

YILDIZ TEKNİK ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

Koşu, Kalp Has. Bil. Sls. Tas. ve Uyg.

YÜKSEK LİSANS TEZİ

Şebnem Tokcan

1989

YILDIZ TEKNİK ÜNİVERSİTESİ
KÜTÜPHANE DOKÜMANTASYON
DAİRE BAŞKANLIĞI

R 368

50

Kot	:	
Alındığı Yer	:	Fen Bil. Ens.
Tarih	:	13.4.95
Fatura	:	
Fiyatı	:	10.800.
Ayniyat No	:	1-4
Kayıt No	:	51028
UDC	:	
Ek	:	

YILDIZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Yılların en önemli gelişmelerinin yapıldığı alanlardan biri de bilgisayarlardır. Bu alanda her geçen yıl insanlığın umurlarında bir yük daha eksilmekte, aklı ve gücü çok daha önemli konulara ve araştırmalara yönelmektedir. Zira bu minimum zaman ve insan gücü demektir.

Bu gereksinim özellikle tıbbi ilgilendiren konularda önemli bir yer tutmaktadır. Özellikle kalp hastalıkları ve damar tıkanıklıkları gibi hastalıkların tedavisi için bilgisayarlı sistemler kullanılmaktadır. Bu sistemler, hastanın durumunu sürekli olarak izleyerek, doktorun emriyle ilaçları otomatik olarak vermektedir. Bu sistemlerin arttırılması ve son yıllarda araştırmaların eline daha kesin ve detaylı bilgiler bırakarak ameliyatı kolaylaştırması.

Çalışmamın ilk aşaması 1988'de yapılan 6. ULUSAL KARDİYOLOJİ Kongresi'nde sunulmuş ve bu çalışmamın ilgisiz konularla ilgili olarak devam etmektedir.

Bu çalışmamın amacı, kalp hastalıkları ile ilgili olarak Prof. Dr. Yahya Kaya ile birlikte araştırma görevlisi Sedat Alar'a ve öğretim görevlisi Dr. Şebnem Tokcan'a bir burs bütçesi.

YÜKSEK LİSANS TEZİ

YILDIZ ÜNİVERSİTESİ BİLGİSAYAR MÜH. ŞEBNEM TOKCAN
İSTANBUL, 1989

İSTANBUL, 1989

ÖNSÖZ

İÇİNDEKİLER

Yüzyılımızın en önemli gelişmelerinin yapıldığı alanlardan biri de bilgisayarlardır. Bu sayede her geçen yıl insanlığın omuzlarından bir yük daha eksilmekte , aklını ve gücünü çok daha önemli konulara ve araştırmalara yöneltebilmektedir ki bu minimum zaman ve insan gücü demektir.

Bu gereksinim özellikle tıbbi ilgilendiren konularda önemli bir yer tutmaktadır. Yapılan çalışmalar da hasta-doktor ilişkisinde önemli bir etken olan zaman faktörünü minimum olduğu kadar düşürmek , böylece tedavi ve bakım hizmetlerinin arttırmak ve sonraki yıllarda araştırmacıların eline daha kesin ve detaylı bilgiler bırakmak amacını gütmüştür.

Çalışma ilk aşamasını 23-26 EKİM 1988'de yapılan 6. ULUSAL KARDİYOLOJİ kongresinde bildiri olarak seçilerek sunulması ile tamamlamış olup çalışmalar halen devam etmektedir.

Bu çalışmamda değerli görüşleri ile bana yön gösteren sayın Prof. M. Yahya KARSLIGİL'e ve araştırma görevlisi Sedat Alış'a ve emeği geçen tüm arkadaşlarıma teşekkürü bir borç bilirim.

YILDIZ ÜNİVERSİTESİ

ŞEBNEM TOKCAN

İSTANBUL, 1989

İÇİNDEKİLER

Özet

Abstract

1. Tıpta bilgisayar

1.1. Giriş	1
1.2. Tarihçe	3

2. Veri tabanı oluşturma

2.1. Tanım	5
2.2. Veri tabanı çevrimi	9
2.2.1 Koşuyolu kalp hastanesi mevcut sistem tanıtımı	
2.2.1.1 Giriş	10
2.2.1.2 Teknik özellikler	14
2.3 Sistem tasarımı	16

3. DOS ve AIX işletim sistemlerinde tasarımın gerçekleştirilmesi

3.1. Dbase III plus	21
3.2. DBMS yönetimi	23

Sonuç

ÖZET

PROGRAMLAR

datamng.h	pr1
mpanel.c	pr 2
hastane.c.....	pr
dökümler	dök1

Kaynaklar

Özgeçmiş

ÖZET

ABSTRACT

Bu projenin hazırlanmasındaki ana hedef hastanelerdeki hasta-doktor ilişkisini , değişik birimlerden gelen hasta verilerinin hatasız ve tam olarak tutulması ile daha verimli kılmanın yanısıra , araştırmaya yönelik çalışan doktorların da çeşitli istatistiki sonuçlara ulaşmasında eldeki veriler üzerinde çok daha az arama yaparak sonuca daha kısa zamanda ve tam olarak ulaşmasını sağlamaktır. Verilerin düzenli ve paylaşımlı olarak tutulması ileride hasta teşhisi ile ilgili çalışmalarında temel kaynağını oluşturacaktır.

Çalışma iki adımda gerçekleştirilmiştir. Birinci adım verilerin her değişik çalışma ortamı için rahatça toplanıp saklanmasına uygun , gerektiğinde artan birim ve bu birimlerin getireceği verileri karşılayabilecek şekilde bir sistemin tasarımıdır. İkinci adım ise seçilen sistemde uygun veri tabanı organizasyonunun oluşturulmasıdır.

Bu organizasyon ise önce dbase III plus daha sonra da IBM 6150 model 125 'de AIX işletim sistemi altında çalışan DBMS veri tabanı dilleri kullanılmıştır.

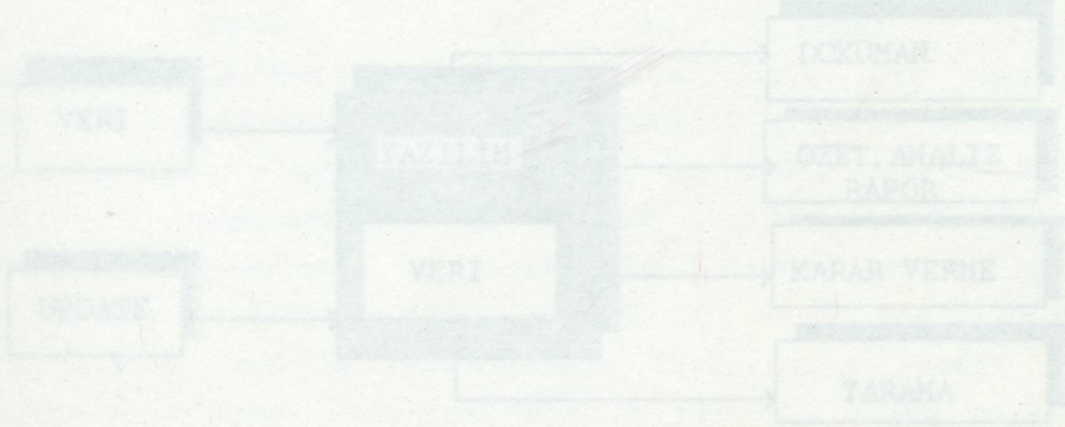
ABSTRACT

The aim of preparing this project is to make relationship between the doctors and his patients more productivity by keeping the data of the operations redundancy which are coming from the different parts of hospital. Also this people helps the doctors who are making research to reach the result of statistics without mentioning details in a very short time and completely.

It is very important to make diagnosis by keeping data in a proper and sharable situation.

This project has been completed with two steps. First step include a system design which is suitable for excepting and storing a data comes from different hospital and a different unit of hospital and even expanding the system. The second step include data-base design and maintenance. While creating the data-base dbase III plus and DBMS which is used in a special data-base management system in a AIX operating system of IBM 6150 model 125 are used.

koymaktadır. Aynı yanı sıra elde edilen sonucu hasta hakkında cümle bilgi verebilecek ve bu bilgi üzerinden değişik uygulamalarla istenen istatistikî değerlere götürebilecek bir rapor türünde olması gerekmektedir. Özetle şekil (1.1) de görüldüğü gibi hasta verilerinden faydalanılarak doküman, rapor türünde bilgilere ulaşmak mümkündür.



Şekil (1.1)

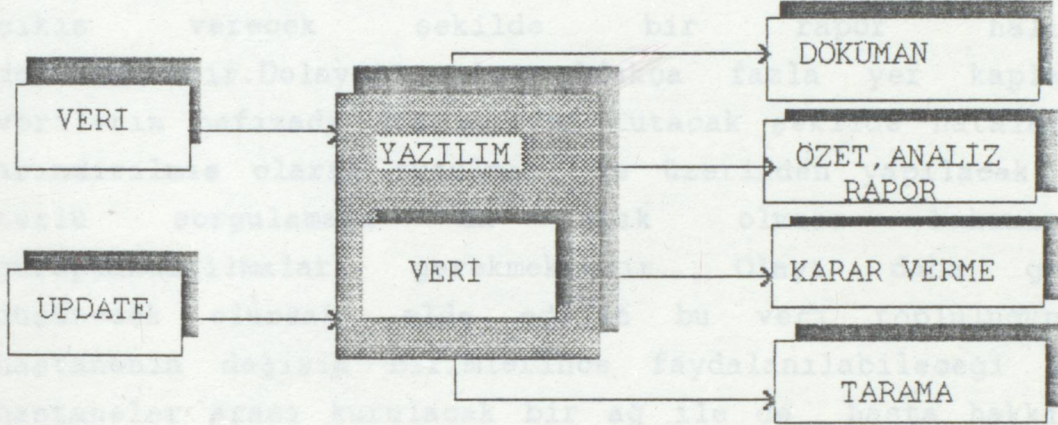
1- TIPTA BILGISAYAR

1.1- Giriş

Günümüzde bilgisayarların kullanımının en yaygın olduğu alanlardan biri de tıp alanıdır. İnsan sağlığının :

- 1- Zamanında müdahale,
- 2- Doğru teşhis ,
- 3- İyi bakım gibi hizmetlere bağlı olması ve bu hizmetlerin yerine getirilmesinde hastadan toplanan verilerin büyük rol oynaması **bilgisayar** kullanımına ihtiyaç doğurmaktadır. Yapılan araştırmalar da iyi bir teşhisin %85'inin hastanın hikayesine bağlı olduğunu ortaya koymakla bu fikri doğrulamaktadır. Geri kalan %15 ise teşhise gidilmede toplanan verilerin doktorun bilgi dağarcığı ve tecrübesi ile desteklenmesi gerektiğini ortaya

koymaktadır. Bunun yanı sıra elde edilen sonucun hasta hakkında özetle bilgi verebilecek ve bu bilgi üzerinden değişik sorgulamalarla istenen istatistikî değerlere götürebilecek bir rapor türünde olması gerekmektedir. Özetle şekil (1.1)'de de görüldüğü gibi hasta verilerinden faydalanılarak döküman , rapor türünde bilgilere ulaşmak mümkündür.



Şekil (1.1)

Şekilde görülen veri topluluğu hastanın başvurusundan çıkışına kadar geçen sürede değişik birimlerden gelen sorgulamalar sonucunda elde edilmektedir. Bu verileri guruplandırarak olursak :

1-Hastanın şikayeti:Hastanın başvurusunda doldurulan hastalığın başlangıç ve seyrini içeren *Fizik ve Anamnez* başlığı altındaki bir formdan ibarettir.

2-Şikayetlerin kronolojik detayları:Hastalığın semptomlarının başlangıç tarihi , sürme şekli ve süresi gibi daha detaylı bilgileri içeren , yine *Fizik ve Anamnez* başlığı altındaki formda kullanılan bir bilgi türüdür.

3-EKG,EKO,Laboratuar bulguları:Eforlu olarak alınan elektrokardiogram (Hastanın bisiklet çevirerek veya herhangi bir şekilde kalbin atışını hızlandıracak hareketler yaparak ölçümlerinin alınması) sonuçları ve KAN , BIYOKİMYA , IDRAR gibi alınan tahlil sonuçlarının *LABORATUAR* başlıklı bir form altında toplanmış şeklidir.

4-Yapılmış ise ameliyat bulguları:Gerekli görüldüğü

takdirde hasta ROMATİZMAL AÇIK KALP CERRAHİSİ ya da DİĞER AMELİYAT türü ile tedavi edilir. Eğer hasta bu tedaviye cevap vermezse veri toplama işlemine geri döner.

Böylece elde edilen verilerin çıkışı, *EPIKRİZ* dediğimiz hastaya ait özet bilgileri içerecek şekilde ya da bu bilgiler üzerinden her türlü sorgulamaya istenen formda çıkış verecek şekilde bir rapor halinde dökülmektedir. Dolayısı ile oldukça fazla yer kaplayan verilerin hafızada çok az yer tutacak şekilde hatalardan arındırılmış olarak tutulması ve üzerinden yapılacak her türlü sorgulamaya da açık olması bakımından guruplandırılmaları gerekmektedir. Olayı daha geniş düşünecek olursak ,elde edilen bu veri topluluğundan hastanenin değişik birimlerince faydalanılabileceği gibi hastaneler arası kurulacak bir ağ ile de hasta hakkında anında bilgi sahibi olmak mümkün olacaktır.

Görüldüğü gibi böyle bir çalışmada iyi bir sistemin seçilmesinin yanısıra iyi bir veri tabanının da oturtulması gerekmektedir. Kurulacak olan bu veri tabanının tıbbın değişik konularına hizmet verebilecek , istendiği takdirde sistemin gelişmesine ayak uydurabilecek , üzerinden yapılacak olan her türlü sorgulamayı karşılayabilecek türde olması gerekmektedir. Bu çalışmada da amaçlanan böyle bir sistemi ortaya koymaktır.

1.2. Tarihçe

Tıpta bilgisayar konulu çalışmalar literatürlerden incelenecek olursa, ilk olarak 1963 yılında KAISER-PERMANENTE (California) , WISCONSIN HOSPITAL gibi hastanelerde hasta kayıtlarının tutulması , gerekli ücret ve yatak kapasitesi takibi şeklinde olduğu görülür. 1980 yıllarında ise bu tür çalışmalara hastanın hastalığı ile ilgili şikayetleri, ekg ve laboratuar bulgularında katılması ile yeni bir yön verilmiştir. Hastanın hikayesini içeren bu veri kümesi basitçe hastaların dolduracakları

EVET ya da **HAYIR** şeklinde cevaplar ile yönlenecek bir hasta-bilgisayar diyalogu ile elde edilmektedir. Bu sistemde hastanın EVET cevabı ile sorgulama o konu üzerinde daha detaylı bir yönde gelişmekte HAYIR cevabı ile ise daha değişik bir soruya atlanılmaktadır, ya da BILMIYORUM cevabı karşısında soru daha detaylı olarak açıklanmaktadır. Böylece hastanın doktora söyleyemeyeceği bir durumunu bilgisayara açması daha kolay olacaktır.

NATIONAL INSTITUTES , TORONTO GENERAL HOSPITAL gibi hastanelerde ise hastanın elektrokardiogramları istendiğinde bir telefon hattı ile bilgisayara ulaşabilmekte ya da tüm laboratuvar ve yoğun bakımdaki kontroller aynı şekilde bilgisayar aracılığı ile saklanıp kontrol edilebilmektedir.

Görüldüğü gibi çalışmaların laboratuvar ile ilgili kısımları **HARDWARE** (Donanım) ağırlıklı olup **SOFTWARE** (Yazılım) kısmı donanıma nazaran daha az geliştirilmiştir..

Ülkemizde ise durum tam tersidir. Veri toplama işlemi daha dikkatli ve detaylı olarak yapılmakla birlikte donanım olarak yapılacak kontroller gerekli elektronik cihazlarını da beraberinde getirdiğinden yazılım üzerine yapılan çalışmalar daha kuvvetlidir.

Bu çalışmalardan bazıları YILDIZ UNIVERSITESI tarafından

CERRAHPAŞA TIP FAKÜLTESİ , LEPRİ HASTANESİ , KOŞUYOLU KALP HASTANESİNE yapılmıştır.

2- VERİ TABANI YAPISI

2.1- TANIM

Bir veri tabanı değişik verilerden istenilen bilgileri elde etmek için bu verileri hatalardan ve fazlalıklardan arındırarak , sistemin en kolay yapılabilirliği yanında saklama kolaylığı , Gelişmekle file sistemi ile karşılaştığında gerekli en büyük farklardan birisi , veri tipleri ile çalışmayıp , gerektiğinde yeni bir file yaratmadan yeni

bilgilere ulaşılabilir. Getirdiği diğer yararlar ise

1- **ENTEGRASYON OLMAZI** : Verilerin fazlalıklardan ve hatalardan arındırılması bir şekilde diskte olduğu az bir alan kaplayacak şekilde aralarında loğak bir bağ kurularak saklanması dolayısı ile erişim hızının artırılmasıdır.

2- **PAYLAŞILABİLİR OLMAZI** : Çok kullanıcıları bir sistemde aynı veriye pek çok kişinin aktif olarak erişebilmesidir. Aynı zamanda herhangi bir file de yapılacak bir güncelleştirme işlemi olarak o file in bağlantıda bulunduğu diğer bütün alanlarda da gerçekleştirilmelidir. Örneğin geleneksel file yapısında böyle bir durum mevcut değildir. Her file bu yapı içerisinde ayrı , ayrı tutulduğundan yeni bir veri için nasıl yeni bir file almak gerektirir. Güncelleştirme yapmak için de aynı zamanda bütün fileler tek , tek elden geçirilmelidir.

3- **STANDART OLMAZI** : Verilerin saklanması ve bu verilere erişim kullanıcının tanımladığı formata uygun şekilde olmaktadır. İstenildiğinde bu format değiştirilip her türlü veri için , sorgulama , güncelleştirme vb gibi işlemler bu yeni formata göre yapılabilir.

4- **GELİŞTİRİLEBİLİR** : Sistem her zaman yeni veri tipleri , sorgulama , yeni bilgiler elde etmeye uygun bir yapıya sahiptir.

2- VERİ TABANI OLUŞTURMA

2.1- Tanım

Bir veri tabanı değişik verilerden istenilen bilgileri elde etmek için bu verileri hatalardan ve fazlalıklardan arınmış , erişimin en kolay yapılabileceği tarzda saklama şeklidir. Geleneksel file sistemi ile karşılaştırıldığında görülen en büyük farklardan birikimsiz veri tipleri ile çalışmayıp gerektiğinde yeni bir file yaratmadan yeni

bilgilere ulaşabilmesidir. Getirdiği diğer yararlar ise :

1- **ENTEĞRE OLMASI** : Verilerin fazlalıklardan ve hatalardan arındırılmış bir şekilde diskde oldukça az bir alan kaplayacak şekilde aralarında lojik bir bağ kurularak saklanması dolayısı ile erişim hızının da arttırılmasıdır.

2- **PAYLAŞILABİLİR OLMASI** : Çok kullanıcılı bir sistemde aynı veriye pek çok kişinin aktif olarak erişebilmesidir. Aynı zamanda herhangi bir file'da yapılacak bir güncelleştirme otomatik olarak o file'ın bağlantıda bulunduğu diğer bütün alanlarda da gerçekleştirilecektir. Oysa geleneksel file yapısında böyle bir durum mevcut değildir. Her file bu yapı içerisinde ayrı , ayrı tutulduğundan yeni bir veri için nasıl yeni bir file açmak gerekiyorsa güncelleştirme yapmak için de aynı yöntemle bütün filelar tek , tek elden geçirilmelidir.

3- **STANDART OLMASI** : Verilerin saklanması ve bu verilere erişim kullanıcının tanımladığı formata uygun şekilde olmaktadır. İstendiğinde bu format değiştirilip her türlü özet bilgi , sorgulama , güncelleştirme ...vb gibi işlemler bu yeni formata göre yapılabilir.

4- **GELİŞTİRİLEBİLMESİ** : Sistem her zaman yeni veri tipleri karşısında yeni bilgiler elde etmeye uygun bir şekilde gelişip büyüebilir. Büyüyebilmenin sınırı sadece bilgisayarın hardware (Donanım) sınırına bağlı olup veri tabanı bu konuda tamamen sınırsız bir yapı teşkil eder. Böylece her yeni veri tipi için ayrı bir file açmak gerekmemektedir ki bu da zaman ve maliyeti oldukça kısar.

5- **GUVENİLİR OLMASI** : Çok kullanıcılı bir sistemde bir file'ın pek çok kişi tarafından kullanılması , geleneksel fil'lara oranla merkezi kontrolde güçleştirmektedir. Bu duruma karşı alınan belli başlı önlemlerin arasında :

1- Veri tabanında tablolar üzerinde **view** dediğimiz bir görüntü alan yaratmak mümkündür. Böylece bazı kullanıcıların erişilmesi istenmeyen bilgilere ulaşması hatta bu bilgilerin varlığından haberdar olması bile önlenmiş olur.

2- Filelara erişimde kullanıcıların bağlantı izinleri de göz önünde tutulur. Bu konuda **FERNANDEZ** , **SUMMERZ** ve **WOOD** tarafından 1981 yılında bir model üretilmiştir. B umodele göre bir tabloda her file'a hangi kullanıcıların ne şartla bağlanacağı yazılmış ve buradan gerekli denetim yapılmıştır. Şekilde de görüldüğü gibi ilk satırda *kayıt* bölümündeki bir kişi laboratuvar sonuçları üzerinde hiç bir şekilde değişiklik yapamaz veya ikinci satırda ki gibi hastane nöbet tablosuna sadece *terminal* kullanıcıları nöbet saatini 6 saatten fazla yazmamak kaydıyla değiştirebilirler.

Kullanıcı	Tablo	Olay	Değer
Kayıt	Laboratuvar	Modify	None
Terminal	Nöbet	Modify	saat LE 6
Doktor	EKG	Delete	None

3- Kullanıcıların sistem dışında kendi tablolarını yukarıdaki tabloya benzer şekilde yaratıp istedikleri filelarını diğer kullanıcılara göstermeye bilirler.

4- Veriler veri tabanı tarafından kodlanarak da saklanabilirler. Bunun yanında kullanıcılar da isterse kendi kodlarını yaratabilirler. Böylece file'ın içeriğinin okunması

imkansız bir hale gelir. Ancak bu kodu tekrar veriye dönüştürecek bir programında sistem tarafından gizli olarak tutulması gerekmektedir.

5- Daha gelişmiş sistemlerde biometrik kontrol ile password'ün dışında ayrı bir kontrol de yapılmaktadır. Bu kontrolde her kullanıcının kredi kartına benzer özel bir kartında el imzası veya parmak izi saklanmaktadır. Sistemin biometrik okuyucusu ile bu kartın içeriği sistemdeki veri ile karşılaştırılmaktadır. Kartın kaybolması halinde ise bu karşılaştırma uymadığından sisteme girilemez.

5- *VERİNİN BAĞIMSIZ OLMASI* : Verinin herhangi bir programa bağımlı olmadan saklanması üzerinde yapılacak bir değişiklik karşısında programın yeniden yazılmasını önlemiş olur. Böylece daima program eldeki verilerin tiplerine göre olaya yön vereceğinden veri üzerindeki değişiklikten etkilenmez.

Sistemin getireceği dezavantajlara gelince :

1- Her gelişen sistem beraberinde personel eğitimini de getireceğinden bu olay bir problem yaratabilir. Sistemin herhangi bir kullanıcı tarafından değiştirilebilir olması bu işin sıkça yapılması halinde kontrolde zorlaştırır.

2- Minimum veri kullanımı aynı zamanda bu verilerin back-up (kopya) larında alınmasını gerekli kılar. Herne kadar veri tabanı bu işlemi otomatik olarak yapsa da alınan koyalara kontrolü karmaşık bir yapıya sebep olabilir.

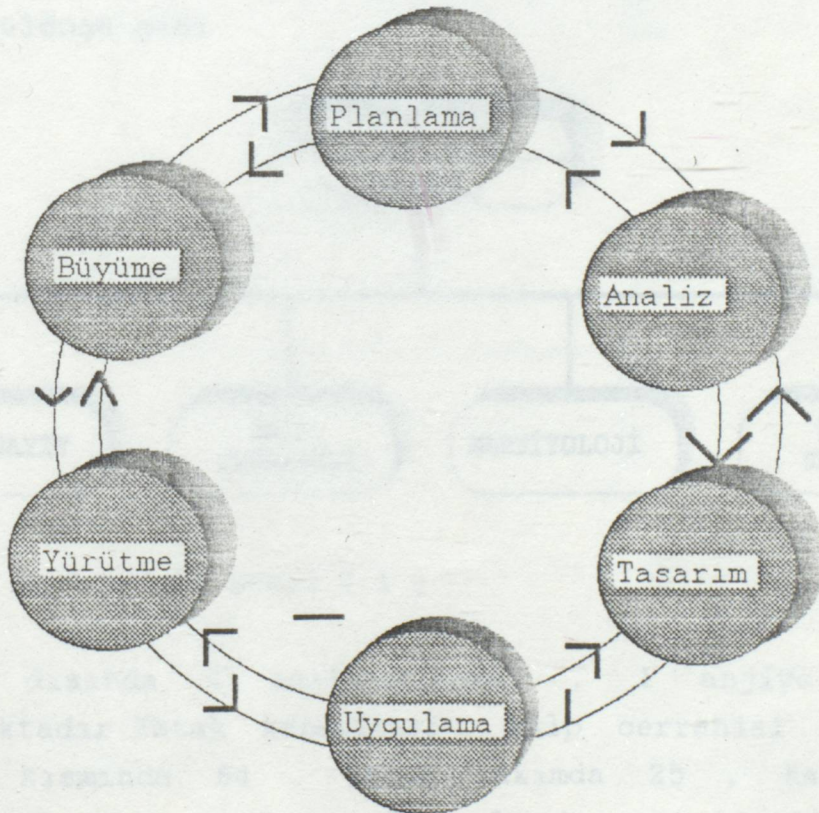
3- Çoklu kullanımın getireceği kolaylıkların yanında bazı dezavantajları da vardır. Bunlar , aynı veri farklı kullanıcılar tarafından değiştirilmek istendiğinde bağlantının senkronize olması halinde verinin bozulabileceği

veya kullanıcıların dead-lock denilen bir kitlenme türüne sebep olabileceği şeklinde sıralanabilir.

2.2- BİR VERİ TABANI ÇEVİRİMİ

Bir veri tabanı çevrimi analizi yapılacak olayın tarifinden yeni bir sistemin oluşturulmasına kadar geçen adımları içerir. Koşuyolu kalp hastanesi sistem analizi bu adımlar ile gerçekleştirilmiştir. Bunlar sırası ile :

- 1- Planlama
- 2- İsteklerin formülleştirilmesi ve analizi
- 3- Tasarım
- 4- Uygulama
- 5- Yürütme
- 6- Büyüme ve değişme



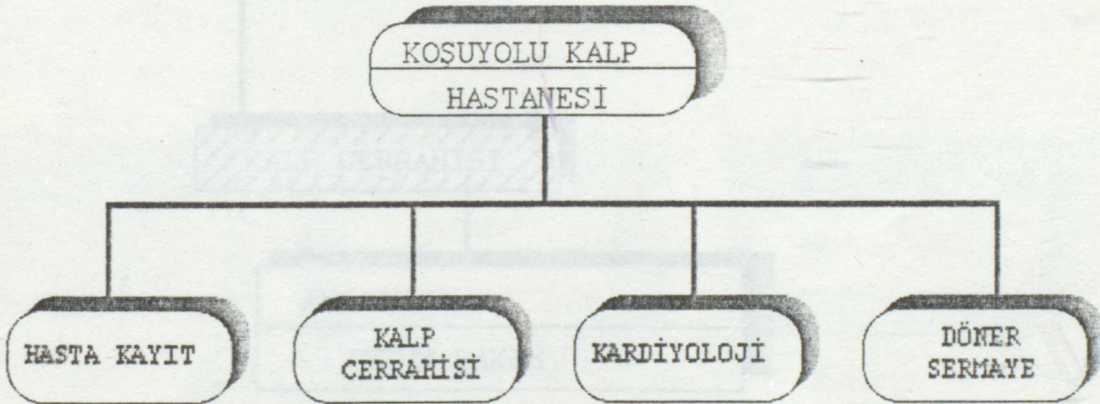
1- **PLANLAMA** : Giriş verilerinden yararlanarak stratejik

bir planın geliştirilmesinde ki adımdır. Bu adımda önce konunun iş analizi yapılır. Bu analizde önce mevcut sistemin sistem kurulmadan önce işleyen çalışma bölümlerinin saptanması bu bölümlerin kimlere bağlı olduğu ve sorumlulukları incelenir. Bölümler arasında hangi verilerin ortak kullanıldığı bu analizde saptanır. Analiz sonunda ise bir sonuç raporu oluşturulur. Koşuyolu kalp hastanesinde bu analiz aşağıdaki şekilde yapılmıştır.

2.2.3- KOŞUYOLU KALP HASTANESİ MEVCUT SİSTEMİN TANITIMI

2.2.3.1- Giris

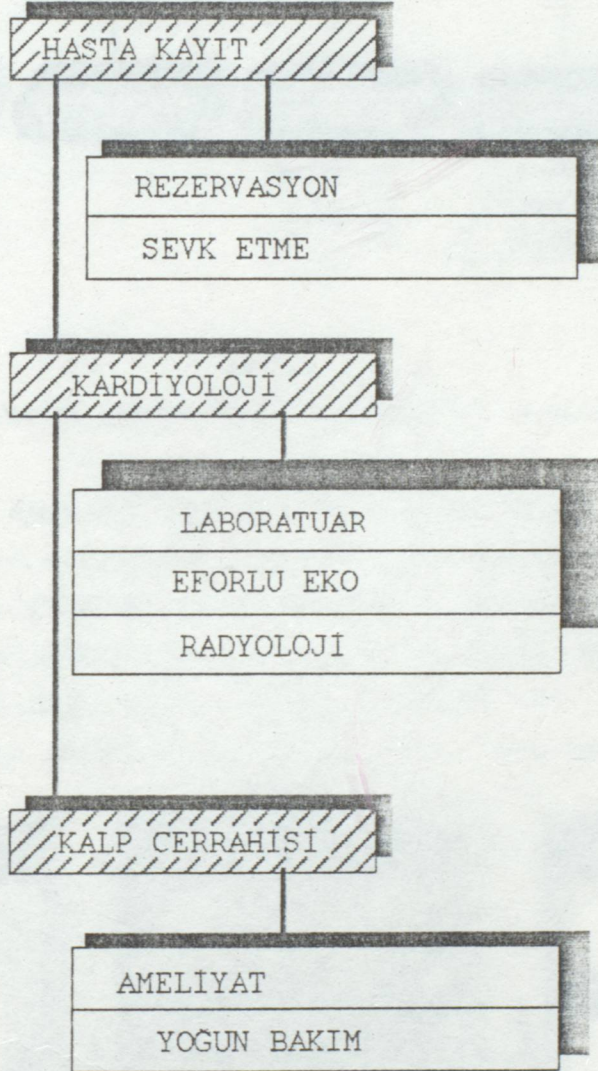
Koşuyolu kalp hastanesi kalp cerrahisi ve kardiyoloji bölümlerini içeren bir araştırma hastanesidir. Şekil (2.1) 'de görüldüğü gibi.



(Şekil 2.1)

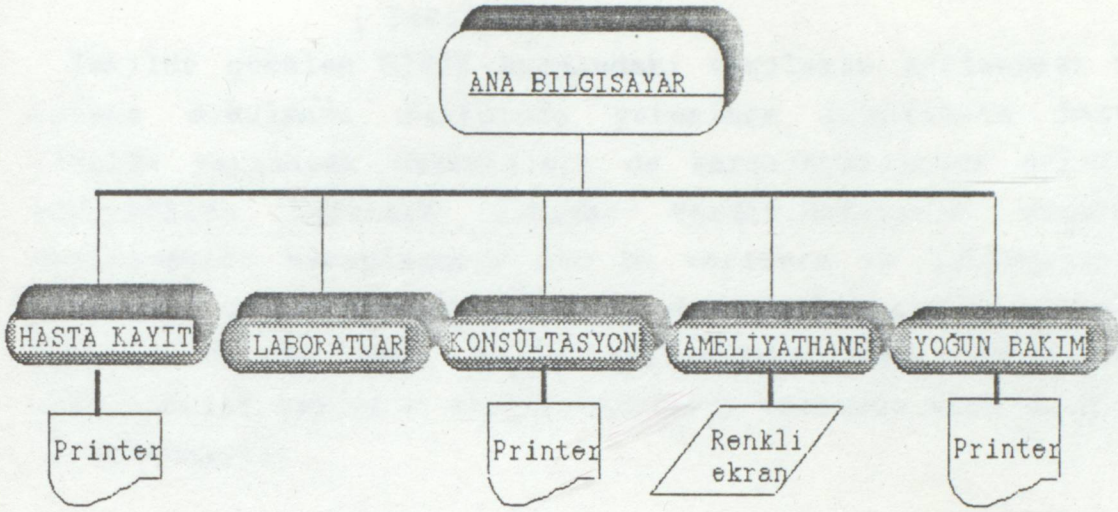
Bunu dışında 4 ameliyathanesi , 1 anjiyo ünitesi bulunmaktadır. Yatak kapasitesi kalp cerrahisi bölümünün servis kısmında 64 , yoğun bakımda 25 , kardiyoloji servisinde 40 , toplam : 129 olarak orta büyüklükte bir hastanedir. Yaklaşık günde 5-6 civarı açık kalp ameliyatı gerçekleştirilmekte, EKO ve eforlu EKG üzerine çalışmalar yapılmakla birlikte senede 4-5 defa bildirilere yapılan

istatistiki ölçümlerle katılmaktadırlar. Sonuçta böyle bir araştırma hastanesinde kayıtların üzerinden istatistiki değerlere varılabilecek şekilde düzenli olarak tutulması gerekmektedir.



(Şekil 2.2)

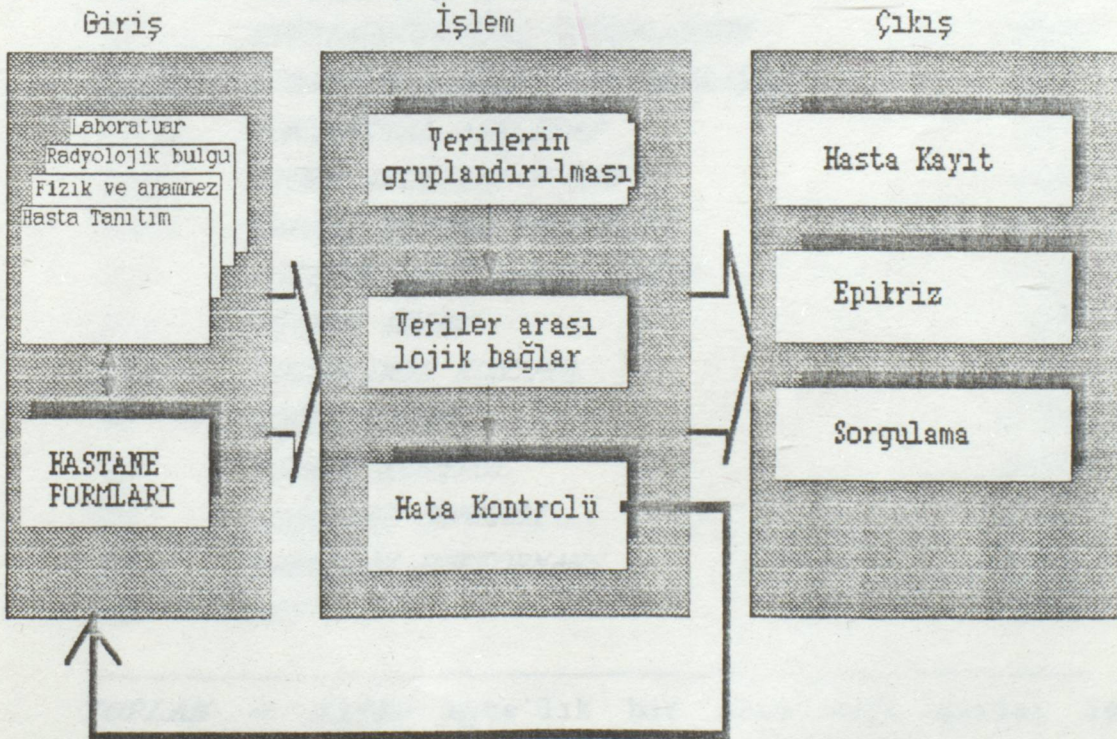
Ancak hastane bugüne kadar elindeki kayıtları değişik birimlere ait formlar halinde tuttuğundan bundan sonraki çalışmalarda bu formların bilgisayar girişine uygun olarak düzenlenmesi ile devam ettirilmiştir. Bu analize dayalı olarak da aşağıdaki şekilde bir ağın kurulmasına karar verilmiştir.



(Şekil 2.3)

2- İsteklerin formülleştirilmesi ve analizi

Yukarıda kurulan ağa çeşitli zamanlarda ve yerlerde veri girişleri ve çıkışları olacaktır. Bu giriş ve çıkış esnasında verileri işleyip kontrol edecek ve gerçekte de programların akışını yönlendirecek bir işlem safhası vardır. Şekil (2.4) 'de de görüldüğü gibi GİRİŞ , İŞLEM ve ÇIKIŞ' ı içeren bu üçlü akış **HIPO** diyagramları ile tasarlanmıştır.



(Şekil 2.4)

Şekilde görülen GİRİŞ kısmındaki verilerin saklanması ve işleme sokulması sırasında yeterince büyüklükte hatta ileride yapılacak eklentileri de karşılayabilecek şekilde büyüyeabilen hafızaya ihtiyaç vardır. Hafızanın minimum büyüklüğünün hesaplanması ise bu verilere ve kullanılacak programın genişliğine bağlıdır. Bu durumda hastane tarafından verilen ve bu güne kadar kullanılmış olan formlardan yararlanarak yaklaşık olarak verilerin tutacağı alan aşağıda hesaplanmıştır.

	FORM	ALAN
F0	HASTA TANITIM	200
F1	ANAMNEZ VE FİZİK İNCELEME	800
F2	RADYOLOJİK İNCELEME	130
F3	EKG İNCELEME	200
F4	EKOKARDIOGRAFIK İNCELEME	280
F5	KALP KATETERİZASYONU	350
F6	SİNTİGRAFIK İNCELEME	200
F7	LABORATUAR BULGULARI	1800
F8	ANESTEZİYOLOJİ	355
F9	EKSTRAKORPOREAL SİRKÜLASYON	900
F10A	ROMATİZMAL AÇIK KALP CERRAHİSİ	850
F10B	KONJENİTAL AMELİYAT	300
F10C	DİĞER AMELİYAT FORMU	580
F10D	KAPALI MITRAL VALVOTOMİ	600
F11	POSTOPERATİF BAKIM FORMU	1240
F12	KLİNİK İZLEM	360
F13	POLİKLİNİK KONTROL	350
G1	YOGUN BAKIM	9750
G2	TEDAVİ KONTROL	3250
G3	AMELİYAT SONRASI	9750
G4	KARDİYAK PERFORMANS	2000
G5	ACT	200

TOPLAM = 33925 byte'lık bir alan veri girişi için gerekmektedir. Senede yaklaşık 1000 hastanın baş vurduğu göz

önüne alınırsa sadece 35 MB 'lık bir alan + kullanılacak programların kaplayacağı alan ile birlikte 70 MB'lık hafızaya ihtiyaç vardır. Bunun yanısıra sistemin genişleyebilme özelliğinin olması, artan çevre birimlerini destekleyebilmesi ve en önemlisi bilgisayar destekli tasarım ile çizime uygun olması gerekmektedir. Buna bağlı olarak hafıza 4 MB ile 8 MB arasında olmalı ve özellikle işletim sisteminin ve kullanılacak yüksek seviyeli dillerin çoklu terminal bağlantısına olanak tanıyacak şekilde olması gerekmektedir. Sistem tasarımı bakımından ise veri tabanı kurmaya uygun dBASE , SQL gibi dillere , sorgulama bakımından ise LISP , PROLOG gibi yüksek seviyeli sorgulama dillerinin kullanımına uygun olmalıdır.

Yukarıdaki bütün bu özellikleri sağlaması açısından kurulacak ağda aşağıdaki şartların aranması gerekmektedir.

2.2.3.2- Teknik özellikler

1 - **DONANIM**

- . Ana hafıza 4 MB ile 8 MB arasında olup gerektiğinde 16 MB'a kadar büyüebilmelidir.
- . 32 bit mikro işlemcisi olmalıdır.
- . Hafızaya direk ulaşım (DIRECT ACCESS STORAGE) için 40 MB ile 70 MB arasında file kullanılabilmeli hatta bu file sayısı gerektiğinde iki katına çıkmalıdır.

2- **GÖRÜNTÜ HAFIZA**

- . Görüntü hafıza (VIRTUAL MEMORY) 40 bit adreslenebilir olup bağlanacak PC ler 80286/80287 mikro işlemcisi kullanabilmelidirler.
- . Kapasite en az 16 MB olmalıdır.

3- **MERKEZİ İŞLEM BİRİMİ**

- . Çoklu iş düzeninde (Multiprogramming) çalışmayı sağlayacak şekilde olmalıdır.
- . Tek başına kullanılabildiği gibi kurulmuş bir ağa da bağlanabilmelidir.
- . LAN (Local area network) her çeşit PC ile kurulabilir olmalı ve maximum 16 PC bağlanabilmeli ayrıca buna benzer bir ağ CAD/CAM 'ede dayalı olarak kurulabilmelidir.

4- İŞLETİM SİSTEMİ

- . UNIX veya benzeri bir işletim sistemi kullanarak maksimum 16 aktif terminal bağlantısına olanak tanıyan bildiği gibi kullanıcılara da çoklu iş düzeninde grafik yapıya uygun ve problem çözümlemede çeşitli yüksek seviyeli dilleri kullanabilecek rahatlık sunulabilmelidir.
- . Veri tabanı (Data Base) yönetimini uygulayabilecek bir yapıya sahip olmalı
- . Ayrıca kurulan ağın dışında diğer kaynakların ve file'ların ortak paylaşımlı kullanımı sağlanmalı
- . Ekranda birden fazla pencere açarak değişik programların aynı anda derlenmesine olanak tanınmalıdır.

5- YAZILIM

- . Sistemde yüksek seviyeli dillerden FORTRAN 77, PASCAL PROLOG , LISP , C 'nin kullanımının yanısıra dbase ,

SQL , ORACLE gibi veri saklama ve aramaya yönelik dillerin de kullanımına olanak sağlanmalıdır.

Hazırlanan teknik şartnameye göre IBM'in bir ürünü olan 6150 serisinin 125 modelinin 1 renkli terminal 4 aktif terminali ile birlikte alınmasına karar verilmiştir. İşletim sistemi AIX (Advance Interactive Executive) :UNIX IV 'in yeni bir derlemesi olup C,LISP,SQL gibi yüksek seviyeli dillerin kullanımına olanak tanımaktadır. Ayrıca kendi yapısında ki DBMS (Data Base Management System) tasarımı ile de veri tabanına uygun file'lar yaratıp bu file'lara istenen şekilde arama ve sorgulamaya yönelik şekilde erişebilmektedir. Aşağıda açıklanan bu sisteme ait detaylı bilgiler C'nin geniş özellikleri ile birleştirildiğinde ortaya SQL 'den daha fazla imkanlar sağlayan bir sistem çıkmaktadır.

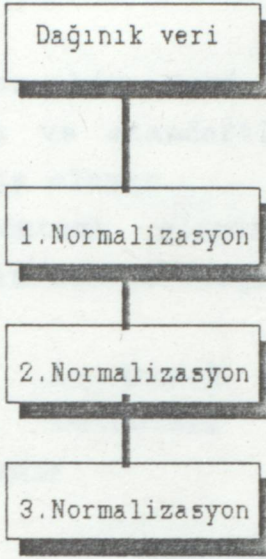
3- Sistem tasarımı

Eldeki veriler incelemeye alındığında bunların önce ekran tasarımı yapılarak veri girişine uygun hale getirilmiştir. Bu tasarımın gerçekleştirilmesinde veriler üzerinde 3. derece normalizasyon yöntemi uygulanmıştır. NORMALİZASYON 3 aşamada gerçekleştirilen ve sistemin oluşturulmasından önce, verileri düzene sokmak bakımından gerekli olan bir yöntemdir. Bu yöntemin,

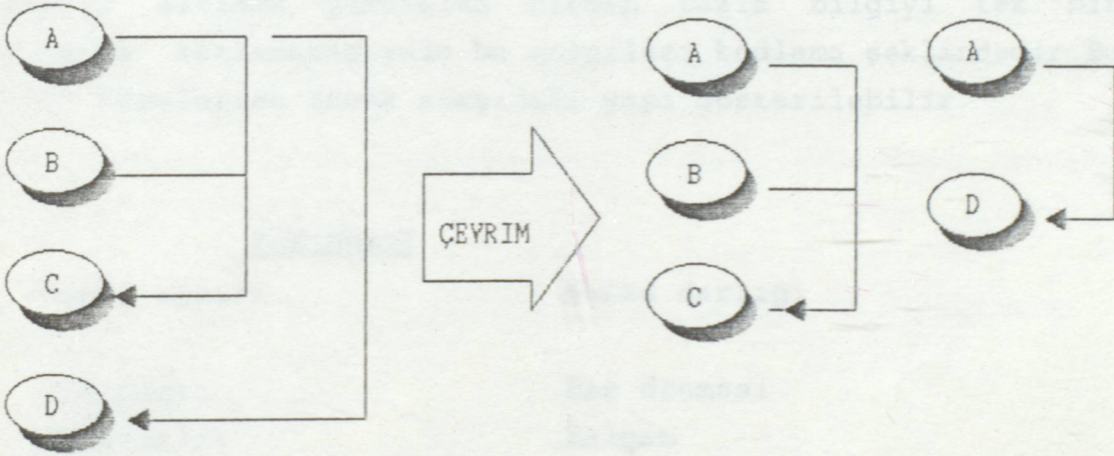
Birinci aşamasında: Eldeki dağınık veriler iki boyutlu bir matrise indirgenir. Böylece hangi verinin kime karşılık düştüğü görülmüş olunur.

İkinci aşamasında : Elde edilen bu iki boyutlu matris üzerinde gruplandırmalar yapılır. Gruplandırma , verilerin bir yerde tekrarını önlediği gibi, diğer veriler ile lojik bağların kurulmasında da kolaylık sağlar.

Üçüncü aşamasında : Düzenlenmiş ve gruplandırılmış veriler guruplarına ve aralarındaki lojik ilişkilere göre ayrı alanlara bölünürler. Böylece tek bir yerde saklanmanın getireceği karmaşıklık önlenmiş olunur. (Şekil 2.6)

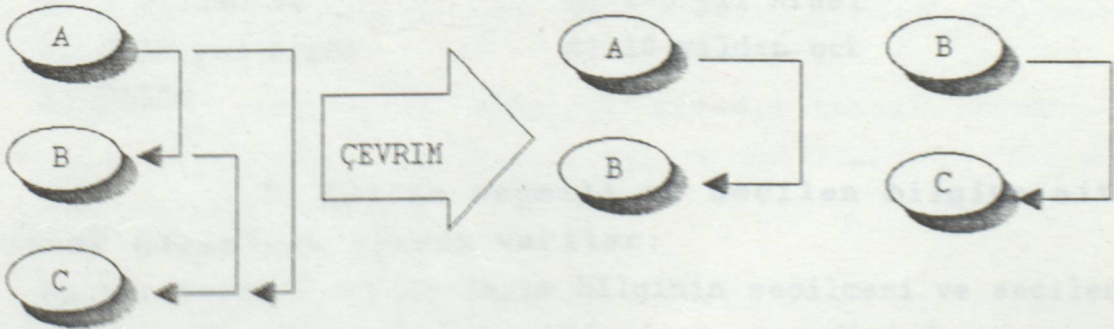


İKİNCİ NORMAL FORMA ÇEVİRME



(Şekil 2.6 b)

ÜÇÜNCÜ NORMAL FORMA ÇEVİRME



(Şekil 2.6 c)

Böylece bir veri tabanının sağladığı bütünlülük , gizlilik ve standartlık iyi bir normalizasyon ile baştan sağlanmış olunur.

Bu yöntemi elimizdeki verilere uyguladığımızda elde ettiğimiz sonucu aşağıdaki şekilde görmek mümkündür.

1- Direk girişler : File'da o bilgiye karşı düşecek sütünlara ara file kullanmadan yapılacak girişlerdir.

DOSYANO :
AD :
SOYAD : gibi

2- Çoktan seçmeli veya sık : Hastaya ait tek başlık altında yürütülen birden fazla bilgiyi tek bir file'da saklamanın yolu bu bilgileri kodlama şeklindedir. Bu veri kümelerine örnek aşağıdaki yapı gösterilebilir.

YAKINMASI

Göğüs ağrısı Nefes darlığı

Çarpıntı Baş dönmesi

Halsizlik Balgam

DİGER :

SEMPTOMLARIN SÜRESİ

- a) 1 yıldan az b) 1-5 yıl arası
c) 5-10 yıl arası d) 10 yıldan çok
e) DİGER :

3- Çoktan seçmeli ve seçilen bilgiye ait direk girişleri içeren veriler:

Bu tür veriler birden fazla bilginin seçilmesi ve seçilen bilgiye ait daha detaylı bilgileri içerdiğinden yapısı

itibari ile parçalı olarak saklanmak zorundadır. Bu tip örnek olarak :

ÖKSÜRÜK

Süre : Nefes darlığı ile
Kuru Geceleri
Sırt üstü yatınca
DİĞER :

MYOKARD INFARKTI

Transmural Nontransmural
Yeri : Tarihi .../.../...

4- MATRİS KONUMU GÖSTERENLER

Birden fazla aynı anda doldurulması gereken verileri içerirler. Bu tür veriler çoktan seçeneklerin kodlanıp daha sonra bunlara ait bilgilerin ayrı bir file'da tutulması ile saklanabilirler. Bu guruplamaya örnek aşağıdaki şekilde gösterilebilir.

	<u>Akut</u>	<u>Subakut</u>	<u>Kronik</u>	<u>Gecirilmiş</u>
SEPTAL	X			
ANTEROSEPTAL			X	
ANTERIOR				
ANTEROAPIKAL				
INFERİOR				X

5- Hastaya ait bilgilerin hikaye şeklinde girilmesi ile

Doktorun ancak uzun cümleler ile hastalığın durumu ve tedavisi şeklinde açıklayabileceği ve epikriz'de önemle göz önünde bulundurulmuş veri şeklindedir.

Veri tabanına uygun şekilde düzenlenen bu verilerin ekrandan girişine de uygun hale getirilmesi ayrı bir ekran

tasarımını da beraberinde getirmektedir.

DOSTA NO

149888

ADY

NESRİN

SOYADI

ERDEMİ

YAKINHAŞI

GÖĞÜS AGRISI

NEFES DARLIĞI

GARFİYİ

BAS DÖĞÜSİ

HALSİZLİK-YORGUNLUK

BALGAM

NEHŞTİZİ

AYAKLARDA ÜŞÜME

YAKINHAŞI

BÖLÜMLERDE AĞRI

BACAKLARDA SIZLIK

BACAKLARINDA AĞRI

KOLLARDA AĞRI

BÖRANNA

BULANTI-KUSMA

GÖRNE DÖĞÜMLERİ

ELLERDE ÜŞÜME

GÖZLEMLERİ-SİZLİK
VE KIRILMA

KULAK ÇİNCİLİĞİ

DİĞER YAKINHAŞI

SEMPTOMLARIN SÜRESİ

1) 1 yıldan az

2) 1-5 yıl

3) 5-10 yıl arası

4) 10 yıldan çok

SEMPTOMLARDA GİDİŞLİLİK

1) yok

2) 5 yıldan az

3) 5 yıldan çok

GÖĞÜS AGRISI

Ağrının Yeri

RETROSTERNAL

SOL-ON GÖĞÜS

SOL YAN

SAG GÖĞÜS

SAG YAN

EPİGASTRUM

K O S U Y O L U K A L P H A S T A N E S I

A N A M N E Z V E F I Z I K İ N C E L E M E

DOSYA NO

149888

ADI

NESRİN

SOYADI

ERDEMLİ

.....Y A K İ N M A S I.....

GOGUS AGRISI

NEFES DARLIĞI

CARPINTI

BAS DONMESİ

HALSİZLİK-YORGUNLUK

BALGAM

HEMOPTİZİ

AYAKLARDA USUME

F1-1

.....Y A K İ N M A S I.....

EKLEMLERDE AĞRI

BACAKLARDA SİSLİK

BACAKLARLARDA AĞRI

KOLLARDA AĞRI

MORARMA

BULANTI-KUSMA

GORME BOZUKLUGU

ELLERDE USUME

EKLEMLERDE SİSLİK
VE KIZARIKLİK

KULAK CİNLAMASI

D İ G E R : YAKINMASIZ

.....S E M P T O M L A R İ N S Ü R E S İ.....

1) 1 yıldan az

2) 1-5 yıl

3) 5-10 yıl arası

4) 10 yıldan çok

.....S E M P T O M L A R D A S İ D D E T L E N M E

1) yok

2) 6 aydan az

3) 6 aydan çok

F1-2

.....G O G U S A G R İ S İ.....

Ağrının yeri

RETROSTERNAL

SOL-ON GOGUS

SOL YAN

SAG GOGUS

SAG YAN

EPIGASTRIUM

Agrinin tipi

SIKISTIRICI

YANICI

BATICI

BASKI TARZINDA

IC SIKINTISI TARZINDA

D I G E R :

F1-3

Agrinin yayilimi

SOL ON GOGUS

SAG ON GOGUS

SOL GOGUS

SAG GOGUS

SOL SKAPULA

SAG SPAPULA

SOL KOL

SAG KOL

BOYUN

SIRT

KARIN

D I G E R :

Agrinin suresi

1) 10'az

2) 10'-30'

3) 30'cok

F1-4

Agriyi baslatan ve/veya artiran faktorler

FIZIKSEL EPOR

STRESS

AGIR YEMekten SONRA

SOGUK HAVA

SEKSUEL AKTIVITE

DERIN NEFES ALMAK

SIRT USTU YATMAK

YAN YATMAK

HEYECANLANMAK

GECE

EFORLA ILGISIZ

1) 1 yaştıkta 2) 2 yaştıkta F1-5
3) yetemiyor

G Agriyi geciren faktorler
O
G
U
S
DINLENMEK ANALJEZİKLER
AYAGA KALKMAK GEGİRMEK
OGUSTURMAK SIRT USTU UZANMAK
KENDİLİGİNDEN DİL ALTI NİTROGLİSERİN

A Agriya eslik eden semptomlar
G NEFES DARLIĞI CARPINTI
R GEGİRTİ BAS DONMESİ
I TERLEME BULANTI-KUSMA
S OKSURUK KOPUKLU BALGAM ÇIKARMA
I HEMOPTİZİ F1-6

G
1) class 1
2) class 2
3) class 3
4) class 4
5) class 5

Class 0 : Angina yok
Class 1 : Ağır eforla ağrı var
Class 2 : Fazla merdiven çıkma, mesafeler kısa
Class 3 : 100-200 metre hızla yürümekle ağrı var
Class 4 : Herhangi bir fiziksel aktivite ile istirahat
baldır ağrı var

F1-4

NEFES DARLIĞI
CARPINTI
BULANTI-KUSMA
OKSURUK
HEMOPTİZİ
KOPUKLU BALGAM

NEFES DARLIĞI
CARPINTI
BULANTI-KUSMA
OKSURUK
HEMOPTİZİ
KOPUKLU BALGAM

O Hasta kac yastikta yatiyor ? 1

- G 1) 1 yastikta 2) 2 yastikta
- U 3) yatamiyor

S Hastada angina pektoris dusunuluyorsa tipi

- A 1) stable 2) unstable
- G 3) atipik 4) akut myokard infarkti

R

I

S

F1-7

G

O Stable angina pektorisin KANADA KALP cemiyetine gore
siniflandirilmesi

G

- U 1) class 0 2) class 1
- S 3) class 2 4) class 3
- 5) class 4

A

Class 0 : Angina yok

G

Class 1 : Agir eforla angina var

R

Class 2 : Fazla merdiven cikma, emosyonel stress, sogukta
kalmakla angina var

I

Class 3 : 100-200 metre duz yolda yurumekle, merdiven cikmakla
angina var

S

Class 4 : Herhangi bir fiziksel aktivite ile istirahatte
bazen angina var

I

F1-8

..... NEFES DARLIGI

EFORLA

ISTIRAHATTE

GECE GELEN

Nefes darligina eslik eden semptomlar

CARPINTI

OKSURUK

KOPUKLU BALGAM

GOGUS AGRISI

HALSIZLIK
MORARMA

BAS DONMESI
KOYU BALGAM

KENDILIGINDEN

ILACLA

KAROTIS MASAJI VEYA
VALSALVA MANEVRASIYLA

..... B A S D O N M E S I

SPONTAN

EPORLA

DEVAMLI

CARPINTI ILE

NEFES DARLIGI ILE

GOGUS AGRISI ILE

POZISYON DEGISTIRMEKLE

BAS-BOYUN
HAREKETI ILE

AYAGA KALKMAK ILE

DIGER

F1-10

DOSYANO 149888

..... B A S D O N M E S I

Bas donmesi ile birlikte gorulen semptomlar

GECICI KORLUK

PARESTEZI

KONUSMA BOZUKLUKLUGU

SIYANOZ

KASILMALAR

D I G E R :

..... B A Y I L M A

GOGUS AGRISI ILE

BAS AGRISI ILE

CARPINTI ILE

SPONTAN

EPORLA

POZISYON ILE

F1-11

..... B A Y I L M A

AYAGA KALKMAK ILE

NEFES DARLIGI ILE

BAS BOYUN HAREKETIYILE

D I G E R :

Veriler Bunlar:

1-Her dbf tablosu ancak 128 tane değışkenin tanımlanması için vermektedir.

2-Her tablo bir satıra 4000 byte'lık yer ayırmaktadır.

3-Aynı zamanda ancak 15 tablonun açılmasına izin vermektedir.

4-Bir scr file'ı kendi içinde 10 ekranı tanımlama hakkı vermektedir.

Bu şartlar altında verilerin saklanması şekli her bir verinin bir dbf file'ı içerisinde saklanması ile gerçekleştirilmiştir. Dbase iii plus ancak bir dbf file'ı içerisinde yapılabilecek sorgulama yönteminde kolaylık sağladığından, bu yöntem dışında formların birer dbf file'ı olarak iletilmesi arama yapmada zorluklar çıkarır. Bu problemi çözmek için bütün dbf file'ları içerikleri ayrı bir tabloda her bir değışkenin hangi tabloya karşı düştüğünü belirtmek tutulmuş, böylece arama önce aranan verilerin tablo isim belirlenmesi daha sonra bu tablo içerisinde dbase iii plus'ın kullanıldığı bir scr file'ı olarak yapılabilecek bir arama şekli daha açık bir şekilde gerçekleştirilmiştir.

3- DOS VE AIX İŞLETİM SİSTEMLERİNDE TASARIMIN GERÇEKLEŞTİRİLMESİ

3.1- DBASE III PLUS ile

IBM'in model 60'ı üzerinde DOS işletim sistemi altında yapılan tasarımda dbase iii plus veri yönetimine uygun dil olarak seçilmiştir. Bu dilde kullanılacak file tiplerini :

1-Veri tabanı file'ı (dbf)

2-Format file'ı (fmt)

3-Ekran file'ı (scr)

sıralamak mümkündür.

Ancak bu dilin tasarımda getirdiği kolaylıkların yanısıra file yapılarında getirdiği bazı kısıtlamalarda

vardır. Bunlar:

1-Her dbf tablosu ancak 128 tane değişkenin tanımlanmasına izin vermektedir.

2-Her tablo bir satıra 4000 byte lık yer ayırmaktadır.

3-Aynı zamanda ancak 15 tablonun açılmasına izin verilmektedir.

4-Bir scr file'ı kendi içinde 10 ekranı tasarlamaya izin vermektedir.

Bu şartlar altında verilerin saklanma şekli her 10 ekranın bir dbf file'ı içerisinde saklanması ile gerçekleştirilmiştir. Dbase iii plus ancak bir dbf file'ı üzerinden yapılacak sorgulama yönteminde kolaylık sağladığından, bu yöntem eldeki formların birkaç dbf file'a bölünmesi ile arama yapmada zorluklar çıkarmıştır. Bu problemi yenmek için bütün dbf file içerikleri ayrı bir tabloda her bir değişkenin hangi tabloya karşı düştüğünü belirtecek tutulmuş, böylece aramada önce aramanın yapılacağı tablo ismi belirlenmiş daha sonra bu tablo üzerinde dbase'in özellikleri uygulanmıştır. ŞEKİL (3.1)'deki akış diyagramında bu arama şekli daha açık bir şekilde belirlenmiştir.

Sistemin getirdiği diğer bir zorluk ise daha sonra geliştirilmesine uygun olmayışıdır. Her dbf file'ının ancak 128 değişken tarifine olanak sağlaması daha sonraki veri eklemelerini bu sınırla imkansız hale getirmektedir. Sistemin paylaşımlı olmaması ise bir DISTRIBUTED DATA BASE (Dağıtılmış veri tabanı) kullanımına olanak vermediğinden hastane sistemine uygun görülmemiştir.

3.2- 6150 Sisteminde DBMS Yönetimi

AIX işletim sistemi altında çalışan veri tabanı organizasyonu iki ana birimi içermektedir. Bunlar :

vardır. Bunlar:

1-Her dbf tablosu ancak 128 tane değişkenin tanımlanmasına izin vermektedir.

2-Her tablo bir satıra 4000 byte lık yer ayırmaktadır.

3-Aynı zamanda ancak 15 tablonun açılmasına izin verilmektedir.

4-Bir scr file'ı kendi içinde 10 ekranı tasarlamaya izin vermektedir.

Bu şartlar altında verilerin saklanma şekli her 10 ekranın bir dbf file'ı içerisinde saklanması ile gerçekleştirilmiştir. Dbase III plus ancak bir dbf file'ı üzerinden yapılacak sorgulama yönteminde kolaylık sağladığından, bu yöntem eldeki formların birkaç dbf file'a bölünmesi ile arama yapmada zorluklar çıkarmıştır. Bu problemi yenmek için bütün dbf file içerikleri ayrı bir tabloda her bir değişkenin hangi tabloya karşı düştüğünü belirtecek tutulmuş, böylece aramada önce aramanın yapılacağı tablo ismi belirlenmiş daha sonra bu tablo üzerinde dbase'in özellikleri uygulanmıştır. Aşağıdaki akış diyagramında bu arama şekli daha açık bir şekilde belirlenmiştir.

1- Record type
2- Field type

Sistemin getirdiği diğer bir zorluk ise daha sonra geliştirilmesine uygun olmayışıdır. Her dbf file'ının ancak 128 değişken tarifine olanak sağlaması daha sonraki veri eklemelerini bu sınırla imkansız hale getirmektedir. Sistemin paylaşımlı olmaması ise bir DISTRIBUTED DATA BASE (Dağıtılmış veri tabanı) kullanımına olanak vermediğinden hastane sistemine uygun görülmemiştir.

3.2- 6150 Sisteminde DBMS Yönetimi

AIX işletim sistemi altında çalışan veri tabanı organizasyonu iki ana birimi içermektedir. Bunlar :

1- **Program arabirimi** : C programları içinde kullanılmak üzere file yaratma ve bu file'dali veriyi istenilen şekilde işlemeyi sağlayan fonksiyonlar bir set halinde toplanmış şeklidir. Bu fonksiyonları 6 gurup şeklinde incelemek mümkündür.

- 1- AIX file yapısı
- 2- Katolog
- 3- Transaction (İş kaydı)
- 4- Çeşitli fonksiyonlar

1.2- **AIX file yapısı** : AIX işletim sistemi altında çalışan bir file'in içeriğinden çok ismi ve bağlantı önceliğinin sistem tarafından bilinmesi gerekir. Bunun anlamı bir file'in içerisinde her çeşit bilgiyi saklamanın , aramanın , eklemenin ve silmenin mümkün olduğudur. Bu işlem ise veri tiplerinin karışık olmasından dolayı oldukça zaman alır. Oysa veri yönetimi sisteminin file yapısı kullanıldığında veriler parçalara bölünerek üzerlerinde işlem yapılmasını kolaylaştırırlar. Bu parçalama işlemi iki tür file yapısına sebep olur. Bunlar :

- 1- Record tipi
- 2- Field tipi

1- **Record tipi** : Veri yönetiminde bu tür file'larda bilgiler değişken veya sabit record ile tutulurlar. Bu yöntemle değişik veri tiplerini saklamak mümkündür. Ancak bu veri tiplerinin değişik formatlar halinde saklanması arama yapma sırasında karışıklıklara sebep olacağından sorumluluk kullanıcıya aittir. Bu durumda file üzerinde bir index anahtarı seçilmesi gerektiğinden en azından bu anahtarın yerinin record üzerinde sabit tutulması gerekmektedir. Aşağıdaki şekilde de görüleceği gibi record türünde saklanan verilerin uzunlukları sabit değildir. Seçilen index anahtarlara bu file yapısında rastgele erişmek mümkündür.

2- *Field tipi* : Bu yapının terminolojisi biraz daha farklıdır.

1-Field türü bir file'a **TABLO**

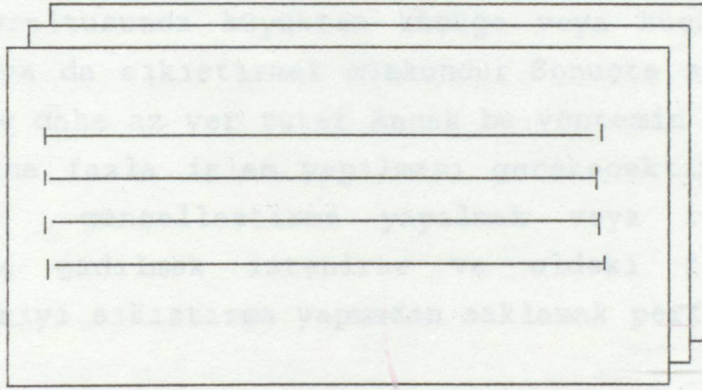
2-Field türü bir record'a **SATIR**

3-Ve bu file'ın bir field'ına **KOLON** ismi verilir.

Aşağıdaki şekilde de bu durum açıkça görülmektedir.

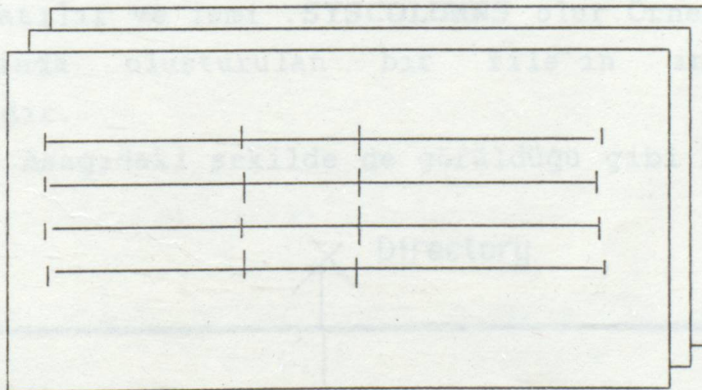
Bu tür file'larda veriler üzerinde yapılacak işlemler bu file'daki kolonların arasındaki ilişkiye bağlıdır. Kısaca bir record file'ında erişim recordlar ile bir field file'ında ise kolonlar ile olmaktadır. Her iki file'da da kolon ya da record uzunluğu 254'ü geçemez.

Record 1



kolon a

Satır 1



field tipi tablo

(Şekil 2.6b)

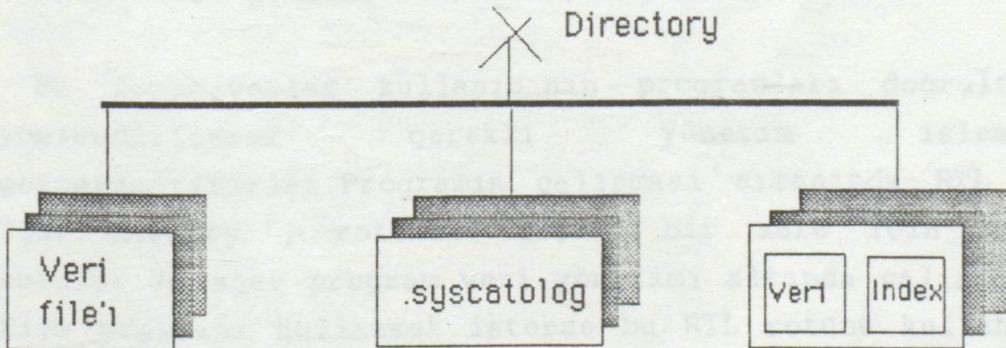
1.3- **TRANSACTION** (İş kaydı) : Veri yönetimi sistemi ile her AIX directory'sinde birden fazla file'ı kapsayan bir katalog file'ı vardır. Directory ile aynı bağlantı iznine

sahiptir. İlk record ya da field türü bir file yaratıldığında otomatik olarak açılır ve son file silindiğinde yine otomatik olarak silinir. İçerik olarak , yaratılan tablolar hakkında kolon sayısı , veri tipi , uzunluğu , index seçilen keylerin tipi ve yeri gibi bilgileri kapsar.

1.3.1- *Index yapısı* : Bir veri yönetimi index'i file recordlarının veya tablo satırlarının listesinden oluşur. Böylece her defasında istenen record veya satır sıralamadan o veriye ulaşılmış olunur. Her index için bir anahtar seçmek ve isim belirlemek gerekir. Bazen bir record üzerinde birden fazla anahtar seçilebilir. Bu durumda bu anahtarlara **COMPOSITE** anahtar denilir, ve her değer bu anahtarın bir parçasıdır. Seçilen anahtarı kullanıcının isteği doğrultusunda büyükten küçüğe veya küçükten büyüğe sıralamak ya da sıkıştırmak mümkündür. Sonuçta sıkıştırılmış veri diskde daha az yer tutar. Ancak bu yöntemin kullanılması halinde daha fazla işlem yapılması gerekecektir. Eğer büyük çapta bir güncelleştirme yapılmak veya rastgele bir sorgulamaya gidilmek istenirse ve eldeki file oldukça küçükse veriyi sıkıştırma yapmadan saklamak performansı daha arttırır.

Record düzeyinde file'lar üzerinde index yapılamaz. Herhangi bir index yaratıldığında index file'ıda otomatik olarak yaratılır ve ismi **.SYSCOLUMNS** olur. Örneğin : HASTANE ismi altında oluşturulan bir file'ın index file'ı **.HASTANE**'dir.

Özetle : Aşağıdaki şekilde de görüldüğü gibi bir AIX file yapısı,



1- Veri filelerini içeren **ANA SİSTEM FILE**lerinden,
 2- **.SYSCATALOG** larını içeren **veri yönetim kataloglarından** ,

3-Veri fileları ve index filelerini içeren veri yönetimi filelerinden oluşur.

Kullanıcının dışında oluşturulan index ve katalog fileları hatasız olarak çalışan AIX sistem fonksiyonlarını saklar.

File isimleri ise :

1-Her veri file'ı ismini yaratıldığı anda alır.

2-Bu file'ın index file'ının ismi file ile aynı olup .periyodludur.

3-Katoloğu ise .syscatalogdur.

2- **Veri yönetimi hizmetleri** : Kullanılan file'ların çeşitli işlemlere tutulmasını sağlayan bazı shell komutlarının bir C dili altında kullanılması şeklindedir. Bu komutlardan bazıları :

1- File veya tabloyu arşivleme

2- Katoloğu sıkıştırma

3- File veya tabloyu kopyalama

4- File veya tabloyu yaratma

5- Silme

6- Tarifleme

7- Liste alma

8- Başka bir yere aktarma

9- Geri getirme olarak sıralanabilir.

Bu fonksiyonlar kullanıcının programları doğrultusunda yönlendirilerek gerekli yönetim işlemlerini gerçekleştirirler. Programın çalışması sırasında RTL (Run Time Library) rotününe geçici bir süre için bağlantı kurulur. Ve eğer program veri yönetimi altında çalışacak bir file yapısını kullanmak isterse bu RTL rotünü kullanılacak komutların bir kopyasını alır. Bir anlamda kopyaların

atıldığı alan ile RTL arasında semafor oluşturulur. Bu tür bir çalışma ancak DMSMAIN adı altında bir EXEC veya FORK ile harekete geçirilebilir. Kullanılan fonksiyonların kod numaraları yeniden çağrıldığında da kullanılmak üzere bir yere atanır.

2.1- VERİ YÖNETİMİNDE GÜVENİLİRLİK

Filelara bağlantı izni olmayan bir kullanıcı asla erişemez. Bunun denetimi ise şu şekilde gerçekleştirilir.

1-Sistemde çalışma sırasında her file'ın owner ve group bağlantı konumları vardır. Bu konumlardan birine girmeyen bir kullanıcı o file'a ulaşamaz.

2-Her file directoryde 9 bitten oluşan bir bağlantı izin koduna sahiptir. Bu bitler okuma , yazma ve işleme koymayı gösterir. Ancak super kullanıcı modundaki bir kişi bu kodu değiştirebilir.

2.2- VERİ YÖNETİMİNDE BÜTÜNOLÜRLÜK VE PAYLAŞILIRLIK

Veri yönetimi sistemi verilere record veya field ulaşımında esneklik sağlamakla kalmaz aynı zamanda bu yapıyı kullanan programların geliştirilmesi ve çoklu paylaşım konusunda da aynı esnekliği getirir. Bu esnekliğin oluşturulmasında kullanıcıların istekleri büyük rol oynar. Örneğin: Bir file'ın okunması , güncelleştirilmesi ve hangi kullanıcılar tarafından paylaşılacağı aşağıda açıklanan bazı parametrelerin belirtilmesi ile mümkün olacaktır. Bunlar sırası ile :

- 1- File kitlenmesi (veya yok edilmesi)
- 2- Record kitlenmesi (veya yok edilmesi)
- 3- File kaydı şeklindedir.

2.3- TASARIM

Bu veri tabanı yönetiminde formlar 4 ana tablo altında

tutulmuşlardır.Bu tablolar :

1-Ana tablo :Hastaların kayıtlarını tutulduğu tablolardır.Bu tablolara girişler direk olarak yada kodlanarak yapılmaktadır.Her form bir tabloya karşılık düşürüldüğünden 12 tablo üzerinden araştırma yapılmaktadır.

2-Tablo içeriklerinin saklandığı tablo : Bu tabloda hastaların kayıtlarının tutulduğu tablo isimleri , kolon sayıları saklanır.Burada amaç alt seçenekli verilerin tümünün bir yerde saklanması olduğundan bu verilere ulaşmak için adres yöntemini kullanmaktır.Böylece her tablonun kolon sayısı ve her kolonun var ise alt seçenekleri ayrı , ayrı iki tabloda toplu olarak saklandığından bu tablolara ancak istenen verilerin adresleri yolu ile erişilebilir.

3-Kolon içeriklerinin saklandığı tablo : Bu tabloda ise her tablonun kolonlarının isimleri ,tipleri, uzunlukları , adresleri saklanır.Tablolardaki adresler o verinin bir sonraki tabloda bulunduğu yeri göstermektedir.Kolonlara içerikleri yukarıda açıkladığımız veri guruplarına göre direk veya kodlu olarak erişilir.Kodlama ancak veri çoktan seçenekli ise olmaktadır.Bu seçeneklerde kendi içinde büyüyebilme özelliği göstermektedir.Dolayısı ile bu kodların hangi seçeneğe karşılık düştüğü diğer bir alt kolon tablosuna bağlanılarak anlaşılabilir.

4- Alt kolon içeriklerinin saklandığı tablo : Kodlu verilerin saklandığı bir tablodur.Bu tabloda veriler sistemin tasarlanması sırasında saptanan boş alanlar ile büyüyebilme özelliğine sahiptirler.Ancak girilmesi gereken nümerik bir değer olursa , kendi başlığı altındaki bir tabloda ayrı olarak tutulur.

SONUÇ

Yapılan analiz öncelikle KOŞUYOLU KALP HASTANESİ hasta-doktor ilişkisi ve sorunlarının öğrenilmesinde yardımcı olmuş bunun yanısıra oluşturulan hasta veri tabanı sistemi ile hastanın girişinden taburcu edilişine kadar geçen sürede hertürlü poliklinik , laboratuvar, koroner bakım , ameliyat ve yoğun bakım işlemi takip edilip , her türlü bilgiye doktorun isteği doğrultusunda erişilmiş ve rapor çıkarılmıştır.Sistemin tasarımı önce IBM PS/2 model 60'da yapılip istenilen sonuca ulaşılmış ve ANKARA'da yapılan 5. ULUSLARARASI TIP KONGRESİNDE bildiri olarak başarı ile sunulmuştur.Ancak hastanenin IBM 6150 sisteme geçmesi ve bu sistemin AIX işletim sistemi altında çalışması, tasarlanan sistemin yeniden C dili ile yazılmasını ve farklı bir veri yönetim metodu tasarlanmasını gerektirmiştir.Bu çalışma halen devam etmektedir.


```

#include "mpanel.h"
#include <dms.h>
#define ISIMINDEX "isimindex"
#define SIRAINDEX "siraindex"
#define ISIM      20
#define TIPI      2
#define DURUMU    1

/* FIELD'LARIN TARIFI */
struct tabloi {
    DMS__VARCHAR isim[ISIM+1];
    DMS__INTEGER kolonsayi;
    DMS__COLUMN columns[2];
} tablo = {"",0,
    {
        { "isim", DMSVARCHAR, ISIM, tablo.isim },
        { "kolonsayi", DMSINTEGER, 0, (char *) &tablo.kolonsayi }
    }
};

struct tabloii {
    DMS__VARCHAR isim[ISIM+1];
    DMS__VARCHAR tipi[TIPI+1];
    DMS__INTEGER uzunluk;
    DMS__INTEGER degisken;
    DMS__INTEGER adres;
    DMS__INTEGER buyume;
    DMS__COLUMN columns[6];
} kolon = {"", "", 0, 0, 0, 0,
    {
        { "isim", DMSVARCHAR, ISIM, kolon.isim },
        { "tipi", DMSVARCHAR, TIPI, kolon.tipi },
        { "uzunluk", DMSINTEGER, 0, (char *) &kolon.uzunluk },
        { "degisken", DMSINTEGER, 0, (char *) &kolon.degisken },
        { "buyume", DMSINTEGER, 0, (char *) &kolon.buyume },
        { "adres", DMSINTEGER, 0, (char *) &kolon.adres }
    }
};

struct tabloiii {
    DMS__VARCHAR isim[ISIM+1];
    DMS__VARCHAR durumu[DURUMU+1];
    DMS__INTEGER sayisi;
    DMS__VARCHAR tipi[TIPI+1];
    DMS__INTEGER uzunluk;
    DMS__COLUMN columns[5];
} altkolon = {"", "", 0, "", 0,
    {
        { "isim", DMSVARCHAR, ISIM, altkolon.isim },
        { "durumu", DMSVARCHAR, DURUMU, altkolon.durumu },
        { "sayisi", DMSINTEGER, 0, (char *) &altkolon.sayisi },
        { "tipi", DMSVARCHAR, TIPI, altkolon.tipi },
        { "uzunluk", DMSINTEGER, 0, (char *) &altkolon.uzunluk }
    }
};

```



```

#include <cur01.h>
#include <cur00.h>
#include <cur04.h>
#include <cur02.h>
#include <cur05.h>
#include <cur03.h>
int ins=0; /* insert off */

struct RDBOX {
    int x,y;
    int len;
    char type;
    char str[21];
    char st[30];
} rdbox[100];

#include <ctype.h>
#include <stdio.h>
#include <string.h>
#undef KEY_ENTER
#define KEY_ENTER 0x000a

int panel();

int panel(rdbox,n)
struct RDBOX rdbox[];
int n;
{
    register char *p,*r;
    register struct RDBOX *q;
    int i,k,m,len,key,dp=0,x,y,type;
    for(k=0; k<n; k++) {
        q=& rdbox[k];
        colorout(NORMAL);
        mvaddstr(q->x,2,q->st);
        mvaddstr(q->x,q->y,q->str);
        mvchgat(q->x,q->y,q->len,REVERSE);
    }
top:
    q=& rdbox[dp];
    k=0;
    x=q->x; y=q->y;
    colorout(NORMAL);
    if(q->type == 'n') {
        type=1;
        mvaddstr(0,10,"Numeric");
    }
    else {
        type=0;
        mvaddstr(0,10,"");
    }
    colorout(REVERSE);
    len=q->len;
    p=q->str;
    m=strlen(p);
    mvaddstr(x,y,p);

```



```

refresh();
key=getch();
switch(key) {
    case KEY_LEFT:
        if(k == 0) goto previous_field;
        k--; y--;
        goto label1;
    case KEY_RIGHT:
        if(++k == len) goto next_field;
        y++;
        if(k > m) {
            p[m++]=' ';
            p[m]='\0';
        }
        goto label1;
    case KEY_DC:
        if(k != m) {
            m--;
            r=p+k;
            strcpy(r,r+1);
            addstr(r);
            addch(' ');
            goto label1;
        }
        break;
    case KEY_IC:
        ins=~ins;
        colorout(NORMAL);
        if(ins) mvaddstr(0,0,"Insert");
        else mvaddstr(0,0,"");
        colorout(REVERSE);
        goto label1;
    case KEY_BTAB:
        goto previous_field;
    case KEY_NPAGE:
        return key;
    case KEY_PPAGE:
        return key;
    case KEY_ENTER:
        goto next_field;
    case KEY_TAB:
        goto next_field;
    case KEY_BACKSPACE:
        if(k == 0) break;
        strcpy(&p[k-1],&p[k]);
        k--; m--; y--;
        mvaddstr(x,y,&p[k]);

```



```

        addch(' ');
        break;
default:
    if(isprint(key) || isspace(key)) {
        if(type != 0) {
            if(!isdigit(key)) {
                beep();
                goto label1;
            }
        }
        if(ins != 0) {
            if(m == len) beep();
            else {
                for(i=m; i>k; i--) p[i]=p[i-1];
                m++;
                p[k]=(char)key;
                addstr(&p[k]);
                y++; k++;
                if(k == len) goto next_field;
            }
            goto label1;
        }
        else {
            p[k]=(char)key;
            if(k == m) p[++m]='\0';
            addch((char)key);
            goto label5;
        }
    }
    beep();
} /*switch*/
goto label1;
next_field:    if(++dp == n) dp=0;
               goto top;
previous_field: if(--dp < 0) dp=n-1;
               goto top;
}

mpanel(filename)
char *filename;

```

```

{
    FILE *fp;
    struct RDBOX *p;
    int i,n;
    initscr();
    noecho();
    keypad(TRUE);

```



```

#include <stdio.h>
#include <conio.h>
#define MAX 100
main()
{
    drawbox(stdscr,1,0,23,80);refresh();
    colorout(REVERSE);
    fp=fopen(filename,"r");
    if(fp == NULL) {
        printf("File %s not found\n",filename);
        endwin();
        exit(0);
    }
    for(n=0 ; ; n++) {
        p=& rdbox[n];
        fscanf(fp,"%d %d %c %d",& p->x,& p->y,& p->type,& p->len);
        fgets(p->st,30,fp);
        p->st[strlen(p->st)-1]=0;
        if(feof(fp)) break;
    };
    fclose(fp);
    panel(rdbox,n);
    endwin();
}

```



```

#include "datamng.h"
DMS__COLUMN *columns;
#define EMPINDEX "empindex"
main ()

{

    register int    ret;
    register int    tabdes;
    char    response[30];
    char    qualifier[100];
    char    orderby[100];
    char    path[DMSMAXPATH];
    char *strcpy(),linker;
    static char *files[] = {"file1.data","file2.data","file3.data"};
    int len,empno=0,m;
    float degisim;
    int    resp;
    DMS__IDEF order;
    DMS__IDEF *orderptr;

    int termerr ();                /* HATA BULMA */

    /* DATA MANAGEMENT FONKSIYON DEKLARASYONU*/
    int dmscmit ();
    int dmsdelt ();
    int dmsinit ();
    int dmsisrt ();
    int dmsferr ();
    int dmsftch ();
    int dmsslct ();
    int dmstclo ();
    int dmstopn ();
    int dmsterm ();
    int dmsupdt ();

    /* AIX Operating System FONKSIYONLARININ DEKLARASYONU */
    double atof ();
    long    atol ();
    char    *gets ();
    char    *strcpy ();

    /* TABLE YARATMA */

    void crtable ();
    void yazek    ();
    void yazpane ();
    void yazfile ();

    /* DATA MANAGEMENT'I TARIFLEME */
    (void) dmsinit ();
    /* PATHNAME KULLANMA */
    printf ("\n path name ==> ");

```



```
(void) gets (path);
```

```
printf ("kacinci tip tabloyu yaratacaksiniz?");
scanf ("%d",&resp); getchar();
switch (resp) {
case 1:
    columns= tablo.columns;
    len=sizeof(tablo.columns)/sizeof(DMS__COLUMN);
    break;
case 2:
    columns= kolon.columns;
    len=sizeof(kolon.columns)/sizeof(DMS__COLUMN);
    break;
case 3:
    columns= altkolon.columns;
    len=sizeof(altkolon.columns)/sizeof(DMS__COLUMN);
};
```

```
for (;;) {
```

```
printf ("\n(c)reate , (a)dd (s)elect (q)uit ");
gets (response);
switch (response[0]) {
case 's':
    if ((tabdes = dmstopn (path, DMSUPSHR)) < 0) {
        printf ("\n Can't open %s: tabdes = %d", path, tabdes );
        exit (4);
    }
```

```
    /* HATA DURUMU */
    (void) dmsferr (tabdes, DMS_INVALID, termerr);
    (void) dmsferr (tabdes, DMS_RECLEN, termerr);
    (void) dmsferr (tabdes, DMS_BDMSF, termerr);
    (void) dmsferr (tabdes, DMS_IO, termerr);

    /* SATIR CAGIRMA */
    /* SART KOSMA */
    printf ("\nKriterinizi giriniz      ==> ");
    (void) gets (qualifier);
    printf ("\n'ORDER BY' istegi <ret> ==> ");
    (void) gets (orderby);
    if (orderby[0] == '\0')
        orderptr = NULL;
    else {
        order.numkey = 1;
        (void) strcpy (order.keydef[0].colname, orderby);
        order.keydef[0].keyorder = DMSASC;
        orderptr = &order;
    }
```

```
    /* SATIR SECME */
    if ((ret = dmsslct (tabdes, "*", qualifier, orderptr)) > 0) {
        for (;;) {
```



```

if ((ret = dmsftch (tabdes, columns)) > 0) {

    yazek(len);

    /* KARAR VERME */
    printf ("\n(u)pdate, (d)elete, (n)ext, or (q)uit ==> ");
    (void) gets (response);
    switch (response[0]) {
        case 'u':
            /* GUNCELLESTIRME */
            yazpane(len);
            printf("cekilecek isim %s",files[resp-1]);
            mpanel (files[resp-1]);
            yazfile(len);
            if ((ret = dmsupdt (tabdes, columns)) == 0) {
                yazek(len);
            }
            else
                printf ("\nError in dmsupdt: ret = %d", ret );
            continue;
        case 'd':
            /* SATIR SILME */
            if ((ret = dmsdelt (tabdes)) == 0)
                printf ("\n ROW DELETED");
            else
                printf ("\n dmsdelt error: ret = %d", ret);
            continue;
        case 'n':
            break;
            continue;
        case 'q':
            break;
            continue;
        default:
            printf ("\n ***invalid option %s", response);
            continue;
    }
    break;
}
else {
    printf ("\n Out of rows! ret = %d", ret);
    break;
}
}
else
    printf ("\n dmsslct error: ret = %d", ret);
(void) dmsscmitt ();
(void) dmstclo (tabdes);
continue;

case 'a':
    /* YENI KISI GIRME */
    if ((tabdes = dmstopn (path, DMSUPSHR)) < 0) {
        printf ("\n Tablo acilamadi %s: tabdes = %d", path, tabdes

```



```

);      exit (4);
}
/* HATA TARAMASI */
(void) dmsferr (tabdes, DMS_INVALID, termerr);
(void) dmsferr (tabdes, DMS_RECLEN, termerr);
(void) dmsferr (tabdes, DMS_BDMSF, termerr);
(void) dmsferr (tabdes, DMS_IO, termerr);

for(m=0; m<len; m++) *rdbox[m].str=0;
mpanel (files[resp-1]);
yazfile(len);

/* YENI SATIR EKLE */
if ((ret = dmsisrt (tabdes, columns)) == 0) {

    yazek(len);

}
else
    printf ("\n Error in dmsisrt: ret = %d", ret);
(void) dmscmit ();
(void) dmstclo (tabdes);
continue;

case 'c':

    /* yeni tablo yaratma */
    crtable (path,len);
    continue;

case 'q':
    break;

default:
    printf ("\n***invalid option %s.", response);
    continue;
}
break;
}

(void) dmstclo (tabdes);

(void) dmsterm ();
}

/* tablo ve indexi yaratma */

void crtable (path,len)
char *path;
int len;

```



```

{
    int    ret;
    DMS__IDEF idef;
    DMS__DIR dmsdir;

    int dmstcre ();
    int dmsicre ();
    int dmsgten ();
    int dmschen ();
    char *strcpy ();
    printf(" crlen=%d",len);

    if ((ret = dmstcre (path, DMSOWNR ; DMSOWNW, len, columns)) == 0)
        printf("\n%s tablo yaratildi",path);
    else
        printf ("\nHata var dmstcre: ret = %d", ret);
    return;
}

/*FUNCTION ERROR */

int termerr (error)

int error;

{

int dmstern ();

printf ("\nError returned was %d; terminating\n", error);
(void) dmstern ();
exit (4);

}

/* VERININ EKRANA YAZDIRILMASI */

void yazek(len)

int len;

{
    int m;

    for (m=0;m < len ; m++) {

        switch(columns[m].type) {

            case DMSVARCHAR :

                printf("%s,",columns[m].valptr);
                break;
            case DMSINTEGER :

                printf("%d,",*((int *)columns[m].valptr));
                break;
            case DMSFLOAT :

                printf("%f,",*((float *)columns[m].valptr));

```



```

        } /* switch kapama */
        } /* loop kapama */
    }

    /* PANELE VERIYI YAZDIRMA */
    void yazpane(len)
    int len;
    {
        int m;

        for (m=0; m < len; m++) {
            switch(columns[m].type) {

                case DMSVARCHAR:
                    strcpy(rdbbox[m].str, columns[m].valptr);
                    break;

                case DMSFLOAT:
                    sprintf(rdbbox[m].str, "%f", columns[m].valptr);
                    break;

                case DMSINTEGER:
                    sprintf(rdbbox[m].str, "%d", columns[m].valptr);
                    break;
            }
        }

        for (m=0; m<len; m++) {
            switch(columns[m].type) {

                case DMSVARCHAR:
                    strcpy(rdbbox[m].str, columns[m].valptr );
                    break;

                case DMSINTEGER:
                    sprintf(rdbbox[m].str, "%d", *((int *)columns[m].valptr) );
                    break;

                case DMSFLOAT:
                    sprintf(rdbbox[m].str, "%f", *((float *)columns[m].valptr) );
            }
        }
    }

```



```

        break;
    }
}

/* FILE A VERININ YAZDIRILMASI */

void yazfile(len)
int len;
{
    int m;
    for (m=0;m < len ; m++) {
        switch(columns[m].type) {
            case DMSVARCHAR:
                strcpy(columns[m].valptr,rdbox[m].str);
                break;
            case DMSFLOAT:
                sscanf(rdbox[m].str,"%f",columns[m].valptr);
                break;
            case DMSINTEGER:
                sscanf(rdbox[m].str,"%d",columns[m].valptr);
                break;
        }
    }
};

```


EPIKRİZ ÇIKARILMASI

DOSYA NO :1383/88

ADI:SAMI

SOYADI:GUNDUZ

YASI 50

DOGUM YERI:BULGARISTAN

CINSIYETI:E

MEDENI DURUMU:E

KAN GURUBU:0 +

KILO: 72

BOY: 2

MESLEGI:EMEKLI

BAGLI OLDUGU KURUM:EMEKLI SANDIGI

ADRESI:ACEMLER CAD.DEGIRMEN SOK.NO/3 BURSA

YATIS TARİHİ:08/22/88

KACINCI YATISI: 1

GELDİĞİ YER:KVC

YAKININ ADI:HANDAN

YAKINININ SOYADI:GUNDUZ

YASI: 22

MESLEGI:TIP OGRENCISI

ADRESİ:MEVLANA KAPI CAD.NO 22 CAPA

TELEFONU: 5851288

DERECESİ:KIZI

BASVURULACAK KISI ADIHANDAN

SOYADGUNDUZ

ADRESİ:MEVLANA KAPI CAD.NO 22 CAPA

TELEFONU 5851288

GOGUS AGRISI

SEMPTOMLARIN SURESI 1

SEMPTOMLARDA SİDDETLENME 1

RETROSTERNAL

SIKISTIRICI

YANICI

BASKI TARZINDA

SOL KOL

SAG KOL

AGRI SURESI 1

EFORLA

STRESSLE

AKTIVITE

GECE GELEN

DINLENMEK

NİTROGLİSERİN

YASTIK 1

PEKTORIS 1

ANEVREZMEKTOMI YAPILAN OLGULARDA İNFARKTUS GECİREN

11:54 1989 scr.out Page 1

INFARKTUS GECIRMIS HASTALARDAN ANEVREZMEKTOMI
UYGULANANLARIN

DOSYANO	ADI	SOYADI
1386/88	IHSAN	KUTOGLU
1236/88	OKTAY	KARAGOZOGLU
481/88	ARAP	KUTLU
552/88	NECMI	ALDUZ

TOPLAM==> 4 KISI

ANEVREZMEKTOMI YAPILAN OLGULARDA INFARKTUS GECIREN
HASTALARIN

DOSYANO	ADI	SOYADI
1386/88	IHSAN	KUTOGLU
1236/88	OKTAY	KARAGOZOGLU
481/88	ARAP	KUTLU
552/88	NECMI	ALDUZ

ANEVREZMEKTOMI GECIRENLERIN SAYISI 5
BUNLARIN ICINDEN INFARKTUS GECIREN HASTALARIN SAYISI 4
INFARKTUS ORANI = % 80

ANEVREZMEKTOMI YAPILAN OLGULARDA INFARKTUS GECIREN
HASTALARDAN SIGARA ICENLERIN

DOSYANO	ADI	SOYADI
481/88	ARAP	KUTLU
552/88	NECMI	ALDUZ

ANEVREZMEKTOMI YAPILAN OLGULARDA INFARKTUS GECIREN
HASTALARDAN SIGARA ICENLERIN SAYISI 2
ORANI = % 50

KAYNAKLAR

- MEDICAL PHYSICS

John R.Cameron

James G.Scofronick

- PRINCIPLES OF DATABASE MANAGEMENT

James Martin (IBM systems research)

- COMPUTER DATA-BASE ORGANIZATION

James Martin

- SOFTWARE ENGINEERING

Roger S.Pressman (1987)

- DATA BASE MANAGEMENT

Fred R.McFadden and Jeffrey A.Hoffer

ÖZGEÇMİŞ

Şebnem TOKCAN ,1965'te Ankara'da doğdu. Liseyi Özel Şişli Terakki Lisesi'nde bitirdi. 1986 yılında Yıldız Üniversitesi Bilgisayar Bilimleri Mühendisliği Bölümü'nden mezun oldu. Ocak 1987 'den bu yana mezun olduğu bölümün Yazılım Anabilim Dalı'nda Araştırma Görevlisi olarak çalışmaktadır.



