

T C
S.DEMİREL ÜNİVERSİTESİ
TEKNİK EĞİTİM FAKÜLTESİ
ELEKTRONİK-BİLGİSAYAR EĞİTİMİ BÖLÜMÜ

2002 YAZ DÖNEMİ STAJ FAALİYET RAPORU

HAZIRLAYAN	ŞEVKET KESER 99/42
BÖLÜM	BİLGİSAYAR SIS. ÖĞRT.
TARİH	14/09/02

2002 YAZ DÖNEMİ
III. SINIF YAZ STAJ RAPORU

2002 YAZ DÖNEMİ STAJI KÜNYE BİLGİLERİ

HAZIRLAYAN :.....**SEVKET KESER**
PROGRAMI :.....**BİLGİSAYAR SİTEMLERİ ÖĞRETMENLİĞİ**
FAKÜLTE NO :.....**9911703042**
DÖNEM :.....**III. SINIF**
STAJ YAPTIĞI KURUM :.....**TEKFEN İNŞAAT VE TESİSAT A.Ş.**
Bilgi Teknolojileri Bölümü

STAJ DÖNEMLERİ

KURUM ADI	KONU	ÇALIŞMA SÜRESİ	
		Başlangıç	Bitiş
TEKFEN İNŞ. VE TES. A.Ş Bilgi Teknolojileri Bölümü	YAZILIM STAJI	01/07/2001	26/07/2001

Not ; Fiili olarak 01/07/02 – 14/08/02 tarihleri arasında toplam 33 iş günü çalışılmıştır.

İÇİNDEKİLER

TANITIM.....	(4)
Tekfen İnşaat ve Tesisat A.Ş tanıtım & şirket bilgileri	
YAZILIM	
0 Bilgi Teknolojileri Bölümü	
0 Yazılım Sistemi Yapısı	
0 Geliştirilen Yazılım Projeleri	(6)
0 Veri Tabanı Yapısı	(10)
0 Örnek Veri Tabanı Analizi	(17)
WEB TEKNOLOJİLERİ	
0 PHP	(19)
0 MYSQL	(22)
LINUX İŞLETİM SİSTEMİ YAPI ANALİZİ	
0 LINUX İŞLETİM SİSTEMİ ÇEKİRDEK YAPISI	(25)
0 ÇEKİRDEK DERLEME	(27)
0 QMAIL SERVER KURULUMU	(30)
TOPLAM İŞ GÜNÜ.....	(20)

Şevket Keser

Kontrol :::

:::

TEKFEN İNŞAAT VE TESİSAT A.Ş.

KURUMSAL KİMLİK, ŞİRKET YAPISI, FAALİYET ALANLARI, BİLGİ TEKNOLOJİLERİ BÖLÜMÜ ALTYAPISI VE GELECEK ÖNGÖRÜLERİ

KURUMSAL KİMLİK

Tekfen İnşaat ve Tesisat A.Ş., küreselleşme kavramının hızla yayıldığı yeni bin yılı karşılamaya hazır 2500 kişilik yönetici, teknik ve idareci kadrosu, bakımlı ve geniş makine/ekipman yelpazesi ile Türkiye'nin en köklü şirketlerinden ve uluslar arası müteahhitlerinden biridir.

ŞİRKET YAPISI

1956 yılında kurulan Tekfen İnşaat ve Tesisat A.Ş. bünyesinde birçok değişik faaliyet alanlarına sahip şirketler barındıran TEKFEN HOLDİNG'in çekirdeği ve amiral gemisi niteliğindedir. Tekfen Holding ayrıca bünyesinde birçok yabancı ortaklı şirket barındırmaktadır. Bu ortaklıklar sayesinde TEKFEN gurubu yurtdışına açılabilmiş ve dünya çapında önemli işlere imza atma başarısını göstermiştir.

Tekfen Holding bünyesindeki Diğer grup şirketlerinden birkaçı ,

Tekfen Mühendislik A.Ş

Tekfen İmalat ve Mühendislik A.Ş

Toros Gübre ve Ziraat İlaç Pazarlama A.Ş.

Mis Süt Ürünleri A.Ş.

Tekfen Dış Ticaret A.Ş

TEKFENBANK

Tekfen Yatırım ve Finansman A.Ş

Tekfen Sigorta ve Aracılık Hizmetleri A.Ş

Yabancı ortaklıklar;

HBM, Hallaesche Mitteldeutsche Bau-AG Almanya,

AZFEN Azerbaycan,

GEOTEK Gürcistan

FAALİYET ALANLARI

Tekfen İnşaat ve Tesisat A.Ş. nin faaliyet alanları: altyapı yatırımlarından ağır sanayi tesislerine; uydu kentlerden sosyal ve ticari yapılara; deniz tesislerinden enerji üretim ve dağıtımına uzanan geniş bir yelpazeyi kapsamaktadır.

BİLGİ TEKNOLOJİLERİ BÖLÜMÜ

Tekfen İnşaat ve Tesisat A.Ş. birimleri arasındaki otomasyon, iletişim ve verimliliği optimum düzeyde sağlamak için güçlü bir intranet ve internet ağı inşasını sağlamıştır. Bilgi teknolojileri Bölümü, TİTAŞ bünyesinde bulunan 150 son kullanıcıya hizmet vermenin yanı sıra, Tekfen Holding bünyesinde bulunan yaklaşık 40 son kullanıcıya da hizmet vermektedir. Bu hizmet teknik servis, ağ desteği, yazılım, bakım ve eğitim seçeneklerinden oluşan geniş bir yelpazeyi kapsamaktadır. Bu durum firmanın bilgi işlem giderlerinde önemli rahatlamalar sağlamaktadır.

Yazılım geliştirme

Firmanın Bilgi teknolojileri Bölümünde gerçekleştirilen en önemli faaliyetlerden biriye; şirketin kendi yazılımlarını kendi içersinde dışarıdan hiçbir yardım almadan geliştirmesidir.

Yazılım geliştirme sonucu ortaya çıkan programlar aşağıdaki kriterleri sağlamaktadır.

- İlgili departmanın tüm ihtiyaçlarını karşılar.
- Üzun vadeli çözümler sunar.
- İnternet teknolojilerine uyumludur. (XML ve ASP)
- Veri tabanı sistemleriyle eşgüdümlü çalışır. (SQL)

Şirket ayrıca 512 Kb büyüklüğünde internet çıkışına ve 1Gb genişliğinde intranet ağına sahiptir.

GELECEK ÖNGÖRÜSÜ & KALİTE POLİTİKASI

Tekfen İnşaat ve Tesisat A.Ş. gelişen ve değişen dünya ve değerleri karşısında iç dinamiğini aşağıdaki kriterler çerçevesinde sürdürmektedir.

- 0 Mükemmelliği hedeflemek ,
- 0 Verimli çalışmak ve katma değer sağlamak,
- 0 İşlerimizde “sürekli gelişme” ilkesine göre hareket etmek ,
- 0 Tüm yönetici ve çalışanlarımız ile birlikte kaliteden sorumlu olmak ,
- 0 Eğitim ile becerilerimizi ve kalite bilincimizi arttırmak ,
- 0 Grup çalışanları ile uyumlu, etkinliği ve yaratıcılığı teşvik etmek ,
- 0 Taşeronlarımız ile her fırsatta ortak gayeler ile sürekli gelişmeye yönelik çalışmak.

GELİŞTİRİLEN YAZILIM PROJELERİ

TİTAŞ Bilgi Teknolojileri Bölümü, şirketin tüm birimlerini otomasyona geçirmek için yaklaşık iki sene önce uzun vadeli bir proje başlamıştır.

Proje; genişleme süreci ,parçadan bütüne doğrudur.Oncelikli hedef, tüm departmanlarının ihtiyacını karşılayacak yazılımlar üretmek ve bu üretilen yazılımları daha sonra; tek veri tabanında birleştirerek, birbirleriyle uyumlu olarak çalışır hale getirmektir.Proje geliştirme sürecinde dikkate alınan kıstaslar aşağıda maddelenmiştir.

- Proje, şirket yapısına uygun olarak geliştirilmelidir.
- İlgili departmanın tüm ihtiyaçlarını karşılamalıdır.
- Geliştirilen diğer projelerle uyumlu çalışabilmelidir..
- Geliştirilmesi düşünülen projelerle uyumlu çalışabilme özelliğine sahip olmalıdır.
- Proje, grup yapısı içinde ve kendi içinde geliştirilmeye müsait olmalıdır.
- Diğer projelerle ortak bir veri tabanını kullanmalıdır.

Şirket içinde geliştirilen projeler ve bölüm bilgileri aşağıda verilmiştir.

Not: Projelerle ilgili ayrıntılı bilgi, yetkililerin izin verdiği kadarıyla verilmiştir.

PERSİS

Personel kontrol ve yönetim sistemi.Firma bünyesinde çalışan personelin kayıtlarının tutulduğu programdır.

Teknik özellikler ;

Geliştirme ortamı : Delphi 5
Veri tabanı sunucusu : MSSQL 7.0
Yapı : exe dosya
Buyukluk : 7,5 Mb
Kullanıcı sayısı : 5
Network yeri : //jupiter/./persis.exe
Table sayısı : 43

**FİNSİS**

Finans kontrol ve yönetim sistemi. Finansman bölümünün kayıtların tutulduğu programdır.

Teknik özellikler ;

Geliştirme ortamı : Delphi 5
Veri tabanı sunucusu : MSSQL 7.0
Yapı : exe dosya
Buyukluk : 3.07 Mb
Kullanıcı sayısı : 3
Network yeri : //jupiter/./finsis.exe

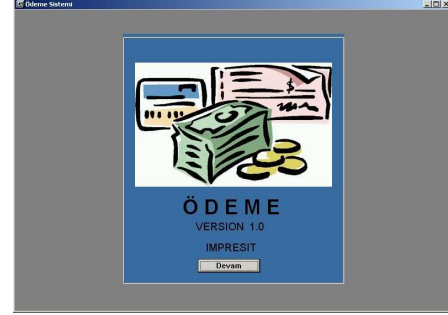


ÖDEMESİS

Tekfen impresit ödeme ve takip sistemi.
Muhasebe bölümü tarafından kullanılan ödeme takip programıdır.

Teknik özellikler ;

Geliştirme ortamı : Delphi 5
Veri tabanı sunucusu : MSSQL 7.0
Yapı : exe dosya
Buyukluk : 1.12 Mb
Kullanıcı sayısı : 3
Network yeri : //jupiter/./odemesis.exe

**BİLGİSİS**

İhale grubu, satınalma firma bilgileri takip bilgileri.

Teknik özellikler ;

Geliştirme ortamı : Delphi 5
Veri tabanı sunucusu : MSSQL 7.0
Yapı : exe dosya
Buyukluk : 1.03 Mb
Kullanıcı sayısı : 14
Network yeri : //jupiter/./bilgisis.exe

**SATSİS ;**

Lojistik grubu satınalma bilgileri takip ve kontrol programıdır.

Teknik özellikler ;

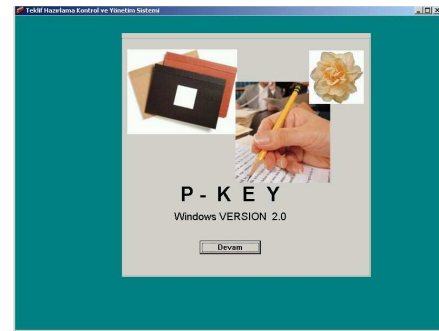
Geliştirme ortamı : Delphi 5
Veri tabanı sunucusu : MSSQL 7.0
Yapı : exe dosya
Buyukluk : 1.47 Mb
Kullanıcı sayısı : 14
Network yeri : //jupiter/./satsis.exe

**TEKLİFSİS**

İhale bölümü teklif hazırlama ve kontrol takip programıdır.

Teknik özellikler ;

Geliştirme ortamı : Delphi 5
Veri tabanı sunucusu : MSSQL 7.0
Yapı : exe dosya
Buyukluk : 1.78Mb
Kullanıcı sayısı : 3
Network yeri : //jupiter/./satsis.exe



İSTEKSİS

Proje istekleri yönetim ve kontrol programıdır. Program tarafından firmanın satınalma sistemi takip edilir. İsteksis iki arabirim tarafından kullanılabilir. Birinci arayüz **delphi 5** ile geliştirilmiş .exe programıdır. Veri tabanına buradan erişilir ve programla ilgili tüm işlemler bu arabirimden gerçekleştirilebilir. Programın kullanılması için ikinci bir arabirimde web arayuzudur. Bu arayüz sayesinde, programa internet üzerinden erişilebilir. Web arayuzuyla, exe arayüzüyle gerçekleştirilebilecek tüm işlemler gerçekleştirilebilir. Bu özellik kullanıcın ofiste ve işte olmadığı zamanlarda da programı hakları doğrultusunda kullanmasını sağlar.

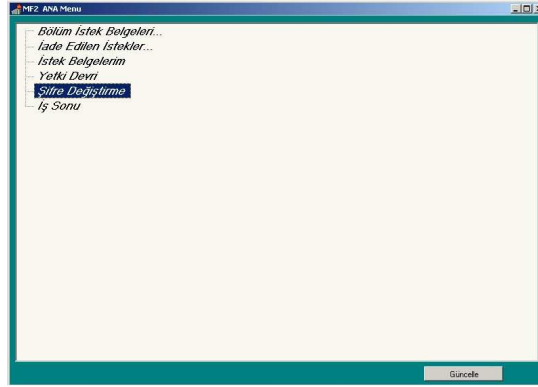
Aşağıda bu arayüzlerle ilgili ayrıntılı bilgi verilmiştir.

İSTEKSİS EXE ARABİRİMİ

İsteksis exe arabirimi yaklaşık 1.01 MB büyüklüğünde bir program parçasıdır.

Teknik özellikler ;

Geliştirme ortamı : Delphi 5
Veri tabanı sunucusu : MSSQL 7.0
Yapı : exe dosya
Buyukluk : 1.01Mb
Kullanıcı sayısı : Bölümlerin satınalma görevlileri
Table Sayısı : 27
Network yeri jupiter/././isteksis.exe

**İSTEKSİS WEB ARABİRİMİ,**

Exe arayüzünden gerçekleştirilen hertürlü program fonksiyonu, bu arabirim sayesinde web üzerinden de gerçekleştirilebilmektedir. Aşağıda programla ilgili teknik bilgiler ve ekran görüntüleri verilmiştir.

İsteksis web Arabirimi

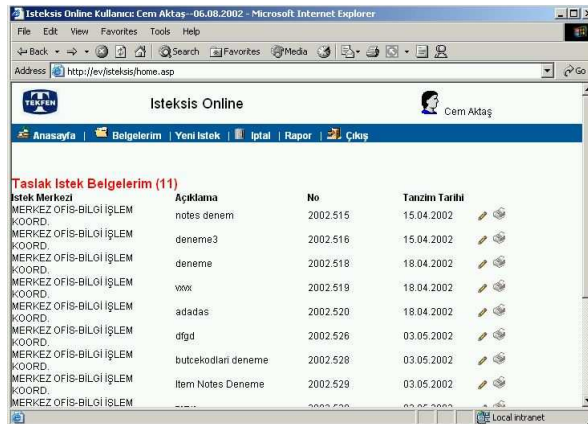
Teknik özellikler ;

Geliştirme ortamı : FrontPage 2000
Veri tabanı sunucusu : MSSQL 7.0
Yapı : xml/asp/http kodları
Buyukluk : 1.01Mb
Kullanıcı sayısı : Bölümlerin satınalma görevlileri

Table Sayısı : 27

Network yeri ;

http://ev/isteksis/bmi2s.asp



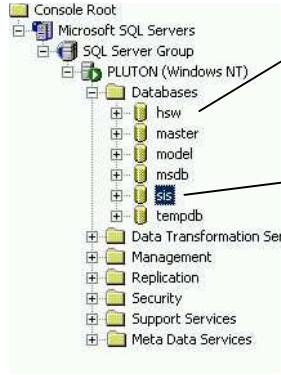
Kontrol :::

:::

VERİ TABANI YAPISI

Bilgi teknolojileri bölümü, firmanın network,donanım,yazılım ihtiyacından başka veri depolama ihtiyacında karşılamaktadır.Bu işi,sistemde MS SQL 7.0 yüklü server karşılamaktadır.Gün boyunca yukarıda anlatılan programlardan üretilen datalar SQL server tarafından, saklanmakta,işlenmekte ve yedeklenmektedir.Bunun dışında kullanıcılar tarafından üretilen datalar, 3 ayrı server da tutulmaktadır.Bu üç serverdaki datalar gün sonunda yedeklenmekte ve yedekler iki hafta boyunca saklanmaktadır.Gün sonunda yedeklenen toplam data büyüklüğü 3X24 GB büyüklüğündedir.

Sistem, “sis” ve “hsw” olarak üzere iki ayrı veri tabanı üzerine kurulmuştur.Aşağıdaki şekilde de görüldüğü gibi;

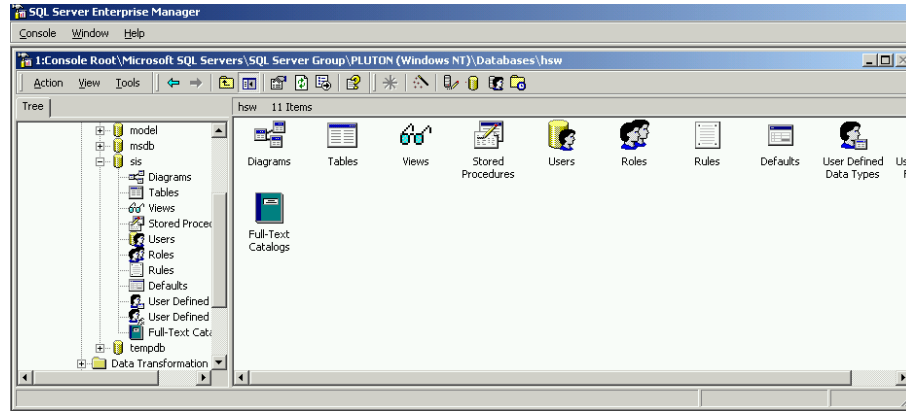


HSW ; Bilgi Teknolojileri Bölümünün bütçe kayıtlarının tutulduğu veri tabanıdır.Günlük Düzenli olarak yedeklenmekte ve Bilgi teknolojileri bölümünün kayıtları sonucu genişlemektedir.

SIS ; Persis,finsis,bilgisis gibi,yukarıda ayrıntılı tanımlanmış programlarının datalarının tutulduğu veri tabanıdır.Bu database kendi içinde butunseldir.Web ve exe arabirimlerden kullanıcılar tarafından girilen tüm veriler bu database’de toplanmaktadır.Bu database düzenli olarak yedeklenir.

SQL Server Grafik Arabirimi Üzerine Bir kaç Uygulama

Sql server üzerindeki güçlü grafik arabirim sayesinde, verilerin tutulduğu table’lar üzerinde etkin olarak çalışılabilir ve değişiklik yapılabilir.Table’lara yeni kayıt eklenebilir, ilişkiler kurulabilir,view ve diagram objeleri sayesinde database üzerinde etkin bir hakimiyet sağlanır.Bunların yanında “enterprise manager” sayesinde database yönetimi daha da kolaylaşır.Aşağıda “enterprise manager” ekranının ayrıntılı ekran görüntüsü bulunmaktadır.



YAPISAL SORGULAMA DİLİ (Structured Query Language,SQL)

İlişkisel Veritabanı Yönetim Sistemleri (Relational Database Management Systems -RDBMS) modeli ilk önce 1970 yılında Dr. E. F .Codd tarafından tarif edilmiştir. SQL veya Structred English Query Language (SEQUEL), IBM firması tarafından Codd'un modelini kullanmak için geliştirilmiştir. SEQUEL sonraları SQL olmuştur. 1979 yılında, Relational Software (şu an, Oracle Corporation), SQL'in ilk ticari uygulamasını geliştirmiştir. Bugün,SQL, ilişkisel veritabanı yönetim sistemleri standardı olarak kabul edilmektedir.

SQL, ilişkisel veritabanlarındaki bilgileri sorgulamak için kullanılan dildir. SQL, bütün kullanıcıların uygulamaların veritabanına erişmek için kullandıkları komutlar bütünüdür. Uygulama programları ve veritabanı araçları kullanıcılara çoğu durumda SQL kullanmadan veritabanına erişim imkanı sunmaktadırlar fakat bu uygulamalar da geri planda SQL kullanmaktadırlar.

SQL, ilişkisel veritabanları ile uygulamaların diyalogunu sağlamaktadır. SQL, temelde verilerle mantıksal seviyede çalışmaktadır. Yani, bir tablodan bir kaç kayıt seçebilmek için, o kayıtları seçebilecek bir şart belirtilir. Şarta uyan bütün kayıtlar bir basamakta gelir ve bunlar kullanıcıya gösterilebildiği gibi, bir başka SQL'e veya bir uygulamaya da gönderilebilir. Kayıtların tek tek nasıl geldiği ve fiziksel olarak veritabanının neresinde ve nasıl tutulduğu ile SQL ilgilenmemektedir.

SQL komutları ile,veri sorgulama,bir tabloya kayıt ekleme, değiştirme ve silme veritabanı nesnelere (database objects) yaratma, değiştirme ve silme,veritabanına ve nesnelere erişimi kontrol etme veritabanı bütünlüğünü ve tutarlılığını sağlama işlemleri yapılabilmektedir.

SQL cümlelerinde kullanılan tabloların yapısı ve içeriği aşağıda dörüldüğü gibidir.

Empno	Ename	Job	Mgr	Sal	Hiredate	Comm	Deptno
7369	SMITH	CLERK	7902	800	17.12.1980		20
7499	ALLEN	SALESMAN	7698	1600	20.02.1981	300	30
7521	WARD	SALESMAN	7698	1250	22.02.1981	500	30
7566	JONES	MANAGER	7839	2975	02.04.1981		20
7654	MARTIN	SALESMAN	7698	1250	28.09.1981	1400	30
7698	BLAKE	MANAGER	7839	2850	01.05.1981		30
7782	CLARK	MANAGER	7839	2450	09.06.1981		10

SQL'E Giriş

Onceleri SEQUEL (Structred English Query Language) olarak adlandırılan bu dilin adı, daha sonra İngilizce söylenişine uygun olarak SQL olarak değiştirildi. SQL, bilgisayar uzmanı olmayan ve yüksek düzeyli, işlemsel olmayan (non-procedural) bir dil aracılığıyla veri tabanı ile iletişim kurmak isteyen kullanıcı kitlelerine yönelik bir veri tabanı dilidir.

SQL KOMUT KUMESİ

Komut	Acıklama
SELECT	Verileri getirmek için kullanılır.
INSERT	Yeni kayıt eklemede kullanılır.
UPDATE	Var olan bir kaydı değiştirmek üzere kullanılır.
DELETE	Var olan bir kaydı silmek için kullanılır.
CREATE	Veri tabanında yeni bir nesne yararmak için kullanılır.(Table, View, Index, vb.)
ALTER	Veri tabanındaki nesne üzerinde değişiklik yapar. (yeni bir kolon eklemek vs.)
DROP	Veri tabanındaki bir nesneyi siler.
GRANT	Kullanıcılara erişim yetkilerinin verilmesinde kullanılır.
REVOKE	Verilen erişim yetkilerinin geri alınmasında kullanılır.

SQL komulan nasıl yazılır;Yazılan SQL cümlesi bir yada daha fazla satır olabilir, SQL cümlesi içerisinde yer alan cümlecikler genelde farklı satırlara yazılır,Komut kelimeleri bölünerek alt satırdan devam edilemez.SQL komutları büyük yada küçük harflarla yazılabilir,Her SQL cümlesi noktalı virgül (;) ile bitirilir. Birden fazla satır olan komutlarda en son satırın sonuna "/" işareti isteğe bağlı olarak konabilmektedir

BASİT SELECT İFADELERİ

En basit biçimiyle bir SELECT ifadesi

- İstenilen tablonun kolonlarının belirtildiği bir SELECT cümlecği,
- Hangi tablodan veri seçileceğini ifade eden bir FROM cümlecği içermelidir.

SELECT DEPTNO,ENAME,MNG FROM EMP;

DEPT NO	ENAME	MNG
10	KING	
20	JONES	7839
30	BLAKE	7839
10	CLARK	7839

SELECT * FROM EMP

Eğer tablodaki bütün kolonlar seçilek istenseydi, * ile hepsi ifade edilebilirdi

Aritmetik ifadeler :Aritmetik ifadeler içerisinde kolon isimleri, sabit sayılar ve aritmetik işleçler kullanılabilir. (+) toplama (-) çıkarma (*) çarpma (/) bölme

Aritmetik ifade içerisinde "*" ve "/" eşit öncelikli, "+" ve "-"ye göre daha önceliklidir. "+" ve "-" de eşit önceliklidirler. Öncelikleri tam ifade etmek için parantezler kullanılabilir

SELECT SAL *12, COMM FROM EMP; SAL kolonunun 12 ile çarpılmış biçimi getirilir.

Kolon isimlendirme; Kolonların başlıkları SQL cümlesinde yazıldıkları biçimi ile getirilirler. Bunu değiştirmek mümkündür.

SELECT ENAME, SAL *12 ANNSAL, COMM FROM EMP;

Kontrol :::

:::

Kolon başlığı olarak SAL *12 yerine ANNSAL kullanılacak.

Değiştirilen kolon başlıkları SQL cümlesi içerisinde artık yeni isimleri ile kullanılırlar.

Birleştirme işleci (||) : Kolonlardan gelen değerleri yan yana yapıştırmak mümkündür. " || "

İşleci ile kolonlar arasında boşluk olmadığına dikkat ediniz.

SELECT EMPNO,ENAME EMPLOYEE FROM EMP;

```
EMPLOYEE      Komutun yaptığı iş yanda görülmektedir.
7839KING
7566JONES
7698BLAKE
7782CLARK
```

Fonksiyonlar : Fonksiyonlar aldıkları kolonun yada sabit değişkenin değerini yeni bir biçime dönüştürürler. NULL değerlerin yakalanması NULL değeri olmayan anlamı taşır ve herhangi bir işleme giren bu değer mutlaka NULL üretir ve istenmeyen bir sonuç alınmasına yol açabilir. Bunu engellemek için NVL fonksiyonu kullanılır.

SELECT SAL *12-NVL(COMM,O) YILLIK FROM EMP;

Eğer NVL fonksiyonu kullanılmamış olsa idi COMM değeri olmayan bütün kayırlar için SAL *12-COMM esaplamasından NULL geri döndürülecekti. NVL fonksiyonu içerisine yazılan kolonun değeri NULL ise NULL yerine ne kullanılacağını belirler. Örnekte COMM kolonunun NULL değerleri için sıfır kullanılmaktadır.

DISTINCT ifadesi :Eğer seçilen kolon aynı satırdan birden fazla getiriyorsa, bunlardan fazla olan satırları atmak mümkündür.

SELECT DISTICT DEPTNO FROM EMP;

Bu sorgu bir bölümde birden fazla çalışan olduğu halde çalışanlar tablosundan sadece bölüm numaralarını seçmekte ve tek karlı satırları elemektedir.

ORDER BY ifadesi: Seçilen kolonların hangi sırada geleceğini belirlemek amacıyla kullanılır.

SELECT ENAME,JOB FROM EMP ORDER BY ENAME;

Örnekte çalışanlar isimlerine göre sıralı olarak getirilmektedir.

ORDER BY ENAME ~ : Küçükten büyüğe sıralar

ORDER BY ENAME ~ : Büyükten küçüğe sıralar

Birden fazla kolon sıralanmak istendiğinde kolon isimleri virgül ile ayrılarak yazılır. (ORDER BY DEPTNO.SAL DESC gibi). Sıralama sırasında NULL değerleri sıralama nasıl olursa olsun ilk sırada yer alırlar.

WHERE ifadesi : Kayıtlar arasından sadece istenilen koşulları taşıyanların seçilebilmesini sağlamak amacıyla WHERE ifadesi kullanılır. Koşulu vermek için mantıksal işlemler kullanılır.

~ Anlamı '=' eşittir '>' büyüktür '<' küçüktür '>=' büyük eşittir '<=' küçük eşittir

SELECT DNAME,DEPTNO FROM DEPT DEPTNO WHERE DEPTNO>20;

20'den büyük olanları seç

SQL işleçleri

İslec	Anlamı
BETWEEN ...AND	iki değer arasında (değerler dahil...)
IN	(liste) listedeki değerler içinden en az biri
LIKE	Benzer ise
IS NULL	Boş değer ise

SELECT ENAME,SAL FROM EMP WHERE SAL BETWEEN 1 000 AND 2000;

Değeri 1000'e eşit ve 1000'den büyük ve 2000'e eşit ve 2000'den küçük olanları seçer.

IN İşleci

SELECT EMPNO,ENAME,SAL,MGR FROM EMP WHERE MGR IN (7902, 7566, 7788);

MGR 7902 yada 7566 yada 7788 olan kayıtlar seçilir.

LIKE İşleci

Bazı durumlarda tam karşılaştırma yapmak işimize yaramayabilir. Örneğin adı 'A' harfi ile başlayanları seçmek istediğimizde. Bu durumda LIKE işleci kullanılır. Karşılaştırmada kullanılacak sabit içerisinde özel semboller kullanılır.

Sembol Anlamı

% Sıfır yada daha fazla karakterler ,Bir karakter

SELECT ENAME FROM EMP WHERE ENAME LIKE 'S%';

Ornekteki SQL cümlesi 'S' harfi ile başlayan isimleri getirir.

SELECT ENAME FROM EMP WHERE ENAME LIKE '!';

İkinci örnekteki SQL cümlesi ise dört harfli isimleri getirir.

/'%' ve '!' sembolleri bir sabit içerisinde aynı anda değişik kombinasyonlarla kullanılabilir.

IS NULL işleci :Bir kolon değerinin NULL olup olmadığını anlamının tek yolu bu işleci kullanmaktır. '=NULL' işleci kullanmak aynı şey değildir ve hiçbir sonuç vermez.

SELECT ENAME,MGR FROM EMP WHERE MGR IS NULL;

Yukarıdaki örnek yöneticisi olmayan çalışanların listesini üretir.

Negatif Test Bir koşulu sağlayan değil de sağlamayanlar seçilmek isteniyor olabilir. Bu durumda işlecin anlamının tersine çevrilmesi gerekir.

!= eşit değildir (V AX için),= eşit değildir (IBM için) <> eşit değildir.

SQL işleci Anlamı

NOT BETWEEN Verilen değerler arasında olmayan değerler

NOT IN (liste) Listedeki değerler içerisinde olmayan

NOT LIKE Benzer olmayan

IS NOT NULL Boş olmayan

Sorguda birden fazla koşul kullanma AND (ve) ve OR (ya da) işleçleri birden fazla verilmiş olan koşul ifadelerini birleştirmek amacıyla kullanılır. AND işleci arasında bulunduğu her iki koşul ifadesinin de doğru olmak zorunda olduğunu, OR işleci arasında bulunduğu her iki koşul ifadesinden sadece birisinin doğru olmasının yeterli olduğunu ifade etmek için kullanılır. Aynı SQL cümlesinde hem AND hem de OR işleçleri koşullar arasında istenilen kombinasyonda kullanılabilir. AND işleci her zaman OR işlecine göre daha önceliklidir ve daha önce işletilir. OR işlecini AND işlecine göre öncelikli hale getirebilmek için parantezlerden "()" faydalanılır.

**SELECT EMPNO,ENAME,JOB,SAL FROM EMP WHERE SAL BETWEEN 1 000 AND 2000
END JOB='CLERK';**

FONKSİYONLAR

Fonksiyonlar aldıkları değerleri, kendilerine yüklenmiş olan göreve göre yeni bir biçime dönüştürürler. Fonksiyonlar, Karakter fonksiyonları, Sayısal fonksiyonlar, Tarih fonksiyonları, Dönüştürme fonksiyonları, Herhangi bir tür veri üzerinde işlem yapabilen fonksiyonlar, Grup fonksiyonları olmak üzere 6 grupta incelenebilir.

1-Karakter Fonksiyonları : Karakter türü değişken ya da sabitleri alarak yüklenmiş olduğuna göre bunlar üzerinde işlem yapar ve karakter ya da sayısal bir değer döndürür.

LOWER(kolon/sabit) : Giriş olarak okuduğu karakter dizisini küçük harflere önüştürür. 'kolonlsabit' şeklindeki gösterim bir tek parametre girileceğini ve bu parametrenin bir kolon adı ya da bir sabit karakter dizisi olabileceğini gösterir.

```
SELECT LOWER(DNAME),LOWER('SQL') FROM DEPT;
```

UPPER(kolonlsabit) : Parametre olarak alınan karakterleri büyük harfe dönüştürür. İfadenin büyük harf veya küçük harf olması sonucu değiştirmez, her durumda büyük harfe çevirir.

INITCAP(kolonlsabit) : Parametre olarak verilen karakter dizisi içerisindeki her kelimenin baş harfini büyük harfe diğer harflerini küçük harfe dönüştürür.

LPAD(kolon/sabit,n,'karakter dizisi') : Parametre olarak verilen kolon ya da sabiti yazmadan önce başına n adet istenilen karakter dizisinden basar. Eğer karakter dizisi verilmez ise boşluk ekler.

```
SELECT LPAD(DNAME,20,*),LPAD(DNAME, 15) FROM DEPT;
```

RPAD(kolonlsabit,n,'karakter dizisi') : Parametre olarak verilen kolon ya da sabitin sonuna n adet istenilen karakter dizisinden basar. Eğer karakter dizisi verilmez ise boşluk ekler.

```
SELECT LPAD(DNAME,20,*),LPAD(DNAME, 15) FROM DEPT;
```

SUBSTR(kolonlsabit,poz,n) : Verilen kolon ya da sabit karakter dizisinin istenilen pozisyonundan başlayarak n adet karakteri getirir.

```
SELECT SUBSTR('Sevket Keser',4,3) FROM DUAL;
```

INSTR(kolonlsabit,'karakter dizisi') : Verilen kolon ya da sabit karakter dizisi içerisinde istenilen karakter ya da karakter dizisinin ilk geçtiği konumu döndürür.

L TRIM(kolonlsabit,'karakter dizisi') : İstenilen karakteri ya da istenilen karakterleri verilen kolon ya da sabit karakter dizisinin başından siler. Dizinin başında yer alan silinecek karakter ard arda yer aldığı sürece silinir. Eğer hiçbir karakter verilmez ise dizinin başındaki boşluklar silinir.

RTRIM(kolonlsabit,'karakter dizisi') : L TRIM ile aynı işlemi yapar; tek farkı istenilen karakteri (ya da karakterler) dizinin sonundan siler.

LENGTH(kolonlsabit) : Verilen karakter dizisinin toplam kaç karakter uzunlukta olduğunu döndürür.

TRANSLATE(kolonlsabit,eski,yeni) : Verilen karakter dizisi içerisinde 'eski' parametresi olarak girilecek karakterleri bularak 'yeni' olarak girilecek olan karakterler ile yer değiştirir.

2-Sayısal Fonksiyonlar Sayısal fonksiyonlar sayısal değerleri parametre olarak alır ve yine sayısal değerler üretirler.

ROUND(kolonlsabit,n) : Verilen değeri virgülden sonra n basamağını dikkate alarak yuvarlar. Eğer n değeri negatif girilirse virgülün soluna kalan n adet rakam yuvarlanır.

```
ROUND(45.923,1) --> 45,9 /.. / ROUND(45.923) --> 46 /../ ROUND(45.923,1) --> 45,3
```

TRUNC(kolonlsabit,n) : Virgülden sonraki n basamak sıfırlanır. Eğer n değeri negatif girilirse virgülin solunda kalan n adet rakam sıfırlanır.

CEIL(kolonlsabit) : Verilen değerden büyük en yakın tamsayıyı döndürür.

FLOOR(kolonlsabit) : Verilen değerden küçük en yakın tamsayıyı döndürür.

POWER(kolonlsabit,n) : Verilen değer n'inci kuvvetini alır.

SQRT(kolonlsabit) : Verilen değer kare kökünü bulur.

SIGN(kolonlsabit) : Eğer verilen değer 0 ise 0, negatif ise -1 , pozitif ise +1 değeri döndürür.

ABS(kolonlsabit) : Verilen değer mutlak değerini döndürür.

MOD(d1 ,d2) : d1 'in d2'ye bölümü sonucu oluşan kalan değerini döndürür .

Tarih Fonksiyonları Oracle DATE türünde tanımlanmış alanlar içerisinde saniye düzeyine kadar tarih bilgisini saklayabilir. Fakat giriş ve çıkış sırasında tarih formatı ayarlanarak bu detayda bilgi girişine gerek duyulmadan tarih bilgisi üzerinde istenilen işlemler yapılabilir.

Tarih : Yüzyıl, Yıl, Ay, Gün, Saat, Dakika, Saniye,

Oracle veri tabanının başlangıç olarak kullandığı tarih formatı (değiştirilmediği sürece) DD-MON-YY şeklindedir. Eğer veri tabanı türkçe modunda çalıştırılıyorsa ayların kısaltmaları türkçe olarak yapılacaktır (MAYIS : MAY, HAZİRAN : HAZ gibi).

Veri tabanından sistem tarihini de okumak aşağıdaki SQL sorgusu ile mümkündür.

SELECT SYSDATE FROM SYSTEM.DUAL;

Bu sorgu DUAL isimli sahte bir tablo kullanarak o andaki sistem tarihini okumamızı sağlamaktadır.

Tarih üzerinde aritmetik işlemler

tarih+sayı : Tarihe istenilen gün sayısı eklenir.

tarih-sayı : Tarihten istenilen gün sayısı çıkartılır.

tarih1-tarih2 : İki tarih arasındaki gün sayısını bulunur

tarih+sayı24 : İstenilen saat sayısını tarihe eklenir.

MONTHS_BETWEEN(tarih1,tarih2) : İki tarih arasında kaç ay olduğunu hesaplar. Eğer tarih2 tarih1 'den daha büyükse sonuç negatif olarak üretilir.

ADD_MONTHS(tarih,n) : İstenilen tarihe n ay ekler.

NEXT_DAY(tarih,gün) : Verilen tarihten bir sonraki haftanın istenilen gününün tarihini döndürür. Girilen gün bir numara yada günün adı olabilir. 'FRIDAY' ve 6 aynı günü ifade eder. Günler pazar gününden itibaren sayılmaya başlanır. Örneğin pazartesi günü için 2 girilmelidir.

LAST_DAY(tarih) : Girilen tarihin ayının son gününün tarihini döndürür.

ROUND(tarih) : Girilen tarihin saat kısmını yuvarlar. Bu genelde saat bilgisi içeren tarihlerin arşılaştırılmasında faydalı olur.

ROUND(tarih,'MONTH') : Girilen tarihi ay bilgisine kadar yuvarlar. Ayın 15'şinden önceki günler ayın ilk gününün tarihini, sonraki günler için sonraki ayın ilk gününün tarihini getirir.

ROUND(tarih,'VEAR') : Girilen tarihi yıl bazında yuvarlar. Girilen tarih yılın ilk yarısını gösteriyor ise o yılın ilk gününün tarihi geri dönderilir, değilse bir sonraki yılın ilk gününün tarihi döndürülür.

TRUNC(tarih,'MONTH') : Verilen tarihin ayının ilk gününün tarihini bulur.

TRUNC(tarih,'VEAR') : Verilen tarihin yılının ilk gününün tarihini bulur.

Dönüştürme Fonksiyonları

TO_CHAR(sayıtarih,['format]) : Verilen rakam ya da tarihi istenilen formatta karaktere dönüştürür.

TO_NUMBER('karakterler') : Karakter türünde verilmiş olan rakamları sayısal türe dönüştürür.

TO_DATE('karakter' , 'format') : Formatı belirli karakter halindeki bir tarihi tarih türüne dönüştürür.

Tarih Formatları (TO_DATE ile kullanılabilen)

Ornek : TO-DATE('27-OCT -95','DD-MON-RR')

Değişik türlerde parametre kabul edebilen fonksiyonlar

DECODE Bu komut ile kolon isimlerini koşullara bağlayarak verilerin durumuna göre değişik kolon isimlerini seçmek mümkündür. Yapısal dillerdeki 'if-then-else' yapısının ilkel bir örneğidir.

DECODE(kolon/ifade,ara1,sonuc1,[ara2,sonuc2, ...,]default)

SELECT ENAME,JOB,

DECODE(JOB,'CLERK','WORKER','MANAGER','BOSS','UNDIFIED')

DECODE_JOB FROM EMP ,14 satırları seçildi.

Grup Fonksiyonları;Grup fonksiyonları veri tabanından seçilen bir dizi satır üzerinde işlem yapar ve sonuç olarak kendisine yüklenmiş olan göreve göre özet bir bilgi üretir.

AVG([DISTINCTIAL] kolon) : Verilen kolon değerlerinin ortalamasını bulur.

COUNT([DISTINCTIAL] kolon*): Verilen kolonun NULL olmayanlarının sayısını getirir. Eğer * kullanılırsa toplam kaç satır sorgulandığı bulunur.

MAX([DISTINCTIAL] kolon) : Maximum değeri getirir.

MIN([DISTINCTIAL] kolon) : Minimum değeri getirir.

SUM([DISTINCTIAL] kolon) : Verilen kolon toplamını bulur.

SELECT COUNT(*) FROM EMP WHERE DEPTNO = 20;

20 numaralı bölümde kaç kişi olduğunu bulur.

GROUP BY ifadesi GROUP BY ifadesi ile sorgulanan satırlar belirli guruplara ayrılarak bu guruplar üzerinde grup fonksiyonları kullanılır.

SELECT JOB,AVG(SAL) FROM EMP GROUP BY JOB;

J

JOB	AVG(SAL)	Her iş için ayrı ayrı ne kadar ortalama maaş verildiğini hesaplar.
ANALYST	3000	SELECT DEPTNO,JOB,AVG(SAL) FROM EMP GROUP BY DEPTNO,JOB;
CLERK	1037.5	
MANAGER	2758.3333	
PRESIDENT	5000	
SALESMAN	1400	

DEPT NO	JOB	AVG(SAL)	
10	CLERK	1300	9 satırları seçildi.
10	MANAGER	2450	Her bölümdeki farklı işler için ne kadar ortalama maaş verildiğini hesaplar. Grup fonksiyonları ile birlikte kullanılan kolon isimleri mutlaka GROUP BY ifadesi içinde yer almalıdır. Tersisi durumda hatalı bir SQL cümlesi yazılmış olur
10	PRESIDENT	5000	
20	ANALYST	3000	
20	CLERK	950	
20	MANAGER	2975	
30	CLERK	950	
30	MANAGER	2850	
30	SALESMAN	1400	

Kontrol :::

:::

VERİ TABANI ANALİZİ

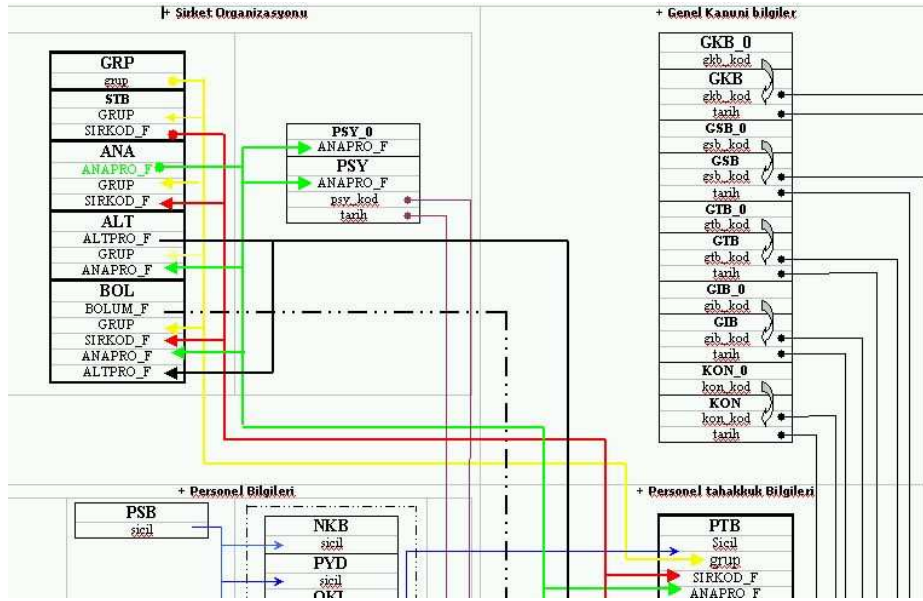
Staj döneminde yukarıda da ayrıntılı olarak bahsedilen, **persis** isimli personel takip ve otomasyon programının analizi gerçekleştirilmiştir. Analiz iki boyutta yapılmıştır. Birinci boyut veri tabanı analizi ikinci boyutta exe arayüzünün analizidir. Aşağıdaki sayfalarda veri tabanı analiz ve exe arayüzünün analizi ayrıntılı olarak anlatılmaktadır.

(A).. Database Analizi

Database analizinde, yapılan işlemler aşağıda bahsedilmiştir.

- Programla ilgili table'lar belirlenmiştir.
- Table'ların field alanları ayrıntılı olarak belirlenmiştir.
- Birbirleriyle ilişkili olan tablolar ve field alanları belirlenmiştir.
- Programın database yapısı diğer programlarla ilişkili olarak belirlenmiştir.
- Belirlenen database yapısı doğrultusunda haritalamaya gidilmiştir. gidilen bu haritalama sonucunda da proje üst bakış sağlanmıştır.

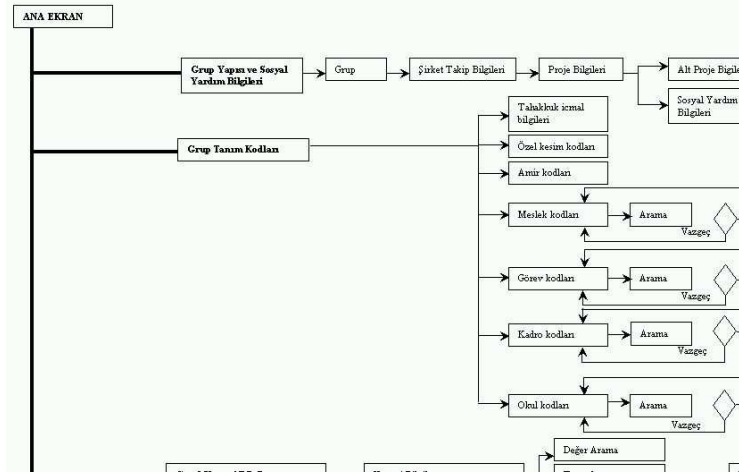
Proje database analizi hakkında yapılan çalışmalar ve database haritası daha sonra bir rapor çatısı altında birleştirilerek sonuca ulaşılmıştır. Aşağıda çıkartılan database haritasıyla ilgili resmi görmekteyiz.



Şekil Veri tabanı yapı analizi / ilişki şeması

(B).. EXE Arayüzü Analizi

Persis Personel takip ve otomasyon programı kullanıcı arabirimi yaklaşık 7.5 MB büyüklüğündedir.Kullanıcılar programın işleyişi doğrultusunda, kayıtlarını bu arayüzden girerler,girilen bu kayıtlar, veri tabanına kaydedilir. Daha sonra kayıtlar daha sonra işlenmek üzere saklanır.Programın exe arayüzünün staj dönemi boyunca ayrıntılı teknik analizi gerçekleştirilmiş ve raporlaması yapılmıştır.Analiz sonucunda elde edilen veriler iki çalışmada sonuçlandırılmıştır.Birinci çalışma, programın akış diyagramıdır.Exe arayüzünün akışdiyagramı çıkartılmış,böylece proje bir üst bakış sağlanmıştır.Aşağıdaki resimdeprogramın blok diyagramı ve hazırlanan raporla ilgili ekran görüntülerini görmekteyiz.

**Şekil : Program blok diyagramı**

Grup Yapısı

1. İlk önce buradan faaliyet alanını seçin.
2. Faaliyet alanıyla ilgili şirket seçilir.
3. Seçilen şirketin ulaşılmak istenen projesi seçilir.
4. Seçilen projenin alt projesi seçilir.
5. Bir üstde seçilen alt projenin ulaşılmak istenen bölümü seçilir.

Sonuç:Yapılan bu çalışmalar sonunda proje sorumlularıyla yapılan değerlendirme toplantılarıyla projenin eksiklikleri ve hataları tespit edilmiş ve bir üst bakış sağlanmıştır.Çalışmayan bölümler düzeltilmiş,programın sağlamlığı ve hızlılığı konusunda yeni adımlar atılmıştır.Projenin genişliğinden dolayı ortaya çıkan dağıtık yapı bir nebze de olsa toparlanmıştır.Ayrıca exe arayüzü için hazırlanan kullanma klavuzundan programı kullananlara dağıtılmıştır.

Kontrol :::

:::

PHP

PHP (eski adıyla Personal Home Page, şimdiki adıyla Hypertext Preprocessor), sunucu tarafında çalışan (server-side) bir betik (script) dilidir, bu nedenle yazılan programlar birer düz yazı dosyasıdır. Web sunucusunun ayarlarına bağlı olarak, yazılan PHP programlarını kaydederken dosya uzantılarını *.php* yada *.phtml* olarak belirtmeliyiz.

PHP dosyalarının bulunduğu makinedeki web sunucusu programı (Örneğin Apache), ziyaretçilerden gelen istekler doğrultusunda hazırladığımız programları, içerdiği PHP modülü sayesinde çalıştırır ve elde ettiği çıktıları saf HTML (ve varsa Javascript, VBScript) kodu olarak ziyaretçiye iletir.

PHP'nin yapısı ve gerekliliği ;Bilişimci gazeteci yazar Dr. Hakkı Öcal, yazılarından birinde HTML'le ilk tanıştığı zamanlarda farkettiği önemli bir sorundan bahsetmektedir. Bu sorun HTML ile nasıl program yapılacağı idi. Daha doğrusu bu, HTML'in en büyük sorunu idi. HTML ile program yapmak, hazırladığımız siteyi bu program çerçevesinde sunmak mümkün değildir. Bu açığı bir nebze olsun kapatmak amacıyla tarayıcı (browser) üreticileri bazı diller geliştirdiler: Javascript ve VBScript. Ama bunlar da yeterli değil çünkü Javascript yada VBScript programları ziyaretçi tarafında (client-side) çalışırlar ve boyutları ne kadar büyürse ziyaretçiyi de o kadar zor durumda bırakırlar. Üstelik bu saydığımız dillerden VBScript sadece Microsoft IE tarafından yorumlanabiliyor, Javascript ise her ne kadar tüm tarayıcılar tarafından yorumlanabiliyorsa da ticari rekabet yüzünden hepsiyle %100 uyumlu değil. İşte bu noktada bize tarayıcılardan bağımsız, ziyaretçiyi yormayacak (yani sunucuda çalışacak) ve bizi site yada web tabanlı program hazırlarken büyük dertlerden kurtaracak (burada bir özyinelemeli (recursive) döngü var: zaten bu çözüm ortaya çıkmasaydı web tabanlı program diye bir şey olmayacaktı) bir çözüm gerekiyor. Bu çözümler birden fazla. Açık kod olarak geliştirilen PHP, Microsoft ürünü ASP (Active Server Pages) ve Sun ürünü JSP (Java Server Pages).

PHP komutları aşağıdaki şekillerde yazılabilir.

```
<?php ... ?>

<? ... ?>

<script language="php"> ... </script>
```

PHP, diğer betik diller gibi çok esnek bir dildir. Oluşturacağımız *.php* uzantılı dosyalarda tamamen PHP komutları bulunmasına gerek yoktur, sayfamızı HTML etiketleri (tag) ile hazırlayıp sadece gerektiği yerlerde yukarıdaki yazım kurallarından biriyle PHP kodumuzu yazabiliriz.

Örneğin;

```
<html>
<head>
  <title>PHP'ye hızlı giriş..</title>
</head>
<body>
  <? print "<h1>Merhaba Dünya</h1>" ; ?>
</body>
</html>
```

Not ; PHP, temelde C notasyonu kullanacak şekilde hazırlandığından her komutun sonuna (:) konmalıdır

Kontrol :::

:::

PHP KOMUTLARI**1- “echo” ve “print”****Genel Kullanımı ;**

```
Echo "ekrana yazdırılacak ifade"  
Print (ekrana yazdırılacak ifade)
```

Echo, tırnaklar arasında verilen ifadeyi ekrana basmayı sağlar.Print, parantez içerisinde verilen ifadeyi ekrana basmayı sağlar.

2- “if”

Karşılaştırma işlemlerinde kullanılır. Eğer şart'dan sonra tek bir işlem yapılacaksa { } bloğunu kullanmaya gerek yoktur.

Genel Kullanımı:

```
if(şart)  
{  
    doğruysa yapılacak işlemler  
}  
else  
{  
    yanlışsa yapılacak işlemler  
}
```

3-“for” döngüsü;**Genel kullanımı**

```
For(döngü değişkeni=değer;şart;artış miktarı)
```

4-“While” döngüsü

```
while(şart)  
{  
    işlemler  
}
```

5- “do while” dongüsü

```
do  
{  
    işlemler  
}while(şart)
```

6-Gettype

Değişkenin hangi tipde olduğunu öğrenmek amacıyla kullanılır.

```
gettype(değişken)
```

7-Settype:

Bir değişkenin tipini değiştirmek için kullanılır.

Kontrol :::**:::**

PHP KOMUTLAR II

8- “strval”, “doubleval”, “intval”:

Bu komutlar değişkeni o an için belirtilen tipe dönüştürerek ekrana basmayı sağlar.

9-“define”

Sabit tanımlamak için kullanılır.

```
Define("sabit", "atanacak değer")
```

10- “switch”

```
switch(ifade)
{
    case durum1:islem1; break;
    case durum2:islem2; break;
    .....
    .....
    default:islem;
}
```

FONKSİYONLAR

Bir bilgisayar programının kalbi, fonksiyonlardır. Fonksiyonlar, bir kere tanımlanır ve bir çok kereler, ihtiyaç olduğu noktada hizmete çağrılır. Fonksiyonlar genellikle bir iş veya hesaplama yaparlar ve bunun sonucu olarak ortaya yeni bir değer çıkartırlar. Bu değere "fonksiyondan dönen değer" denir.

PHP'nin kullanılmaya hazır bir çok fonksiyonu vardır. Şu ana kadar sürekli kullandığımız `echo()` ve `print()` bu tür PHP fonksiyonlarıdır. PHP'nin kendi fonksiyonlarını göreve çağırarak, yine yukarıdaki örneklerde olduğu gibi, program içinde gerektiği yere adını ve kullanacağı değeri yazmakla olur:

```
print ("merhaba!");
```

PHP bizim kendi fonksiyonlarımızı kullanmamıza da izin verir. Fakat bunun için önce fonksiyonu tanımlamamız gerekir.

Fonksiyon Tanımlama ve Çağırma

PHP'de fonksiyonlar *function* komutu ile oluşturulur. Tanımladığımız fonksiyon, kendisini göreve çağırarak komuttan, yapacağı işlemde kullanmak üzere değer alacaksa, bu değerlere vereceğimiz değişken isimleri fonksiyon adının yanında parantez içinde gösterilir. Fonksiyon birden fazla değer bekleyecekse, bunların değişken adlarının arasına virgül konur. Fonksiyona ulaştırılan değerlere argüman denir. Fonksiyon, kendisine bir değer ulaştırılmasını beklemese bile içi boş parantez koymamız gerekir. Buna göre PHP'de fonksiyon şöyle yazılır:

```
function fonksiyonun_adi (argüman1, argüman2, ... argümanN)
{
    fonksiyonun yapacağı işe ilişkin komutlar
}
```

Fonksiyon adları, değişken adları ile aynı kurallara tabidir. Fonksiyon içinden fonksiyon çağrılabilir; fonksiyonların içinde PHP'nin kendi fonksiyonları kullanılabilir

MYSQL

SQL yani Structural Query Language (Yapısal Sorgulama Dili) hemen her türlü veritabanını kullanmak için hazırlanmış bağımsız bir sorgulama dilidir. Bu dili anlayabilen veritabanlarından biri de bu dökümanda üzerinde durduğumuz **MySQL**'dir. **MySQL** açık kod lisansı ile dağıtılan bir veritabanı sunucusudur. Sunucu olarak çalıştığı için kendisi ile bağlantı kurabilen ve anlaşılabilen her platform üzerinden (biz PHP ile ilişkisini anlatacağız) bu veritabanına erişip verilerimizi kendi platformumuza aktarabiliriz.

SQL dili çok gelişmiştir. En doğal işlevlerinden olan "tablo oluştur", "tabloya veri ekle", "tablodan veri çek", "tabloyu güncelle"nin yanında çok daha karmaşık tablolar arası ilişkilendirmeleri içeren işlemleri yapılabilir. Bazı durumlarda bilhassa matematiksel verilerin işlenmesinde hemen hemen bütün yükü SQL ifadesinde halledebiliriz.

SQL, (Structured Query Language, Yapısal Sorgu Dili) veritabanı endüstrisinin standart dilidir. Fakat HTML konusunda farklı firmaların sadece kendi Browserlarının tanıdığı türden etiketler geliştirmeleri gibi, SQL alanında da farklı firmaların ekleri ve dilde yaptıkları değişiklikler vardır. Bir veritabanında farklı sayıda tablolar olabilir. Veritabanı dosyasında yer alan tablolarda, sütunlar ve satırlar vardır: bunların keşistikleri yerlere biz verilerimizi yazarız. Bir SQL veritabanı dosyasında her bir kutunun ne tür veri kabul edeceği önceden belirlenir; dolayısıyla türü sayı olan verinin yazılacağı kutuya, alfanümerik değer yazılmaz. MySQL, sayısal verilerin türünü **INT**, belirli bir sınıra kadar değişen sayıda karakterden oluşacak verilerin türünü **VARCHAR** komutuyla belirler.

mysql veri türleri

MySQL'de bir çok veri türü oluşturulabilir. Ancak Web programları açısından önemli olan bir kaçını özellikleri şöyle sıralanabilir ;

INT	Tamsayı: -2147483648'den 2147483647 kadar değişen diziye " <u>signed</u> " (işaretli), :0'dan 4294967295'e kadar değişenine " <u>unsigned</u> " (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.

MYSQL KOMUTLARI -I

MySQL'de bir tablo oluşturmak için gerekli CREATE TABLE komutu şöyle kullanılır:

```
CREATE TABLE uyeler (adi VARCHAR(30), soyadi VARCHAR(30), uye_no INT);
```

Bu komutla, "uyeler" isimli üç sütunlu bir tablo oluşturulur: birinci ve ikinci sütunlarda en fazla 30, karakterlik değişen boyutta alfanümerik değerler yer alırken, üçüncü sütunda sadece tam sayı olan değerler bulunabilir. Bu komutla oluşturulan tabloya INSERT INTO komutuyla veri girebilirsiniz:

```
INSERT INTO uyeler (adi, soyadi, uye_no) VALUES ('Muharrem','Taç','1234');
```

Bir tablonun oluşturulması ile içine veri yerleştirilmesi komutları ayrı ayrı zamanlarda, ayrı işlemler olarak yapılabileceği gibi, toplu bir metin halinde, otomatik olarak da yapılabilir.

MySQL veritabanından bilgi edinmek için SELECT komutunu kullanırız:

```
SELECT * FROM uyeler ;
```

Bu, MySQL'e, uyeler adlı tablodaki bütün değerlerin okunmasını bildirir. Buradaki "*" işareti, "bütün sütunlardaki bütün değerler" anlamına gelir. Diyelim ki yukarıda oluşturduğumuz tablonun sadece "adi" ve "soyadi" sütunlarındaki bilgileri almak isteseydik, bu komutu şöyle yazacaktık: **SELECT adi soyadi FROM uyeler ;**

Bir veritabanındaki bilgilerin yenileriyle değiştirilmesini, yani veritabanı dosyasının güncelleştirilmesini UPDATE komutu sağlar. Bu komutu kullanarak veritabanımızdaki bazı

Kontrol :::

:::

Yapılan iş (A)..Web Teknolojileri

(23)

kutucukların içindeki bilgileri değiştirebiliriz. Veritabanı dosyalarını güncelleştirme zorunluğu bulunması ise bize veritabanı tasarımının çok önemli olduğunu gösterir. Örneğin:

```
UPDATE uyeler SET adi = "tuba" ;
```

Bu komut, veritabanındaki bütün satırlarda, birinci sütundaki değerleri "tuba" olarak değiştirmekle sonuçlanırdı. Amacımız bu ise, sorun değil; ancak çoğu kez MySQL'e hangi satırda (veritabanı tekniğindeki terimle söylersek, hangi kayıtlarda) değişiklik yapılacağını daha ayrıntılı söylememiz gerekir. Veritabanı dosyamızı oluştururken, her kaydın diğer kayıtlarda olmayan (*unique*) bir sütun (bunu da veritabanı tekniğindeki terimle söylersek, alan) bulunmalıdır, ki MySQL'e yapılacak değişikliğin tam yerini söyleyelim. Örneğin

```
UPDATE uyeler SET adi = "tuba" WHERE uye_no = 1234;
```

MySQL bu komutu alınca sadece üye numarası 1234 olan kişinin (yani uye_no alanındaki değer 1234 olan kaydın) "adi" alanındaki değeri silecek ve yerine verdiğimiz yeni değeri yazacaktır. Böyle birincil alanı bulunan, iyi düşünülmüş bir veritabanından seçim yapmak da kolay olur. Örneğin:

```
SELECT adi soyadi FROM uyeler WHERE uye_no >= 123;
```

deyimi ile tablomuzda bulunan kayıtlardan sadece üye numarası 123'den büyük olanları seçebiliriz.

Bir MySQL veritabanındaki kaydı silmek için DELETE komutunu kullanırız:

```
DELETE FROM uyeler WHERE uye_no = 1234;
```

Veritabanında sadece bir kayıta üye numarası 1234 olacağı için bu komutla sadece bir satır silinecektir.

Bu komutu, diyelim ki üyelik kaydını yenilememiş kişilerin tümünü silmek için de kullanabiliriz.

Veritabanımızda üyelik kaydının yenilenme tarihini gösteren bir alan bulunduğunu varsayalım:

```
DELETE FROM uyeler WHERE yenileme_tarihi < 2001-08-31;
```

Bu komutla, üyeliğini yenileme tarihi 31 Temmuz 2001'den eski olan bütün üyelerimizin kaydını veritabanından silmiş oluruz.

MySQL'in sorgulanmasından geriye dönen sonuç tablosundaki veriler de `mysql_result()` komutuyla alınır.

```
$isimler = mysql_query(select * from ogrenciler");
if (mysql_error()) {echo ("MySQL hatası oluştu. Hata no :"); echo
mysql_error();}
$ad = mysql_result($isimler,$i,"ad");
$soyad = mysql_result($isimler,$i,"soyad");
```

Veritabanından dönen tablodaki ilk satırın satır numarası 0'dır. Her sonucun bir satır olduğu bu tabloda tablo alanının ismiyle satır numarası verilerek sonuçlar çağırılabilir. Öğrenci numarası, adı ve soyadının bulunduğu "ogrenciler" isimli tabloya yapılacak olan genel sorguda geriye dönen sonuç tablosunun her satırında bir öğrenci numarası, ad ve soyad bulunacaktır. Bu sonuç tablosundan sonuçların birer birer alınarak kullanılması için genellikle while döngüsü kullanılır. Sonuç tablosundaki satır sayısını öğrenmek için `mysql_numrows()` komutu kullanılır.

```
$isimler = mysql_query(select * from ogrenciler");
if (mysql_error()) {echo ("MySQL hatası oluştu. Hata no :"); echo
mysql_error();}
$sayi = mysql_numrows($isimler); $i = 0;
while ( $i < $sayi ) :
    $ad = mysql_result($isimler,$i,"ad");
    $soyad = mysql_result($isimler,$i,"soyad");
    echo ("Ad : $ad Soyad : $soyad");
    $i++; endwhile;
```

Sonuç tablosunun Php'de kullanılmasını sağlayan `mysql_query()` komutuyla tüm tablodaki değerler bir sayı kullanılarak tanımlanır. Sonuçlarsa `mysql_result()` kullanılarak metin değişkenine atanırlar. Eğer sonuçların bir dizi değişkeninde (array) toplanması isteniyorsa `mysql_fetch_array()` komutu kullanılır.

Kontrol :::

:::

Yapılan iş (A)..Web Teknolojileri

(24)

PHP ile Veritabanı

PHP ile veritabanına dayanan sayfalar yapmak için mutlaka ve sadece MySQL sürücülerini kullanmak gerekmez. Bunun yerine Microsoft'un ODBC sürücülerini kullanabilirsiniz. Ancak Web sitenize ev sahipliği yapan firmanın size ODBC desteği verip vermediğini öğrenmeniz gerekir. Aslında aynı şekilde, ev sahibi firmanın MySQL desteği verip vermediğini de öğrenmek zorundasınız. Yukarıda, bir PHP programı ziyaretçiden Form ile alacağı bilgileri, yine kendisine gönderebileceği ve bu bilgileri işleyebileceği öncelikle bilinmelidir..

PHP programımızın bu bilgileri bir veritabanına işlediğini ve ziyaretçi bir başka sayfaya gitmek istediğinde o sayfanın içeriğinin, ziyaretçinin birkaç saniye önce yaptığı tercihlere göre belirlendiğini düşününüldüğünde veri tabanı kullanımı ön plana çıkar.. Bir başka örnek, Web sitemizdeki sayfalarda yer alan unsurların güncelleştirilmesi için her güncelleştirmede çok sayıda programı yeniden yazmak veya sayfayı yeniden yapmak zorunluğu varsa, bu program ve sayfaların içeriklerini bir veritabanından alıyor olması halinde, sadece veritabanını güncelleştirmekle onlarca, belki de yüzlerce, HTML ve PHP belgesini yeniden yazmış gibi olabiliriz.

PHP İLE MYSQL DATABASE BAĞLANTISI ÖRNEĞİ

Php ile mysql bağlantısı komutlarına örnek olması ve php nin önemli bir özelliği olan html kodları arasına gömülü olarak yazılmasına bir örnek olması açısından aşağıdaki kodlar önemlidir.

```
<html>
<?php          /* bildiğiniz gibi html kodundan php koduna geçiş
                için kullanılır */
$tarikh=date(ymd); /* yıl ay gun formatında ki server zamanını tarih
                isimli değişkene atanmakta */
$saat=date(His); /* saat dakika saniye formatında ki server
                zamanını saat isimli değişkene atamakta */
/* localdeki mysql serverın rootuna şifresiz halde bağlanılır, hata
varsa hata mesajı çıkar */

MYSQL_CONNECT("localhost","root","") OR DIE ("Veritabanına
bağlanamadı");

/* databasel seçilir */

@mysql_select_db("databasel") or die("Veritabanı bulunamadı");

/* hata mesajı verdirmenin başka bir yolu da if komutuyladır. If komutu
yanında parantez içindeki olay hatasız olursa kayıt tamam yazılır ,
hata çıkarsa else'ten sonraki {} parantezleri içindeki işlemler
gerçekleşir. */

if (MYSQL_QUERY("INSERT INTO table1 (ad, mail, mesaj, tarih, saat)
VALUES ('$ad', '$mail', '$mesaj', $tarikh, $saat) "))
{ echo "kayıt tamam.."; } else { echo "kayıt yapılamadı."; }
?>
/* php kodundan html koduna geçiş için. */

<br>Tekrar forma dönmek için <a href="form.htm">tıklayınız..</a>
</html>
```

Kontrol :::

Yapılan iş (A)..Linux İşletim Sistemi Yapı Analizi

(25)

LINUX ÇEKİRDEĞİ

Linux çekirdeği , işletim sisteminde yazılım ve donanım arasında aracı olarak çalışan ve genel organizasyonu sağlayan temel programdır.

Çekirdeğin temel görevleri;

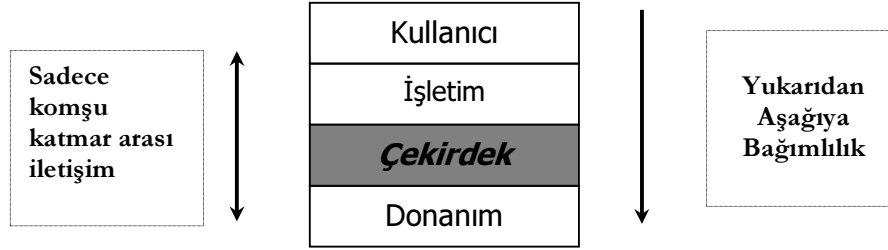
- Süreç kontrolü
- Hafıza yönetimi
- Dosya sistemi organizasyonu
- Giriş/Çıkış kontrollerini gerçekleştirir

Bunun dışında linux çekirdeği

- Donanım tarafını kullanıcıdan saklamak,
- Kolay programlanabilir bir arayüz sunmak. gibi işlevleri de gerçekleştirir.

Linux çekirdeği ;

- Tek başına işlevsizdir
- İşlevsel bir yapı için yardımcı katmanlar sağlar



Çekirdek yapısında ; iletişimin,sadece komşu eşit katmanlar arası olması ve bağımlılığın yukarıdan aşağıya doğru olması sağlamlığı getirir.

ÇEKİRDEK MİMARİLERİ**(A)..Monolitik Çekirdek Yapısı**

Linux işletim hali hazırda bu çekirdek yapısını kullanmaktadır. Tek başına, büyük bir programdır, yaklaşık 70 MB..

Avantajları :

- Konvansiyonel teknikler ile görevler(*ing. task*) arası iletişim. (doğrudan hafıza erişimi, semaphore, mutex, vb.)
- Çekirdek süreçleri ve görevleri arasında hızlı geçiş.

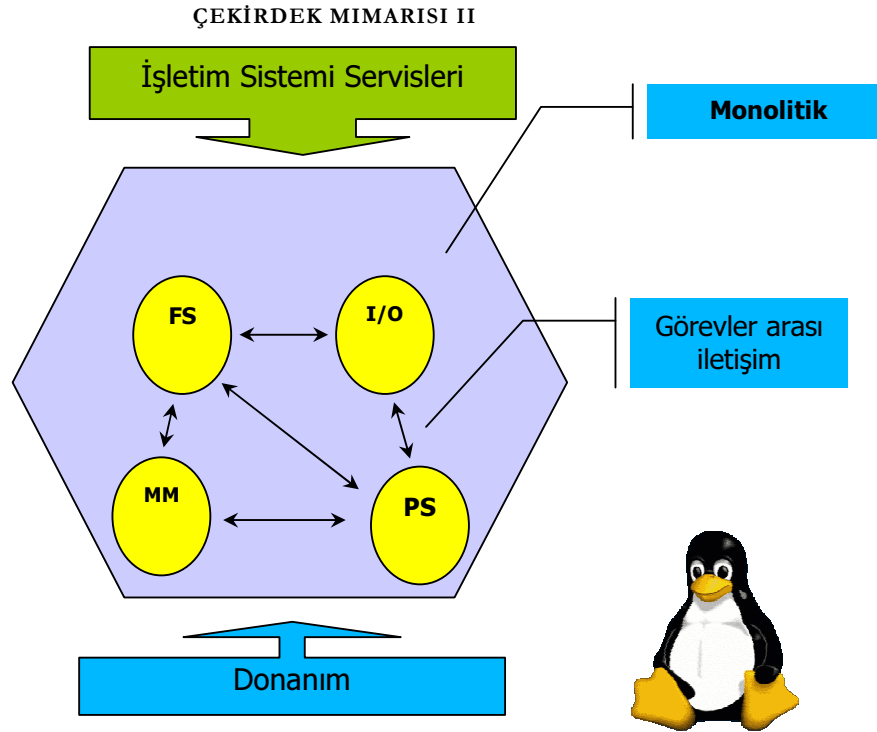
Dezavantajları :

- Proje büyüdükçe karmaşıklaşan geliştirme yükü

Kontrol :::

Yapılan iş (A)..Linux İşletim Sistemi Yapı Analizi

(26)

**Çekirdek Mimarisi Alternatifleri;****(B)..Mikro çekirdek yapısı**

Mikro çekirdek yapısı, işlevsel parçalar, mesaj geçirecek birbirleri ile haberleşen modüller üzerine kurulmuştur.

Avantajları :

- Mimariler arasında yüksek taşınabilirlik
- Bağımsız gruplar tarafından geliştirilebilme imkanı
- Ölçeklenebilirlik
- Az ve basit görevli küçük çekirdek

Dezavantajları

- Mesaj geçirme sisteminin getirdiği yüksek yük
- Öngörülemeyen modüler uyumsuzluk

Linux çekirdeği monolitik çekirdek mimarisi kullanmaktadır ve bu değişebilmesi muhtemel bir özellik değildir.

(C)..Linux Modeli

Bu çekirdek mimarisinde Monolitik çekirdeğin özelleştirilebilme problemine yaratıcı çözüm getirilmeye çalışılmıştır. Bu mimarinin özellikleri ise “Dinamik Yüklenebilir Çekirdek Modülleri” Spesifik donanım için sadece modülü derleyerek sisteme katabilme olanağı, ilgili aygıtlar ve özellikler kullanılacağı zaman dinamik olarak yüklenebilme olanağı ve kısıtlı sistem kaynakları altında dahi düşük çekirdek yükü olarak sıralanabilir.

Çekirdek tasarım Kriterleri

- Daha fazla hız
- Eski hatalardan arındırılmışlık
- Yeni donanımlara destek
- Güvenilirlik, kararlılık

Kontrol :::

:::

Yapılan iş (A)..Linux İşletim Sistemi Yapı Analizi

(27)

ÇEKİRDEK DERLEME

Aşağıdaki ekran görüntüsünde bilgislem merkezindeki linux server'la gerçekleştirilmiş telnet bağlantısını görmekteyiz bağlantı sırasında görülen komutlar verilerek,işletim sistemini çekirde sürümü,kullanıcı kimliği gibi bilgilerle erişilmiştir.

```

C:\WINNT\System32\telnet.exe
Red Hat Linux release 7.2 (Enigma)
Kernel 2.4.7-10 on an i686
login: skeser
Password:
Last login: Fri Aug  9 17:12:58 from 10.1.1.88
[skeser@wood skeser]# who i am
wood.tekfen.com.tr!skeser pts/0  Aug  9 17:14 (10.1.1.88)
[skeser@wood skeser]# uname -r
2.4.7-10
[skeser@wood skeser]# uname -a
Linux wood.tekfen.com.tr 2.4.7-10 #1 Thu Sep 6 17:27:27 EDT 2001 i686 unknown
[skeser@wood skeser]# _
  
```

- Telnet bağlantısı kabul edildiğinde gelen default satır.
- Login neme ve password yazılır.
- Kullanıcı kimliği "who i am" komutuyla sisteme doğrulattılır.
- **uname -r** komutuyla çekirdek sürümü öğrenilir.
- **uname -a** komutuyla ayrıntılı çekirdek sürümü öğrenilir.

2.4.7

- Birincil çekirdek sürüm numarası
- Değişmesi için Linux mimarisinin neredeyse tamamının değişmesi gerekir
- İkincil çekirdek sürüm numarası
- Tek sayılar "geliştirme" çekirdekleri
- Çift sayılar "kararlı" çekirdekler
- Ciddi mimari değişimleri
- Alt sürüm numarası
- İkincil sürümler içi yeniliklerde değişir.

Neden Derleme ?

Linuxda çekirdeğin yeni özellikler ve güncellemeler doğrultusunda tekrar derlenebilmesi işletim sistemi için önemli bir esnekliktir.Diyelim ki yeni bir donanım edindiniz, mevcut linux sürümünüz NTFS dosya sistemini desteklememekte veya çekirdekte kaynaklanan bir güvenlik açığı söz konusu ise internette,bu istediğiniz özellikleri sağlayan çekirdeği edinin ve aşağıdaki adımlara göre derlemeye başlayın. Böylece yeni bir linuxunuz olsun,güncel ve eskisinden de sağlam.

Başlıyoruz;

Mutlaka bir Linux açılış disketi oluşturun ve sisteminizi bir kez bu disketten açarak test edin. İşler ters giderse bu disket sisteminizi açmanızı sağlayacak. Bir açılış disketi oluşturmak için

```
# mkbootdisk --device /dev/fd0 2.4.16
```

komutunu kullanabilirsiniz. 2.4.16 sisteminizdeki çekirdek sürüm numarasını açıklamaktadır. Siz bu numarayı kendi sisteminizde yüklü bulunan çekirdek sürüm numarası ile değiştirebilirsiniz.

En yeni ve kararlı Linux Çekirdeğini sisteminize indirin. Bu, muhtemelen sürüm adını taşıyan ve .tar.gz ile biten bir dosya olacak. *.tar.gz ile biten bir dosya tar ile birleştirilmiş ve gzip ile sıkıştırılmış bir dosya demektir. Çekirdeği indireceğiniz yerde *.tar.bz2 ile biten dosyalar da olabilir, bu dosyalar da bzip ile sıkıştırılmıştır. Sıkıştırılmış dosya boyutu yaklaşık 28 MB 'dır.

Kontrol :::**Yapılan iş** (A)..Linux İşletim Sistemi Yapı Analizi

(28)

Dosyayı /home/kullanici/ dizinine indirdiğimizi varsayarsak, Sisteme root olarak login olduktan sonra, Kendi ev dizinimize indirdiğimiz çekirdeği açmadan önce ilgili yerine taşıyoruz.

```
[skeser@skeser skeser]# mv /home/baris/linux-2.4.17.tar.gz /usr/src/
```

Eski çekirdeğin sembolik bağlantısını siliyoruz

```
[skeser@skeser skeser]# rm /usr/src/linux
```

Çekirdek kaynak kodunu açıyoruz

```
[skeser@skeser skeser]# cd /usr/src
[skeser@skeser skeser]# tar -zxvf linux-2.4.17.tar.gz
```

/usr/src dizini altında çekirdek sürüm numarası ile yeni bir dizin oluştu. Bu dizinin adını değiştiriyoruz

```
[skeser@skeser skeser]# mv linux linux-2.4.17
```

Biraz önce sildiğimiz sembolik bağlantıyı yeni çekirdek için yeniden kuruyoruz

```
[skeser@skeser skeser]# ln -s /usr/src/linux-2.4.17 linux
```

Şimdi geldik işin en zor kısmına. Çekirdeği kaynak kodundan derlemeye başlıyoruz. Konsoldaysanız

```
make menuconfig
```

X ortamındaysanız

```
make xconfig
```

komutunu vererek çekirdek derleme menüsünü hazırlayın.

Şu andan itibaren yapacağınız çok önemli. Çekirdeği hiç hatasız derleyebilmek için son derece dikkatli olmalı ve herhangi bir soruya 'Yes' ya da 'OK' komutunu vermeden önce mutlaka okumalısınız. Çekirdeğe eklemek istediklerinizi menüden ekleyebilir, istediklerinizi çıkarabilir ya da modül olarak eklenmesini sağlayabilirsiniz. Modül olarak yaptığınız ekler çağrıldığı zaman çalışacaktır. Bu menü Linux' u öğrenmek için bulabileceğiniz en değerli kaynaktır. Bütün seçenekleri tek tek okumanızı tavsiye ederim. Şu anda Linux' un ruhuyla yüzyüze konuşma şansını yakaladınız, keyfini çıkarın. Seçenekleri belirledikten sonra 'Save' ve 'Exit' komutlarını verin. Yaptığınız değişiklikler bir metin dosyasında saklanır ve çekirdeğin derlenmesi sırasında okunur. Sırasıyla

```
[skeser@skeser skeser]# make dep
[skeser@skeser skeser]# make clean
[skeser@skeser skeser]# make bzImage
```

komutlarını verin. Eğer çekirdeğe modül olarak birşeyler eklediyseniz ayrıca

```
[skeser@skeser skeser]# make modules
[skeser@skeser skeser]# make modules_install
```

komutlarını da çalıştırmanız gerekiyor.

Çekirdeği derledik. Şimdi onu uygun bir yere taşımamız gerekiyor. Lütfen dikkat, burası son derece önemli.

```
[skeser@skeser skeser]# cp /usr/src/linux/arch/i386/boot/bzImage
/boot/linux-2.4.17
```

Şimdi de System.map dosyasını gerekli yere taşıyalım

```
[skeser@skeser skeser]# cp /usr/src/linux-2.4.17/System.map
/boot/System.map-2.4.17
```

Değiştirmemiz gereken birkaç sembolik bağlantı daha var. /boot dizinine geçin

```
[skeser@skeser skeser]# cd /boot
[skeser@skeser skeser]# rm /boot/vmlinux
```

Kontrol :::

Yapılan iş (A)..Linux İşletim Sistemi Yapı Analizi

(29)

:::

```
[skeser@skeser skeser]# rm /boot/System.map
[skeser@skeser skeser]# ln -s /boot/System.map-2.4.17 /boot/System.map
[skeser@skeser skeser]# ln -s /boot/vmlinuz-2.4.17 /boot/vmlinux
```

Az kaldı ha gayret. Şimdi LILO' ya yeni çekirdeğimizin yerini göstermeliyiz. /etc/lilo.conf dosyasını dilediğiniz bir metin editörü ile açın.

```
[skeser@skeser skeser]# pico /etc/lilo.conf
```

lilo.conf' un eski hali şuna benziyor olmalı :

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux
image=/boot/vmlinuz-2.2.13
label=linux
root=/dev/hda5
read-only
```

Sizin kendi sisteminize uygun yapacağınız olarak yapacağınız değişiklik sonrası lilo.conf aşağıdakine benzer olacaktır :

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux-yeni
image=/boot/vmlinuz-2.4.17
label=linux-yeni
root=/dev/hda5
read-only
image=/boot/vmlinuz-2.2.15
label=linux-eski
root=/dev/hda5
read-only
```

Gördüğümüz gibi yeni çekirdeği önseçimli hale getirdik ve ona linux-yeni etiketini verdik. LILO' yu yeniden yüklüyoruz.

```
[skeser@skeser skeser]# /sbin/lilo
```

LILO' yu yeniden çalıştırınca aşağıdaki mesajı almanız gerekli

```
Added linux-yeni *
Added linux-eski
```

Yanında * işareti bulunan çekirdek önseçimlidir ve açılışta bir çekirdek seçmezseniz bu çekirdek yüklenecektir. Bilgisayarınızı yeniden başlatın

```
[skeser@skeser skeser]# shutdown -hfr now
```

Sisteminiz yeniden açıldığında yeni derlediğiniz çekirdekle çalışacaktır.

Kontrol :::

Yapılan iş (A)..Linux İşletim Sistemi

(30) :::

Mail Server Name : venus.tekfen.com.tr
www.obua.org'dan qinstaller-1.1.tar.gz dosyası download edilir.
 qinstaller-1.1.tar.gz dosyasını /home/caktas'dizinine kopyalanır.

```
# tar -zxvf qinstaller-1.1.tar.gz
# cd qinstall
# vi qinstall.conf
```

Bu konfigürasyon dosyasının içine, default host kısmına 'tekfen.com.tr' yazılır,

```
# cd /home/caktas
# get http://www.qmail.org/qmail-patch
```

Bir kaç ön hazırlık yapalım

2.konsol ekrana geçelim , login olalım,

```
# cp /home/caktas/qmail-patch /usr/local/src/qinstall/qmail
# cd /usr/local/src/qinstall
# patch -p0 < qmail-queue-patch
```

bu satırı yazıp konsolun aklında tutmasını sağlıyoruz
 Qmail installer dosyaları açıp hemen kurmaya kumaya başladığından
 Qmail'i patch etmek için vakit kalmıyor
 Bu sebeple bu komutları hazır olması gerekir.

Tekrar 1.ekrana dönüyoruz.

```
# cd qinstall
# ./qinstall
```

!! hemen ikinci (2) ekrana geçiyoruz. !!

```
# cp /home/caktas...
# cd/usr/local/src/...
# patch-p0...
```

Şimdi 1.ekrana geçip kurulumun bitmesini bekleyeceğiz.

Patch ederken hatalar olursa bazı dosyaların kurulu olmamasından kaynaklanabilir.
 Örn: dbl-devel ve gdbm-devel dosyalarında
 Qmail başarılı bir şekilde kuruldu. !!

Şimdi Qmail scanner için maildrop 1.4.0 kurulumu ;

```
#tar-zxvt maildrop
#cd maildrop*
#configure
#make
#make install-strip
#make man
```

F-Prot Kurulumu (Antivirüs Taraması İçin)

Qmail-Scanner Time::Hires istiyor
 (/home/caktas dizinindeyim)

```
#wget http://www.cpan.org/authors/id/D/DE/DEWEG/Time-Hires-01.20.tar.gz
```

Kontrol :::

Yapılan iş (A)..Linux İşletim Sistemi

(31)

```
#tar-zxvf time*
#cd time*
#per makefile.PL
#make
#make test
#make install
#cd
#cd qmail-s*
# ./configure-install
#cd contrib
# ./test-installation.sh
# ./test-installation.sh --doit
# vi /home/vpopmail/etc/tcp.smtp
```

Açılan tcp.smtp dosyasının içine
;allow;QMAILQUEUE="/var/qmail/bin/qmail-scanner-queue.pl" satırını eklenir

```
# etc/init.d/qmail cdb
```

Şimdi sıra vpopmail manager kurulumunda , (vqadmin) www.inter7.com vqadmin

```
#tar -zxvf vqadmin.2.3.2 tar.gz
#cd vqadmin
#./configure
#make
#make install-strip
```

Kurulumdan sonra ;

```
#vi /etc/httpd/conf/httpd.conf /* httpd.conf dosyasının içinde
```

```
#<Directory "var/www/cgi-bin/vqadmin">
```

```
#Deny from all
```

```
#Options exec CGI
```

```
#Allow Override AuthConfig
```

```
#Order deny allow
```

```
#</Directory>
```

satırları eklendi ..

```
#vi /var/www/cgi-bin/vqadmin/.htaccess
```

"Authuserfile /usr/local/apache/conf/vqadmin.passwd" Satırını;

"Authuserfile /etc/httpd/conf/vqadminpasswd user password" Olarak değiştirdim..

```
#httpasswd -bc /etc/http/conf/vqadmin.passwd user password
#vi /var/www/cgi-bin/vqadmin/vqadmin.acl
```

default satırı kalacak

senior * admin satırını **senior * caktas** olarak değiştirdim

```
#chmod 600 /var/www/cgi-bin/vqadmin/.htaccess
```

#web serveri yeniden başlatıldı, fakat **forbidded** hatası verdi

```
#vi /var/www/cgi-bin/vqadmin/vqadmin.acl
```

senior*caktas satırını **admin*caktas** olarak değiştirilir .başarılı!!!

Son olarak server time sone ayarları ;

```
ln -s /usr/share/zoneinfo/Europe/Istanbul /etc/local/localtime
```

Kontrol :::

:::