

YILDIZ TEKNİK ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

Türkçe Database Simülasyonu

Ü. Gülsüm Yakut Demirbaş

Yüksek Lisans Tezi

YILDIZ TEKNİK ÜNİVERSİTESİ
KÜTÜPHANE DOKÜMANTASYON
DAİRE BAŞKANLIĞI

Kot : R 368
46

Alındığı Yer : Fen Bil. Ens.

Tarih : 13.4.95

Fatura :

Fiyatı : 16.400?

Ayniyat No : 1-4

Kayıt No : 51024

UDC :

Ek :

YILDIZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÖZET

ABSTRACT

1. SÖZGE (FILE) DÜZENLERİ

1.1 Dizin

1.2 Klasik Veri

1.2.1 Erişim

1.2.2 Erişim

1.3 Veri

1.3.1 Avantajlar

TÜRKÇE DATABASE SİMÜLASYONU

2. DATABASE TİPLERİ

2.1 Dizin

2.2 Hiyerarşik

2.3

2.4

2.5

2.6

2.7

2.8

2.9

2.10

2.11

2.12

2.13

2.14

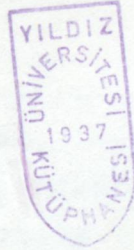
2.15

2.16

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜH. Ü.GÜLSÜM YAKUT DEMİRBAŞ

İSTANBUL, 1988



SONUÇ	2.4.3 Security.....	2	-17
	2.3.4 Örnek Problem.....	2	-18
PROBLEMLER	2.3.5 SQL DDL Komutları.....	2	-19
	2.3.6 SQL DML Komutları/Katalogları.....	2	-20
	2.2.7 Örnek Sorgulamalar.....	2	-21

3. MENU DRIVEN

3.1 Giriş	33	-1
3.2 Database Yazılımlarının Yaklaşımı.....	3	-2

4. YLDZDB SİSTEMİNİN TANITIMI

4.1 Giriş	4	-1
4.2 Gerçekleştirilen Model.....	4	-1
4.3 Gerçeklenmesi.....	4	-2
4.3.1 Seçilen Dil.....	4	-2
4.3.2 Sistem Katalogları.....	4	-2
4.3.3 Katalogların Yapısı.....	4	-3
4.3.4 File Tipleri ve Anlamları.....	4	-5
4.4 Değişkenlerin Sınırları.....	4	-7
4.5 Kullanılan Altprogramlar.....	4	-8
4.6 Hata Mesajları.....	4	-17
4.7 Utility Programlarmalar.....	4	-17
4.8 Öneriler.....	4	-20
4.9 Çalıştırılması.....	4	-20
4.A Blok Diagramlar.....	4A-1	

SONUÇ

ONSÖZ

PROGRAMLAR

YLDZDB.C	Pr1
YLDZDBI.C	Pr2
YLDZDBB.BAT	Pr3
YLDZDBR.BAT	Pr4
YLDZDBL.C	Pr5

KAYNAKLAR

SÖZLÜK

ÖZGEÇMİŞ

YILDIZ TEKNİK ÜNİVERSİTESİ

YILDIZ TEKNİK ÜNİVERSİTESİ

1998

ÖNSÖZ

Günümüzde insanoğlu, yaptığı işleri bilgisayarlara yaptırma çabasıdadır. Son yıllarda uzman sistemler kurarak, insanın unutabileceği küçük ayrıntıları dahi bilgisayarlara yükleyerek, güvenilir uzman bilgisayarlar yaratmaktadırlar.

Yapay Zeka konusundaki çalışmaların temelinde; toplanacak verilerin konacağı bir database kavramı yatmaktadır. Bilgisayara kazandırılacak her türlü yeteneğin gerekli bilgilerinin toplanabileceği bir veriler grubu (database) kurulması gerekmektedir. Dolayısıyla bu tip uygulamalarda database amaç değil araç olarak kullanılır.

Bu projede amaç, Türkçe bilen bir kişinin rahatlıkla menüler aracılığı ile kullanabileceği bir database sistemi kurmaktır.

Bu projenin hazırlanmasında değerli yardımlarından dolayı Sayın Prof. M.Yahya KARSLIGİL'e , tezimin yazılmasında emeği geçen tüm arkadaşlarıma ve eşime teşekkür ederim.

YILDIZ UNİVERSİTESİ

Ü.Gülsüm YAKUT DEMİRBAŞ

İSTANBUL, 1988

Bu kataloglarda oluşabilecek class bir konularına karşı
YLDZDBB komutu kullanılarak backup alınmalıdır. Herhangi bir
bozukluk durumunda ise YLDZDBB komutu kullanılarak ya son
backup a geri dönme sağlanmalıdır.

ÖZET

Konusu Türkçe Database Simulasyonu olan bu projede ana
hedef, Türkçe düşünen bir kişinin menüler aracılığı ile
kullanabileceği bir database sistemini gerçekleştirmektir.

Sistem YLDZDB adını taşımakta olup Hiyerarşik ve Relational
Modellerin karışımından oluşur. Kayıtlara sıralı erişimi
gerçeklediği için Hiyerarşik, verinin bünyesinde pointer
içermediği içinse Relational Model kategorisine
sokulabilir.

YLDZDB sisteminin kurulması için MS/DOS altında YLDZDBI
komutu çalıştırılmalıdır. Bu komut ile Sistemin ihtiyaç
duyduğu sistem katalogları yaratılır. Yazılımın
çalışabilmesi için bu kataloglar disk üzerinden asla
silinmemelidir.

Bilgiler tablolar halinde tutulurlar. Sistemdeki her
tabloya ait bir takım bilgiler SYSCAT ve SYSCOL isimlerini
taşıyan sistem kataloglarında birer satır halinde
bulunurlar. Sistemde yapılacak her türlü iş için bu
kataloglara referans gösterilir.

Bu kataloglarda oluşabilecek olası bir bozulmaya karşı YLDZDBB komutu kullanılarak backup alınmalıdır. Herhangi bir bozukluk durumunda ise YLDZDBR komutu kullanılarak en son backup a geri dönüş sağlanmalıdır.

The main aim of this project whose subject is 'THE TURKISH Bozuk veri içeren kataloglar ile yapılan çalışma istenen sonucu vermeyecektir. YLDZDBL komutu ile bu katalogların listesi alınabilir.

The system is called YLDZDB and is a combination of Hierarchical and Relational Database Models. Since providing sequential access to files, it is hierarchical and also it is relational because of not containing any pointer in the structure of data.

In order to install YLDZDB system YLDZDBI command should be used under MS/DOS operating system. By this command the system catalogues which are needed by system are created. To keep on the system active, these catalogues must never be removed on disk.

Informations are stored as tables. A set informations about each table of the system, represented as a line of SYSCAT and SYSCOL catalogues. For all the process in the system the catalogues are referenced.

To overcome a possible error in the catalogues there should be a backup facility and it may be realized when an error occurs by using YLDZDBI command. This provides the opportunity to reach the previous check point.

ABSTRACT

The main aim of this project whose subject is 'THE TURKISH DATABASE SIMULATION' is ; to create a menu driven database system which can be used by anyone who knows Turkish.

The system is called YLDZDB and is a combination of Hierarchical and Relational Database Models. Since providing sequential access to files , it is hierarchical and also it is relational because of not containing any pointer in the structure of data.

In order to install YLDZDB system YLDZDBI command should be used under MS/DOS operating system. By this command the system catalogues which are needed by system are created. To keep on the system active these catalogues must never be removed on disk.

Informations are stored as tables. A set informations about each table of the system , represented as a line of SYSCAT and SYSCOL catalogues. For all the process in the system the catalogues are referenced.

To overcome a possible error in the catalogues there should be a backup facility and it may be realized when an error occurs by using YLDZBR command. This provides the opportunity to reach the previous check point.

The process which use errorness tables will obviously produce faulty results. YLDZDBL command gives a list of these catalogues.

Oyle ki yeni yazimlar gelistirmek icin gerekenden daha
 cok zaman ve para harcaniyor, ancak BIM'ler istekleri
 gereklilikleri gibi kendilerine uygun guven ve yitirme
 kabiliyetleri degiyorlardir. O guverde BIM'lerdeki
 guvenin yitmesi ve yetkiye inancin gibi kaynak
 yetersizlikleri de bu sonuclara en onemli etkenlerdendir.

Oyleyse bugun bilginin tekneliklari BIM'lere ozellikle gercek
 zamanli, interaktif uygulamalar ile ilgili gerceklestirme
 guvenini yitmesi ve yitirme alanlari
 olmaktadır. Bu guvenin yitilmesi zaman gecip de
 bir yandan guvenin yitilmesi icin yeni yazimlar
 gereklilikleri gerektirmektedir.



1. KÜTÜK (FILE) DÜZENLEME YÖNTEMLERİ

1.1. Giriş

BİM'ler kuruluşlara önce muhasebe, bordro türü uygulamalar
 ile işletimsel düzeyde girmiş; zamanla yönetimlerin
 planlama, denetim, karar alma işlevlerinde 'bilgi' nin
 önemini belli etmesi, BİM'lerin hizmet ve alanını
 genişletmiştir.. Ancak, kuruluşların her düzeyine bilgi
 üretmeye çalışan BİM'ler, yeni programlar geliştirmenin;
 özellikle zaman kısıtlamasından doğan zorluklarını, eski
 programlarda ekleme ve değişikliklerle gidermeye
 yönelmişler, sonuçta kısa dönemli çözümler, uzun dönemde
 hizmetin niteliğini bozmuştur. Birbirinden ayrı yürütülen
 bilgi işlem uygulamalarının sonuçları yöneticilere
 aktıkça, ortak bilgilerde tutarsızlıklar saptanmış ve
 bunları yok etmek amacı ile ,kurulu yazılımlar üzerinde
 daha karmaşık işlemler yapılmaya başlanmıştır.

Öyle ki,yeni yazılımlar geliştirmek için gerekenden daha çok zaman ve çaba harcanıyor,ancak BİM'ler istekleri kaşılayamadığı gibi kendilerine duyulan güveni de yitirme tehlikesini yaşıyorlardı.O günlerde BİM'lerdeki donanım,yazılım ve yetişmiş insangücü gibi kaynak yetersizlikleri de bu sonuçlarda en önemli etkenlerdendi.

Oysa bugün,bilişim teknikbilimi,BİM'lere özellikle gerçek zamanlı , interaktif uygulamalar ,iletişim gerçekleştirme yönünde gelişmiş donanım ve yazılım olanakları sunmaktadır.Bu konudaki yetişmiş insan gücü de küçümsenemeyecek boyutlara ulaştığı için yeni yazılımlar üretilmeye başlanmıştır.

Bu doğrultuda bilginin ancak sağlıklı verilerden elde edilebileceği ve koşulların var olan bilgi erişme,işleme sistemini zorladığı gözlemlenerek üst yönetim ve diğer kullanıcılar arasındaki buzları eritmek amacı ile yeni bir sistem ;DataBase Management System(DBMS) ya da Türkçe adı ile Veri Tabanı Yöneti Sistemi(VTYS) fikri ortaya atılmıştır.

Amaç öncelikle sağlıklı bilgiye ulaşmak olduğu ve bilgiyi üretmek için veriler tek başına yeterli olmadığı için bulunacak çözüm, veriler arası ilişkileri de kapsamalı ve bunlar tutarsızlıkları giderici biçimde olmalıdırlar.İşte bu çözüm VERİ TABANI(VT)'dir ve şöyle tanımlanabilir 'Bir kuruluşun değişik uygulamalarında kullanılmak üzere gereksiz yinelemelerden arınmış olarak,merkezi denetimle bellekte tutulan ilişkili verilerin bütününe VT adı verilir.

1.2.Klasik Veri Düzenleme Yöntemleri

Bir uygulama programında, alışılan yaklaşım ,uygulama bilgilerinin file larda tutarak ,bunları işletim sistemi ve genel amaçlı bir programlama dilinin file düzenleme ve erişim yöntemleri ile işlemektir.

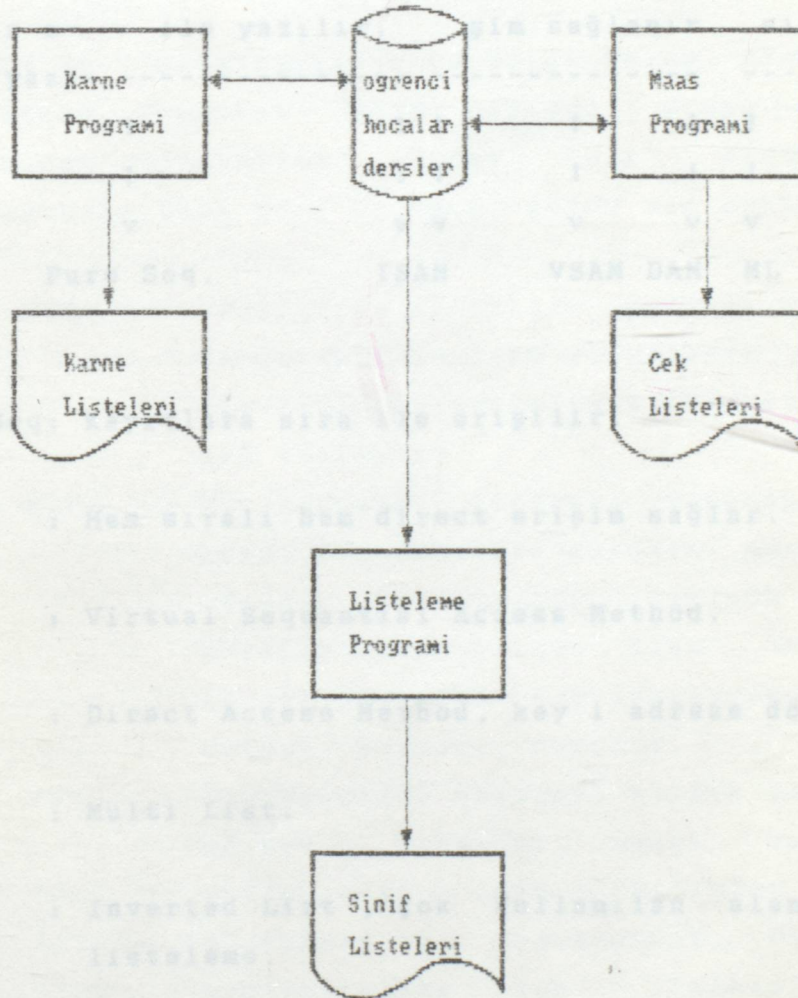
Eğer, aynı ya da ilişkili veriler ,değişik uygulama amaçları için ayrı kütüklerde yineleniyorsa, bellek savurganlığının ötesinde bir sorunla karşılaşmaktadır.Verinin, değişik file larda farklı güncellik seviyesinde bulunması tutarsızlıklara yol açmakta,sonuçta kullanıcılar sağlıklı bilgi edinememekten yakınmaktadırlar.

Şekil 1.1 Klasik Veri Düzenleme Yöntemlerinde karşılaşılan tipik bir durumu göstermektedir.

1.2.1.Erişim Metodları

Ana bellekte adrese kolaylıkla erişilebileceği için bir veri depolamak ve bu depolanmış veriye erişmek oldukça kolaydır.Oysa disk,teyp ya da disket gibi yan bellek adı altında topladığımız bellek tiplerinde bir veriyi depolamak ve bu depolanmış veriye ulaşmak bir hayli zaman kaybına sebep olur.Bu erişimi sağlamak için geliştirilmiş ACCESS METHOD larından bazılarını şöylece sayabiliriz ;Serial Access Method, Sequential Access Method(SAM), Random Access Method ve Linked List ana grupları altında Indexed Sequential Access Method(ISAM), Direct Access Method(DAM), Inverted List,.... gibi dallara ayırabiliriz.

1.2.1. Erişim Yöntemlerinin Sanstik Gösterimi

Klasik Düzenleme Yöntemleri
Storage Structure

SERIL 1.1 : KLASIK VERI DÜZENLEME YÖNTEMİ

1.2.2.Erişim Metodlarının Şematik Gösterimi

Kütük Düzenleme Yöntemleri							
Storage Structure							
Serial	Sequential	Random	Linked List				
Gelen kayıt	Kayıtlar sıra	Rastgele eri-	Keylere göre				
sirasız o-	ile yazılır.	şim sağlanır.	sıralıdır.				
larak yazı-							
lır.	!	!	!	!	!	!	!
	!	!!	!	!	!	!	!
	v	v v	v	v	v	v	v
Pure Seq.	ISAM	VSAM DAM	ML	IL	Trees		

Pure Seq: Kayıtlara sıra ile erişilir.

ISAM : Hem sıralı hem direct erişim sağlar.

VSAM : Virtual Sequential Access Method.

DAM : Direct Access Method, key i adrese dönüştürür.

ML : Multi List.

IL : Inverted List , çok kullanılan alanlara göre listeleme.

1.3. Veri Tabanı Yönetim Sistemleri

Veri Tabanı Yönetim Sistemlerinde, veri Klasik Veri Düzenleme Yöntemlerindeki erişim metodlarından biri kullanılarak ve kullanıcıya gösterilmeyen oldukça karmaşık işlemlerden geçerek depolanır. Kullanıcı bu yöntemleri ve bunları nasıl kullanacağını bilmek zorunda değildir. Sadece ona sunulan basit veri tanımlama ve sorgulama komutlarını bilmesi yeterli olup veri paylaşımı mümkün kılındığı için veri tekrarı da önlenmiştir. Hafızayı en fazla fayda verecek şekilde kullandığını bile bilmesi gerekmez.

Şeki 1.2 de 'Bilgisayarlar ve Bilgi İşlem Amerikan Ulusal Standartlar Komitesi VYTS Çalışma Grubu (ANSI/SPARC) tarafından önerilen örnek bir VTYS mimarisi görülmektedir. Buna göre üç genel düzey söz konusudur:

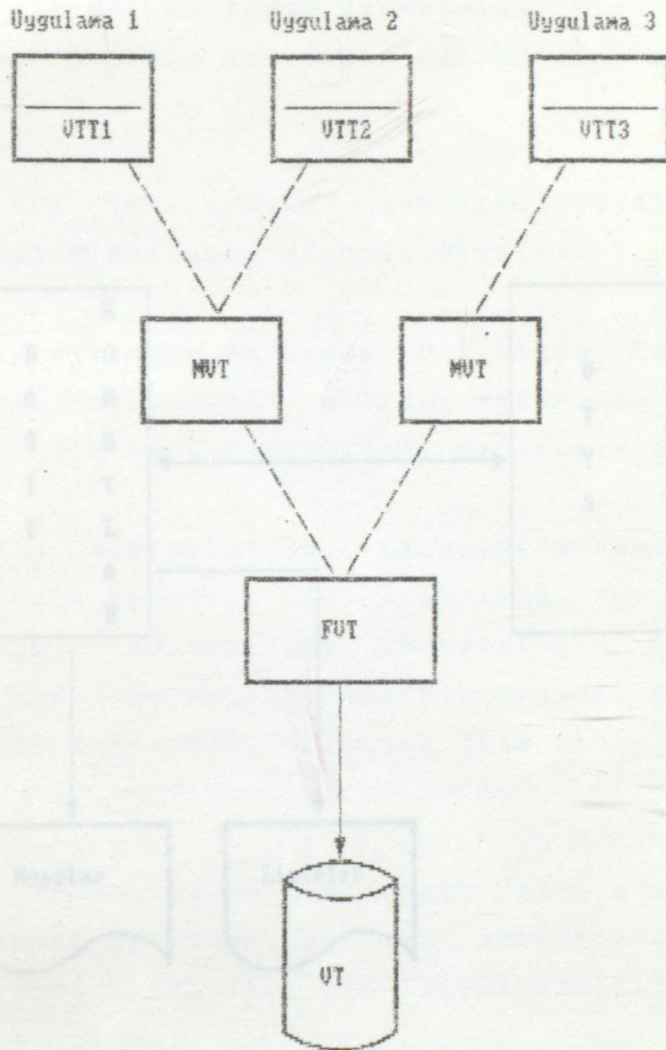
1. FVT : Veriler ve ilişkiler ,çeşitli veri yapıları ve kütük düzenleme yöntemleri kullanılarak bir fiziksel yapıya oturtulmuştur ki bu FVT nin tanımıdır.
2. MVT : Kullanıcıları fiziksel yapıdan soyutlamak için, uygulamalara ilişkin veri, ilişkiler bütünüün belirli bir biçimde düzenlenmesi gerekir . Mantıksal olan bu düzenleme gerçek yaşamdaki uygulamaların bir modelini ortaya koymayı amaçlar.
3. VTT : Kullanıcılar verileri alışık olduğu biçimde görmek isterler, ancak bellekte her kullanıcıya özgü biçimler yaratmak olanaksız ve anlamsızdır. Bu durumda, kullanıcılara tek fiziksel yapı ve kavramsal şemadan yararlanarak,

gereksinimlerine uygun mantıksal veri görünümleri sunulur . Bu da verilerin tanımlanması aşamasında önem kazanır.

Görüldüğü gibi, bir VT kullanıcısı, MVT ile FVT den habersizdir; ancak kullanıcının VTT ile oluşturduğu bir istek FVT de karşılanmalıdır. Öyleyse, düzeyler arasında bir geçiş, dönüşüm yolu olmalıdır ki bu da VTYS mimarisinde düzeyler arası iletişim programları olarak yerini almıştır. Bu iletişim programlarının değişikliklere açık olmasıyla , kullanıcı ve uygulamalar , hem bellekteki veri tutma düzenine (fiziksel veri bağımsızlığı), hem de kavramsal şemadaki veri ve ilişki düzenlenmesinde (mantıksal veri bağımsızlığı) oluşacak değişikliklerden korunmaya çalışılır.

VTYS şemaları, bilgisayar ortamında yine VTYS'lerin sunduğu dil olanakları ile geliştirilip kullanılır ve VTYS'ler amaçlarına bu diller aracılığı ile erişirler. Söz konusu diller genel olarak şöyle sınıflandırılabilir:

1. VERİ İŞLEME DİLLERİ (DML): Kullanıcıların dış şemayı kolayca kullanması ve uygulamalar geliştirilmesi amacıyla erişim, ekleme, silme, güncleme gibi veri işleme olanakları sağlayan dillerdir. Tek başlarına ya da genel amaçlı bir programlama dili içinde kullanılabilen komutlar içerirler. Genellikle uzman olmayan kullanıcılara yönelik olduklarından doğal dile yatkındırlar.
2. ŞEMA TANIMLAMA DİLLERİ (DDL): VTYS şemaları ile eşlemeleri, veriyle ilişkilerin özellik ve yapılarını temel alarak ; tanımlamak amacıyla geliştirilmiş ve yetişmiş uzmanlarca kullanılabilen dillerdir.



SERIL 1.1 : VERI TABANI YAPISIN SIKLINI

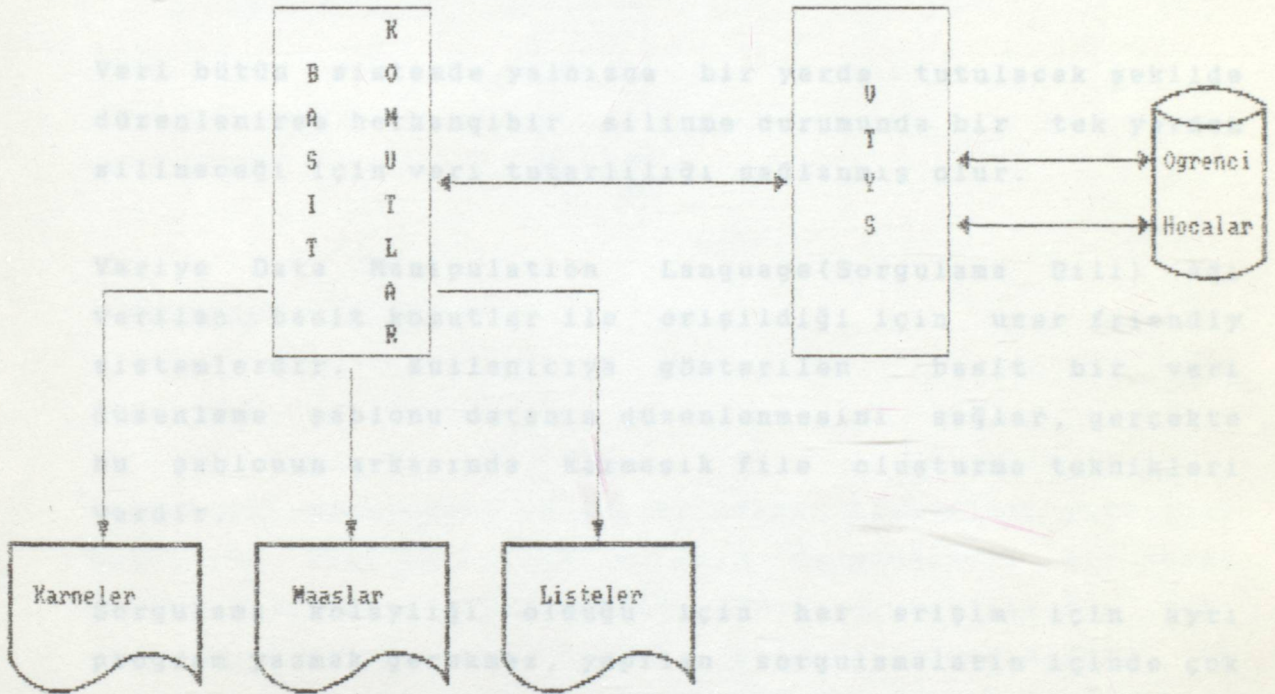
- VTT : Veri Tabani Tanimi
- MVT : Mantiksal Veri Tabani
- FVT : Fiziksel Veri Tabani
- VT : Veri Tabani

SERIL 1.2 : GENEL BIR UTYS SEMASI

1.3.1. Avantajları / Dezavantajları

Veri paylaşımı mümkün kılınarak klasik veri düzenleme yöntemlerinin en en büyük dezavantajı olan veri tekrarı ortadan kalkar, aynı zamanda kullanıcıların kullanımını da kolaylaştırır.

Kullanıcıları verilerdeki değişikliklerden soyutlayarak gerçek ve kolay kullanım olanağı yaratır.



SEKIL 1.3 : VERİ TABANI YONETİM SİSTEMİ

1.3.1. Avantajları / Dezavantajları

Veri paylaşımı mümkün kılınarak klasik veri düzenleme yöntemlerinin en en büyük dezavantajı olan veri tekrarını önler, aynı zamanda hafızanın iyi kullanımını da sağlamış olur.

Kullanıcıları verilerdeki değişikliklerden soyutlayarak esnek ve kolay kullanım olanağı yaratır.

Veri bütün sistemde yalnızca bir yerde tutulacak şekilde düzenlenirse herhangi bir silinme durumunda bir tek yerden silineceği için veri tutarlılığı sağlanmış olur.

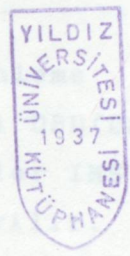
Veriye Data Manipulation Language (Sorgulama Dili) adı verilen basit komutlar ile erişildiği için user friendly sistemlerdir. Kullanıcıya gösterilen basit bir veri düzenleme şablonu datanın düzenlenmesini sağlar, gerçekte bu şablonun arkasında karmaşık file oluşturma teknikleri vardır.

Sorgulama kolaylığı olduğu için her erişim için ayrı program yazmak gerekmez, yapılan sorgulamaların içinde çok farklı veriler birbirleri ile ilişkilendirilebilir.

Tüm bu avantajları yanında; pahalı oluşları, çok hafıza götürmeleri, bilgisayar sisteminin performansını düşürmeleri ve güvenliğinin zor sağlanması gibi birtakım dezavantajları da vardır. Ancak kuruluşlar pahalı olan yazılım mühendisi çalıştırmak yerine VTYS kullanmayı tercih etmektedirler.

2.2. Hiyerarşik Model

bu database sistemi olarak kabul edilmiş. IBM'ün IBM/Information Management System) adındaki ilk relational database için bir hiyerarşik gibi düşünülmüş. Her veri bir nesne olarak kabul edilmiştir. Bu sistem, sorgular için olan DE/1(Data Language/1) aracılığıyla PL/1, COBOL, ASSEMBLY gibi yüksek seviyeli dillerde de çalıştırılabilir.



2. DATABASE TIPLERİ

2.1. Giriş

Gerçek yaşamdaki veri ve aralarındaki ilişkileri yansıtmak amacı ile veri ve ilişkiler için de mantıksal bir model geliştirilir. Buna VERİ MODELİ (VM) adı verilir. Bir veri ögesi çoğunlukla tek başına anlamlı olmadığından, VM geliştirmek için veriler arası ilişkilerin nitelikleri göz önüne alınmalıdır.

Database fikrinin doğmasından itibaren sırası ile Hiyerarşik Model, Network Model ve Relational Model olmak üzere 3 tip model geliştirilmiş olup her yeni model kendinden önceki modeller dezavantajlarını kompanse etmek için ortaya atılmıştır.

2.2.Hiyerarşik Model

İlk database sistemi olarak kabul edilen IBM ürünü IMS(Information Management System) aslında fiziksel database lerin bir birikimi gibi düşünülebilir.Hem batch hem online olarak kullanılabilen bu sistem sorgulama dili olan DL/I(Data Language/1) aracılığı ile PL/I,COBOL,ASSEMBLY gibi yüksek seviyeli dillerin içinden de çağırılabilir.

IMS in veri düzenleme için kullandığı DDL(Data Definition Language) i DBD(DataBase Definition) adını taşır ve bu şablon ile,veriler IMS in kendi yöntemlerince kontrol edeceği bir hale getirilir.

Hiyerarşik düzende verinin yanısıra; sayısı seçilen erişim yöntemine bağlı olarak değişen miktarda fiziksel pointer lar da tutulduğu için yapı, veri düzeni karmaşıklaştıkça hem bu yapının dizaynını yapan kişi hem de durumu dışardan inceleyen kişi için komplike bir hal alır. Ayrıca n-->m li ilişkiler kurulması gerekiyor işe tüm bu tanımlara ek olarak Logical Linking Pointer ları da ekleneceğinden durum oldukça karmaşık hale gelir.

2.2.1.Kuralları, Avantajları/Dezavantajları

Kurallar :

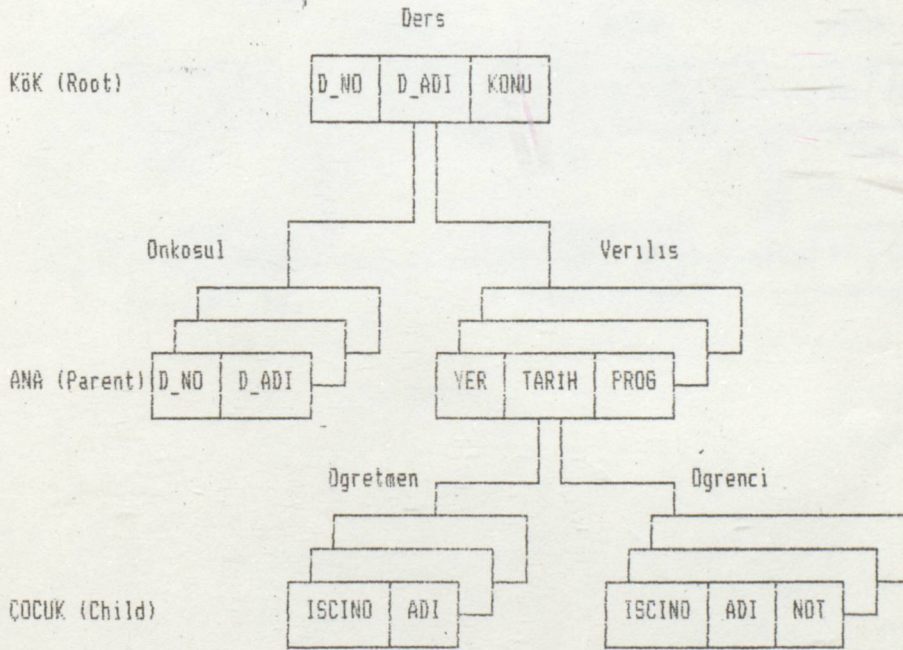
- o) Herhangibir hiyerarşik model veri tabanında tek tip kök kaydı vardır.
- o) Her kök kaydın deęişen sayıda çocuk kaydı olabilir.
- o) Bir çocuk kayıt bir başka çocuk kaydın babası olabilir.
- o) Alt düzeydeki her ana kayıt için deęişen sayıda çocuk kaydı olabilir.
- o) Hiçbir çocuk kaydı baęlı olduęu ana kayıt olmadan var olamaz. Ana kayıt silinirse ona baęlı olan tüm çocuk kayıtlar da silinir.

Avantajları / Dezavantajları :

- i) Bir veri birden çok ilişkiye girdiğinde veri yinelenmesi olur.
- ii) Sebep olduęu bilgi tekrarını önlemek için izlenecek yöntemler önemli ölçüde zaman ve yer kaybettirir.
- iii) n --> m li ilişkilerde güçlkle ve zorlamayla uygulanır.
- iV) Pointer lar ile hızlı erişim yapılır.

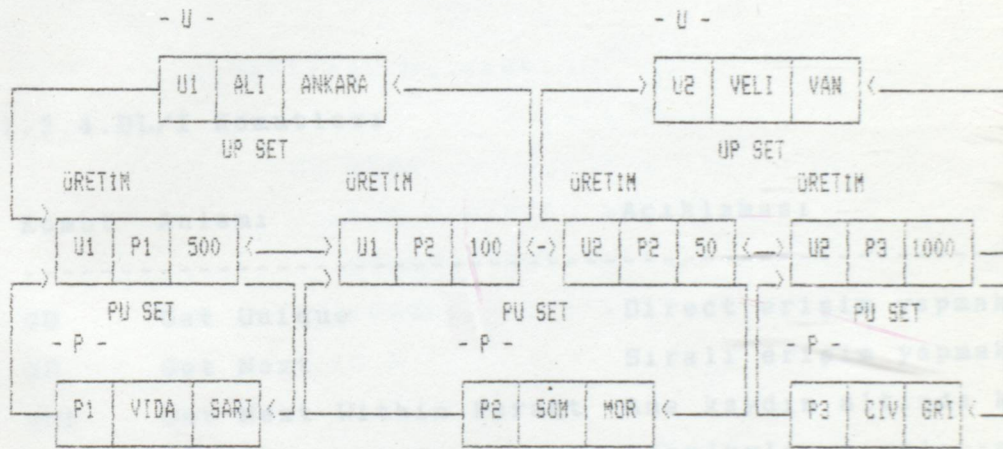
2.2.2.Örnek Problem

Kendi bünyesinde bir eğitim bölümü açan bir firmada çalışanlara değişik şehirlerde değişik öğretmenlerce bir dizi kursun verildiğini ve öğretmen ile öğrencilerin bu firmanın çalışanları olduğunu belirten bir database kurunuz. Bu database in şematik gösterimi ;



2.2.2.Örnek Problem

Kendi bünyesinde bir eğitim bölümü açan bir firmada çalışanlara değişik şehirlerde değişik öğretmenlerce bir dizi kursun verildiğini ve öğretmen ile öğrencilerin bu firmanın çalışanları olduğunu belirten bir database kurunuz. Bu database in şematik gösterimi ;



2.2.3.DBD Örneği

```

DBD NAME=EGITIM
SEGM NAME=DERS, BYTES=256
FIELD NAME=(D_NO, SEQ), BYTES=3, START=1
FIELD NAME=D_ADI, BYTES=33, START=4
FIELD NAME=KONU, BYTES=220, START=37
SEGM NAME=ONKOSUL, PARENT=DERS, BYTES=36
FIELD NAME=(D_NO, SEQ), BYTES=3, START=1
FIELD NAME=D_ADI, BYTES=33, START=4

```

2.2.4.DL/I Komutları

Komut	Anlamı	Açıklaması
GU	Get Unique	Direct erişim yapmak için
GN	Get Next	Sıralı erişim yapmak için
GNP	Get Next Within Parent	Ana kaydın altında kalmak koşuluyla sıralı erişim yapmak için
GHN	Get Hold Next	Sıralı erişimi yaptığı kayda pointer 1 sabit kılar.
GHNP	Get Hold Next Within Parent	Ana kaydın altında kalmak koşuluyla sıralı erişim yaparken bulunan kayda pointer 1 sabit kılar.
ISRT	Insert	Kayıt ekler.

DLET Delete Pointer ın sabit olduğu
 kaydı silmek için
 REPL Replace Pointer ın sabit olduğu
 kaydı yenilemek için

2.2.5.Örnek Sorgulamalar

Sorgu 1 :Ankara'da verilen ilk derse ait bilgileri bulunuz.

```

VERILIS(TARİH=0873,YER='İZMİR')
GU DERS
VERILIS(YER='ANKARA')

```

Sorgu 2 :Ankara'da verilen ilk dersten başlayarak yalnız
 'A' notu alan

öğrencileri bulunuz.

```

GU DERS
VERILIS(YER='ANKARA')
OGRENCI(NOT='A')
N : GN OGRENCI(NOT='A')

```

goto N

Sorgu 3 :Ağustos 1973 tarihinde İzmir'de verilen 3 nolu
 dersi alan

GÜNAY'ın notunu 'B' yapınız.

```

GHU DERS(D_NO=3)
VERILIS(TARİH=0873,YER='İZMİR')
OGRENCI(ADI='GÜNAY')
(NOT='B')
REPL

```


Sorgu 4 :Ağustos 1973 tarihinde İzmir'de verilen 3 nolu derse 'GÜNAY'

Kuralları IS_NO=1235 ve NOT='B' kaydını ekleyiniz.

```

      (IS_NO=1235)
      (ADI='GÜNAY')
      (NOT='B')
      ISRT DERS(D_NO=3)
      VERILIS(TARİH=0873,YER='İZMİR')
      OGRENCI
  
```

Sorgu 5 :Herhangibir dersten 'A' alan öğrencileri bulunuz.

Kayıt Ekimdeki Üyelik Tipleri :

```

      GU DERS
      VERILIS
      OGRENCI(NOT='A')
N:    GN OGRENCI(NOT='A')
      Goto N
  
```

2.3.Network Model

1971 Nisan ayında toplanan DataBase Task Group yeni tip bir DBMS sistemi ortaya atmıştır.Bu model tümüyle COBOL programlama diline uygun olarak yazılmış olup Hiyerarşik modelde olduğu gibi bünyesinde fiziksel pointer lar içeren bir sistemdir.Veriler kayıt bazında tutulup,belirlenen setlere bağlanırlar.

2.3.1.Kuralları, Avantajları/Dezavantajları

Kurallar :

- o) Tanımlanan her setin mutlaka bir owner ı vardır.
- o) Tanımlanan her setin sıfır ya da daha çok member ı olabilir.
- o) Bir setin member ı başka bir setin owner ı olabilir.

Kayıt Eklemedeki Üyelik Tipleri :

- i) AUTOMATIC : Kayıt eklendiği zaman sistem hangi sete ve hangi bağlantılara konacağını kendisi bulur.
- ii) MANUAL : Kayıt eklendiği zaman hiç bir sete bağlanmaz . Gerekli yere kaydı yaratan kişi koyar.

Kayıt Silmedeki Üyelik Tipleri :

- i) FIXED : Bir setin owner ı silinirse ona bağlı tüm member larda silinir.
- ii) MANDATORY : Kayıt bir owner a bağlı olmak koşulu ile aynı set içerisinde owner değiştirebilir.
- ii) OPTIONAL : Kayıtlar bir set içerisinde hiçbir owner a bağlı olmadan varolabilir.

Avantajları/Dezavantajları :

i) Verinin bünyesinde pointer içermesi en büyük dezavantajdır.

ii) Bağların kurulması ve koparılmasındaki bir yanlışlık bilgi kaybına neden olur;

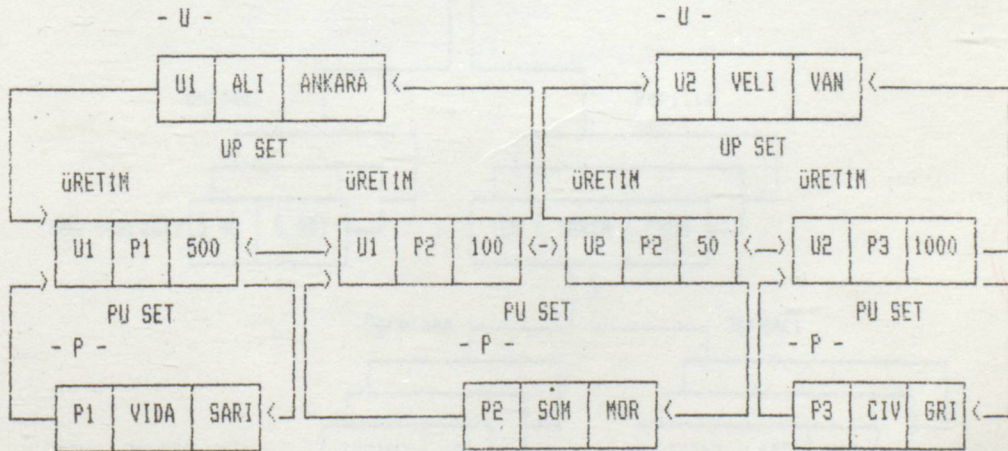
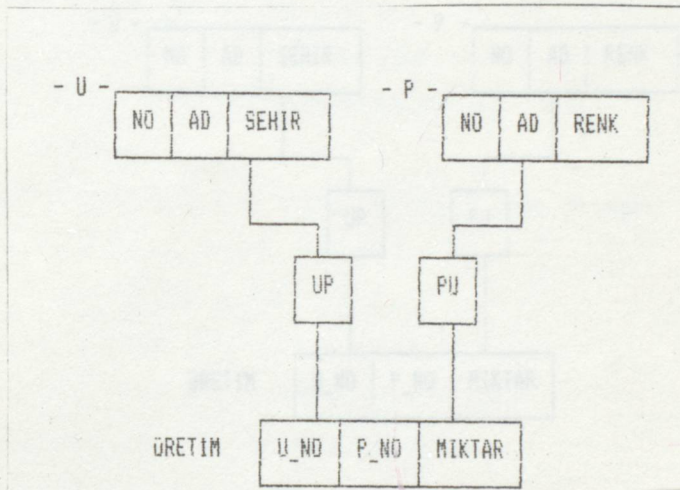
iii) Bir veri birden çok ilişki içindeyse , veri tutanağı birçok bağ bilgisi içerir;

iv) İlişkili veri tutanaklarına erişebilmek için bir noktadan yola çıkılır ve tutanaktaki bağlardan, istenen ilişkiye ait olanı izlenerek diğer tutanaklara geçilir ;böylece bağlar boyunca gezinti yapılır bu da zaman kaybına sebep olur.

v) n-->m li ilişkileri sağlar.

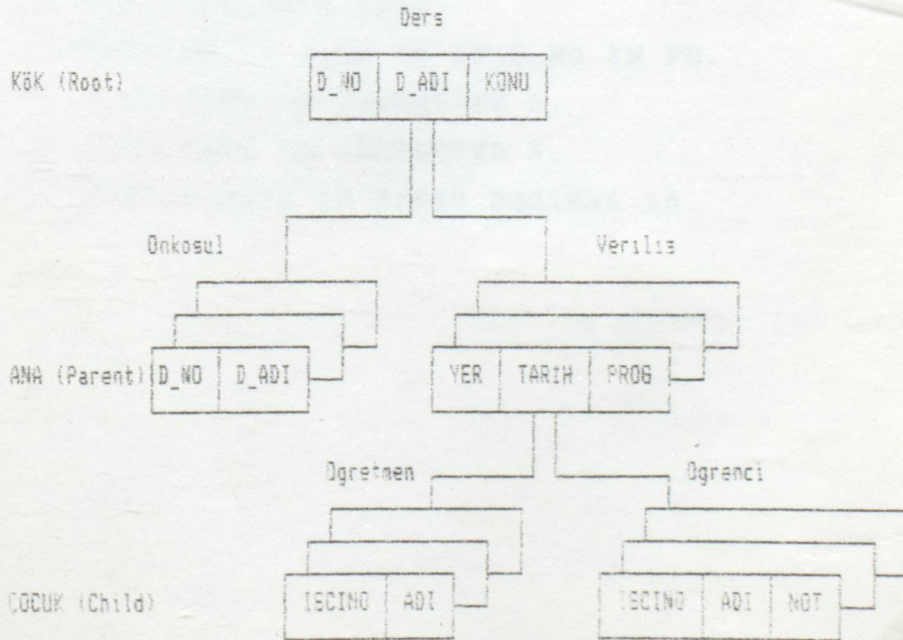
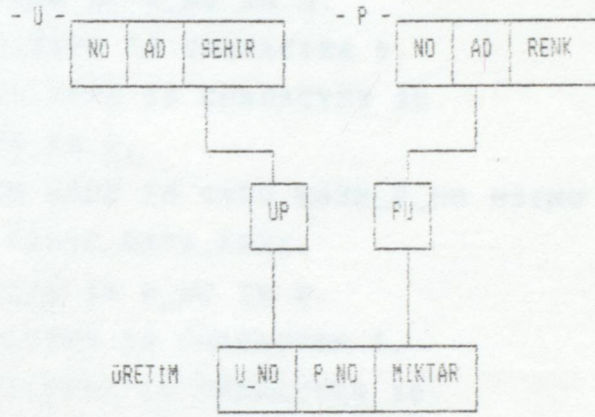
2.3.2. Örnek Problem

Üretici ve parça bilgilerini üretim ile ilişkilendirecek şekilde bir database kurunuz. Bu database in şematik gösterimi ;



2.3.2.Örnek Problem

Üretici ve parça bilgilerini üretim ile ilişkilendirecek şekilde bir database kurunuz.Bu database in şematik gösterimi ;



2.3.3.DBTG/DDD Örneđi

SCHEMA NAME IS URETICI_PARCA.

AREA NAME IS BASIC_DATA_AREA.

AREA NAME IS LINK_DATA_AREA.

RECORD NAME IS U;

LOCATION MODE IS CALC HASH_U_NO USING U_NO IN U;

WITHIN BASIC_DATA_AREA;

IDENTIFIER IS U_NO IN U.

02 U_NO;TYPE IS CHARACTER 5.

02 U_ADI;TYPE IS CHARACTER 20.

RECORD NAME IS P;

LOCATION MODE IS CALC HASH_P_NO USING P_NO IN P;

WITHIN BASIC_DATA_AREA;

IDENTIFIER IS P_NO IN P.

02 P_NO;TYPE IS CHARACTER 5.

02 P_ADI;TYPE IS CHARACTER 10.

02 RENK;TYPE IS CHARACTER 10.

RECORD NAME IS URETIM;

LOCATION MODE IS SYSTEM DEFAULT;

WITHIN LINK_DATA_AREA;

IDENTIFIER IS U_NO IN UP,P_NO IN PU.

02 U_NO;TYPE IS CHARACTER 5.

02 P_NO;TYPE IS CHARACTER 5.

02 MIKTAR;TYPE IS FIXED DECIMAL 10.

SET NAME IS UP;
 OWNER IS U;
 MEMBER IS URETIM;
 INSERTION IS AUTOMATIC
 RETENTION IS FIXED;
 KEY IS ASCENDING P_NO IN URETIM
 DUPLICATES ARE NOT ALLOWED
 NULL IS NOT ALLOWED;
 SET SELECTION IS THRU UP OWNER
 IDENTIFIED BY IDENTIFIER U_NO IN U;
 SET NAME IS PU;
 OWNER IS P;
 MEMBER IS URETIM;

2.3.4.DBTG/DML Komutları

Komut	Açıklaması
FIND	Belirlenen kaydı current hale getirir.
GET	Current kayda erişimi sağlar.
MODIFY	Current kayıta değer değişikliği yapar.
CONNECT	Current kaydı bir ya da daha çok current sete ekler.
DISCONNECT	Current kaydı bir ya da daha çok current setten çıkarır.
STORE	Yeni kayıt yaratır ve bunu otomatik olarak current yapar.

ERASE kayıt PERMANENT Fix ya da mandatory tipi üyelik-
 lerde owner+member 1 siler. Optio-
 nalda yalnızca owner 1 siler.
 MOVE SELECTIVE Optional da başka owner a ait ol-
 mayan kayıtları siler. Sette mem-
 ber varsa silmez.
 GET ALL Tüm member ları siler.

2.3.5. Örnek Sorgulamalar

Sorgu 1 : U1 hakkındaki bilgileri veriniz.

```
MOVE 'U1' TO U_NO IN U.  
FIND ANY U.  
GET U.
```

Sorgu 2 : U1 in ürettiği tüm parçaları bulunuz.

```
MOVE 'U1' TO U_NO IN U.  
FIND ANY U.  
IF UP SET IS EMPTY GOTO SON.  
N : FIND NEXT URETIM IN WITHIN UP.  
IF ENDSET='YES' GOTO SON.  
GET URETIM
```

```
MOVE 'U1' TO U_NO IN U.  
FIND ANY U.  
GOTO N
```

SON :

```
FIND ANY U.  
MOVE 'U1' TO U_NO IN U.  
FIND ANY U.  
CONNECT U TO USER.
```


Sorgu 3 :U4 ün P5 i hangi miktarda ürettiğini bulunuz.

MOVE 'U4' TO U_NO IN U.

FIND ANY U.

MOVE 'P5' TO P_NO IN URETIM.

FIND URETIM WITHIN UP CURRENT USING P_NO IN CURRENT.

IF ENDSET='YES' GOTO YOK.

GET URETIM

YOK :

GHU DERS(D_NO=3)

VERILIS(TARİH=0873,YER='İZMİR')

OGRENCI(ADI='GÜNAY')

(NOT='B')

REPL

Sorgu 4 :U1 in bulunduğu şehri 'ANKARA' olarak değiştiriniz.

MOVE 'U1' TO U_NO IN U.

FIND ANY U.

MOVE 'ANKARA' TO SEHIR IN U.

MODIFY U.

Sorgu 5 :U1 kaydını USET isimli sette 'STATU='10A') olan owner a bağlayınız.

MOVE '10A' TO STAT IN STATU.

FIND ANY STATU.

MOVE 'U1' TO U_NO IN U.

FIND ANY U.

CONNECT U TO USET.

2.4.Relational Model

1969 ve 1970 yıllarında E.F.CODD tarafından teorisi ortaya atılan model 1974 yıllarında gerçekleştirilmiş olup bugün SYSTEM R,Query BY Example(QBE), SQL/DS(ya da DB2) gibi çeşitli yazılımlarda kullanılmaktadır.Bu modelde veriler tablolar halinde depolanır ve birbirleri ile mantıksal olarak ilişkilendirilmeleri sağlanır.Verinin bünyesinde fiziksel pointer içermemesi diğer iki modelden en farklı yanıdır.

2.4.1.Kuralları, Avantajları/Dezavantajları

Kurallar :

- o) Tablo yaratmada teoride hiçbir ön koşul yoktur.
- o) Birden fazla tablo birbiri ile ilişkilendirilecekse ortak alanlar tanımlanmalıdır.

Avantajları/Dezavantajları :

- i) Veriler arası integrity sağlamak için özel komutlara sahiptir.
- ii) Veri içinde pointer olmadığı için daha az yere ihtiyaç gösterir.
- iii) Data Sharing diğer iki modelden daha gerçekçi olarak sağlanır.

iV) DML ve DDL komutları oldukça basit ve kullanımı kolaydır.

V) Birden fazla tablonun birleşimini gerektiren sorgulamalara cevap verebilir.

2.4.2. Integrity (Tutarlılık)

SQL/DS (VM/SP işletim sistemi altındaki sistemin adı) ya da DB2 (MVS veya DOS/VSE altındaki sistemin adı) da Entity Integrity Rule ve Referential Integrity Rule başlığı altında iki tip integrity den sözedilebilir.

Entity Integrity Rule :

Bir tablo içinde key olarak seçilen bir alan unique olarak tanımlanmalıdır. Böylece unique key olarak belirtilen alanda boş ya da tekrar edilen kayıtlar bulunamadığı için kayıtların tekrarı önlenmiş olur.

Referential Integrity Rule :

Herhangibir tablo ile ilişkilendirilen bir tablodan bir veri silindiği zaman referans ettiği tablo ya da tablolardaki tüm ilişkin kayıtlar da silinmeli; eklendiği zaman da referans ettiği tüm tablo ya da tablolara eklenmelidir.

Bu amaçla tablolar yaratıldıktan sonra bir defaya mahsus olmak üzere belli kuralları koyan TRIGGER lar set edilir. Herhangibir işlem yapıldığında bu trigger ı ilgilendiren bir durum varsa kural işletilir.

2.4.3.Security

SQL/DS te view define edilerek yeni virtual tablolar yaratılabilir ve bazı yetkisiz kişilere sadece bu view lar kullandırılarak istenmeyen kişilerin istenmeyen alanlara erişimi engellenebilir. Ayrıca SQL/DS te yetki sınırlandırılması GRANT komutu ile kullanıcılara verilen yetkilerle sağlanır. Bir kullanıcıya yetki verme yetkisi de verilebilir. Ancak bu kullanıcının yetkisi alındığında onun verdiği tüm yetkiler de geri alınmış olur.

2.4.4. Örnek Problem

Supplier, part ve bunlar arasındaki her supplier in ürettiği her parçanın ne miktarda üretildiğine ilişkin bir database kurunuz. Bu database in şematik gösterimi;

- S -

SNO	SADI	STATUSU	SEHIR
S1	Smith	20	Londra
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	Londra
S5	Adams	30	Ankara

- P -

PNO	PADI	RENK	AGIR
P1	Nut	Kırmızı	12
P2	Bolt	Yeşil	17
P3	Screw	Mavi	17
P4	Screw	Kırmızı	14
P5	Coa	Mavi	12
P6	Cog	Kırmızı	19

- SP -

SNO	PNO	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

2.4.5. SQL DDL Komutları

```
CREATE TABLE S(SNO CHAR(3),SADI CHAR(10),STATUSU INT,
SEHIR CHAR(10))
```

```
CREATE TABLE P(PNO CHAR(3),PADI CHAR(10),AGIR INT,
RENK CHAR(10))
```

```
CREATE TABLE SP(SNO CHAR(3),PNO CHAR(3),QTY INT)
```

```
CREATE UNIQUE INDEX SI ON S(SNO)
```

```
CREATE UNIQUE INDEX PI ON P(PNO)
```

```
CREATE UNIQUE INDEX SPI ON SP(SNO,PNO)
```

```
DEFINE TRIGGER ON
```

```
  DELETION OF S:
```

```
  DELETE SP
  WHERE SNO='X'
```


2.4.6. SQL DML Komutları/Katalogları

Komut	Açıklaması
CREATE	Tablo, index veya view yaratma da kullanılır.
DELETE	Tablodan kayıt silmek için kullanılır.
DROP	Tablo, view veya index silmek için kullanılır.
GRANT	Kullanıcılara yetki vermek için kullanılır.
INPUT	Tabloya bulk insertion yapmakta kullanılır.
INSERT	Tabloya tek kayıt eklemek için kullanılır.
REVOKE	Verilen yetkiyi geri almak için kullanılır.
SELECT	Tablodan kayıt seçmek için kullanılır.
UPDATE	Kayıt yenileme için kullanılır.

Kataloglar :

Kullanıcının SQL/DS sisteminde yaptığı her aktif iş birtakım kataloglara bazı entry lerin girmesine neden olur; bu aktif iş bir tablo yaratmak, kayıt eklemek, silmek, index yaratmak, ... gibi çeşitli işler olabilir. Database de herhangi bir işlem yapılacağı zaman önce bu kataloglar kontrol edilir buralardan alınan bilgilere göre cevap verilir.

2.4.7. Örnek Sorgulamalar

Sorgu 1: Paris'te oturanları S ve SP tablosundan silen bir trigger hazırlayınız.

```

WHERE SNO=
DEFINE TRIGGER ON SNO
DELETION OF S: SP
DELETE SP WHERE PNO=
WHERE SNO= (SELECT PNO
SELECT SNO
FROM S WHERE COLOR='KIRMIZI')
AND SEHIR='PARIS' WHERE SEHIR='PARIS'
OR STATUSU > 20

```

Sorgu 2: Paris'te oturan ve kırmızı parçaları üreten ya da stat
fazla olan üreticileri bulunuz.

```

SELECT SNAME
FROM S
WHERE STATUSU > 20 OR SEHIR='PARIS' AND SNO=
SELECT SNO
FROM SP
WHERE PNO=
SELECT PNO
FROM P

```

Sorgu 3: Paris'te oturan ve kırmızı parçaları üreten ya da stat
fazla olan üreticileri bulunuz.

```

SELECT SNAME
FROM S
WHERE SNO=
SELECT SNO
FROM SP GROUP BY SNO
WHERE COUNT(PNO) > 3

```


YA DA

```

UPDATE S
(SELECT SNAME
FROM S SNO='S1'
WHERE SNO=

```

Sorgu 6: P1' i silinir (SELECT SNO
FROM SP

```

DELETE FROM S WHERE PNO='P1'

```

```

(SELECT PNO

```

Sorgu 7: S tablosu içerisindeki SADI ve SADI'yi gösteren
bir view kurunuz.

```

FROM P

```

```

WHERE COLOR='KIRMIZI'))

```

```

AND SEHIR='PARIS'

```

```

OR STATUSU > 20 ) (SNO,SADI)

```

```

AS SELECT SNO,SADI

```

Sorgu 3: Tüm parçaları üreten üreticileri bulunuz.

Sorgu 7 (SELECT SNAME
FROM S
WHERE SNO=

```

GRANT SELECT (SELECT SNO

```

```

TABLE SP FROM SP

```

```

WITH GRANT WHERE PNO=ALL

```

```

(SELECT PNO

```

```

FROM P)))

```

Sorgu 4: 3 veya daha çok parça üreten üreticileri bulunuz.

```

SELECT SNAME

```

```

FROM S

```

```

WHERE SNO=

```

```

SELECT SNO

```

```

FROM SP GROUP BY SNO

```

```

WHERE COUNT(PNO) >= 3

```

Sorgu 5:S1 in statusunu 30 yapınız.

```
UPDATE S
SET STATUSU=30
WHERE SNO='S1'
```

Sorgu 6:P1 i siliniz.

```
DELETE FROM P WHERE PNO='P1'
```

Sorgu 7:S tablosu üzerinden sadece SNO ve SADI nı gosteren bir view kurunuz.

```
DEFINE VIEW NEW_S(SNO,SADI)
AS SELECT SNO,SADI
FROM S
```

3. KNO DRIVEN

Sorgu 7:S tablosu üzerinden seçim yapma hakkını ALI'ye GRANT option 1 ile veriniz.

```
GRANT SELECT ON
TABLE EMP TO BOB
WITH GRANT OPTION
```




3.MENU DRIVEN

3.1.Giriş

Her türlü bilgisayar yazılımının amacı;sonuçta kişilere hizmet olduğu için , bu hizmeti sunarken uç kullanıcıların program ve programlama dili hakkında en az bilgiye sahip olmak koşulu ile bir yazılımı kullanılıp yönetilebilmesi son yıllarda en çok hedeflenen bir amaç durumuna girmiştir.Bu amaçla micro bilgisayarların yazılımlarında başlayıp işletim sistemlerine sıçrayan 'Menu Driven' modası macro bilgisayarların yazılımları ve işletim sistemlerine kadar uzanmıştır.

Daha önceleri hazırlanmış olan yazılımlar önlerine eklenen 'Menu Driven' yazılımları veya 'Menu Selection'lar ile tuş kullanarak ya da 'Mouse' adı verilen, kullanım kolaylığı sağlayan aletlerle bu akıma uygun hale getirilmiştir.

Bu akımın ana amacı, programı kullanan kişilere en az sorumluluk yükleyip, programları, neredeyse sokaktaki adamın kullanabileceği düzeyde basitleştirmek olarak belirtilebilir. Yani , kullanıcı; programı kullanırken programın getirdiği 'Komut Formatı (Syntax)' kısıtından bağımsız olarak sürekli bir menüler sisteminde dönmek suretiyle, en az hata yaparak kullanılabilir. Her programı kullanmak için özel eğitim görmüş kişilere ihtiyaç duyulmasını önleyici bir etkidir.

3.2. Database Yazılımlarının Yaklaşımı

Bu akıma uyularak daha önceden yazılmış olan database yazılımlarının içine menüler aracılığı ile erişimi sağlayacak bazı komutlar eklenmiştir

Örneğin ; DBASE III deki 'Assist' komutu kullanılarak menüler aracılığı ile kullanılabilir hale getirilebilir.

SQL/DS sisteminde; varolan SQL/E (Structured Query Language/Edit), IBUD (Interactive Build Update Delete) gibi programları kullanarak menüler aracılığı ile komutlar kullanılabilir.

DL/I komutları , QUERY DL/I product ı kurulmak suretiyle menüler aracılığı ile kullanılabilir hale gelir.

Data üzerinde aramalarda 3 adet pointer gerektirilmekte
yazılmaktadır ve ilgili ortadan kaldırılması da bu pointerler
kullanılarak gerçekleştirilir. Pointer yapısı ise bir
adet 1. sütun üzerinde iki adet olacak şekilde
piktürdür. Sütun üzerinde bulunan pointer'lerden biri
karakterler halinde , diğeri kelime halinde orijinal yazılmak
için kullanılır. Tasarlanan sistemin menu driven olması
ve 3. Bölümde anlatıldığı gibi seçilmiş olup
geliştirilmiştir. Yöntem program paketleri halinde
kullanılmaktadır. Yazılım Türkçe dilinde bir kişi rahatlıkla
kullanabilmektedir.



4.2. Gerçekleştirilen

4. YLDZDB SİSTEMİNİN TANITIMI

4.2.1. Seçilen Dil

4.1. Giriş

Temel olarak bir database tablo yaratma, silme, listeleme; kayıt ekleme, silme, yenileme olmak üzere altı ana fonksiyona sahiptir. Bu ana fonksiyon iskeleti üzerine ek bazı olanaklar da konarak sistemler birbirlerine göre farklılaştırılırlar.

4.2. Gerçekleştirilen Model

Tek modelin temsilcisi olmayan proje; Hiyerarşik ve Relational Modellerin karışımından oluşmuş olup menüler aracılığı ile çalışacak şekilde düzenlenmiştir. Kayıtlara sıralı erişimi sağladığından Hiyerarşik modele , veri yapısında pointer içermediği için de Relational Modele dahil edilebilir.

Data üzerinde aramalarda 3 adet pointer, gerektiğinde yaratılır ve ihtiyaç ortadan kalktığı zaman da bu pointerlar serbest bırakılırlar. Pointer yapısı satır üzerinde bir adet ; sütun üzerinde iki adet olacak şekilde planlanmıştır. Sütun üzerinde gezinen pointer'lardan biri karakter bazında, diğeri kelime bazında erişimi sağlamak için kullanılır. Tasarlanan sistemin menu driven olması ise 3. Bölümde anlatılan nedenlerden dolayı seçilmiş olup gelişmeye yönelik bir program paketi halinde hazırlanmıştır. Yazılımı Türkçe bilen bir kişi rahatlıkla kullanabilir.

4.3. Gerçeklenmesi

4.3.1. Seçilen Dil

Projenin gerçekleştirilmesinde C Dili kullanılmıştır. Bu dilin alçak seviyeli bir dil olması, dilin yapısından gelen bir takım zorlamalar ve kısıtlamalar projeyi yönlendirici faktörler olmuştur. En büyük zorluklardan birisi doğrudan erişimin sadece BYTE boyutunda yapılabilirliği olmuştur. Bu sorunu halledilebilmek için birtakım algoritmalar geliştirilmiştir.

4.3.2. Sistem Katalogları

YLDZDB sisteminde yapılan bir takım aktif işler bazı sistem kataloglarına kayıt eklenmesine sebep olurlar. Bu kataloglarda, veri tabanında tutulan tabloların kolon sayıları; her bir kolonun adı, tipi, uzunluğu gibi bilgiler tutulur. Veri tabanında yapılan her türlü işlem için önce bu kataloglar incelenir, bilgilere göre karar verilir.

Yeni bir tablo yaratılmak istendiğinde sistem katalogunda aynı isimli tablonun varlığı bu tablo yaratma isteğinin rededilmesine sebep olması bu konuya açıklık kazandıran bir örnektir.

YLDZDB 'nin iki ana katalogu vardır. Bunlar SYSCAT ve SYSCOL adı altında; YLDZDBI (YLDZDB Installation) çalıştırıldığı anda yaratılır ve database sisteminin çalışabilmesi için sürekli disk üzerinde okunup yazılabilecek durumlarda saklanır. Aksi takdirde bu katalogların silinmesi ya da disk üzerinde oluşacak bir donanım hatası yüzünden erişilememesi durumlarında YLDZDB sisteminde tanımlı olan hiçbir tabloya erişim sağlanamaz. Yapılan her türlü işlem sonuçta kataloglarla bağlantılı olduğundan database in son hali bu kataloglardaki durumla çok sıkı ilişki içindedir. Bu sebeplerden dolayı YLDZDBB (YLDZDB Backup) utility'si çalıştırılarak periyodik olarak bu kataloglar yedeklenmelidir. YLDZDBB programı çalıştıktan sonra oluşan ve tipi 'BAC' olan file lar kesinlikle diskten silinmemelidir.

4.3.3. Katalogların Yapısı

SYSCAT katalogu, tablo ismi (tname) ve tabloda bulunacak kolon sayısı (colcount) nı tutar. YLDZDB sistemi tarafından tanınan her tabloya ait bir bilgi satırı tutulur.

SYSCOL katalogu; tablo ismi (tname) ve SYSCAT katalogunda tutulan kolon sayısı kadar kolon ismi (colname), kolon tipi (coltype) ve kolon uzunluğu (collen) alanlarından oluşmuştur. SYSCAT katalogundaki her satıra birebir karşı düşen satırlarda tablo isimleri; birebir karşı düşen sayıda kolon bilgileri bulunmalıdır.

SİSTEM KATALOGLARININ GENEL GÖRÜNÜMÜ

```

!-----+-----+ !-----+-----+-----+
! tname  ! colcount ! ! tname  ! colname ! coltype! collen!
!        !          ! !        !         !         !       !
!=====+=====+ !=====+=====+=====+
!        !          ! !        !         !         !       !
! Tablo1 !        1  ! ! Tablo1 !  col1  ! char  !  3    !
! Tablo2 !        2  ! ! Tablo2 !  col1  ! int   !  2    !
! Tablo3 !        1  ! !        !  col2  ! char  !  1    !
! .      !        .  ! ! Tablo3 !  col1  ! int   !  4    !
! .      !        .  ! ! .      !      .  ! .     ! .     !
! .      !        .  ! ! .      !      .  ! .     ! .     !
!-----+-----+ !-----+-----+-----+

```

SYSCAT

SYSCOL

Bir tablonun YLDZDB tarafından işlenmesi için gene YLDZDB tarafından dan yaratılmış olması gerekmektedir. Aksi takdirde tablo tanınmayabilir. Eğer herhangi bir şekilde kataloglardaki bilgiler aldatılarak tanıma aşaması atlatılsa bile işlemler sırasında istenmeyen sonuçlar doğabilir ve en önemlisi bu işlemler beklenmeyen anlarda katalog bozulmalarına sebep olabilirler.

YLDZDBL programı çalıştırılarak bu katalogların listeleri de çekilebilir.

4.3.4. File Tipleri Ve Anlamları

Program çalışırken aşağıda belirtilen bazı file'lar kullanır, bunların bir kısmı disk üzerinde kalıcı olmak zorunda olup bir kısmı da bazı optionlar seçildiğinde geçici olarak yaratılırlar.

! FILE İSMİ	! KULLANILDIĞI	! ADIM	! AÇIKLAMA
! PRT.PRT	! seçeneğinde	! listenin	! yerleştirildiği! disk üzerinde kalır. ! ! file'dır. !
! WORK.WRK	! silineceğinde	! kullanılır.	! Tablodan kayıt ! Adım başarı ile ! ! sonuçlanırsa silinir. !
	! yenileneceğinde	! kullanılır.	! Tabloda kayıt ! Adım başarı ile ! ! sonuçlanırsa silinir. !

```

=====+=====+=====+
!           !           !           !
! FILE İSMİ ! KULLANILDIĞI ! AÇIKLAMA !
!           ! ADIM !           !
=====+=====+=====+
!           !           !           !
!           ! Tablo silinmesi! Adım başarı ile !
! SYSCAT.WRK ! adımında ! sonuçlanırsa silinir.!
!           ! kullanılır. !           !
-----+-----+-----+
!           !           !           !
!           ! Tablo silinmesi! Adım başarı ile !
! SYSCOL.WRK ! adımında ! sonuçlanırsa silinir.!
!           ! kullanılır. !           !
-----+-----+-----+
!           !           !           !
!           ! Her adımda ! HİÇ !
! SYSCAT ! referans göste-! SİLİNMEMELİDİR. !
!           ! rilen sistem !           !
!           ! kataloğudur. !           !
-----+-----+-----+
!           !           !           !
!           ! Her adımda ! HİÇ !
! SYSCOL ! referans göste-! SİLİNMEMELİDİR. !
!           ! rilen sistem !           !
!           ! kataloğudur. !           !
-----+-----+-----+
!           !           !           !
! SYSCAT.BAC ! Backup alınınca! RESTORE yapılması !
! SYSCOL.BAC ! oluşur. ! isteniyorsa silin- !
!           ! ! memelidir. !
=====+=====+=====+

```


4.4. Değişkenlerin Sınırları

Program çalışırken kullanıcıdan değeri istenen bazı değişkenler sınırlandırılmış olup bu sınırların dışında değerler verildiğinde ya kabul edilmez ya da beklenmeyen cevaplar alınabilir.

Aşağıdaki tablolarda bu değişkenlere ait kısıtlamalar görülmektedir :

! DEĞİŞKEN İSMİ !	! TİPİ !	! UZUN-! LUÇU !	! MİN ! DEĞ.!	! MAX ! DEĞ.!	! ALABİLECEĞİ ! DEĞERLER !
! Tablo İsmi	! Char !	! 8 !	! - !	! - !	! - !
! Tablo Sayısı	! Int !	! 3 !	! 0 !	! 255 !	! - !
! Kolon İsmi	! Char !	! 8 !	! - !	! - !	! - !
! Kolon Tipi	! Char !	! 4 !	! 1 !	! 255 !	! Int/Char !
! Kolon Sayısı	! Int !	! 3 !	! 1 !	! 255 !	! - !

4.5. Kullanılan Altprogramlar

Altprogram Adı : menu()

Amacı : Ekran ana menüyü çıkarmak.

Parametreleri

Giriş : -

Dönüş : Seçeneğin numarası.

Local : -

Kullandığı

Rutinler : -

Geçici

File'lar : -

Altprogram Adı : inputs

Amacı : Verilen bir prompt'un yanında istenen bir değişkene istenen uzunlukta değer okunması.

Parametreleri :

Giriş : prompt,variable,length.

Dönüş : Değişkenin değeri.

Local : -

Kullandığı

Rutinler : -

Geçici

File'lar : -

Altprogram Adı : yarat

Amacı : Tablo yaratmak.

Parametreleri :

Giriş : -

Dönüş : -

Local : Tablo ismi, alan sayısı, her alanın adı, tipi, uzunluğu.

Kullandığı

Rutinler : inputs

err0

err1

err2

dön

findin

Geçici

File'lar : -

Altprogram Adı : ekle

Amacı : Tabloya kayıt eklemek.

Parametreleri :

Giriş : -

Dönüş : -

Local : Tablo ismi,

Kullandığı

Rutinler : inputs
 : erro
 : dön
 : findin

Geçici

File'lar : -

Altprogram Adı : sil

Amacı : Tabloyu ya da tablodaki kayıtları silmek.

Parametreleri

Giriş : -
Dönüş : -
Local : Kayıt mı/Tablo mu silinecek,
 Tablo ismi,
 kayıt ise belirleyici özelliği.

Kullandığı

Rutinler : inputs
 : erro
 : baslık
 : dön
 : findin

Geçici

File'lar : WORK.WRK
 : SYSCAT.WRK
 : SYSCOL.WRK

Altprogram Adı : list

Amacı : Şartlı yada şartsız kayıt listelemek.

Parametreleri

Giriş : -

Dönüş : -

Local : Şart var mı/yok mu?

Tablo ismi,

Şart varsa belirleyici özelliği.

Kullandığı

Rutinler : inputs

err3

başlık

dön

findin

Geçici

File'lar : PRT.PRT

Altprogram Adı : yenile

Amacı : Kayıt güncelleştirme.

Parametreleri

Giriş : -

Dönüş : -

Local : Tablo ismi,

Kaydın belirleyici özelliği.

Kullandığı

Rutinler : inputs offset word

Dönüş : err3'deki satır sayıları

Local : başlık

dön

Kullandığı : findin

Geçici : -

File'lar : WORK.WRK

File'lar : -

Altprogram Adı : başlık

Altprogram Adı : Viracın

Amacı : Verilen tablonun SYSCAT'tan colcount
değerinin bulunması. 7. satır

yolmak.

Parametreleri

Girişleri : tname

Dönüş : colcount

Local : -

Local : -

Kullandığı

Rutinler : -

Geçici : -

File'lar : -

File'lar : -

Altprogram Adı : findin

Amacı : Verilen tabloda belirtilen kelimeyi
aramak.

Parametreleri

Giriş : tname,offset,word
Dönüş : tname'deki satır sayısı.
Local : -

Kullandığı

Rutinler : -
Geçici
File'lar : -

Altprogram Adı : clrscrn

Amacı : Belirtilen sayı kadar boş satır
yazmak.

Parametreleri

Giriş : count
Dönüş : -
Local : -

Kullandığı

Rutinler : -
Geçici
File'lar : -

Altprogram Adı : don

Amacı : Ana menüye dönüş log'unu yazdırmak.

Parametreleri

Giriş : -

Dönüş : -

Local : -

Kullandığı

Rutinler : -

Geçici : -

File'lar : -

Altprogram Adı : err0

Amacı : SYSCAT'ın okuma amacı ile açılıp

açılmadığının tetkikini yapmak.

Parametreleri

Giriş : -

Dönüş : -

Local : -

Kullandığı

Rutinler : -

Geçici : -

File'lar : -

Altprogram Adı : err1

Amacı : SYSCOL'un append amacıyla açılıp
açılmadığının tetkikini yapmak.

Parametreleri

Giriş : -

Dönüş : -

Local : -

Kullandığı

Rutinler : -

Geçici

File'lar : -

Altprogram Adı : err2

Amacı : SYSCAT'ın append amacıyla açılıp
açılmadığının testini yapmak.

Parametreleri

Giriş : -

Dönüş : -

Local : -

Kullandığı

Rutinler : -

Geçici

File'lar : -

Altprogram Adı : err3

Amacı : Verilen tablonun okuma amacı ile
açılıp açılmadığını test etmek.

Parametreleri

Giriş : tname
Dönüş : -
Local : -

Kullandığı

Rutinler : -
Geçici : -
File'lar : -

SYSCAT KATALOGU İLE DİSK: Bu tabloya erişilemez.
ARASINDA UYUŞMAZLIK. : Tablo disk üzerinde
SYSTEM ERROR. : alınmış değildir.

TABLO SİSTEM KATALOGUNDA: Tabloya erişilemez. YERDE
KAYITLI DEĞİL. : sisteminde kayıtlı değil.
: Tabloyu önce yaratın.

TABLO KOLONLARI SİSTEM : Tabloya erişilemez.
KATALOGUNDA YOK. : Katalog uyumsuzluğa var.

TABLO BOŞUK. : Disk üzerinde bulunmaz
: sorunu var. Diskli temir
: ettiriniz.

KATALOG HATASI. : Kataloga erişilemiyor.
SYSTEM ERROR. : Diskli temir ettiriniz.

4.6.HATA MESAJLARI

Aşağıdaki tabloda programın çalışması sırasında alınan hata mesajları ve bu durumlarda yapılması gerekenler gösterilmiştir.

ALINAN MESAJ	YAPILMASI GEREKEN İŞLEM
TABLO HALEN TANIMLIDIR.	Başka bir tablo ismi ile yeniden deneyiniz.
SYSCAT KATALOGU İLE DİSK ARASINDA UYUŞMAZLIK. SYSTEM ERROR.	Bu tabloya erişilemez. Tablo disk üzerinden silinmiş demektir.
TABLO SİSTEM KATALOGUNDA KAYITLI DEĞİL.	Tabloya erişilemez.YLDZDB sisteminde kayıtlı değil. Tabloyu önce yaratın...
TABLO KOLONLARI SİSTEM KATALOGUNDA YOK.	Tabloya erişilemez. Katalog uyumsuzluğu var.
TABLO BOZUK.	Disk üzerinde donanım sorunu var.Diski tamir ettiriniz.
KATALOG HATASI... SYSTEM ERROR...	Kataloga erişilemiyor. Diski tamir ettiriniz.

```

!=====+=====+
!
! ALINAN MESAJ YAPILMASI GEREKEN İŞLEM
!
!=====+=====+
! KATALOG UYUSMAZLIGI. ! Tabloya erişilemez. !
!-----+-----!
! GIDERILEMEYECEK HATA. ! Disk ile ilgili donanım !
! hatası var.Diski tamir !
! ettiriniz... !
!-----+-----!
! SARTA UYGUN KAYIT ! Başka koşulla deneyiniz. !
! BULUNAMADI... !
!-----+-----!
! SYSxxx KATALOGU BOZUK. ! Backup tan katalogları !
! SYSTEM ERROR... ! geri yükleyiniz... !
!-----+-----!
! TABLO ACILAMADI. ! Diskte donanım hatası var.!
! FILE OPEN ERROR... ! Diski tamir ettiriniz. !
!-----+-----!

```


4.7.UTILITY PROGRAMLAR

YLDZDB sistemi MS/DOS işletim sistemi altında çalışan bir takım yardımcı programlara sahiptir. Bunlar aşağıdaki tabloda ayrıntılı olarak anlatılmıştır.

```

!=====+=====+=====+=====+
!           !           Kullandığı           !           !
! İSMİ      !           file lar           ! Çalıştırılma !
!           !-----+-----! sıklığı      !
!           ! GİRİŞ      ! ÇIKIŞ      !           !
!=====+=====+=====+=====+
! YLDZDBI   !           ! SYSCAT     ! Sadece kuruluş !
! (YLDZDB   !           !           ! aşamasında bir !
! INSTALL) !           !           ! kere ...       !
!-----+-----+-----+-----!
! YLDZDBB   ! SYSCAT     ! SYSCAT.BAC! Kullanımına     !
! (YLDZDB   ! SYSCOL     ! SYSCOL.BAC! göre belirlenir!
! BACKUP)   !           !           ! Normalde hafta-!
!           !           !           ! da bir alınma- !
!           !           !           ! lıdır.         !
!-----+-----+-----+-----!
! YLDZDBR   ! SYSCAT.BAC! SYSCAT     ! Herhangibir     !
! (YLDZDB   ! SYSCOL.BAC! SYSCOL     ! nedenle geri    !
! RESTORE)  !           !           ! dönüş gerekirse!
!-----+-----+-----+-----!
! YLDZDBL   ! SYSCAT     !           ! Sistem katalog-!
! (YLDZDB   ! SYSCOL     !           ! larının liste- !
! LIST)     !           !           ! si istendiğinde!
!-----+-----+-----+-----!

```

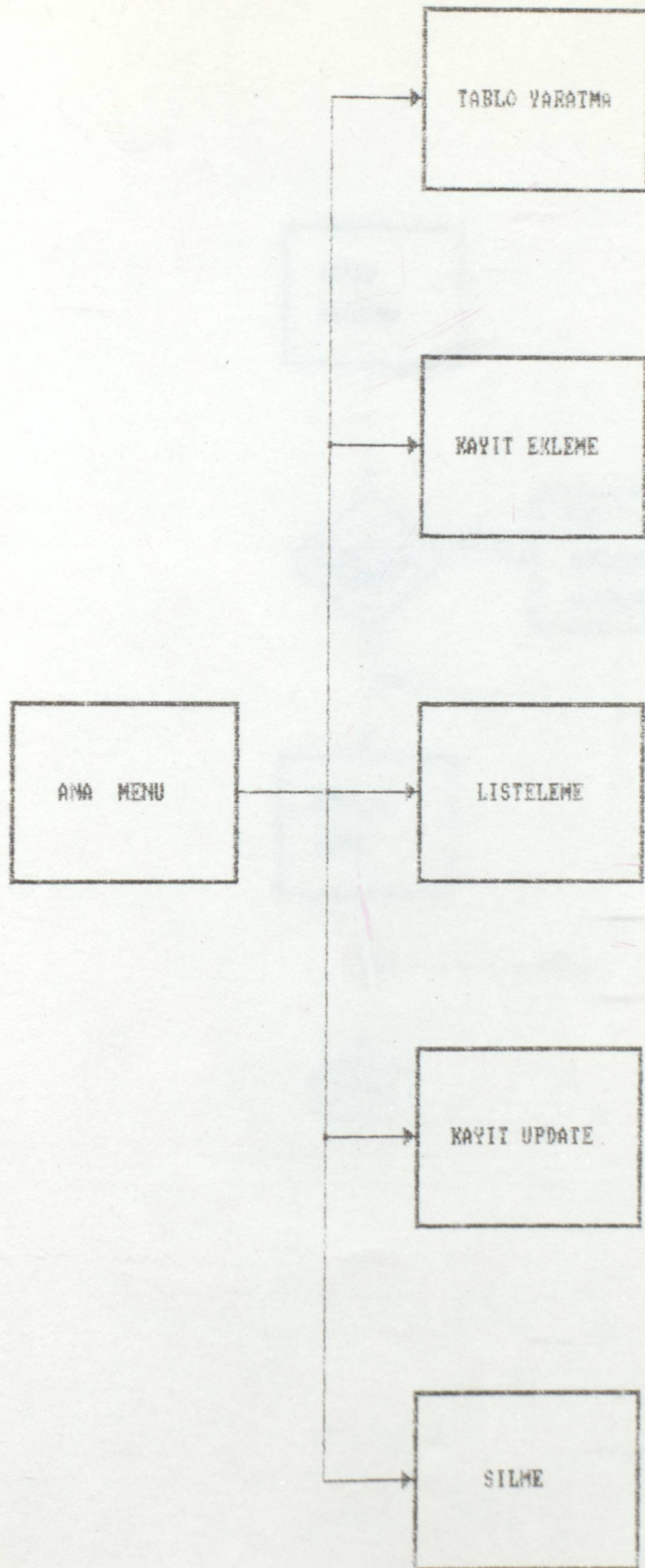

4.8.ÖNERİLER

Yaratacılacak olan tabloların isimlerinin büyük ya da küçük harf olması farklılık getirdiğinden kullanımda bir takım sorunlara yolaçabilir.Bu nedenle oluşabilecek sorunlara çare bulabilmek amacı ile tablo isimlerinin sürekli büyük harf verilmesi ve yaratılırken de büyük harf olarak kullanılması önerilmektedir.

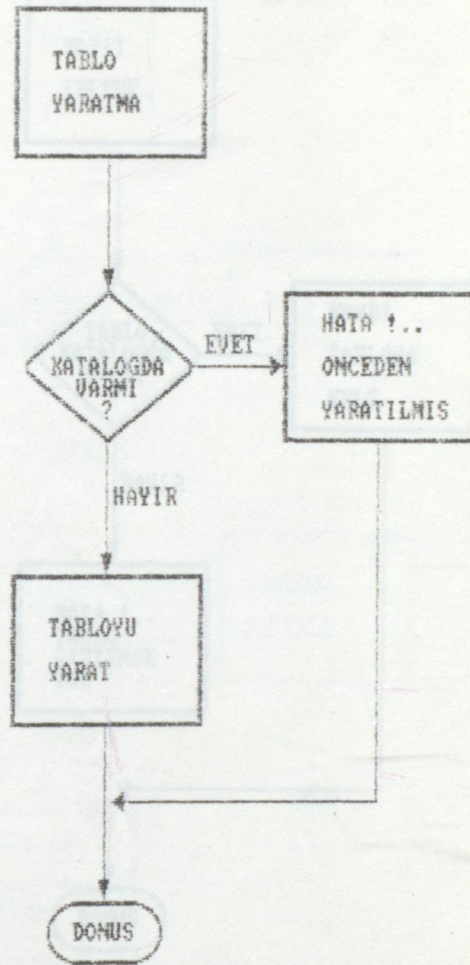
Tablolarda 'Unique Key' kavramı satır bazında değil bütün tablo bazında tanımlandığından eğer şartlı arama yapılacaksa bu durumun göz önünde bulundurulması gerekmektedir.Bu tip isteklerin gerçekleştirileceği tabloların ilk kolonlarının %/1,%/2,.... gibi artan sayıda ve bir daha tablo içinde tekrar olasılığı olmayan karakterlerden oluşan bir karakter alan olarak tariflenmesi önerilir.

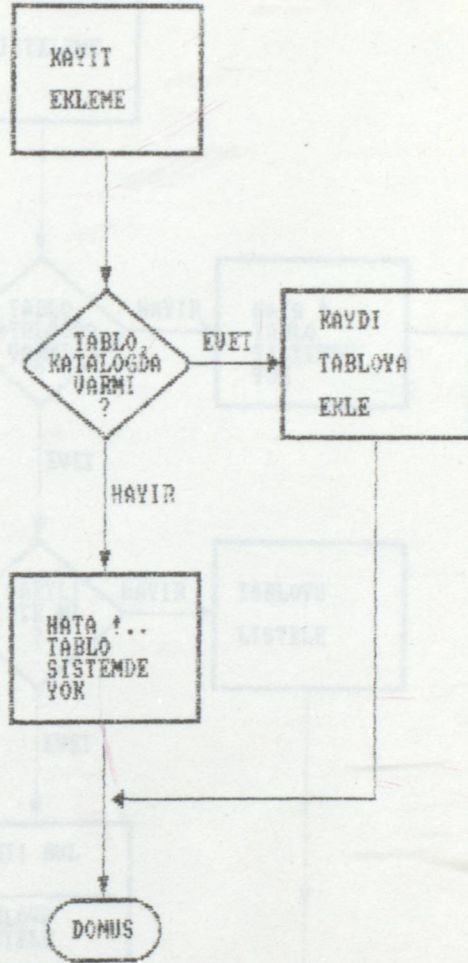
4.9.ÇALIŞTIRILMASI

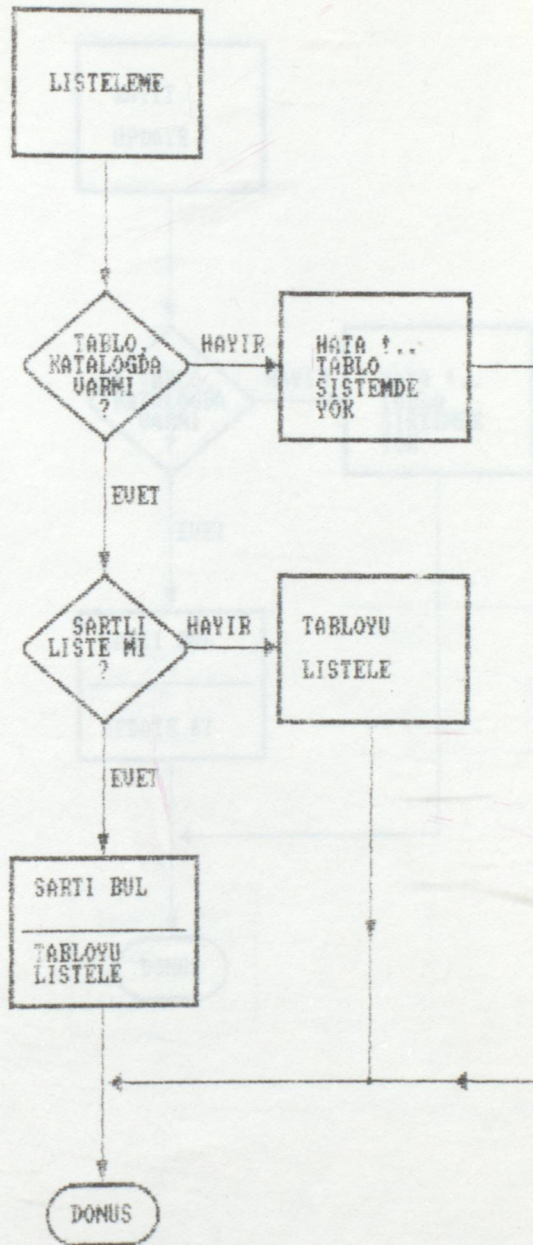
YLDZDB sisteminin çalıştırılması için MS/DOS ortamından YLDZDB komutunun girilmesi yeterlidir.Bu komutun girilmesi ile yazılımın oluştuğu menüler sistemi çalışmaya başlar.Bu aşamadan sonra her adım menu seçenekleri ve soru cevap şeklinde devam eder.

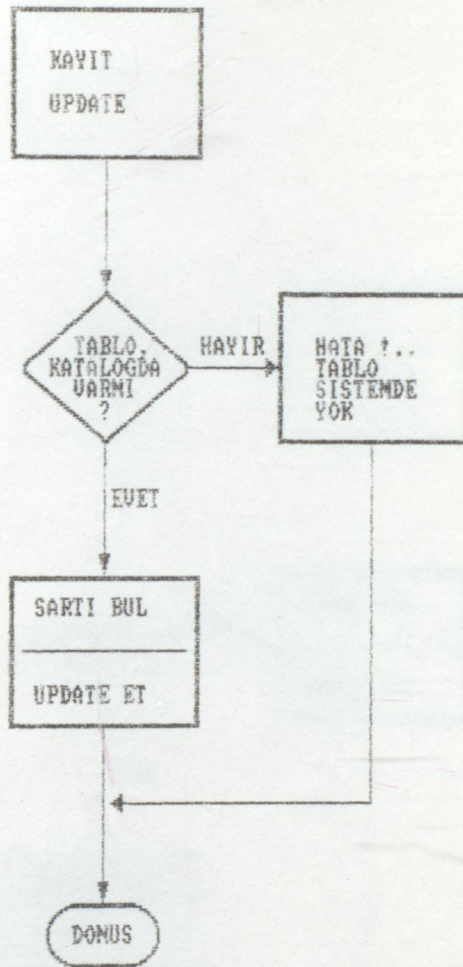


SERIL 4.1 : BLOK DIAGRAM





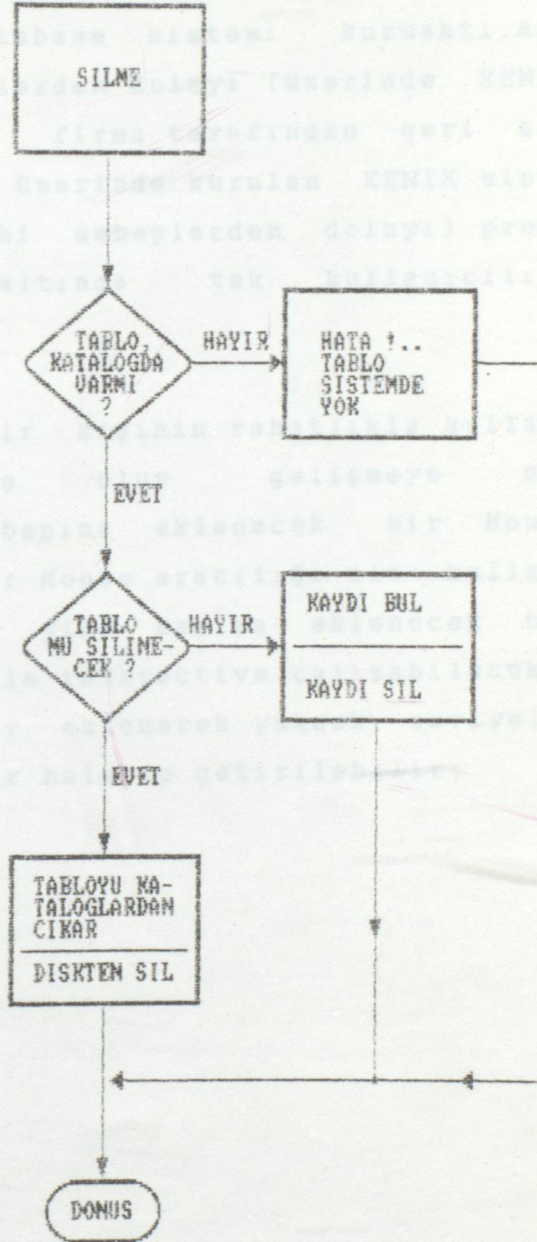




DÖNÜŞ

Bu projede önce XENIX işletim sistemi altında çok kullanıcıya sahip bir dosya sistemi kurulmuştur. Ancak elde olmayan teknik sebeplerden dolayı XENIX işletim sistemi bilgileri verilen firma tarafından yapılmış ve bu nedenle bilgileri burada kurulan XENIX sistemlerinin parçaları çalışması için çalışmadan dolayı proje MS/DOS işletim sistemi altında tek kullanıcıya sahip olarak yapılmıştır.

Sistem Yalnızca bir dosya sistemi olarak çalışabileceği şekilde tasarlanmıştır. Sistemin başlangıcında bir House Driver olarak 'i' ile başlayan bir dosya sistemi oluşturulmuştur. Bu dosya sistemi bir dosya sistemi olarak çalışır. Bir Compiler kullanılarak bu dosya sistemi için kullanılabilir hale getirilmiştir.



SONUÇ

Bu projede amaç XENIX işletim sistemi altında çok kullanıcılı bir database sistemi kurmaktır. Ancak elde olmayan teknik sebeplerden dolayı (üzerinde XENIX çalışan bilgisayarın verilen firma tarafından geri alınması , eldeki bilgisayarlar üzerinde kurulan XENIX sistemlerinin pürüzlü çalışması gibi sebeplerden dolayı) proje MS/DOS işletim sistemi altında tek kullanıcılı olarak uyarlanmıştır.

Sistem Türkçe bilen bir kişinin rahatlıkla kullanabileceği şekilde tasarlanmış olup gelişmeye açık bir yapıdadır. Programın başına eklenecek bir Mouse Driver Rutine 'i ile menüler Mouse aracılığı ile kullanılır hale gelebileceği gibi ; yine başına eklenecek bir Syntax Analyzer aracılığı ile interactive çalışabilecek bir hale getirilir. Bir Compiler eklenerek yüksek seviyeli dillerin içinden kullanılabilir hale de getirilebilir.

```

/* ANA PROGRAM..... */
#include <process.h>
#include <io.h>
#include <stdio.h>
#include <string.h>
char tus;
main()
{
    int secim;
    int sayac;
    do{
        clrscrn(30);
        secim=menu();
        switch(secim){
            case 0:exit(0);
            case 1:clrscrn(30);
                printf(" TABLO YARATMA YAPILIYOR...\n");
                clrscrn(10);
                yarat();
                break;
            case 2:clrscrn(30);
                printf(" KAYIT GIRISI YAPILIYOR...\n");
                clrscrn(10);
                ekle();
                break;
        }
    }
}

```




```

    case 3:clrscrn(30);
printf("    TABLO LİSTELEME YAPILYOR...\n");
printf("    clrscrn(10);
printf("    list();
printf("    break;
case 4:clrscrn(30);
printf("    KAYIT YENİLEME YAPILYOR...\n");
printf("    clrscrn(10);
printf("    yenile();
printf("    break;
case 5:clrscrn(30);
printf("TABLO / KAYIT SILME YAPILYOR...\n");
printf("    clrscrn(10);
printf("    sil();
printf("    break;
}
}while(1);
}
/* MENU ALTPROGRAMI...
ANA MENUYU EKRANA GETİRİR.
YAPILAN SECİMİ ANA PROGRAMA DONDURUR. */
menu()
{
char s[80];
int c;

```

```

printf("0 ==> CIKIS\n");
printf("1 ==> TABLO YARATMA\n");
printf("2 ==> KAYIT EKLEME\n");
printf("3 ==> TABLO LISTELEME\n");
printf("4 ==> KAYIT YENILEME\n");
printf("5 ==> TABLO-KAYIT SILME\n");
do{
    printf("\n SECIMINIZI GIRINIZ : ");
    gets(s);
    c=atoi(s);
}while(c<-1||c>5);
return c;
}
/* BELIRTILEN BASAMAK SAYISI KADAR DEGISKENE
DEGER OKUYAN ALTPROGRAM...
DONUS PARAMETRESI DEGISKENIN DEGRIDIR. */

inputs(prompt,s,count)
char *prompt;
char *s;
int count;
{
    char p[255];
    do{
        printf(prompt);
        gets(p);
    }
}

```



```

    if(strlen(p)>count) {
        printf("\n    UZUN VERI GIRDINIZ... \n");
    }
}while(strlen(p)>count);

strcpy(s,p);
}

/*  TABLO YARATMAYI YAPAN ALTPROGRAM...
    KATALOGLARI INCELEYEREK YENI BIR
    TABLO YARATILIP YARATILMAMASINA IZIN VERIR.  */

yarat()
{
FILE *pp;
FILE *fp;

char tname[255];
char ch,tname[8];
char colname[255];
char coltype[255];
char colc[3];
char coll[255];
int collen;
int m,colcount,t;

inputs("  TABLO ISMINI VERINIZ : ",tname,8);
err0();
pp=fopen("SYSCAT","r");
fscanf(pp,"%s\n",tname);

```

```

ch=getc(pp);
while(ch!=EOF){
    if(strcmp(ttname,tname)==0){
        if((fp=fopen(tname,"r"))!=0){
            printf("    %s ISIMLI TABLO HALEN TANIMLIDIR...\n",tname);
            don();
            return(1);
        }
        printf("    SYSCAT KATALOGU ILE DISK ARASINDA UYUSMAZLIK...
            SYSTEM ERROR\n");
        don();
        return(2);
    }
    ch=getc(pp);
    fscanf(pp,"%s\n",ttname);
}
fclose(pp);
inputs("    ALAN SAYISINI VERINIZ : ",colc,3);
colcount=atoi(colc);
do{
    if(colcount<=0||colcount>255){
        printf("\n VERECEGINIZ DEGER 0 ILE 255 ARASINDA OLABILIR...\n");
        inputs("    ALAN SAYISINI VERINIZ : ",colc,3);
        colcount=atoi(colc);
    }
}

```



```

}while(colcount>255);
err1();
fp=fopen("SYSCOL","a");
err2();
pp=fopen("SYSCAT","a");
fprintf(pp,"%s %d \n",tname,colcount);
fprintf(fp,"%s ",tname);
printf("  ALANLARININ ISMINI TIPINI UZUNLUGUNU VERINIZ...\n");
for(t=1;t<=colcount;++t){
    scanf("%s %s %s\n",colname,coltype,coll);
    collen=atoi(coll);
    fprintf(fp,"%s %s %d ",colname,coltype,collen);
}
fprintf(fp,"\n");
fclose(pp);
fclose(fp);
if((fp=fopen(tname,"w"))==0){
    printf("  %s TABLOSU YARATILAMADI...",tname);
    don();
    return(1);
}
fprintf(fp,"\n");
fclose(fp);
return;

```

```

/* BIR TABLOYA BIR KAYIT EKLEME
   ALTPROGRAMI...
   SISTEM KATALOGLARINA GORE
   IZIN VERIR.
*/

```

```

sayac=indiri("SYSCOL",D,tname);

```

```

ekle()
{

```

```

    printf("   %s ISIMLI TABLONUN KOLONLARI SISTEM KATALOGUNDA
    YOK...\n",tname);

```

```

    FILE *fp;

```

```

    FILE *pp;

```

```

    char tname[8];

```

```

    int colcount;

```

```

    char colname[8];

```

```

    int collen;

```

```

    char deger[255];

```

```

    int ddeg;

```

```

    char coltype[9];

```

```

    int i;

```

```

    char dummy[255];

```

```

    char ch;

```

```

    int sayac;

```

```

    inputs("   TABLO ISMINI VERINIZ : ",tname,8);

```

```

    sayac=baslik(tname);

```

```

    if(sayac==0){

```

```

        printf("   %s ISIMLI TABLO SISTEM KATALOGUNDA KAYITLI
        DEGIL...\n",tname);

```

```

    don();

```

```

    return;
}

```



```
}  
colcount=sayac;  
sayac=findin("SYSCOL",0,tname);  
if(sayac<=1){  
    printf("    %s ISIMLI TABLONUN KOLONLARI SISTEM KATALOGUNDA  
        YOK...\n",tname);  
    don();  
    return;  
}  
if((pp=fopen(tname,"a"))==0){  
    printf("    %s ISIMLI TABLO BOZUK...\n",tname);  
    don();  
    return;  
}  
if((fp=fopen("SYSCOL","r"))==0){  
    printf("    KATALOG HATASI....\n");  
    exit;  
}  
for(i=1;i<sayac;++i){  
    ch=fgetc(fp);  
    while(ch!='\n'){  
        ch=fgetc(fp);  
    }  
}  
fscanf(fp,"%s ",dummy);  
for(i=1;i<=colcount;++i){
```

```

char dummy[8];

fscanf(fp,"%s %s %d ",colname,coltype,&collen);
inputs(colname,deger,collen);
if(strcmp(coltype,"int")==0){
    ddeg=atoi(deger);
    fprintf(pp,"%d ",ddeg);
}
else{
    fprintf(pp,"%s ",deger);
}
}

fprintf(pp,"\n");
fclose(fp);
fclose(pp);
return;
}

/* KAYIT YA DA TABLO SILEN ALTPROGRAM...
KATALOG INCELEMESINDEN SONRA SILMEYE
IZIN VERIR. */

sil()
{
FILE *fp;
FILE *pp;
char tname[8];
char *rtname;
int i,k;

```



```

char dummy[8];
char tname[8];
int sayac;
int colcount;
int say,count[255];
int *p,secim;
char s[80];
int c1,j,collen;
char colname[8],coltype[3];
char ch,c,deger[255];
printf("                                0 ==> ANA MENUYE DONUS\n");
printf("                                1 ==> TABLO SILNEK\n");
printf("                                2 ==> SATIR SILNEK\n");
do{
    printf("\n  SECIMINIZI GIRINIZ  :");
    gets(s);
    c1=atoi(s);
}while(c1<-1||c1>2);
switch(c1){
    case 0:return;
    case 1:inputs("  SILINECEK TABLONUN ISMINI VERIN :",tname,8);
        sayac=findin("SYSCAT",0,tname);
        if(sayac==0){
            printf("  %s ISIMLI TABLO SYSTEM CATALOGUNDA TANIMLI
            DEGIL,...\n",tname);

```

```

    don();
    return;
}
err0();
fp=fopen("SYSCAT","r");
if((pp=fopen("SYSCAT.WRK","w"))==0){
    printf("  GIDERILEMEYECEK HATA...\n");
    don();
    return;
}
p=count;for(i=0;i<255;++i) *p++=0;
p=count;k=0;
fprintf(pp,"\n");
for(i=2;i<sayac;++i){
    fscanf(fp,"%s %d",ttname,&colcount);
    fprintf(pp,"%s %d \n",ttname,colcount);
    *p++=colcount;
    k++;
}
fscanf(fp,"%s %d",ttname,&colcount);
*p++=colcount;
++i;
++k;
while(!feof(fp)){
    fscanf(fp,"%s %d",ttname,&colcount);

```



```
for(i=2; i<=sayac; i++)  
  
    if(!feof(fp)){  
        fprintf(pp,"%s %d \n",ttname,colcount);  
        *p++=colcount;  
        ++i;  
        ++k;  
    }  
}  
  
fclose(fp);  
fclose(pp);  
unlink("SYSCAT");  
rename("SYSCAT","SYSCAT.WRK");  
sayac=findin("SYSCOL",0,tname);  
if(sayac<=1){  
    printf("  KATALOG UYUSMAZLIGI !.....\n");  
    don();  
    return;  
}  
  
err1();  
fp=fopen("SYSCOL","r");  
if((pp=fopen("SYSCOL.WRK","w"))==0){  
    printf("  GIDERILEMEYECEK HATA\n");  
    don();  
    return;  
}  
  
fprintf(pp,"\n");k=0;
```

```

for(i=2;i<sayac;++i){
    fscanf(fp,"%s ",ttname);
    fprintf(pp,"%s ",ttname);
    for(j=1;j<=count[i-2];++j){
        fscanf(fp,"%s %s %d",colname,coltype,&collen);
        fprintf(pp,"%s %s %d ",colname,coltype,collen);
    }
    fprintf(pp,"\n");k++;
}
fscanf(fp,"%s ",ttname);
for(j=1;j<=count[i-2];++j){
    fscanf(fp,"%s %s %d ",colname,coltype,&collen);
}
++i;k++;
while(!feof(fp)){
    fscanf(fp,"%s ",ttname);
    if(!feof(fp)){
        fprintf(pp,"%s ",ttname);
        for(j=1;j<=count[i-2];++j){
            fscanf(fp,"%s %s %d ",colname,coltype,&collen);
            fprintf(pp,"%s %s %d ",colname,coltype,collen);
        }
    }
    fprintf(pp,"\n");i++;
}

```



```

fclose(fp);
fclose(pp);
unlink("SYSCOL");
rename("SYSCOL","SYSCOL.WRK");
unlink(tname);
break;
case 2:inputs("  TABLO ISMINI VERINIZ : ",tname,8);
inputs("  KAYDIN BELIRLEYICI OZELLIGINI VERINIZ : ",deger,25);

err3(tname);
fp=fopen(tname,"r");
if((pp=fopen("WORK.WRK","w"))==0){
printf("  GIDERILEMEYECEK HATA...\n");
don();
return;
}
fprintf(pp,"\n");
k=findin("SYSCAT",0,tname);
if(k<=1){
printf("  %s ISIMLI TABLO SYSCAT KATALOGUNDA
YOK...\n",tname);
don();
return;
}
say=findin(tname,0,deger);
if(say<=1){

```

```

    printf(" SARTA UYGUN KAYIT BULUNAMADI...\n");
    don();
    return;
}

k=baslik(tname);
if(k==0) return;
for(j=2;j<say;++j){
    for(i=1;i<=k;++i){
        fscanf(fp,"%s ",deger);
        fprintf(pp,"%s ",deger);
    }
    fprintf(pp,"\n");
}
for(i=1;i<=k;++i){
    fscanf(fp,"%s ",deger);
}
while(!feof(fp)){
    for(i=1;i<=k;++i){
        fscanf(fp,"%s ",deger);
        fprintf(pp,"%s ",deger);
    }
    fprintf(pp,"\n");
}
fclose(fp);
fclose(pp);

```



```

        unlink(tname);
        rename(tname,"WORK.WRK");
    if (sayac1==0)
        break;
    }
}
return;
}

/* SARTLI YA DA SARTSIZ TABLO
   LİSTELEMESİ YAPAN ALTPROGRAM...
   KATALOG İNCELEMESİNDEN SONRA
   KARAR VERİR. */

list()
{
    FILE *pp;
    FILE *fp;
    char ara[8],deger[20],tname[8];
    char pcc[3],dd[8],yaz[255],scim[3];
    int secim;
    int flag,colcount,n,count[255],*p,sayac1,sayac,i,collen;
    char nn[3],dummy,pc,ch,ttname[8],colname[8],coltype[3];
    inputs(" TABLO İSMİNİ GIRINIZ : ",tname,8);
    sayac=baslik(tname);
    if(sayac==0) return;
    err1();
    p=count;
    fp=fopen("SYSCOL","r");
    sayac1=findin("SYSCOL",0,tname);

```

```

if(sayac1<=1){
    printf("    KATALOG UYUSMAZLIGI....\n");
    don();
}
return;
}
for(i=1;i<sayac1;++i){
    ch=fgetc(fp);
    while(ch!='\n'){
        ch=fgetc(fp);
    }
}
if((pp=fopen("PRT.PRT","w"))==0){
    printf("    GIDERILEMEYECEK HATA ... \n");
    don();
    return;
}
inputs("    KOLON UZERINDEN ARAMA YAPMAK ISTERSENIZ 1
        YAZIN : ",scim,1);
secim=atoi(scim);
if(secim==1){
    colcount=baslik(tname);
    inputs("    KACINCI KOLON UZERINDEN ARAMA YAPILACAK : ",nn,3);
    n=atoi(nn);
    if(n>colcount){
        printf("    VERILEBILECEK MAX COLON SAYISI %d OLABILIR\n",
            colcount);
    }
}

```



```

flag=0;return;
}
}
inputs(" ARADIGINIZ DEGERI VERINIZ : ",dd,8);
}
for(i=1;i<=sayac;++i)
fscanf(fp,"%s ",tname);
for(i=1;i<=sayac;++i){
fscanf(fp,"%s %s %d ",colname,coltype,&collen);
*p++=collen;
if((strlen(colname))>collen){
strncpy(ara,colname,collen);
ara[collen]=0;
strncpy(colname,ara,collen);
colname[collen]=0;
strncpy(ara," ",collen);
ara[collen]=0;
}
fprintf(pp,"%*s ",collen,colname);
printf("%*s ",collen,colname);
}
printf("\n");
fprintf(pp,"\n");
fclose(fp);
err3(tname);
fp=fopen(tname,"r");
while(!feof(fp)){

```

```

flag=0;
strcpy(yaz," ");
yaz[0]=0;
for(i=1;i<=sayac;++i){
    fscanf(fp,"%s ",deger);
    if(secim==1 && i==n && (strcmp(dd,deger)==0)) flag=1;
    sprintf(&yaz[strlen(yaz)],"%s ",count[i-1],deger);
    printf("%d count[i-1],i degeri %d\n",count[i-1],i);
    printf("%s %s deger ve yaz in degeri\n",deger,yaz);
    if(secim!=1){
        printf("%s ",count[i-1],deger);
        fprintf(pp,"%s ",count[i-1],deger);
    }
}
if(flag==1){
    printf("%s \n",yaz);
    fprintf(pp,"%s \n",yaz);
}
else{
    fprintf(pp," \n");
    printf("\n");
}
}
inputs(" LİSTEYİ YAZICIDAN ALMAK İSTERSENİZ 1 YAZIN : ",pcc,1);
pc=atoi(pcc);

```



```

if(pc==1){
    inputs("YAZICININ HAZIR OLDUGUNDAN EMIN OLUNCA ENTER A BASINIZ ",
    dummy,1);

    system("TYPE PRT.PRT>PRN");
}

return;

}

/* KAYIT UPDATE ALTPROGRAMI....
   KATALOG INCELEMESINDEN
   SONRA KARAR VERIR.          */

yenile()
{
    FILE *fp;
    FILE *pp;
    char tname[8],deger[255],dd[255];
    int say,i,j,k;
    inputs(" TABLO ISMINI VERINIZ : ",tname,8);
    inputs(" KAYDIN BELIRLEYICI OZELLIGINI VERINIZ : ",deger,255);
    err3(tname);
    fp=fopen(tname,"r");
    if((pp=fopen("WORK.WRK","w"))==0){
        printf(" GIDERILEMEYECEK HATA...\n");
        don();
        return;
    }
    fprintf(pp,"\n");
}

```

```

if(strcmp(dp,"?")!=0){
k=findin("SYSCAT",0,tname);
if(k<=1){
printf("  %s ISIMLI TABLO SYSCAT KATALÖGÜNDA YOK...\n",tname);
don();
return;
}
say=findin(tname,0,deger);
if(say<=1){
printf("  SARTA UYGUN KAYIT BULUNAMADI...\n");
don();
return;
}
k=baslik(tname);
if(k==0) return;
for(j=2;j<say;++j){
for(i=1;i<=k;++i){
fscanf(fp,"%s ",deger);
fprintf(pp,"%s ",deger);
}
fprintf(pp,"\n");
}
for(i=1;i<=k;++i){
fscanf(fp,"%s ",deger);
printf("%s ",deger);
inputs("  YENI DEGERINI YAZIN YA DA / A BASIN : ",dd,255);
}
}

```



```

    if(strcmp(dd,"/")!=0){
        char tname[8];
        strcpy(deger,dd);
        colcount=0;
    }
    scr2();
}
sayac1=findin("SYSCAT",0,tname);
fprintf(pp,"\n");
if(sayac1!=1){
while(!feof(fp)){
    printf("  ISIRILI TABLO CATALOG SA YOK...\n",tname);
    for(i=1;i<=k;++i){
        colcount=0;
        fscanf(fp,"%s ",deger);
        fclose(fp);
        fprintf(pp,"%s ",deger);
        return colcount;
    }

    fprintf("\n");
}
for(i=2;i<=sayac1;i++){
fclose(fp);
fscanf(fp,"%s %d ",tname,&colcount);
fclose(pp);

unlink(tname);
rename(tname,"WORK.WRK");
return;
}

/* VERILEN TABLONUN KOLON
   SAYISINI BULAN ALTPROGRAM... */
DELIRILEN KALIMAYI ARAYAN
ALTPROGRAM...

baslik(tname)
char tname[8];
{
    FILE *fp;
    int i,colcount,sayac1;

```

```

char tname[8],ch;
colcount=0;
err2();
sayac1=findin("SYSCAT",0,tname);
if(sayac1<=1){
    printf("    %s ISIMLI TABLO CATALOG DA YOK...\n",tname);
    colcount=0;
    fclose(fp);
    return colcount;
}

fp=fopen("SYSCAT","r");
for(i=2;i<sayac1;++i){
    fscanf(fp,"%s %d ",tname,&colcount);
}

fscanf(fp,"%s %d ",tname,&colcount);
fclose(fp);

return colcount;
}

/*  VERILEN TABLONUN ICINDE
    BELIRTILEN KELIMEYI ARAYAN
    ALTPROGRAM... */

findin(tname,offset,word)

int offset;

char tname[8];

char *word;

```



```
{  
FILE *pp;  
int k,j,i,sayac;  
char dummy[80];  
char ch;  
err3(tname);  
pp=fopen(tname,"r");  
sayac=1;  
i=0;  
fseek(pp,offset,0);  
for(j=0;j<81;j++) dummy[j]='\0';  
ch=fgetc(pp);  
while(ch!=EOF){  
    if(ch=='\n'){  
        sayac++;  
        goto don;  
    }  
    if(ch!=' '){  
        dummy[i]=ch;  
        i++;  
        goto don;  
    }  
    i=0;  
    if(strcmp(word,dummy)==0){  
        fclose(pp);
```

```

return sayac;
}

for(j=0;j<81;j++) dummy[j]='\0';
don: ALTPROGRAM...

    ch=fgetc(pp);
}

fclose(pp);
sayac=0;
return sayac;
}
exit(2);
/* VERILEN SAYI KADAR BOS
SATIR YAZAN ALTPROGRAM... */
fclose(pp);
clrscrn(count)

int count;
/* SYSOL KATALOGU) EKLEME
( AMACI ILE ACHAYA CALISAN
ALTPROGRAM... */
int i;
for(i=1;i<=count;i++){
    printf("\n");
}
}
/* ANA MENUYE DONUS EKRANINI
GETIREN ALTPROGRAM... */
don()

char tus;

```



```

inputs(" ANA MENUYE DONMEK ICIN HERHANGIBIR TUSA BASINIZ...", &bus,
/* SYSCAT KATALOGUNU EKLEMEK
return; ILE ACMAYI DENEYEN
ALTPROGRAM... */
}

/* SYSCAT KATALOGUNU OKUMAK ICIN
ACIP ACAMAYACAGINI KONTROL
EDEN ALTPROGRAM... */

FILE *pp;
err0()
{
if((pp=fopen("SYSCAT", "a"))==0)
FILE *pp;
if((pp=fopen("SYSCAT", "r"))==0){
printf(" SYSCAT KATALOGU BOZUK...SYSTEM ERROR\n");
fclose(pp);
exit(2);
}
fclose(pp);
}

/* SYSCOL KATALOGUNU EKLEME
AMACI ILE ACMAYA CALISAN
ALTPROGRAM... */

err1()
{
FILE *fp;
if((fp=fopen("SYSCOL", "a"))==0){
printf(" SYSCOL KATALOGU BOZUK...SYSTEM ERROR\n");
exit(2);
}
fclose(fp);
}

```

```
}

/* SYSCAT KATALOGUNU EKLEMEK
   AMACI ILE ACMAYI DENEYEN
   ALTPROGRAM... */

main()
err2()
{
    FILE *pp;
    if((pp=fopen("SYSCAT","a"))==0){
        printf(" SYSCAT KATALOGU BOZUK...SYSTEM ERROR\n");
        exit(2);
    }
    fclose(pp);
}

/* VERILEN TABLOYU OKUMA AMACI
   ILE ACMAYI DENEYEN ALTPROGRAM... */

err3(tname)
char tname[8];
{
    FILE *tp;
    if((tp=fopen(tname,"r"))==0){
        printf(" %s ISIMLI TABLO ACILAMADI...FILE OPEN ERROR\n");
        exit(2);
    }
    fclose(tp);
}
```



```
#include <stdio.h>
```

```
main()
```

```
{
```

```
FILE *pp;
```

```
FILE *fp;
```

```
if((pp=fopen("SYSCOL","a"))!=0){
```

```
    fprintf(pp,"\n");
```

```
    fclose(pp);
```

```
}
```

```
if((fp=fopen("SYSCAT","a"))!=0){
```

```
    fprintf(fp,"\n");
```

```
    fclose(fp);
```

```
}
```

```
}
```

ECHO OFF

ECHO " SİSTEM KATALOGLARI RESTORE EDİLİYOR...."

ECHO ON

copy syscat.bac syscat

ECHO OFF

ECHO " SİSTEM KATALOGLARININ BACKUP İ ALINIYOR...."

copy syscat syscat.bac

copy syscol syscol.bac

ECHO " BACKUP İSLEMI SONA ERMİSTİR...."

ECHO ON.


```
ECHO OFF
```

```
ECHO "   SISTEM KATALOGLARI RESTORE EDILİYOR...."
```

```
copy syscat.bac syscat
```

```
copy syscol.bac syscol
```

```
ECHO "   RESTORE İSLEMİ SONA ERMİŞTİR..."
```

```
ECHO ON
```

```
int prt;
```

```
char ar;
```

```
printf("LİSTELERİ YATICIDAN İSTİYORSANIZ İ YATINIZ...");
```

```
scanf("%s",prt);
```

```
if(prt=="1")
```

```
printf("YATICININ HAZIR OLDUGUNDAN EMİN OLUNCA SİZ TUGA...");
```

```
scanf("%c",&ar);
```

```
system("type syscat.pcn");
```

```
system("type syscol.pcn");
```

```
printf("SYSCAT KATALOGUNU");
```

```
system("type syscat");
```

```
printf("SİZ SYSCOL KATALOGUNU");
```

```
system("type syscol");
```

```
#include <stdio.h>

#include <process.h>

main()
{
    int prt;
    char a;

    printf("LISTELERI YAZICIDAN ISTIYORSANIZ 1 YAZINIZ...\n");

    scanf("%d",&prt);

    if(prt==1){
        printf("YAZICININ HAZIR OLDUGUNDAN EMIN OLUNCA BIR TUSA BASINIZ.");

        scanf("%c",&a);

        system("type syscat>prn");

        system("type syscol>prn");
    }

    printf("\n SYSCAT KATALOGU\n");

    system("type syscat");

    printf("\n SYSCOL KATALOGU\n");

    system("type syscol");
}
```


KAYNAKLAR

- An Introduction to Database Systems
C.J.DATE

- SQL/DS Database Administrator Guide
IBM Product

VİYS : Veri Tabanı Yönetim Sistemi

- C Made Easy İslam Nekeci
McGraw Hill
H.Schildt Tabanı

- Veri Tabanının Zamani (Makale)
Hülya Polat

FVT : Fiziksel Veri Tabanı

DDL : Data Definition Language

DML : Data Manipulation

VM : Veri Modeli

SÖZLÜK

DBMS : DataBase Management System

VTYS : Veri Tabanı Yönetim Sistemi

BİM : Bilgi İşlem Merkezi

VT : Veri Tabanı

MVT : Mantıksal Veri Tabanı

FVT : Fiziksel Veri Tabanı

DDL : Data Definition Language

DML : Data Manipulation

VM : Veri Modeli

ÖZGEÇMİŞ

Ü.Gülsüm YAKUT DEMİRBAŞ, 1966 yılında İstanbul'da doğdu.Liseyi G.O.P. Plevne Lisesi'nde bitirdi.1986'da Yıldız Üniversitesi Mühendislik Fakültesi Bilgisayar Bilimleri Mühendisliği Bölümünden mezun oldu. 1987 yılında evlendi.1988 yılında çocuk sahibi oldu. — Halen özel bir kuruluştta Database Administrator olarak çalışmaktadır.

