

126913

**BİLGİ SİSTEMLERİNİN ÇÖZÜMLENMESİ
(JEODEZİK BİLGİ SİSTEMİ)**

YÜKSEK LİSANS TEZİ
Müh. Yalkın ÇAĞLAR
501991046

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

Tezin Enstitüye Verildiği Tarih: 02 Ocak 2002
Tezin Savunulduğu Tarih: 22 Ocak 2002

Tez Danışmanı : Prof. Dr. Orhan ALTAN Y.

Prof. Dr. Sıtkı KÜLÜR

Diğer Jüri Üyeleri Prof. Dr. Gönül TOZ (İ.T.Ü)

Prof. Dr. Ahmet YAŞAYAN (Y.T.Ü.)

OCAK 2002

126913

İÇİNDEKİLER

KISALTMALAR	v
ŞEKİL LİSTESİ	vi
ÖZET	vii
SUMMARY	x
1. GİRİŞ	1
2. BİLGİ SİSTEMİ KAVRAMI	3
3. BİLGİ SİSTEMİN KURULUŞLARDAKİ ROLÜ	6
3.1. Bilgi Sistemi Kullanımı İçin Kuruluş ve Çevre Şartları	6
3.2. Bilgi Sistemi Hayat Çarkı	8
3.3. Veri Tabanı Uygulama Sistemi Hayat Çarkı	9
4. VERİ TABANI TASARIM SÜRECİ	11
4.1. Aşama 1: Gereksinimlerin Derlenmesi ve Çözümlemesi	12
4.1.1. Veri Akış Çizgeleri	15
4.2. Aşama 2: Kavramsal Veri tabanı Tasarımı	17
4.2.1. Aşama 2a: Kavramsal Şema Tasarımı	17
4.2.1.1. Veri Modelleri	18
4.2.1.2. Kavramsal Şema Tasarımına Yaklaşımlar	21
4.2.1.3. Şema Tasarımı İçin Stratejiler	23
4.2.1.4. Şema (Görünüm) Birleştirme	25
4.2.2. Aşama 2b: Veri Tabanı İşlemi (transaction) Tasarımı	28
4.3. Aşama 3: Veri Tabanı Yönetim Sisteminin Seçimi	30
4.4. Aşama 4: Veri Modeli Dönüşümü (Mantıksal Veri Tabanı Tasarımı)	32
4.5. Aşama 5: Fiziksel Veri Tabanı Tasarımı	33
4.6. Aşama 6: Veri Tabanı Sisteminin Uygulanması	34
5. COĞRAFİ BİLGİ SİSTEMİ KAVRAMI	36
5.1. Coğrafi Bilgi Sisteminin Bileşenleri	38
5.1.1. Coğrafi Veriler	38
5.1.2. Coğrafi Bilgi Sistemi Donanımı	39
5.1.2.1. Veri Giriş Elemanları	40
5.1.2.2. Veri Depolama ve İşleme Elemanları	40
5.1.2.3. Veri Sunuş Elemanları	40
5.1.3. Coğrafi Bilgi Sistemi Yazılımı	40
5.1.3.1. Coğrafi Veri Girişi ve Veri İşleme Yazılımları	42
5.1.3.2. Coğrafi Veri Depolama, Erişim ve Coğrafi Veri Tabanı Yönetimi Yazılımları	44
5.1.3.3. Coğrafi Veri Sorgulama ve Çözümleme Yazılımları	44
5.1.3.4. Coğrafi Veri Sunuş (Görüntüleme ve Sonuç Ürünler Hazırlama) Yazılımları	45

5.1.3.5. Kullanıcı Arabirimi	46
5.1.4. Coğrafi Bilgi Sistemi Personeli	46
5.1.5. Organizasyon (Yöntemler)	47
6. UYGULAMA (JEODEZİK BİLGİ SİSTEMİ)	49
6.1. Aşama 1: Gereksinimlerin Derlenmesi ve Çözümlemesi	49
6.2. Aşama 2: Kavramsal Veri tabanı Tasarımı	50
6.3. Aşama 3: Veri Tabanı Yönetim Sisteminin Seçimi	52
6.4. Aşama 4: Veri Modeli Dönüşümü (Mantıksal Veri Tabanı Tasarımı)	52
6.5. Aşama 5: Fiziksel Veri Tabanı Tasarımı	54
6.6. Aşama 6: Veri Tabanı Sisteminin Uygulanması	58
7. SONUÇ	63
KAYNAKLAR	67
EKLER	70
ÖZGEÇMİŞ	77



KISALTMALAR

4GL	: Fourth Generation Language
BS	: Bilgi Sistemi
BKİ	: Bilgi Kaynağı İdaresi
CBS	: Coğrafi Bilgi Sistemleri
CPU	: Central Processing Unit
CVT	: Coğrafi Veri Tabanı
DDL	: Data Definition Language
DEC	: Digital Equipment Corporation
DML	: Data Manipulation Language
DVTYS	: Dağıtık Veri Tabanı Yönetim Sistemi
f.k.	: foreign key
GVI	: Gelişmiş Varlık – İlişki
JBS	: Jeodezik Bilgi Sistemi
SDL	: Storage Definition Language
SGİÇ	: Sıradüzensel Girdi İşlem Çıktı
STD / STÇ	: Sorun Tanımlama Dili / Sorun Tanımlama Çözümleyicisi
VAC	: Veri Akış Çizgeleri
vd	: ve diğerleri
VI	: Varlık – İlişki
VTS	: Veri Tabanı Sistemi
VTY	: Veri Tabanı Yöneticisi
VTYS	: Veri Tabanı Yönetim Sistemi
YÇTY	: Yapısal Çözümleme ve Tasarım Yordamı

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1 : Anlam Seviyesine Göre Veri, Bilgi, Birikim	4
Şekil 4.1 : Büyük Veri Tabanları İçin Veri Tabanı Tasarım Aşamaları	14
Şekil 4.2 : Bilgi Akış Modeli	16
Şekil 4.3 : Bilgi Akış İyileştirmesi	17
Şekil 4.4 : Sıradüzensel Veri Modeli.....	20
Şekil 4.5 : Ağ Veri Modeli	20
Şekil 4.6 : İlişkisel Veri Modeli	20
Şekil 4.7 : Nesneye Yönelik Veri Modeli	21
Şekil 4.8 : Varlık - İlişki Gösterim İşaretleri Özeti	22
Şekil 4.9 : Varlık – İlişki Veri Modeli	23
Şekil 4.10 : Yukardan – Aşağı İyileştirme Örnekleri. (a) Yeni Bir Varlık Türü Türetme. (b) Bir Varlık Türünü İki Varlık Türü ve Bir İlişki Olarak Ayrıştırma	25
Şekil 4.11 : Aşağıdan – yukarı İyileştirme Örneği. (Yeni Bir İlişkinin Bulunması ve Eklenmesi)	26
Şekil 4.12 : Birleştirmeden Önce Uyuşum İçin Görünümlerin Düzeltilmesi ..	27
Şekil 4.13 : Görünüm Birleştirmesi İçin Farklı Stratejiler	29
Şekil 5.1 : CBS Bileşenleri	38
Şekil 5.2 : CBS Donanım Bileşenleri	39
Şekil 5.3 : CBS Yazılım Bileşenleri	42
Şekil 5.4 : CBS Yazılım Bileşeni	45
Şekil 6.1 : JBS Veri Akış Çizgeleri (nirengi).....	51
Şekil 6.2 : Bütünlük Koşullarını İçeren Bir İlişkisel Veri Tabanı Şeması	53
Şekil 6.3a : Tablolar	54
Şekil 6.3b : Belirler Tablosu Tasarım Görünümü	54
Şekil 6.3c : Kot Türü Tablosu Tasarım Görünümü	55
Şekil 6.3d : Personel Tablosu Tasarım Görünümü	55
Şekil 6.3e : Nirengi Tablosu Tasarım Görünümü	56
Şekil 6.3f : Tesis Türü Tablosu Tasarım Görünümü	56
Şekil 6.3g : Ölçü Tablosu Tasarım Görünümü	57
Şekil 6.3h : Protokol Tablosu Tasarım Görünümü	57
Şekil 6.3i : Fiziksel Tasarım	58

ÖZET

Günümüzün teknolojik harikası olan bilgisayarın yalnızca bir hesaplama aracı olmayıp bir bilgi işlem ve bir bilgi iletişim aracı olduğu bir gerçektir. Genelde bilgi işlem olarak adlandırılan etkinliklerin hammaddesinin VERİ, ürününün BİLGİ (işlenmiş veri) olduğu da göz önüne alınırsa veri ve bilgilerin organize edilmesi sorunu ön plana çıkmaktadır. Bilgi Sistemi (BS) ve Veri Tabanı Sistemi (VTS) bu amaçla geliştirilmiştir. Veri Tabanı teknolojisi bir veri organizasyonunun bütün olarak işlenmesini ve farklı uygulamalar için kullanıcıların verilere daha ekonomik bir biçimde ulaşmasını sağlar. Coğrafi varlıklara ait veri toplanması arazide yapılan ölçümlerle olur. Bu değerler doğrulanır, işlenir ve bir bütün olarak organize edilerek çok çeşitli amaçlar için kullanılır. Jeodezik Bilgi Sistemi (JBS)'nin temelini teşkil eden bu verilerin elde edilmesi uzun zaman aldığı gibi önemli bir harcamayı da gerektirmektedir.

Bu çalışmada JBS' nin oluşturulmasında, veri tipleri ve sistem içindeki mantıksal akışından yaralanarak sistemin alt birimlere ayrılmasını esas alan ve sistemin mantıksal olarak organizasyonunu; veri akışının mantığı ile sistemi oluşturan alt birimler arasındaki işlevsel ilişkiye bağlı olarak gerçekleştiren veri akışı – güdümlü yöntembilimi izlenmiştir.

Kuruluşlarda verinin hacmi ve ortak kullanım gereksinimi vb. arttıkça bilgilerin bir merkezden yönetilme ve dosya sisteminden veri tabanı sistemine geçiş gereksinimi de artmıştır.

Genellikle bilgi sistemi hayat çarkına **makro hayat çarkı**, veri tabanı sistemi hayat çarkına ise **mikro hayat çarkı**, adı verilir. Tipik bir **makro hayat çarkı**, şu aşamaları içerir:

1. Olurluk çözümlemesi (analizi): Olası uygulama alanlarının çözümlenmesini, ön kar – maliyet çalışmalarını ve uygulamalar arası önceliklerin belirlenmesini içerir.
2. Gereksinimlerin belirlenmesi ve çözümlenmesi: Gereksinimlerin ayrıntıları, kullanıcılarla yapılan görüşmelerle derlenir.
3. Tasarım: Bu aşama iki dalda ilerler. Veri tabanının tasarımı ve bu veri tabanını kullanıp, işleyen uygulama sistemlerinin (yazılımlarının) tasarımı
4. Uygulama: Bilgi sistemi uygulanır, veri tabanı yüklenir, veri tabanı işlemleri uygulanır ve denenir.
5. Değerlendirme ve kabul testi: Sistem, kullanıcı gereksinimlerini karşılama ve başarımlı ölçütleri (kriterleri) anlamında değerlendirilir. Sistem, başarımlı ölçütlerine ve davranış şartlarına karşı denenir.
6. Kullanma: Bundan önce, eski sistemdeki kullanıcıların dönüşümü ve kullanıcı eğitimi gelebilir. Kullanma aşaması tüm sistem işlevleri kullanılabilir olduğunda başlar. Yeni gereksinim ve uygulamalar türedikçe sisteme katılıp kullanılmaya başlamadan önce, baştaki tüm aşamalardan

geçer. Sistem başarımının gözlenmesi ve sistem bakımı kullanma aşamasının önemli etkinlikleridir.

Veri Tabanı Uygulama Sistemi (mikro) hayat çarkı ile ilgili etkinlikler aşağıdaki aşamaları içerir:

1. Sistem tanımlaması: Veri tabanı sisteminin kapsamı, kullanıcıları ve onun uygulamaları tanımlanır.
2. Tasarım: Bu aşamanın sonunda, seçilmiş veri tabanı yönetim sistemi üzerinde tüm mantıksal ve fiziksel tasarım hazır hale gelmiş olur.
 - a. Gereksinimlerin derlenmesi ve çözümlenmesi: Bu aşamada son kullanıcılar arasından seçilen personel ile yapılan görüşme ve anketlere dayanarak Veri Akış Çizgeleri (VİÇ) hazırlanır.
 - b. Kavramsal veri tabanı tasarımı: Bu aşamada Varlık – İlişki (Vİ) çizgeleri hazırlanır. Sorgulamalar planlanır.
 - c. Veri tabanı yönetim sisteminin seçimi: Tanımlı ölçütler değerlendirilerek en uygun veri tabanı yönetim sistemi seçilir.
 - d. Veri modeli dönüşümü (mantıksal veri tabanı tasarımı olarak da adlandırılır): Vİ çizgeleri veri tabanı tablolarına dönüştürülür.
 - e. Fiziksel veri tabanı tasarımı: Veri tabanı dosyalarındaki depolama yapıları ve erişim yollarının tasarlandığı aşamadır.
3. Uygulama: Bu aşama kavramsal, dış ve iç veri tabanı tanımlarının yazılması, boş veri tabanı dosyalarının yaratılması ve yazılımların uygulanmasını içerir.
4. Yükleme veya veri dönüşümü: Veri tabanı, ya verilerin doğrudan yüklenmesi ya da hazır dosyaların veri tabanı formatına dönüştürülmesi ile doldurulur.
5. Uygulama dönüşümü: Önceki sistemden kalma tüm yazılım uygulamaları yeni sisteme dönüştürülür.
6. Test ve değerlendirme: Yeni sistem denenir ve değerlendirilir.
7. Kullanım: Veri tabanı sistemi ve uygulamaları kullanıma konur.
8. İzleme ve bakım: Kullanım aşaması boyunca sistem sürekli izlenir ve bakımı yapılır. Hem veri içeriğinde hem de yazılım uygulamalarında büyüme ve genişleme olabilir. Zaman zaman büyük değişiklikler ve yeniden yapılanmalar gerekebilir.

2, 3 ve 4 etkinlikleri daha büyük olan bilgi sistemin tasarım ve uygulama aşamalarının parçasıdır.

Coğrafi bilgi sistemleri (CBS) için kısaca “bilgi sistemlerinin coğrafi veriye uygulanmış özel bir formudur” denilebilir.

Bir CBS’ nin kurulabilmesi ve kendinden beklenen işlevleri yerine getirebilmesi için gerekli olan temel elemanlar şunlardır:

1. Coğrafi Veriler
2. Coğrafi Bilgi Sistemi Donanımı
 - a. Veri Giriş Elemanları
 - b. Veri Depolama ve İşleme Elemanları
 - c. Veri Sunuş Elemanları
3. Coğrafi Bilgi Sistemi Yazılımı
 - a. Coğrafi Veri Girişi ve Veri İşleme Yazılımları
 - b. Coğrafi Veri Depolama, Erişim ve Coğrafi Veri Tabanı Yönetimi Yazılımları
 - c. Coğrafi Veri Sorgulama ve Çözümleme Yazılımları

- d. Coğrafi Veri Sunuş (Görüntüleme ve Sonuç Ürünler Hazırlama) Yazılımları
- e. Kullanıcı Arabirimi
4. Coğrafi Bilgi Sistemi Personeli
5. Organizasyon (Yöntemler)

Bu çalışmada bu aşamalar JBS için ele alınıp onun önemli bir parçasını oluşturan nirengi ile ilgili bölüm için uygulaması yapılmıştır. Bu uygulama sonucunda şunlar görülmüştür:

1. Çözümlemenin başında kapsamın çok geniş tutulması ve bunun için yeterli personel ve zamanın iyi planlanmaması, sistem çözümleme uygulamasının sonuçlandırılmama riskini arttırmaktadır. Bunu engellemek için sistem mutlaka alt sistemlere ayrılmalı ve öncelikle bu alt sistemlerdeki hedefler konularak bunlara ulaşılmaya çalışılmalıdır. Daha sonra tüm ara hedeflere ulaşırsa ana hedefe yönelmelidir.
2. Ulaşılmış bir ara hedef olan nirengi verilerinin bir VTS içerisinde tutularak kullanıcılara sunulması sırasında şu zorluklarla karşılaşmıştır:
 - a. Sistem çözümlemesinin doğasında olan geri beslemeler (örneğin mantıksal veri tabanı tasarımı aşamasında yeni gereksinimlerin ortaya çıkması) fazlalaştığında geriye dönmekten ilerleme sağlanamamakta bu da çözümçüler arasında bıkkınlık ve yılgınlığa neden olmaktadır. Bunu önlemek için baştan sistemin sınırlarının ve ara hedeflere ulaşmadaki zamanların iyi belirlenmesi gerekmektedir.
 - b. Son kullanıcılarla çözümçüler çok farklı alanlardan geldiğinde bir iletişim kopukluğu ortaya çıkmakta, bu da mevcut sorunların teşhisinde yanlışlıklara neden olmaktadır. Bunu önlemek için bu iki grubun mümkün olduğunca birbirlerini anlayabilecek insanlardan seçilmesi gerekmektedir.
 - c. Çözümlemenin ömrünün uzaması başta kullanılmasına karar verilmiş donanım ve yazılımların demode olmasına ve son gereksinimleri karşılamakta yetersiz kalmasına sebep olmaktadır. Bunu için daha önce de değinildiği gibi uzun zaman alacak mükemmel bir sistem çözümlemesine ulaşmak yerine kısa zamanda ulaşılabilecek ara hedefler belirlenerek ulaşılan her hedefin kullanıma sunulmasının daha uygun olduğu değerlendirilmektedir.
3. Tüm bunların yanı sıra uygulamanın sonuçlarında şu yararlar sağlanmıştır:
 - a. Kullanıcılar en son bilgileri içeren büyük hacimli verilere daha hızlı ve kolay bir şekilde ulaşabilmişlerdir.
 - b. Veriler arasındaki karmaşık ilişkilerin modellenmesi ve verilerin ortak bir alanda korunması ile veri güncellemesi kolaylaşmıştır.
 - c. Çeşitli bilgi kaynakları birleştirilerek karmaşık kararlar almak daha kolay ve hızlı olarak yapılabilmeye başlanmıştır.
 - d. Uygulamalar arası veri bağımsızlığı sağlandığından verilerin bozulması ve kaybolması riski ortadan kaldırılmıştır.
 - e. Veriler arası ilişkiler kurulup bunlardan görünümüler oluşturulduğundan disk hacminden büyük tasarruf sağlanmıştır.
 - f. Yeni uygulama geliştirimi çok hızlı ve ucuza yapılabilir hale gelmiştir.

Sonuç olarak, her sistemin bir gün bilgi sisteme geçişi kaçınılmazdır. Bunun da bir yöntembilim içerisinde yapılması zorunludur.

SUMMARY

It is a fact that computers, today's technological wonder, are not only "computing" tools but also means of information and communication. Considering the raw material of so – called data processing activities usually as DATA and production as INFORMATION, the problem of organizing the data and information arises automatically. Information System (IS) and DataBase System (DBS) are developed in this sense. Database technology, helps a data organization to be processed integrately and different users to attain the data much economically. The data collection of geographical materials is done by measurements in the field. These values are verified, processed and organized all together to be used by various objectives. Obtaining these data which constitutes the basics of Geodetic Information System (GdetIS) is a very time – and – money – consuming activity.

In this study, in the construction of GdetIS, data flow – oriented methodology which examines data types and their flow in the system and which realizes the logical organization of the system together with data flow logic and the functional relations between the subunits of the system is considered.

As the size of the data and the need to share it increased, the need to manage them from a center and to migrate into database system from file system also increased.

The Information System life – cycle is usually called macro life – cycle and Database System life – cycle is called micro life – cycle. The macro life – cycle typically includes the following phases:

1. Feasibility analysis: This is concerned with analyzing potential application areas, performing preliminary cost – benefit studies, and setting up priorities among applications.
2. Requirements collection and analysis: Detailed requirements are collected by interacting with potential users to identify their particular problems and needs.
3. Design: This phase has two aspects: the design of database system, and the design of the application systems (programs) that use and process the database.
4. Implementation: The information system is implemented, the database is loaded, and the database transactions are implemented and tested.
5. Validation and acceptance testing: The acceptability of the system in meeting users' requirements and performance criteria is validated. The system is tested against performance criteria and behavior specifications.
6. Operation: This may be preceded by conversion of users from an older system, as well as by user training. The operational phase starts when all system functions are operational and have been validated. As new requirements or applications crop up, they pass through all the previous phases until they are validated and incorporated into the system. Monitoring

of system performance and system maintenance are important activities during the operational phase.

Activities related to the Database Application System (Micro) Life – Cycle include the following phases:

1. System definition: The scope of the database system, its users, and its applications are defined.
2. Design: At the end of this phase, a complete logical and physical design of the database system on the chosen DBMS is ready.
 - a. Requirements collection and analysis: In this phase, based on the interviews held by the chosen end users, Data Flow diagrams (DFDs) are prepared.
 - b. Conceptual database design: In this phase Entity – Relation (ER) diagrams and queries are prepared.
 - c. Choice of a DBMS: Evaluating the defined parameters, the best DBMS is chosen.
 - d. Data model mapping (also called logical database design): ER diagrams are mapped into the chosen DBMS tables.
 - e. Physical database design: This is the phase where storage structures and access path are designed.
3. Implementation: This comprises the process of writing the conceptual, external, and internal database definitions, creating empty database files, and, implementing the software applications.
4. Loading and data conversion: The database is populated either by loading the data directly or by converting existing files into the database system format.
5. Application conversion: Any software applications from a previous system are converted to the new system.
6. Testing and validation: The new system is tested and validated.
7. Operation: The database system and its applications are put into operation.
8. Monitoring and maintenance: During the operational phase, the system is constantly monitored and maintained. Growth and expansion can occur in both data content and software applications. Major modifications and reorganization may be needed from time to time.

Activities 2, 3, and 4, together are part of the design and implementation phases of larger information life – cycle.

Geographical Database Systems (GIS) can simply be defined as “a special form of information system which is implemented to geographical data”.

The basic components for a GIS to be able to be constructed and to perform the expected functions are as follows:

1. Geographical Data
2. GIS Hardware
 - a. Data Entry Components
 - b. Data Storage and Processing Components
 - c. Components
3. GIS software
 - a. Geographical Data Input and Data Processing Software
 - b. Geographical Data Storage, Access and Geographical Database Management Software

- c. Data Querying and Analyzing Software
- d. Geographical Data Presentation (Viewing and Preparing Result Productions) Software
- e. User Interface
- 4. GIS Personnel
- 5. Organization (Methods)

In this study these phases are carried out for GdetIS and is applied to triangulation part which plays significant role in it. In this implementation the following results are drawn:

1. Overrating the scope of analysis and insufficient planning of personnel and time, increases the risk of analysis the process to fade. To prevent this, the system must be divided into subsystems and the objectives for these subsystems must be defined and tried to be achieved. After all intermediate objectives are achieved the main objective should be considered.
2. While achieving the intermediate objective of keeping the triangulation data by a database system and presenting the information over it, the following problems are met:
 - a. If the feedbacks, which comes natural in system analysis (ex: new requirements arise during logical database design), increases too much, it may not be possible to look ahead because of looking back so frequently. This makes the analyzers wearied and daunted. To prevent this, the borders of the system and the arrival times of the intermediate objectives must be defined strictly.
 - b. If the end users and analyzers come from very different areas there happens to be a communication lack between them and this causes misdiagnosis of the present problems. To prevent this, these two groups must be as related as possible.
 - c. Prolonging the analyzing process causes the hardware and software bought to be used at the beginning to be out – of – date. To prevent this, instead of aiming at a perfect system analysis that may take a relatively long time, one must define intermediate objectives to be achieved and put them into use respectively, as stated above.
3. Despite all of the facts mentioned above, the following usages are encountered:
 - a. The users accessed the large volumes of data that produced the latest information in a much easier and economic way.
 - b. The modeling of the complex relations among the data and the updating of the data by keeping them in a shared storage became easier.
 - c. Combining various information resources made complex decisions easier and faster.
 - d. Data independence protects application programs from destructing and losing data.
 - e. By establishing relations among data and forming external views from them vast amount of disk space is saved.
 - f. Developing new applications became very fast and cheap.

As a result, the transition to information system for every system is inevitable. And this must be done in a methodology.

1. GİRİŞ

Günümüzün teknolojik harikası olan bilgisayarın yalnızca bir hesaplama aracı olmayıp bir bilgi işlem ve bir bilgi iletişim aracı olduğu bir gerçektir. Genelde bilgi işlem olarak adlandırılan etkinliklerin hammaddesinin VERİ, ürününün BİLGİ (işlenmiş veri) olduğu da göz önüne alınırsa veri ve bilgilerin organize edilmesi sorunu ön plana çıkmaktadır. Nitekim pek çok projenin geliştirildiği ve uygulandığı büyük kuruluşlarda her projede pek çok program ve her programın kullandığı pek çok veri kütükleri mevcuttur. Veri ve bilgilerin depolandığı bu kütükler her programın kendi mantıksal yapısına göre şekillenmiştir. Ancak bu projeler ve programlar birbirleriyle ilişkili ve bütünler nitelikte olabileceklerinden kullanılan veri kütüklerinin de ilişkili ve ortak olması gerekmektedir. Bu kütüklerin içeriğinin büyük bir bölümünün aynı olması ve birden fazla yerde saklanması yer ve performans kaybına neden olmaktadır. Öte yandan kütükler işlendikçe (ekleme, silme, güncelleme vb.) aynı verileri içeren kütükler arasında tutarsızlıklar ortaya çıkmaktadır. Kullanılan veri kümesinin çok yoğun olması durumunda bu tutarsızlıkların daha büyük boyutlarda olacağı açıktır. İşte Bilgi Sistemi (BS) ve Veritabanı Sistemi (VTS) bu tür sorunları gidermek üzere geliştirilmiştir. Veritabanı teknolojisi bir veri organizasyonunun bütün olarak işlenmesini ve farklı uygulamalar için kullanıcıların verilere daha ekonomik bir biçimde ulaşmasını sağlar.

Coğrafi varlıklara ait veri toplanması arazide yapılan ölçümlerle olur. Bu değerler doğrulanır, işlenir ve bir bütün olarak organize edilerek çok çeşitli amaçlar için kullanılır. Jeodezik Bilgi Sistemi (JBS)'nin temelini teşkil eden bu verilerin elde edilmesi uzun zaman aldığı gibi önemli bir harcamayı da gerektirmektedir. Büyük bir ekonomik değere sahip bu verilerin hız – ekonomi – doğruluk optimizasyonu ile yönetilmelerinin gereği de kendiliğinden ortaya çıkmaktadır. Bu çalışmada bu tür büyük veri yığınlarının, bir veri yönetimi tekniği ile nasıl yönetileceği ve kullanıcılara sunulan hizmetin nasıl otomatize edileceği araştırılmaktadır.

Çalışmanın giriş bölümünden sonra gelen ikinci bölümünde bilgi sistemleri hakkında bilgi verilerek üçüncü bölümde bilgi sistemlerinin kuruluşlardaki rolüne değinilmiş, dördüncü bölümde veritabanı tasarım süreci aşamalarla ele alınmış ve beşinci bölümde bir bilgi sistemi olarak coğrafi bilgi sistemleri hakkında açıklamalarda

bulunulmuştur.

Altıncı bölümde, gerçekleştirilen uygulama hakkında detaylı bilgi verilmiştir. Bu bölümde örnek bilgi sistemi olarak **Jeodezik Bilgi Sistemi** ele alınmış ve anlatılan aşamaların uygulaması yapılmıştır.

Son olarak yedinci bölümde ise bu çalışmadan çıkarılan sonuçlar ve yapılan öneriler sunulmuştur.



2. BİLGİ SİSTEMİ KAVRAMI

Sistem terimi günümüzde pek çok disiplinde ortak olarak kullanılmaktadır. 1950’li yılların sonunda elektronik veri işlemindeki hızlı gelişmeler ve bazı ara disiplinlerin de ortaya çıkmasıyla sistem sözcüğünün kullanımı yaygınlaşmıştır.

Sistem terimi kısaca; insan, makine, işlem, doküman, veri veya önceden belirlenmiş gayelere ulaşmak için çevre ile iletişimde bulunan varlıkların örgütlenmiş bir bütünü olarak tanımlanabilir [1].

Bilgi sözcüğü (information), günümüzde diğer sözcüklerle çok karıştırılan bir sözcüktür. Bilgi (information) sözcüğü, İngilizce’de kullanılan data ve knowledge sözcükleri ile çok ilişkilidir. Ancak bu terimleri açıklamadan önce bir başka terim olan mesaj sözcüğünün açıklanması gereklidir.

Mesaj, bir kuruluşun bilgi sisteminden alınan, işlenen ve tekrar depolanan bir grup karakterden oluşur. Kuruluşların bilgi sistemlerinde akmakta olan bu mesajlar, içerdikleri anlam seviyesine bağlı olarak dardan genişe sırandığında ya veri (data) ya bilgi (information) yada birikim (knowledge) olarak adlandırılır.

Veri en dar anlamdadır ve bir grup karakterden oluşur. Kısaca ham gerçek ve düşüncelerden ibarettir. Bilgi veriye göre daha geniş anlamlıdır ve daha çok içinde bulunulan karar ortamında işe yararlar. Birikim ise en geniş anlama sahip olup geleceğe yönelik kararlarda potansiyel bilgi olarak yararlanılır (Şekil 2.1) [2].

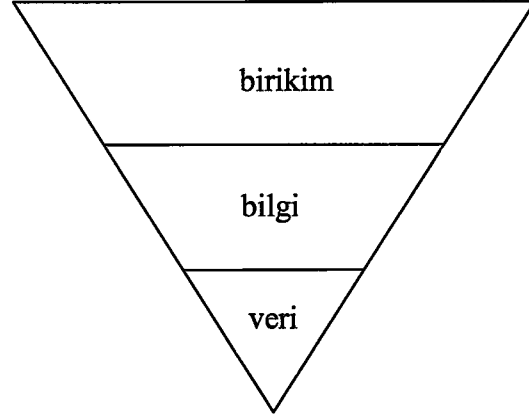
Bir kuruluşun bilgi sistemini, her yönetim seviyesinde gerekli olduğu yer ve zamanda bilgi sağlayan bir sistem olarak tanımlayabiliriz. Böyle bir sistem; herhangi bir yolla elde edilen bilgiyi almalı, depolamalı, depodan geri getirebilmeli, işlemeli ve bilgisayar sistemlerini veya başka ortamları kullanarak çevresi ile iletişim kurabilmelidir. Kısacası, kuruluşun istediği hedefleri yerine getirebilmelidir.

Yöntem (metot); herhangi bir şeyi yaparken veya işlerken izlenen sistematik yol, yordam (teknik) demektir. Yöntem kullanıldığında ortaya çıkan sonuç her zaman doğrudur.

Yöntembilim (metodoloji); bilimin, sanatın veya bir disiplinin kullandığı varsayım,

kural, kavram ve yöntemlerin bir bütünüdür. Yöntembilimler uygulandıktan sonra doğru sonuçlardan birine ulaşılma olasılığı yüksektir ancak kesin değildir.

Dolayısıyla başarılı bir bilgi sisteminin geliştirilmesinde herkes tarafından kullanılan standart bir yöntem bulunmamaktadır. Sadece değişik yöntembilimler vardır.



Şekil 2.1 Anlam Seviyesine Göre Veri, Bilgi, Birikim

Bilgi sistemi geliştirilmesinde kullanılan mevcut yöntembilimleri üç ana gruba ayırabiliriz [3].

1. İşlevsel (Fonksiyonel) yöntembilimler
2. Veri – güdümlü yöntembilimler
3. Kurallı yöntembilimler

İşlevsel yöntembilimler, sistemin ilgili alt sistemlere ayrışması üzerinde dururlar. Bu yüzden her bir alt sisteminde anlaşılması, tasarlanması ve uygulanması karmaşık değildir. Sistemin işlevleri temel amaç olduğundan “İşlevsel Yaklaşım” olarak da adlandırılırlar.

Kurallı yöntembilimler ise, sistem geliştirmesine yardımcı olacak bilgisayarlı yöntemlerdir. Genellikle yazılım sistemi gelişiminde kullanılırlar. Temel amaç, analizcileri yazılım tasarımının ayrıntılı yordamlarından korumak olup sistem özelliklerinin çözümlenmesi (analiz edilmesi) ve ilgili yazılımın üretilmesini sağlamaktır.

Veri – güdümlü yöntembilimler, işlenecek veriye ait özellikler üzerinde dururlar. İki alt sınıfa ayrılırlar:

- a. Veri akışı – güdümlü yöntembilimler
- b. Veri yapısı – güdümlü yöntembilimler

Veri akışı – güdümlü yöntembilimler genellikle veri tipleri ve sistem içindeki mantıksal akışından yararlanarak sistemin alt birimlere ayrılması esasına dayanırlar. Sistemin mantıksal olarak organizasyonu; veri akışının mantığı ile sistemi oluşturan alt birimler arasındaki işlevsel ilişkiye bağlı olarak gerçekleştirilir.

Veri yapısı – güdümlü yöntembilimler ise; temelde sisteme giren / çıkan veri yapılarının belirlenmesi ile uğraşırlar. Belirlenen yapılar; sistemin yapısı için bir baz oluştururlar. Sistemin elemanları veya alt birimlerin oluşturulması ve aralarındaki işlevsel ilişkilerin bulunması, sistemin yapısından yararla gerçekleştirilir.

Bu çalışmada veri akışı – güdümlü yöntembilimi izlenecektir.

3. BİLGİ SİSTEMİN KURULUŞLARDAKİ ROLÜ

Bu bölümde bilgi sistemlerinin örgütsel yapılara katkıları hakkında bilgi verilecektir.

3.1 Bilgi Sistemi Kullanımı İçin Kuruluş ve Çevre Şartları

Bilgi istemlerinde 1960'larda daha çok dosya sistemleri hakimdi. Kuruluşlar burdan veritabanı sistemlerine 1970 başlarında adım adım geçtiler. Bir çok kuruluş veritabanı hayat çarkı etkinliklerini idare ve denetim için bir veritabanı yöneticisine (VTY) bağlı idari birimler kurdular. Benzer olarak Bilgi Kaynağı İdaresi (BKİ) bir çok büyük kuruluş tarafından başarılı bir iş idaresi için anahtar kabul edildi. Bunun nedenleri şunlardı:

- Kuruluşlarda, en son bilgileri içeren büyük hacimli verilere ihtiyacı arttıran daha fazla işlevin bilgisayarla halledilir hale gelmesi
- Verilerin ve uygulamaların karmaşıklığı arttıkça veriler arasındaki karmaşık ilişkilerin modellenmesi ve korunması ihtiyacı
- Bilgi kaynaklarının birleştirilmesi eğilimi

Veritabanı sistemleri yukarıdaki üç ihtiyacı büyük ölçüde karşılarlar. İlave iki özellik daha büyük veritabanlarının tasarım ve idaresinde oldukça önemlidir:

- **Veri Bağımsızlığı** uygulama yazılımlarının, temeli oluşturan mantıksal kuruluştaki, fiziksel erişim yollarında ve depolama yapılarında değişiklik yapmalarını engeller.
- **Dış Görünümler** aynı verinin, her biri ayrı görünüme sahip bir çok uygulama tarafından kullanılmasına izin verir.

Veritabanı sistemine geçişi haklı kılan bir değer neden ise eski dosya sistemiyle karşılaştırıldığında çok düşük kalan yeni uygulama geliştirme maliyetidir. Bu çoğunlukla başlangıçtaki veritabanı tasarımı ve sistem dönüşümündeki yüksek maliyeti dengeler. Yüksek – seviye veri erişim dilleri veritabanı uygulama geliştirimini basitleştirirken, yüksek – seviye sorgulama dilleri yüksek – seviye idareciler için bilginin anında sorgulanmasını mümkün kılar.

Eğilim 1970 başlarından 1980 ortalarına kadar büyük, merkezi veri depoları yaratarak bunların bir veritabanı yönetim sistemi (VTYS) tarafından yönetilmesi yönündeydi. Son zamanlarda bu eğilim aşağıdaki sebeplerden dolayı tersine döndü:

1. Kişisel bilgisayarlar ve VISICALC, LOTUS 1 – 2 – 3, SYMPHONY, PARADOX, DBASE IV – V ve MS – ACCESS gibi veritabanı yönetim sistemi benzeri yazılımlar, daha önceden seyrek ve düzensiz veritabanı kullanıcısı kategorisinde yer alan kullanıcılar tarafından yoğun olarak kullanılmaya başlandı. Birçok yönetici, mühendis, bilim adamı, mimar vb. meslek grubu bu kategoride yer almaktadır. Sonuç olarak, **kişisel veritabanları** kurma uygulaması yaygınlık kazandı. Artık bir veritabanı sunucusunda tutulan büyük bir veritabanının bir bölümünün kopyasını alarak kişisel bir çalışma istasyonunda çalışmak ve onu sunucuya geri yüklemek mümkün olmaktadır. Benzer olarak, kullanıcılar kendi veritabanlarını tasarlayıp yaratarak daha büyük veritabanlarına katılabilmektedir.
2. Dağıtık Veritabanı Yönetim Sistemleri (DVTYS) 'nin ortaya çıkması daha iyi yerel denetim ve daha hızlı yerel işlem için veritabanını çeşitli bilgisayar sistemlerine dağıtma seçeneğini sunmaktadır. Aynı zamanda yerel kullanıcılar uzak veriye dağıtık veritabanı yönetim sistemi kolaylıklarını kullanarak erişebilmektedirler.
3. Birçok kuruluş artık, veritabanları için meta – veriyi (yani veritabanı yapısı, koşulları, yetkilendirmeleri vb. bilgileri) idare eden bir mini veritabanı yönetim sistemi olan **veri sözlüğü sistemlerini** kullanmaktadır [4]. Bunlar çoğunlukla bilgi kaynağı yönetimi için *bütünleşik araçlar* olarak kullanılırlar. Yararlı bir veri sözlüğü sistemi aşağıdaki türden bilgileri depolamalı ve yönetmelidir:
 - a. Veritabanı sistem şemalarının tanımları
 - b. Depolama yapıları, erişim yolları ve dosya ve kayıt boyutları gibi fiziksel veritabanı tasarımı ile ilgili ayrıntılı bilgi
 - c. Veritabanı kullanıcılarının, sorumluluklarının ve erişim haklarının tanımlamaları
 - d. Veritabanı işlemlerinin (sorgulama dizileri – transactions), uygulamalarının ve işlemlerle kullanıcı ilişkilerinin yüksek – seviye tanımları
 - e. Veritabanı işlemleri ile bunların kullandıkları veri öğeleri arasındaki ilişkiler (Bu belirli bir veri tanımı değiştiğinde hangi işlemlerin etkilendiğinin bulunmasında yararlıdır).

- f. Sorgulama, işlem sıklıkları ve veritabanının değişik bölümlerinin ulaşım adetleri gibi kullanım istatistikleri

Bu bilgi veritabanı yöneticileri, tasarımcıları ve yetkili kullanıcılara anında sistem belgeleme (on – line system documentation) sağlaması açısından yararlıdır. Bu da veritabanı yöneticisinin bilgi sistemi üzerindeki denetimini ve kullanıcıların sistemi anlamasını ve kullanmasını kolaylaştırır. Birçok büyük kuruluşta veri sözlüğü sistemi, veritabanı yönetim sistemi kadar önemli kabul edilir.

Son zamanlarda yüksek başarılı (performanslı) (hizmet sektöründe kullanılan ve 24 saat kullanım gerektiren) **işlem yürütme sistemleri** (transaction processing systems)'ne büyük önem verilmektedir. Bu veritabanları çoğunlukla yerel ve uzak uçbirimler (terminaller) tarafından dakikada yüzlerce kez erişilmektedir. Dakikadaki ortalama işlem miktarı ve ortalama ve en uzun cevaplama zamanı türünden ifade edilen işlem başarımı bu uygulamalar açısından kritiktir. Kuruluşun işlem – yürütme ihtiyaçlarını karşılayan dikkatli bir veritabanı tasarımı bu tür sistemler için şarttır.

Bazı kuruluşlar bilgi kaynağı yönetimlerini belli veritabanı yönetim sistemleri ve veri sözlüğü ürünleri ile yapmaktadırlar. Çok büyük ve karmaşık sistemlerin tasarım ve uygulamalarına yaptıkları yatırımlar yeni veritabanı yönetim sistemlerine geçmelerini zorlaştırır ve böylece kuruluşlar mevcut veritabanı yönetim sistemlerinde kilitli kalırlar. Böyle büyük ve karmaşık sistemler söz konusu olduğunda değişken gereksinimlerin neden olduğu olası değişikliklerin önemi asla azımsanamaz. Büyük bir karmaşık sistem gelişemez ve diğer bir veritabanı yönetim sistemine geçişi zorunlu kılırsa maliyet çok yüksek olabilir.

3.2 Bilgi Sistemi Hayat Çarkı

Büyük bir kuruluşta veritabanı sistemleri çoğunlukla kuruluşun bilgi kaynaklarının yönetiminde kullanılan çok daha büyük bir bilgi sisteminin bir parçasıdır. Bir **bilgi sistemi**, kuruluş içerisindeki bilginin derlenmesi, yönetimi ve dağıtımının kapsadığı tüm kaynakları içerir. Bilgisayarlı bir ortamda bu kaynaklar verinin kendisini, veritabanı yönetim sistemi yazılımını, bilgisayar sistemi donanımını ve depolama medyasını, veriyi kullanan ve yöneten personeli (veritabanı yöneticisi, son kullanıcılar, parametrik kullanıcılar, vb...), veriye ulaşım güncelleyen uygulama yazılımını ve bu yazılımı hazırlayan yazılımcıları kapsar. Dolayısıyla veritabanı sistemi çok daha büyük bir örgütsel bilgi sisteminin sadece bir parçasıdır.

Bu bölümde bir bilgi sisteminin tipik hayat çarkı ve veritabanı sisteminin buna nasıl uyduğu incelenecektir. Genellikle bilgi sistemi hayat çarkına **makro hayat çarkı**, veritabanı sistemi hayat çarkına ise **mikro hayat çarkı**, adı verilir. Tipik bir **makro hayat çarkı**, şu aşamaları içerir [5]:

1. Olurluk çözümlemesi (analizi): Olası uygulama alanlarının çözümlenmesini, ön kar – maliyet çalışmalarını ve uygulamalar arası önceliklerin belirlenmesini içerir.
2. Gereksinimlerin belirlenmesi ve çözümlenmesi: Gereksinimlerin ayrıntıları, kullanıcılarla yapılan görüşmelerle derlenir.
3. Tasarım: Bu aşama iki dalda ilerler. Veritabanının tasarımı ve bu veritabanını kullanıp, işleyen uygulama sistemlerinin (yazılımlarının) tasarımı
4. Uygulama: Bilgi sistemi uygulanır, veritabanı yüklenir, veritabanı işlemleri uygulanır ve denenir.
5. Değerlendirme ve kabul testi: Sistem, kullanıcı gereksinimlerini karşılama ve başarımlı ölçütleri (kriterleri) anlamında değerlendirilir. Sistem, başarımlı ölçütlerine ve davranış şartlarına karşı denenir.
6. Kullanma: Bundan önce, eski sistemdeki kullanıcıların dönüşümü ve kullanıcı eğitimi gelebilir. Kullanma aşaması tüm sistem işlevleri kullanılabilir olduğunda başlar. Yeni gereksinim ve uygulamalar türedikçe sisteme katılıp kullanılmaya başlamadan önce, baştaki tüm aşamalardan geçer. Sistem başarımlının gözlenmesi ve sistem bakımı kullanma aşamasının önemli etkinlikleridir.

3.3 Veritabanı Uygulama Sistemi Hayat Çarkı

Veritabanı Uygulama Sistemi (mikro) hayat çarkı ile ilgili etkinlikler aşağıdaki aşamaları içerir [6]:

1. Sistem tanımlaması: Veritabanı sisteminin kapsamı, kullanıcıları ve onun uygulamaları tanımlanır.
2. Tasarım: Bu aşamanın sonunda, seçilmiş veritabanı yönetim sistemi üzerinde tüm mantıksal ve fiziksel tasarım hazır hale gelmiş olur.
3. Uygulama: Bu aşama kavramsal, dış ve iç veritabanı tanımlarının yazılması, boş veritabanı dosyalarının yaratılması ve yazılımların uygulanmasını içerir.
4. Yükleme veya veri dönüşümü: Veritabanı, ya verilerin doğrudan yüklenmesi yada hazır dosyaların veritabanı formatına dönüştürülmesi ile doldurulur.
5. Uygulama dönüşümü: Önceki sistemden kalma tüm yazılım uygulamaları yeni

sisteme dönüştürülür.

6. Test ve değerlendirme: Yeni sistem denenir ve değerlendirilir.

7. Kullanım: Veritabanı sistemi ve uygulamaları kullanıma konur.

8. İzleme ve bakım: Kullanım aşaması boyunca sistem sürekli izlenir ve bakımı yapılır. Hem veri içeriğinde hem de yazılım uygulamalarında büyüme ve genişleme olabilir. Zaman zaman büyük değişiklikler ve yeniden yapılanmalar gerekebilir.

2, 3 ve 4 etkinlikleri daha büyük olan bilgi sistemin tasarım ve uygulama aşamalarının parçasıdır. Bir sonraki bölümde daha çok veritabanı tasarımını kapsayan ikinci etkinlikten bahsedilecektir. Kuruluşlardaki veritabanları tüm hayat çarkı etkinliklerinden geçerler. Dönüşüm adımları (4 ve 5) hem veritabanı hem de uygulamalar yeniyse uygulanabilir değildirler. Bir kuruluş, kurulu eski bir sistemden bir yenisine geçerken 4 ve 5 etkinlikleri en çok zaman alan aşamalardır ve bunları gerçekleştirmek için gereken çaba çoğunlukla azımsanır. Her aşamada sıkça yeni gereksinimler doğduğundan çeşitli adımlar arası geri beslemeler olur.

4. VERİTABANI TASARIM SÜRECİ

Veritabanı tasarım sorunu şu şekilde ifade edilebilir: Bir kuruluşteki kullanıcıların tanımlı bir uygulama kümesi için olan bilgi gereksinimlerini karşılamak üzere bir veya daha fazla veritabanının mantıksal ve fiziksel tasarımını yapma [7].

Veritabanı tasarımının hedefleri çeşitlidir: belirli kullanıcı ve uygulamaların bilgi gereksinimlerini karşılamak; doğal ve anlaması kolay bir bilgi yapılanması sağlamak; son olarak veri işleme gereksinimlerini ve cevaplama zamanı, işlem zamanı, depolama kapasitesi gibi başarımlar hedeflerini desteklemek. Bunlar gerçekleştirilmesi ve ölçülmesi zor hedeflerdir. Veritabanı tasarım sürecinin gayri resmi ve zayıf tanımlanmış hedeflerle başlaması, bu hedefleri ulaşılması daha zor bir hale sokar. Halbuki tasarım etkinliği, veritabanı uygulanmaya başladığında değiştirilmesi kolay olmayan, katı bir şekilde tanımlanmış bir veritabanı görünümüyle sonuçlanır. Veritabanı tasarım sürecinde altı ana aşama belirlenebilir:

1. Gereksinimlerin derlenmesi ve çözümlenmesi
2. Kavramsal veritabanı tasarımı
3. Veritabanı yönetim sisteminin seçimi
4. Veri modeli dönüşümü (mantıksal veritabanı tasarımı olarak da adlandırılır.)
5. Fiziksel veritabanı tasarımı
6. Veritabanı sisteminin uygulanması

Tasarım süreci Şekil 4.1’de gösterildiği gibi iki paralel etkinliği içerir. İlk etkinlik veritabanının **veri içeriği ve yapısının** tasarımını içerir. İkincisi ise veritabanının **işlenmesi ve yazılım uygulamaları** ile ilgilidir. Bu iki etkinlik oldukça iç içe geçmiştir. Örneğin, veritabanında tutulacak veriler veritabanı uygulamaları çözümlenerek belirlenebilir. Buna ek olarak, veritabanı dosyalarının depolama yapısının ve erişim yollarının seçildiği fiziksel veritabanı tasarım aşaması, bu dosyaları kullanacak olan uygulamalara bağlıdır. Diğer taraftan, veritabanı uygulamalarının tasarımı genellikle ilk etkinlik sırasında oluşturulan veritabanı şema yapılarına başvurularak hazırlanır. Bu iki etkinliğin birbirlerini çok güçlü bir şekilde etkilediği açıktır.

Yukarıda bahsedilen altı adım kesin bir dizi içerisinde ilerlemeyebilir. Birçok durumda erken aşamalarındaki tasarımların sonraki aşamalarda değiştirilmesi zorunlu olabilir. Aşamalar arasındaki ve içindeki bu geri besleme ilmekleri (lupları) veritabanı tasarımı sırasında yaygındır. Şekli karıştırmamak için Şekil 4.1’de bunlar gösterilmemiştir. Bu şekildeki Aşama 1 veritabanı kullanımına yönelik bilgi toplamayı içerirken, Aşama 6 veritabanı uygulaması ile ilgilidir. Aşama 1 ve 6 bazen veritabanı tasarımının bir parçası olarak değil de daha genel bir bilgi sisteminin hayat çarkının bir parçası olarak düşünülür. Veritabanı tasarım işleminin kalbi aşağıda kısaca özetlenecek olan Aşama 2, 4 ve 5’tir.

- **Kavramsal veritabanı tasarımı (Aşama 2):** Bu aşamanın hedefi veritabanı yönetim sisteminden bağımsız bir kavramsal veritabanı şeması üretmektir. Bu aşamada çoğunlukla Varlık – İlişki (Vİ) yada Geliştirilmiş Vİ (GVİ) gibi yüksek – seviye bir veri modeli kullanılır. Ek olarak belli bir veritabanı yönetim sisteminden bağımsız bir gösterim (notasyon) kullanılmak suretiyle bilinen veritabanı uygulamalarının ve işlemlerinin mümkün olduğunca fazlası belirlenir.
- **Veri modeli dönüşümü (Aşama 4):** Buna mantıksal veritabanı tasarımı da denir. Bu aşama sırasında Aşama 2’de kullanılan yüksek – seviye veri modelinde oluşturulan kavramsal veri şeması, Aşama 3’te seçilmiş olan veritabanı yönetim sisteminin veri modeline dönüştürülür. Bu aşamaya uygulama veri modelinin seçiminden sonra belli bir veritabanı yönetim sisteminin seçimini beklemeden (örneğin ilişkisel veritabanı yönetim sistemi kullanılmasına karar verilmiş ama adı konmamış ise) başlanabilir. Buna **sistem – bağımsız** (ama veri modeli – bağımlı) mantıksal tasarım denir. Bu aşamanın sonucu seçilmiş bir veri modelinde **kavramsal şema**dır. Ek olarak bu aşama boyunca belli uygulamalar için dış şemalar sıklıkla yapılır.
- **Fiziksel veritabanı tasarımı (Aşama 5):** Bu aşama sırasında depolanmış (doldurulmuş) veritabanına ait fiziksel depolama yapısı, kayıt yapma ve erişim yolları anlamında şartlar tasarlanır.

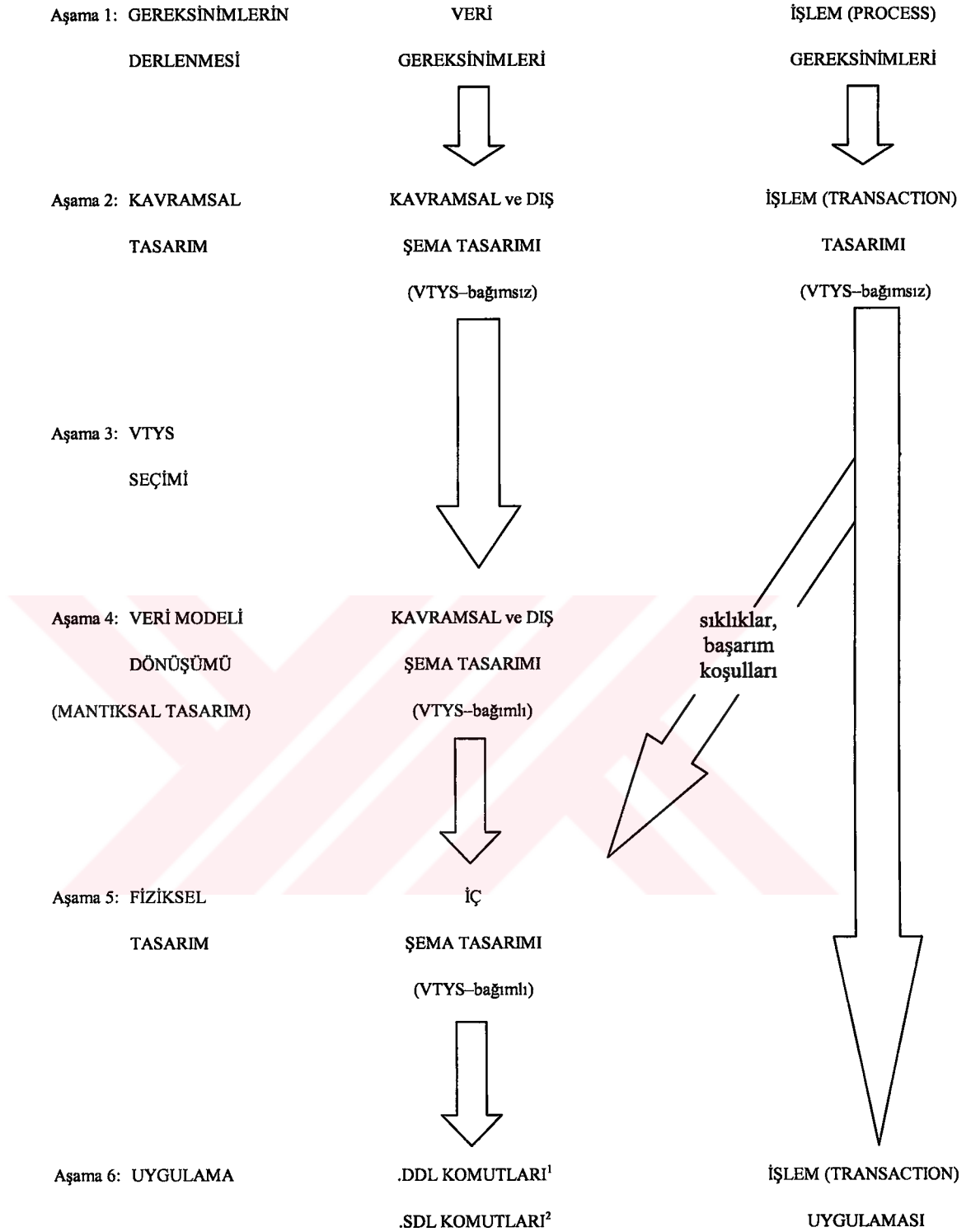
4.1 Aşama 1: Gereksinimlerin Derlenmesi ve Çözümlemesi

Bir veritabanının etkin olarak tasarlanmasından önce kullanıcı beklentilerinin ve veritabanının kullanılma eğiliminin olabildiğince ayrıntılı olarak bilinmesi gerekmektedir. Kullanılma eğiliminin belirlenmesi ve çözümlemesi işlemi gereksinimlerin belirlenmesi ve çözümlemesi olarak da adlandırılır.

Gereksinimlerin tanımlanması için öncelikle, bilgi sisteminin veritabanı ile etkileşecek bölümlerinin belirlenmesi gerekir. Bunlar eski ve yeni kullanıcı ve uygulamaları içerir. Bu kullanıcı ve uygulamaların gereksinimleri daha sonra belirlenir ve çözümlenir. Aşağıdaki etkinlikler bu aşamanın tipik bölümleridir:

- Veritabanını kullanacak ana uygulama alanları ve kullanıcı grupları belirlenir. Her bir grupta gereksinim derlenmesi ve belirlenmesinin alt aşamalarında ana katılımcı olacak anahtar personel seçilir.
- Önce uygulamalar ile ilgili hazır belgeler (varsa) incelenir. Diğer belgeler (kılavuzlar, formlar, raporlar ve örgütsel grafikler) gereksinim derleme ve tanımlama işlemine etkisinin olup olmayacağını belirlenmesi için gözden geçirilir.
- Mevcut işletim çevresi ve bilginin planlanan kullanılma amacı araştırılır. Bu adım işlem türlerinin ve sıklıklarının olduğu kadar sistemdeki veri akışının incelenmesini kapsar. İşlemlere girdi ve çıktı olan veriler belirlenir.
- Olası veritabanı kullanıcılarından soru kümelerine olan yazılı yanıtları (anketler) toplanır. Bu sorular, kullanıcıların çeşitli uygulamalara verdikleri önem ve öncelikleri kapsar. Anahtar personel ile bilginin değerinin biçilmesi ve önceliklerinin kurulması konularında yardım almak için görüşmeler yapılabilir.

Yukarıda bahsedilen gereksinim derleme yolları zayıfça yapılandırılmış ve çoğunlukla gayri resmi gereksinimleri verir. Bunlar daha sonra daha resmi **gereksinim belirleme yordamları** (teknikleri)’nden biri kullanılarak daha iyi yapı bir hale dönüştürülür. Bu yordamlar şunlardır: Sıradüzensel Girdi İşlem Çıktı – SGİÇ (Hierarchical Input Process Output – HIPO), Yapısal Çözümleme ve Tasarım Yordamı – YÇTY (Structured Analysis and Design Technique – SADT), Veri Akış Çizgeleri – VAÇ (Data Flow Diagrams – DFDs), Orr – Warnier Çizgeleri, ve Nassi – Schneiderman Çizgeleri. Tüm bunlar bilgi işleme gereksinimlerini düzenleme ve sunma için kullanılan çizgesel yöntemlerdir. Bu çizgeler yöntemine göre sıra düzenler, akış çizgeleri, ardışık ve ilmik yapıları vb. gösterimi kullanırlar. Metin, tablo, çizge şeklindeki ilave belgeler ve karar gereksinimleri genellikle çizgelere eşlik eder. Tüm konusu bu aşamayı tartışmak olan kitaplar bulunabilir [8] [9] [10]. Bir sonraki bölümde bu çalışmada kullanılacak olan VAÇ hakkında bilgi verilecektir. Bu aşama oldukça zaman alıcı olabilir, fakat bilgi sisteminin gelecekteki başarısı için çok önemlidir.



Şekil 4.1 Büyük Veritabanları İçin Veritabanı Tasarım Aşamaları

Gereksinimlerin derlenmesi ve çözümlemesi işi için bazı bilgisayar – yardımcı yordamlar üzerinde bazı çalışmalar yapılmıştır. Bu yordamlar koşulların bütünlüğünü

¹ Veri Tanımlama Dili – Data Definition Language

² Depo Tanımlama Dili – Storage Definition Language

ve kararlılığının denetimi için gereksinim çözümlemelerinde kullanılacak otomatik araçları içerir. Gereksinimler, genellikle tasarım veritabanı adı verilen tek bir ambarda depolanır ve tasarım ilerledikçe buradan görüntülenebilir ve güncellenebilir (Örn: SORUN TANIMLAMA DİLİ VE SORUN TANIMLAMA ÇÖZÜMLEYİCİSİ – STD / STÇ, PROBLEM STATEMENT LANGUAGE and PROBLEM STATEMENT ANALYZER – PSL / PSA [11]).

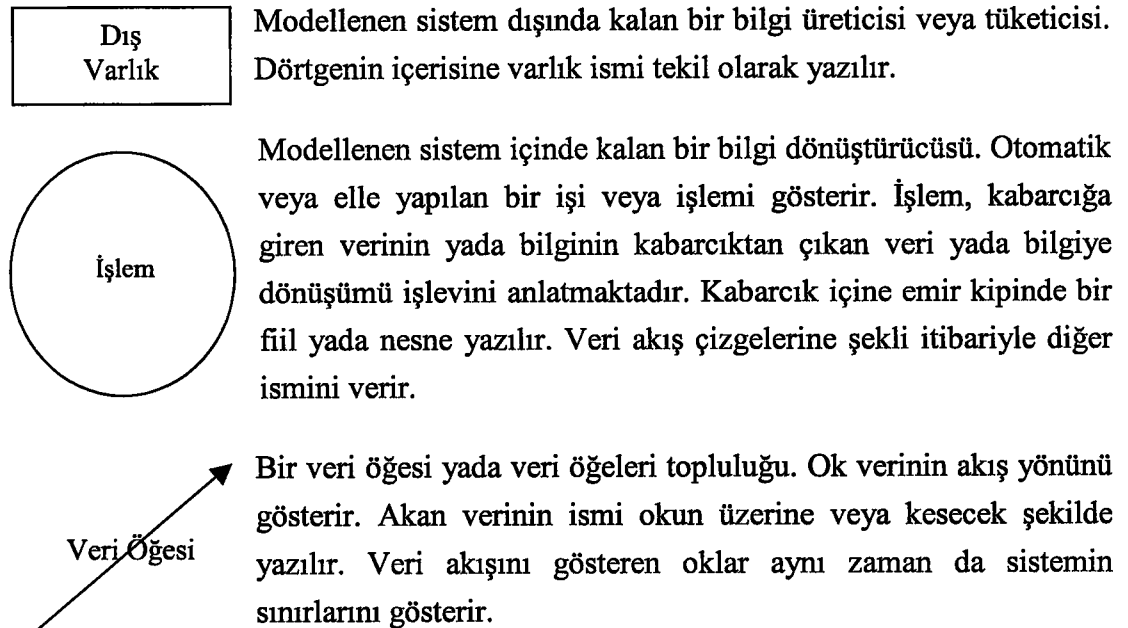
4.1.1 Veri Akış Çizgeleri

Veri akış çizgeleri; sistemin karmaşıklık seviyesi ne olursa olsun o sistemin mantıksal seviyede anlaşılmasında ve model düzenlemelerinin yapılması amacıyla çözümleme safhasında kullanılan çizgesel bir araçtır. Veri akış çizgelerine kabarcık çizgeleri (bubble chart) de denir.

Veri akış çizgeleri ;

- Sistemlerin alt sistemlere olan ayrışmasını,
- Sistem içindeki veri akışını,
- Veri depolarını ve buralara giren – çıkan verileri ve
- Sisteme göre dış varlık olarak bilinen kullanıcıları göstermektedir.

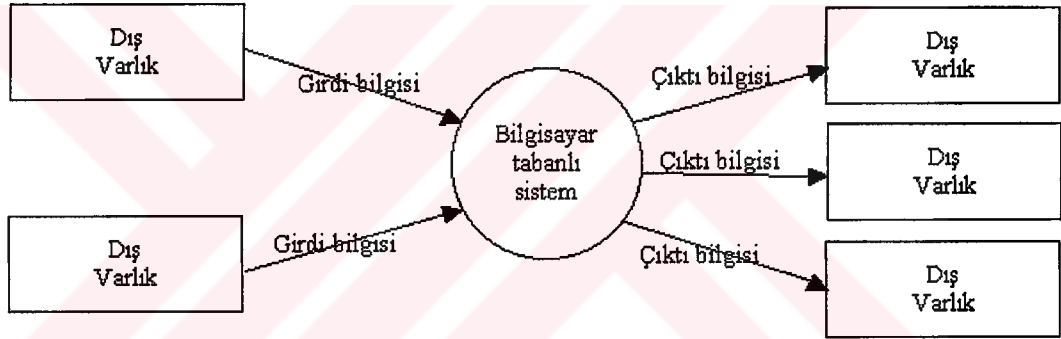
Veri akış çizgelerinde kullanılan işaretler şunlardır:



Veri Deposu

Bir veya daha fazla işlemin kullanacağı verinin depolanacağı bir havuz. Bir kuyruk yada arabellek kadar basit yada ilişkisel bir veritabanı kadar karmaşık olabilir. İçlerine depoladığı veriyi en iyi şekilde anlatan tekil isim yazılır. Veriler bir işlemten bir işleme doğrudan nadiren gönderilirler; depoların kullanımı burada ortaya çıkmaktadır. Depolar aslında verilerin bir işlemten bir işleme gönderilmeden önce sıralanmak veya başka işlemler yapılmak üzere alıkonuldukları havuzlardır. Dolayısıyla sistem başarımını yavaşlatan bir unsur olarak görülürler. Depo işaretinin içine yazılacak olan isim giren – çıkan veriyi iyi bir şekilde tanımlıyorsa veri akışının üzerine giren – çıkan verinin ismi yazılmayabilir.

Veri akış çizgeleri hazırlanırken genelde kaynak dış varlıklar kağıdın solunda hedef dış varlıklar kağıdın sağında, işlemler ortada, depolar ise alt ve üst kısımda çizilir.

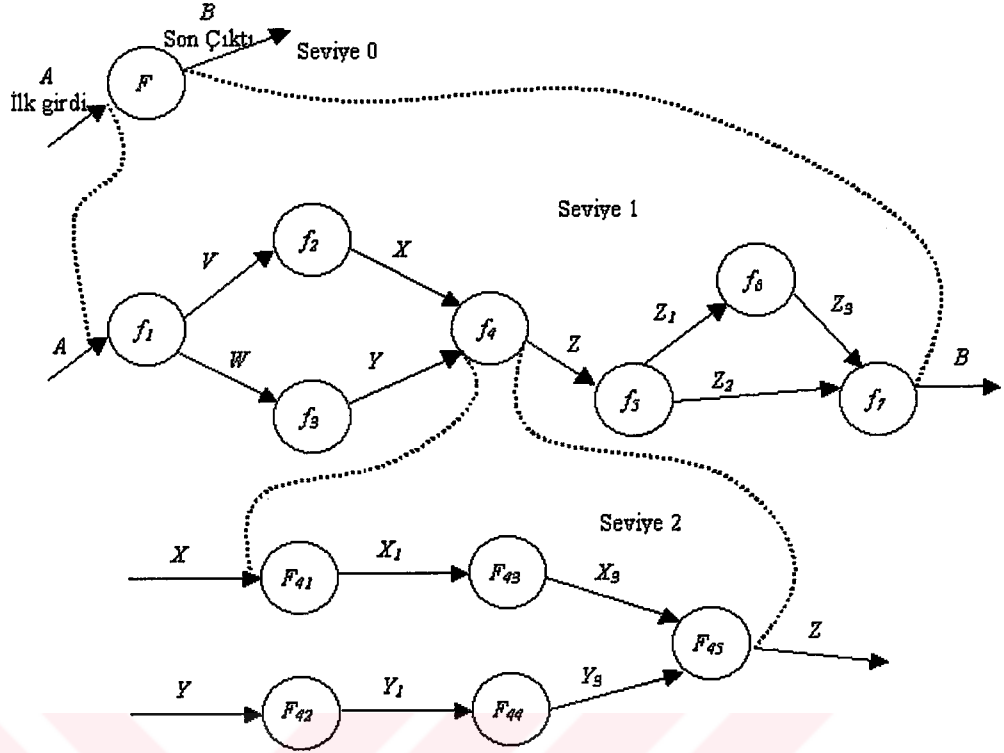


Depo, ve dış varlıklar gerekirse birden fazla gösterilirler (Şekil 4.2).

Şekil 4.2 Bilgi Akış Modeli

Veri akış çizgeleri içerdikleri anlam itibarıyla şu kategorilere ayrılır (Şekil 4.3) [12];

1. Ana (Context) Çizge – Seviye 0: Düşünülen sistemin sınırlarını bir bütün olarak belirleyen çizgedir. Bu çizgede sistem ve dışında kalan kullanıcılar gösterilir.
2. Tasarım (Overview) Çizgesi – Seviye 1: Sistem içinde kullanılan veri depolarını, giren ve çıkan tüm veri akışlarını ve sorgulamalarını gösterir. Ayrıca sistem; birden fazla alt sistemden oluşuyorsa her biri için ayrı bir tasarım çizgesi hazırlanabilir.
3. Detaylı (Detailed) Çizge – Seviye 2: Tasarım çizgelerindeki işlemlerin basitleştirilmesi amacıyla kullanılan alt – seviye çizgelerdir.



Şekil 4.3 Bilgi Akış İyileştirmesi

4.2 Aşama 2: Kavramsal Veritabanı Tasarımı

Veritabanı tasarımının ikinci aşaması iki paralel etkinlik içerir. Bunların ilki olan kavramsal şema tasarımında, Aşama 1'den elde edilen veri gereksinimleri incelenir ve bir kavramsal veritabanı şeması üretilir. İkincisi olan işlem tasarımında Aşama 1'de irdelenen veritabanı uygulamaları incelenir ve bu işlemler için yüksek seviyeli tanımlar üretilir.

4.2.1 Aşama 2a: Kavramsal Şema Tasarımı

Bu aşamada üretilen kavramsal şema çoğunlukla VTYS' den bağımsız yüksek seviyeli bir veri modeli ile yapıldığından doğrudan veritabanına uygulanamaz. VTYS – bağımsız bir kavramsal şema aşağıdaki nedenlerden dolayı önemlidir:

1. Kavramsal şema tasarımının hedefi; veritabanı yapısının anlamının (semantiğinin), ara ilişkilerin ve koşulların tam olarak anlaşılmasıdır. Her veritabanı yönetim sisteminin kendine has yordam ve kısıtlamalarından kaynaklanabilecek etkileri bertaraf etmek için kavramsal şema tasarımı veritabanı yönetim sisteminden bağımsız olmalıdır.

2. Kavramsal şema tasarımı veritabanı içeriğinin sabit tanımlamalarını içermez. Veritabanı yönetim sistemi seçimi ve sonraki tasarım kararları VTYS – bağımsız kavramsal şema tasarımı değişimini gerektirmeden değişebilir.
3. Veritabanı kullanıcıları ve uygulama tasarımcıları için kavramsal şemanın iyi anlaşılması ve bunun sağlanması için veritabanı yönetim sisteminin veri modelinden daha genel ve açıklayıcı bir yüksek – seviye veri modelinin kullanımı çok önemlidir.
4. Kavramsal şemanın çizgesel tanımlaması veritabanı kullanıcıları, tasarımcıları ve çözümçüleri arasında mükemmel bir iletişim aracı olarak hizmet edebilir. Yüksek – seviye veri modelleri, VTYS – bağımlı daha düşük seviyeli veri modellerine göre daha kolay anlaşılır kavramlara dayandığından, şema tasarımı ile ilgili herhangi bir iletişim, daha aracısız ve kesin olarak meydana gelir.

Veritabanı tasarımının bu aşamasında, yüksek seviyeli semantik veya kavramsal veri modeli de denilen aşağıdaki özelliklere sahip bir veri modelinin kullanılması önemlidir [13]:

1. Anlamlılık: Veri modeli değişik tip verileri, ilişkileri ve koşulları ayrı anlamlarla ifade edebilmelidir.
2. Basitlik: Vasıfsız bir kullanıcının kavramlarını anlayıp kullanabileceği kadar basit olmalıdır.
3. Azlık (minimalite): Model, anlam olarak ayrı ve örtüşmeyen az sayıda temel kavrama sahip olmalıdır.
4. Çizgesel gösterim: Model, kavramsal şemayı temsil için yorumlaması kolay, çizgesel bir görünüme sahip olmalıdır.
5. Kurallılık: Bir veri modeli ile vurgulanan bir kavramsal şema belirsizliği bulunmayan kurallı bir veriyi temsil etmelidir, yani, model kavramları doğru ve belirsizliğe yer bırakmayacak şekilde tanımlanmalıdır.

Bu gereksinimler bazen birbiriyle çelişir. Özellikle gereksinim – 1 diğerleriyle çelişir. Veritabanı tasarımı için birçok yüksek seviye kavramsal model önerilmiştir [14] [15] [16]. Aşağıdaki 4.2.1.1 bölümünde veri modelleri hakkında kısa bilgi verilmiştir. Bu çalışmada bu modellerden Varlık – İlişki, Vİ (Entity – Relationship, ER) modeli kullanılacaktır.

4.2.1.1 Veri Modelleri

Veritabanlarının mantıksal yapıları belli biçimlerde modellendirilmektedir. Veri

modeli, fiziksel depolama ve erişim yolları başta olmak üzere veritabanının birçok özelliklerini biçimlendirmekte ve buna göre her veritabanı yönetim sistemi yazılımı belli bir modeli esas almaktadır. Bu nedenle “Model” ve “Veri Modeli” kavramlarına açıklık getirmek gereklidir.

Model; gerçek dünyadaki varlıklar, olayların ve bunlar arasındaki ilişkilerin temsil ediliş şeklidir.

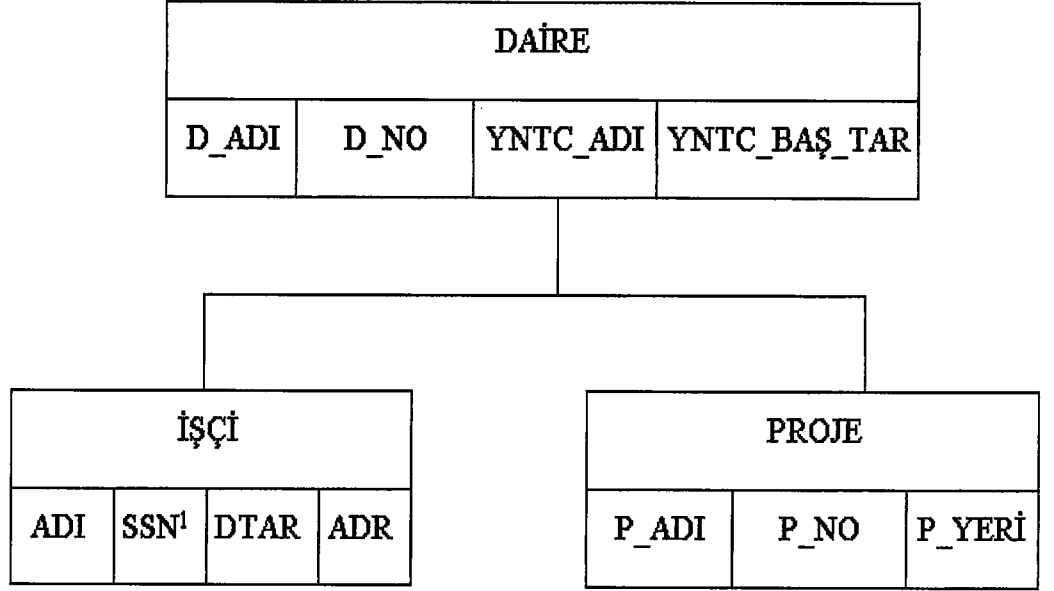
Veri modeli; gerçek dünyadaki varlıklar, olaylar, etkinlikler ve bunlar arasındaki ilişkiler hakkındaki verilerin temsil ediliş şeklidir.

Varlıklar ve olayların kendi aralarında üç çeşit ilişkisi vardır:

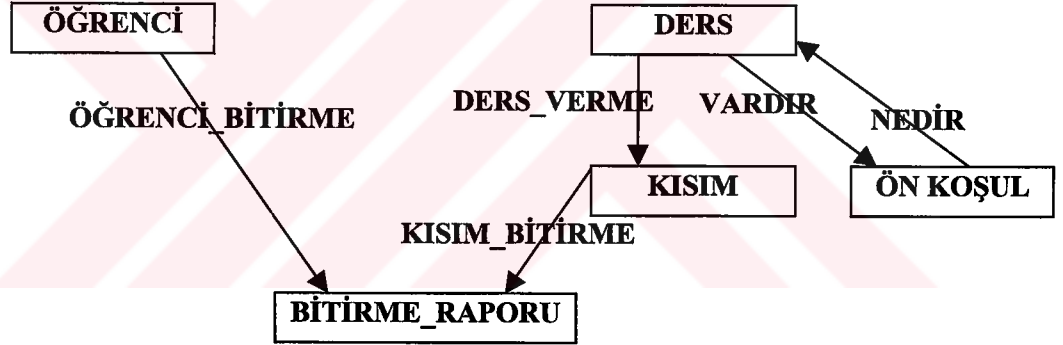
- Bire bir (1 : 1) ilişki (one – to – one)
- Bire çok (1 : M) ilişki (one – to – many)
- Çokla çok (M : N) ilişki (many – to – many)

Bu ilişkiler ışığında şu veri modelleri tanımlanabilir;

- Sıradüzensel veri modeli: Varlıklar arası ilişkiler bire – çok şeklindedir. İlişkiler değişik düzeylerdeki bağlantılarla kurulur. Alt düzeydeki bir varlık sadece bir üst düzeydeki varlıklarla bağlantılıdır (Şekil 4.4).
- Ağ veri modeli: Varlıklar arası ilişkiler çokla – çok şeklindedir. İlişkiler değişik düzeylerdeki bağlantılarla kurulur. Alt düzeydeki bir varlık üst düzeydeki birden fazla varlık ile bağlantılı olabilir (Şekil 4.5).
- İlişkisel veri modeli: Bu model veritabanını, her bir tablonun bir dosyada tutulduğu bir tablolar topluluğu olarak alır (Şekil 4.6).
- Nesneye – yönelik veri modeli: Bu model veritabanını nesneler, onların özellikleri ve onların işlemleri cimsinden tanımlar. Aynı yapı ve davranıştaki nesneler bir sınıfa aittir ve sınıflar sıradüzensel veya dönüşsel olmayan (acyclic) çizgeler olarak düzenlenirler (Şekil 4.7).
- Yüksek – seviye kavramsal veri modelleri (Veri – İlişki): Gösterim anlamları Şekil 4.8’de verilen bu model, veriyi varlık, ilişki ve öznitelik olarak tanımlar ve bunların belirlenmesinden meydana gelir (Şekil 4.9) [17] [18].



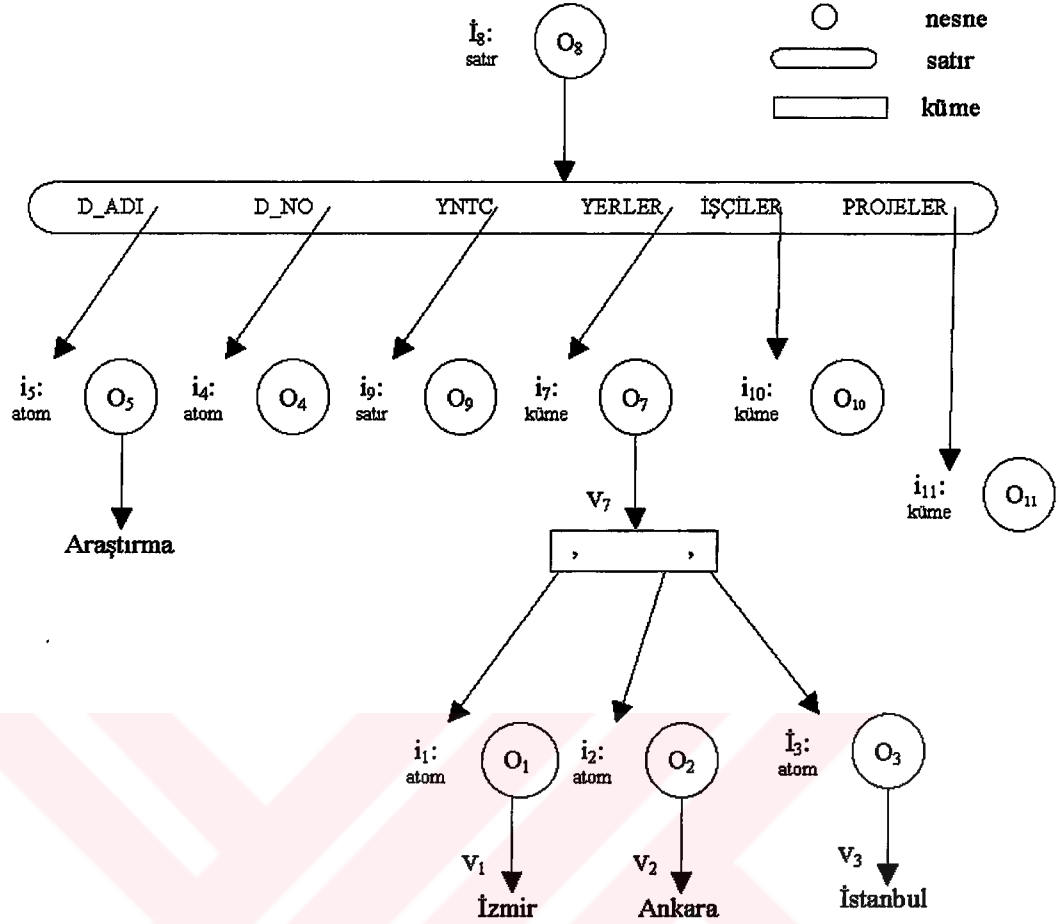
Şekil 4.4 Sıradüzensel Veri Modeli



Şekil 4.5 Ağ Veri Modeli

ÖĞRENCİ	Adı	ÖğrenciNo	Sınıf	Branş
	Erol	12	1	SOS
	Ebru	23	2	FEN

Şekil 4.6 İlişkisel Veritabanı



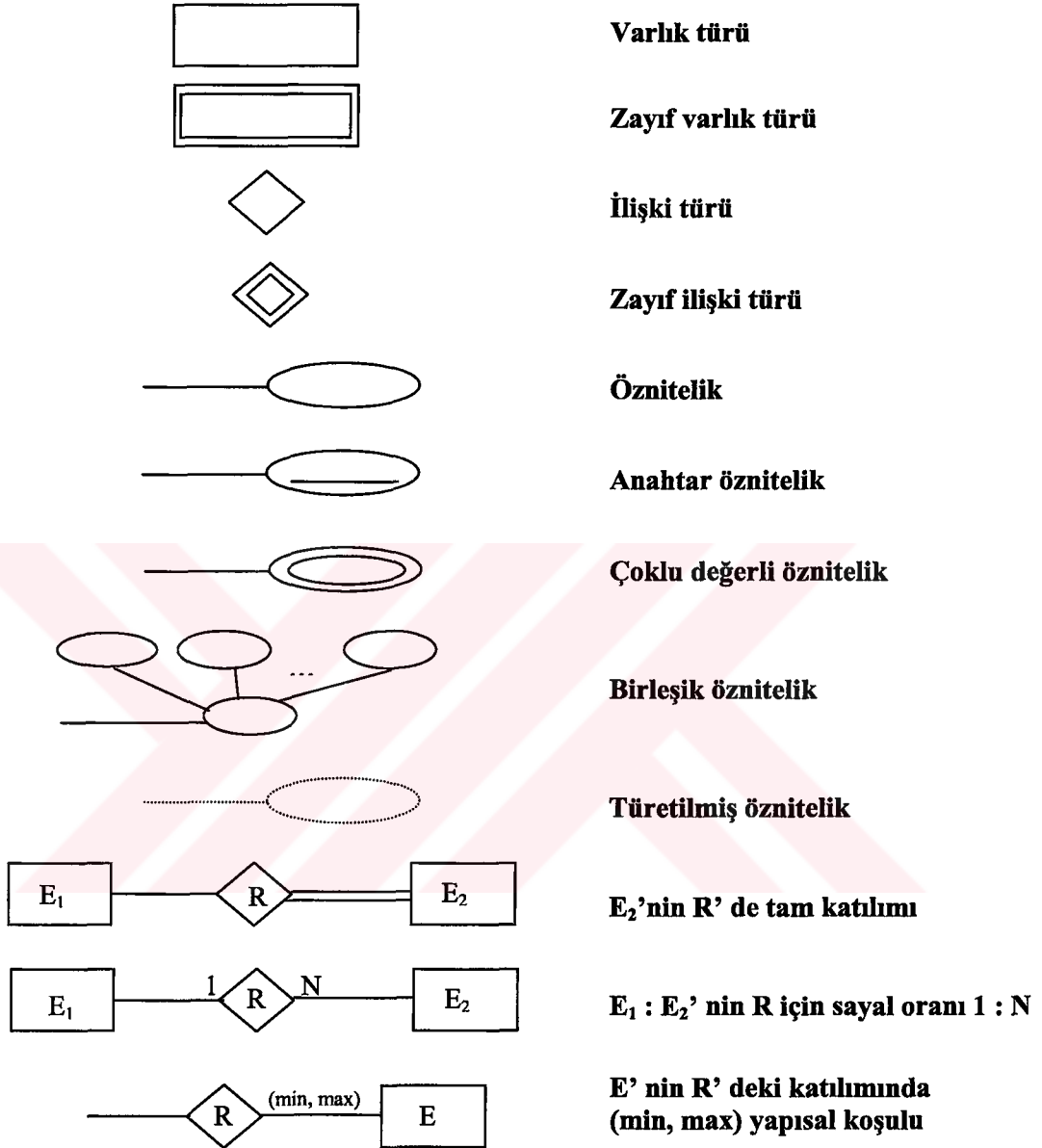
Şekil 4.7 Nesneye Yönelik Veri Modeli

4.2.1.2 Kavramsal Şema Tasarımına Yaklaşımlar

Kavramsal şema tasarımı için önce şemanın şu temel bileşenlerinin belirlenmesi gerekir: Varlık tipleri, ilişki tipleri ve öznitelikleri. Ayrıca anahtar öznitelikler, ilişkilerdeki katılım ve önem koşulları, zayıf varlık türleri, sıradüzenlerin özelleştirilmesi / genelleştirilmesi işlemleri de tanımlanmalıdır. Bu tasarım Aşama 1'de derlenen gereksinimlerden türetilir.

Kavramsal şema tasarımına iki tür yaklaşım vardır [13]: **Merkezi (tek – atım) şema tasarımı yaklaşımı** adı verilen birincisinde, Aşama 1'den gelen farklı uygulama ve kullanıcı grupları *kavramsal şema tasarımı başlamadan* bir gereksinim kümesi içinde birleştirilir. Daha sonra birleştirilmiş bu gereksinim kümesine karşılık tek bir şema tasarlanır. Çok sayıda uygulama ve kullanıcı olduğunda tüm gereksinimleri birleştirmek zor ve zaman alıcı bir görev olabilir. Bu yaklaşımın ardındaki kabul merkezi bir otoritenin (veritabanı yöneticisi) farklı kullanıcı ve uygulamaların gereksinimlerinin nasıl birleştirileceğine karar vermekten ve tüm veritabanı için

kavramsal şemayı tasarlamaktan sorumlu olmasıdır. Kavramsal şema tasarımı sona erdiğinde çeşitli kullanıcı grup ve uygulamaları için dış şemalar veritabanı yöneticisi tarafından belirlenebilir.

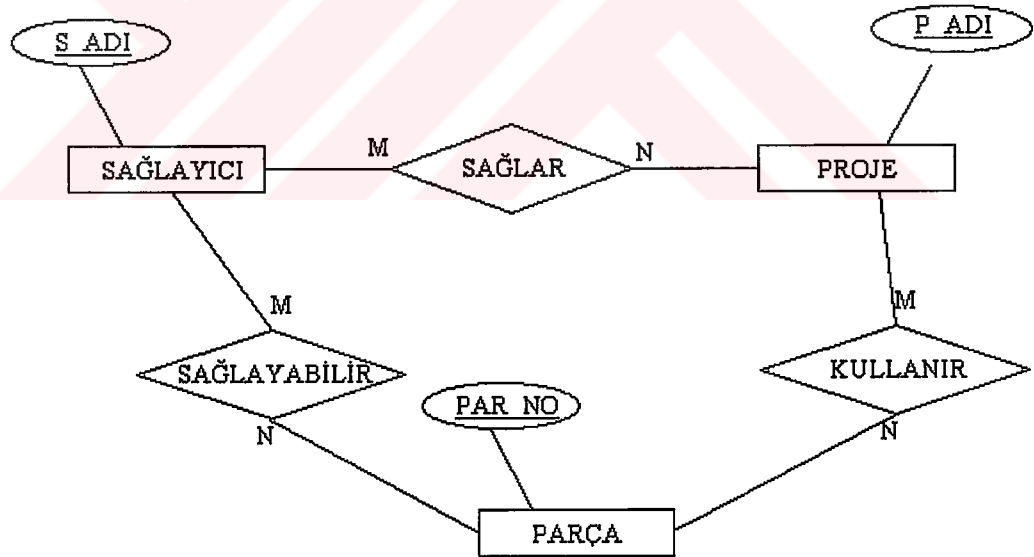


Şekil 4.8 Varlık - İlişki Gösterim İşaretleri Özeti

Görünüm birleştirme yaklaşımı adı verilen ikinci yaklaşımda, gereksinimlerin birleştirilmesi yerine her bir grup veya uygulamanın sadece kendi ihtiyaçlarına dayalı bir şema (veya görünüm) tasarlanır. Sonuç olarak, her grup ve uygulama için tek bir yüksek – seviye şema geliştirilir. Sonraki görünüm birleştirme aşaması sırasında, bu şemalar, tüm veritabanı için tek bir genel (global) kavramsal şemada birleştirilirler. Görünüm birleştirmesinden sonra her bir görünüm ayrı ayrı dış şemalar olarak yeniden oluşturulabilirler.

Bu iki yaklaşım arasındaki temel fark çok sayıda kullanıcı ve uygulamaya ait çeşitli görünüm ve gereksinimlerin bağdaştırılması safhasındaki tarzda yatmaktadır. Merkezi yaklaşımda, bağdaştırma şemaların tasarımından önce veritabanı yönetimi çalışanları tarafından elle yapılır ve Aşama 1’de derlenen gereksinimlere doğrudan uygulanır. Bu yaklaşım, kavramsal şema tasarımı uygulamaya konmadan genel gereksinimlerde açık ve anlaşılır tanımlar oluşturmak için kullanıcı grupları arasındaki farklılık ve çakışmaların bağdaştırılması konusunda veritabanı yönetimi çalışanlarına yığılı yükü rağmen çoklukla kullanılmıştır. Bu görevin yerine getirilmesindeki güçlükler yüzünden Görünüm Birleştirme Yaklaşımı günümüzde daha çok kabul görmeye başlamıştır [19].

Görünüm birleştirme yaklaşımında, her bir kullanıcı grubu veya uygulama, aslında kendi gereksinimlerinde kendi Vİ çizgesini tasarlar. Daha sonra bu çizgeler genel bir birleştirilmiş çizge oluşturmak için veritabanı yöneticisi tarafında bir işleme tabi tutulur. Görünüm birleştirme elle yapılabilmesine rağmen, onlarca kullanıcı içeren bir veritabanına uygulanabilmesi bir yöntem-bilim ve birleştirmeyi kolaylaştıracak araçların kullanımını gerektirir. Özniteliklerin uygunluğu, çeşitli görünümlerdeki varlık ve ilişki türleri, birleşimden önce belirlenmelidir. Ek olarak, çakışan görünümün birleştirilmesi ve belirli çizgeler arası uygunlukların kararlılığının doğrulanması gibi sorunlar çözülmelidir.



Şekil 4.9 Varlık – İlişki Veri Modeli

4.2.1.3 Şema Tasarımı İçin Stratejiler

Tek bir kullanıcı için yada bir kullanıcı grubu için bir gereksinim kümesi verildiğinde, bu gereksinimleri karşılayacak bir kavramsal şema yaratılmalıdır. Bu

tür bir şemanın tasarımı için çeşitli stratejiler vardır [20]. Bu stratejilerin çoğu artan bir yaklaşım izlerler; yani, gereksinimlerden türetilen şema yapıları ile başlarlar ve sonra artan bir şekilde bunları düzeltir, iyileştirir veya inşa ederler. Bu stratejiler şunlardır:

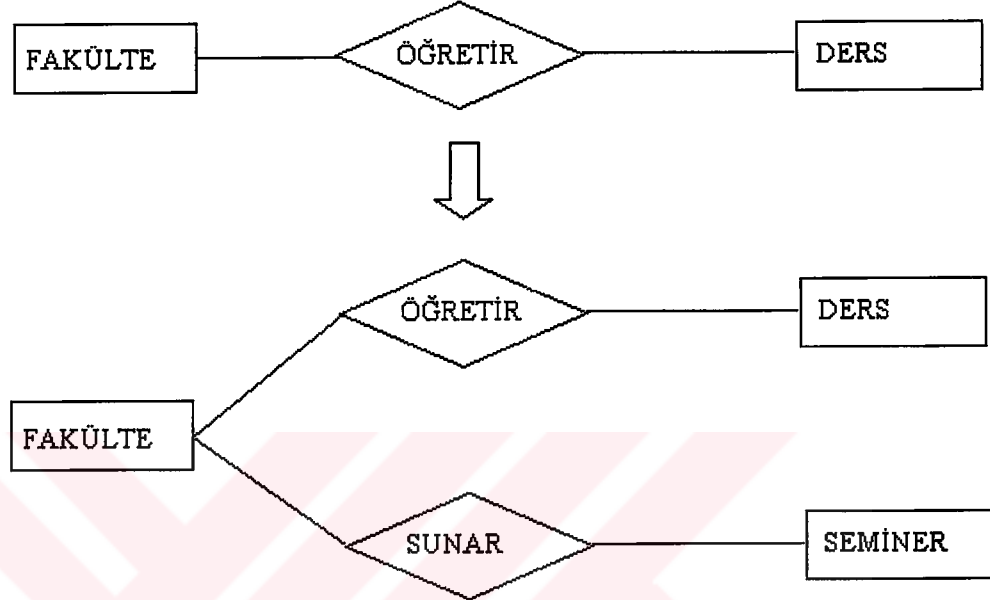
1. Yukardan – aşağı (top – down): Yüksek – seviye kavramlar (abstractions) içeren bir şema ile işe başlanır ve sonra ardışık yukardan – aşağı iyileştirmeler uygulanır. Örneğin; sadece birkaç yüksek – seviye varlık türünün belirlenmesi ile başlanabilir. Sonra, öznitelikler belirlenmeye başladıkça bunlar düşük – seviye varlık ve ilişkilere ayrıştırılabilir. Bir varlık türünü alt sınıflara ayrıştırarak iyileştirmek için yapılan belirleme işlemi de diğer bir yukardan – aşağı tasarım stratejisi örneğidir.
2. Aşağıdan – yukarı (bottom – up): Temel kavramlar içeren bir şemayla başlanır, sonra bunlar birleştirilir veya bunlara eklemeler yapılır. Örneğin; öznitelikler ile başlanabilir ve bunlar varlık türleri ve ilişkilerle gruplanabilirler. Alt sınıfları yüksek – seviye genel sınıflara genelleştirme işlemi de diğer bir aşağıdan – yukarı tasarım stratejisi örneğidir.
3. İçten – dışı (inside – out): Bu, dikkatin en açık kavramlardan oluşan kümede yoğunlaştırıldığı özel bir aşağıdan – yukarı tasarım stratejisi uygulamasıdır. Modelleme daha sonra hazır kavramlar etrafındaki yeni kavramların ele alınmasıyla içeriden dışarıya genişler. Örneğin; şemada birkaç aşık varlık türü belirlenip, her biri diğeriyle ilişkili diğer varlık türlerini ve ilişkileri bunlara ekleyerek devam edilebilir.
4. Karma (mixed): Tasarım boyunca özel bir strateji belirlemek yerine, gereksinimler yukardan – aşağı stratejiye göre bölümlenir ve her bir bölümün şema kısmı, aşağıdan – yukarı stratejiye göre tasarlanır. Değişik şema kısımları daha sonra birleştirilir.

Şekil 4.10 ve 4.11 sırasıyla yukardan – aşağı ve aşağıdan – yukarı iyileştirmeleri göstermektedir. İlkel bir yukardan – aşağı iyileştirmeye örnek bir varlık türünün birkaç varlık türüne ayrıştırılmasıdır. Şekil 4.10(a) DERS varlığının DERS ve SEMİNER varlıklarına, ÖĞRETİR ilişkisinin ÖĞRETİR ve SUNAR ilişkilerine ayrışmasını gösterir. Şekil 4.10(b) bir DERS_VERME varlığının aralarında bir ilişki olan iki varlık türüne ayrışmasını gösterir. Şekil 4.11 ilkel bir aşağıdan – yukarı iyileştirme olan varlık türleri arasında ilişki kurmayı gösterir.

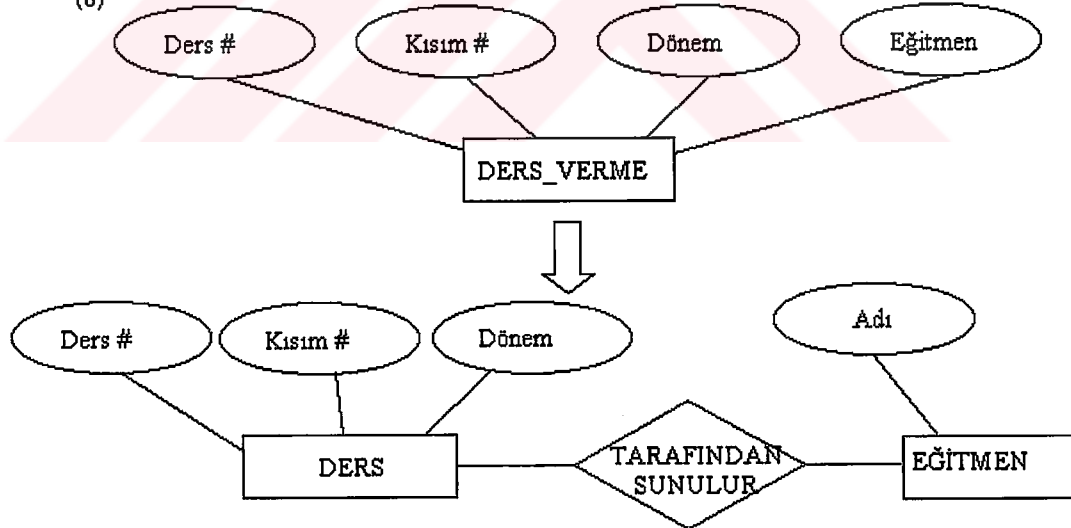
4.2.1.4 Şema (Görünüm) Birleştirme

Çok sayıda kullanıcı ve uygulamaya sahip büyük veritabanları için bireysel şemalar tasarlayıp birleştirmeden oluşan görünüm birleştirme yaklaşımı kullanılabilir. Bireysel görünümeler nispeten küçük olduğundan, şema tasarımı basitleştirilmiş olur.

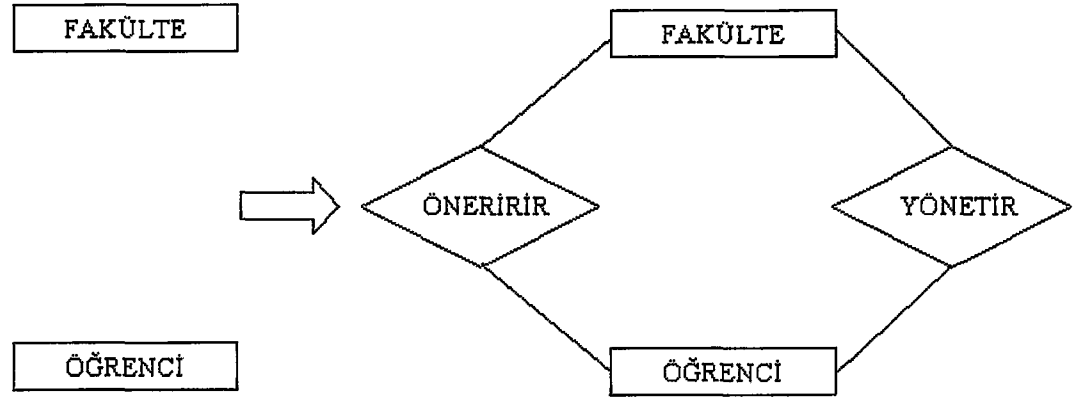
(a)



(b)



Şekil 4.10 Yukardan – Aşağı İyileştirme Örnekleri. (a) Yeni Bir Varlık Türü Türetme. (b) Bir Varlık Türünü İki Varlık Türü ve Bir İlişki Olarak Ayırma.



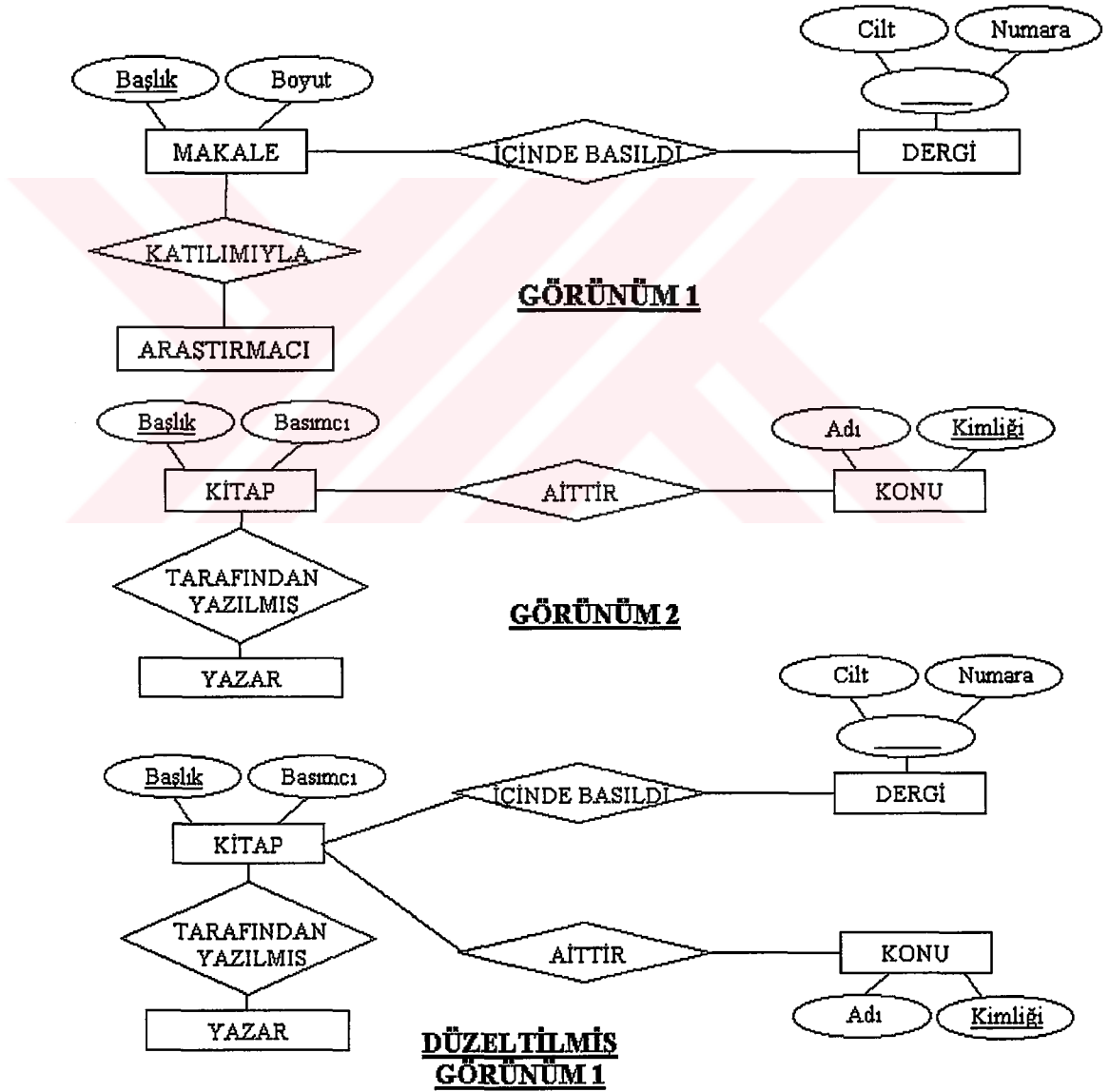
Şekil 4.11 Aşağıdan – yukarı İyileştirme Örneği. (Yeni Bir İlişkinin Bulunması ve Eklenmesi)

Bununla birlikte, görünümleri genel bir şemada birleştirmek için bir yöntem bilime ihtiyaç vardır. Şema birleştirimi aşağıdaki alt görevlere bölünebilir [21]:

1. Şemalar arası benzerlik (correspondence) ve çakışmaların bulunması: Şemalar bireysel olarak tasarlandığı için, aynı gerçek – dünya kavramını temsil edecek şema yapılarının belirlenmesi gerekmektedir. Birleştirme ilerlemeden önce bu benzerlikler bulunmalıdır. Bu işlem sırasında şemalar arası şu türden çakışmalar bulunabilir:

- a. İsimlendirme çakışmaları: Bunlar iki türdür: eş anlamlılar ve eşsesliler. Eş anlamlılar farklı şemalar aynı kavram için farklı isimler kullandığında ortaya çıkar. Örneğin; bir şemadaki MÜŞTERİ varlık türünün kastettiği anlam başka bir şemada ALICI varlık türünün kastettiği anlamla aynı olabilir. Eşseslilik ise farklı şemalarda farklı kavramlar için aynı isim kullanıldığında ortaya çıkar. Örneğin; PARÇA varlık türü bir şemada bilgisayar parçası anlamına gelirken bir başkasında mobilya parçası anlamına gelebilir.
- b. Tür çakışmaları: Aynı kavram farklı şemalarda farklı modelleme yapısı ile temsil edilebilir. Örneğin; ŞUBE kavramı bir şemada varlık türü iken bir başkasında öznitelik olabilir.
- c. Tanım kümesi çakışması: Bir öznitelik farklı şemalarda farklı tanım kümelerine sahip olabilir. Örneğin; SSN (Sosyal Sigorta Numarası) bir şemada tam sayı olarak kullanılırken bir başkasında karakter dizgisi olarak kullanılabilir. Bir şemada AĞIRLIK libre bir başkasında kilogram olarak temsil edildiğinde ölçü birimi çakışması meydana gelmiş olur.

2. Görünümleri birbirine uydurma: Bazı şemalarda diğer bazı şemalara daha çok uymaları için değişiklikler yapılabilir. Adım 1’de bahsedilen çakışmaların bazıları bu adımda giderilir.
3. Görünümleri birleştirme: Genel şema, bireysel şemaların birleştirilmesiyle oluşturulur. Birbirine benzeyen kavramlar genel şemada tek olarak temsil edilirler ve görünümlerin kendi aralarındaki ve genel görünümle aralarındaki eşlemeler (mappings) belirlenir.
4. Yeniden yapılandırma: İsteğe bağlı son bir adım olarak, genel şema çözümlenebilir ve gereksizlikleri (redundancies) gidermek için yeniden yapılandırılabilir.



Şekil 4.12 Birleştirmeden Önce Uyuşum İçin Görünümlerin Düzeltilmesi

Bu fikirlerin bazıları Şekil 4.12’de oldukça basit örneklerle gösterilmiştir. Şekil

4.12’de bir “Kaynakça Veritabanı” yaratmak için iki görünümün nasıl birleştirildiği gösterilmektedir. İki görünüm arasındaki benzerliklerin belirlenmesi sırasında, ARAŞTIRMACI ve YAZAR’ın, TARAFINDAN_DAĞITILIR ve TARAFINDAN_YAZILIR gibi eşanlamlı (bu veritabanı dikkate alındığında) olduğu bulunur. Sonra, GÖRÜNÜM 2’ye uyması için GÖRÜNÜM 1’e, MAKALE için bir KONU içermesi değişikliği yapılır.

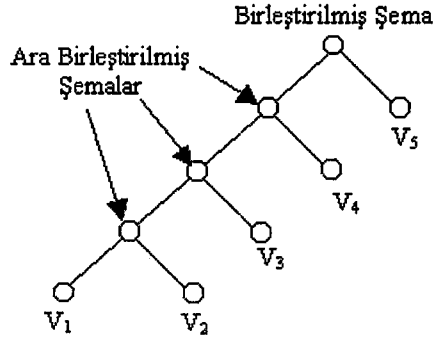
Görünüm birleştirme işlemi için çeşitli stratejiler önerilmiştir [22]. Şekil 4.13’de gösterildiği gibi bunlar şunlardır:

1. İkili merdiven birleştirme: İlk önce birbirine çok benzeyen iki şema birleştirilir. Sonra bu sonuç şema bir başkasıyla birleştirilir ve bu işlem tüm şemalar birleştirilene dek tekrar edilir. Bütünleştirme için şemaların sıralanmasında ölçüt olarak şema benzerliği alınabilir. Bu strateji izlediği adım – adım yaklaşımdan dolayı elle birleştirme için uygundur.
2. Çoklu birleştirme: Benzerliklerin çözümlenmesi ve belirlenmesinden sonra tüm görünüm tek bir işlemle birleştirilir. Bu strateji, büyük tasarım sorunlarından dolayı bilgisayarlı araçlar gerektirir.
3. İkili dengeli strateji: Önce şema çiftleri birleştirilir; sonra ileri birleştirmeler için sonuç şemalar eşleştirilir ve bu işlem sonuçta genel bir şema oluşturulana dek tekrar edilir [23].
4. Karma strateji: Başta, şemalar benzerlikleri göz önüne alınarak gruplara bölünür ve her bir grup ayrı olarak birleştirilir. Ara şemalar tekrar gruplanır ve birleştirilir ve böylece devam eder.

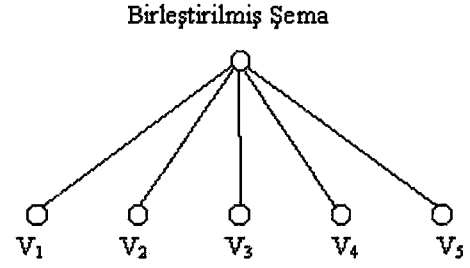
4.2.2 Aşama 2b: Veritabanı İşlemi (transaction) Tasarımı

Aşama 2a ile paralel ilerleyen bu aşamanın amacı bilinen veritabanı işlemlerinin özelliklerini VTYS – bağımsız bir şekilde tasarlamaktır. Bir veritabanı tasarlanırken tasarımcılar, veritabanı uygulamaya konduğunda veritabanı üzerinde çalışacak bilinen pek çok uygulamanın (işlemin) farkındadırlar. Bu işlemlerin işlevsel özelliklerinin erkenden tasarım işlemi sırasında belirlenmesi veritabanı tasarımının önemli bir bölümünü oluşturur. Böylece veritabanı şemasının bu işlemler tarafından gereken tüm bilgiyi içermesi garanti altına alınmış olur. Ek olarak, çeşitli işlemlerin göreceli önemlerinin bilinmesi ve bunların ümit edilen kullanılma oranları fiziksel veritabanı tasarımının (Aşama 5) önemli bir bölümünü oluşturur. Genel olarak, tasarım sürecinde sadece bazı veritabanı işlemleri bilinir; veritabanı sistemi uygulanmaya başladıktan sonra, yeni işlemler sürekli olarak tanımlanır ve

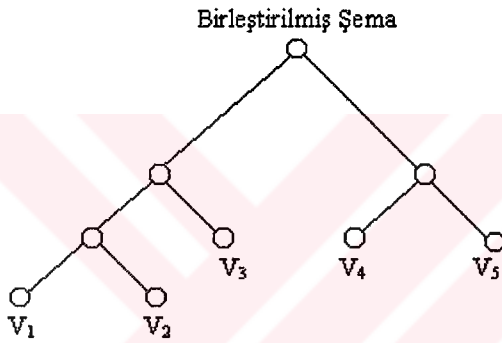
uygulanır. Bununla birlikte, çoğu önemli işlemler sistemin uygulanmasından önce bilinir ve ilk evrelerde tanımlanmalıdır.



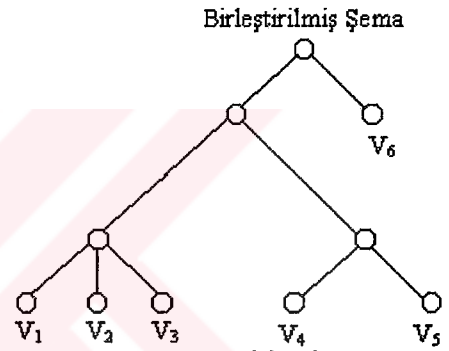
İkili Merdiven Birleştirme



Çoklu Birleştirme



İkili Dengeli Birleştirme



Karma Birleştirme

Şekil 4.13 Görünüm Birleştirmesi İçin Farklı Stratejiler

Kavramsal düzeyde işlem belirlemenin yaygın bir yordamı onların girdi / çıktı ve işlevsel davranışlarının tanımlanmasıdır. Girdi verinin, çıktı verinin ve denetimin iç işlevsel akışının belirlenmesiyle tasarımcılar bir işlemi kavramsal ve sistem – bağımsız bir şekilde belirleyebilirler. İşlemler genellikle üç kategoride gruplanırlar: istemsel (retrieval) işlemler, güncelleme işlemleri ve karma işlemler. İstemsel işlemler ekranda gösterim yada bir rapor üretimi için veri isteminde kullanılırlar. Güncelleme işlemleri, veritabanına yeni veri girişi veya varolan verinin değiştirilmesinde kullanılırlar. Karma işlemler, bazı istem ve güncellemeler yapan daha karmaşık uygulamalarda kullanılır. Bir havayolları rezervasyon veritabanı tasarlandığı varsayılırsa; belli bir tarihte iki şehir arasındaki sabah uçuşlarının listesini almak istemsel işleme, bir müşteri için bir uçuşa yer ayırmak güncelleme işlemine örnek verilebilir. Karma işleme, herhangi bir uçuştaki müşteri rezervasyonunu görüntülenmesi gibi önce veri gösterimi sonra rezervasyonun silinerek iptal edilmesi gibi bir veri güncellemesi, örnek gösterilebilir.

Gereksinim belirlemesi için çeşitli yordamlar, birkaç işlemin birleşiminden oluşan

işlem süreçlerinin belirlenmesi için işaret gösterimi içerirler. İşlem belirlemesindeki diğer öneriler ise TAXIS [24], GALILEO [25] ve GORDAS'dır [26]. İşlem tasarımı aynen şema tasarımı kadar önemlidir. Maalesef, yürürlükteki pek çok yöntem bilimlerini birini öbürüne üstün görmektedirler. Aşama 2a ve Aşama 2b geri besleme ilmleri ve iyileştirmeler kullanılarak kararlı bir şema ve işlem tasarımına ulaşılana dek paralel olarak devam etmelidir [27].

4.3 Aşama 3: Veritabanı Yönetim Sisteminin Seçimi

VTYS' nin seçiminde birkaç etmen vardır. Bazı etmenler teknik, bazıları ekonomik diğer bazıları ise kuruluş politikası ile ilgilidirler. Teknik etmenler, veritabanı yönetim sisteminin eldeki göreve uygunluğu ile ilgilendirler. Bu başlık altında şu hususlar değerlendirilir; veritabanı yönetim sistemi türü (ilişkisel, ağ, sıradüzensel, nesneye yönelik, vd.), veritabanı yönetim sisteminin desteklediği depo yapıları, erişim yolları, uygun kullanıcı ve yazılımcı ara yüzleri, yüksek – seviye sorgu dili türleri vb. Bu bölümde veritabanı yönetim sistemi seçimini etkileyen ekonomik ve kuruluşla ilgili etmenler üzerinde durulacaktır. Veritabanı yönetim sistemi seçiminde aşağıdaki maliyetler dikkate alınmalıdır [27]:

1. Yazılım alım maliyeti: Dil seçenekleri, formlar, ekranlar gibi değişik ara yüzler, yedekleme / kurtarma seçenekleri, özel erişim yöntemleri ve belgeleme yetenekleri ile birlikte yazılımın alınmasındaki başlangıç maliyetidir.
2. Bakım maliyeti: Veritabanı yönetim sistemini güncel tutmak için satıcının sürekli sağlaması gereken standart bakım hizmetidir.
3. Donanım alım maliyeti: Hafıza, uçbirim (terminal), disk birimi veya özel veritabanı yönetim sistemi belleği gibi yeni donanım gereksinimi olabilir.
4. Veritabanı yaratımı ve dönüşümü maliyeti: Veritabanı sistemini sıfırdan yaratma yada varolan sistemi yeni veritabanı yönetim sistemi yazılımına çevirme maliyetidir.
5. Personel maliyeti: Veritabanı yönetim sistemi yazılımının bir kuruluşa ilk alımını genellikle veri işleyen birimlerin yeniden örgütlenmesi izler. Veritabanı yönetim sistemini yeni edinen pek çok kuruluştaki veritabanı yönetimi çalışanları için yeni kadrolar açılır.
6. Eğitim maliyeti: Veritabanı yönetim sistemlerinin çoğunlukla karmaşık sistemler olması nedeniyle veritabanı yönetim sistemini kullanmak ve programlamak için çoğu personelin eğitimi gerekir.

7. İşletim maliyeti: Veritabanı sisteminin sürekli işletimi maliyeti, seçilen veritabanı yönetim sisteminden bağımsız olduğundan, genelde seçeneklerin değerlendirmesi gibi bir şey söz konusu değildir.

Veritabanı yönetim sistemi alımının yararları kolay ölçülüp nicelenemez. Veritabanı yönetim sisteminin geleneksel dosya sistemlerine göre kullanım kolaylığı, verinin daha yaygın kullanılabilmesi ve bilgiye daha hızlı ulaşım gibi, soyut avantajları vardır. Daha somut yararlar ise uygulama geliştirme maliyetinin azlığı, gereksiz (redundant) veri kullanımının azlığı ve daha iyi denetim ve güvenlik olarak sıralanabilir. Kuruluşlar yapacakları kar – maliyet çözümlemelerine göre veritabanı yönetim sistemine geçiş zamanına karar vermelidirler. Bu geçişi genellikle aşağıdaki etmenler etkiler;

- Veri karmaşıklığı: Veri ilişkileri karmaşıklaştıkça veritabanı yönetim sistemine olan ihtiyaç güçlenir.
- Uygulamalar arası paylaşım: Uygulamalar arası paylaşım arttıkça dosyalardaki gereksiz veri artar, dolayısıyla veritabanı yönetim sistemi gereksinimi artar.
- Devingen (dynamic) olarak büyüyen ve gelişen veri: Veri sürekli olarak değişiyorsa veritabanı yönetim sistemi kullanarak bu değişikliklerle başa çıkmak dosya sistemlerine oranla çok daha kolaydır.
- Anlık veri gereksinimlerinin sıklığı: Dosya sistemleri anlık veri istemine kesinlikle uygun değildir.
- Veri hacmi ve denetim gereksinimi: Yalnızca veri hacmi ve denetim gereksinimi bile bazen veritabanı yönetim sistemini gerektirir.

Son olarak, bazı ekonomik ve kuruluşla ilgili etmenler veritabanı yönetim sistemleri arasındaki seçimi etkiler;

1. Veri yapısı: Saklanacak veri sıradüzensel bir yapı içindeyse sıradüzensel türde bir veritabanı yönetim sisteminin alınması düşünülmelidir. Pek çok ara ilişkisi olan veriler için ağ yada ilişkisel sistem daha uygun olabilir. İlişkisel yordambilimler (teknolojiler) yaygınlık kazanmaktadır. Karmaşık veri yapıları ve veri türleri için neneye yönelik sistemler uygun olabilir.
2. Personelin sisteme aşinalığı: Kuruluştaki yazılımcılar belli bir veritabanı yönetim sistemine aşina iseler, eğitim maliyetini ve öğrenme zamanını azaltmak için o sistem tercih edilebilir.
3. Satıcı hizmetlerinin uygunluğu: Satıcı hizmet olanaklarının yakında olması sistemle ilgili sorunların çözümüne yardım açısında arzu edilir. VTYS olmayan

bir çevreden VTYS olan bir çevreye geiş genellikle büyük bir yükttür ve başlangıta satıcı yardımını çoka gerektirir.

Bir veritabanı yönetim sistemi satın alınmadan önce, kuruluş veritabanı yönetim sistemi tarafından gereken donanım / yazılım yapılanışı (konfigürasyonu) ve veritabanı yönetim sisteminin farklı donanım türlerine uygunluğunu dikkate almalıdır. Çok sayıda ticari veritabanı yönetim sistemi yazılımı farklı donanım / yazılım yapılanışlarında (yada platformlarda) çalışabilen sürümlere sahiptirler. Yedekleme, kurtarma, başarıml, bütünlük ve güvenlik için uygulama gereksinimleri de göz önünde tutulmalıdır. Çok sayıda veritabanı yönetim sistemleri hali hazırda kuruluşlardaki bilgi – işleme ve bilgi – kaynak yönetimi gereksinimlerine *toplam çözüm* olacak şekilde tasarlanmaktadır. Çoğu veritabanı yönetim sistemi satıcısı ürünlerini 4GL (fourth generation language, dördüncü kuşak diller) olarak kategorize ettikleri aşağıdaki seçeneklerle veya yerleşik (built – in) özelliklerle birleştirmektedirler:

- Metin düzenleyici ve göz atıcı
- Rapor üreticiler ve listeleme kolaylıkları
- İletişim yazılımları (çoğunlukla uzışlem görüntü birimleri “teleprocessing monitors” de denir)
- Otomatik düzenleme yetenekleri olan form, ekran ve menüler gibi veri giriş ve görüntüleme özellikleri
- Grafik tasarım araçları

Bazı durumlarda bir veritabanı yönetim sistemi kullanmak uygun olmayabilir, bunun yerine uygulamalar için kuruluş içi bir yazılım geliştirilmesi tercih edilebilir. Bu ise ancak uygulamalar önceden çok iyi tanımlı ve tümü biliniyorsa mümkündür. Böyle bir durumda, kuruluş içi, özel tasarım bir sistem bilinen uygulamaların en etkin şekilde gerçekleştirilmesi için uygun olabilir. Bununla birlikte, çoğu durumda tasarım sürecinde öngörölmeyen yeni uygulamalar sistem uygulanmaya başladıktan sonra ortaya çıkar. Veritabanı yönetim sistemlerinin yaygınlaşmasının asıl sebebi de budur; yeni uygulamaların hali hazır sisteme eklenmesini kolaylaştırır.

4.4 Aşama 4: Veri Modeli Dönüşümü (Mantıksal Veritabanı Tasarımı)

Veritabanı tasarımının sonraki adımı kavramsal bir şema ve seçili veritabanı yönetim sisteminin veri modeli cinsinden dış şemalar yaratma işlemidir. Bu da Aşama 2a’da üretilen kavramsal ve dış şemaların yüksek – seviye veri modelinden veritabanı

yönetim sisteminin veri modeline dönüştürülmesidir. Dönüşüm iki evrede ilerler:

1. Sistem – bağımsız dönüşüm: Bu adımda veritabanı yönetim sisteminin veri modeline dönüşüm sırasında veri modelinin veritabanı yönetim sistemine uygulanmasında hiçbir özel durum veya özellik dikkate alınmaz. VTYS – bağımsız Vİ şemasından ilişkisel, nesneye yönelik, sıradüzensel veya ağ şemasına dönüşümü anlatan kaynaklar da mevcuttur [27].
2. Şemaların belli bir veritabanı yönetim sistemine uyarlanması: Farklı veritabanı yönetim sistemleri özel modelleme özellikleri ve koşulları kullanarak veri modelini uygularlar. Adım 1’de elde edilen şemaları seçili veritabanı yönetim sisteminde kullanılan veri modelinin özelliklerine uydurmak için yeniden düzenlemek gerekebilir.

Dönüşüm sonucu elde edilen tablolar “Normalizasyon”a tabi tutulur. Bu aşamanın sonucu seçili veritabanı yönetim sistemi dilinden kavramsal ve dış düzeyde şemaları tanımlayan DDL (Veri Tanımlama Dili – Data Definition Language) komutları olmalıdır. Fakat DDL komutları bazı fiziksel tasarım ölçütleri içeriyorsa tam bir DDL tanımlaması, fiziksel veritabanı tasarımı evresi sonuçlanana kadar beklemelidir.

4.5 Aşama 5: Fiziksel Veritabanı Tasarımı

Fiziksel veritabanı tasarımı, çeşitli veritabanı uygulamalarında iyi bir başarıma ulaşabilmek için veritabanı dosyalarındaki depolama yapıları ve erişim yollarının seçilmesi işlemidir. Her veritabanı yönetim sistemi dosya organizasyonu ve erişim yolları için değişik seçenekler sunmaktadır. Bunlar genellikle dizinleme (indexing) türleri, ilgili kayıtların disk üzerinde kümelenmesi, ilgili kayıtların imleçler (pointers) ile bağlanması ve çeşitli sıralama (hashing) türleridir. Belli bir veritabanı yönetim sistemi seçildikten sonra fiziksel veritabanı tasarımı işlemi veritabanı dosyalarının yapısı için veritabanı yönetim sisteminin sunduğu seçenekler içinden en uygun olanının seçimi ile sınırlıdır. Aşağıdaki ölçütler fiziksel veritabanı tasarım seçeneklerinin seçiminde kılavuzluk ederler:

1. Yanıtlama süresi: Bir veritabanı işleminin gönderilmesi ve yanıtın alınması arasındaki zamandır. Veritabanı yönetim sistemi denetimindeki yanıtlama süresindeki ana etki, işlemin başvurduğu veri öğeleri için veritabanı erişim zamanıdır. Yanıtlama zamanı veritabanı yönetim sistemi denetimde olmayan sistem yüklenmesi, işletim sistemi zamanlaması (scheduling) veya iletişim gecikmeleri gibi etmenlerden de etkilenir.

2. Alan kullanımı: Veritabanı dosyaları ve erişim yol yapıları tarafından kullanılan depolama alanı miktarıdır.
3. İşlem ürünü: Veritabanı sistemi tarafından dakikada işlenebilen ortalama işlem sayısıdır; havayolu rezervasyon ve banka gibi işlem sistemlerinde kritik bir ölçüttür.

Genel olarak, anlatılan ölçütlerdeki ortalama ve en kötü durum sınırlar sistem başarımlarının gereksinimlerinin bir parçası olarak belirlenir. Önürünleme (prototyping) ve benzetim (simulation) gibi çözümsel (analitik) veya deneysel yordamlar, belirli başarımların gereksinimlerinin karşılanıp karşılanmadığının belirlenmesi için çeşitli fiziksel tasarım kararlarında ortalama ve en kötü durum değerlerinin tahmininde kullanılırlar.

Başarım kayıt büyüklüğüne ve dosyadaki kayıt sayısına bağlıdır. Bu yüzden her dosya için bu ölçütlerin tahmin edilmesi gerekir. Ek olarak, dosyanın tüm işlemlerden birikimli olarak güncelleme ve erişim şekilleri tahmin edilmelidir. Kayıtların seçimi için kullanılan özniteliklerin birincil erişim yolları ve ikincil dizinleri olmalıdır. Yeni özniteliklerin eklenmesinden yada kayıtların artmasından kaynaklanan dosya büyümesinin tahmini de fiziksel veritabanı tasarımı sırasında hesaba katılmalıdır.

Fiziksel veritabanı tasarımının sonucu veritabanı dosyalarına ait depo yapılarının ve erişim yollarının *ilk* belirlemesidir. Veritabanı uygulanmaya başladıktan sonra veritabanı sisteminin ölçülen başarımları temel alınarak, tasarımda ince ayarlar hemen hemen her zaman gereklidir. Pek çok sistem, sonraki çözümlerinde kullanılmak üzere sistem katalog ve veri sözlüklerinde tutulan başarımların istatistikleri toplamak için bir izleme kolaylık yazılımı içerir. Bu istatistikler önceden tanımlı işlem veya sorgulamaların çağrılma sayıları, dosyalardaki girdi / çıktı etkinlikleri, dosya sayfalarının yada dizin kayıtlarının sayısı ve dizin kullanımının sıklığı gibi değerlerdir. Veritabanı sistemi gereksinimleri değiştikçe, bazı dosyaları yeni dizinler oluşturarak veya birincil erişim yöntemlerini değiştirerek yeniden düzenlemek gerekir.

4.6 Aşama 6: Veritabanı Sisteminin Uygulanması

Mantıksal ve fiziksel tasarımlar tamamlandıktan sonra, veritabanı sistemi uygulamaya konabilir. Seçili veritabanı yönetim sisteminin DDL (Veri Tanımlama Dili – Data Definition Language) ve SDL (Depo Tanımlama Dili – Storage Definition Language) komutları derlenir ve veritabanı şema ve (boş) veritabanı

dosyalarını yaratmak için kullanılır. Bundan sonra veritabanı veriyle **yüklenebilir** (doldurulabilir). Eğer veri önceki bir bilgisayarlı sistemden dönüştürülecekse, veriyi yeniden düzenlemek ve yeni veritabanına yüklemek için **dönüşüm yordamları** gerekebilir.

Bu evrede veritabanı işlemleri uygulama yazılımcıları tarafından uygulanmalıdır. İşlemlerin kavramsal tanımlamaları incelenir ve bunlara ilişkin gömülü DML (Veri İşleme Dili – Data Manipulation Language) komutları ile yazılım kodları yazılır ve sınanır. İşlemler hazır olduğunda ve veri yüklendiğinde, tasarım ve uygulama evresi bitmiş ve veritabanı sisteminin işletim evresi başlamış demektir.



5. COĞRAFI BİLGİ SİSTEMİ KAVRAMI

Coğrafi bilgi sistemleri (CBS) için kısaca “bilgi sistemlerinin coğrafi veriye uygulanmış özel bir formudur” denilebileceği gibi çok daha ayrıntılı ve kapsamlı tanımlamalar da yapmak mümkündür. Coğrafi bilgi sistemleri için günümüze kadar bir çok tanımlama yapılmıştır. Bir kısmının aşağıda sıralandığı bu tanımlamalarda coğrafi bilgi sistemleri değişik bakış açıları ve değişik vurgulamalar ile açıklanmıştır.

Coğrafi bilgi sistemleri, karmaşık planlama ve yönetim sorunlarının çözülebilmesi için tasarlanan, mekandaki konumu ve öznitelikleri belirlenmiş verileri içeren, yönetimi, işlenmesi, analiz edilmesi, modellenmesi ve görüntülenebilmesi işlemlerini kapsayan, kullanıcıların daha doğru kararlar vermesi, üretimi arttırması, zaman, para ve işgücü tasarrufu sağlayan donanım, yazılım, yöntemler ve personelden oluşan bir bilgi sistemidir.

Coğrafi bilgi sistemleri, konuma bağlı varlıklara ait grafik ve grafik olmayan bilgilerin toplanması, bir coğrafi veritabanı biçiminde organize edilmesi, işlenmesi, sorgulanması ve kullanıcılara çizimler, ekran görüntüleri veya raporlar biçiminde sunulmasını sağlayan yazılım / donanım sistemleridir.

Coğrafi bilgi sistemleri, belli bir amaç ile yeryüzüne ait verilerin toplanması, depolanması, sorgulanması, transferi ve görüntülenmesi işlevlerini yerine getiren araçların tümüdür.

Coğrafi bilgi sistemleri, kartografik verilerin depolanması, sorgulanması, analizi ve görüntülenmesine yarayan bir bilgisayar programıdır.

Coğrafi bilgi sistemleri, kullanıcının daha iyi karar verebilmesi, verimliliğin geliştirilmesi, zaman, ekonomi ve insan gücünde kazanç sağlanması amaçlarıyla bir araya getirilmiş coğrafi veri, bilgisayar donanımı ve yazılımı, personel ve ilgili özniteliklerin bir bütünüdür.

Coğrafi bilgi sistemleri, coğrafi veya konumsal koordinatlar ile temsil edilen verilerle çalışacak şekilde tasarlanmış bir bilgi sistemidir. Diğer bir deyişle coğrafi bilgi sistemleri, hem konuma ilişkin veriler için özel yetenekleri olan veritabanı sistemi

hem de veriler üzerinde çalışacak işlemlerden oluşan bir kümedir.

Yukarıda sıralanan tanımların ışığında coğrafi bilgi sistemlerini şu şekilde tam olarak ifade edebiliriz: Coğrafya ile ilgili grafik ve grafik olmayan verilerin kullanıcı ihtiyaçlarını karşılayacak biçimde çeşitli kaynaklardan toplanması, depolanması, işlenmesi, analiz edilmesi, yönetilmesi ve sunulması fonksiyonlarını bütünleşik olarak yerine getiren coğrafi veri, donanım, yazılım ve personel bileşenlerinden oluşan bir organizasyondur.

En basit ifade ile coğrafi verileri saklayan, sorgulayan ve kullanan bir bilgi sistemidir.

Genel anlamdaki tanımlamalar ile ilgili problem, coğrafi bilgi sistemleri ile sokak adresleri, posta kodları gibi konuma ilişkin diğer bilgi sistemlerinin ayrımındaki gerçeğin göz ardı edilmesidir. Bazı sistemler, hem isim olarak hem de fonksiyonel olarak coğrafi bilgi sistemlerine benzemelerine karşın yukarıda tanımlandığı şekilde gerçek coğrafi bilgi sistemi değildirler. Bu tür sistemler coğrafi bilgi sistemlerinin yapabildiği ölçüde karmaşık coğrafi analizleri yapabilme olanaklarına sahip değildirler.

Coğrafi bilgi sistemleri, her ne kadar değişik ölçeklerde, değişik projeksiyonlarda ve değişik renklerde harita yaratabiliyorsa da, harita yapımı için kullanılan basit bir bilgisayar sistemi değildir. Coğrafi bilgi sistemleri analiz eden bir alettir. Coğrafi bilgi sistemlerinin asıl avantajı, harita özellikleri arasındaki konumsal ilişkileri tanımlamaya olanak vermesidir.

Coğrafi bilgi sistemleri haritaları ve resimleri tutmaz, veritabanını tutar. Veritabanı kavramı coğrafi bilgi sistemlerinin kalbidir. Ayrıca coğrafi bilgi sistemleri veritabanı da depolanmış verileri kullanarak, harita üzerindeki detaylara ilişkin yeni bilgiler hesaplar.

Coğrafi bilgi sistemleri veri yapıları yönünden diğer bilgi sistemlerinden farklıdır. Çünkü coğrafi bilgi sistemlerindeki varlıklar coğrafi varlıklardır ve bu nedenle grafik olmayan bilgilerin yanında grafik bilgilerin de bütünleşik olarak tutulması gerekmektedir. Diğer yandan coğrafi bilgi sistemlerinde varlıklar arasında alışagelmış ilişkilerin dışında konuma bağlı ilişkiler de (komşuluk, içermeye vb.) yer almaktadır.

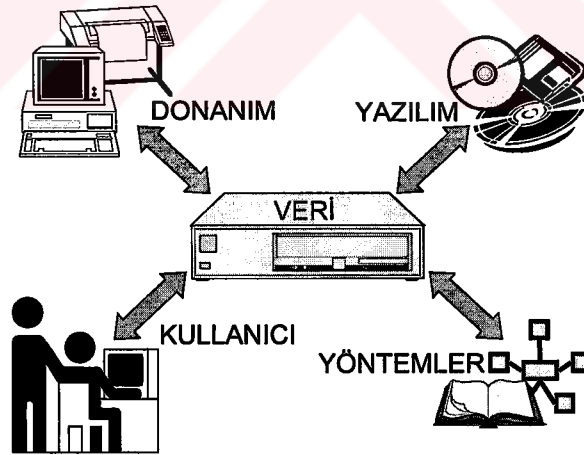
5.1 Coğrafi Bilgi Sisteminin Bileşenleri

CBS' lerine fonksiyonel, teknolojik ve yönetim açılarından bakıldığında farklı bileşenlerden oluştuğu görülür.

Fonksiyonel açıdan bakıldığında bir CBS 'de veri aktarma, veri depolama, veri işleme, coğrafi analiz ve veri sunma bileşenlerinin var olması gerekir. Teknolojinin rolü ise bu fonksiyonların gerçekleştirilmesini sağlayan donanım ve yazılım araçları sunmaktan ibarettir. Yönetim ise fonksiyonel ve teknolojik bileşenlerle insan ve mali kaynakların yönetimini birleştirip bir bütün oluşturmayı ve amaca ulaşmayı hedefler.

Bir CBS' nin kurulabilmesi ve kendinden beklenen işlevleri yerine getirebilmesi için gerekli olan temel elemanlar şunlardır (Şekil 5.1);

1. Coğrafi veriler
2. CBS donanımı
3. CBS yazılımı
4. Personel
5. Organizasyon (yöntemler)



Şekil 5.1 CBS Bileşenleri

5.1.1 Coğrafi Veriler

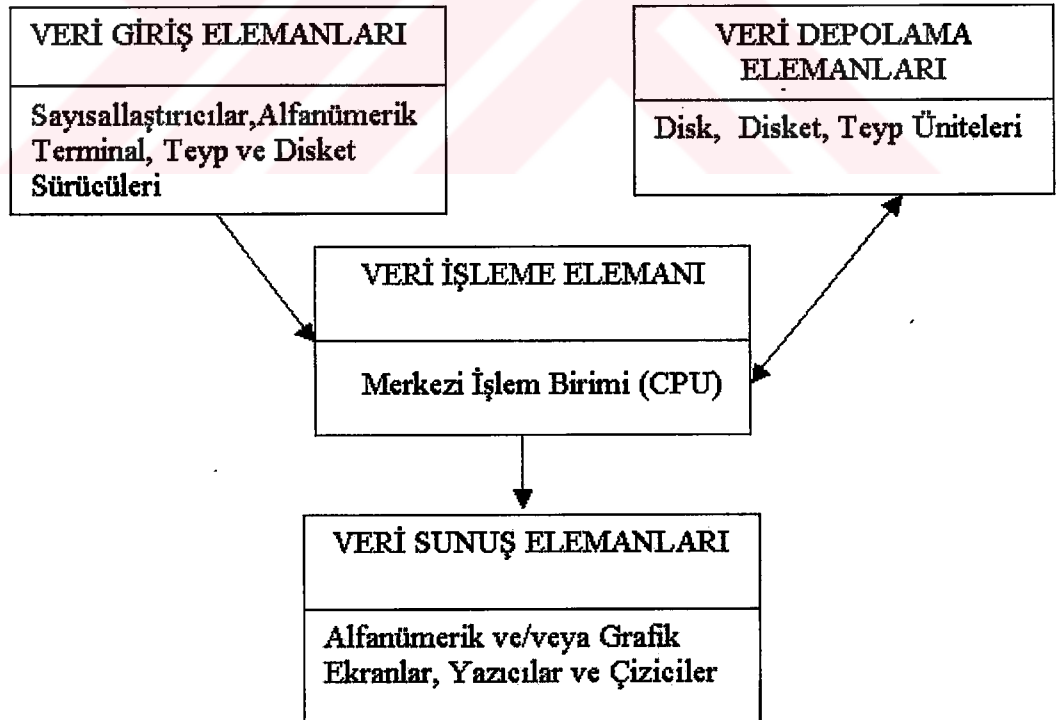
Coğrafi veriler; konumsal (spatial; mekansal, grafik), öznitelik (tanımsal, grafik olmayan) ve topolojik verilerdir. Coğrafi veriler veritabanında depolanırlar. Toplanması uzun zaman alması ve oldukça yüksek maliyetler gerektirmesi bakımından CBS' nin en önemli bileşenidir. Çok iyi geliştirilmiş olsa bile, veri

bileşeni olmadan bir CBS' nin çalıştırılması mümkün değildir.

Bilgisayar teknolojisindeki hızlı gelişim ve CBS projelerinde çalışan personelin değişimi CBS bileşenlerinde devingenlik (dinamizm) sağlamaktadır. Bu nedenle CBS geliştiricileri genellikle veri – merkezli (data – centered) yaklaşımları desteklemektedirler. Veri, kendi amaçlarını gerçekleştirmek için tek bir kullanıcı yada tek bir kurum tarafından toplanabilmeli, aynı zamanda birden çok kullanıcı ve birden çok yazılım / donanım tarafından paylaşılabilmelidir. Diğer bileşenler ise verinin işlenmesi için gerekli olan desteği sağlamalıdır. Bütün CBS paketleri coğrafi varlıkların hem konum bilgilerini hem de bu varlıklara ait öznitelik bilgileri depolama, işleme ve analiz kabiliyetlerine sahiptirler.

5.1.2 Coğrafi Bilgi Sistemi Donanımı

CBS donanım bileşeni; çeşitli kaynaklardan verilerin toplanması, işlenmesi, depolanması ve sunulmasını sağlayan elemanlardan oluşmaktadır. CBS donanım bileşeni, veri toplamada kullanılan veri giriş birimleri, verilerin üzerinde işlemler yapıp depolandığı bilgisayarlar ve bilgilerin sunulmasında kullanılan veri sunuş birimleri olmak üzere üç gruba ayrılır (Şekil 5.2).



Şekil 5.2 CBS Donanım Bileşenleri

5.1.2.1 Veri Giriş Elemanları

Bunlar manuel sayısallaştırıcılar, raster tarayıcılar, vektör tarayıcılar, video sayısallaştırıcılar, yarı otomatik sayısallaştırıcılar, görüntü algılayıcıları, elektronik takeometreler , GPS alıcıları, sayısal görüntü kıymetlendirme cihazları, alfanümerik terminal, teyp, disket ve CD - ROM sürücülerdir. Çeşitli kaynaklardan verilerin toplanmasını sağlarlar.

5.1.2.2 Veri Depolama ve İşleme Elemanları

Bunlar büyük bellek kapasitesine sahip bilgisayarlar ya da çalışma istasyonları (workstations), kişisel bilgisayarlar ve mini bilgisayarlardır. Bilgisayarların; görüntü birimi, sistem birimi ve klavye olmak üzere fiziksel üç temel bileşeni bulunmaktadır. Monitör olarak da isimlendirilen görüntü birimi yazı ve grafik şekillerin görüntülenmesi işlemlerinde kullanılmaktadır. Sistem biriminin içinde, bilgisayarın beyni yani merkezi işlem birimi (CPU : Central Processing Unit), kontrol devreleri, bellek ve disk üniteleri bulunmaktadır. Klavye ise bilgisayarın temel veri giriş birimidir. Bilgisayar belleği ve disk veri depolama elemanı, merkezi işlem birimi ise veri işleme elemanıdır.

5.1.2.3 Veri Sunuş Elemanları

Bu son grup ise verinin karar vericilere sunuş bilgisayar çıktılarını verecek cihazlardan oluşur. Bunlar; alfanümerik ve/veya grafik ekran ve terminaller, yazıcılar, çiziciler, film kayıt aletleri, film pozlayıcılar, çinko kalıp pozlayıcılar ve direkt baskı makineleri ile teyp, disket ve CD-ROM sürücüler gibi yardımcı çevre birimlerinden oluşmaktadır.

Donanım, CBS için oldukça esnek bir yapıya sahiptir. CBS için bir konfigürasyon oluştururken bu birimlerden hangilerinin kullanılacağını belirlemede dikkate alınacak en önemli husus, yapılacak uygulamanın gereksinimleridir. Kullanıcılar ihtiyaçları doğrultusunda değişik bilgisayar yapıları belirleyebilirler. Daha sonraları ise teknolojik gelişmelere paralel olarak, konfigürasyonda değişiklikler yapılabilir.

5.1.3 Coğrafi Bilgi Sistemi Yazılımı

Bilgisayarların kavramsal bileşenlerini oluşturan yazılımlar, kullanım amaçları açısından işletim sistemi yazılımları, programlama dilleri ve uygulama yazılımları olmak üzere üç ana gruba ayrılmaktadır.

Sistem yazılımları, bilgisayarların işletilmesi ile ilgili birçok işlevi yerine getirmeleri nedeniyle işletim sistemi yazılımı şeklinde de ifade edilmektedirler. İşletim sistemi yazılımı bilgisayarların açıldıklarında ilk ve kapanmadan önce son çalışan yazılımlardır. İşletim sistemleri diğer uygulama programlarının işlemlerini ve çevre birimleri ile olan giriş – çıkış işlemlerini kontrol ederler. Ayrıca kullanıcıların bilgisayarı kullanmalarını, veri depolamalarını ve bilgiye erişmelerini sağlarlar. Yaygın işletim sistemi yazılımları;

- Kişisel bilgisayarlarda MS-DOS, OS/2, LINUX, Win95/98/ME, WinNT/2000,
- Çalışma istasyonlarında UNIX yada türevi veya benzeri olan AIX, XENIX, LINUX, CLIX, ULTRIX vb.,
- Ana bilgisayarlarda genellikle donanım üreticisi tarafından geliştirilen; Digital Equipment Corp. (DEC) firmasının VAX serisi bilgisayarlarda VMS, Prime serisi bilgisayarlarda PRIMOS, IBM bilgisayarlarda CMS, Data General bilgisayarlarda AOS/VS vb. yazılımlardan oluşmaktadır.

Programlama dilleri ve derleyiciler, kullanıcılar ile merkezi işlem biriminin tanıyabildiği makine kodu arasında çevirici işlevi oluşturmaktadırlar. Delphi, C, Pascal, Fortran, BASIC gibi yüksek seviyeli programlama dilleri CBS gibi uygulama yazılım paketlerinin geliştirilmesinde kullanılırlar. Ayrıca Arc/Info için AML, MicroStation için MDL ve AutoCAD için AutoLISP gibi bazı uygulama yazılımlarının kullanımında yararlanılan özel amaçlı programlama dilleri de bulunmaktadır.

Uygulama yazılımları, işletim sistemi işlemlerinin gerçekleştirilmesi ve program yazma işlemlerinin kapsamı dışında kalan tüm işlemlerin yapılmasında kullanılan yazılımlardır. CBS paketleri, kelime işlemciler, hesap çizelgesi paketleri, Bilgisayar Destekli Tasarım Sistemleri, Veritabanı Yönetim Sistemleri, uçuş kontrol sistemleri, benzetim (simulator) sistemleri gibi bir çok uygulama yazılımı örneği sıralamak mümkündür. Bununla birlikte uygulama yazılımları genellikle kullanıcılar tarafından amaca ve uygulama alanına göre geliştirilmekte ve isimlendirilmektedir. Örneğin ; Ankara Kent Bilgi Sistemi, Orman Bilgi Sistemi.

CBS yazılımı ise genel amaçlı hazırlanmış ve birçok analiz türünü içeren ticari bir yazılımdır. Piyasada bulunan mevcut CBS yazılımlarına örnek olarak Arc/Info, ArcView, MGE, TNTmips ve TIGRIS verilebilir.

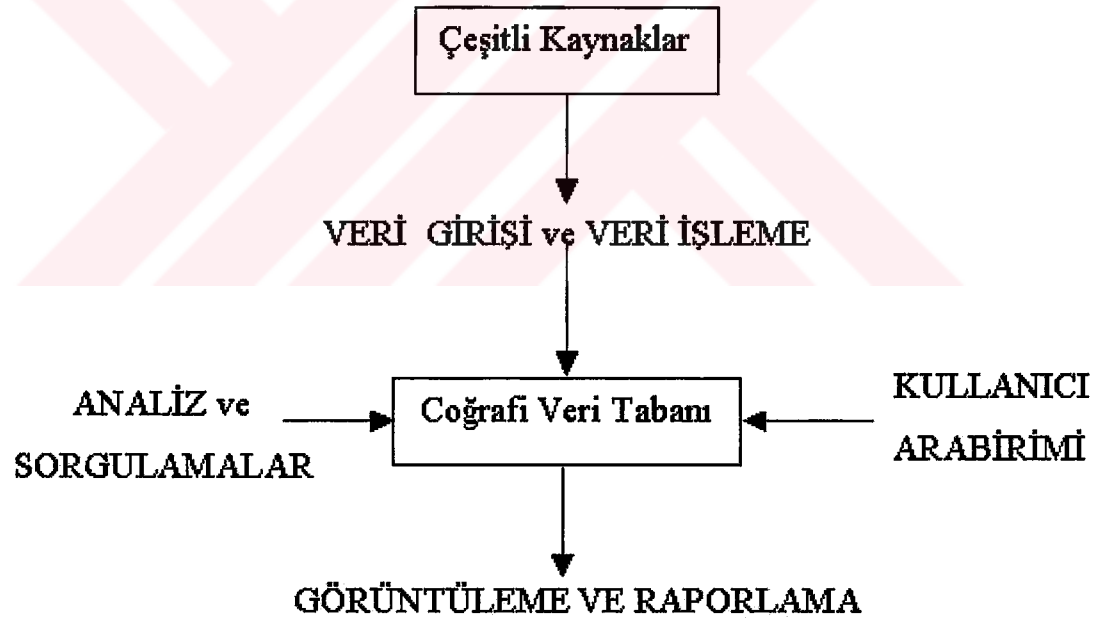
Bir CBS yazılımı beş ana bileşenden oluşmaktadır (Şekil 5.3):

- Coğrafi veri girişi ve veri işleme yazılımları

- Coğrafi veri depolama, erişim ve coğrafi veritabanı yönetim yazılımları
- Coğrafi veri sorgulama ve analiz yazılımları
- Coğrafi veri sunuş (görüntüleme ve sonuç ürünler hazırlama) yazılımları
- Kullanıcı arabirimi

5.1.3.1 Coğrafi Veri Girişi ve Veri İşleme Yazılımları

Değişik türdeki coğrafi verileri (çizgisel haritalar, hava fotoğrafları, arazi ölçmeleri gibi), uygun CBS veri giriş elemanları (sayısallaştırıcılar, alfanümerik terminal gibi) kullanılarak bilgisayar tarafından anlaşılır formlara dönüştürme işlemlerini yapan ve daha sonra bilgisayar ortamına aktarılmış verilerin doğruluğunu kontrol etmek ve hatalı olanları düzeltmek için kullanılan yazılımlardır. Örneğin ARC/INFO yazılımı içinde Arcedit, APS, Tables/INFO, Editplot yazılımları, kağıt haritaların sayısallaştırmalarında kullanılan sayısallaştırma masaları ve tarayıcı yazılımları ile uydu algılayıcı verileri ve arazi ölçme verilerinin manyetik ortamlarda depolanması ve bilgisayar ortamlarına aktarılmasını sağlayan yazılımların tümü.



Şekil 5.3 CBS Yazılım Bileşenleri

Veri girişi yazılımı dış dünya ile ilişkidir (Şekil 5.4 – A). Kullanıcılar bu yazılımı kullanırlar. Bu veriler birçok ara programdan geçerek bilgisayar ortamına gelir. Veri girişi yazılımları olabildiğince değişik kaynaklardan veri toplayabilmelidir. İyi bir CBS çeşitli türde sayısallaştırıcılar, tarayıcılar, etkileşimli grafik terminaller gibi birimler yardımıyla bu kaynaklardan veri toplayabilmeli, ayrıca diğer sistemler (uydu

algılayıcıları veya diğer CBS'leri) tarafından toplanmış ve manyetik ortamda bulunan verileri de gerekli dönüşümlerle okuyabilir özellikte olmalıdır.

Veri işleme yazılımları bilgisayar ortamına girilen verilerin doğruluğunu kontrol etmek ve hatalı olanlarını düzeltmek için kullanılan yazılımlardır. Grafik olmayan verilerin girişi sırasında yapılan hatalar, hatalı verilerin silinerek yeniden girilmesi şeklinde manuel giderilebilmesine karşın; grafik verilerin girişinde yapılan hataların manuel olarak giderilmesi kısmen mümkün olmakla beraber oldukça güç ve karmaşıktır. Çünkü;

- Grafik veri hatalarının manuel düzeltilmesi sırasında yeni hataların oluşma olasılığı yüksektir.
- Grafik veri hatalarının manuel belirlenmesi insanın görme duyarlılığına bağlıdır.
- Grafik veri hatalarının manuel düzeltilmesi insanın el hareketi duyarlılığına bağlıdır.

Belirlenecek ve düzeltilecek hatalı verilerin çok miktarda ve birbirine yakın olması durumunda bunları birbirinden ayırmak güçtür.

Veri işleme yazılımlarının içerdiği fonksiyonlar iki ana gruba ayrılır :

- Veri bakımı ve idaresi fonksiyonları
- Veri kullanımı ve analizi fonksiyonları

Veri bakımı ve idaresi fonksiyonları aşağıdaki işlemleri kapsar :

- Grafik detayların güncelleştirilmesi (ekleme, silme, değiştirme, güncelleştirme)
- Haritanın / grafik detayın ölçeğinin değiştirilmesi, ötelenmesi, döndürülmesi,
- Haritanın projeksiyon sisteminin başka bir projeksiyon sistemine dönüştürülmesi
- Komşu haritaların kenarlaştırılması
- Pan / zoom / window işlemleri
- Vektör veri yapısından raster veri yapısına dönüşüm ve ters işlemi
- Kartografik genelleştirme, çizgi yumuşatma, yeniden sınıflandırma işlemleri
- Distorsiyon giderme (rubber sheeting) işlemi

Veri Kullanımı ve Analizi Fonksiyonları aşağıdaki işlemleri kapsar :

- Çevre ve alan hesabı
- Münhaniden matris veya matristen münhani türetme
- 3 boyutlu görüntü elde etme
- Gölgeleme, kesit çıkarma, perspektif görüşler elde etme, meyil tarama
- Simulasyon

5.1.3.2 Coğrafi Veri Depolama, Erişim ve Coğrafi Veritabanı Yönetimi Yazılımları

Coğrafi veri giriş yazılımları ile girilen ve daha sonra veri işleme yazılımları ile hatalardan arındırılan ve istenilen şekle getirilen coğrafi verileri, belli bir veritabanı modeline uygun şekilde coğrafi veri depolama tekniklerinden birini kullanarak bilgisayar ortamında depolayan ve yöneten coğrafi veritabanı yönetimi sistemi yazılımlarıdır.

Coğrafi veri depolama teknikleri ile grafik türdeki coğrafi veriler veri yapılarına göre iki grupta toplanır:

- Raster yapıdaki coğrafi veri depolama yöntemleri
- Vektör yapıdaki coğrafi veri depolama yöntemleri

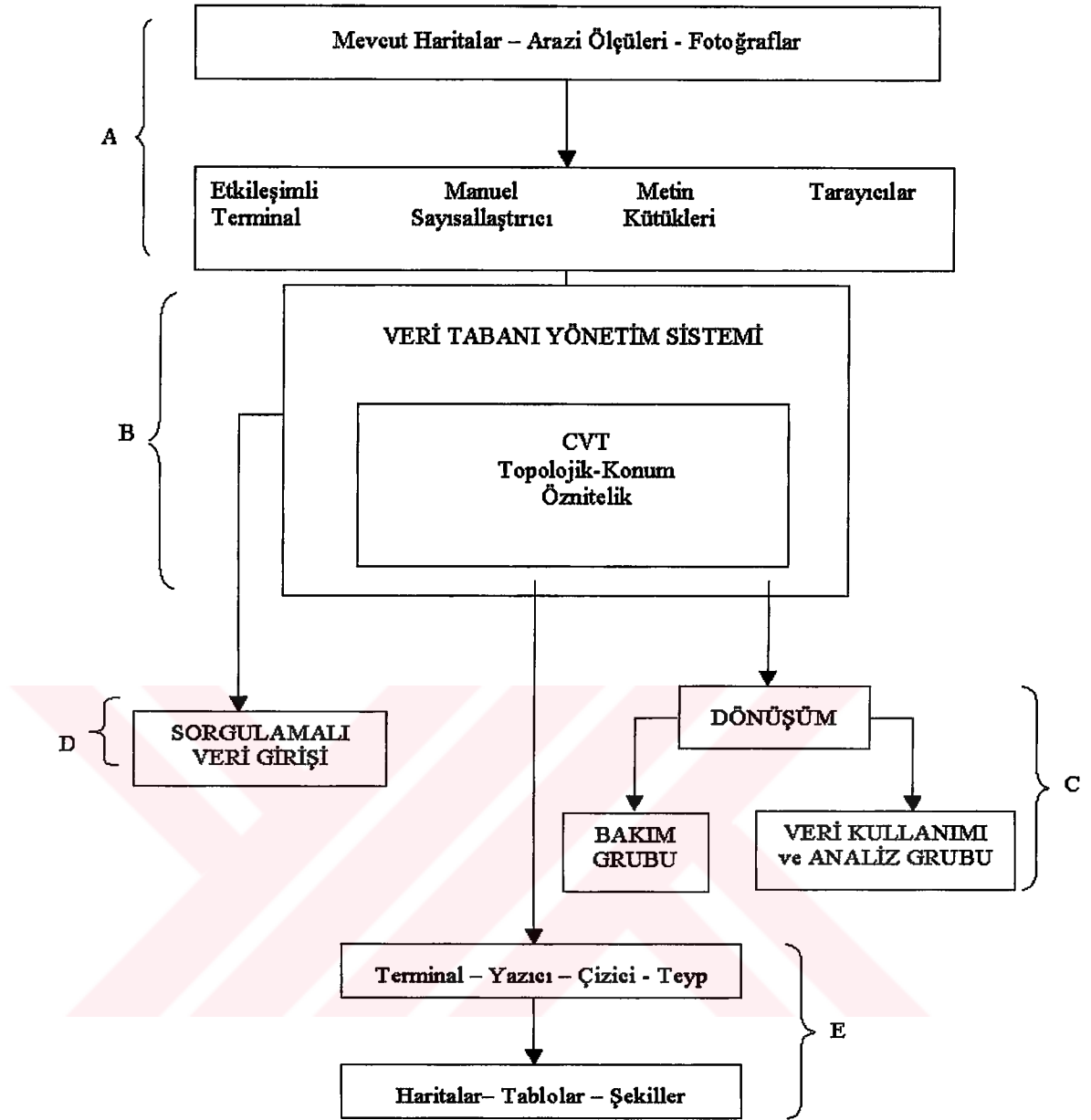
Çeşitli kaynaklardan gelen veriler Coğrafi Veritabanında (CVT) toplanır. (Şekil 5.4 – B). CVT bir VTYS ile yönetilir. Veriler arasında topolojik, konumsal ve öznitelik ilişkileri kurulur. Veriler burada yapılandırılır. CBS' nin kalbi coğrafi veritabanıdır.

5.1.3.3 Coğrafi Veri Sorgulama ve Çözümleme Yazılımları

CBS' nin diğer sistemlerden ayıran önemli özellikleri içeren yazılım bileşenidir. Coğrafi verilerin konumları ile ilgili analizleri (çakışma, kesişme, komşuluk vb.) gerçekleştiren, elde mevcut verilere göre sorgulamalar ve istatistik analizler yapabilen yazılımlardır (Şekil 5.4 – D). Kullanıcının yapacağı sorgulama ve analizlere göre veri girişi yapılmalıdır.

İyi bir CBS yazılımında klasik bir bilgi sistemindeki alışılmış sorgulamalara ek olarak konuma bağlı sorgulamaların da yapılabilmesi mümkün olmaktadır. Konuma bağlı sorgulamalara aşağıdakiler örnek olarak verilebilir; 7813 numaralı parsel nerede, alanı 40 hektardan büyük olan bahçeler hangileridir, eğimi %30 u geçen yerlerdeki toprak sınıfları nelerdir, X yerindeki toprak derinliği nedir? Coğrafi analizlere örnek olarak da; B den D kadar mesafede kaç tane A var?, X ten Y ye

giden ve P den geçen en kısa yol hangisidir?



Şekil 5.4 CBS Yazılım Bileşeni

Bu sorgulama ve analiz sayılarını çoğaltmak mümkündür. Kullanıcı ihtiyaçlarının çok iyi belirlenmesi sorgulama imkanlarını arttıracaktır. Çünkü bu sorgulamaların yapılması, verilerin sorgulanmaya imkan verecek şekilde girilmesiyle artacaktır.

5.1.3.4 Coğrafi Veri Sunuş (Görüntüleme ve Sonuç Ürünler Hazırlama) Yazılımları

Bilgisayar ortamındaki harita bilgilerinin veya analiz, sorgulama gibi işlemlerin sonuçlarının kullanıcının istediği formlarda grafik veya listeler şeklinde ekranda veya

izici, yazıcı ve pozlayıcı gibi aralar yardımı ile sunulmasına yarayan yazılımlardır.

eřitli niteliklerdeki terminallerden; yazıcılara, izicilere, manyetik ortama kadar uzanan birimler aracılığı ile eřitli nitelikte haritalar, tablolar ve řekiller retebilen modllerden oluřur. Veritabanından bilgiler alınır ve kullanıcılara sunulur (řekil 5.4 – E).

5.1.3.5 Kullanıcı Arabirimi

CBS ile kullanıcılar arasında iletiřimi saėlayan ve verilerin dzenlenmesi (silme, ekleme, deėiřtirme vb.), sorgulama, sorulara yanıt alınması, sistem kullanımının ėrenilmesi, sistemin izlenmesi vb. iřlemlerin yapılmasına olanak saėlayan menlerden oluřmaktadır. Ayrıca kullanıcı tarafından sistemi ynlendirecek programlar yazmaya yarayan makro programlama dilleri ve sorgulama dilleri de kullanıcı arabirimi iinde deėerlendirilmektedir.

Bu yazılım bileřenlerinden anlařılacağı zere CBS, klasik grafik olmayan bilgi sistemlerinden, rneėin bir personel bilgi sisteminden konuma baėlı bilgileri ve bylece topolojik iliřkileri iermesi bakımından farklılařmaktadır. Buna baėlı olarak da etkileřimli analiz ve sorgulama olanakları da olduka zengindir.

5.1.4 Coėrafi Bilgi Sistemi Personeli

CBS' de bileřenlerin organizasyonunu, planlanması ve tasarımını doėru biimde yapabilecek ve ynetecek olan personeldir. Tm bileřenleri transfer ederek uygulama ve geliřtirmek mmkndr. Ancak, sistemin bařarısı bu teknolojileri kullanacak personel ve yneticilerin eėitimine baėlıdır ve en nemli faktr bu konuda 'yetiřmiř insan' dır.

Bir CBS' nin personel bileřenini oluřturan personel on grupta toplanabilir:

- Sistem yneticisi
- Sistem zmleyicisi (analizcisi)
- Veritabanı yneticisi
- Veri iřleme uzmanı
- Veri reticisi
- Veri giriř operatr
- Harita mhendisi

- Bilgisayar mühendisi / teknisyeni
- Uygulama programcısı
- Son kullanıcılar

Yöneticiler izlenecek politikaları, işlem adımlarını belirlerler, gerekli amaç ve araçları tespit ederler ve zaman planlaması yaparlar. Veri üreticileri CBS' de kullanılacak grafik ve grafik olmayan veriyi toplayan personeldir. CBS' de konuma bağlı analizlerin yapılabilmesi için toplanan verilerin de bu işlemlere uygun yapıda olması gerekmektedir. Veri üreticilerinin de veriyi uygun şekilde toplayabilmeleri için bu özelliklerden haberdar olmaları ve CBS gereksinimlerini bilmeleri gerekmektedir. Sistem kullanıcıları (veri işleme uzmanı, veri giriş operatörü, programcılar ve mühendisler) CBS' nin yazılım ve donanımını aktif olarak kullanan personeldir. Sistem kullanıcıları problem çözümlerinde CBS' nin nasıl uygulanacağı konusunda gelişmiş teknik bilgiye sahiptirler. Sistem kullanıcıları sistemin sadece günlük kullanımından değil aynı zamanda sistemin yaşatılması ve güncel tutulmasından da sorumludurlar. Son kullanıcılar ise CBS tarafından üretilen bilgileri kullanırlar. Teknik bilgiye sahip olmalarına gerek yoktur. Bunun yanında, sistemin olanakları ve kısıtlarını bilmek açısından sistem kullanıcıları ile sürekli iletişimde bulunmalıdırlar. Son kullanıcılar günlük işlemlerinde CBS kullanımının gerekli olup olmadığına karar verebilmek açısından CBS' nin fayda ve maliyetlerini çok iyi bilmek durumundadırlar.

5.1.5 Organizasyon (Yöntemler)

CBS bileşenlerinde organizasyon (yöntemler) kapsamında veri yönetimi, personel idaresi, sistem geliştirme, sistem bakımı, planlama, maliye ve prosedürler yer alır.

Başarılı bir CBS, her kuruluşun kendine özel çalışma deneyimleri ve modellerinden oluşan iyi tasarlanmış planlara ve iş kurallarına bağlı olarak işlevini sürdürür. CBS'leri, coğrafi bilgilerin depolanması, erişilmesi ve görüntülenmesi için geliştirilmiş bir araç olmasına rağmen, verilerinin kaliteli olmasına ve devamlılığının sağlanmasına özen gösterilmelidir.

Birçok yeni CBS proje geliştiricisi için bir CBS çözümünün seçilmesi ve yürütülmesi genellikle karmaşık ve oldukça uğraştırıcı bir işlem olmaktadır. Queen ve Blinn, CBS çözümlerinin kullanıldığı her türlü durumda, mevcut veya yeni tüm CBS kullanıcılarına sistemin temini, projeye başlama veya var olan bir projede yeni bir projenin başlatılması işlemlerinden önce stratejik bir planlama işlemine zaman ayırmalarını tavsiye etmektedirler. Ayrıca CBS çözümlerini kullanmaya başlayan

yeni kullanıcıların, kendileri ile benzer işlemleri yapan mevcut CBS kullanıcıları ile teknolojinin kullanımına yönelik yöntemler hakkında görüş alışverişinde bulunmaları, projelerin yürütülmesinde yararlı olacaktır.



6. UYGULAMA (JEODEZİK BİLGİ SİSTEMİ)

Bu bölümde anlatılan aşamalara ait yapılan Jeodezik Bilgi Sistemi uygulamasından nirengi ile ilgili bölüm alınarak bahsedilecektir.

6.1 Aşama 1: Gereksinimlerin Derlenmesi ve Çözümlemesi

Bir örneği aşağıda verilen ankete benzer anketler anahtar personele verilmiş, veri alışverişinde kullanılan formlar (EK – A) tespit edilmiştir [28] [29]:

1. Nirengilere ait konum bilgileri nelerdir?

Enlem ve boylam (coğrafi koordinatlar). Bunlar UTM (sağa, yukarı ve dilim orta meridyeni – dom) cinsinden de ifade edilir. Ayrıca yükseklik bilgisi de tutulur.

2. Bu bilgiler nasıl üretilir?

Arazide yapılan açı (yatay ve düşey) ve mesafe ölçümlerinin dengelenmesiyle üretilir.

3. Bu bilgileri hangi idari birim üretir?

Jeodezi Şubesi.

4. Bu bilgileri hangi idari birim sunar?

Jeodezik ve Jeofizik Değerler Şubesi (JJDS).

5. Bir nirengi noktası tek anlamlı olarak nasıl tanımlanır?

İçinde bulunduğu 1 / 100.000 ölçekli paftanın adı ve nokta numarası ile.

6. Bu bilgiler kimlere sunulur?

Fotogrametri Dairesi, Tapu ve Kadastro Gn. Md.lüğü, İller Bankası, firmalar, diğer bazı KİTler.

7. Nirengi değerleri ile ilgili konumdan başka ne tür bilgiler, hangi formatta tutulmalıdır?

- Nokta numarası (101)
- İçinde bulunduğu pafta (Elbistan K36 –a1)
- Nokta derecesi (I, II, III, IV)
- Kot verilme türü (Trigonometrik, Nivelman)
- Tesis edilme türü (Beton Blok, Pilye, Bina Üstü, Kayada Bronz, Minare)
- Tesis yılı (1966)

8. Bu bilgileri hangi tarzda istenmektedir?

- Adı verilen bir pafta içine giren belirli nitelikte (örneğin; İnci derece, Beton Blok) noktalar
- 1 / 100.000 ölçekli pafta adı ve nokta numarası verilmek suretiyle istenen noktalar
- Köşe noktaları ile tanımlanan bir alan içine giren noktalar
- Konumu tanımlanan bir noktaya belli bir uzaklıktaki noktalar
- Pafta üzerinden görerek seçilen noktalar.

9. Bu bilgilerin sunumunda ne tür formlar kullanılır?

EK – A’ de verilen formlar ve protokoller kullanılır.

10.Nirengilere ait sizin tutulması gerektiğini düşündüğünüz başka bilgi var mıdır?

Mevcut sistemde araziye çıkanlardan bazı noktaların tahrip olduğuyula ilgili bilgi gelmekte fakat bu bilgi saklanmamaktadır. Bu bilginin de tutulmasının yararlı olduğu değerlendirilmektedir.

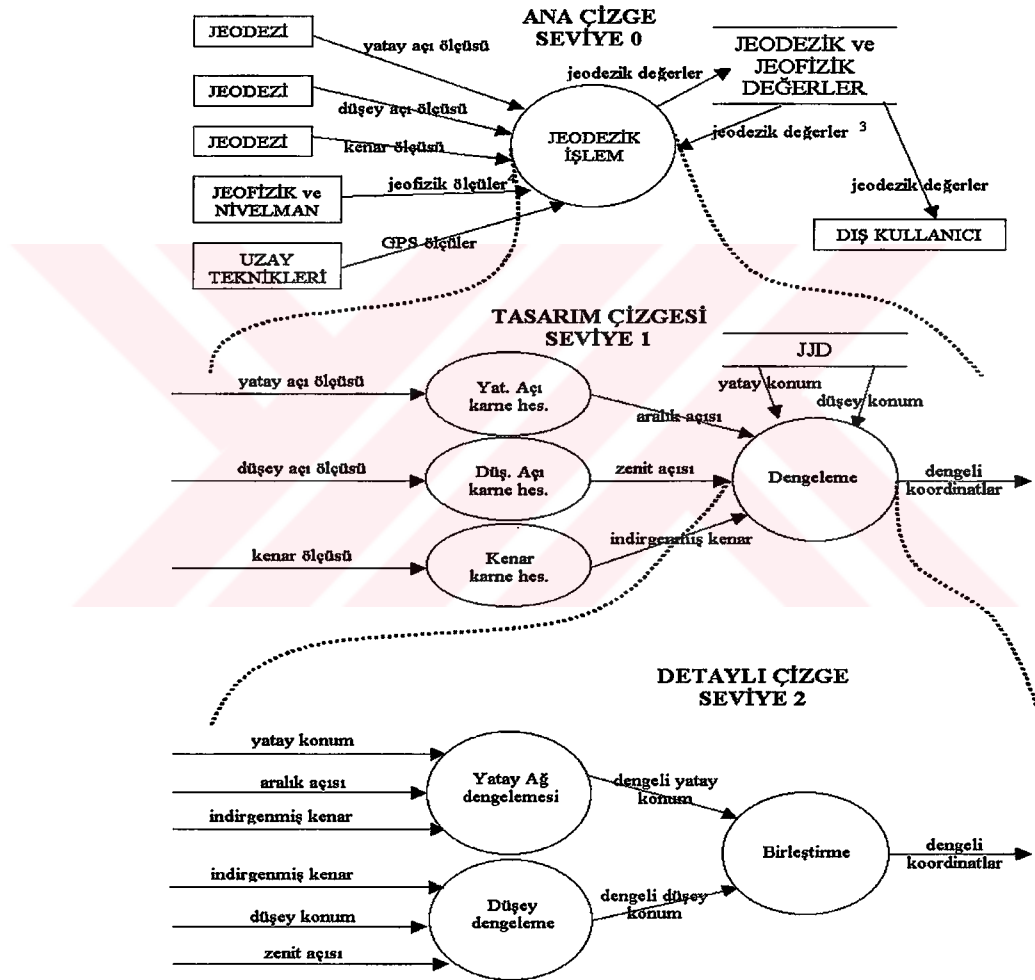
Bu kapsamda Şekil 6.1’ deki VAÇ’leri oluşturulmuştur.

6.2 Aşama 2: Kavramsal Veritabanı Tasarımı

Bölüm 4.2’ de anlatıldığı gibi veritabanı tasarımının ikinci aşaması iki paralel etkinlik içerir. Bu bağlamda hazırlanan Vİ çizgesi EK – B’ de sunulmuştur. Aşama 2b’de değinilen işlem tasarımı için de istemsel işleme örnek teşkil edebilecek yukarıdaki anketin sekizinci sorusuna verilen cevaplardaki gereksinimlerin bazılarını karşılamak üzere aşağıdaki SQL komutları tasarlanmıştır:

- Adı verilen bir pafta içine giren belirli nitelikte (örneğin; İnci derece, Beton Blok) noktalar:

select * from nirengi
 where paf250 = 'Elbistan' and
 paf100 = 'K36' and
 paf50 = 'a' and
 paf25 = '1' and
 tesis_türü = '1' and
 nokder = '2'



Şekil 6.1 JBS Veri Akış Çizgeleri (nirenge)

- 1/ 100.000 ölçekli pafta adı ve nokta numarası verilmek suretiyle istenen noktalar

select * from nirengi
 where paf100 = 'K36' and nok_no = '101'

6.3 Aşama 3: Veritabanı Yönetim Sisteminin Seçimi

Bu çalışmanın amaçlarına uygunluğu, yaygın kullanımı ve kolay anlaşılması gibi etmenlerden dolayı Microsoft ACCESS 2000 yazılımı VTYS olarak seçilmiştir.

6.4 Aşama 4: Veri Modeli Dönüşümü (Mantıksal Veritabanı Tasarımı)

Bu aşamada Vİ çizgelerinden tablolar üretilmiştir. Bunun için aşağıda uygulanarak anlatılan şu adımlar izlenmiştir [27] (burada f.k. (foreign key) yabancı anahtar anlamına gelmekte olup birincil anahtarların da altı çizilidir):

1. Zayıf olmayan her bir varlık için tüm özniteliklerini içeren bir tablo yaratılır ve birincil anahtar olacak öznitelik belirlenir.
 - Personel [sicil, adı, soyadı]
 - Nirengi [paf100, nokno, paf250, paf50, paf25, sağa, yukarı, dom, eder, edak, esan, bder, bdak, bsan, yük, tesyıl, drc]
 - Kot Türü [kod, tür]
 - Tes. Türü [kod, tür]
2. Her bir zayıf varlık için tüm özniteliklerini içeren ayrıca sahibinin birincil anahtarını yabancı anahtar olarak alan bir tablo yaratılır. Sahibinin ve varsa kendisinin birincil anahtarlarının birleşimi tablonun birincil anahtarı olur.
 - Ölçü [PerSicil(f.k), Tarih, YDurNok, YBakNok1, YBakNok2, AraAçı, DDurNok, DBakNok, Zenit, KDurNok, KBakNok, Mesafe]
 - Protokol [NirPaf100(f.k.), Nirnokno(f.k.), tarif, kroki, kaba değer]
3. Tüm bire bir ilişkilerde ilişkinin bir tarafındaki varlığa K diğerine H denirse (tam katılımı olana K demek daha doğru olur), H'nin birincil anahtarı K'ye yabancı anahtar yapılır.
 - Nirengi [paf100, nokno, paf250, paf50, paf25, sağa, yukarı, dom, eder, edak, esan, bder, bdak, bsan, yük, tesyıl, drc, KotKod(f.k.), TesKod(f.k.),]
4. Tüm bire çok ilişkilerde ilişkiye çoklu (N) olarak katılan varlığa K diğerine H denirse, H'nin birincil anahtarı K'ya yabancı anahtar olarak eklenir.
 - Protokol [NirPaf100(f.k.), Nirnokno(f.k.), tarif, kroki, kabadeğer,

PerSicil(f.k.)]

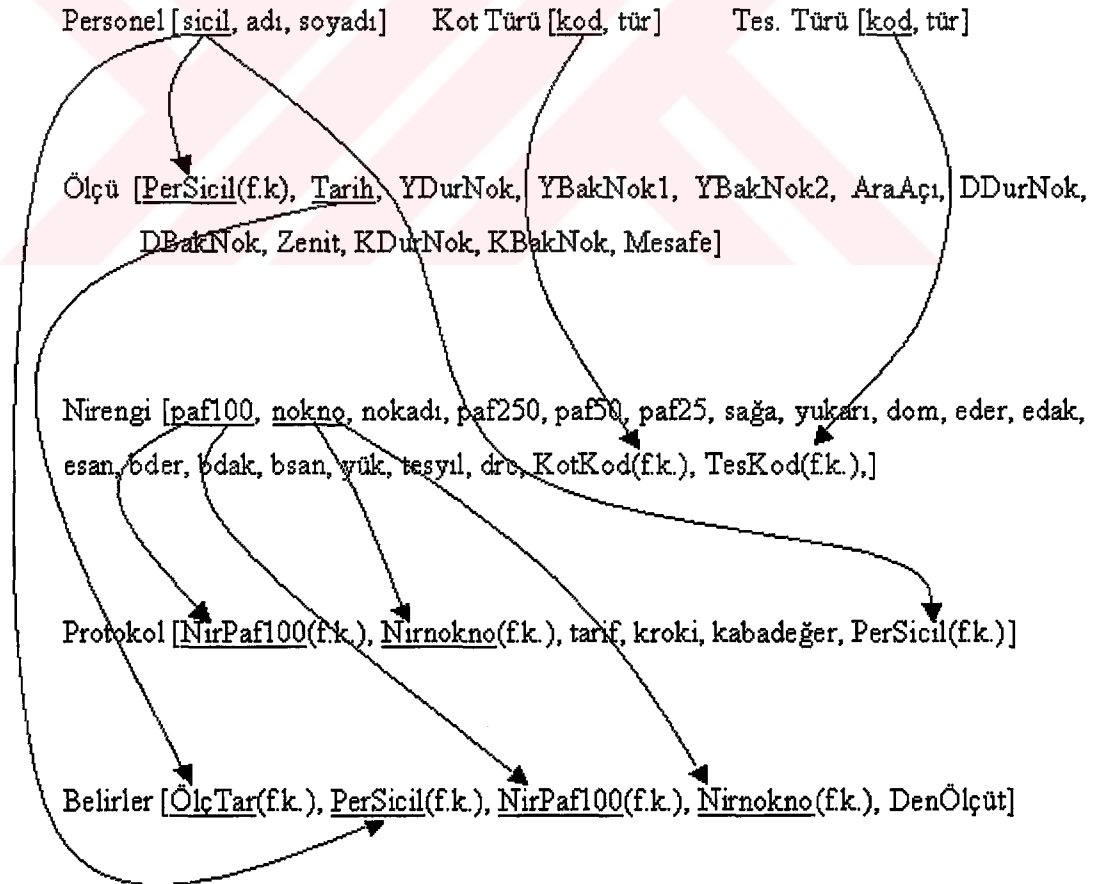
5. Tüm çoka çok ilişkiler için bir tablo yaratılarak ilişkiye katılan tüm varlıkların birincil anahtarları bu tabloya yabancı anahtar olarak eklenir ve bunların birleşimi tablonun birincil anahtarı olur. Varsa ilişkinin öznitelikleri de bu tabloya eklenir.

• Belirler [ÖlçTar(f.k.), PerSicil(f.k.), NirPaf100(f.k.), Nirnokno(f.k.), DenÖlçüt]

6. Tüm çoklu değerli öznitelikler için bir tablo yaratılır. Çoklu değer ait olduğu tablonun birincil anahtarı ve çoklu değer kendisi bu tabloya öznitelik olarak girer. Bunların birleşimi de birincil anahtar olur. Kullanılan örnekte çoklu değer bulunmamaktadır.

7. Tüm üçlü ilişkiler için bir tablo yaratılarak ilişkiye katılan tüm varlıkların birincil anahtarları bu tabloya yabancı anahtar olarak eklenir ve bunların birleşimi tablonun birincil anahtarı olur. Varsa ilişkinin öznitelikleri de bu tabloya eklenir. Kullanılan örnekte üçlü ilişki bulunmamaktadır.

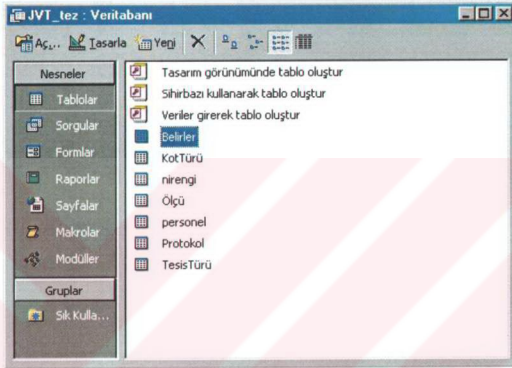
Oluşturulan bu tablolara ait başvurusal bütünlük koşullarını (referential integrity constraints) da içeren bir ilişkisel veritabanı şeması Şekil 6.2’de görülmektedir.



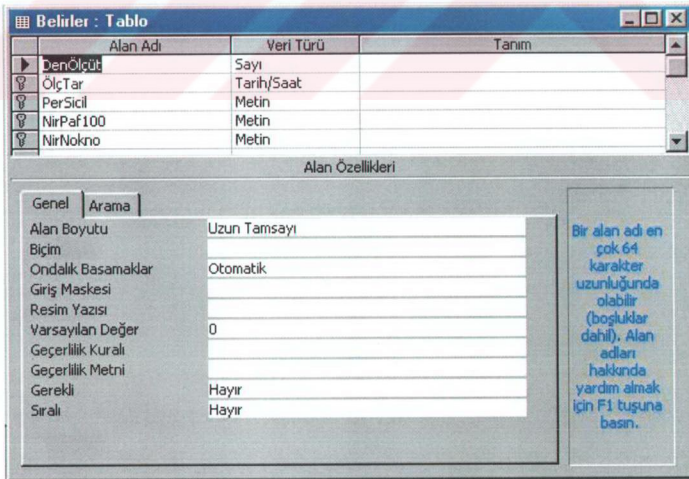
Şekil 6.2 Bütünlük Koşullarını İçeren Bir İlişkisel Veritabanı Şeması

6.5 Aşama 5: Fiziksel Veritabanı Tasarımı

Bölüm 6.4'te oluşturulan kavramsal şema ve tablolar bu aşamada MS ACCESS2000 yazılımı kullanılarak fiziksel ortamda hazırlanmıştır. Bunlar Şekil 6.3'te görülmektedir. Sık kullanılacağı yapılan anketlerle belirlenen sorgulamalara yanıtlama süresini azaltmak için "nirengi" tablosundaki paf250, paf100, paf50, paf25 öznitelikleri indekslenmiştir.



Şekil 6.3a Tablolar



Şekil 6.3b Belirler Tablosu Tasarım Görünümü

KotTürü : Tablo

	Alan Adı	Veri Türü	Tanım
?	kod	Otomatik Sayı	
	tür	Metin	

Alan Özellikleri

Genel | Arama

Alan Boyutu: Uzun Tamsayı
Yeni Değerler: Artan
Biçim:
Resim Yazısı:
Sıralı: Evet (Yineleme Yok)

Alan yerleşimini gösterir. Önceden tanımlanmış bir biçim seçin veya özel bir biçim girin. Biçimler hakkında yardım almak için F1 tuşuna basın.

Şekil 6.3c Kot Türü Tablosu Tasarım Görünümü

personel : Tablo

	Alan Adı	Veri Türü	Tanım
?	Sicil	Metin	
	adı	Metin	
	soyadı	Metin	

Alan Özellikleri

Genel | Arama

Alan Boyutu: 20
Biçim:
Giriş Maskesi:
Resim Yazısı:
Varsayılan Değer:
Geçerlilik Kuralı:
Geçerlilik Metni:
Gerekli: Evet
Sıfır Uzunluk İzni: Hayır
Sıralı: Evet (Yineleme Yok)
Unicode Sıkıştırma: Evet

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna

Şekil 6.3d Personel Tablosu Tasarım Görünümü

nirengi : Tablo

Alan Adı	Veri Türü	Tanım
hol_no	Metin	
nok_adi	Metin	
paf250	Metin	
paf100	Metin	
paf50	Metin	
paf25	Metin	
drc	Metin	
dom	Sayı	
sağa	Sayı	
yukarı	Sayı	
eder	Sayı	
edak	Sayı	
esan	Sayı	
bder	Sayı	
bdak	Sayı	
bsan	Sayı	
yük	Sayı	
tesyl	Sayı	
teskod	Sayı	
kotkod	Sayı	

Alan Özellikleri

Genel	Arama
Alan Boyutu	9
Biçim	
Giriş Maskesi	
Resim Yazısı	
Varsayılan Değer	
Geçerlilik Kuralı	
Geçerlilik Metni	
Gerekli	Hayır
Sıfır Uzunluk İzni	Hayır
Sıralı	Hayır
Unicode Sıkıştırma	Hayır

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1.

Şekil 6.3e Nirengi Tablosu Tasarım Görünümü

TesisTürü : Tablo

Alan Adı	Veri Türü	Tanım
kod	Otomatik Sayı	
tür	Metin	

Alan Özellikleri

Genel	Arama
Alan Boyutu	Uzun Tamsayı
Yeni Değerler	Artan
Biçim	
Resim Yazısı	
Sıralı	Evet (Yineleme Yok)

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.

Şekil 6.3f Tesis Türü Tablosu Tasarım Görünümü

Ölçü : Tablo

Alan Adı	Veri Türü	Tanım
YDürNok	Metin	
YBakNok1	Metin	
YBakNok2	Metin	
AraAçı	Sayı	
DDürNok	Metin	
DBakNok	Metin	
Zenit	Sayı	
KDürNok	Metin	
KBakNok	Metin	
Mesafe	Sayı	
PerSicil	Metin	
Tarih	Tarih/Saat	

Alan Özellikleri

Genel | Arama

Alan Boyutu: 50

Biçim:

Giriş Maskesi:

Resim Yazısı:

Varsayılan Değer:

Geçerlilik Kuralı:

Geçerlilik Metni:

Gerekli: Hayır

Sıfır Uzunluk İzni: Hayır

Sıralı: Hayır

Unicode Sıkıştırma: Evet

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.

Şekil 6.3g Ölçü Tablosu Tasarım Görünümü

Protokol : Tablo

Alan Adı	Veri Türü	Tanım
tarif	Metin	
kroki	OLE Nesnesi	
kabadeğer	Sayı	
NirPaf100	Metin	
NirNokno	Metin	
PerSicil	Metin	

Alan Özellikleri

Genel | Arama

Alan Boyutu: 250

Biçim:

Giriş Maskesi:

Resim Yazısı:

Varsayılan Değer:

Geçerlilik Kuralı:

Geçerlilik Metni:

Gerekli: Hayır

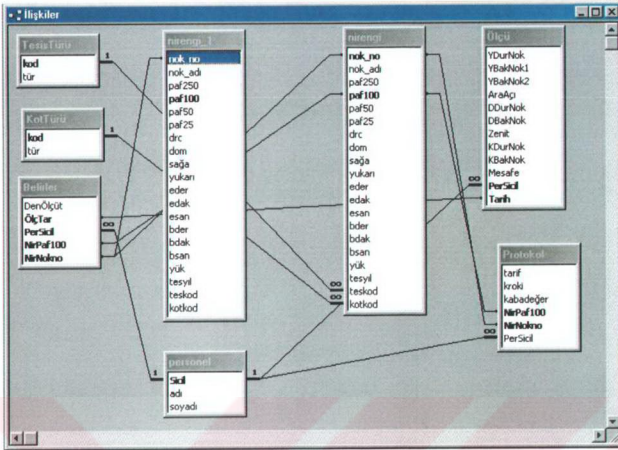
Sıfır Uzunluk İzni: Hayır

Sıralı: Hayır

Unicode Sıkıştırma: Evet

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.

Şekil 6.3h Protokol Tablosu Tasarım Görünümü



Şekil 6.3i Fiziksel Tasarım

6.6 Aşama 6: Veritabanı Sisteminin Uygulanması

Tüm aşamalar göz önünde bulundurularak aşağıda kullanıcı arayüzleri şekillerle anlatılan yazılım Delphi derleyicisi kullanılarak hazırlanmıştır.

Yandaki iletişim penceresi ile kullanıcılar bir pafta tanımlayarak onun içine giren noktaların listesini ya da protokollerini görebilmektedirler. Ayrıca pencerenin alt kısmında yer alan “Ayrıntılar” bölümü yardımıyla da sorgulamayı göre daraltabilmekte ve listedeki koordinatları 3^0 ya da 6^0 olarak hazırlatabilmektedirler.

kısımında yer alan “Ayrıntılar” bölümü yardımıyla da sorgulamayı tesis türü, kot verilme türü, nokta derecesine göre daraltabilmekte ve listedeki koordinatları 3⁰ yada 6⁰ olarak hazırlatabilmektedirler.

Koordinat Tanımlayınız...

Hazırlayan: Tülay GÜVENER

Kontrol Eden: Abdül TOK

Coğrafi: UTM

Enlem: 37° 47' 17"

Boylam: 36° 53' 41"

Noktasına 5000 metre uzaklıktaki noktalar

Kapat Liste Protokol Ayrıntılar <<

Alınacak noktalar

1 / 25 000 1 / 5 000 Tümü

Kot verilme türü

Trigonometrik Nivelman Tümü

Tesis Türü

Beton Blok Pilye Bina Üstü

Kayada Bronz Minare

Nokta Derecesi Dilim Genişliği

I. II. III. IV. 3° 6° Coğ.

Koordinat Tanımlayınız...

Hazırlayan: Tülay GÜVENER

Kontrol Eden: Abdül TOK

Coğrafi: UTM

Sağa: 242588 Dilim Genişliği

Yukarı: 4311683 6°

Dilim Numarası Dilim Orta Meridyeni

39

Noktasına 3000 metre uzaklıktaki noktalar

Kapat Liste Protokol Ayrıntılar <<

Alınacak noktalar

1 / 25 000 1 / 5 000 Tümü

Kot verilme türü

Trigonometrik Nivelman Tümü

Tesis Türü

Beton Blok Pilye Bina Üstü

Kayada Bronz Minare

Nokta Derecesi Dilim Genişliği

I. II. III. IV. 3° 6° Coğ.

Yukarıdaki iletişim pencereleri ile kullanıcılar coğrafi yada UTM olarak bir noktanın koordinatlarını ve metre olarak da ona olan mesafeyi girmek suretiyle bir alan tanımlayarak içine giren noktaların listesini yada protokollerini görebilmektedirler. Ayrıca pencerenin alt kısmında yer alan “Ayrıntılar” bölümü yardımıyla da sorgulamayı tesis türü, kot verilme türü, nokta derecesine göre daraltabilmekte ve listedeki koordinatları 3⁰ yada 6⁰ olarak hazırlatabilmektedirler.

7. SONUÇ

Doğru kararlar almaktaki en büyük etmen olan doğru bilgiye ulaşımın hızlanması için verilerin iyi organize edilmesi gerektiği, bunun içinde çağımızın teknolojik harikası olan bilgisayarın kullanılması zorunluluğu insanlığın önüne günden güne daha açık bir şekilde çıkmıştır. Dünyadaki büyük kuruluşların sayısı arttıkça bilgiye giden yolun ilk ayağı olan verinin iyi organize edilebilmesi için bilimsel bir perspektifin gerekliliğinin anlaşılması 1950'li yıllara uzanmaktadır. O günlerden günümüze gelişen bu yordambilim (teknoloji) büyük ve karmaşık yapıları kuruluşlardaki büyük hacimli verilerin depolanmasının, akışının, güncellenmesinin ve hızlı bir şekilde karara etki edecek olan bilgiye dönüşmesinin otomasyonu ile ilgilenmiştir.

Bu bağlamda ortaya çıkan bilgi sistem kavramının anlamının anlaşılabilmesi, veri yönetiminin, bilgi kaynağı idaresinin anlaşılmasından geçmektedir. Bu çalışmada, bu amaçla geliştirilmiş olan birkaç yöntembilimden biri olan ve genellikle veri tipleri ve sistem içindeki mantıksal akışından yararlanarak sistemin alt birimlere ayrılması esasına dayanan, sistemin mantıksal olarak organizasyonunu; veri akışının mantığı ile sistemi oluşturan alt birimler arasındaki işlevsel ilişkiye bağlı olarak gerçekleştiren veri akışı – güdümlü yöntembilimlerin yaklaşımı esas alınmıştır.

Tüm bunları ortaya koymuş ve bilgi sistemini çözümleme kararı almış bir kuruluş için yapılması gerekenlerden ilki bilgi sistem (makro) hayat çarkına bakmaktır:

1. Olurluk çözümlemesi (analizi)
2. Gereksinimlerin belirlenmesi ve çözümlenmesi
3. Tasarım
4. Uygulama
5. Değerlendirme ve kabul testi
6. Kullanma

Daha sonrada veritabanı uygulama sistemi (mikro) hayat çarkı irdelenmelidir:

1. Sistem tanımlaması: Veritabanı sisteminin kapsamı, kullanıcıları ve onun

uygulamaları tanımlanır.

2. Tasarım: Bu aşamanın sonunda, seçilmiş veritabanı yönetim sistemi üzerinde tüm mantıksal ve fiziksel tasarım hazır hale gelmiş olur.
 - a. Gereksinimlerin derlenmesi ve çözümlenmesi: Bu aşamada son kullanıcılar arasından seçilen personel ile yapılan görüşme ve anketlere dayanarak Veri Akış Çizgeleri (VİÇ) hazırlanır.
 - b. Kavramsal veritabanı tasarımı: Bu aşamada Varlık – İlişki (Vİ) çizgeleri hazırlanır.
 - c. Veritabanı yönetim sisteminin seçimi: Tanımlı ölçütlere göre bir değerlendirilme yapılarak en uygun veritabanı yönetim sistemi seçilir.
 - d. Veri modeli dönüşümü (mantıksal veritabanı tasarımı olarak da adlandırılır.): Vİ çizgeleri veritabanı tablolarına dönüştürülür.
 - e. Fiziksel veritabanı tasarımı: Veritabanı dosyalarındaki depolama yapıları ve erişim yollarının seçildiği aşamadır.
 3. Uygulama: Bu aşama kavramsal, dış ve iç veritabanı tanımlarının yazılması, boş veritabanı dosyalarının yaratılması ve yazılımların uygulanmasını içerir.
 4. Yükleme veya veri dönüşümü: Veritabanı, ya verilerin doğrudan yüklenmesi yada hazır dosyaların veritabanı formatına dönüştürülmesi ile doldurulur.
 5. Uygulama dönüşümü: Önceki sistemden kalma tüm yazılım uygulamaları yeni sisteme dönüştürülür.
 6. Test ve değerlendirme: Yeni sistem denenir ve değerlendirilir.
 7. Kullanım: Veritabanı sistemi ve uygulamaları kullanıma konur.
 8. İzleme ve bakım: Kullanım aşaması boyunca sistem sürekli izlenir ve bakımı yapılır. Hem veri içeriğinde hem de yazılım uygulamalarında büyüme ve genişleme olabilir. Zaman zaman büyük değişiklikler ve yeniden yapılanmalar gerekebilir.
- 2, 3 ve 4 etkinlikleri daha büyük olan bilgi sistemin tasarım ve uygulama aşamalarının parçasıdır.

Bu çalışmada bu aşamalar JBS için ele alınıp onun önemli bir parçasını oluşturan nirengi ile ilgili bölüm için uygulaması yapılmıştır. Bu uygulama sonucunda şunlar görülmüştür:

1. Çözümlemenin başında kapsamın çok geniş tutulması ve bunun için yeterli personel ve zamanın iyi planlanmaması, sistem çözümü uygulama uygulamasının sonuçlandırılmama riskini arttırmaktadır. Bunu engellemek için sistem mutlaka alt sistemlere ayrılmalı ve öncelikle bu alt sistemlerdeki hedefler konularak bunlara ulaşılmaya çalışılmalıdır. Daha sonra tüm ara hedeflere ulaşırsa ana hedefe yönelmelidir.
2. Ulaşılmış bir ara hedef olan nirengi verilerinin bir VTS içerisinde tutularak kullanıcılara sunulması sırasında şu zorluklarla karşılaşmıştır:
 - a. Sistem çözümlemesinin doğasında olan geri beslemeler (örneğin mantıksal veritabanı tasarımı aşamasında yeni gereksinimlerin ortaya çıkması) fazlalaştığında geriye dönmekten ilerleme sağlanamamakta bu da çözümçüler arasında bıkkınlık ve yılgınlığa neden olmaktadır. Bunu önlemek için baştan sistemin sınırlarının ve ara hedeflere ulaşmadaki zamanların iyi belirlenmesi gerekmektedir.
 - b. Son kullanıcılarla çözümçüler çok farklı alanlardan geldiğinde bir iletişim kopukluğu ortaya çıkmakta, bu da mevcut sorunların teşhisinde yanlışlıklara neden olmaktadır. Bunu önlemek için bu iki grubun mümkün olduğunca birbirlerini anlayabilecek insanlardan seçilmesi gerekmektedir.
 - c. Uygulamanın ömrünün uzaması başta kullanılmasına karar verilmiş donanım ve yazılımların demode olmasına ve son gereksinimleri karşılamakta yetersiz kalmasına sebep olmaktadır. Bunu için daha önce de değinildiği gibi uzun zaman alacak mükemmel bir sistem çözümlemesine ulaşmak yerine kısa zamanda ulaşılacak ara hedefler belirlenerek ulaşılan her hedefin kullanıma sunulmasının daha uygun olduğu değerlendirilmektedir.
3. Tüm bunların yanı sıra uygulamanın sonuçlarında şu yararlar sağlanmıştır:
 - a. Kullanıcılar en son bilgileri içeren büyük hacimli verilere daha hızlı ve kolay bir şekilde ulaşabilmişlerdir.
 - b. Veriler arasındaki karmaşık ilişkilerin modellenmesi ve verilerin ortak bir alanda korunması ile veri güncellemesi kolaylaşmıştır.
 - c. Çeşitli bilgi kaynakları birleştirilerek karmaşık kararlar almak daha kolay ve hızlı olarak yapılabilmeye başlanmıştır.

- d. Uygulamalar arası veri bağımsızlığı sağlandığından verilerin bozulması ve kaybolması riski ortadan kaldırılmıştır.
- e. Veriler arası ilişkiler kurulup bunlardan görünümüler oluşturulduğundan disk hacminden büyük tasarruf sağlanmıştır.
- f. Yeni uygulama geliştirimi çok hızlı ve ucuza yapılabilir hale gelmiştir.

Sonuç olarak her sistemin bir gün bilgi sisteme geçişi kaçınılmazdır. Bunun da bir yöntembilim içerisinde yapılması zorunludur.



KAYNAKLAR

- [1] **Fry, J. and Sibley, E.**, 1976. Evolution of Database Management Systems, Association for Computing Machinery Computing Surveys, **Vol.8: No.1**.
- [2] **Date, C.**, 1990. An Introduction to Database Systems, Volume 1, Fifth Edition, Addison – Wesley.
- [3] **Korth, H. and Silberschartz, A.**, 1991. Database System Concepts, Second Edition, McGraw – Hill Inc.
- [4] **Navathe, S. and Kerschberg, L.**, 1986. Role of Data Dictionaries in Database Design, Information and Management, **Vol.10: No.1**.
- [5] **Elmasri, R., Larson, J. and Navathe, S.**, 1986. Schema Integration Algorithms for Federated Databases and Logical Database Design, Honeywell CSDD, Technical Report CSC-86-9: 8212.
- [6] **Teorey, T.**, 1990. Database Modeling and Design: The Entity – Relationship Approach, Morgan Kaufmann.
- [7] **Ceri, S.**, 1983. Methodology and Tools for Database Design, North – Holland.
- [8] **Gane, C. and Sarson, T.**, 1977. Structured System Analysis: Tools and Techniques, Improved System Technologies Inc.
- [9] **Pressman, Roger S.**, 1992. Software Engineering: A Practitioners Approach, McGraw – Hill Inc.
- [10] **Stevens, W. P., Myers, G. J. and Constantine L. L.**, 1974. Structured Design, IBM Systems Journal, **Vol.13: No.2**, pp. 115 – 139.
- [11] **Eick, C. and Lockemann, P.**, 1985. Acquisition of Terminological Knowledge Using Database Design Techniques, SIGMOD.

- [12] **Bruyn, W. et al.**, 1988. ESML: An Extended Systems Modeling Language Based on the Data Flow Diagram, Association for Computing Machinery Software Engineering Notes, **Vol.13: No.1**, pp. 58 – 67.
- [13] **McFadden F. and Hoffer, J.**, 1988. Database Management, Second Edition, Benjamin / Cummings Publishing Co. Inc.
- [14] **Abrial, J.**, 1974. Data Semantics, North – Holland.
- [15] **Verheijen, G. and VanBekkum, J.**, 1982. NIAM: An Information Analysis Method, North – Holland.
- [16] **Codd, E.**, 1974. Extending the Database Relational Model to Capture More Meaning, Association for Computing Machinery Transaction on Database Systems (journal), **Vol.2: No.4**.
- [17] **Chen, P.**, 1976. The Entity – Relationship Model – Toward a Unified View of Data, Association for Computing Machinery Transaction on Database Systems (journal), **Vol.1: No.1**.
- [18] **ER Conference**, 1992. Proceedings of the Eleventh International Conference on Entity – Relationship Approach, Pernul, G. and Tjoa, A. (editors), Karlsruhe, Germany.
- [19] **Batini, C., Lenzerini, M. and Navathe S.**, 1987. A Comparative Analysis of Methodologies for Database Schema Integration, Association for Computing Machinery Computing Surveys, **Vol.18: No.4**.
- [20] **Yao, S.**, 1985. Principles of Database Design, Volume 1: Logical Organizations, Prentice – Hall.
- [21] **Navathe, S., Elmasri, R. and Larson, J.**, 1986. Integrating User Views in Database Design, Computer Magazine (journal) of Institute of Electrical and Electronics Engineers (IEEE), **Vol.19: No.1**.
- [22] **Casanova, M. and Vidal, V.**, 1982. Toward a Sound View Integration Method, Proceedings of the Association for Computing Machinery Symposium on Principles of Database Systems.
- [23] **Teorey, T. and Fry, J.**, 1982. Design of Database Structures, Prentice – Hall.

- [24] **Mylopoulos, J., Bernstein, P. and Wong, H.**, 1980. A Language Facility for Designing Database Intensive Applications, Association for Computing Machinery Transaction on Database Systems (journal), **Vol.5: No.2.**
- [25] **Albano, A., Cardelli, L. and Orsini, R.**, 1985 GALILEO: A Strongly – Typed Interactive Conceptual Language, Association for Computing Machinery Transaction on Database Systems (journal), **Vol.10: No.2.**
- [26] **Elmasri, R., and Wiederhold, G.**, 1981, GORDAS: A Formal, High – Level Query Language for the Entity – Relationship Model, ER Conference 1981, Washington DC.
- [27] **Elmasri, R. and Navathe, S.**, 1995. Fundamentals of Database Systems, Second Edition, Benjamin / Cummings Publishing Co. Inc.
- [28] **Ata, M.**, Karşılıklı görüşme.
- [29] **Kalkar, M.**, Karşılıklı görüşme.



EKLER

EK-A KARNELER

EK-A1 DİSTOMAT (KENAR) ÖLÇÜ KARNESİ

DISTOMAT UZUNLUK ÖLÇÜSÜ VE İNDİRGEME HESABI

GÖZLEMCI :	TARİH :	I		
		II		
İSTASYON :	HEDEF :	I+II=T		
		400-T=K		
ALET YÜK :	İŞARET YÜK:	I		
		1/2K'		
		I+1/2K=Z ₁		
ÖLÇÜLER		I		
t = °C	S ₁ = , m	II		
p = mb	S ₂ = , m	I+II=T		
ppm =	S ₃ = , m	400-T=K		
	S _{ort} = , m	I		
		1/2K		
		I+1/2K=Z ₂		
		Z = (Z ₁ +Z ₂)/2		
FORMÜLLER	DEĞERLER	AÇIKLAMA		
S _{ort}		Meteorolojik düzeltme almış uzunluk (Birimi = Metre)		
$\Delta H = H_a - H_b$ *		Alet ve hedef yüksekliği farkı (Birimi = Metre)		
$H_m = \frac{H_a + H_b}{2}$		Alet ve hedef yükseklikleri ortalaması (Birimi = Metre)		
R		Yeryuvarlağı eğrilik Yarıçapı (Birimi = Metre)		
$K_1 = \frac{(\Delta H)^2 + \Delta H^4}{2S_{ort}^3 \theta S_{ort}^3}$		Yatay düzleme indirgeme düzeltmesi		
$K_2 = -\frac{H_m}{R} S_{ort}$		Deniz Yüzeyine (veya elipsoide) indirgeme düzeltmesi		
$K_3 = \frac{S_{ort}^3}{24 R^2}$		Deniz Yüzeyindeki (veya elipsoiddeki) Kirişten yay uzunluğuna indirgeme düzeltmesi		
[K]		[K] = K ₁ + K ₂ + K ₃		
S _o = S _{ort} + K		Deniz (veya elipsoid) Yüzeyindeki yay uzunluğu		
$ds = \frac{1}{6R^2} (y_1^2 + y_1y_2 + y_2^2) S_o$		Gauss - Kruger Yüzeyine indirgeme düzeltmesi		
S = S _o + ds		Gauss - Kruger Yüzeyindeki uzunluk		

★ Denizden olan Yükseklik alınırsa, deniz seviyesine ;
elipsoidten olan yükseklik alınırsa, elipsoid yüzeyine indirgeme yapılmış olur.

NOT : Bu indirgemeler geçicidir.

EK-A2 ZENİT (DÜŞEY AÇI) ÖLÇÜ KARNESİ

Zenit ölçüleli Karnesi (Zelss)											
İstasyon				İşaret				Resatçı			
Teodolit No.				Alet				Gün ve Hava			
Noktalar											
I											
II											
I + II = Toplam											
400 - Toplam = k											
I											
1/2 K											
I + 1/2 K = Z ₁											
I											
II											
I + II = Toplam											
400 - Toplam = k											
I											
1/2 K											
I + 1/2 K = Z ₂											
Z ₁ + Z ₂ Ortalama											
g											

EK-A3 YATAY AÇI ÖLÇÜ KARNESİ - 1

İstasyon No. :	Adı :	Yatay Açı Ölçüleri	Alet yüksekliği : İşaret :
Rasit : Yazım :	Gün : .. / .. / 98 Saat : ..	Eskilen Hedef	İşaret Helyotrof Pirüdak
		Hava:	Yakın Kırsal Bölge Yatış
		Rüzgarlı	Hafif orta güddeli

Eskilen Noktaların		Sil. No.	D. V.	I				I + 2 Ortalama	I ve II Ortalama				Mebdelerin altına çevrilmesi			
No.	Adı			g	e	ce	cc		g	e	ce	g	e	ce		
			I													
			II													
			I													
			II													
			I													
			II													
			I													
			II													
			I													
			II													
			I													
			II													
			I													
			II													
			I													
			II													
			I													
			II													
			I													
			II													
			I													
			II													

Herşeyi :
İmza :
Oran No. : ..

Stok No : 1830 - 3/2

[illegible]

DEĞER CEDVELİ
Paftası : **BANDIRMA-618-C2**

Dilim orta meridyeni = $\lambda_0 = 27^\circ$
 Dilim genişliği 6°
 Küçültme faktörü 0.9996

[illegible]

Yazıldığı gün : 24.09.2001
Yazarın adı : Ayken Gökçek
İmzası : 

Kontrol eden : *Abdullah Fatah*
 Ingeasi : *29*

73

EK-A6 DEĞER CETVELİ – 2

DEĞER CETNELİ

PAFTASI : ELBİSTAN K38-a1
DİLİM ORTA MERİDYENİ : 39°
DİLİM GENİŞLİĞİ : 3°
KÜÇÜLTME FAKTÖRÜ : 1.0000

[illegible]

YAZILDIĞI TARİH : 24.09.2001
HAZIRLAYAN :Aytan Gökçe
İMZASI :

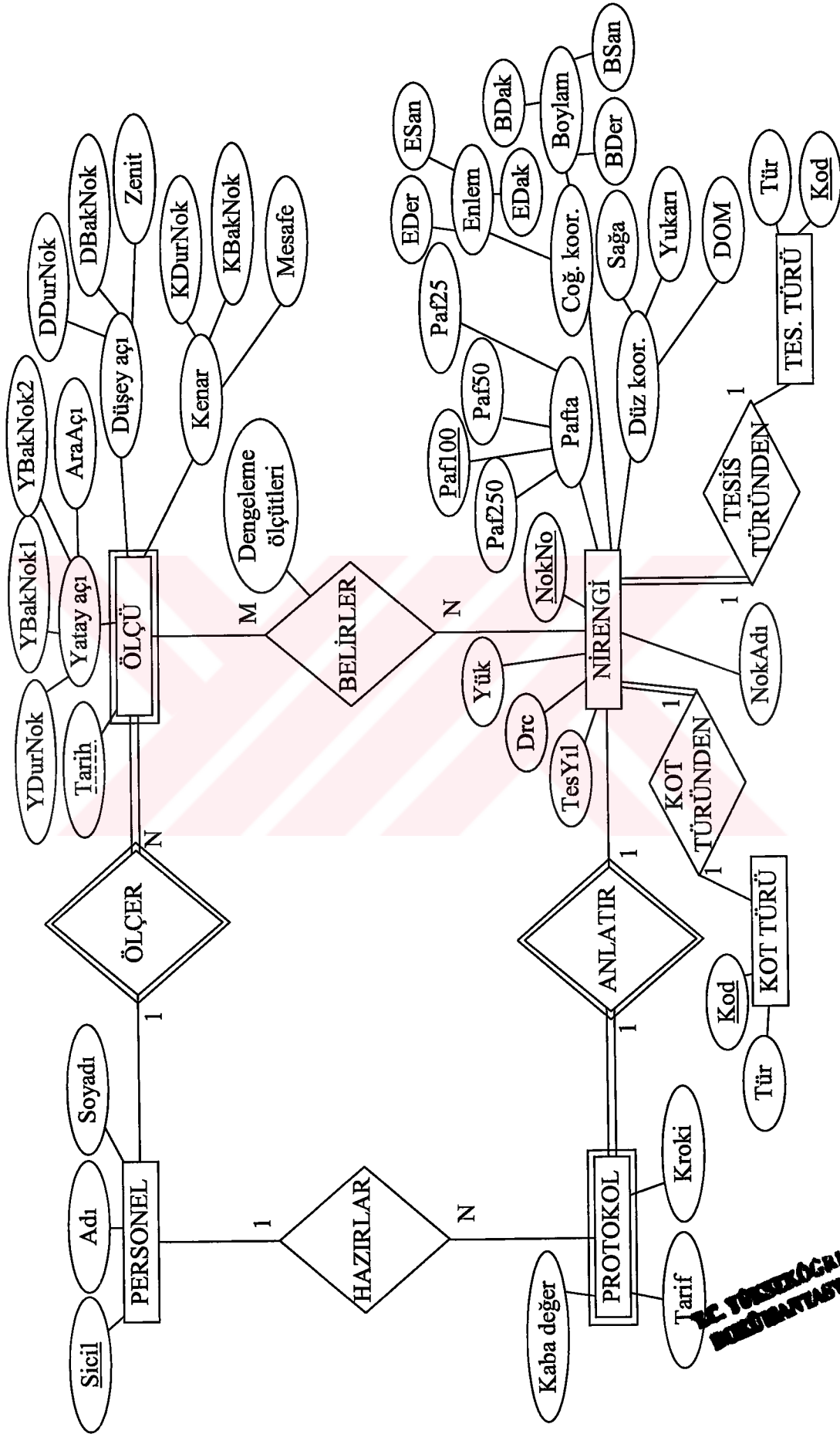
KONTROL EDEN :Ahmet TAN
İMZASI :

EK-A7 NİVELMAN (YÜKSEKLİK) ÖLÇÜSÜ KARNESİ

T.D. NİVELMAN ÖLÇÜ KARNESİ
VE
KART DELGİ FORMU

SAYFA NO :
SIRA NO :
KAYIT NO :

NOKTASINDAN					NOKTASINA					VARILAN NOKTANIN TARİFİ										GİDİŞ DÖNÜŞ		TARİH					İSİ		KOD															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
FORTE NO	BAKİŞ					SOL ORTA					SAĞ ORTA					GERİ - İLERİ					ST		SOL GERİ					SOL İLERİ					GERİ		İLERİ									
						1 2 3 4 5 6 7 8 9 10 11 12					1 2 3 4 5 6 7 8 9 10 11 12					{ + } { - }					ÜST KİL		13 14 15 16 17 18 19 20 21 22																					
	İLERİ					23 24 25 26 27 28 29 30 31 32 33 34															ALT KİL		35 36 37 38 39 40 41 42 43 44																					
	GERİ					1 2 3 4 5 6 7 8 9 10 11 12															ÜST KİL		13 14 15 16 17 18 19 20 21 22																					
	İLERİ					23 24 25 26 27 28 29 30 31 32 33 34															ALT KİL		35 36 37 38 39 40 41 42 43 44																					
	GERİ					1 2 3 4 5 6 7 8 9 10 11 12															ÜST KİL		13 14 15 16 17 18 19 20 21 22																					
	İLERİ					23 24 25 26 27 28 29 30 31 32 33 34															ALT KİL		35 36 37 38 39 40 41 42 43 44																					
	GERİ					1 2 3 4 5 6 7 8 9 10 11 12															ÜST KİL		13 14 15 16 17 18 19 20 21 22																					
	İLERİ					23 24 25 26 27 28 29 30 31 32 33 34															ALT KİL		35 36 37 38 39 40 41 42 43 44																					
	GERİ					1 2 3 4 5 6 7 8 9 10 11 12															ÜST KİL		13 14 15 16 17 18 19 20 21 22																					
	İLERİ					23 24 25 26 27 28 29 30 31 32 33 34															ALT KİL		35 36 37 38 39 40 41 42 43 44																					
	GERİ					1 2 3 4 5 6 7 8 9 10 11 12															ÜST KİL		13 14 15 16 17 18 19 20 21 22																					
	İLERİ					23 24 25 26 27 28 29 30 31 32 33 34															ALT KİL		35 36 37 38 39 40 41 42 43 44																					
	GERİ					1 2 3 4 5 6 7 8 9 10 11 12															ÜST KİL		13 14 15 16 17 18 19 20 21 22																					
	İLERİ					23 24 25 26 27 28 29 30 31 32 33 34															ALT KİL		35 36 37 38 39 40 41 42 43 44																					
	GERİ					1 2 3 4 5 6 7 8 9 10 11 12															ÜST KİL		13 14 15 16 17 18 19 20 21 22																					
	İLERİ					23 24 25 26 27 28 29 30 31 32 33 34															ALT KİL		35 36 37 38 39 40 41 42 43 44																					
																																			</									



ÖZGEÇMİŞ

Yalkın ÇAĞLAR, 1970 yılında Urla / İZMİR’de doğdu. İlk öğrenimini İzmir Balçova İlkokulu’nda 1981 yılında, orta öğrenimini de İzmir Güzelyalı Ortaokulu’nda 1984 yılında tamamladı. Aynı yıl Maltepe Askeri Lisesi’nde öğrenimine başladı ve 1988 yılında buradan mezun olarak Kara Harp Okulu’nda öğrenimine devam etti. 1992 yılında Kara Harp Okulu’ndan Harita Subayı olarak mezun olarak M.S.B. Harita Yüksek Teknik Okulu’nda iki yıl süreyle mühendislik eğitimi gördü.

1994 yılında Harita Yüksek Teknik Okulu’ndan Harita Mühendisi diploması alarak aynı yıl Harita Genel Komutanlığı, Jeodezi Dairesi’nde göreve başladı ve halen aynı dairede Jeodezik ve Jeofizik Değerler Kısım Amiri olarak görevine devam etmektedir.