/doğru, değilse false/yanlış sonuç

verir.

6.14.3 Dosyanın Okunabilirliğinin Kontrolü (is_readable())

Programda kullanmaya karar vermeden önce bir dosyanın erişilebilir ve PHP tarafından okunabilir olup olmadığını sınayan bu fonksiyon, dosya okunabilir ise true/doğru, değilse false/yanlış sonuç verir. Örnek:

```
if ( is_readable ( "bir_dosya.txt" ) )

print ("Bu dosya okunabilir!");
```

Dosya okunabilir değilse program "Bu dosya okunabilir!" yazmadan yoluna devam edecektir. (Unix ortamında varlığı görülebilen her dosyanın okuma izni bulunmayabilir.)

6.14.4 Dosyanın Yazılabilirliğinin Kontrolü (is_writable())

Bir dosyanın yazılabilir olup olmadığını sınayan bu fonksiyon, dosya yazılabilir ise true/doğru, değilse false/yanlış sonuç verir. Örnek:

```
if ( is_writable ( "bir_dosya.txt" ) )

print ("Bu dosyaya yazılabilir!");
```

Dosya yazılabilir değilse program "Bu dosyaya yazılabilir!" yazmadan yoluna devam edecektir. (Unix ortamında varlığını görebildiğimiz hatta okuyabildiğimiz her dosyanın yazma izni bulunmayabilir.)

6.14.5 Dosyanın Çalıştırılabilirliğinin Kontrolü (is_executable())

PHP programında kimi zaman sunucunun kullanımına izin verdiği harici programlar çalıştırmak gerekebilir. PHP programının düzgün işlemesi bu harici programa bağlı olabilir. Bir dosyanın çalıştırılabilir olup olmadığını sınayan bu fonksiyon, dosya çalıştırılabilir ise true/doğru, değilse false/yanlış sonuç verir. Örnek:

```
if ( is_executable ( "bir_dosya" ) )

print ("Bu dosya çalıştırılabilir!");
```

6.14.6 Dosya Oluşturma ve Silme

PHP ile yapılabilecek önemli dosya işlemlerinin başında, olmayan bir dosyayı oluşturmak ve olan bir dosyayı silmek gelir. PHP' nin dosya oluşturma komutu touch() fonksiyonudur. Bu

fonksiyona oluşturulması istenen dosyanın adı argüman olarak gönderilmelidir.

```
<?php
$dosya_dizin = "/inetpub/wwwroot/";
touch ("$dosya_dizin/yeni_belge.txt");
print ("yeni_belge adlı bir dosya oluşturuldu!");
?>
```

Bu programı gerçek sunucuda çalıştırabilmek için yazma/okuma izni bulunan ve Web sunucunun erişebileceği bir dizinin adı verilmelidir.

PHP ile mevcut bir dosyayı silmek için unlink() fonksiyonu kullanılır. Bu fonksiyon da silinecek dosyanın adı ile birlikte yolunu ister. Örnek:

```
<?php
$dosya_dizin = "/wwwroot/mycgiserver.com/members/uNhM13Qnm/";
unlink ("$dosya_dizin/yeni_belge.txt");
print ("yeni_belge adlı dosya silindi!");
?>
```

Bu komut Windows sistemlerinde işlemeyebilir.

6.14.7 Dosya Açma ve Kapama

PHP' de bir dosyanın içeriğini alarak sayfalarda kullanma veya bir dosyanın içeriğini değiştirmek gibi işlemler için önce dosyanın açılmış olması gerekir. Bunu gerçekleştiren fopen() fonksiyonudur. Bu fonksiyonla bir dosyayı okumak ('r'), yazdırmak ('w') veya ek yapmak ('a') için açabiliriz. Bu fonksiyon dosyanın başarıyla açılması halinde bir tamsayı verecektir.

```
$dosya = fopen( "bir_dosya.txt" , 'r' );
```

Açılan bir dosyanın bütün işlemler bittikten sonra, kapatılması gerekir. Dosya kapatma işlemini fclose() fonksiyonu yapar. Bu fonksiyona parametre olarak dosya adını değil, dosyanın işaretçisi olan değişkenin adını veririz. Örnek:

```
fclose ( $dosya );
```

6.14.8 Dosya Okuma: (fgets(), fread() ve fgetc())

PHP' de bir dosyanın içeriğini satır satır okuma fgets() fonksiyonu ile sağlanır. Bu fonksiyona daha önce açılmış olan dosyanın işaret değişkenin adını ve okunmasını istediğimiz asgari byte ölçüsünü parametre olarak veririz. fgets() fonksiyonu verdiğimiz uzunluk ölçüsüne ulaşmadan önce dosyada bir yeni satır işareti görürse, veya dosyanın sonuna ulaşırsa, okumaya son verir. Bu fonksiyonu çoğu zaman bir döngünün içinde kullanırız. Ancak döngünün hata vermemesi için, PHP' nin dosyanın sonuna ulaştığında döngüyü durdurmamız gerekir. fgets() fonksiyonunun okuyacağı satırı bir fonksiyona değer olarak verebilir ve daha sonra bu değer programlarda kullanabilir. Örnek:

```
<?php
$dosya_dizin = "/inetpub/wwwroot/";
if ($dosya = (fopen ("$dosya_dizin/bir_dosya.txt" , 'r') ) ) {
        print ("Dosya açıldı!<br>");
} else {
        print ("Dosya açılamadı!");
while ( ! feof ($dosya ) ) {
        $satir = fgets ( $dosya, 1024 ) ;
        print ("$satir<br>");
}
fclose ($dosya);
?>
```

Bu işlemin içinde yapıldığı while() döngüsünün devam şartı olarak kullanılan ifade yer alan feof() fonksiyonu bir dosyanın dosya-sonu (end-of-file) durumuna ulaşıp ulaşmadığını sınar. fread() fonksiyonu da daha önce açılmış olan dosyanın işaret değişkenin adını ve okunması istenilen asgari byte ölçüsünü parametre olarak alır.

6.14.9 Dosyaya Yazma ve Ek Yapma (fwrite() ve fputs())

Bir dosyaya yazma veya ek yapma, PHP açısından aynı işlemdir; sadece dosyaların açılışında fark vardır. Bir dosyayı yazmak amacıyla açmak için:

```
$dosya = fopen( "bir_dosya.txt" , 'w' ) or die ("Dosya açılamıyor!") ;
```

ekleme amacıyla açmak için ise

```
$dosya = fopen( "bir_dosya.txt" , 'a' ) or die ("Dosya açılamıyor!") ;
```

kodunu yazmamız gerekir. Daha sonra yapılacak yazma ve ekleme işlemlerinin farkı, 'w' parametresi ile açılan dosyaya yazma işlemi en başından başlar ve devam eder; 'a' parametresi ile açılan dosyaya yazma işlemi ise en sondan başlar ve devam eder. PHP' nin bir dosyaya metin yazdırma fonksiyonları olan fwrite() ve fputs() aynı biçimde yazılır ve aynı işlevi yerine getirirler; aralarında fark yoktur. Örnek:

```
<?php
$dosya_adi = "/inetpub/wwwroot/bir_dosya.txt";
$dosya = fopen ($dosya_adi , 'w') or die ("Dosya açılamadı!");
$metin = "Bu satır dosyaya yazılacak: Merhaba Dünya!\n";
fwrite ( $dosya , $metin ) ;
fputs ( $dosya , "Bu satır ise sonradan eklenecek\n" ) ;
fclose ($dosya);
?>
```

6.15 Temel Alfanümerik Fonksiyonları

PHP' nin metin olarak gördüğü değişkenlere alfanümerik (String) değişkenler denir. PHP' nin alfanümerik fonksiyonları, bu tür değişkenlerin değerlerinin içinden bir bölümü alma, atma veya değiştirme imkânı sağlar.

substr(): Bir alfanümerik değişkenin değerinin veya bir metnin tanımlanan bölümünü verir. İkisi zorunlu, biri seçmeli üç parametre ile kullanılır. Yazılışı:

```
substr($degisken, başla, [boyut] );
```

trim(): Bir alfanümerik değişkenin değerinin baş ve son tarafındaki boşlukları atar. Yazılışı:

trim($degisken)

chr(): Parametre olarak ASCII değerini belirttiğimiz karakteri sağlar. Örnek:

echo (chr(34);

ord(): Parametre olarak yazdığınız karakterin ASCII değerini sağlar. Örnek:

echo (ord("A");

strlen(): Bir alfanümerik değişkenin değerinin kaç karakter içerdiğini bildirir. Yazılışı:

strlen($degisken);

printf()/sprintf(): Bu fonksiyonlar bir değişkeni biçimlendirmekte kullanılır. Birincisinin elde ettiği sonuç ziyaretçinin inceleyici penceresine gönderilir; ikincisinin elde ettiği sonuç ise değer olarak döner. Bu fonksiyonlarla kullanabilecek biçim parametreleri;

%	Yüzde işareti. Yanında biçim parametresi gerekmez.
b	Değişken tamsayı olarak işlem görür ve ikili sayı olarak döner.
c	Değişken tamsayı olarak işlem görür ve ASCII değerinin karşılığı olan karakter olarak döner.
d	Değişken tamsayı olarak işlem görür ve ondalık sayı olarak döner.
f	Değişken kesirli sayı olarak işlem görür ve kesirli sayı olarak döner.
o	Değişken tamsayı olarak işlem görür ve sekiz-tabanlı (octal) sayı olarak döner.
s	Değişken alfanümerik olarak işlem görür ve alfanümerik olarak döner.
x	Değişken tamsayı olarak işlem görür ve 16 tabanlı (hexadecimal) sayı olarak döner. (Harfler, küçük harf olur).
X	Değişken tamsayı olarak işlem görür ve 16 tabanlı (hexadecimal) sayı olarak döner. (Harfler, büyük harf olur).

Her iki fonksiyonun da kullanılış biçimi aynıdır:

printf("biçim" , $degisken1, $degisken2, ... "metin");

number_format(): Türü sayı olan değişken değerlerini bin-basamakları şeklinde biçimlendirmekte kullanılır. Parametre olarak sayı içeren değişkenin adını, ondalık bölümün kaç haneli olacağını, ondalık hanesini ve binler basamaklarını ayırmakta kullanılacak karakteri kabul eder. Örnek:

```
$degisken = 123456.123 ;

echo (number_format($degisken, 4 chr(44) , ".") ); //chr(44)=virgül
```

Bu deyimle 123456.123 şeklindeki değer, browser penceresine "123.456,123" şeklinde yazdırılacaktır.

6.16 PHP-MySQL İlişkisi

PHP' nin yaygınlaşmasında MySQL programının çok büyük etkisi vardır. Windows ve Unix versiyonları ücretsiz olarak dağıtılan MySQL' in PHP ile beraber gelişmesi sayesinde hem PHP hem de MySQL karlı çıktılar. Gelişmiş veritabanı sistemlerinde olan ilişkisel veri tabanı mantığıyla geliştirilen MySQL, çoklu bağlantı desteği ve performansıyla uzunca süreden beri İnternet' te adını duyurmayı başarmıştı. Kısa bir süre önce GPL lisansıyla kaynak kodları açıklanan MySQL' in PHP ile stratejik işbirliği sayesinde PHP 4 ve MySQL birlikte diğer veritabanı ve dil ikililerine karşı konulamaz performans üstünlüğü sağlıyorlar. PHP' nin, MySQL ve Apache ile birlikte derlenerek kullanıldığı sistemlerde veritabanı ve Web sunucu arasında kurulan kalıcı bağlantılar (persistent connection) sayesinde Web sunucu ve veritabanı tek bir yazılımmış gibi çalışabiliyor. Bu sayede çalışan iki ayrı yazılıma göre %400' lere varan performans üstünlüğü de sağlanmış oluyor. Kısa bir süre önce İnternet'te dünyanın en çok kullandığı site olan Yahoo' da bazı bölümlerinde MySQL kullanmaya başlamıştır..

Bir PHP programı yazılarak veritabanındaki kayıtlar okunabilir. PHP programlarının veritabanından yararlanabilmesi için programın önce Web sunucusu aracılığıyla veritabanı dosyası ile bağlantı kurması gerekir. MySQL açısından ise bu bağlantı, veri sunucusunda yeni bir oturum açılması anlamına gelir. İki program arasındaki bu ilişkiyi PHP' nin mysql_connect() fonksiyonu yapar. Bu fonksiyonun alabileceği üç parametre vardır:

```
$veri_yolu = mysql_connect ("localhost" , "root" , "parola" );
```

Burada "localhost" yerine MySQL programının parçası olarak çalıştığı sunucunun adı yazılır. "root", MySQL sunucusunda açılacak oturumun kimin adına açılacağını belirler. "root" kelimesi, sunucunun yönetici oturumu açacağı anlamına gelir: "parola" kelimesinin yerine de

MySQL' i kurarken belirlenen bir kullanıcı parolası varsa, o yazılır. Bu komutta yer alan $veri_yolu değişkeni, açılacak veri yolunun, PHP ile MySQL veritabanı sunucusu arasındaki bağın tanıtıcı işareti olacaktır. Veri sunucusu ile veri yolu bağlantısı kurulursa, bu değişken değer tutar hale gelir; bağlantı kurulamazsa bu değişken boş kalır. mysql_connect() fonksiyonunun başarılı olup olmadığını bu değişkenin durumunu sınayarak anlayabiliriz.

```
$veri_yolu =mysql_connect("ogis", "root");

if ( ! $veri_yolu) die ("MySQL ile veri bağlantısı kurulamıyor!);
```

Burada ikinci satırdaki if deyimi, $veri_yolu değişkeninin değer içerip içermediğine bakıyor ve değişkende bir değer yoksa, bağlantı kurma girişini durdurarak, ziyaretçiye hata mesajı gönderiyor. Bağlantı başarıyla kurulduktan sonra PHP programı, bu yoldan, veritabanı sunucusuna, hangi veritabanı dosyasından yararlanmak istediğini bildirmelidir. Buna veritabanı dosyası seçme işlemi denir ve mysql_select_db() fonksiyonu ile yapılır:

```
mysql_select_db( "veritabanın_adı" , $veri_yolu ) or die ("Veritabanı açılamıyor!". mysql_error() );
```

Bu fonksiyonun başarıyla icra edilip edilmediği fonksiyondan dönen değerin true/doğru veya false/yanlış olmasından anlarız. Bu değer false ise bu deyimin die() bölümü icra edilecek ve inceleyici penceresine veritabanının açılamadığı mesajıyla birlikte MySQL' in hata mesajı da gönderilecektir. PHP' nin MySQL veritabanını seçememesi çoğu zaman kullanıcı yetkilerinin Internet ziyaretçilerini kapsayacak şekilde düzenlenmemiş olmasından kaynaklanır. Bu durum gerçek Web sunucusunda ortaya çıkarsa, Web sunucusu yönetimine başvurmak gerekir.

```
<HTML>

<TITLE>PHP ile Veri Örnegi</TITLE>

<meta http-equiv="content-type" content="text/html; charset=ISO-8859-9">

<meta http-equiv="Content-type" content="text/html; charset=windows-1254">

</HEAD>

<BODY>

<?php

$veri_yolu = mysql_connect("ogis", "root");
```

```
if ( ! $veri_yolu ) die ("MySQL ile veri bağlantısı kurulamıyor!");

mysql_select_db("veri" , $veri_yolu)

or die ("Veritabanına ulaşılamıyor!" . mysql_error() );

$sonuc = mysql_query("SELECT * FROM ogrenciler", $veri_yolu);

printf("Adı: %s<br>\n", mysql_result($sonuc,0,"adi"));

printf("Soyadı: %s<br>\n", mysql_result($sonuc,0,"soyadi"));

printf("Numarası: %s<br>\n", mysql_result($sonuc,0,"no"));

?>

</BODY>

</HTML>
```

Burada, mysql_connect() fonksiyonu ile "ogis" isimli sunucuda root adına MySQL sunucu ile bağ kurduktan sonra mysql_select_db() fonksiyonu ile bu bağı kullanarak veri isimli veritabanından veri çekileceği bildiriliyor. Daha sonra mysql_query() fonksiyonu ile bu veritabanındaki "ogrenciler" isimli tablonun tüm kayıtları seçiliyor ve kayıtlar $sonuc dizi-değişkeninde toplanıyor. $sonuc değişkenin değerlerini görüntülemek için PHP' nin özel bir fonksiyonu olan mysql_result() fonksiyonu kullanılıyor. mysql_query() fonksiyonu, PHP' nin SQL dilini kullanarak veritabanı işlemleri yapmasını sağlayan başlıca aracıdır. SQL komutlarıyla yazılacak bütün "query" deyimleri bu fonksiyon ile icra ettirilir. mysql_result() ise SQL değil, Data Manipulation Language (DML) denen başka bir veri-biçimlendirme dilinin inceliklerinden yararlanılmasını sağlar. Burada $sonuç değişkeninde veritabanı kayıt biçiminde tutulan verileri PHP' nin ve dolayısıyla HTML' in anlayacağı biçime çeviren bu fonksiyondur.

```
<?php

// Form doldurulduktan sonra program buradan başlıyor

if ( isset ( $HTTP_POST_VARS )) {

$veri_yolu = mysql_connect("server", "root");

if ( ! $veri_yolu ) die ("MySQL ile veri bağlantısı kurulamıyor!");
```

```
mysql_select_db("veri" , $veri_yolu) or die ("Veritabanına ulaşılamıyor!" . mysql_error() );

$ekle = mysql_query("INSERT INTO calisanlar ( adi , soyadi , adres , pozisyon ) VALUES
('$adi', '$soyadi', '$adres', '$pozisyon' )", $veri_yolu );

echo ("<HTML><HEAD>

<TITLE>PHP'de Veritabanı</TITLE>

<meta http-equiv='content-type' content='text/html; charset=ISO-8859-9'>

<meta http-equiv='Content-Type' content='text/html; charset=windows-1254'>");

$sonuc = mysql_query("SELECT * FROM calisanlar", $veri_yolu);

echo ("<TABLE><TR><TD><B>Uzmanın Adı</B></TD>

<TD><B>Çalıştığı Yer</B></TD><TD><B>Görevi</B></TD></TR>\n");

while ($satir = mysql_fetch_row($sonuc)) {

        printf("<TR><TD>%s %s</TD><TD>%s</TD></TD><TD>%s</TD></TR>\n",

$satir[1], $satir[2], $satir[3], $satir[4]);

}

echo ("</TABLE>\n<p><B>Teşekkür ederiz.</B></P>

<A HREF='index.php'>Ana sayfaya dönmek için tıklayınız</A>

");

}

// program ilk kez açılıyorsa buradan başlayacak

else {echo ("<HTML><HEAD>

<TITLE>PHP'de Veritabanı</TITLE>

<meta http-equiv='content-type' content='text/html; charset=ISO-8859-9'>

<meta http-equiv='Content-Type' content='text/html; charset=windows-1254'>

</HEAD>
```

```
<BODY>
<p><B>Mevcut Üyelerimiz</B></P>");
$veri_yolu = mysql_connect("server", "root");
mysql_select_db("veri", $veri_yolu);
$sonuc = mysql_query("SELECT * FROM calisanlar", $veri_yolu);
echo ("<TABLE><TR><TD><B>Uzmanın Adı</B></TD>
<TD><B>Çalıştığı Yer</B></TD><TD><B>Görevi</B></TD></TR>\n");
while ($satir = mysql_fetch_row($sonuc)) {
printf("<TR><TD>%s %s</TD><TD>%s</TD></TD><TD>%s</TD></TR>\n",
$satir[1], $satir[2], $satir[3], $satir[4]);
}
echo ("</TABLE>\n<p></p><p><B>Siz de aramıza katılmak ister misiniz?</B></P>
<FORM ACTION='$PHP_SELF' METHOD='POST'><TABLE>
<TR><TD>Adınız: </TD><TD><INPUT TYPE='TEXT' NAME='adi'></TD></TR>
<TR><TD>Soyadınız: </TD><TD><INPUT TYPE='TEXT' NAME='soyadi'></TD></TR>
<TR><TD>İş Yeriniz: </TD><TD><INPUT TYPE='TEXT' NAME='adres'></TD></TR>
<TR><TDALIGN='left'>Göreviniz: </TD>
<TD><INPUT TYPE='TEXT' NAME='pozisyon'></TD></TR>
<TR><TD ALIGN='center'><INPUT TYPE='SUBMIT' VALUE='Defteri imzala!'></TD>
<TD ALIGN='center'><INPUT TYPE='RESET' VALUE='Tümünü sil!'></TD></TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

");

}

?>

Program, ilk kez çalıştığında, çalışmaya ikinci yarısındaki else() deyiminden başlayarak itibaren icra ediliyor; ziyaretçilere mevcut üyelerin listesini veriyor ve üye olmak isteyip istemediğini soruyor. Arzu edenin üye olabilmesi için gerekli Form' u da sunuluyor. Programın her iki bölümünde de veri okuyan ve bunu görüntüleyen, mysql_fetch_row() fonksiyonudur. PHP' nin DML araçlarından olan bu fonksiyonun marifeti, bir veritabanından elde edilen sonucu satır-satır okumasıdır. Nitekim, burada bu fonksiyondan dönen değeri $satir adını verdiğimiz dizi-değişkene yazıyoruz ve sonra printf() bu dizinin elemanlarını sırayla browser penceresine gönderiyor. Programın iki bölümü arasındaki tek fark, $HTTP_POST_VARS dizi-değişkeninin bir değer tutması halinde (yani ziyaretçi sayfayı açtığında karşısına çıkan Form' u doldurduğu ve gönderdiği zaman) çalışan birinci bölümünde, mysql_query() fonksiyonunun bu kez veritabanı dosyasına ziyaretçinin verdiği bilgileri işlemek üzere farklı bir SQL deyimi içermesidir. Bu fonksiyon "ogrenciler" tablosundaki dört alana, dört değişkenin değerlerini SQL' in INSERT komutuyla ekliyor.

PHP' nin MySQL ile yapabileceğimiz veritabanı yönetimi için 20' ye yakın fonksiyonu vardır;

6.16.1 PHP' de Kullanılabilecek MySQL Komutları :

mysql_affected_rows : Bir önceki işlemde etkilenen satır sayısı

mysql_close : Belirtilen MySQL bağlantısını kapatır

mysql_connect : Sunucuya veritabanı bağlantısı açar

mysql_create_db : MySQL' de veritabanı açar

mysql_data_seek : Sonuç satırında belirtilen sıraya geçer

mysql_db_query : MySQL' e sorgu gönderir

mysql_drop_db : Sunucudan veritabanı siler

mysql_errno : Bir önceki işlemdeki MySQL hata numarasını verir

mysql_error : Bir önceki işlemdeki MySQL hata mesajını verir

mysql_fetch_array : Sonuçları dizi değişkeni olarak alır

mysql_fetch_field : Sonuç tablosundaki alan adını obje olarak alır

mysql_fetch_lengths : Sonuç tablosundaki dizi değişkenin uzunluğunu alır

mysql_fetch_object : Sonuç satırını obje olarak alır

mysql_fetch_row : Sonuç tablosundan dizi değişkeni olur

mysql_field_name : Sonuç tablsundaki sonucun tablodaki alan adını verir

mysql_field_seek : Sonuç tablosunda sıra indisini belirtilen yere götürür

mysql_field_table : Alan adı verilen sonucun tablo adını verir

mysql_field_type : Sonuçtaki alanın hangi tip olduğunu belirtir

mysql_field_flags : Sonuçtaki alanın hangi tür ekstra parametrelerle tanımlandığını belirtir

mysql_field_len : Sonuçtaki alanın veritabanındaki uzunluğunu verir

mysql_free_result : Sonuçlar için atanan hafızayı boşaltır

mysql_insert_id : Bir önceki veri yerleştirmede oluşan otomatik veri değerini verir

mysql_list_fields : Sonuçtaki tüm tablo alanlarını listeler

mysql_list_dbs : Sunucudaki tüm veritabanlarını listeler

mysql_list_tables : Veritabanındaki tüm tabloları listeler

mysql_num_fields : Sonuçtaki alan sayısını verir

mysql_num_rows : Sonuçtaki satır sayısını verir

mysql_pconnect : Sunucuya kalıcı bir bağlantı tanımlar

mysql_query : Veritabanına sorgu gönderir

mysql_result : Sorgudan dönen sonuçları alır

mysql_select_db : Sunucudan veritabanı seçer

mysql_tablename : Verilen alanın ait olduğu tablo adını verir

6.17 PHP' de Oturum (Session)

PHP4 ile gelen en önemli değişiklik şüphesiz oturum yönetimi özelliğidir. Oturum, kullanıcının siteye girdiği andan tamamen çıktığı ana kadar geçen süreyi tanımlamak için kullanılır. Bu süre içinde kullanıcı pek çok sayfa çağırabilir. Temel olarak kullanıcı siteye ilk gelişinde kendisine bir numara (session_id) verilir ve site gezisi sırasında bu numara ile takip edilir. Kullanıcıya atanan oturum numarasının saklanmasında iki değişik yol kullanılabilir.

Çerezler: Kullanımı daha kolaydır. Ancak kullanıcının tarayıcısının çerez özelliğini kapatmış olması durumunda başarısız olur.

Adrese ekleme: Bu numara sayfalardaki bütün göreceli dizinli bağlantılara otomatik olarak eklenir. Ancak, kullanıcı bildiği bir sayfaya, bağlantıyı tıklamak yerine adres çubuğuna yazarak geçerse, kullanıcının izi kaybedilir.

PHP yukarıdaki her iki seçeneğe de destek vermektedir.

7. SONUÇLAR ve ÖNERİLER

Uzaktan eğitimde kullanılan çeşitli yöntemler bulunmaktadır. Bu yöntemlerin herbirinin eğitmen, öğrenci ve kurum açısından avantaj ve dezavantajları bulunmaktadır. Ortaya çıkan gereksinimler doğrultusunda oluşturulacak yapılarda gözönünde bulundurulması gereken insan, organizasyon türü ve maliyet etkenleridir. Böyle bir sistemde eğitmen ile öğrenciler arasında birebir etkileşim özelliğinin tamamen ortadan kalkması engellenmelidir.Internet tabanlı sistemlerde ise elektronik posta (e-posta) yolu ile soru sorma veya sıkça sorulan sorular bölümü ile bu etkileşimi sağlamak gibi çözümler kullanılmalıdır.

Internet olanakları kullanılarak oluşturulacak böyle proje için Linux, PHP ve MySQL programları çok güçlü bir altyapı sağlar. Bu programlar ücretsiz olarak temin edilebilmesi, kararlı ve güvenli bir sistem kurulalabilmesi yönüyle rakiplerinden avantajlıdır. Unix tabanlı bir sistem olduğu için elbette Windows ortamında kullanılan programlar kadar kullanışlı ve kolay değildir. Ama elde edilebilecek sonuclar ve performans, harcanan daha fazla çabaya değecektir.

KAYNAKLAR

Akerdem,S. ve Usta,S. (2000),Unix,Alfa Yayıncılık,İstanbul.

Bahar,N. ve Türkmen,D.(1998),Internet Unleashed,Sistem Yayıncılık,İstanbul.

Çetin,G. (2000),Linux İşletim Sistemi Red Hat 6.2,Seçkin Yayıncılık,Ankara.

Çetin,G. Çelik,K. (2000),Linux Ağ Yönetimi,Seçkin Yayıncılık,Ankara.

Daft,R.L. ve Lengel R.H.(1986), A Proposed Integration Among Organizational Requirements, Media Richness and Structural Design. Management Science,32,554-571.

Mc Cormack,C. ve Jones,D.(1997), Building A Web-Based Education System. New York: john Wiley & Sons.

Öcal,H. (2000),PHP ,İhlas Yayıncılık,İstanbul

Özkan,Y. (1996),Unix İşletim Sistemi,Alfa Yayıncılık,İstanbul.

Petersen, R. (2001),The Complete Reference Linux, Alfa Yayıncılık,İstanbul

Şamlı,M. (2002),PHP ile Web Programcılığı,Pusula Yayıncılık,İstanbul

Uysal,M. (1994),SQL Veri Tabanı Sorgulama Dili,Beta Yayıncılık,İstanbul

Yurt,B.,Ötkünç,B.ve Karaçayır,A. (2000),Html-Java-Cgi-Vrml-Sgml Unleashed,Sistem Yayıncılık,İstanbul.

INTERNET KAYNAKLARI

[1]www.linux.org.tr

[2]www.php.org.tr

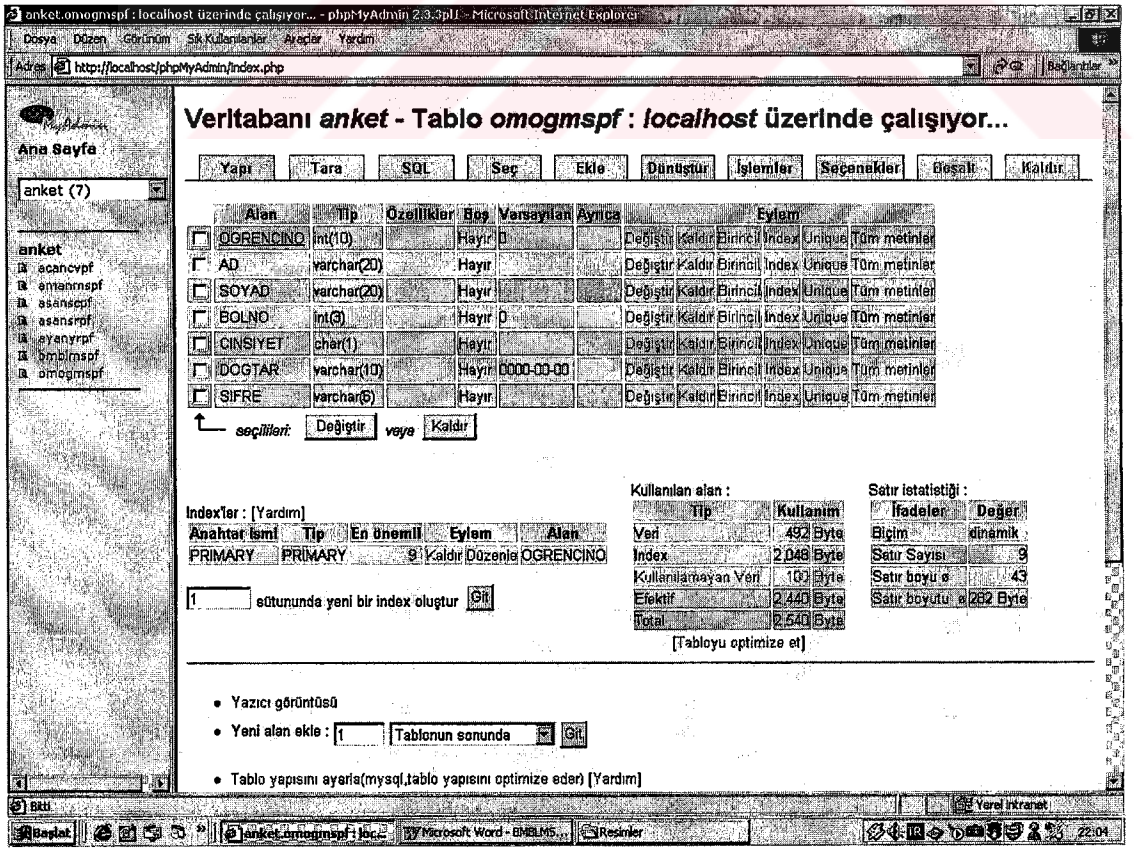
EKLER

Ek 1 Anket Uygulaması

Bu uygulama; kullanıcıya web üzerinden hazırlayıp, yayınlayabileceği ve sonuçlarını yine bu ortam üzerinden değerlendirebileceği bir anket ortamı sunmaktadır. Değişik konularda, soruları ve cevap şıkları tanımlanabilen anket formları hazırlanabilir. Ayrıca anket sorularının dışında anket formunu dolduracak kişilerin konu hakkındaki ek fikirlerini yazabilecekleri alanlar tanıtabilir. Anket uygulaması Apache, PHP ,MySQL yazılmları kullanılarak geliştirilmiştir. Veritabanı yönetiminde yine bir PHP uygulaması olan PHPMyAdmin'den yararlanılmıştır.

Tablolar

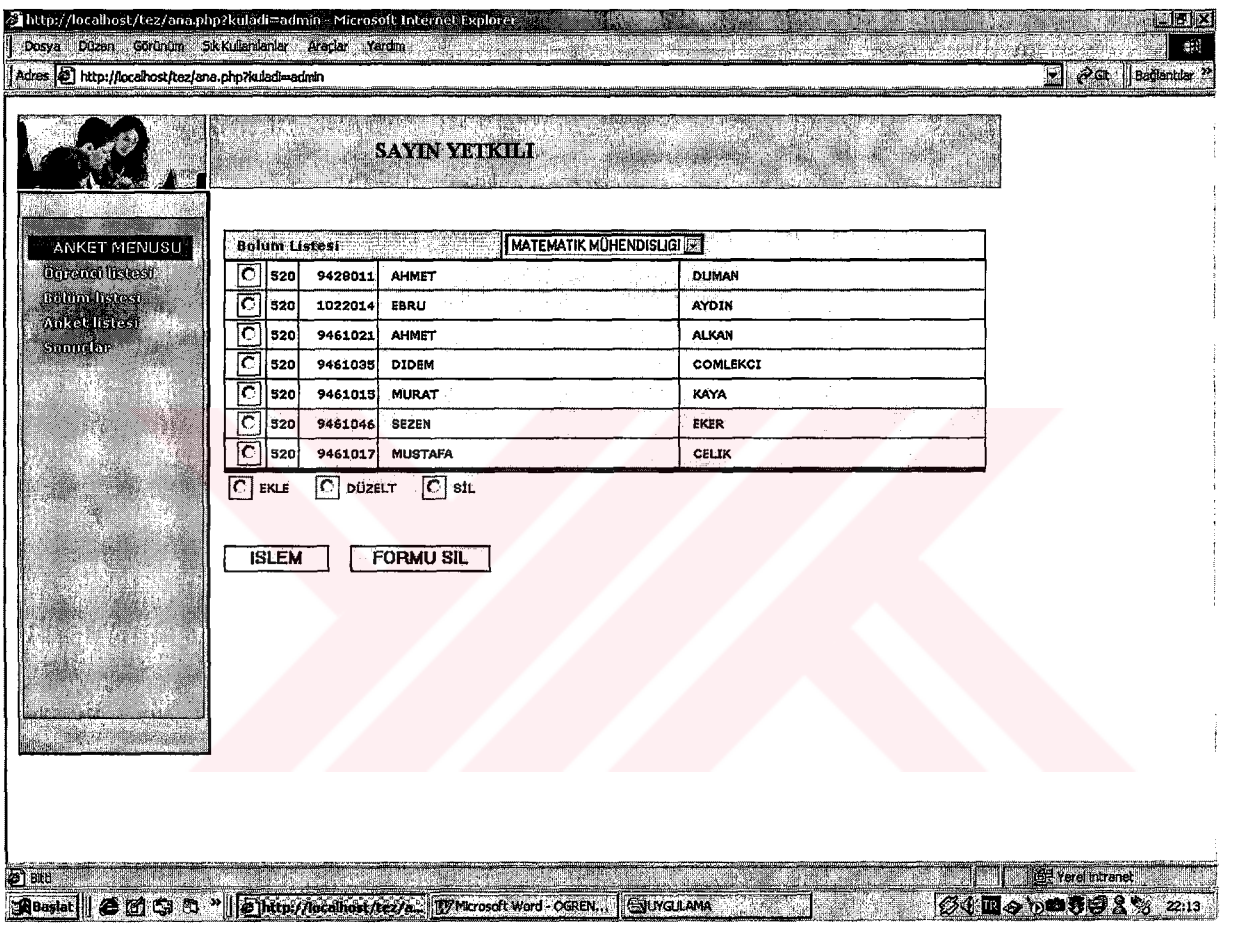
Uygulamada yedi adet tablo yer almaktadır. Bu tablolarda öğrenci bilgileri, bolüm bilgileri, anket tanımları, anket seçenekleri, anket soruları, kullanıcıların verdiği cevaplar, konu hakkındaki yorumları depolanmaktadır. Aşağıda örnek bir tablonun PHPMyAdmin programındaki görüntüsü yer almaktadır. Bu tabloda hazırladığımız anket formlarını dolduracak öğrencilerin bilgileri yer saklanmaktadır. Bu sayfadan tablonun yapısı değiştirilebilir ve kayıt giriş işlemleri yapılabilir.



Şekil Ek 1.1 Örnek veritabanı tablosu (PHMyAdmin)

Ögrenci Listesi

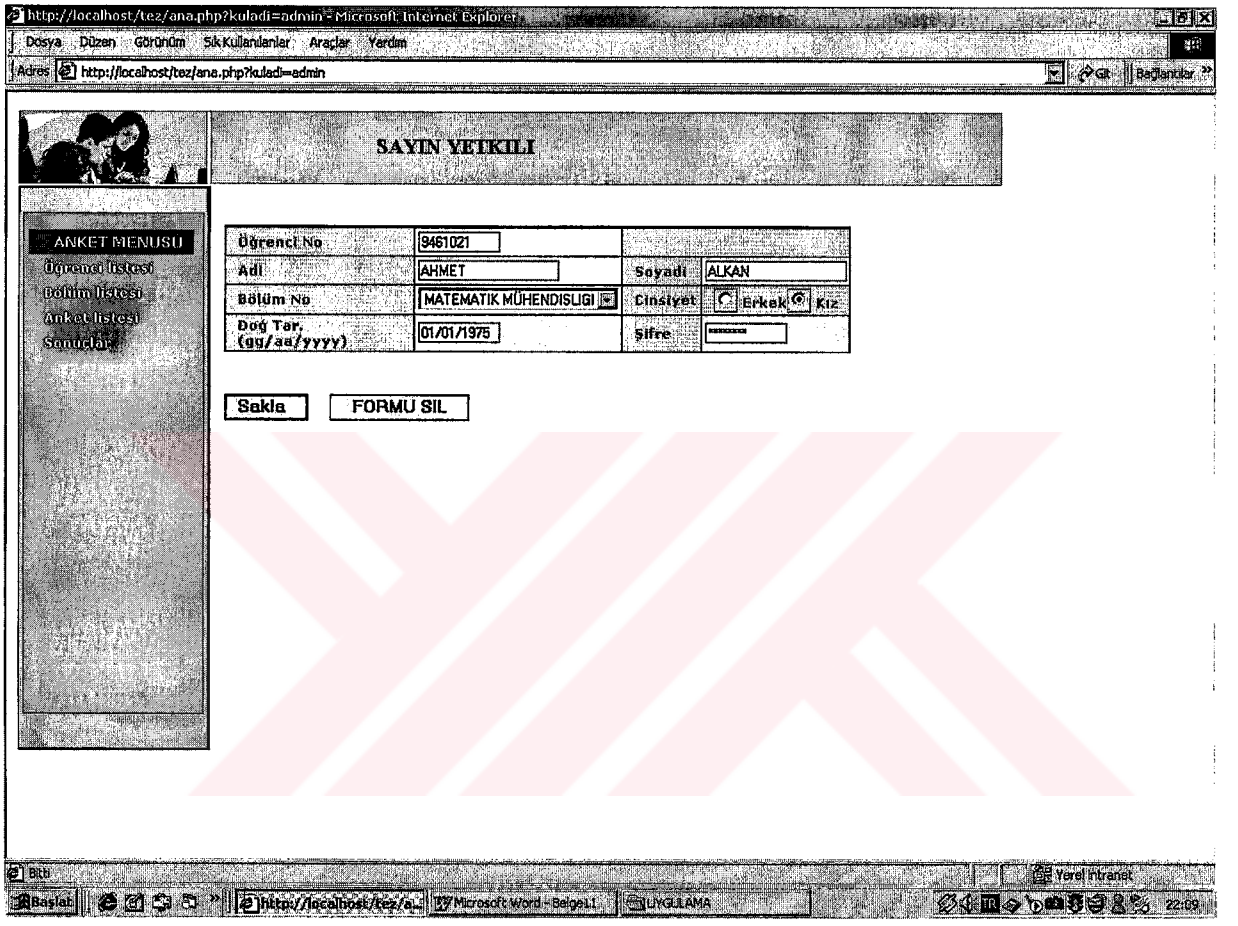
Bu ekrandan farklı bolum öğrencilerinin listesi alınabilir. Ayrıca sayfanın alt tarafında bulunan secenekler ile de listeye yeni bir öğrenci eklenebilir, bir öğrencinin bilgisi değiştirilebilir veya öğrencinin kayıdı silinebilir.



Şekil Ek 1.2 Öğrenci Listesi

Öğrenci Bilgileri Girişi

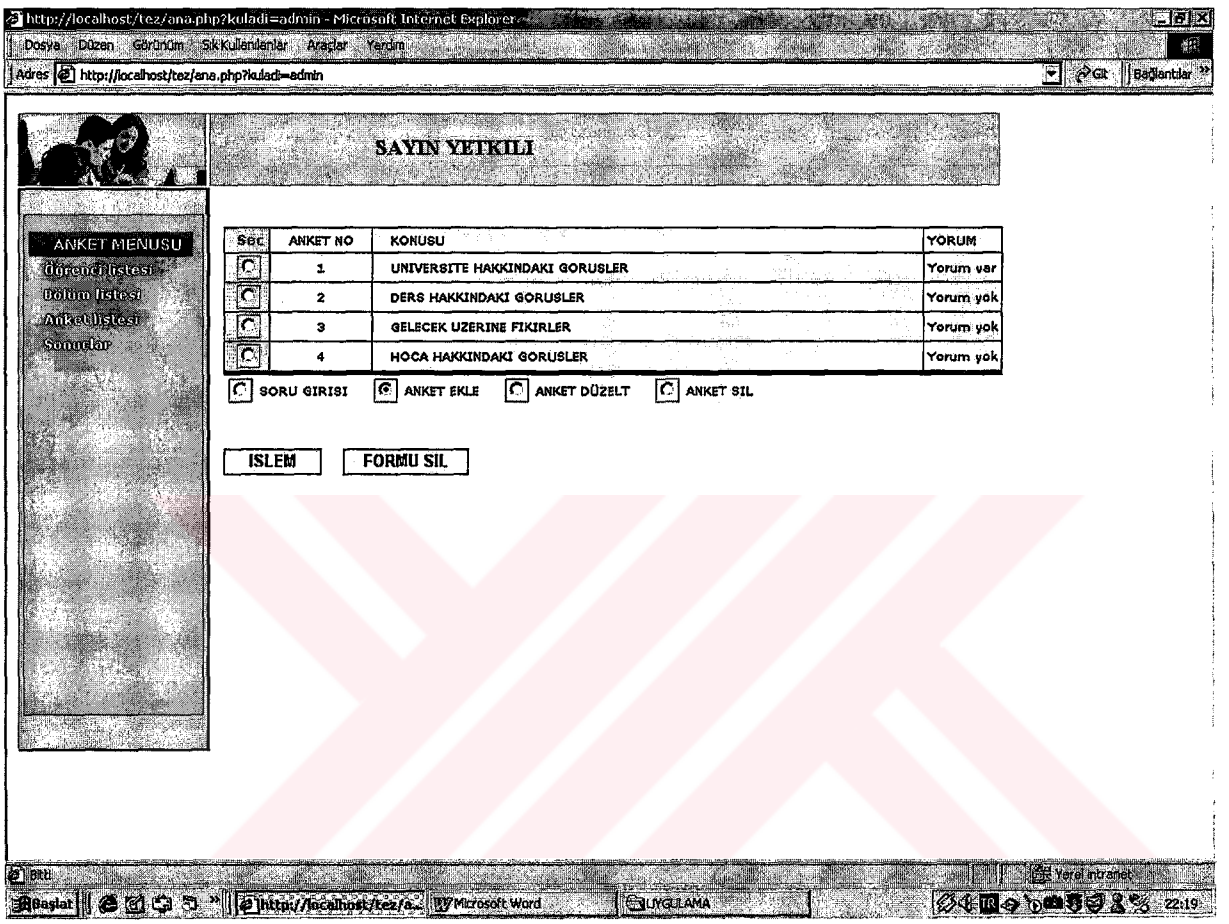
Bu ekrandan öğrenci bilgileri girişi ve düzeltmesi yapılır. Sakla butonuna basıldığında bilgiler anket veritabanında yer alan omogmspf adlı tabloya sql'in insert into veya update komutu ile işlenir.



Şekil Ek 1.3 Öğrenci Tanımlama

Anket Listesi

Bu ekranda sisteme daha önce tanıtılmış anket tanımları konularıyla birlikte listelenir. Anketlerin değiştirilmesi ve düzenlenmesi burada yapılacak seçimlere göre olur.

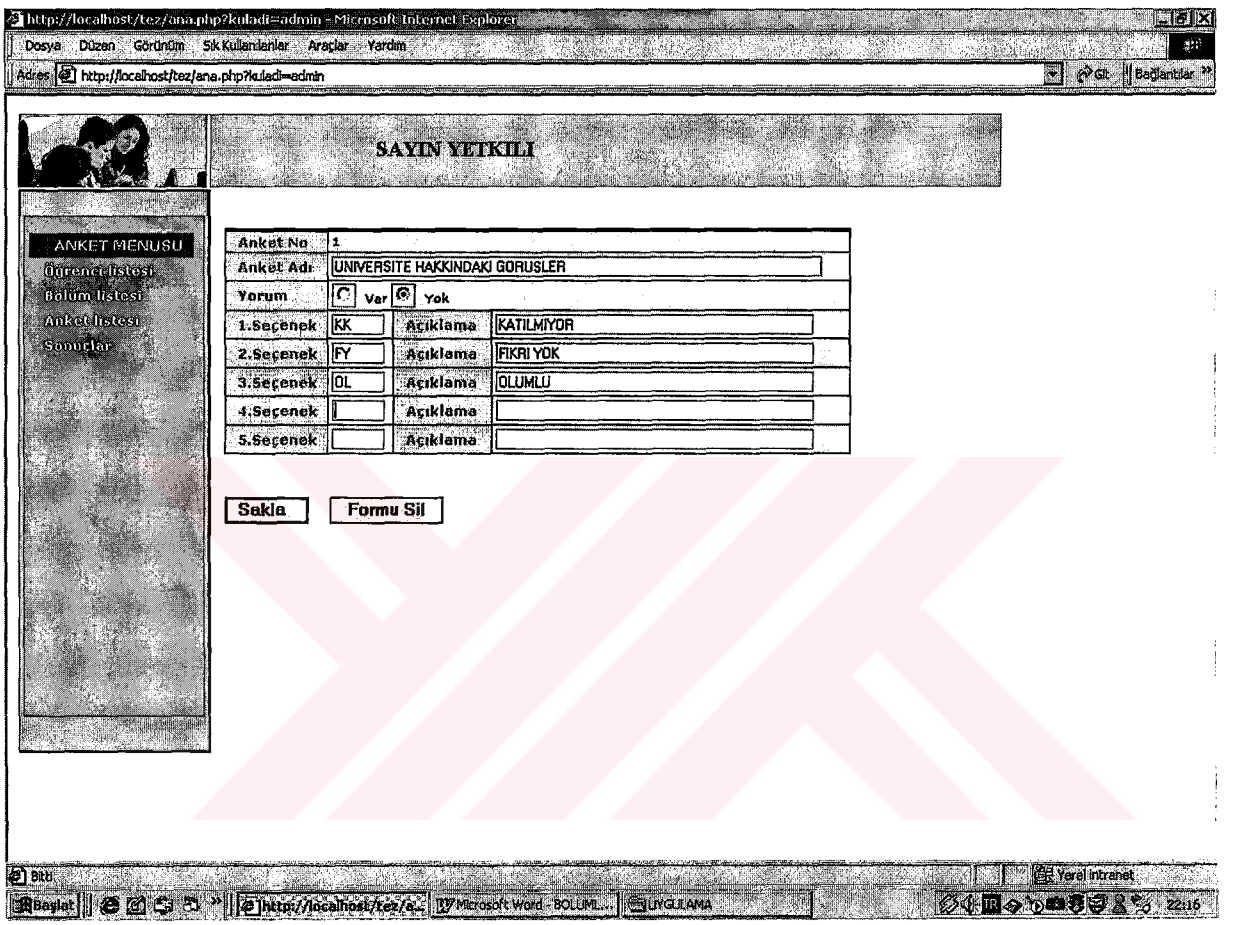


Şekil Ek 1.4 Anket Listesi

Anket Tanımlama

Anket konusu ve ankete yer alacak seçeneklerin tanımlanması bu ekran vasıtası ile yapılır.

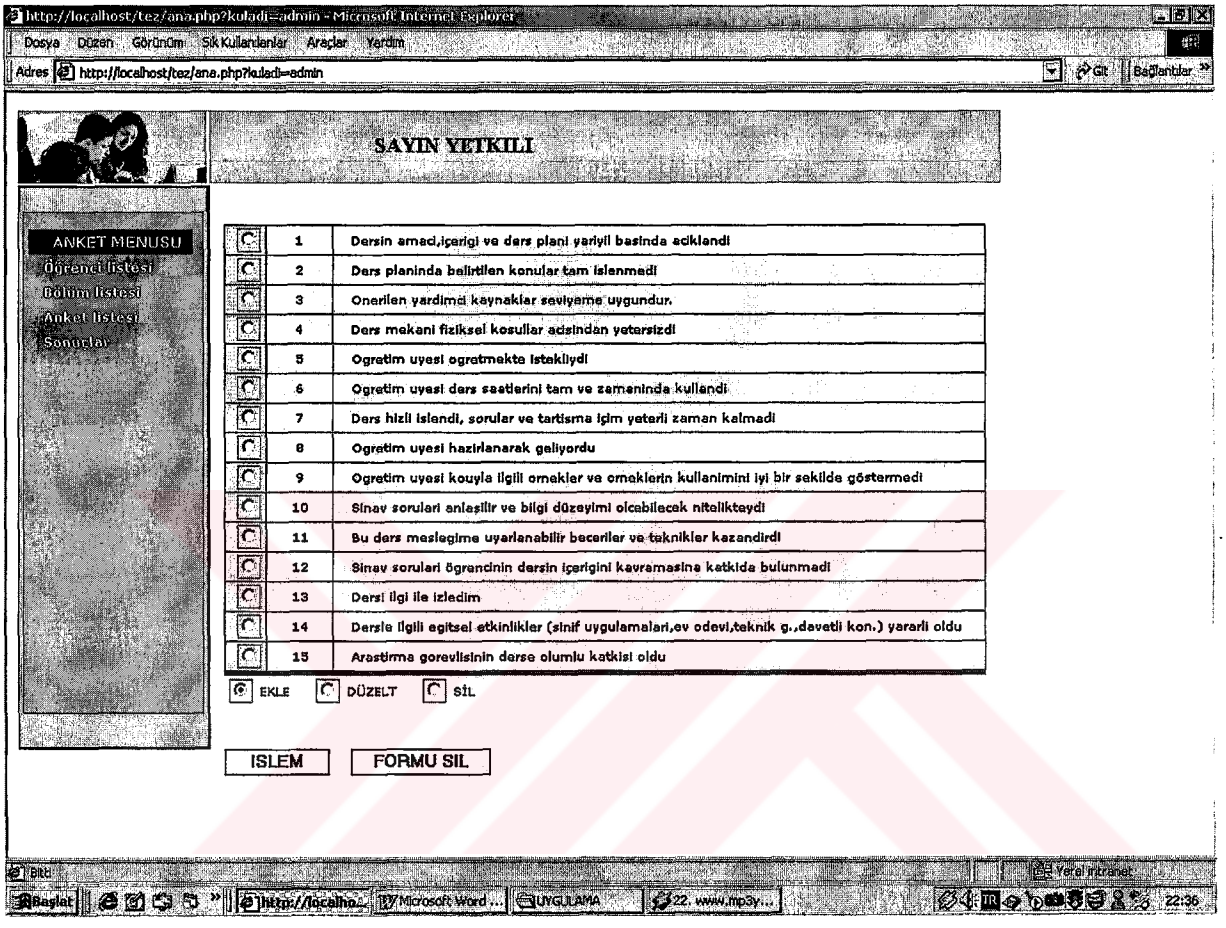
Anket konusu hakkında anketi doduranların yorum yapıp yapamayacağıda burada belirtilir.



Şekil Ek 1.5 Anket ve Seçenek Tanımlama

Anket Soruları Listesi

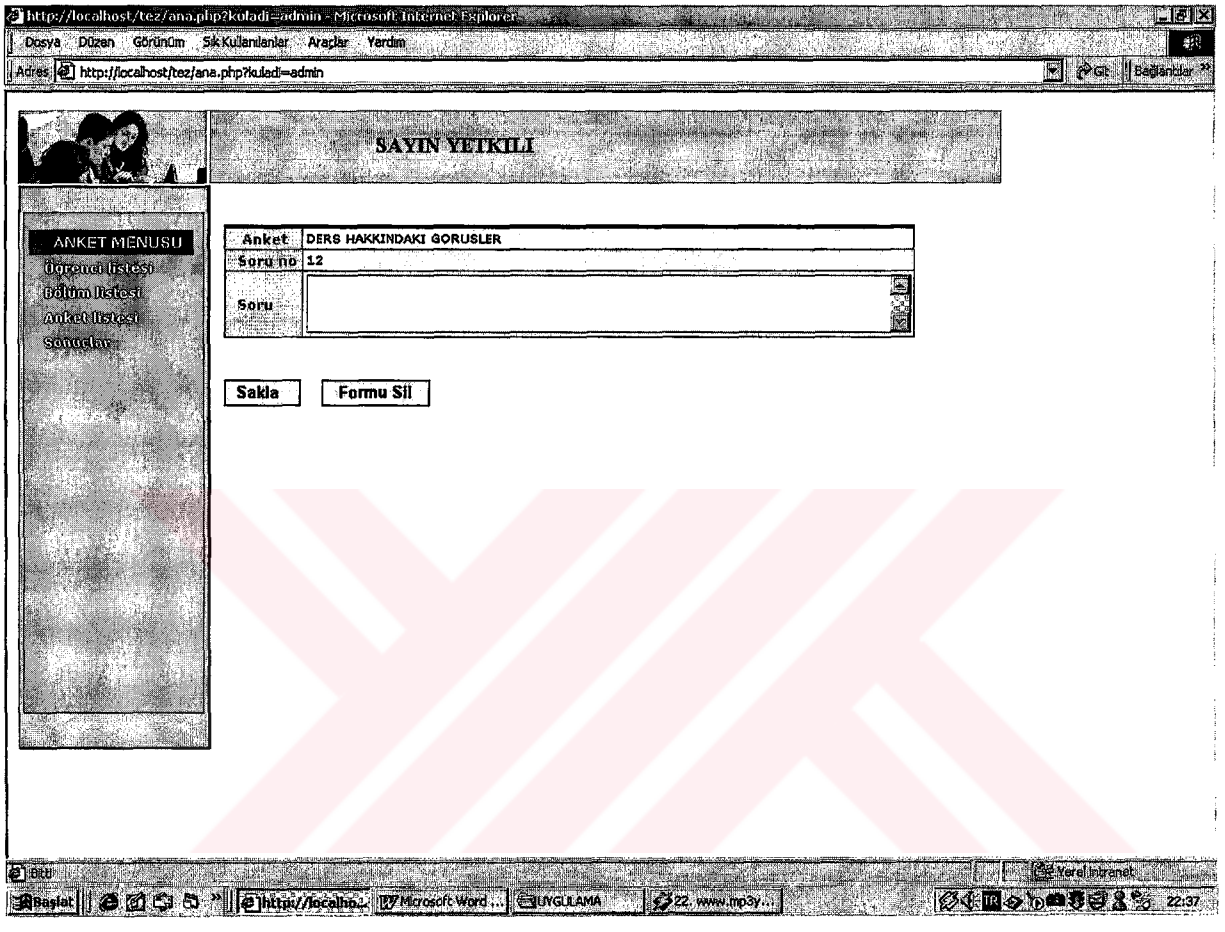
Ankette yer alacak soruların listelendiği ekrandır. Buradan yapılacak seçimlerle yeni soru girilebilir, varolan bir soru değiştirilebilir veya silinebilir.



Şekil Ek 1.6 Soru Giriş/Düzeltme/Silme

Ankete Soru Ekleme

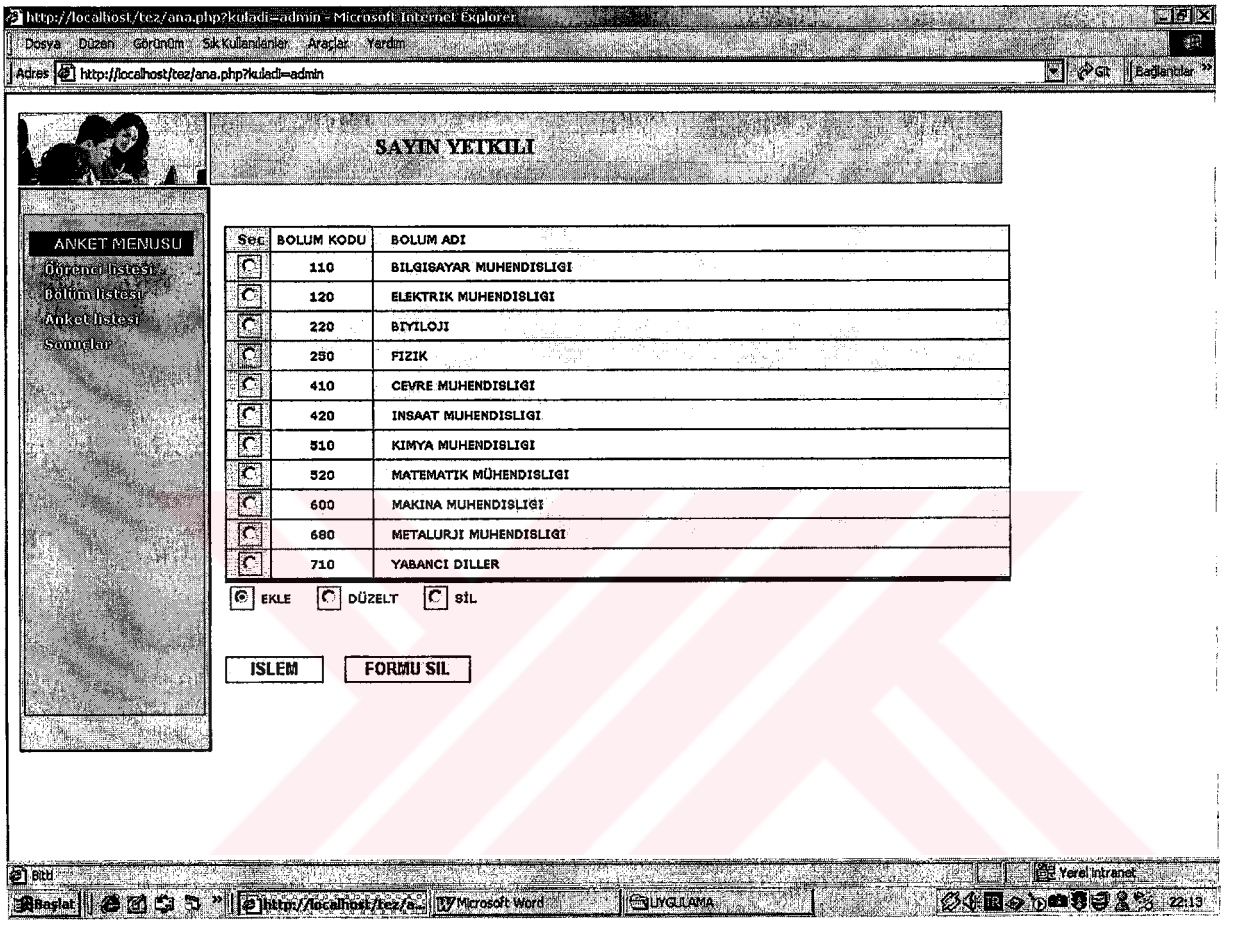
Yeni soru bu ekran vasıtası ile ankete eklenir. Soru numaraları otomatik olarak program tarafından atanır. Varolan soruları değiştirmek içinde bu ekran kullanılır.



Şekil Ek 1.7 Soru Giriş/Düzeltme

Bölüm Kodları Listesi

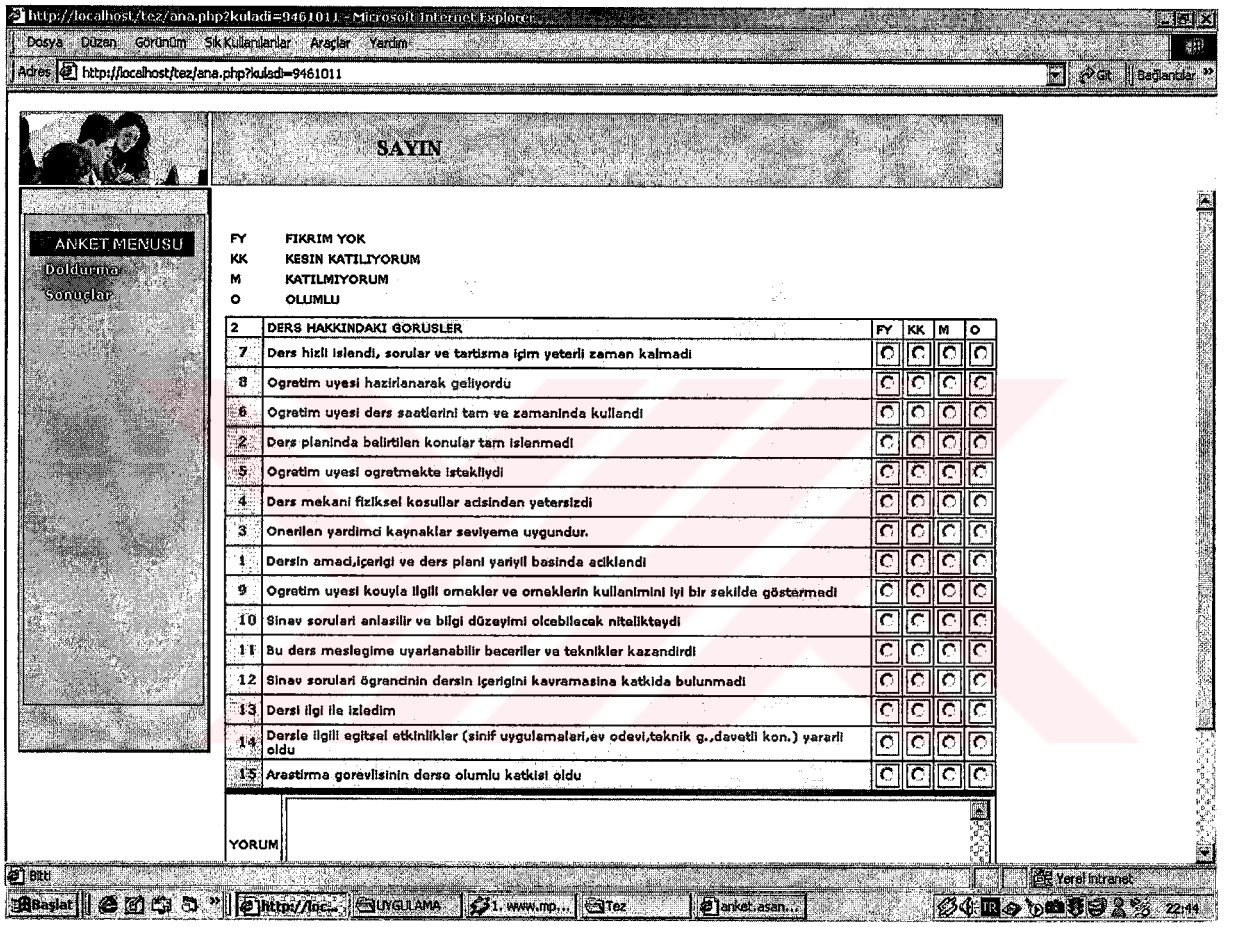
Bu ekrandan bölüm bilgilerinin yönetimi sağlanır. Yeni bölüm tanımlama veya varolan bölümlerin bilgilerinin değiştirilmesi bu ekrandaki seçenekler yardımı ile yapılır.



Şekil Ek 1.8 Bölüm Listesi

Anket Doldurma Formu

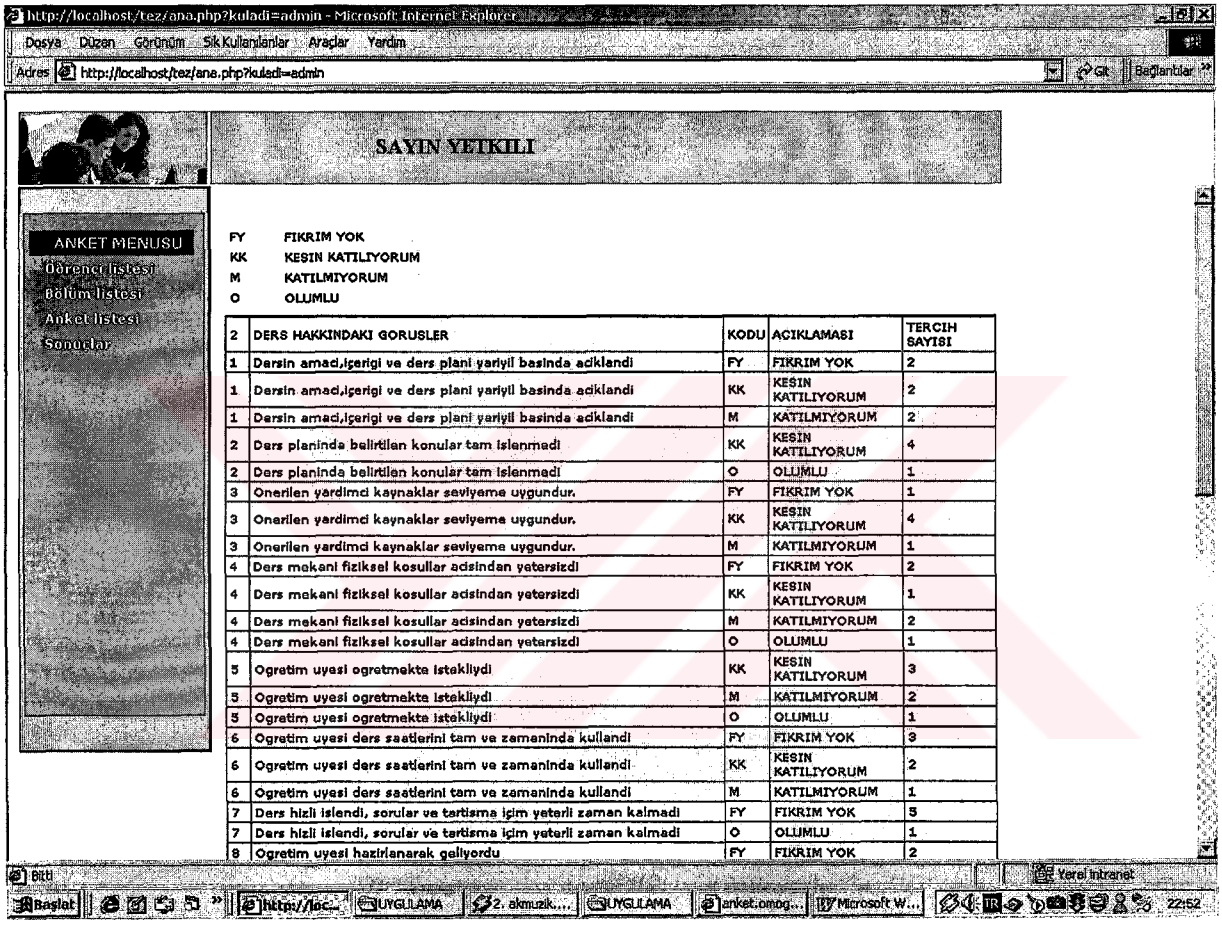
Hazırlanan anketlerin kullanıcılar tarafından doldurulduğu form budur. Kullanıcı yandaki seçenekler vasıtası ile tercihlerini belirtir. Ayrıca konu hakkındaki ek fikirlerinide sayfanın altında yer alan yorum alanına yazabilir. Yorum penceresinin gelip gelmeyeceği anketi düzenleyen kişinin tercihine bağlıdır.



Şekil Ek 1.9 Anket Doldurma Formu

Anket Sonuçları

Bu ekranda; anket formları doldurulduktan sonra dolduran kişilerin yaptıkları tercihler istatistiksel olarak yer alır. Bir soruya kaç kişinin ne şıkkı işaretlediği bu ekrandan görülebilir. Ayrıca anket tanımlanırken kullanıcılara yorum yapma hakkıda verilmişse ekranın en altında yer alan yorumlar butonuyla konu hakkında yazılan yorumların yer aldığı sayfaya geçilebilir.

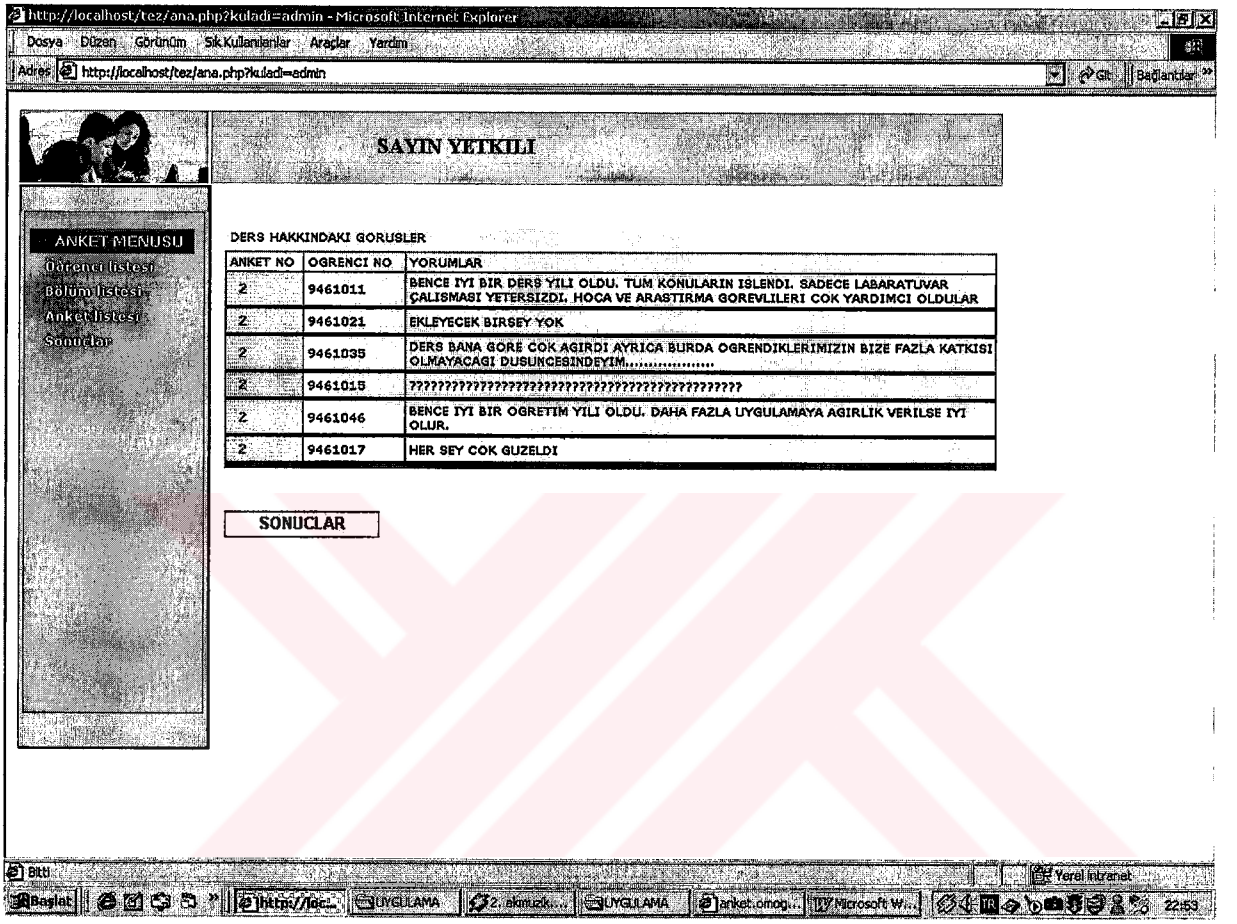


2	DERS HAKKINDAKI GORUSLER	KODU	ACIKLAMASI	TERCIH SAYISI
1	Dersin amacı,içerigi ve ders plani yariyil basinda aciklandi	FY	FIKRIM YOK	2
1	Dersin amacı,içerigi ve ders plani yariyil basinda aciklandi	KK	KESIN KATILIYORUM	2
1	Dersin amacı,içerigi ve ders plani yariyil basinda aciklandi	M	KATILMIYORUM	2
2	Ders planinda belirtilen konular tam islenmedi	KK	KESIN KATILIYORUM	4
2	Ders planinda belirtilen konular tam islenmedi	O	OLUMLU	1
3	Onerilen yardimci kaynaklar seviyeme uygundur.	FY	FIKRIM YOK	1
3	Onerilen yardimci kaynaklar seviyeme uygundur.	KK	KESIN KATILIYORUM	4
3	Onerilen yardimci kaynaklar seviyeme uygundur.	M	KATILMIYORUM	1
4	Ders mekani fiziksel kosullar acisindan yetersizdi	FY	FIKRIM YOK	2
4	Ders mekani fiziksel kosullar acisindan yetersizdi	KK	KESIN KATILIYORUM	1
4	Ders mekani fiziksel kosullar acisindan yetersizdi	M	KATILMIYORUM	2
4	Ders mekani fiziksel kosullar acisindan yetersizdi	O	OLUMLU	1
5	Ogretim uyesi ogretmekte istekliydi	KK	KESIN KATILIYORUM	3
5	Ogretim uyesi ogretmekte istekliydi	M	KATILMIYORUM	2
5	Ogretim uyesi ogretmekte istekliydi	O	OLUMLU	1
6	Ogretim uyesi ders saatlerini tam ve zamaninda kullandi	FY	FIKRIM YOK	3
6	Ogretim uyesi ders saatlerini tam ve zamaninda kullandi	KK	KESIN KATILIYORUM	2
6	Ogretim uyesi ders saatlerini tam ve zamaninda kullandi	M	KATILMIYORUM	1
7	Ders hizli islendi, sorular ve tartisma icim yeterli zaman kalmadi	FY	FIKRIM YOK	5
7	Ders hizli islendi, sorular ve tartisma icim yeterli zaman kalmadi	O	OLUMLU	1
8	Ogretim uyesi hazirlanarak geliyordu	FY	FIKRIM YOK	2

Şekil Ek1.10 Anket Sonuçları

Konu İle İlgili Yorumlar

Anket tanımında yorum yapılmasına izin verilmişse buradan kullanıcıların yazdığı yorumlar okunabilir.



ANKET NO	OGRENCI NO	YORUMLAR
2	9461011	BENCE IYI BIR DERS YILI OLDU. TUM KONULARIN ISLENDI. SADECE LABARATUVAR ÇALISMASI YETERSIZDI. HOCA VE ARASTIRMA GOREVLILERI COK YARDIMCI OLDULAR
2	9461021	EKLEYECEK BIRSEY YOK
2	9461035	DERS BANA GORE COK AGIRDI AYRICA BURDA OGRENDIKLERIMIZIN BIZE FAZLA KATKISI OLMAYACAGI DUSUNCESINDEYIM.....
2	9461015	??
2	9461046	BENCE IYI BIR OGRETIM YILI OLDU. DAHA FAZLA UYGULAMAYA AGIRLIK VERILSE IYI OLUR.
2	9461017	HER SEY COK GUZELDI

Şekil Ek 1.11 Yapılan Yorumlar

Giris.php

```
<head>
<script language='JavaScript'>
 function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
}
</script>
<link href='formstyle.css' rel='stylesheet' type='text/css'>
<meta http-equiv='content-type' content='text/html; charset=iso-8859-9'>
<title>Anket &#304;&#351;lemleri</title>
</head>
<body leftmargin=0 topmargin=0 marginwidth=0 marginheight=0>
<table border="0" cellpadding="0" cellspacing="0" width="100%" bgcolor="#E33635">
<tr><td></td></tr></table><br><br>
<TD><CENTER><h1><FONT COLOR="BLUE">Sisteme Giri&#351;</h1><TD>
<br><form action='kontrol.php' method='post'>
<center><HR><table border=0 cellpadding=0 cellspacing=0 bgcolor='#444444'><tr><td>
<table border=0 class='tablecol col1' cellpadding=0 cellspacing=1>
<tr><th>Kullan&#305;c&#305; Ad&#305;</th><td><input type='text' name='kuladi'
size=24></td></tr>
<tr><th>&#350;ifre</th><td><input type='password' name='sifre' size=24></td></tr>
</table></td></tr></table><HR>
<H1><FONT COLOR="RED"><CENTER>
<?  if ($hata == 1 ) {
   print("Kullan&#305;c&#305; kodu/&#351;ifre bo&#351; geçilemez!");
   }
?>
</H1>  <br><br><input type="submit" value="Tamam" name="B1">
<br><br></center></form>
```

Kontrol.php

```
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
    if ( $kuladi == "" or $sifre =="") {
      header("Location:giris.php?hata=1");
    }
    if ($kuladi == "admin") {
      if ($sifre == "qaz") {
         header("Location:ana.php?kuladi=$kuladi"); exit;
      } else {
         header("location:giris.php?hata=1");  exit;
      }
                          }
      $table="SELECT * FROM omogmspf where ogrencino=$kuladi and sifre='$sifre'";
```

```
        $sorgu=mysql_query($table);
        if (mysql_num_rows($sorgu) >0) {
           header("location:ana.php?kuladi=$kuladi"); exit;
        } else {
           header("location:giris.php?hata=1"); exit;
        }
?>
```

Ana.php

```
<head>
 <input type="hidden" name="kuladi" value="$kuladi">
 <FRAMESET ROWS="12%,88%">
 <FRAME  NAME="frmtop"  SCROLLING="no"
SRC="top.php?kuladi=<?echo($kuladi)?>" frameborder ="0" noresize="true" >
 <FRAMESET cols="17%,83%">
 <FRAME NAME="frmleft"  SRC="menu.php?kuladi=<?echo($kuladi)?>" frameborder="0"
noresize="true">
 <FRAME NAME="frmright" SRC="bos.php" frameborder="0" noresize="true">
 </FRAMESET>  </FRAMESET></head>
```

Top.php

```
<?
 if ($kuladi != "admin") {
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
    $table="SELECT AD,SOYAD FROM omogmspf where ogrencino=$kuladi";
    $sorgu=mysql_query($table);
    $data=mysql_fetch_array($sorgu);
 }
?>
<html><head><title>DHTML Menu Sample</title></head><body >
<TABLE width="830" cellpadding="0" cellspacing="0">
  <TBODY>
    <TR bgcolor="#87cefa">
      <TD ><IMG  border="0" height="78" width="163" src="pics/topimage.gif" ></TD>
        <td width="528" style='font-size: 12px; color: black;  font-weight: bold;'>
      <?
        if ($kuladi == "admin") {
           echo ("<h3>SAYIN YETKILI</h3>");
        }else{
           echo ("<h3>SAYIN $data[0] $data[1]</h3>");
        }
      ?>
    </TR>
    <input type="hidden" name="kuladi" value="$kuladi">
  </TBODY>
</TABLE>
</body>
```

Menu.php

```
<html><head><title>WebDesignHelper.co.uk - Vertical Menu 5</title></head>
<STYLE>
{
  text-decoration: none;
  color: #FFFFFF;
  font-family: verdana, arial, helvetica, sans-serif;
  font-size: 11px;
  font-weight: bold
}
</STYLE></HEAD>
<BODY LEFTMARGIN=10 TOPMARGIN=0 RIGHTMARGIN=0 BOTTOMMARGIN=0
MARGINWIDTH=0 MARGINHEIGHT=0 >
<table border=0 cellpadding=0 cellspacing=0 WIDTH="163">
<tr><td bgcolor=#00008b WIDTH=1 height=2
COLSPAN="3"><imgsrc="images/spacer.gif" WIDTH=1 HEIGHT=1></td></tr>
<tr><td bgcolor=#00008b WIDTH=2><img src="images/spacer.gif" BORDER=0 WIDTH=1
HEIGHT=1></td>
<td bgcolor=#87cefa align="center" width=124><img src="images/spacer.gif" BORDER=0
WIDTH=1 HEIGHT=20></td>
<td bgcolor=#00008b width=2><img src="images/spacer.gif" WIDTH=1 HEIGHT=1></td>
</tr></table>
<table border=0 cellpadding=0 cellspacing=0 bgcolor="#FFFFFF" WIDTH="163"
height=80%> <tr>
<td bgcolor=#00008b WIDTH=2 rowspan=5><img src="images/spacer.gif" BORDER=0
WIDTH=1 HEIGHT=1></td>
<td bgcolor=#87cefa width=4 rowspan=4><img src="images/spacer.gif" WIDTH=1
HEIGHT=1></td>
<td BGCOLOR=#00008b colspan=5 height=1><img src="images/spacer.gif" WIDTH=1
HEIGHT=1></td>
<td bgcolor=#87cefa width=4 rowspan=4><img src="images/spacer.gif" WIDTH=1
HEIGHT=1></td>
<td bgcolor=#00008b width=2 rowspan=5><img src="images/spacer.gif" WIDTH=1
HEIGHT=1></td>
</tr><tr>
<td BGCOLOR=#00008b WIDTH=1><img src="images/spacer.gif" WIDTH=1
HEIGHT=1></td>
<td BGCOLOR=#4169e1 WIDTH=5><img src="images/spacer.gif" WIDTH=1
HEIGHT=1></td>
<td BGCOLOR=#4169e1 WIDTH="170" VALIGN="TOP" ALIGN="left">
<table BORDER=0 CELLPADDING=1 CELLSPACING=0 >
 <tr><td colspan=2><font size=1> </td></tr>
<tr BGCOLOR=#00008b><td width=10> </td><td><font face=Verdana size=2
COLOR='#FFFFFF'><b>  ANKET MENUSU  </b></font></td>
</tr>
<? if ($kuladi == "admin"): ?>
    <tr BGCOLOR=#4169e1>
    <td width=10> </td>
   <td><A HREF="Ogrlist.php" target="frmright">Öğrenci listesi</A></td>
</tr><tr BGCOLOR=#4169e1>
```

```
   <td width=10> </td>
 <td><A HREF="Bolumlist.php" target="frmright">Bölüm listesi</A></td>
</tr>
     <tr BGCOLOR=#4169e1>
     <td width=10> </td>
     <td><A HREF="anketlist.php" target="frmright"> Anket listesi</A></td>
</tr>

<? else: ?>
     <tr BGCOLOR=#4169e1>
     <td width=10> </td>
     <td><A HREF="anketsec.php?kuladi=<?echo($kuladi)?>"
target="frmright">Doldurma</A></td>
      </tr>
<? endif ?>
         <tr BGCOLOR=#4169e1>
         <td width=10> </td>
         <td><A HREF="Sonucsec.php?kuladi=<?echo($kuladi)?>"
target="frmright">Sonuçlar</A></td>
         </tr><tr> <td colspan=2><font size=1> </td>
         </tr></table><br>
         </TD>
<td BGCOLOR=#4169e1 WIDTH=5><img src="images/spacer.gif" WIDTH=1
HEIGHT=1></td>
<td BGCOLOR=#00008b WIDTH=1><img src="images/spacer.gif" WIDTH=1
HEIGHT=1></td></TR>
<TR><TD BGCOLOR=#00008b colspan=5 height=1><img src="images/spacer.gif"
WIDTH=1 HEIGHT=1></td>
</TR><TR><TD BGCOLOR=#87cefa WIDTH=1 COLSPAN=5><img
src="images/spacer.gif" WIDTH=1 HEIGHT="10"></td>
</TR><TR>
<TD BGCOLOR=#00008b WIDTH=1 COLSPAN="7" height=2><img
src="images/spacer.gif" WIDTH=1 HEIGHT=1></td>
</TR></TABLE>
     <input type="hidden" name="kuladi" value="<?echo($kuladi)?>">
</Body>
</Html>
```

Ogrlist.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="tr" lang="tr" dir="ltr">
<HEAD><META http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
 <META name="GENERATOR" content="IBM WebSphere Studio">
 <link rel="stylesheet" type="text/css" href="formstyle.css">
 <TITLE>Bölüm listesi</TITLE>
<script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
```

```
        else
                button.className = 'ButtonOut';
} </script>
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
?>
</HEAD><BODY><tr><td><BR>
<form method="POST" action="ogrgir.php">
<TABLE width="642" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
<tr><TD>
  <TABLE width="100%" class='tablecol col3' cellpadding=2 cellspacing=1>
  <TBODY>
   <TR><Th>Bolum Listesi</Th> <td>
       <select name ="bolno">
      <?
        $table="SELECT * FROM bmblmspf ORDER BY BOLNO";
          $sorgu=mysql_query($table);
        while($data=mysql_fetch_array($sorgu)) {
          if ($bolno == "") $bolno = $data[0];
          if ($bolno == $data[0]) {
                echo("<option value='$data[0]' selected>$data[1]</option>");
          }else{
            echo("<option value='$data[0]'>$data[1]</option>");
          }
        }
        ?>
      </select></td></TR> </TBODY>
  </TABLE> </td></tr>
</table>
<TABLE width="642" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
<tr><TD>
<TABLE width="" class='tablecol col4' cellpadding=2 cellspacing=1>
<TBODY>
<?
        $table="SELECT * FROM omogmspf where bolno=$bolno";
        $sorgu=mysql_query($table);
        while($data=mysql_fetch_array($sorgu)) {
            print("<tr><FONT HEIGHT=35 WIDTH=35><th><input type=radio
name=\"ogrencino\" value=$data[0]></th>\n");
          print("<td><center>$data[3]</td></td><td>   $data[0]</td>\n" );
          print("<td WIDTH=300>  $data[1]</td></td><td
width=300>  $data[2]</td></tr>\n" );
     }
?>
            </TBODY></TABLE>
</td></tr><tr></tr><tr></tr><tr></tr></table>
<TABLE width="" class='tablecol col3' cellpadding=2 cellspacing=1>
<tr><td><input type=radio name=islem value="EKLE"></td><td>EKLE
   </td>
```

```
   <td><input type=radio name=islem value="DUZELT"></td><td>DÜZELT
   </td>
   <td><input type=radio name=islem value="SIL"></td><td>SİL
   </td>
</tr></table>
<tr FONT HEIGHT=60 WIDTH=60 COLOR="RED">
   <? if ($hata == 1) print("<td>işlem tipi seçmelisiniz..</td>");
     if ($hata == 2) print("<td>Bolum kodu seçmelisiniz..</td>");
   ?>
</tr>
<br><br>
   <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""
type="submit">  ISLEM  </BUTTON>
      
   <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="reset"
value="FORMSIL">FORMU SIL</BUTTON>
     
   <br><br>
</TD></TR></TBODY></TABLE></TD></TR>
 </TD></TR>
</TBODY>
</TABLE>
</form>
  </BODY>
</HTML>
```

Ogrgir.php

```
<?
  mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
  mysql_select_db("anket") || die("veritabanı açılamadı");
  if ($islem == "") {
      header("location:Ogrlist.php?bolno=$bolno");
  }
  if (($islem == "DUZELT" || $islem == "SIL") && ($ogrencino =="")) {
      header("location:Ogrlist.php?hata=1");
  }
  if ($islem == "DUZELT") {
    $table = "SELECT * FROM omogmspf where ogrencino=$ogrencino";
    $sorgu = mysql_query($table);
    if (!mysql_num_rows($sorgu)) {
      header("location:Ogrlist.php?hata=3");
    }
    $data=mysql_fetch_array($sorgu);
    }elseif ($islem == "SIL") {
      $table="DELETE FROM omogmspf where ogrencino=$ogrencino";
      $sorgu = mysql_query($table);
      header("location:Ogrlist.php?bolno=$bolno");
```

```
    }
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
  <HEAD>
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
    <META name="GENERATOR" content="IBM WebSphere Studio">
<link rel="stylesheet" type="text/css" href="formstyle.css">
    <TITLE>Kullanıcı Bİlgileri Giriş</TITLE>
    <script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
}
</script>
  </HEAD>  <BODY>  <tr><td>  <BR>
<form method="POST" action="ogrsakla.php">
   <TABLE width="528" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
<tr><TD>
<TABLE width="100%" class='tablecol col3' cellpadding=2 cellspacing=1>
<TBODY> <tr>  <Th>Öğrenci No</Th>
 <TD>
 <? if ($islem != "DUZELT") {
     print("<input size=10 maxLength=10  type='text' name='ogrencino'    value='$data[0]'>");
  } else {
     print("$data[0]");
     print("<input type='hidden' name='ogrencino' value='$data[0]'>");
  }
 ?>
 </TD> <Th colspan=2></Th> </tr> <tr> <Th>Adi </Th>
 <TD><INPUT maxlength="20" size="20" type="text" name="ad"
value="<?print("$data[1]");?>">
 </TD>  <Th>Soyadi </Th>
 <TD><INPUT maxlength="20" size="20" type="text" name="soyad"
value="<?print("$data[2]");?>"> </TD>
  </tr> <TR> <Th>Bölüm No</Th>
<td><select name ="bolno">
<?
                    $table = "SELECT * FROM bmblmspf ORDER BY BOLNO";
                    $sorgu = mysql_query($table);
                    while($data1=mysql_fetch_array($sorgu))
                                {
                   if ($data1[0] == $data[3]) {
                        echo("<option value='$data1[0]' selected>$data1[1]</option>");
                        }else{
                         echo("<option value='$data1[0]'>$data1[1]</option>");
                        }
 ?>
                  </select></td>
```

```
                    <Th>Cinsiyet </Th>
                    <Th>
<?
        if ($data[4] == "E") {
         echo("<input  type='radio' name='cinsiyet' value='E' checked> Erkek");
         echo("<input  type='radio' name='cinsiyet' value='K'> Kız");

                          }
        else{ echo("<input  type='radio' name='cinsiyet' value='E'> Erkek");
        echo("<input  type='radio' name='cinsiyet' value='K' checked> Kız");
         }
?>
 </Th> </TR>  <TR>
<Th>Doğ Tar.(gg/aa/yyyy) </Th> <TD>
<input  size="10" maxLength="10"  type="text" name="dogtar"
value="<?print("$data[5]");?>">
                    </TD>
                    <th >Şifre </th>
                      <TD>
                    <input  size="10" maxLength="10"  type="password" name="sifre"
value="<?print("$data[6]");?>">
                    </TD>
                    </TR>
                    <input type=hidden name="islem" value="<?print("$islem");?>">
                </TBODY>
            </TABLE>
</td></tr></table>
<br><br>
   <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="submit"
value="SAKLA">Sakla  </BUTTON>
     
   <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="reset"
value="FORMSIL">FORMU SIL</BUTTON>
     
   <br><br>
</TD></TR></TBODY></TABLE></TD></TR>
         </TD>
       </TR>
   </TBODY>
</TABLE>
</form>

  </BODY>
</HTML>
```

Ogrsakla.php

```
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
    if ($islem == "DUZELT") {
        $table="UPDATE OMOGMSPF SET
ad='$ad',soyad='$soyad',bolno=$bolno,cinsiyet='$cinsiyet',dogtar='$dogtar',sifre='$sifre'
where ogrencino=$ogrencino";
    }else{
        $table="INSERT INTO OMOGMSPF
VALUES($ogrencino,'$ad','$soyad',$bolno,'$cinsiyet','$dogtar','$sifre')";
    }
    $sorgu=mysql_query($table);
    header("location:Ogrlist.php?bolno=$bolno");
?>
```

Bolumlist.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="tr" lang="tr" dir="ltr">
  <HEAD>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
    <META name="GENERATOR" content="IBM WebSphere Studio">
<link rel="stylesheet" type="text/css" href="formstyle.css">
    <TITLE>Bölüm listesi</TITLE>
    <script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
}
</script>
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
?>
  </HEAD>
  <BODY>
  <tr><td>
        <BR>

<form method="POST" action="bolgir.php">
<TABLE width="" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>

<tr><TD>
<TABLE width="" class='tablecol col3' cellpadding=2 cellspacing=1>
<TBODY>
          <tr><FONT HEIGHT=35 WIDTH=35>
```

```
            <Th>Sec</th>
            <td width=80><center>BOLUM KODU</td><td
width=530>   BOLUM ADI</td></tr>
<?
            $table="SELECT * FROM BMBLMSPF ORDER BY BOLNO";
            $sorgu=mysql_query($table);
            while($data=mysql_fetch_array($sorgu)) {
                  print("<tr><FONT HEIGHT=35 WIDTH=35><th><input type=radio
name=bolno value='$data[0]'></th>");
            print("<td width=80><center>$data[0]</td></td><td
width=530>   $data[1]</td></tr>" );
         }

?>
            </TBODY>
         </TABLE>
</td></tr>
<tr></tr><tr></tr><tr></tr>
</table>
<TABLE width="" class='tablecol col3' cellpadding=2 cellspacing=1>
<tr><td><input type=radio name=islem value="EKLE" CHECKED></td><td>EKLE
   </td>
  <td><input type=radio name=islem value="DUZELT"></td><td>DÜZELT
   </td>
  <td><input type=radio name=islem value="SIL"></td><td>SİL
   </td>
</tr>
</table>
<tr FONT HEIGHT=60 WIDTH=60 COLOR="RED">
  <? if ($hata == 1) print("<td>işlem tipi seçmelisiniz..</td>");
    if ($hata == 2) print("<td>Bolum kodu seçmelisiniz..</td>");
  ?>
</tr>
<br><br>
  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""
type="submit">  ISLEM  </BUTTON>
     
  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="reset"
value="FORMSIL">FORMU SIL</BUTTON>
    
  <br><br>
</form>
</BODY>
</HTML>
```

Bolgir.php

```
<?
  mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
  mysql_select_db("anket") || die("veritabanı açılamadı");
  if ($islem == "") {
      header("location:bolumlist.php?hata=1");
  }
  if (($islem == "DUZELT" || $islem == "SIL") && ($bolno =="")) {
      header("location:bolumlist.php?hata=2");
  }
  if ($islem == "DUZELT") {
    $table="SELECT * FROM bmblmspf where bolno=$bolno";
    $sorgu = mysql_query($table);
    if (!mysql_num_rows($sorgu)) {
      header("location:bolumlist.php?hata=3");
    }
    $data=mysql_fetch_array($sorgu);
  } elseif ($islem == "SIL") {
    $table="DELETE FROM bmblmspf where bolno=$bolno";
    $sorgu = mysql_query($table);
    header("location:bolumlist.php?hata=0");
  }
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
  <HEAD>
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
    <META name="GENERATOR" content="IBM WebSphere Studio">
<link rel="stylesheet" type="text/css" href="formstyle.css">
    <TITLE>Kullanıcı Bİlgileri Giriş</TITLE>
    <script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
}
</script>

  </HEAD>
  <BODY>
  <tr><td>
        <BR>

<form method="POST" action="bolsakla.php">
<TABLE width="528" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
<tr><TD>
<TABLE width="100%" class='tablecol col3' cellpadding=2 cellspacing=1>
<TBODY>
                <tr>
```

```
                    <Th>Bölüm No</Th>
                    <TD>
                    <? if ($islem != "DUZELT") {
                        print("<input size=3 maxLength=3  type='text' name='bolno'
value='$data[0]'>");
                    } else {
                        print("$data[0]");
                        print("<input type='hidden' name='bolno' value='$data[0]'>");
                    }
                    ?>
                    </TD>

                  </tr>
                  <tr>
                    <Th>Bölüm Adı </Th>
                    <TD><INPUT maxlength="30" size="30" type="text" name="bolad"
value="<? print("$data[1]") ?>">
                    </TD>
                  </tr>
                    <input type=hidden name="islem" value="<?print("$islem");?>">
                </TBODY>
              </TABLE>
</td></tr></table>
<br><br>
  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="submit"
value="SAKLA">SAKLA  </BUTTON>
    
  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="reset"
value="FORMSIL">FORMU SIL</BUTTON>
    
  <br><br>
</TD></TR></TBODY></TABLE></TD></TR>
        </TD>
      </TR>
  </TBODY>
</TABLE>
</form>

  </BODY>
</HTML>
```

Bolsakla.php

```
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
    if ($islem == "DUZELT") {
      $table="UPDATE BMBLMSPF SET BOLAD='$bolad' where bolno=$bolno";
```

```
    }else{
      $table="INSERT INTO BMBLMSPF VALUES($bolno,'$bolad')";
    }
    $sorgu=mysql_query($table);
    header("location:bolumlist.php?hata=0");
?>
```

Anketlist.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="tr" lang="tr" dir="ltr">
  <HEAD>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
    <META name="GENERATOR" content="IBM WebSphere Studio">
<link rel="stylesheet" type="text/css" href="formstyle.css">
    <TITLE>Bölüm listesi</TITLE>
    <script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
}
</script>
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
    $table="SELECT * FROM AMANMSPF ORDER BY ANKETNO";
    $sorgu=mysql_query($table);
?>
  </HEAD>
  <BODY>
  <tr><td>
        <BR>

<form method="POST" action="anketgir.php">
<TABLE width="" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>

<tr><TD>
<TABLE width="" class='tablecol col3' cellpadding=2 cellspacing=1>
<TBODY>
          <tr><FONT HEIGHT=35 WIDTH=35>
            <Th>Sec</th>
            <td width=80><center>ANKET NO</td><td
width=460>   KONUSU</td><TD>YORUM</TD></tr>
<?
              while($data=mysql_fetch_array($sorgu)) {
                    print("<tr><FONT HEIGHT=35 WIDTH=35><th><input type=radio
name=anketno value='$data[0]'></th>");
```

```
                print("<td width=80><center>$data[0]</td></td><td
width=460>   $data[1]</td>");
                if ($data[2] == 0) {
                        print("<td>Yorum yok</td></tr>");
                }else{
                        print("<td>Yorum var</td></tr>");
                }
        }

?>
            </TBODY>
          </TABLE>
</td></tr>
<tr></tr><tr></tr><tr></tr>
</table>
<TABLE width="" class='tablecol col3' cellpadding=2 cellspacing=1>
<tr>
  <td><input type=radio name=islem value="SORUGIR"></td><td>SORU GIRISI
   </td>
  <td><input type=radio name=islem value="EKLE" CHECKED></td><td>ANKET EKLE
   </td>
  <td><input type=radio name=islem value="DUZELT"></td><td>ANKET DÜZELT
   </td>
  <td><input type=radio name=islem value="SIL"></td><td>ANKET
SIL   </td>
</tr>
</table>
<tr FONT HEIGHT=60 WIDTH=60 COLOR="RED">
  <? if ($hata == 1) print("<td>işlem tipi seçmelisiniz..</td>");
    if ($hata == 2) print("<td>Anket kodu seçmelisiniz..</td>");
  ?>
</tr>
<br><br>
  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""
type="submit">  ISLEM  </BUTTON>
     
  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="reset"
value="FORMSIL">FORMU SIL</BUTTON>
    
  <br><br>
</TD></TR></TBODY></TABLE></TD></TR>
        </TD>
      </TR>
  </TBODY>
</TABLE>
</form>
  </BODY>
</HTML>
```

Anketgir.php

```
<?
  mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
  mysql_select_db("anket") || die("veritabanı açılamadı");
  if ($islem == "") {
      header("location:Anketlist.php?hata=1");
  }
  if (($islem == "DUZELT" || $islem == "SIL" || $islem == "SORUGIR") && ($anketno
==""))  {
      header("location:Anketlist.php?hata=2");
  }
  if ($islem == "SORUGIR") {
      header("location:Sorulist.php?anketno=$anketno");
  }
  if ($islem == "DUZELT") {
    $table="SELECT * FROM amanmspf where anketno=$anketno";
    $sorgu = mysql_query($table);
    if (!mysql_num_rows($sorgu)) {
      header("location:Anketlist.php?hata=3");
    }
    $data=mysql_fetch_array($sorgu);
    $table="SELECT * FROM asanscpf where anketno=$anketno";
    $sorgu = mysql_query($table);
  }  elseif ($islem == "SIL") {
    $table="DELETE FROM amanmspf where anketno=$anketno";
    $sorgu = mysql_query($table);
    $table="DELETE FROM asanscpf where anketno=$anketno";
    $sorgu = mysql_query($table);
    header("location:Anketlist.php?hata=0");
  } else {
    $table="SELECT max(anketno) as anketno FROM amanmspf";
    $sorgu=mysql_query($table);
    $anketno=mysql_result($sorgu,0,"anketno");
    $anketno++;
  }
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
  <HEAD>
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
    <META name="GENERATOR" content="IBM WebSphere Studio">
<link rel="stylesheet" type="text/css" href="formstyle.css">
    <TITLE>Kullanıcı Bİlgileri Giriş</TITLE>
    <script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
}
```

```
</script>
  </HEAD>
  <BODY>
  <tr><td>
        <BR>

<form method="POST" action="anketsakla.php">
   <TABLE width="528" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
<tr><TD>
<TABLE width="100%" class='tablecol col3' cellpadding=2 cellspacing=1>
<TBODY>
                <tr>
                  <Th>Anket No</Th>
                  <Td colspan=3><? echo($anketno) ?></TD>
                  <input type=hidden name="anketno" value="<? echo($anketno) ?>">
                 </tr>
                       <tr>
                  <Th>Anket Adı</Th>
                  <Td colspan=3><INPUT maxlength="50" size="79" type="text"
name="anketad" value="<?echo($data[1])?>"></TD>
                  <input type=hidden name="islem" value="<?echo($islem)?>">
                </tr>
                <tr>
                  <Th>Yorum </Th>
                  <? if ($data[2] == 0): ?>
                    <TD colspan=3>
                    <INPUT type="radio" name="yorum" value="1">  Var
                              <INPUT type="radio" name="yorum" checked
Value="0">  Yok
                            </TD>
                  <? else: ?>
                    <TD colspan=3>
                    <INPUT type="radio" name="yorum" checked
value="1">  Var
                              <INPUT type="radio" name="yorum"
Value="0">  Yok
                    </TD>
                  <? endif ?>
                </tr>
                <tr>
                  <? if ($data=mysql_fetch_array($sorgu)) {
                       $degisken1=$data[1];
                       $degisken2=$data[2];
                     } else {
                       $degisken1="";
                       $degisken2="";
                     }
                  ?>
                  <Th>1.Seçenek</Th>
                  <Td><INPUT maxlength="2" size="5" type="text" name=seckod[]
value="<?print("$degisken1")?>"></td>
```

```
                <th>Açıklama</th>
                <td><INPUT maxlength="20" size="50" type="text" name=secad[]
value="<?print("$degisken2")?>"></TD>
            </tr>
              <? if ($data=mysql_fetch_array($sorgu)) {
                  $degisken1=$data[1];
                  $degisken2=$data[2];
                } else {
                  $degisken1="";
                  $degisken2="";
                }
              ?>
            <tr>
                    <Th>2.Seçenek</Th>
                <Td><INPUT maxlength="2" size="5" type="text" name=seckod[]
value="<?print("$degisken1")?>"></td>
                <th>Açıklama</th>
                <td><INPUT maxlength="20" size="50" type="text" name=secad[]
value="<?print("$degisken2")?>"></TD>
            </tr>
              <? if ($data=mysql_fetch_array($sorgu)) {
                  $degisken1=$data[1];
                  $degisken2=$data[2];
                } else {
                  $degisken1="";
                  $degisken2="";
                }
              ?>
            <tr>
              <Th>3.Seçenek</Th>
              <Td><INPUT maxlength="2" size="5" type="text" name=seckod[]
value="<?print("$degisken1")?>"></td>
              <th>Açıklama</th>
              <td><INPUT maxlength="20" size="50" type="text" name=secad[]
value="<?print("$degisken2")?>"></TD>
            </tr>
              <? if ($data=mysql_fetch_array($sorgu)) {
                  $degisken1=$data[1];
                  $degisken2=$data[2];
                } else {
                  $degisken1="";
                  $degisken2="";
                }
              ?>
            <tr>
                    <Th>4.Seçenek</Th>
              <Td><INPUT maxlength="2" size="5" type="text" name=seckod[]
value="<?print("$degisken1")?>"></td>
              <th>Açıklama</th>
              <td><INPUT maxlength="20" size="50" type="text" name=secad[]
value="<?print("$degisken2")?>"></TD>
```

```
                </tr>
                  <? if ($data=mysql_fetch_array($sorgu)) {
                       $degisken1=$data[1];
                       $degisken2=$data[2];
                     } else {
                       $degisken1="";
                       $degisken2="";
                     }
                  ?>
                <tr>
                  <Th>5.Seçenek</Th>
                  <Td><INPUT maxlength="2" size="5" type="text" name=seckod[]
value="<?print("$degisken1")?>"></td>
                  <th>Açıklama</th>
                  <td><INPUT maxlength="20" size="50" type="text" name=secad[]
value="<?print("$degisken2")?>"></TD>
                </tr>

              </TBODY>
            </TABLE>
</td></tr></table>
<br><br>
  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="submit"
value="SAKLA">Sakla  </BUTTON>
    
  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="reset"
value="FORMSIL">Formu Sil</BUTTON>
    
  <br><br>
</TD></TR></TBODY></TABLE></TD></TR>
        </TD>
      </TR>
  </TBODY>
</TABLE>
</form>

 </BODY>
</HTML>
```

Anketsakla.php

```
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
    if ($islem == "DUZELT") {
       $table="DELETE FROM amanmspf where anketno=$anketno";
       $sorgu = mysql_query($table);
       $table="DELETE FROM asanscpf where anketno=$anketno";
```

```
        $sorgu = mysql_query($table);
    }
    $table="INSERT INTO AMANMSPF VALUES($anketno,'$anketad',$yorum)";
    $sorgu=mysql_query($table) ;
    $table="SELECT max(anketno) as anketno FROM amanmspf";
    $sorgu=mysql_query($table);
    $anketno=mysql_result($sorgu,0,"anketno");
    for ($sayac=0; $sayac < 4 ; $sayac++) {
      if ($seckod[$sayac] != "") {
        $table="INSERT INTO ASANSCPF
VALUES($anketno,'$seckod[$sayac]','$secad[$sayac]')";
        $sorgu=mysql_query($table) ;
      }

    }
  header("location:Anketlist.php?hata=0:");
?>
```

Sorulist.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="tr" lang="tr" dir="ltr">
  <HEAD>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
    <META name="GENERATOR" content="IBM WebSphere Studio">
<link rel="stylesheet" type="text/css" href="formstyle.css">
    <TITLE>Bölüm listesi</TITLE>
    <script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
}
</script>
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
?>
  </HEAD>
  <BODY>
  <tr><td>
        <BR>

<form method="POST" action="sorugir.php">
<TABLE width="" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
```

```
<tr><TD>
<TABLE width="" class='tablecol col3' cellpadding=2 cellspacing=1>
<TBODY>

<?
        print("<input type=hidden name='anketno' value=$anketno>");
            $table="SELECT * FROM ASANSRPF where anketno=$anketno ORDER BY
SIRA";
            $sorgu=mysql_query($table);
            while($data=mysql_fetch_array($sorgu)) {
                print("<tr><FONT HEIGHT=35 WIDTH=35><th><input type=radio
name=sira value='$data[1]'></th>");
            print("<td width=50><center>$data[1]</td></td><td
width=544>   $data[2]</td></tr>" );
        }

?>
          </TBODY>
        </TABLE>
</td></tr>
<tr></tr><tr></tr><tr></tr>
</table>
<TABLE width="" class='tablecol col3' cellpadding=2 cellspacing=1>
<tr><td><input type=radio name=islem value="EKLE" CHECKED></td><td>EKLE
   </td>
  <td><input type=radio name=islem value="DUZELT"></td><td>DÜZELT
   </td>
  <td><input type=radio name=islem value="SIL"></td><td>SİL
   </td>
</tr>
</table>
<tr FONT HEIGHT=60 WIDTH=60 COLOR="RED">
  <? if ($hata == 1) print("<td>işlem tipi seçmelisiniz..</td>");
    if ($hata == 2) print("<td>Bolum kodu seçmelisiniz..</td>");
  ?>
</tr>
<br><br>
  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""
type="submit">  ISLEM  </BUTTON>
     
  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="reset"
value="FORMSIL">FORMU SIL</BUTTON>
    
  <br><br>
</TD></TR></TBODY></TABLE></TD></TR>
        </TD>
      </TR>
  </TBODY>
</TABLE>
```

```
</form>
  </BODY>
</HTML>
```

Sorugir.php

```
<?
  mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
  mysql_select_db("anket") || die("veritabanı açılamadı");
  if ($islem == "") {
      header("location:Sorulist.php?anketno=$anketno");
  }
  if (($islem == "DUZELT" || $islem == "SIL") && ($sira =="")) {
      header("location:Sorulist.php?anketno=$anketno");
  }
  $table="SELECT anketad FROM amanmspf where anketno=$anketno";
  $sorgu=mysql_query($table);
  $anketad=mysql_result($sorgu,0,"anketad");
  if ($islem == "DUZELT") {
    $table = "SELECT * FROM asansrpf where anketno=$anketno and sira=$sira";
    $sorgu = mysql_query($table);
    if (!mysql_num_rows($sorgu)) {
      header("location:Sorulist.php?anketno=$anketno");
    }
    $data=mysql_fetch_array($sorgu);
    }elseif ($islem == "SIL") {
      $table="DELETE FROM asansrpf where anketno=$anketno and sira=$sira";
      $sorgu = mysql_query($table);
      header("location:Soruist.php?anketno=$anketno");
    } else {
      $table="SELECT max(sira) as sira FROM asansrpf where anketno=$anketno";
      $sorgu=mysql_query($table);
      $sira =mysql_result($sorgu,0,"sira");
      $sira++;
    }
?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
  <HEAD>
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
    <META name="GENERATOR" content="IBM WebSphere Studio">
<link rel="stylesheet" type="text/css" href="formstyle.css">
    <TITLE>Kullanıcı Bİlgileri Giriş</TITLE>
    <script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
```

```
}
</script>
  </HEAD>
  <BODY>
  <tr><td>
        <BR>

<form method="POST" action="sorusakla.php">
   <TABLE width="528" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
<tr><TD>
<TABLE width="100%" class='tablecol col3' cellpadding=2 cellspacing=1>
<TBODY>
                <tr>
                  <Th> Anket  </Th>
                        <td><?echo($anketad)?></td>
                        <input type="hidden" name="anketno"
value="<?echo($anketno)?>">
                        <input type="hidden" name="islem" value="<?echo($islem)?>">

                      </tr>
                <tr>
                  <Th>Soru no</Th>
                        <td><?echo($sira)?></td>
                        <input type="hidden" name="sira" value="<?echo($sira)?>">

                      </tr>
                 <tr>
                  <Th>Soru</Th>
                  <Td><TEXTAREA ROWS=3 COLS=70   name="soru" value=<?
echo("$data[2]") ?>></TEXTAREA></TD>
                </tr>

              </TBODY>
            </TABLE>
</td></tr></table>
<br><br>
   <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="submit"
value="SAKLA">Sakla  </BUTTON>
     
   <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="reset"
value="FORMSIL">Formu Sil</BUTTON>
   <br><br>
</TD></TR></TBODY></TABLE></TD></TR>
        </TD>
      </TR>
   </TBODY>
</TABLE>
</form>
```

```
  </BODY>
</HTML>
```

Sorusakla.php

```
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
    if ($islem == "DUZELT") {
        $table="UPDATE ASANSRPF SET soru='$soru' where anketno=$anketno and
sira=$sira";
    }else{
        $table="INSERT INTO asansrpf VALUES('$anketno','$sira','$soru')";
    }
    $sorgu=mysql_query($table);
    header("location:Sorulist.php?anketno=$anketno");
?>
```

Anketsec.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
  <HEAD>
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
    <META name="GENERATOR" content="IBM WebSphere Studio">
    <link rel="stylesheet" type="text/css" href="formstyle.css">
    <TITLE>Kullanıcı Bilgileri Giriş</TITLE>
    <script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
}
</script>
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
    $table="SELECT * FROM amanmspf ORDER BY ANKETNO";
    $sorgu=mysql_query($table);
?>
  </HEAD>
  <BODY>
  <tr><td>
   <FORM method="POST" action="anket_doldur.php">
   <TABLE width="650" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
   <tr><TD>
    <TABLE width="100%" class='tablecol col3' cellpadding=2 cellspacing=1>
```

```
    <TBODY>

                <TR>
                  <input type=hidden name=kuladi value="<?echo($kuladi)?>">
                  <Th>Anket Doldurma</Th>
                        <td>
                             <select name ="anketno">

                  <?
                   while($data=mysql_fetch_array($sorgu))
                                {
                                echo("<option value='$data[0]'>$data[1]</option>");
                                }
                         ?>
                         </select>

                              <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="submit"
value="SAKLA">GIT  </BUTTON>
                                  </td>

                </TR>
     </TBODY>
    </TABLE>
    </td></tr></table>
              <? if ($hata == 1)  { print("<TR></TR><TR></TR><FONT
COLOR='RED'><H1>BU ANKETI ZATEN DOLDURMUSSUNUZ. !</TR>");} ?>
</FORM>


</TD></TR></TBODY></TABLE></TD></TR>
        </TD>
      </TR>
    </TBODY>
</TABLE>
  </BODY>
</HTML>
```

Anket_doldur.php

```
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
    $table="SELECT * FROM amanmspf where anketno=$anketno";
    $sorgu=mysql_query($table);
    $data=mysql_fetch_array($sorgu);
    $anketad=$data[1];
    $yorum=$data[2];
    $table="SELECT * FROM asanscpf where anketno=$anketno ORDER BY SECKOD";
    $sorgu=mysql_query($table);
```

```
    while($data=mysql_fetch_array($sorgu)) {
          $seckod1[]= $data[1];
        $secad[]  = $data[2];
    }
    $table="SELECT * FROM acancvpf where anketno=$anketno and ogrencino=$kuladi";
    $sorgu=mysql_query($table);
    if(mysql_num_rows($sorgu)) {
      header("location:Anketsec.php?kuladi=$kuladi&hata=1");
    }
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
  <HEAD>
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
    <META name="GENERATOR" content="IBM WebSphere Studio">
<link rel="stylesheet" type="text/css" href="formstyle.css">
    <TITLE>Kullanıcı Bİlgileri Giriş</TITLE>
    <script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
}
</script>
  </HEAD>
  <BODY>
  <tr><td>
        <BR>

<form method="POST" action="Cevapsakla.php">
<TABLE width="650" cellpadding=0 cellspacing=0 border=0 bgcolor='WHITE'>
<tr><TD>
<TABLE width="650" class='tablecol col3' cellpadding=2 cellspacing=1>
  <TBODY>
      <?
       for($i=0; $i<5; $i++) {
          if ($seckod1[$i] != "") {
              print("<tr><td WIDTH=40>$seckod1[$i]</td><td>$secad[$i]</td></tr>");
          }
       }
      ?>
   </TBODY>
  </TABLE>
</td></tr>
<tr></tr><tr></tr><tr></tr><tr></tr><tr></tr><tr></tr><tr></tr>
</table>

<TABLE width="650" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
<tr><TD>
<TABLE width="650" class='tablecol col3' cellpadding=2 cellspacing=1>
```

```
  <TBODY>
                <tr>
              <td><? echo($anketno) ?></td>
              <td><? echo($anketad) ?></td>
              <CENTER>
                <?
              foreach ($seckod1 as $deger) {print("<td>$deger</td>");}
            ?>
              </CENTER>
            </tr>
                <?
                $table="SELECT * FROM asansrpf where anketno=$anketno";
                $sorgu=mysql_query($table);
                while($data=mysql_fetch_array($sorgu)) {
                print("<input type='hidden' name='sira[]' value='$data[1]'\n");
                        print("<tr><th> $data[1] </th><td> $data[2] </td><CENTER>\n" );
                        foreach ($seckod1 as $deger) {print("<td><input type='radio'
name='seckod$data[1]' value='$deger'></td>");}
                        print("<CENTER></tr>");
            }
            ?>
   </TBODY>
  </TABLE>
</td></tr>
<tr></tr><tr></tr><tr></tr>
</table>
<TABLE width="650" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
<tr><TD>
<TABLE width="650" class='tablecol col3' cellpadding=2 cellspacing=1>
  <TBODY>
<?
   if ($yorum) {
      echo("<tr><td>YORUM</td><TD><TEXTAREA ROWS=5 COLS=82
name='yorum'></TEXTAREA></tr>");
   }
?>
  </TBODY>
  </TABLE>
</td></tr>
<tr></tr><tr></tr><tr></tr>
</table>

<br><br>
   <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="submit"
value="SAKLA">Sakla  </BUTTON>
     
   <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="reset"
value="FORMSIL">Formu Sil</BUTTON>
     
```

```
   <br><br>
<input type=hidden name="kuladi" value="<?echo($kuladi)?>">
<input type=hidden name="anketno" value="<?echo($anketno)?>">
</form>
</BODY>
</HTML>
```

Cevapsakla.php

```
<?
   mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
   mysql_select_db("anket") || die("veritabanı açılamadı");
   foreach($sira as $sayac) {
      $kod=${"seckod$sayac"};
      $table="INSERT INTO acancvpf VALUES($anketno,$kuladi,$sayac,'$kod')";
      $sorgu=mysql_query($table);
   }
   if ($yorum != "") {
      $table="INSERT INTO ayanyrpf VALUES($anketno,$kuladi,'$yorum')";
      $sorgu=mysql_query($table);
   }
   header("location:Anketsec.php?anketno=$anketno&kuladi=$kuladi");
?>
```

Sonucsec.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
  <HEAD>
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
    <META name="GENERATOR" content="IBM WebSphere Studio">
    <link rel="stylesheet" type="text/css" href="formstyle.css">
    <TITLE>Kullanıcı Bilgileri Giriş</TITLE>
    <script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
}
</script>
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
    $table="SELECT * FROM amanmspf ORDER BY ANKETNO";
    $sorgu=mysql_query($table);
?>
  </HEAD>
```

```
<BODY>
<tr><td>
 <FORM method="POST" action="Sonuclar.php">
 <TABLE width="650" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
 <tr><TD>
  <TABLE width="100%" class='tablecol col3' cellpadding=2 cellspacing=1>
  <TBODY>

            <TR>
              <input type=hidden name=kuladi value="<?echo($kuladi)?>">
              <Th>Anket Sonucları Görme</Th>
                    <td>
                          <select name ="anketno">

              <?
               while($data=mysql_fetch_array($sorgu))
                              {
                              echo("<option value='$data[0]'>$data[1]</option>");
                              }
                    ?>
                    </select>

                  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""  type="submit"
value="SAKLA">GIT  </BUTTON>
                              </td>

            </TR>
   </TBODY>
  </TABLE>
  </td></tr></table>
          <? if ($hata == 1)  { print("<TR></TR><TR></TR><FONT
COLOR='RED'><H1>BU ANKETI HENUZ KIMSE DOLDURMAMIS. !</TR>");} ?>
</FORM>


</TD></TR></TBODY></TABLE></TD></TR>
       </TD>
     </TR>
  </TBODY>
</TABLE>
  </BODY>
</HTML>
```

Sonuclar.php

```
<?
   mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
   mysql_select_db("anket") || die("veritabanı açılamadı");
   $table="select count(*) as sayi from acancvpf where anketno=$anketno";
```

```
    $sorgu=mysql_query($table);
    $data= mysql_fetch_array($sorgu);
    if($data[0] == 0) {
     header("location:Sonucsec.php?hata=1");
    }

    $table="SELECT * FROM amanmspf where anketno=$anketno";
    $sorgu=mysql_query($table);
    $data=mysql_fetch_array($sorgu);
    $anketad=$data[1];

    $table="SELECT count(*) FROM ayanyrpf where anketno=$anketno";
    $sorgu=mysql_query($table);
    $data=mysql_fetch_array($sorgu);
    $yorum=$data[0];

    $table="SELECT * FROM asanscpf where anketno=$anketno ORDER BY SECKOD";
    $sorgu=mysql_query($table);
    while($data=mysql_fetch_array($sorgu)) {
        $seckod1[]= $data[1];
        $secad[]  = $data[2];
        $secsay++;
    }
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="tr" lang="tr" dir="ltr">
  <HEAD>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
    <META name="GENERATOR" content="IBM WebSphere Studio">
<link rel="stylesheet" type="text/css" href="formstyle.css">
    <TITLE>Bölüm listesi</TITLE>
    <script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
}
</script>
 </HEAD>
  <BODY>
  <tr><td>
        <BR>

<form method="POST" action="Yorumlar.php">
<TABLE width="650" cellpadding=0 cellspacing=0 border=0 bgcolor='WHITE'>
<tr><TD>
<TABLE width="650" class='tablecol col3' cellpadding=2 cellspacing=1>
  <TBODY>
     <?
      for($i=0; $i<5; $i++) {
```

```
        if ($seckod1[$i] != "") {
            print("<tr><td WIDTH=40>$seckod1[$i]</td><td>$secad[$i]</td></tr>");
        }
      }
    ?>
  </TBODY>
 </TABLE>
</td></tr>
<tr></tr><tr></tr><tr></tr><tr></tr><tr></tr><tr></tr>
</table>
<TABLE width="650" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
<tr><TD>
<TABLE width="650" class='tablecol col3' cellpadding=2 cellspacing=1>
    <TBODY>
              <tr>
            <td><? echo($anketno) ?></td>
            <td><? echo($anketad) ?></td>
            <td>KODU</td>
            <td>ACIKLAMASI</td>
            <td>TERCIH SAYISI</td>
          </tr>
            <?
            $table="select d1.sira,d1.soru,d2.seckod,d3.secad,count(*) as sayi from
asansrpf d1, acancvpf d2, asanscpf d3 where d1.anketno=$anketno and
d1.anketno=d2.anketno and d1.sira=d2.sira and d2.anketno=d3.anketno and
d2.seckod=d3.seckod group by  d1.sira,d1.soru,d2.seckod,d3.secad order by
d1.sira,d1.soru,d2.seckod";
          $sorgu=mysql_query($table);
            while($data=mysql_fetch_array($sorgu)) {

print("<tr><tD>$data[0]</tD><td>$data[1]</td><td>$data[2]</td><td>$data[3]</td><td>$d
ata[4]</td></tr>\n" );
          }
          ?>
          <input type=hidden name="anketno" value="<?echo($anketno)?>"
    </TBODY>
</TABLE>
</td></tr>
<tr></tr><tr></tr><tr></tr>
</table>
<br><br>
<? if($yorum): ?>
  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""
type="submit">  YORUMLAR  </BUTTON>
     
<?endif?>
  <br><br>
</TD></TR></TBODY></TABLE></TD></TR>
       </TD>
     </TR>
```

```
    </TBODY>
</TABLE>
</form>
  </BODY>
</HTML>
```

Yorumlar.php

```
<?
    mysql_connect("localhost","root","qwerty") || die ("makinaya erişilemedi");
    mysql_select_db("anket") || die("veritabanı açılamadı");
    $table="SELECT * FROM amanmspf where anketno=$anketno";
    $sorgu=mysql_query($table);
    $data=mysql_fetch_array($sorgu);
    $anketad=$data[1];
    $table="select * from ayanyrpf where anketno=$anketno";
    $sorgu=mysql_query($table);
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="tr" lang="tr" dir="ltr">
  <HEAD>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
    <META name="GENERATOR" content="IBM WebSphere Studio">
<link rel="stylesheet" type="text/css" href="formstyle.css">
    <TITLE>Bölüm listesi</TITLE>
    <script language='JavaScript'>
function ButtonOver(button,isOver) {
        if (isOver)
                button.className = 'ButtonOver';
        else
                button.className = 'ButtonOut';
}
</script>
 </HEAD>
  <BODY>
  <tr><td>
        <BR>
<form method="POST" action="Sonuclar.php">
<TABLE width="650" cellpadding=0 cellspacing=0 border=0 bgcolor="WHITE">
<tr><TD>
<TABLE width="650" class='tablecol col3' cellpadding=2 cellspacing=1>
    <TBODY>
                <tr>
            <td><? echo($anketad) ?></td>
           <input type=hidden name="anketno" value="<?echo($anketno)?>"
           </tr>
    </TBODY>
</TABLE>
</td></tr>
<tr></tr><tr></tr><tr></tr>
```

```
</table>
<TABLE width="650" cellpadding=0 cellspacing=0 border=0 bgcolor='#444444'>
<tr><TD>
<TABLE width="650" class='tablecol col3' cellpadding=2 cellspacing=1>
    <TBODY>
                <tr>
              <td>ANKET NO</td>
              <td>OGRENCI NO</td>
              <td>YORUMLAR</td>
            </tr>
                <?
              while($data=mysql_fetch_array($sorgu)) {
               print("<tr><th width=50>$data[0]</th><td
width=80>$data[1]</td><td>$data[2]</td><tr>\n" );
            }
            ?>
    </TBODY>
</TABLE>
</td></tr>
<tr></tr><tr></tr><tr></tr>
</table>
<br><br>
  <BUTTON class='ButtonOut' onMouseOver="ButtonOver(this,1)"
onMouseOut="ButtonOver(this,0)" name="CMD" onclick= ""
type="submit">  SONUCLAR  </BUTTON>
     
  <br><br>
</TD></TR></TBODY></TABLE></TD></TR>
        </TD>
      </TR>
  </TBODY>
</TABLE>
</form>
  </BODY>
</HTML>
```

Bos.php

```
<html>
<head>
<title>DHTML Menu Sample</title>
</head>

<body bgcolor="#FFFFFF" text="#000000" leftmargin="5" topmargin="10"
marginwidth="0" >
<input type="hidden" name="kuladi" value="$kuladi">
</body>
```

ÖZGEÇMİŞ

Doğum tarihi	02.03.1975	
Doğum yeri	Sivas	
Lise	1989-1992	İstanbul Şişli Lisesi
Lisans	1994-1998	Yıldız Teknik Üniversitesi Kimya Metalurji Fak. Matematik Mühendisliği Bölümü
Yüksek Lisans	1999-2002	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Mühendisliği Anabilim Dalı

Çalıştığı kurum

1998-2000	YTÜ Bilgi İşlem Müdürlüğü - Çözümleyici
2000-Devam ediyor	Güneş Sigorta A.Ş. Bilgi İşlem Müdürlüğü Programcı