

T.C.
İSTANBUL ÜNİVERSİTESİ
EDEBİYAT FAKÜLTESİ
KÜTÜPHANECİLİK BÖLÜMÜ
DOKÜMANTASYON VE ENFORMASYON ANABİLİMPALİ

VERİ TABANI YAKLAŞIMI
VE KÜTÜPHANELERDE UYGULANMASI :

BİR ÇÖZÜM ÖNERİSİ

LİSANS TEZİ

AYKUT ARIKAN
9008 DE 005

İSTANBUL, 1994





İÇİNDEKİLER

Sayfa

ÇİZİMLER	1v
ÖZ/ABSTRACT/ABSTRAKT	v
ÖNSÖZ	vi
GİRİŞ	1
1 VERİ TABANI YAKLAŞIMI	4
1.1 VERİ TABANI KAVRAMI	4
1.2 VERİ	5
1.2.1 BİLGİ	8
1.3 VERİ TABANI	9
1.3.1 VERİ TABANI YAKLAŞIMININ AMAC VE KAPSAMI	14
1.3.2 VERİ TABANI GELİŞTİRME SÜRECİ	17
1.3.2.1 VERİ ARAŞTIRMASI	19
1.3.2.2 VERİ ÖRNEKLEMESİ	21
1.3.2.2.1 VERİ ÖRNEĞİNİN ÖGELERİ	24
1.3.2.2.1.1 İLİŞKİLER	24
1.3.2.2.1.2 VARLIKLAR VE YÜKLEMLER	26
1.3.2.2.1.3 YÖRELER VE TANIMLAYICILAR	27
1.3.2.2.2 VERİ ÖRNEĞİNİN SAPTANMASININ AŞAMALARI	27
1.3.2.3 VERİ TABANI TASARIMI	28
1.3.3.4 VERİ TABANI İŞLETİMİ	28
1.3.3.5 VERİ TABANI AYARLAMASI	29
1.3.3 VERİ TABANI TÜRLERİ	29
1.3.3.1 HİYERARŞİK VERİ TABANLARI	31
1.3.3.2 AĞ YAPILI VERİ TABANLARI	31

Gifted By

Mesul Yayıncı

1.3.3.3	İLİŞKİSEL VERİ TABANLARI	32
1.3.3.4	NESNE YÖNELİMLİ VERİ TABANLARI	41
1.3.3.5	DİĞER	45
1.4	VERİ TABANI YÖNETİM SİSTEMLERİ	45
1.4.1	VERİ TABANI YÖNETİM SİSTEMLERİNİN İŞLEVLERİ	46
1.4.1.1	TASARIM	46
1.4.1.2	GÜNCELLESTİRME	47
1.4.1.3	SORGULAMA VE RAPORLAMA	47
1.4.1.4	DAĞITIK VERİ YÖNETİMİ	48
1.4.1.5	DİĞER GENEL İŞLEVLERİ	49
1.5	VERİ TABANI UYGULAMA YAZILIMLARI	50
1.5.1	VERİ TABANI UYGULAMA DİLLERİ	53
1.5.1.1	XBASE/CLIPPER AİLESİ	53
1.5.5.2	SQL	54
1.5.5.3	OBJECTPAL	55
1.5.2	UYGULAMA YAZILIMLARININ GELİŞTİRİLMESİNDE DİKKAT EDİLMESİ GEREKENLER	55
1.5.2.1	DİL	55
1.5.2.2	KULLANICI DOSTLUĞU	56
1.6	VERİ TABANI YAKLAŞIMININ GERİRDİKLERİ	58
2	VERİ TABANI YAKLAŞIMININ KUTUPHANELERDE UYGULANMASI VE BİR ÇÖZÜM ÖNERİSİ	62
2.1	VERİ TABANI YAKLAŞIMININ KUTUPHANELERE GETİRDİĞİ YENİLİKLER	62
2.1.1	TOMLESİK YAPI	62

2.1.2	STANDARDIZASYON VE İŞBİRLİĞİ OLANAKLARI VE MARC YAKLAŞIMININ DEĞİŞİMİ	63
2.1.3	KÜTÜPHANECİNİN DEĞİŞEN KONUMU	66
2.2	KÜTÜPHANELERDE VERİ TABANI YAKLAŞIMINA DAYALI UYGULAMALARI İÇİN PARADOX FOR WINDOWS VERİ TABANI YÖNETİM SİSTEMİ ALTINDA GELİSTİRİLEBİLECEK BİR ÇÖZÜM ÖNERİSİ	67
2.2.1	PARADOX FOR WINDOWS'A DAYALI ÇÖZÜMLERİN UYGULANABİLİRLİĞİ	67
2.2.1.1	VERİ TABANI YAPISI VE TASARIM	67
2.2.1.2	İŞLETİM VE DENETİM	69
2.2.1.3	SORGULAMA VE RAPORLAMA	70
2.2.2	ÇÖZÜM ÖNERİSİ : B.K.V.	71
8	SONUÇ	83
	KAYNAKLAR	87

ÇİZİMLER

SAYFA

ÇİZİM 1	ANSI/SPARC MODELİ	16
ÇİZİM 2	İLİŞKİSEL VERİ TABANI ÖRNEĞİ	37
ÇİZİM 3	SEÇME İŞLEMİ	38
ÇİZİM 4	İZDÜŞÜM İŞLEMİ	39
ÇİZİM 5	BİRLEŞTİRME İŞLEMİ	39
ÇİZİM 6	KESİŞİM İŞLEMİ	40
ÇİZİM 7	AYRIM İŞLEMİ	40
ÇİZİM 8	İLİŞKİSEL VERİ TABANI ÖRNEĞİ	42
ÇİZİM 9	NESNE YÖNELİMLİ VERİ TABANI ÖRNEĞİ	43
ÇİZİM 10	SAUER ŞEMASI	58
ÇİZİM 11	GENİŞLETİLMİŞ SAUER ŞEMASI	60

öz

Kuramsal temel eksikliği nedeniyle, sorunlar yaşanan veri tabanı uygulamalarına, kuramsal destek sağlanması amaçlanıyor. Veri tabanı yaklaşımını ortaya çıkaran, ve konuyla ilgili diğer kavramlar çözümlenerek, yaklaşımın sağladığı olanaklar saptanıyor. Kütüphanelere veri tabanına dayalı uygulamalar tartışılarak, Paradox for Windows ilişkisel veri tabanı yönetim sistemi için geliştirilen Bölünmüş Katalog Veri Tabanı Yapısı, B.K.V., adlı veri tabanına dayalı kütüphane kataloğu uygulaması, çözüm olarak öneriliyor.

ABSTRACT

It is intended to support database applications, where problems are faced, because of lack of a theoretical base, with theoretical aspects. Concepts that brought the database approach to light, and related concepts are analysed to indicate the opportunities, supplied by the approach. Database applications in libraries are discussed, and a database based library catalogue application, called B.K.V., developed for the relational database management system Paradox for Windows, is proposed as a solution.

ABSTRAKT

Es wird beabsichtigt, Datenbank Anwendungen, bei denen, wegen Mangel einer theoretischen Basis, Probleme auftauchen, aus theoretischen Aspekten zu unterstützen. Begriffe, die das Datenbank-Verfahren verursacht haben, und betreffende Begriffe, werden analysiert, um Möglichkeiten des Datenbank-Verfahrens darzustellen. Datenbank-Anwendungen in Bibliotheken werden diskutiert, und eine Datenbank basierte Anwendung für Bibliothekskataloge, namens B.K.V., die für das relationale Datenbank Verwaltungssystem Paradox for Windows entwickelt wurde, wird als Lösung vorgeschlagen.

ÖNSÖZ

Bu araştırma, Perran Özben Kütüphanecilik Ödülü yarışması birincisi "Veri tabanı yönetim sistemleri ve kütüphanelerde uygulanması : bir çözüm önerisi" adlı ataçtırmanın gözden geçiriliş, düzeltilmiş ve genişletilmiş halidir.

Ödül kazanan araştırmanın MARC yaklaşımını tartışma konusu yapması nedeniyle kimi çevrelerden oldukça olumsuz tepki almasına karşın, kimi çevrelerce de birincilik ödülüne uygun görülecek derecede olumlu karşılanması, bu konuda yaşanan boşluğun en somut göstergesidir.

Özellikle MARC yaklaşımını ele alan kaynaklardan sonradan derlediğim sonuçların tezimi desteklemesi ise sanırım herkese bu konudaki boşluğu doldurma fırsatını verecektir.

"Ortaya ortak bir bibliyografik veri tabanı yapısı konursa ulusal düzeyli bir MARC biçimine gerek kalmaz" dediğimde bunu ortaya koymam konusunda bana gerekli motivasyonu veren ve sözkonusu araştırma metnini 1994 Perran Özben Kütüphanecilik Ödülü yarışmasına göndermeme neden olan Prof. Dr. Meral ALPAY'a tesekkürü bir borç bilirim.

Beni tezim boyunca destekleyen danıřman hocam Do. Dr. Aysel Yontar'a bu desteęi iin teřekkr ederim.

Yrd. Do. Dr. Hasan S. Keseroęu'na zellikle B.V.K.'nin geiřtirilmesi konusundaki desteklerinden dolayı teřekkr ederim.

Ayrıca arařtırma evresinde beni yazılım, belge ve bilgi ynnden destekleyen Bimeks Bilgi iřlem ve Dıř Tic. Ltd. řti. ın genel mdr Do. Dr. Vedat AKGİRAY'a da bu iten desteęi iin teřekkr ederim.

Aykut Arıkan

31.5.1994

Bu araştırmanın amacı öncelikle Türkiye'de kuramsal temelsizlik ve bilgi eksikliği nedeniyle uygulamalarında yanlışlara ve aksaklıklara rastlanan veri tabanı yaklaşımı konusuna kuramsal açıdan genel bir temel sağlamak, uygulamalara yöntem, yaklaşım ve kapsam konusunda ışık tutarak onları desteklemektir.

Bu nedenle araştırmanın ilk bölümünde veri tabanı yaklaşımı tarihsel gelişimi içinde, kuram ve yöntem açısından bir bütün olarak irdelenmiştir. Bu bağlamda öncelikle veri tabanı terimi kavramsal açıdan tartışılmıştır. Buna bağlı olarak da veri tabanı yönetim sistemi ve veri tabanı uygulama yazılımı kavramlarına açıklık getirilmiştir.

Bu tartışma ve açıklamaların ışığında, veri tabanı yaklaşımının kapsam ve amaçları ile veri tabanı geliştirme süreci incelenmiştir. Veri tabanı kavramının zaman içinde geçirdiği yapısal değişimi yansıtan değişik veri tabanı türleri de buna bağlı olarak ele alınmıştır.

İlk bölümün sonunda veri tabanı yönetim sistemlerinin işlevleri ve veri tabanı uygulama yazılımları ayrıntılı olarak ele alınmışlardır.

Kütüphane ve .Bilgi bilimleriyle bu kadar yakından ilişkili olan bir konunun kütüphane alanındaki uygulamaları, getirdiği yenilikler, uygulamalardaki eksiklikler ve yapılması gerekenler ikinci bölümde, sözkonusu kuruluşlarda geliştirilecek uygulamalarda özellikle kataloglama alanında yaşanan karmaşaya çözüm getirebilmek amacıyla tasarlanmış, B.K.V. olarak da adlandırılan, somut bir çözüm önerisi çerçevesinde ele alınmıştır.

B.K.V.'nin seçeneksiz bir tekbiçim olmayıp, yapısal açıdan çözüm için bir öneri niteliğini taşımasının nedeni özellikle diğer çözüm yollarını, alışlagelmiş zihniyetin aksine, yadsımayarak ya da görmezlikten gelmeyerek, bilgi teknolojisi, bilgi hizmetleri, kütüphanecilik ve iletişim alanlarında yaşanan kimi sıkıntıların başlıca nedeni olan uyumsuzluk sorununu ortadan kaldırmaktır. B.K.V. ya da diğer geleneksel çözüm yollarından hangisinin daha etkin olduğunu zaman içinde geliştirilecek ve gerçekleştirilecek uygulamaların sonuçları belirleyecektir.

Veri tabanı yönetim sistemi olarak önerilen Paradox for Windows adlı ticari veri tabanı yönetim sistemi yazılımı, konuya yatkınlığı açısından tercih edilmiş, ve geliştirilen örnek uygulamalardan, tatmin edici sonuçlar alınmıştır. Ancak bunun ötesine geçip bu ürün için "en iyisi" demek ne olasıdır ne de bilimsel açıdan tutarlı bir yaklaşımdır.

Bu araştırma, . getirdiđi çözüm önerisi çerçevesindeki yeni bakış açısıyla Türkiye için yeni bir yaklaşımın başlangıcı olabilirse amacına ulaşmış sayılacaktır.

1 Veri Tabanı Yaklaşımı

1.1 Veri Tabanı Kavramı

Günümüzde, gerek kütüphanecilik, gerekse bilgisayar konulu yayınlarda, veri tabanı kavramı, çoğunlukla kendisinden başka bir kavram olan, veri tabanı yönetim sistemi kavramının yerine, sanki onun bir kısaltmasıymış gibi, yanlış olarak kullanılmaktadır. Oysa veri tabanı kavramı, aşağıda da ele alınacağı gibi, özellikle bilgisayar ortamındaki uygulamalarda, verilerin yüklü bulunduğu, örgütlü, teknik düzenegi (kütüphanecilik bakış açısıyla bir belgeyi) belirtirken, veri tabanı yönetim sistemi kavramı, bu düzenek ile işlem yapmaya, onu söz konusu örgütlenme içinde, yönetmeye yarayan bir tür uygulama yazılımını belirtmektedir.

İlk bakışta hiç önemsenmeyen başka bir hata ise, özellikle bilgisayar piyasasında, veri tabanı yönetim sistemi olarak, hiç bir ayırım yapılmadan, veri tabanı yönetim sistemleri pazarında "baskın bir eğilimi temsil ettiğinden"¹, değişik türlerdeki veri tabanı yönetim sistemlerinden sadece biri olan, ilişkisel veri tabanı yönetim sistemlerinin kast edilmesidir.

1 ilişkisel sistemler ve diğerleri. — s. 38-44 . — Bilgisayar . — (Temmuz 1993). — s. 38

Bu hata halkaları birbirlerine eklenip de ortaya çıkan zincirleme hata yukarki gibi çözümlendiğinde, konunun gerçek boyutları ortaya çıkmaktadır: günümüzde, veri tabanı kavramıyla aslında kast edilen, veri tabanı yönetim sistemlerinin bir türü olan, ilişkisel veri tabanı yönetim sistemleridir. Bu bakımdan veri tabanı kavramının ortaya çıkmasını sağlayan kavramlarla konuyla ilişkili diğer kavramların tek tek incelenerek, konunun çözümlemeli olarak ele alınması gereklidir.

1.2 Veri

Kavramın ingilizce karşılığı olan "date" [çoğ. data] Latince'deki "dare" [vermek] fiilinin "miş" yapılı ortacı olan "datum" sözcüğünden türetilmiştir²; sonuçta "verilmiş olan" gibi bir anlam çıkmaktadır. Dolayısıyla da, "veri" sözcüğünün kavramı tam olarak karşıladığı söylenebilir.

Türkçe Sözlük, kavramı "bir araştırmanın, bir tartışmanın, bir uslamamanın [zihinsel bir yargılamanın] temeli olan ana öge"³ olarak ele almaktadır.

2 American heritage dictionary. — 2nd college ed. — Boston : Houghton Mifflin Company, c. 1982. — s. 366-367

3 Türkçe sözlük. — genişl. 7. bs. — [Ankara] : Türk Dil Kurumu 1983. — s. 1253

Öte yandan American Heritage Dictionary kavramı "bilgi, özellikle çözümleme için örgütlenmiş ya da bir kararın temellendirilmesi için kullanılan bilgi olarak"⁴ tanımlamaktadır. Bu noktada, kavramın en önemli yanlarından biri ortaya çıkıyor: kavram bilgi kavramıyla yakından ilişkilidir.

Veri, zihinsel bir yargılamaya temel olan başlıca ögedir ve sonuçta bilgiden farklı bir şey olmayan (ya da bir tür çıkarsanmış bilgi olan) bir yargı veya kararı ortaya çıkartmaktadır. Yani bilgi, veri olarak adlandırılan başka bilgilerden çıkarsamalar yoluyla türetilir.

Öyleyse veri, işlenerek yeni bilgiler türetilen bilgidir, yani bir tür işlenmemiş, ham bilgidir. Ancak burada kullanılan "ham bilgi" ifadesi aslında yanıltıcı bir ifade. Konuyu bu noktada örneklerle açmak gerekir:

Örneğin 3 sayısı bir veridir. Aynı zamanda kendi anlamı dışında herhangi bir bilgi vermediğinden, ham bir bilgidir. Bu sayı, her hangi bir çocuğun yaşı, bir öğrencinin aldığı not ya da bir nesneye ait bir ölçü (en. boy. yükseklik v.b. gibi büyüklükler) olabilir. Sayı diğer her hangi bir nesne ile tanımlanmış bir ilişkiye girdiği zaman, ortaya bir bilgi

4 American heritage dictionary...

çıkmaktadır. Örneğin "3" sayısı "araba" adlı nesne ile, "metre cinsinden uzunluğu olma" olarak tanımlanan ilişkiye girdiğinde, ortaya "arabanın metre cinsinden uzunluğu 3'tür" gibi, önerme biçiminde bir bilgi çıkar.

Ancak sözgelimi, yukardaki önermede anılan "araba" adlı nesne, "garaj" adlı kümenin bir ögesi olsun. Aynı kümenin diğer iki ögesi de "kamyonet" ve "otobüs" adlı nesneler olsun. "Kamyonet" adlı nesne "4" sayısı ile, "otobüs" adlı nesne de "5" sayısı ile, "metre cinsinden uzunluğu olma" olarak tanımlanan ilişkiye girdiğinde, şu önermelere ulaşılır:

- (1) "arabanın metre cinsinden uzunluğu 3'tür"
- (2) "kamyonetin metre cinsinden uzunluğu 4'tür"
- (3) "otobüsün metre cinsinden uzunluğu 5'tir"

Aslında başlı başına birer bilgi olan bu üç önermeden, çıkarısama yoluyla, bir dördüncüsüne de ulaşılabilir:

- (4) "garajdaki araçların metre cinsinden uzunluklarının ortalaması 4'tür"

İşte ham bilgi kavramının yanıltıcılığı da burada yatar: 1, 2 ve 3 numaralı önermeler, ya da önerme biçimindeki bilgiler, aslında 4 numaralı önermeye ulaşmak için, çıkarsamaya tutulacak birer veridirler; ancak bunlara ham

bilgi demek doğru değildir, çünkü üçü de aynı zamanda başlı başına birer bilgidir.

Bu nedenle, veri kavramını, işlenmeye hazır bilgi ögesi olarak açıklamak, yukarda sözü edilen yanlışlığı ortadan kaldıracığından, daha uygundur.

1.2.1 Bilgi

Bu noktada bilgi kavramının, Türkçe'nin yapısından kaynaklanan, ve çelişkilere yol açabilecek bir eşeslilik nedeniyle, kendi içindeki iki anlamını ayırmsamak gereklidir: Bilgi terimi, birbirinden farklı iki kavram olan "enformasyon" ve "birikim" (ing. knowledge) kavramlarını karşılayabilmektedir⁵. Ancak birikim, "akılda kalan bilgi"dir⁶. Bu araştırmada ise, genel kullanım gözönünde bulundurularak, "bilgi" terimi, doğrudan doğruya "enformasyon" kavramı yerine kullanılmıştır.

Kavramın "veri" kavramı ile olan ilişkisi göz önünde bulundurulacak olursa, şu sonuca da ulaşılabilir: "bilgi, işlenmeye hazır bilgi öğeleri ya da veriden çıkarsanacak sonuç, varılacak yargıdır".

5 Aysel Yontar. Bilgi bilime giriş [ders notları]. — [İstanbul] : [y.y.], [1992]. — Not : Metin teksirdir. — s. 6

6 Michael H. Brackett. Developing data structured databases. — Engelwood Cliffs, New Jersey : Prentice-Hall, 1987. — s. 214

1.3 Veri tabanı

Yukarda da belirtildiği gibi veri tabanı kavramı, özellikle bilgisayar ortamındaki uygulamalarda, verilerin yüklü bulunduğu örgütlü teknik düzenegi (kütüphanecilik bakış açısıyla bir belgeyi) belirtir. Ancak gene de kavramı çeşitli bakış açılarından irdelemekte yarar vardır:

Yazılım pazarındaki ticari ilişkisel veri tabanı yönetim sistemi yazılımlarından biri olan Paradox, veri tabanı kavramını özellikle kütüphane ve bilgi bilimlerinin yaklaşımlarına koşutluk gösterecek tarzda, "örgütlü bir bilgi ya da veri dermesi"⁷ olarak tanımlamaktadır. Bilgi dermesinin örgütlenmesinin bu iki bilim dalının temel alanlarından biri olduğu düşünülecek olursa kavramın özellikle bilgi bilim açısından önemi açıkça ortaya çıkmaktadır: kütüphaneler, arşivler ve belge merkezleri, örgütlü bir belge dermesinin yönetildiği sistemlerdir; bilgi merkezleri ise bunun diğer bir boyutu olan örgütlü bilgi dermelerinin yönetildiği sistemlerdir. Bu nedenle, veri tabanları, bilgi merkezlerinin dermesidirler.

Buna göre denebilir ki, veri tabanları bilgi merkezlerinin temel unsurudur. Veri tabanı yönetim

7 Paradox for Windows. Getting started. — Scotts Valley, California : Borland International, 1992. — s. 2

sistemleri, bilgi merkezleri ile işlev bakımından koşut, hatta aynıdırlar.

Dermenin bilgi dermesi mi, yoksa veri dermesi mi olduğu sorusunun bu tanımda yanıtlanmamış olması, yukarda da söz edilen veri ve bilgi kavramlarının anlam yakınlığından kaynaklandığı bir gerçektir. Ancak bu iki kavramın ayrımı da göz önünde bulundurularak, veri kavramını benimsemek gerekmektedir.

Tanımın çözümlenmesi sonucunda ulaşılan sonuç, veri tabanının bir veri dermesi olduğu, ve ayırd edici özelliğinin, örgütlenmesi olduğudur.

Ancak önemli olan bir diğer nokta, yukarki tanımda veri tabanının fiziksel biçimine değinilmemesidir. Özellikle bilgisayar çevrelerinde yapılan önemli bir hata da burada yatmaktadır: kullanılagelenin aksine, veri tabanı kavramı özellikle bilgisayara ait bir kavram değildir, yani veri tabanının fiziksel biçimi, ille de bir bilgisayar ortamı değildir. Örgütlü olan her biçimdeki, her türden ve her ortamdaki veri dermesi bir veri tabanıdır: Örneğin bir telefon rehberi, bir sözlük, her hangi bir çizelge özellikle de "kütüphanelerin kart katalogları" hep birer veri tabanıdırlar.

Kütüphane otomasyonu konusundaki arařtırmaları ve alıřmalarıyla uluslararası düzeyde bir üne sahip olan Robert M. Hayes, veri tabanı kavramını "kullanıma uygun, makine dilindeki bir veri dosyası"⁹ olarak tanımlamaktadır. Bu veri tabanı kavramının özellikle bilgisayar ortamı için kullanılagelen fiziksel biçimidir: veri tabanı, makine diline uygun bir dosyadır, yani bir bilgisayar dosyasıdır.

Yetkin danışma kaynaklarından American Heritage Dictionary, veri tabanı kavramını şöyle tanımlamaktadır: "örneğin bilgisayarda olduđu gibi, kolay ve hızlı bilgi erişimini sağlayacak şekilde düzenlenmiş veri topluluđu"¹⁰.

Tanımda ilk göze arpan nokta, Paradox'un tanımına kořut olarak, bilgisayar dosyasının veri tabanının fiziksel biçimlerinden sadece biri olduđudur. Ancak bilgisayar ortamının hızlı ve kolay bilgi erişimini sağlaması, bu tanımın asıl önemli yanıdır. Bu hız ve kolaylıđı ise, söz konusu veri dermesinin düzeni, yani veri tabanı örgütlenmesi sağlamaktadır.

Veri tabanının örgütlenme düzenini, yazılım piyasasındaki ticari veri tabanı yönetim sistemi

9 Robert Hayes. Handbook of data processing for libraries . — 2nd ed. — Los Angeles : Melville Publishing Company, 1974. — s. 636

10 American heritage dictionary... — s. 366

yazılımlarının, özellikle 80'li yıllarda en yaygın olarak kullanılanlarından biri olan, Dbase 3 Plus adlı yazılım, veri tabanı kavramına önerdiği tanımda şöyle açıklamaktadır: "bir çok varlık hakkındaki aynı türden bilgi"¹¹. Yani bir veri tabanı dosyasının yapısında birden çok varlık hakkında bilgi vardır, ve bu bilgi hepsi için aynı düzendedir: örneğin değişik elementlere ait bir veri tabanında, her elementin atom numarası, özgül ağırlığı, elektron sayısı v.b. gibi elementlerin doğasını açıklayan bilgiler bulunabilir. Bu bilgilerin düzeni, tüm elementler için aynıdır, yani her elementin atom numarası, özgül ağırlığı v.b., bu veri tabanında yer almaktadır.

Bu tanım, ilerde de ele alınacağı gibi, söz konusu yazılımın doğası gereği özellikle ilişkisel veri tabanlarının yaklaşımına koşutluk göstermektedir. Ancak gene de, veri tabanlarının örgütlenme düzeni hakkında bir fikir verdiğinden, kayda değerdir.

Veri tabanı kavramıyla sıkça karıştırılan bir kavram da veri bankası kavramıdır. İki kavram arasındaki ayrım, özellikle son zamanlarda ortadan kalktığından, her iki kavram veri tabanı terimi altında birleşmiştir. Bunun nedenini, veri bankası kavramına Hayes'in önerdiği tanımla açıklamak

11 Dbase 3 Plus. Introduction to Dbase 3 Plus [bilgisayar dosyası]. — [y.y.] : Ashton-Tate. [1986?]

olasıdır. Hayes veri bankasını, "genellikle çok büyük ölcekte olup, çok kullanıcı amaçlı olan bir veri tabanı"¹² olarak tanımlamaktadır. Buna göre veri bankası, bir tür veri tabanıdır; ancak veri tabanından ayrıldığı nokta, nicel olarak daha büyük olması, yani daha çok öge hakkında veri içermesi, ve birden fazla kullanıcı tarafından erişilebilir ve işlenebilir olmasıdır.

Ancak geçen zaman içinde gelişen veri tabanı teknolojisi bu ayrımı ortadan kaldırmıştır. Artık neredeyse her veri tabanı birden fazla kullanıcı tarafından erişilebilir ve işlenebilir durumdadır; ve örneğin Paradox for Windows ile yaratılan veri tabanı dosyalarına, iki milyar adet civarında öge hakkında veri girilebilmektedir. Bu nedenle de veri tabanı terimi, veri bankası kavramını da artık içerdiğinden, özellikle günümüzde kullanılan veri tabanı kavramı ile veri bankası kavramının ayrımı ortadan kalkmıştır. Buna koşut olarak da veri bankası teriminin bir geçerliliği kalmamıştır.

Bu arada veri tabanı terimi, kimi zaman da belirli konularda veri içeren bilgi sistemi anlamında da kullanılmaktadır. Terimin bu yanlış kullanımının yaygınlığına karşın, ortaya çıkabilecek yanlış anlaşımaların önlenmesi

12 Hayes...

için veri tabanı ve bilgi sistemi terimlerinin birbirlerinden ayrılmasında yarar vardır.

1.3.1 Veri Tabanı Yaklaşımının Amaç ve Kapsamı

Veri tabanı yaklaşımı, 1960'ların sonlarına doğru¹³, gelişmekte olan bilgisayar teknolojisiyle birlikte, anlık işlemler ve hesaplamalar yerine, belli mantıkal yapılar içinde saklanacak bilgilere dayanarak, yüksek düzeyli, karmaşık ve her şeyden önce, planlamaya/karar vermeye veya denetimi sağlamaya yönelik olarak¹⁴, sorgulamaların yapılabileceği sistemler anlayışının ağırlık kazanmasından sonra, gündeme gelmeye başlamıştır.

Veri tabanı düşüncesinin, o zamana kadar geçerliliğini korumuş olan "geleneksel dosya yaklaşımı"ndan¹⁵ temel farkı, her hangi bir konudaki verinin "ayrı dosyalarda saklanan bağımsız birimlerde değil, ilişkilendirilmiş bir biçimde, tek birimde"¹⁶ saklanması düşüncesidir. Bu düşünce doğrultusunda ortaya çıkan sonuç, verinin derlenerek belli

13 Hermann Sauer. Relationale Datenbanken : Theorie und Praxis, inklusive SQL-2 . — 2. korrig. u. erw. Aufl. — Bonn... [ve öte.] : Addison-Wesley, 1992. — s. [1]

14 Elizabeth A. Oxborrow. Databases and database control systems : concepts and issues. — 2nd ed. — Bromley, Kent, England : Chartwell-Bratt. c. 1989. — s. 3

15 Oxborrow...

16 Oxborrow...

bir örgütlenme içinde tek dosyada. kısaca aynı belli tabanda, yani yukarda da sözü edilen veri tabanı dosyasında saklanmasıdır.

Burada özellikle belirtilmesi gereken nokta, geleneksel dosya yaklaşımında, verinin saklandığı birbirlerinden bağımsız dosyaların kendilerini yönetecek olan yazılıma olan bağımlılıklarıdır¹⁷. Bu yaklaşımın en büyük yitirimi, aynı verinin ayrı işlevleri yüklenmiş olan değişik yazılımlarla birlikte, yazılımlar ancak birbirlerinin ardı sıra çalıştırılabildiğinden, gereksiz yere tekrar edilmesidir¹⁸. Oysa devrim niteliğini taşıyan veri tabanı yaklaşımında, merkezi bir veri sözlüğünün, veri dosyalarından ve uygulama yazılımlarından bağımsız olarak, değişik veri tabanı dosyalarının içeriklerini tanımlayabilmesi nedeniyle, aynı veri tabanı dosyası, birden çok uygulama yazılımı tarafından işlenebilmekte, veya anlık işlemler yapılabilmektedir. Sözelimi, geleneksel dosya yaklaşımında kütüphane yazılımları sağlama, kataloglama, dolasım ve işbirliği gibi, ayrı yazılımlardan oluşmaktadır. Sağlamaya ait veri, yalnızca sağlama yazılımında kullanılabildiğinden, aynı veri kataloglama yazılımı için bilgisayara tekrar yüklenmekte. eğer kataloglanan kitabın ait veri dolasım ya da işbirliği yazılımlarında kullanılacaksa, aynı veri tekrar

58 Sauer... — s. 23

17 Sauer... — s. 13

tekrar bilgisayara girilmekte, gereksiz yere bellek harcanmaktadır. Oysa veri tabanı yaklaşımında, katalog verisi sadece katalog veri tabanında saklanmakta, sağlama, kataloglama, dolaşım ve işbirliği yazılımları, ya da çağdaş programlama yaklaşımıyla, tümleşik bir üst yazılım olan kütüphane yönetimi yazılımının alt yazılımları olarak değerlendirilebilecek, yordamları, bu veri tabanı dosyasını ortaklaşa kullanmaktadırlar.

1975'te yayınlanan, ANSI/SPARC Modeli olarak da anılan, konuya ilişkin standart, veri tabanı yaklaşımını birbirleri ile dikey bağlantılı üç temel katmanda ele almaktadır¹⁹:

KATMAN	BAKİŞ AÇISI
Dış katman	Kullanıcı bakışı
.....	
kavramsal katman	Mantıksal genel bakış
.....	
İç katman	Fiziksel bakış

ÇİZ. 1, ANSI/SPARC MODELİ

ANSI/SPARC Modeli'ne göre, bu üç katman da birbirlerinden fiziksel olarak tamamen bağımsız birimler olmak zorundadırlar²⁰. \öylelikle verinin fiziksel tutanak

19 Sauer... — s. 12

20 Sauer...

yapısının, kullanıcının bakış açısında tamamen bağımsız olması sağlanmış. Burada bulunan kavramsal katman ile kullanıcıya mantıksal açıdan bir genel bakış sağlanarak buradan fiziksel olarak ayrı yerleşimlere sahip olan değişik veri tabanı dosyalarına, kurulan mantıksal bağlantılarla, merkezi erişim olanağı verilmiştir.

Ancak E. F. Codd, bu modeli, tanımlarının kesin olmaması ve yoruma bir hayli açık olması nedeniyle sakıncalı bulur²¹. Model Codd'a göre, çok önceden, dış katman teriminin yerine bakış (views) teriminin, iç katman teriminin yerine yükleme göstergeleri (storage representations) teriminin ve mantıksal katman yerine de taban ilişkileri (base relations) teriminin kullanıldığı ilişkisel modelde tanımlandığını belirtir²².

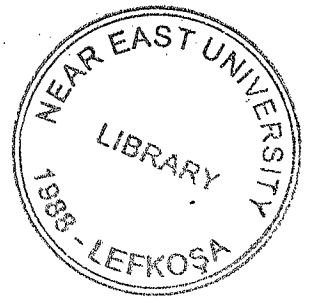
1.3.2 Veri Tabanı Geliştirme Süreci

Oxborrow çeşitli araştırmacılarca önerilen veri tabanı geliştirme yöntemlerini, özet olarak, aşağıda amaçları ve sonuçta elde edilmek istenen son ürünleri belirtilen, beş alt süreçten oluşan, şu sürece yansıtmıştır²³:

21 Edgar F. Codd. The relational model for database management : version 2. — rep. with corr. — Reading, Massachusetts... [ve öte.] : Addison-Wesley, 1990. — s. 33

22 Codd... — s. 34

23 Oxborrow... — s. 17



1. Veri araştırması:

Amaç : Verinin doğasının ve kullanımının tanımlanması

Son ürün : İskelet, veri sözlüğü

2. Veri örnekleme:

Amaç : Gerçekliğe ait olguların kendilerini yansıtacak bir örneğe yüklenmesi

Son ürün : Genellenmiş mantıksal örnek

3. Veri tabanı tasarımı:

Amaç : Mantıksal örneğin v.t.y.s.'ne özgü veri örneğine yansıtılması

Son ürün : v.t.y.s.'ne özgü mantıksal örnek

4. Veri tabanı işletimi:

Amaç : Olgusal verinin fiziksel veri tabanına yüklenmesi

Son ürün : İç katman ve veri tabanı dosyası

5. Veri tabanı ayarlaması:

Amaç : Veri tabanının kullanımının düzenlenmesi ve gerek duyulan durumlarda veri tabanının yeniden yapılandırılması

Son ürün : Elverişli veri tabanı

Oxborrow tarafından oraya konan, sistem analizi sürecini andıran, bu süreci ortaya çıkaran alt süreçleri incelemek, konuya yaklaşımı daha da anlaşılır kılacaktır.

1.3.2.1 Veri Araştırması

Veri araştırması alt süreci her yönden veri tabanı geliştirme sürecinin en önemli alt sürecidir. Bu önemin nedeni gerçekleştirilecek veri araştırmasının tutarlılığı, doğruluğu, konuya ve amaca uygunluğu, gerçekliği yansıtma derecesi v.b. niteliklerinin, veri tabanının niteliklerini, verimini ve etkinliğini doğrudan etkileyecek olmasıdır. Yukarda da ele alındığı gibi, amacı verinin doğasının ve kullanımının tanımlanması olan, veri araştırması, bu nedenle kesin olarak geliştirilmek istenen veri tabanı hangi konuda ise o konunun, araştırılacak verinin sistem içindeki konumunu saptayabilecek, sistem analizi bilgisi olan, uzmanınca gerçekleştirilmelidir.

Araştırılarak derlenecek "bilgi. sistem içinde yer alacak nesnelere özgü ayrıntıları, nesnelere yönelik kısıtlamaları, birbirleriyle kurdukları ilişkileri, fiziksel niteliklerini v.b.'ni içermelidir"²⁴.

Buna ek olarak kullanılacak verinin sistem içinde nasıl işleneceği de belirlenmelidir²⁵. Böylelikle bir sonraki alt süreçte belirlenecek olan veri örneğinin sisteme uygunluğunun sağlanmasına da olanak tanınacaktır.

Bu alt süreçte derlenecek olan bilgi, ya da yukarıda da belirlenmiş olan iki kavramın ayrımı göz önünde bulundurularak, veri, veri tabanında yer alacak verinin doğasına özgü veridir. Bu tür veri metaveri (ing. metadata), yani veriye ilişkin veri, olarak adlandırılmaktadır²⁶.

Sonuç olarak, ortaya çıkacak olan iki son üründen ilki veri tabanının iskeletinden başka bir şey olmayacaktır. Diğerisi ise, veri sözlüğü olarak adlandırılan, veri tabanı yönetim sisteminin veriyi tanımasını sağlayacak olan, ve doğrudan doğruya metaveriyi içerecek olan üründür²⁷. Veri sözlüğü, ikinci alt süreçte oluşturulacak mantıksal veri örneğinin de yapısını belirleyecektir.

25 Oxborrow... — s. 18

26 Oxborrow...

27 Oxborrow...

1.3.2.2 Veri Örneklemesi

Veri örneklemesi alt sürecinin amacı, veri araştırmasında derlenen veriyi, yani metaveriyi, gerçekliğin mantıksal bir örneğine yansıtmaktır.

Veri örneklemesinin temellerinin neo-pozitivist felsefe akımının üyelerinden Ludwig Wittgenstein'in (1889-1951) bilgi kuramı (epistemoloji) konusundaki çalışmalarında yatabileceği konusunda, bazı spekülatif göstergeler dışında, şu anda her hangi bir bilimsel kanıt olmamasına karşın, her iki konunun birbiriyle örtüşme derecesinde ilişkili olmaları, tartışmasız bir gerçektir.

Wittgenstein yedi temel felsefi önerme ve bunların alt önermelerinden oluşan "Tractatus logico-philosophicus"²⁸ adlı incelemesinde, önce gerçekliğin ve olguların ne olduğunu tartışır.

Sonuçta su önermeye ulaşır:

"Bizler olguların resimlerini yaparız." (Önerme 2.1).

28 Ludwig Wittgenstein. Tractatus logico-philosophicus = Logisch-philosophische Abhandlung. — Frankfurt am Main : Suhrkamp Verlag, c. 1963. 1989. — s. 16-18

Buna baęlı olarak da "resmin gereklięin bir rneęi"
olduęunu belirler (nerme 2.12).

Wittgenstein daha sonra řu nermeyi saptar:

"Nesneleri resimde resmin geleri karřılar" (nerme
2.13).

Buna gre, gereklięe ait nesneleri, mantıksal rnekte,
rneęin geleri karřılamaktadır.

"Resim, gelerinin birbirleriyle belli bir tarz ve
biimde iliřki kurmalarından oluřur" (nerme 2.14).

Yani mantıksal rnek, kendisini oluřturan gelerin belli
bir tarz ve biimde birbirleriyle iliřki kurmalarından
oluřur. Wittgenstein saptamalarına devam eder:

"gelerin bu iliřkisi, resmin yapısını. [resmin]
olasılıęı da resimlemenin biimini ifade eder"
(nerme 2.15).

"Buna gre resme, bir de onu resim yapan resimleyen

iliski aittir" (önerme 2.1513).

"Resimleyen iliski nesneleri resmin ögelerinin karşılarnasından oluşur" (önerme 2.1514).

"Olgu resim olabilmek için, resimlenen ile ortak bir şeye sahip olmalıdır" (önerme 2.16).

"Resim ve resimlenende, birinin diğernin resmi olabilmesi için, bir şeyin özdes olması gerekir" (önerme 2.161).

"Resmin, gerçekliğı kendi tarz ve biçiminde -doğru ya da yanlış- resimleyebilmesi için, gerçeklik ile ortak sahip olması gereken şey resimlemenin biçimidir" (önerme 2.17).

"Hangi biçimde olurlarsa olsunlar, bütün resimlerin gerçekliğı -doğru ya da yanlış- resimleyebilmeleri için, gerçeklikle ortak olarak sahip olmaları gereken şey, mantıksal biçimdir. ki bu da gerçekliğın biçimidir" (önerme 2.18).

Kısacası, mantıksal örneğın yapısı, doğrudan doğruya gerçekliğın mantıksal biçimini yansıtmalı, hatta ona koşut olmalıdır. Bu noktada, veri örneğini oluşturan ögeleri tanıtmakta yarar vardır.

1.3.2.2.1 Veri Örneğinin Öğeleri

1.3.2.2.1.1 ilişkiler

Çeşitli araştırmacılar tarafından ortaya konmuş olan ilişki gruplamaları vardır. Ancak bunların ötesinde, veri örnekleme için etkili olabilecek nokta, ilişkilerin değişik türleridir.

İlişki türlerinin iki belirleyici özelliği vardır: ilişkinin derecesi ve varlığı²⁹.

İlişkinin derecesi ya tanımsızdır, ya bire bir ilişki söz konusudur, ya bire çok ilişki ya da çoka çok ilişki.

Tanımsız ilişkiler, bu araştırmanın konusu dışıdır.

Bire bir ilişkiye verilebilecek en güzel örnek, öğrenci öğrenci numarası ilişkisidir. Her öğrencinin sadece bir öğrenci numarası vardır; ve her öğrenci numarası sadece bir öğrenciye aittir.

Bire çok ilişki, anne (ya da baba) ile çocukları arasında vardır. Her çocuğun mutlaka bir annesi (ya da

babası) vardır; ancak her annenin (ya da babanın) birden çok çocuğu olabilir.

Çok çok ilişki ise, öğrenciler ile dersler arasında vardır. Her öğrenci birden çok dersi alır ve her dersi birden çok öğrenci alır.

İlişkinin varlığı, ya zorunludur, ya seçmeli ya da olası.

Zorunlu ilişki, gene öğrenci öğrenci numarası ilişkisiyle açıklanabilir. Her öğrencinin sadece bir öğrenci numarası olmalıdır ve her öğrenci numarası sadece bir öğrenciye ait olmalıdır.

Seçmeli ilişki öğrenci ile yüksekokul arasında vardır: her öğrenci bir yüksekokula gidebilir; ve her yüksek okula değişik öğrenciler gidebilir. Görüleceği gibi burada bir zorunluluk değil, aksine bir seçim vardır.

Olası ilişki, ilişkilerin en karmaşık olanıdır. Burada bir taraf için bir zorunluluk varken, diğeri için seçim vardır: her kitabın mutlaka bir yazarı vardır, ancak her yazarın mutlaka bir kitap yazmış olması gerekmez.

1.3.2.2.1.2 Varlıklar ve Yüklemler

"Varlıklar, gerçeklikte var olan, bizim ilgimizi çeken çeşitli nitelikleri taşıyan nesnelerdir. Yüklemler ise, bir varlığın niteliklerinin her biri olarak tanımlanabilir"³⁰. Örneğin, kitap bir varlıktır; kitabın yazarı, çevirmeni, yayıncısı, yayın tarihi, sayfa sayısı v.b. nitelikleri, ya da özellikleri, kitap adlı varlığın yüklemleridirler.

Varlıklar, Wittgenstein'a göre gerçekliğin nesneleridirler. Yazar 2.01231 sayılı önermesinde "Bir nesneyi tanımak için, dış özelliklerini olmasa bile, iç özelliklerini tanımalıyım." der. Buna göre nesnenin iç ve dış özellikleri vardır; ancak nesnelerin, sonradan kazandıkları, dış özelliklerinin, kazanıldıktan bir süre sonra, nesnenin varoluşunun bir parçası, kısaca iç özelliği haline geldikleri düşünülecek olursa, varlıklara ait yüklemler için, böyle bir ayırımın gerekli olmadığı görülür. Dolayısıyla, varlık için bir yüklemler olma özelliğini kazanmış olan tüm iç ve dış niteliklerinin, varlık ile ilişkilendirilmeleri, her hangi bir ayırım olmaksızın, olasıdır.

1.3.2.2.1.3 Yöreler ve Tanımlayıcılar

Yöre terimi bu araştırmada, Türkçe kaynaklarda tam bir karşılığı bulunamamış olan "domain" terimi için nerilmektedir. Terim, varlıkların ve yüklemelerin ait oldukları özel sınıf³¹, yüklemelerin içerdikleri "değerleri aldıkları, soyut bir değerler havuzu"³² anlamına geldiği söylenebilir. "Belli bir türdeki, olası tüm değerleri içeren yöreler"³³ özellikle veri tabanı geliştirme sürecinin bir sonraki alt süreci olan, veri tabanı tasarımı, verinin doğasının belirlenmesinde, büyük önem taşırlar; yani veri bir tarih mi olacaktır, bir sayı mı yoksa bir kişi adı mı, v.b.

Tanımlayıcılar, gerçekliğin nesnelerinin mantıksal örnekteki soyut karşılığı olacaklardır.

1.3.2.2.2 Veri Örneğinin Saptanmasının Aşamaları

Veri örneği, tüm bu öğelere dayanılarak, şu akış içinde saptanır³⁴:

31 Oxborrow... — s. 26

32 C. J. Date. Relational database writings, 1985-1989 / by C. J. Date: with a special contribution by Andrew Warden. — repr. with corr. — Reading, Massachusetts... [ve öte.] : Addison-Wesley, 1990. — s. 29

33 Date...

34 Oxborrow... — s. 27-37

1. Varlık kümeleri ve ilişkiler tanımlanır
2. Yüklemler tanımlanır
3. İlişki kümeleri tanımlanır
4. Tanımlayıcılar saptanır
5. Yöreler belirlenir
6. Bütün bu öğeler bir araya getirilerek, tam mantıksal örnek kurulur

1.3.2.3 Veri Tabanı Tasarımı

Veri tabanı tasarımı, yukarıda kurulan mantıksal veri örneğinin, oluşturulmak istenen veri tabanının geliştirileceği, veri tabanı yönetim sisteminde yeni baştan tasarlanması, sistemin olanakları göz önünde bulundurularak mantıksal sistemin veri tabanı yapısına uyarlanmasıdır. Yukarıda da belirtildiği gibi elde edilecek son ürün, veri tabanı dosyasının bilgisayar ortamındaki mantıksal yapısıdır.

1.3.2.4 Veri Tabanı İşletimi

Veri tabanı işletimi alt sürecinde, veri tabanında dermesi oluşturulacak veri, artık fiziksel olarak bilgisayar ortamındaki veri tabanına girilmeye başlanır. Bu süreç içinde, gerekli görülen girdi düzeltmeleri yapılır. Son ürün olarak veri tabanı dosyası elde edilir.

Bu arada geliştirilebilecek diğer bir ürün de, bu dosyaya ait işletim yordamlarıdır.

1.3.2.5 Veri Tabanı Ayarlaması

Bu alt süreçte, artık veri tabanının yapısına ilişkin son ayarlamalar yapılır. Bu aşamada, yapıya ufak tefek müdahaleler yapılabilir. Bu gibi başlı başına yeniden yapılandırma çalışmalarına da gidilebilir.

Veri tabanı geliştirme süreci aslına bakılacak olursa hiç bitmeyecek bir süreçtir. Zaten veri tabanının yapısının ve içeriğinin güncel şartlara ve gereksinimlere uyum sağlayabilmesi için, en sağlıklı yaklaşım da bu doğrultudadır.

1.3.3 Veri Tabanlarının Türleri

Veri tabanları, örgütlenme yapıları açısından zaman içinde farklılıklar göstermişler, ve böylece de veri tabanlarının değişik türleri ortaya çıkmıştır. Ancak tarihsel açıdan konuya yaklaşıldığında, yaklaşım olarak veri tabanı düşüncesinin değişmediği, aksine değişimin yapısal nitelikli olduğu görülür. Bu yüzden ki, yeni bir tür ortaya çıktığında eskisi etkisini yitirmektedir.

Veri tabanlarında yaşanan bu yapısal değişikliklerin başlıca nedeni, veri tabanının yapısının giderek gerçekliğin bir örneğinin bilgisayar ortamındaki yansıması olmaktan, uygulama amacına yönelik olarak yeniden biçimlendirilmiş bir yapıya yönelmesidir. Örneğin ilk türlerden olan hiyerarsik veri tabanlarında veri tabanı yapısı uygulama (özellikle de sorgulama) kaygısı taşımaksızın, gerçekliğin felsefi anlamda tamamen köşut bir yansımasıdır. Ağ yapılı veri tabanlarında, bu yapı biraz daha uygulamaya yönelik bir biçime sokulmaya çalışılmış, ancak elde edilen, hiyerarsik veri tabanının yapısındakinden biraz daha büyük bir karmaşa olmuştur.

Veri tabanı dünyasında ilk devrim niteliğini taşıyan gelişme yapı bakımından diğerlerinden oldukça yalın, özellikle sorgulamalarda büyük kolaylıklar sağlayan, ilişkisel veri tabanlarının ortaya çıkmasıdır.

İlişkisel veri tabanlarındaki kimi aksaklıklar ve kısıtlamaların giderilmeye çalışılmasıyla da, özellikle 90'lı yıllara damgasını vuracak gibi görünen, nesne yönelimli veri tabanları ortaya çıkmıştır.

Bu noktada tarihsel açıdan başlı başına eğilimleri temsil eden bu dört türü özet niteliğinde genel olarak ele almak, diğerlerine de sadece isim olarak da olsa, değinmekte yarar vardır.

1.3.3.1 Hiyerarşik Veri Tabanları

Oldukça karmaşık bir yapıya sahip olan hiyerarşik veri tabanlarında, veri tabanının yapısı, bir kökteki yüklemle bir ağacın dallarını andıracak biçimde³⁶ hiyerarşik olarak ilişkilendirilmiş başka yüklemelerden oluşur. Kendi içinde tutarlı olan bu yapılanmada yüklemeler başka bir köke de ait yüklemeler oldukları zaman, yani birden çok hiyerarşik yapı içinde yer almaya başladıkları zaman, veri tabanının yapısı, doğrudan doğruya ağ yapısına dönüşür.

Hiyerarşik veri tabanlarında, girişler ancak hierarşinin en üst düzeyindeki yüklemelerden yapılabilir³⁷.

1.3.3.2 Ağ Yapılı Veri Tabanları

Ağ yapılı veri tabanları özellikle CODASYL (COMmittee on DATA SYstems Languages = Veri Sistemleri Dilleri Komitesi) adlı gurubun çalışmaları sonucunda, 1960'ların sonlarında ortaya çıkmıştır³⁸.

36 İlişkisel sistemler ve diğerleri... — s. 38

37 Brackett... — s. 6

38 Oxborrow... — s. 68

Ağ yapılı veri tabanında, veri tabanının fiziksel örgütlenmesi yüklemeler ve varlıkların ilişkilerinden kurulan kümelerin, bir ağ biçiminde bir araya getirilmesinden oluşmaktadır. Böylesine karmaşık bir yapı, beraberinde karmaşık yönetim işlemlerini de getirmekte, bu da sistemin uygulamalarda zorluk çıkarmasına neden olmaktadır³⁹.

Ağ yapılı veri tabanlarında, girişlerin ağın her hangi bir yerinden yapılabilmesine karşın, kullanıcı bu karmaşık yapıyı çok iyi bilmek zorundadır⁴⁰.

Gerek ağ yapılı gerekse hiyerarşik veri tabanlarında karşılaşılan en önemli sorun, bu karmaşık yapının öğelerinin manyetik ortamda fiziksel olarak değişik adreslere dağıtılmış olmaları, ve bu nedenle ileride yapılmak istenebilecek bir yeniden yapılandırmanın çok güç olmasıdır⁴¹.

1.3.3.3 İlişkisel Veri Tabanları

İlişkisel veri tabanı düşüncesi. E. F. Codd'un 1970 yılında ortaya koyduğu kuramsal yaklaşımın bir ürünüdür⁴².

39 Oxborrow... — s. 68-77

40 Brackett...

41 Sauer... — s. 14-15

42 Sauer... — s. 19

Codd, ilişkisel veri tabanları için, yirmi yıla yakın bir sürede geliştirdiği, bu kuramsal yaklaşımı, daha sonra 1989 yılında, ilişkisel veri tabanlarının temel kuramı sayılabilecek "The relational model for database management : version 2" adlı kitabında⁴³ yayınlamıştır.

İlişkisel veri tabanlarının örgütlenmesi, tablolar biçiminde gerçekleşir⁴⁴. Örgütlenme yapısı ilişkisel ögeler, ilişkisel bütünlük kuralları ve ilişkisel işlemlerin tanımlarıyla belirlenmiştir⁴⁵.

İlişkisel ögeler, kısaca şöyle ele sıralanabilir⁴⁶: yöreler, ilişkiler, derece, yüklem, tutanaklar, birincil anahtar, ikincil anahtar, aday anahtar, yabancı anahtarlar ve karma anahtarlar.

Yöreler, ilişkiler ve yüklem, ki çoğu veri tabanı yönetin sisteminde bunlara "alan" da denir, ilişkisel veri tabanlarında da diğerleriyle aynı işlevleri taşırlar, ancak ilişkisel yapıda ilişkilerin yapılandırılmasına, hiyerarsik ve ağ yapılı veri tabanlarının aksine, önem verilmemiş, yüklemler arası ilişkiler, doğrudan yüklemelerin dilsel

43 Codd...

44 Paradox for Windows. Getting started...

45 Sauer... — s. 19-20

46 Sauer... — s. 20-27

mantığında gerçekleştirilmiştir. Bunun sonucunda, her varlık için yapı bakımından doğrusal ilişkiler kurulmuş, aynı doğruda yer alan yüklemeler, o varlık için ortaya çıkan tabloda bir satırı oluşturmuşlar, buna da tutanak adı verilmiştir. Böylelikle bir veri tabanını oluşturan, aynı sınıftan ya da türden farklı varlıklara ait tutanaklar, yapı bakımından birbirleriyle özdeş hale gelmişlerdir. Bunun sonucunda ortaya çıkan veri tabanı yapısı, yukarki örnekte de görüldüğü gibi, son derece yalın ve kolay anlaşılır bir biçime girmiş, buna kosut olarak da, sorgulama işlemleri hiç görülmediği kadar kolaylaşmış, ve dolayısıyla da hızlanmışlardır.

Bunun yanı sıra tabloların şu 4 özelliği ilişkisel yapı açısından çok önemlidir⁴⁷:

1. Aynı tabloda, birbirinin aynı olan, yani yüklemelerinin değerleri aynı olan, iki tutanak bulunamaz. Sözelimi aynı öğrenci için iki tutanak tutulamayacağı gibi.
2. Tutanakların sırası belirsizdir. Buna göre tutanakların hangi sıra ile tabloya girileceği tamamen kullanıcının tutumuna bağlıdır.

47 Sauer... — s. 24

3. Y klemlerin sırası belirsizdir. hangi y klemin, hangi sırada tabloda yer alacağı, gene tamamen kullanıcının tutumuna bağıdır.
4. Y klemlerin deęerleri b l ms zd r. Yani bir y klemin alacağı deęer iki ayrı deęerden oluşamaz.

Bu  zelliklerin ışığında ilişkisel  gelerden anahtarları anlamak daha kolaydır.

Birincil anahtar, yukarda anılan ilk  zelliğın bir sonucudur. Tabloda yer alacak her tutanak birbirinden farklı olmak zorundaysa, tutanağının en az bir y kleminin deęerinin, o y klemin dięer tutanaklar i in i erdiği deęerlerden farklı olması gerekmektedir. Bu y klem ise, birincil anahtarı oluřturur. Birincil anahtarın alacağı deęerler tek ve "biricik" (ing. unique) olmalıdır.

İliřkisel yapı gereęi, aynı tabloda hepsi birbirinden farklı olan kayıtlar bulunur; ancak bu kayıtların hangi y klem deęerinde farklılık g stereceęi, bilinemez.  rneğın bir n fus tutanakları tablosunda, kisilerin ya adları, ya soyadları, ya doęum tarihleri, ya anne-baba adları ya da doęum yerleri v.b.  zellikleri farklı olacaktır. Aynı deęeri taşıyan bir insanın varlığı, mantıksal olarak olası deęildir. Ancak farklılığın nerede olabileceğini deęil bilmek, kestirmek dahi olanaksızdır.

Özellikle belli öğelere ait verinin arandığı sorgulamalarda, o öğenin tutanağının, her değeri diğer değerlerden farklı olan bir yüklemi olması gerekmektedir. Bu, belirli bir tutanağa erişimde anahtar görevini üstlenmiş olan yükleme, ilişkisel yapıda birincil anahtar adı verilmiştir. Birincil anahtarların değerleri, bir okulda öğrenci numarası, bir ürün için seri numarası, değişik alanlardaki sicil numaraları, kişiler için ulusal düzeyde ulusal ya da sosyal güvenlik numaraları, kütüphane ya da belge-bilgi merkezlerindeki belgeler için demirbaş numaraları, olabilirler.

İkincil anahtarlar, ki bunlara kimi zaman seçmeli anahtar da denir, gene bir tablo içinde birincil anahtar dışında anahtar özelliğini taşıyan yüklemelerdir. Aday anahtarlar da bu özelliği taşırlar ancak ikincil anahtardan farklı olarak, aynı ilişkilere başka bir bakış açısı verilip, bazı yüklemeler bu bakış açısına göre, başka tabloları oluşturduklarında, aday anahtarlar bu tablolarda birincil anahtar rolünü üstlenirler.

Yabancı anahtarlar ise, bir tabloda yer alan, ancak başka bir tablonun da birincil anahtarı olan aday anahtarlardır.

Karma anahtarlar, bir varlığa ait tutanağı, diğerlerinden ayırmak için⁴⁰ iki ya da daha fazla yüklemden oluşan⁴¹ anahtarlardır.

Görüldüğü gibi, ilişkisel veri tabanında. yüklem verliklerinin temelini, ille de olgusal gerçeklikten almazlar. Sorgulma gibi tamamen uygulamaya yönelik konular da veri tabanının yapısını etkiler.

Basit bir ilişkisel veri tabanı. su çizimle gösterilebilir:

Birincil anahtar

yüklemler /\		
Öğrenci #	Soyadı	Adı
9008DE005	Arıkan	Aykut
9008KU031	Doğanay	Hakan
9008KU054	Ekici	Aynur

) tutanak

ÇİZ. 2. Basit bir ilişkisel veri tabanı

İlişkisel veri tabanlarında. su iki ilişkisel bütünlük kuralına dikkat etmek gerekir⁴²:

48 Brackett... — s. 40

49 Brackett... — s. 212

50 Sauer... — s. 28

1. Eger bir yüklem birincil anahtarsa, bunun değeri asla boş bırakılamaz (varlık bütünlüğü kuralı).
2. Yabancı anahtara sahip olan bir tabloda, bu yabancı anahtar eğer başka bir tablonun birincil anahtarına yönlendirme yapıyorsa, ilk tablonun yabancı anahtarının değeri, diğer tablonun birincil anahtarlarının değeriyle ya aynı olmalı ya da boş bırakılmalıdır (yönlendirme bütünlüğü kuralı).

İlişkisel veri tabanlarında, şu işlemler özellikle büyük önem taşırlar:

1. Seçme (select):

Bu işlemin amacı, veri tabanından, belirli yüklemelere ait şart koşulan değerleri taşıyan tutanakların⁵¹ seçilmesidir. İşlem şu çizimle de gösterilebilir⁵²:

A



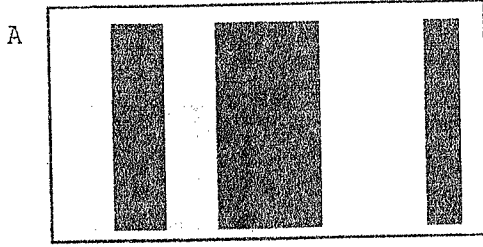
ÇİZ.3. Seçme işlemi

51 Codd... — s. 69

52 İlişkisel sistemler ve diğerleri... — s. 42

2. İzdüşüm (project):

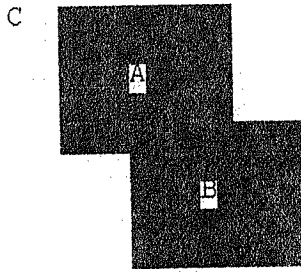
Bu işlemin amacı, veri tabanından, adları belirtilmiş yüklemelere ait tüm değerlerin seçilmesidir. İşlem şu çizimle de gösterilebilir:



ÇİZ. 4. İzdüşüm işlemi

3. Birleştirme (union):

Bu işlemin amacı, iki ayrı veri tabanını, her iki veri tabanında da bulunan tutanakları teke indirgeyerek, ayrı bir veri tabanında birleştirmektir. İşlem şu çizimle de gösterilebilir:



ÇİZ.5. Birleştirme işlemi

53 Codd... — s. 67

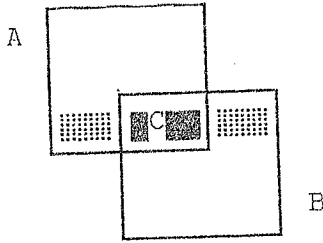
54 ilişkisel sistemler ve diğerleri... — s. 42

55 Codd... — s. 78

56 ilişkisel sistemler ve diğerleri... — s. 42

4. Kesişim (intersection):

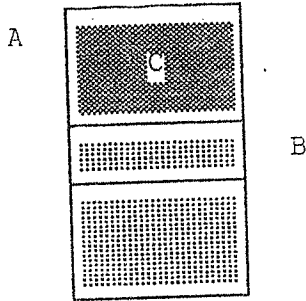
Bu işlemin amacı, iki ayrı veri tabanının, her ikisinde de bulunan tutanakları teke indirgeyerek, ayrı bir veri tabanında birleştirmektir⁵⁷. İşlem şu çizimle de gösterilebilir⁵⁸:



ÇİZ. 6. Kesişim işlemi

5. Ayrım (difference):

Bu işlemin amacı, bir veri tabanında bulunup, diğerinde bulunmayan tutanakları ayrı bir veri tabanında birleştirmektir⁵⁹. İşlem şu çizimle de gösterilebilir⁶⁰:



ÇİZ. 7. Ayrım işlemi

57 Codd... — s. 81

58 ilişkisel sistemler ve diğerleri... — s. 42

59 Codd... — s. 82

60 Codd... — s. 78

Nesne yönelimlilik yaklaşımı, özellikle 1990'lı yılların başlarından itibaren bilgi işlem çevrelerindeki güçlü bir programlama yöneliminin adıdır. Doğrudan veri tabanlarına da yansıyan yaklaşım, ilişkisel veri tabanlarının bir takım aksaklıkları karşısında, akılcı ve uygulanabilir çözümler sunduğundan, ilişkisel sistemlere henüz son verememekle birlikte, kimi çevrelerde çok çabuk kabul görmüştür.

Nesne yönelimlilik yaklaşımı, ilişkisel veri tabanı yaklaşımının taraftarlarınca, nesne yönelimlilik kavramının yeterince tanımlanamadığı⁶¹, ilişkisel veri örneği ile kıyaslanabilece nesne yönelimli bir veri örneği bulunmaması ve yaklaşımın veri tabanı teknolojisi için bir yenilik getirip getirmediğinin kesin olmaması⁶² gibi savlar ile eleştirilmektedir.

İlişkisel veri tabanlarının en büyük yitimlerinden biri, hiç kuşkusuz basit tablo yapısı içinde, özellikle karmaşık sorgulara yönelik karmaşık yapıları barındıramamalarıdır⁶³.

61 Date... — s. 50

62 Codd... — s. 480

63 Andreas Heuer. Objektorientierte datenbanken : Konzepte, Modelle, Systeme. — Bonn... [ve öte.] : Addison-Wesley, 1992. — s. 52

Örneğin katalog dosyası tutanak yapısında konu başlığı yüklemi için, birden çok değer alamamaktadır. Oysa kataloglanacak olan kitap, genelde birden çok konu başlığıyla donatılır. Bu durumda, ya tutanak bileşik bir konu başlığıyla donatılacak, ya da diğer konu başlıkları için, tutanaktaki diğer yüklemelerin değerleri, ikinci bir tutanakta diğer konu başlıklarıyla donatılarak, veri tabanı yaklaşımının tam aksine, gereksiz yere yinelenenlerdir. Basit bir hesaplama, bir katalog tutanağı ortalama 3 konu başlığı ile donatılacaksa, veri tabanının harcadığı bellek alanı 3 katına çıkacaktır.

Ancak bu durum, nesne yönelimli veri tabanları için sözkonusu değildir. Çünkü "nesne yönelimli veri tabanlarında, nesnelerin niteliklerini belirlemek için, standart veri yapılarının dışında, bunların tekrarlanabilir öğeleri de vardır"⁶⁴.

Yani, ilişkisel veri tabanında şu biçimde yer alacak katalog tutanağı:

Yazar adı	Kitap adı	Konu başlığı
Heuer, Andreas	Objektorientierte...	VERİ TABANLARI
Heuer, Andreas	Objektorientierte...	NESNE YÖNELİMLİLİK

ÇİZ. 8. ilişkisel veri tabanı örneği

64 Heuer, Andreas...

nesne yönelimli veri tabanlarında, şu biçimde yer alacaktır:

Yazar adı	Kitap adı	Konu başlıkları
Heuer, Andreas	Objektorientierte...	VERİ TABANLARI NESNE YÖNELİMLİLİK

ÇİZ. 9, Nesne yöneliml veri tabanı örneği

Kısacası ilişkisel veri tabanlarında, tutanaklar alanlardan oluşurken, nesne yönelimli veri tabanlarında tutanaklar nesne ile doğrudan ilişki kurabilen, ve sayıca sınırsız olabilen, alt alanlardan da oluşabilir.

Nesne yönelimli veri tabalarında bu nedenle, nesnenenin yapısında zamanın akışı içinde ortaya çıkabilecek değişiklikler, tutanak yapısı diğer nesneler için bozulmaksızın, veri tabanına yansıtılabilmektedir⁶⁵. Bu nedenle de, nesne yönelimli olarak adlandırılmaktadırlar. Hatırlanacağı gibi ilişkisel veri tabanlarında ilişki tablolarındaki tutanakların yapısında bu tür bireysel değişikliklere gitmek olanaksızdı.

Nesne yönelimli veri tabanlarında, nesnelerin üç önemli özelliği vardır⁶⁶:

65 Heuer, Andreas... — s. 56

66 Jon Udell. "Componentware" Byte Türkiye. — c. 1, s. 4 (Mayıs 1994). — s. 84

1. Veraset:

Bu özellik sayesinde yukarda da ele alındığı gibi, yapılarında zamanın akışı içinde ortaya çıkabilecek değişiklikler, tutanak yapısı diğer nesneler için bozulmaksızın, veri tabanına yansıtılabilmektedir.

2. Çokbiçimlilik:

Nesnelere yönelik tutanakların yapısı sabit değildir, böylelikle nesnelere değişik yüklem değerlerinden erişmek olasıdır.

3. Kapsama:

Nesnelere yönelik olan veri, nesnenin yapısı içinde, kendisini yorumlayacak olan bilgisayar kodu ile birlikte yer alır. Örneğin veri bir ses kaydı ise bu ses olduğu gibi bilgisayarda dinlenebilir.

Bu arada nesne yönelimli oldukları söylenen kimi ürünler için aslında nesne tabanlı demek doğru olacaktır. Çünkü bunlar yaklaşıma olan yakınlıklarına karşın, bu üç özelişin ya sadece birini ya da ikisini taşımaktadırlar; dolayısıyla da tam anlamıyla nesne yönelimli oldukları söylenemez. Bunlar⁴⁷:

- Belgeler, klasörler, çöp kutuları v.b. araçları kullanan, masaüstü mantığıyla çalışan işletim sistemleri için geliştirilmiş uygulama yazılımları.
- Nesneleri ikonlar (çizgisel göstergeler) ile gösteren sistemler, ve
- Grafik özellikleri taşıyan çeşitli kullanıcı arayüzleridir.

1.3.3.5 Diğer

Bu türlerin dışında bir de artık geçerlilikleri kalmamış ve kullanılmayan, Sırasız Liste olarak adlandırılan veri tabanları da vardır.

1.4 Veri Tabanı Yönetim Sistemi

Veri tabanı yönetim sistemi veri tabanı kavramının aksine, özellikle bilgisayara yönelik bir kavramdır. Tanımı fazla tartışmalı olmayan kavram. kısaca bilgisayar ortamındaki veri tabanlarını tanımlayan, yaratan, yapılarını ve içeriklerini değiştirerek, eklemeler ve çıkartmalar yaparak hepsinden önemlisi sorgulayarak işleten; ve sonuçta bilgisayar işletmeninin veri tabanından yararlanmasını

sağlayan, çoklukla ticari bir genel uygulama yazılımı, olarak tanımlanabilir.

Burada üstünde titizlikle durulması gereken bir nokta vardır: Aslında çoğu özel amaçlı uygulama yazılımı bir tür özel veri tabanı yönetim sistemidir. Bunların genel amaçlı veri tabanı yönetim sistemlerinden ayrılan yanı, özel amaçlı uygulama yazılımlarında kullanılan veri tabanı yapılarının belirli ve çoklukla değiştirilemez olması, ve gerçekleştirilebilecek işlemlerin de buna koşut olarak belirli ve kısıtlı olmaları, ancak genel amaçlı veri tabanı yönetim sistemlerinde bunun aksine, gerek veri tabanı yapılarının, gerekse işlemlerin çeşitlerinin sınırsız olmaları, yani tamamen kullanıcının gereksinimlerine göre düzenlenebilir olmalarıdır.

1.4.1 Veri Tabanı Yönetim Sistemlerinin İşlevleri

1.4.1.1 Tasarım

Tasarım, doğrudan doğruya, veri tabanı geliştirme sürecinin veri tabanı tasarımı alt sürecinde gerçekleştirilen işlemleri üstlenen bir işlevdir. Yani mantıksal veri örneği, bu işlev sayesinde veri tabanının manyetik ortamdaki yapısına dönüşür.

Yapılan işlemler sırasıyla şunlardır:

1. Yükleme (alan) adlarının belirlenmesi
2. Yüklemlerin (alanların) cinslerinin belirlenmesi (sayı, tarih, alfanumerik dizi v.b.)
3. Yükleme değerlerinin uzunluklarının belirlenmesi (alfanumerik diziler için)
4. Veri tabanı dosyasının yapısının, bir ad ile saklanması

Sonuçta ortaya, içi boş, veri girilmeye hazır, bir veri tabanı çıkar.

1.4.1.2 Güncelleştirme

Veri tabanı işletimi alt sürecini gerçekleştiren işlemdir. Bu işlevde ya yapısal ya da içeriksel değişikliklere gidilerek, veri tabanı güncel tutulur. Veri tabanına yeni tutanakların eklenmesi işlemi de, veri tabanında içeriksel değişikliğe yol açtığı için bir güncelleştirme işlemidir.

1.4.1.3 Sorgulama ve Raporlama

Bu işlev veri tabanı yönetim sisteminin en önemli işlevidir. Veri tabanı yönetim sisteminin birincil amacı ne veri tabanlarını kurmak, ne de onları güncelleştirmektir;

veri tabanı yönetim sistemlerinin birincil amacı, kullanıcıya bilgi sağlamaktır. Bunun yolu da sorgulamadan geçer.

Sorgulama işlevi ne kadar yetkinse, veri tabanı da aslına bakılacak olursa o kadar etkindir. Bu nedenle sorgulama kesin olarak çok amaçlı bir yapıya sahip olmalıdır. Yani örneğin, ilişkisel sistemlerde, sorgulama yalnızca anahtarlar üzerinden değil, diğer yüklemelerin alacağı değerler üzerinden de yapılabilmelidir. Tıpkı kütüphane kataloglarının kişi adı, başlık, konu başlığı, dizi adı v.b. öğelere ait girişlerden taranması gibi. Sorgulama bunun yanı sıra hata paylarını karşılayacak derecede esnek olmalıdır.

Raporlama sorgulamanın bir ilerki aşamasıdır. Sorgulamada elde edilen karşılıklar bu aşamada kullanıcıya isteğine ve gereksinimlerine bağlı olarak, her hangi bir biçimde ekran ya da yazıcı çıktısı olarak kullanıcıya sunulurlar. Yazıcıdan alınacak raporlar için kimi veri tabanı yönetim sistemleri, grafik ve değişik font seçenekleri gibi olanakları da kullanıcıya sağlamaktadırlar.

1.4.1.4 Dağıtık Veri Yönetimi

Günümüzde çoğu veri tabanı yönetim sistemi bilgisayar ağlarında da çalışabilme özelliğini göstermektedirler. Yani bir bilgisayarda işleyen bir veri tabanı yönetim sistemi, diğer bir bilgisayarda işleyen, uyumlu olduğu bir veri

tabanına, ya yerel ağ bağlantıları ya da modemler üstünden telefon veya veri iletişimi hatlarını kullanarak erişebilmekte, onu sorgulayabilmekte, verilerini paylaşabilmektedir.

Günümüzde Dağıtık Veri Yönetimi (Distributed Data Management) yaklaşımıyla, değişik bilgisayar sistemlerinin bir tabanda bir araya getirilebilmekte olması⁶⁸, özellikle veri tabanı yönetim sistemlerini, bu yaklaşıma uyumlu hale gelmeye, ya da en azından desteklemeye yöneltmektedir. Bu yaklaşım, özellikle bilgisayara dayalı kütüphaneler arası işbirliği alanında yaşanan bazı sorunlara da köktenci çözümler getirebilecek gibidir.

1.4.1.5 Diğer Genel İşlevleri

Veri tabanı yönetim sistemlerinin diğer genel işlevleri ise, veri tabanı yapısından bağımsız olarak, çoğaltma, silme, yeni ad verme gibi işletim sisteminin, ya da dosya yönetim sistemlerinin işlevleridir. Ancak bu işlevlerin veri tabanı yönetim sistemi içinde yer alması, kullanıcıya büyük rahatlık verdiği için, bunları veri tabanı yönetim sistemi içersinde tümleşik olarak bulunmaları, çoğunlukla tercih edilen bir durumdur. Aksi hallerde, kullanıcı veri tabanı yönetim

68 Inside IBM's Distributed Data Management architecture / by R. A. Demers... [ve öte.]. — s. 459-487. — IBM Systems Journal. — vol 31. No. 3 (1992). — 463

sistemi yazılımında işletim sistemine çıkmak. işi bittiğinde de veri tabanı yönetim sistemine dönmek zorunda kalacaktır.

1.5 Veri Tabanı Uygulama Yazılımları

Veri tabanı uygulama yazılımları, veri tabanı yönetim sistemlerinde, kimi özel uygulamalarda sıkça tekrarlanan yordamları oluşturan komutların, kullanıcı tarafından belirlenen üst başlıklar altında bir araya getirilmesinden oluşan yazılımlardır. Bu yazılımların tek amacı, veri tabanı yönetim sistemleri ile gerçekleştirilen belirli işlemleri kolaylaştırmak ve tümleşik bir yapıya oturtmaktır.

Bu tür yazılımların, veri tabanı yönetim sistemlerinin işleyişinden hiç bir farkları yoktur. Belirlenmiş olan komutlar veri tabanı yönetim sistemine ait komutlardır. Uygulama yazılımı işlerken, kullanıcı, aslında veri tabanı yönetim sisteminin işlediğinin, ayırımında değildir, ve sistemi uygulama yazılımının olanakları içinde, kendi ihtiyaçlarına göre, ayrıntılı olarak yapılan işlemleri bilmeksizin ve görmeksizin, yönlendirir.

Uygulama yazılımları: çoklukla kendi başlarına çalışmazlar; ancak bir yorumlayıcı yazılım, yani veri tabanı yönetim sistemi altında işlerler. Buna karşın son zamanlarda bu tür yazılımları kendi başlarına işler biçime dönüştürecek

derleyiciler de, fazla yaygınlık kazanmamakla beraber, pazarda yerlerini almışlardır. Codd bu tür derleyicilerin, derleme özelliklerinin yanı sıra, derlenmiş yazılımları geri kaynak koduna çevirecek, geri-derleme özeliğinin de bulunması gerektiğini savunur⁶⁹.

Özellikle, yoğun bir veri işleme ihtiyacı olan işletmelerde, bu tür uygulama yazılımlarının derlenmeden veri tabanı yönetim sisteminde yorumlatılarak kullanılması daha akılcı bir çözümdür, çünkü derlenmemiş bir yazılımın kaynak kodlarına girerek zaman içinde gerek duyulabilecek değişikliklerin, ekleme ve çıkartmaların yapılması olasıdır, ancak bu, derlenmiş olan yazılımlar için, yazılımın kaynak kodları makine koduna çevrilmiş olduğundan, söz konusu değildir. Yani derlenmiş bir yazılımda yapılabilecek işlemler, kurulabilecek olan veri tabanı dosyalarının yapıları, belirli ve sabitken, kaynak kodundaki derlenmemiş yazılımlarda bunları değiştirme olanakları sınırsızdır.

ANSI/SPARC modeliyle kurulan mantıksal bağlantıların sağladığı bir olanak ise, uygulama yazılımlarının son kullanıcıya, veri tabanı yönetim sisteminin olanaklarını, veri tabanı yönetim sisteminden bağımsız olarak sağlamasıdır.

Modelin özellikle Türkiye'ye sağladığı, yabancı literatürde, gereksinim duyulmadığı için hiç ele alınmayan, çok önemli başka bir olanak ise, son kullanıcıyı dil sorunundan kurtarmasıdır. Yazılım pazarında kullanıcıya sunulan veri tabanı yönetim sistemleri, sözcük işlemciler, hesap tabloları v.b. genel amaçlı uygulama yazılımlarının büyük bir bölümü, yurt dışında üretildiklerinden, Türkçe değildirler. Bu da son kullanıcıyı, bu uygulama yazılımlarından yararlanabilmek için, yabancı dil, özellikle de İngilizce, bilmek zorunda bırakmaktadır. Ancak ANSI/SPARC Modeline göre, geliştirilecek olan özel uygulama yazılımının dili Türkçe olabileceğinden bu yazılımı herkes kullanabilecektir.

Bunun en güzel kanıtı, yazılım pazarında son kullanıcıya sunulan özel amaçlı Türkçe uygulama yazılımlarının bazılarının, derlenerek veri tabanı yönetim sisteminden dil bakımından tamamen bağımsız bir biçime çevrilmiş yazılımlar olmalarıdır.

Veri tabanı uygulama yazılımları, geliştirildikleri veri tabanı yönetim sistemiyle uyumlu, onun tarafından yorumlanabilen, özel bir uygulama dili ile kodlanmışlardır.

1.5.1

Veri tabanı Uygulama Dilleri

Bu bölümde, özellikle yazılım pazarında satışa sunulan ticari yazılımlar hakkında, genel bilgi verilecektir.

1.5.1.1

Xbase/Clipper Ailesi

Xbase birbirleriyle uyumlu veri tabanı yapılarına sahip olan, yaygın bir ilişkisel veri tabanı yönetim sistemleri gurubudur. En çok tanınan üyeleri olan Dbase, günümüzde işletmelerde en çok kullanılan veri tabanı yönetim sistemidir. Bu kullanım çokluğunun nedeni sistemin gelişmişliği değil, aksine az gelişmişliğidir; çünkü bu nedenle fazla karmaşık uygulamalara gitmek istemeyen orta ve küçük çaplardaki kullanıcıların en çok tercih ettikleri sistemdir. Bu arada özellikle belirtmekte yarar vardır ki, gene aynı nedenden dolayı, yoğun olarak kullanılan Dbase sürümü son sürümlerden olan Dbase IV 2.0 değil bundan yaklaşık 9 yıl öncesinin sürümü olan Dbase III Plus'tır.

Geniş bir kullanıcı kesimi tarafından kullanılan bu sürüm, çoklukla, kendisi ile am uyumlu olan özgün derleyicisi, satışa kendisinden yıllar sonra sunulduğu için, o sıralarda vasat bir Xbase ailesi derleyicisi olup, kendisiyle tam uyumlu olmayan Clipper adlı derleyici ile derlenmiş, ortaya çıkan uygulamalar kişisel telefon rehberi

gerksinimlerini başarıyla karşılamış, ancak kütüphane uygulamaları gibi karmaşık uygulamalarda, sonuç doğal olarak başarısızlık ve hayal kırıklığı olmuştur; Mikrobeta Kutup örneğinde olduğu gibi.

1.5.1.2 SQL

IBM araştırmacıları tarafından 1972 yılında geliştirilen SQL (Structured Query Language = Yapılandırılmış Sorgulama Dili)⁷⁰ zaman içinde uluslararası bir veri tabanı uygulama dili standardı olmuştur ISO9075-199x SQL-2 (1989)⁷¹. Bunda özellikle kişisel bilgisayarlardan, iş istasyonları ve büyük ağ sistemlerinde uygulanabilir olması⁷² yatmaktadır. Kimi üreticiler ürünlerinde SQL standardını desteklemekte, kimileri de kendi uygulama dillerini SQL ile uyumlu hale getirmektedirler.

Codd, SQL'in uluslararası bir standart haline gelmesine karşın, bunun nedeninin standardın ilk önce Amerikan Ulusal Standartlar Enstitüsü ANSI tarafından kabul edilmesine bağlar; buna karşın, kendisinin 1971 yılında ortaya attığı ALPHA adlı dile dayanan, QUEL dilinin, SQL'den daha başarılı olduğunu savunur⁷³.

70 Codd... — s. 12

71 Sauer... — s. 12

72 Sauer... — s.43

73 Codd... — s. 12-13

1.1.5.1.3 ObjectPAL

ObjectPAL, Object Based Paradox Application Language (Türkçe'si Nesne Tabanlı Paradox Uygulama Dili) adlı uygulama dilinin kısaltılmış adıdır. ObjectPAL, Windows işletim sistemi altında çalışan Paradox for Windows veri tabanı yönetim sisteminde, Windows temelli tümleşik veri tabanı uygulamalarının geliştirilebileceği bir dildir⁷⁴. ObjectPAL ile geliştirilecek uygulamalar ürünün üreticisi tarafından geliştirilen, ve satışa sunulan, ürünle tam olarak uyumlu, RUNTIME adlı derleyici ile derlenerek, bağımsız bir uygulama yazılımı biçimine de dönüştürülebilir.

SQL standardını sorgulamaları ile destekleyen sistem, özellikle Windows gibi bir grafik kullanıcı arayüzünün verdiği olanaklarla, ilişkisel sistemin basit yapısını birleştirmiş, ve her düzeyde kullanıcı için Paradox for Windows gibi güçlü bir sistemi kullanılabilir kılmıştır.

1.5.2 Uygulama Yazılımlarının Geliştirilmesinde Dikkat Edilmesi Gerekenler

1.5.2.1 Dil

Veri tabanı uygulama yazılımlarında, dilin oynadığı rol, aslında diğer yazılımların geliştirilmesinde oynadığı rolle

74 Paradox for Windows. Guide to ObjectPAL. — Scotts Valley, California : Borland International, 1992. — s.1

aynıdır. Kullanıcı-sistem diyalogunda kullanılacak dil, sade ve anlaşılır olmalıdır. Ancak özel konulu uygulamalarda, genel terimlerin yerine konuya yönelik özel (kimi zaman da teknik) terimler kullanılabilir.

Ancak veri tabanı yönetim sistemine yönelik özel terimler, kullanıcıya yansıtılmamalı. kavramlar herkesin anlayacağı genel terimler ile ifade edilmeye çalışılmalıdır: örneğin yüklem yerine alan ya da hane, tutanak yerine de kayıt terimlerinin kullanılması gibi.

1.5.2.2 Kullanıcı Dostluğu

Tüm yazılımların kullanıcı dostu olması, çok önemlidir. Yazılımda ekran görüntüleri, gözü rahatsız edecek türden olmamalı: parlak renklerin yerine gözü daha az yoracak, sade ve uyumlu renklerin ve renk bileşimleri kullanılmalıdır.

Yazılımın işeyişi, kullanıcının gereksinimleri doğrultusunda olmalı. yordamlarda gereksiz işlemler ya da tekrarlar bulunmamalıdır.

Yazılım kesin olarak fare destekli olmalı, kullanıcı klavyeden yapılacak sayısal ya da alfabetik değer girişlerine duyarlı menülerden kurtarılmalı, fare ile ekran üstünden menü seçimi yapmaya yarayan, yerinde açılan pencerelerden oluşan,

"pop-up" ve yerinden aşağıya doğru açılan pencerelerden oluşan, "pull-down" menüler kullanılmalıdır.

Bu nedenle yazılımlar iyi bir grafik kullanıcı arayüzü olan, Windows ya da OS2 altında çalışarak, ekran üstünden fareyle menü seçmeye yarayan pencere mantığı, düğme menüsü gibi grafik kullanıcı arayüzlerinden yararlanabilmelidir.

Özellikle veri tabanı uygulamalarında, bilginin çizgisel olarak ekran üstünden sunumu⁷⁵ önemlidir

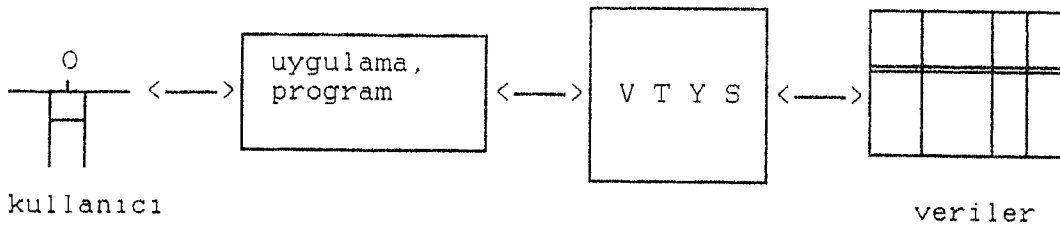
Yazılımlarda özellikle dikkat edilmesi gereken nokta, yazılımın yapısının tümleşik olmasıdır. Kullanıcı gereksinimlerini tek yazılımdan karşılayabilmeli, değişik işlemler için başka yazılımları yüklemek zorunda kalmamalıdır. Bu ise diğer yazılımların alt yordamlar biçiminde ana yazılımın yapısında tümleşik olarak bulunmasıyla olur. Yordamlar arası geçişler kolay, dolaysız ve çabuk olmalıdır.

1.6 Veri tabanı Yaklaşımının Getirdikleri

Ataştırmanın şimdiye kadarki kısmında ele alınan kavramların çözümlenmesinden, şu sonuçlara ulaşılabilir:

1. Veri tabanı örgütlü bir veri dermesini içeren bir bilgisayar dosyasıdır.
2. Veri tabanı yönetim sistemi, veri tabanı dosyalarını yönetmeye yarayan, genel amaçlı ticari yazılımdır.
3. Veri tabanı uygulama yazılımları, veri tabanı yönetim sistemlerinde kimi özel uygulamalarda sıkça tekrarlanan yordamları oluşturan komutların kullanıcı tarafından belirlenen üst başlıklar altında bir araya getirilmesinden oluşan yazılımlardır.
4. Veri tabanı uygulama yazılımları, veri tabanı yönetim sistemlerinde, kimi özel uygulamalarda sıkça tekrarlanan yordamları oluşturan komutların, kullanıcı tarafından belirlenen üst başlıklar altında bir araya getirilmesinden oluşan yazılımlardır.

Bu kavramlar arasındaki ilişkileri Sauer aşağıki çizimle özetlemeye çalışmıştır⁷⁶:



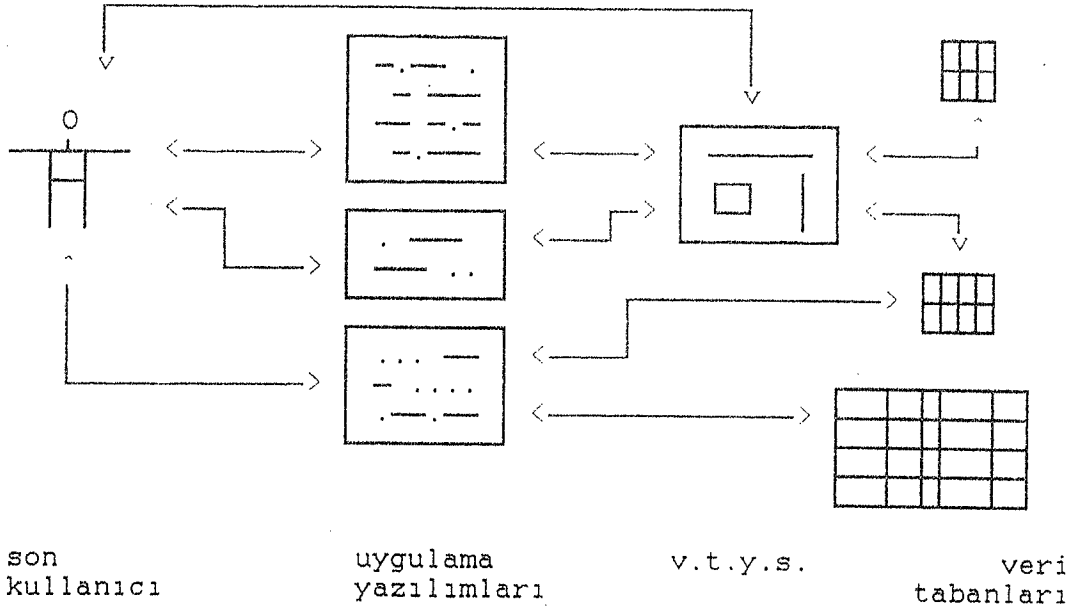
ÇİZ. 10. SAUER şeması

Ancak görüleceği gibi bu çizimi kimi yönlerden eksiktir. Öncelikle veri tabanı yönetim sistemi, tek bir veri tabanı dosyasını işliyor görünmektedir, oysa bir veri tabanı yönetim sistemi, birden fazla veri tabanı dosyasını işleyebilir. Ayrıca veri tabanı yönetim sistemini, tek bir uygulama yazılımı yönlendiriyor gözükmekte, oysa bir veri tabanı yönetim sistemi, değişik uygulama yazılımları tarafından yönlendirilebilir.

Bunların yanı sıra, çizimden kullanıcı, ki buna son kullanıcı demek daha doğru olur, veri tabanı yönetim sistemini uygulama yazılımı aracılığıyla, önceden belirlenmiş kalıplara göre, işletebiliyor izlenimi doğmaktadır. Ancak son kullanıcı hem uygulama yazılımlarını kullanabilir, hem de doğrudan doğruya veri tabanı yönetim sisteminde anlık işlemler gerçekleştirebilir.

Bunun dışında uygulama yazılımı yalnızca veri tabanı yönetim sisteminin bir yorumlayıcı birimi kullanıyor gözükmektedir, oysa son kullanıcı veri tabanı dosyalarını, doğrudan doğruya derlenmiş, bağımsız bir yazılım olan uygulama yazılımından da yönetebilir.

Buna göre Sauer'in çizimini şu biçimde geliştirmek daha akılcıdır:



ÇİZ. 11. Genişletilmiş SAUER şeması

Veri tabanı yaklaşımının getirdiği başlıca yenilikler özetle şöyle sıralanabilir:

1. Gereksiz tekrarlar önlenmiş. bellek tasarrufu sağlanmıştır.
2. Değişik veri tabanı dosyalarına veri tabanı yönetim sisteminden merkezi erişim sağlanmıştır.
3. Merkezi denetim sonucu veri güvenliği sağlanmıştır.
4. Veri tabanı yönetim sisteminden merkezi erişim sağlanarak, basit işlemler için, yazılım geliştirme gereği ortadan kalkmıştır.

5. Veri tabanı yapısı nedeniyle, erişim hızı artmıştır.
6. Veri tek dosya yapısına yüklendiğinden verilerin arasındaki tutarsızlıklar tamamen ortadan kalkmıştır.
7. Veri dosyalarının yapısında standartlaşma sağlandığından. Özellikle kütüphaneler gibi kuruluşların arasında, veri alışverişi kolaylaşmış, birbirlerinden bağımsız sistemler telefon hattıyla birbirlerine veri aktarmanın ötesinde birbirlerinin veri tabanlarına erişerek. hatta onlarla işlem yaparak, zaman ve veri taşıma masraflarından büyük kazançlar sağlanmıştır.

2 Veri Tabanı Yaklaşımının Kütüphanelerde Uygulanması ve bir Çözüm Önerisi

2.1 Veri Tabanı Yaklaşımının Kütüphanelere Getirdiği Yenilikler

2.1.1 Tümlşik Yapı

Gerek arşivlerin, gerek kütüphanelerin gerekse belge-bilgi merkezlerinin örgütlenme yapıları sistemattir. Bu sistemi, birbirleriyle tümlşik olarak işleyen alt sistemler oluşturur: sağlama, kataloglama, yararlanılma v.b. Bu nedenle, veri tabanı yönetim sistemleri. ve bu sistemlerde geliştirilecek uygulama yazılımlarının, yukarda da ele alındığı gibi kullanıcılarına sunduğu tümlşik yaklaşım, sözkonusu kuruluşlar için yaşamsal önem taşır. Hele ki Türkiye gibi otomasyon konusunda systemsizliklerin yaşandığı gelişme ve değışme sancılarını yaşayan, üçüncü dünya ülkelerinde.

Veri tabanı uygulaması ile geliştirilecek olan bir sağlama katalogu. veri tabanı sağlama yordamına ait veri olduğu. gibi asıl katalog veri tabanına aktarılabilir. bu veri tabanı okur tarafından sorgulanarak. bibliyografik tarama yapılabilir.

Bu kolaylık söz konusu kuruluslara işlerini tümleşik bir yapı içinde, eşgüdümlü bir tarzda yapmaları olanağını verecektir.

2.1.2 Standardizasyon ve İşbirliği Olanakları ve MARC Yaklaşımının Değişimi

MARC yaklaşımının ortaya atıldığı yıllarda her kütüphane kendi kayıtlarını kendi özgün veri yapılarında tutmaktadır. Bu kayıtların diğer kütüphaneler tarafından kullanılabilirmeleri için hepsinin ortak bir iletişim ya da değişim biçimine oturtulması gerekmektedir. MARC bu gereksinime yanıt vermek için tasarlanmış bir ortak iletişim formatıdır.

Yaklaşım göre katalog verisi, bulunduğu özgün veri yapısından MARC yapısına çevrilir, bu yapı taşınabilir manyetik bir ortama yüklenir. bu ortam diğer kütüphaneye çoğunlukla postayla gönderilir. diğer kütüphane bu mantetik ortamda MARC yapısında kayıtlı bulunan veriyi, kendi veri yapısına çevirerek kullanabilirdi. Gelişen bilgisayar ağıları ile birlikte ulaştırma sorununun ortadan kalkmasına karşın, çevirme işlemleri, başlı başına sorun ve zorluk olmaya devam etmişlerdir.

MARC yaklaşımı, genel olarak iki açıdan ele alınabilir.

1. Tutanakların fiziksel yapısı
2. Tutanakların mantıksal yapısı

Tutanakların fiziksel yapısı, tutanağın manyetik ortam üstünde alacağı biçimdir. Bu biçim, aslında sistemin tutanağa fiziksel erişiminin türünü de belirler. Erişim üç türlü olabilir:

- a) Tutanakların, aralarına ayraçlar konarak, bir dizi halinde ard arda sıralanmalarını öngören, diğerlerine göre bellek alanından kazandırmasına karşın çok daha yavaş gerçekleşen, Sıralı Erişim
- b) Tutanakların yerlerinin önceden belirlenerek, diğerlerine göre hızlı gerçekleşen, Rastgele Erişim
- c) Tutanakların dizinlenerek dizilmesine yarayan, ancak genede yavaş işleyen, Dizini Erişim

Sistemin tutanaklara nasıl erişim sağlayacağı tutanakların bulunduğu dosyanın yapısına bağlıdır. Buna göre sıralı erişimi kullanan dosyaya, Sıralı Dosya, rastgele erişimi kullanan dosyaya Rastgele Erişimli Dosya, dizinli erişimi kullanan dosyaya da Dizimli Dosya denir.

MARC yaklaşımı bu erişim türlerinden özellikle sıralı erişime uygundur⁷⁷. Bu nedenle de bellek alanından kazandırmasına karşın çok daha yavaş erişim sağlanır. Oysa veri tabanı yaklaşımı, doğrudan doğruya rastgele erişimi kullanır; bu nedenle de, daha fazla bellek alanına gereksinim duymasına karşın, erişimi çok daha hızlı sağlar.

Veri tabanı yaklaşımıyla gündeme gelen dağıtık veri yönetimi uygulamalarının sağladığı önemli bir gelişme. kurulan ağ yapıları ile farklı bilgisayarlardaki farklı veri tabanlarına erişimdir. Bu sayede sistemler başka sistemlere ait veri tabanlarını sorgulayabilmekte ve onların üstünde işlemler yapabilmektedirler. Bu ise makinaca okunabilir bibliyografik verinin bibliyografik kuruluşlar arası değişimini amaçlayan⁷⁸ MARC biçiminin fiziksel tutanak yapısına gereksinimi teorik olarak ortadan kaldıracaktır. Bu nedenledir ki, veri tabanı yaklaşımını benimseyen bir çok kütüphane MARC biçimindeki tutanaklarının fiziksel yapılarını kendi veri tabanlarının yapısına çevirmektedirler⁷⁹. Örneğin A.B.D.'de az sayıda kütüphane otomasyonu sistemi USMARC'ı

77 Walt Crawford. MARC for library use : understanding the USMARC formats. — White Plains, NY: London : Knowledge Industry Publications. 1984. — s. 157

78 Alan Hopkinson. UNIMARC handbook. — London : IFLA International Office for UBC. 1983. — s. 1

79 Crawford...

kullanırken, bir çoğu MARC-uyumlu sistemler kullanmaktadırlar⁸⁰.

MARC-uyumlu sistemler ise, aslında MARC yaklaşımının fiziksel tutanak yapısını değil, mantıksal tutanak yapısını kullanan sistemlerdir.

MARC yaklaşımının mantıksal tutanak yapısı ise alan kodlarından ve alt alan göstergelerinden oluşur⁸¹. Bunlar veri tabanlarında yüklem değerlerini tanımlayan değerler olarak işlenebilir, ve tutanağın mantıksal yapısı böylelikle sağlanmış olur.

2.1.3 Kütüphanecinin Değişen Konumu

Kütüphanelerde veri tabanı yaklaşımına dayalı uygulamaların artması ile, ki genel eğilim artacağı yönünde, özellikle yönetici konumundaki kütüphaneci, yakın gelecekte veri tabanı yöneticisi olmaya adaydır.

Bu nedenle, geleceğin kütüphanecileri öncelikle kütüphane sisteminin tümleşik yapısını özümsemek

80 Crawford... — s. 151

81 Hopkins... — s. 14-262

zorundadırlar. Buna kosut olarak, ileri bir veri tabanı yönetim sistemi bilgisi de temel sartlardan biri olacaktır. Özellikle kendi uygulamasını geliştirmek isteyen kütüphaneciler, veri tabanı geliştirme sürecini ayrıntılı bir sekilde tanımak ve bunu gerçekleştirebilmek zorundadırlar.

2.2 Kütüphanelerde Veri Tabanı Yaklaşımına Dayalı Uygulamalar için Paradox for Windows Veri Tabanı Yönetim Sistemi Altında Geliştirilebilecek Bir Çözüm Önerisi

2.2.1 Paradox for Windows'a Dayalı Çözümlerin Uygulanabilirliği

2.2.1.1 Veri Tabanı Yapısı Ve Tasarım

Sistem, özellikle tasarım konusunda, sanki kütüphaneler için tasarlanmış gibidir.

Bilindiği gibi kataloglar, Anglo Amerikan Kataloglama Kuralları 2 (1988) gereğince, zorunlu, seçmeli, ya da isteğe bağlı⁸² alanlardan oluşurlar. Kurallar gereği, alanlar, her zaman hepsi birden bir arada bulunmayabile alt alanları içerirler. Bu alanların ise uzunlukları deęiskendir.

82 Hasan S. Keseroğlu, Aynur Dursun, "Kataloglama politikası" Kütüphanecilik dergisi : belge bilgi kütüphane araştırmaları. — s.3 (1993). — İstanbul Üniversitesi Edebiyat Fakültesi, 1992. — s. 136

Diğer veri tabanı yönetim sistemlerinde alfa numerik alanların uzunlukları önceden belirlenen değerde sabit kalırken, Paradox for Windows bu sorunu "Memo" adlı değişken uzunluklu alfanumerik alan türüyle çözmüştür⁸³. Memo alanlarının uzunlukları, eğer dosyanın içinde saklanacaklarsa 1-240 karakter arasındadır. Eğer dosya dışında saklanacaklarsa, bu sayı sadece manyetik bellek ünitesindeki, yani kişisel bilgisayarların hard disk'lerindeki bellek alanı ile sınırlanabilmektedir. Kullanıcı sistem tarafından kurulan mantıksal ilişkiler sonucunda alan dosya dışında saklı olsa bile bunu fark etmeyecek ve sistemin ilişkiselliği bozulmayacaktır.

Bunun yanı sıra MEMO alanlarına sabit bir uzunluk verilerek, o uzunluk kadarı dosya içinde kalanı dosya dışında saklanarak, hem diğer sistemlerle kıyaslanamayacak kadar bellek alanı, hem de böylelikle erişim süresi azaltılarak zaman kazanılır⁸⁴.

Böylelikle katalog verisinin değişken uzunluk sorunu da kendiliğinden ortadan kalkmaktadır.

Gene Paradox for Windows'a özgü bir alan türü olan "OLE" (Object Linking and Embedding) alanında baslı başına

83 Paradox for Windows. User's guide. — Scotts Valley, California : Borland International, 1992. — s. 15.

84 Paradox for Windows. User's guide... — s. 289

görsütsel belgeler (resim, çizim, fotoraf v.b.) saklanabilmektedir⁸⁵. Bu ise belge hizmeti veren arşivlere, özel kütüphanelere v.b. kuruluşlara veri tabanında oluşturulmuş katalogda yapılacak tarama sonucunda istenen belgenin görüntüsünü ekrana yansıtabilme olanağını verecektir.

2.2.1.2 İşletim ve Denetim

Windows'a dayalı grafik kullanıcı arayüzü sayesinde Paradox for Windows'da veri tabanı işletimi yapmak son derece basitleşmiştir. Paradox for Windows'un kullanıcı tanımlı veri girişi "form"ları özellikle bir takım rutin işlevlerin yüklenebileceği düğmelerle donatılabilmektedir⁸⁶. bu ise veri tabanı yönetim sisteminden uygulama yazılımına geçmeden basit işlemlerin yapılmasını inanılmaz kadar kolaylaştırabilmektedir.

Sistem belli bir "form"dan birden çok veri tabanına giriş yapmaya olanak tanımakta. böylelikle kullanıcı, veri tabanı değişik dosyalara pratik nedenlerle bölünmüş olsa bile bunları bir arada tek veri tabanında işletim yapar gibi işletebilecektir.

85 Paradox for Windows. User's guide... — s. 16

86 Paradox for Windows. User's guide... — s. 431

Bu tümleşik yapı sayesinde kütüphaneci katalog betimlemesi ve sınıflama gibi değişik işleri aynı ekrandan yapabilme olanağına kavuşmuştur.

Katalog verileri kullanıcı verileri ile ilişkilendirilerek, ödünç verme işlemleri yönetilebilecek, hem de bu veriler başka yönetsel veriler ile ilişkilendirilerek, istatistik, bütçe işlemleri de gerçekleştirilebilecektir.

2.2.1.3 Sorgulama ve Raporlama

Sistemin en can alıcı işlevi olan sorgulama hem güçlü sorgulama parametreleri⁸⁷ ile hem de QBE (query by example = örneğe dayalı sorgu) yaklaşımı⁸⁸ ile bibliyografik veri tabanında en karmaşık sorgulara herhangi bir uygulama yazılımına gerek kalmadan daha veri tabanı yönetim sistemindeyken çok esnek bir tarzda olanak vermektedir.

Raporlamada kütüphanecilik açısından en büyük kazanım sistemin ASCII dizilimini kullanan çoğu sistemin aksine Türkçe alfabetik dizilimi desteklemesi⁸⁹ ve raporlarını bu doğrultuda alfabetik olarak üretebilmesidir. Bunun yanı sıra

87 Paradox for Windows. User's guide... — s. 219-221

88 Paradox for Windows. User's guide... — s. 148

89 Nadir Çolak. "Nedir bu Paradox Engine?" Byte Türkiye. — c. 1. s. 4 (Mayıs 1994). — s. 148

raporlar bir çok font ve sayfa düzeni seçeneğine sahiptir⁹⁰.
Bu ise sisteme gerekli donanım eşliğinde katalog kartından
bibliyografyaya kadar bir çok çıktı olanağını sağlamaktadır.

Sistem bunun yanı sıra manyetik ortamda saklı bulunan
çeşitli belge görüntülerinden de kullanıcıya çıktı olanağı
vermektedir.

2.2.2 Çözüm Önerisi : B.K.V.

B.K.V., yani Bölünmüş Katalog Veri Tabanı Yapısı
özellikle Paradox For Windows veri tabanı yönetim sistemi
için tasarlanmış bir katalog veri tabanı önerisidir.

Katalog yapısı için Anglo-Anmerikan Kataloglama
Kuralları 2'de⁹¹ ele alınan bibliyografik betimleme
kuralları temel alınmıştır, yani yapı Anglo-Amerikan
standardına uygundur.

Veri tabanının değişik veri tabanı dosyalarına bölünmüş
olmasının altında pratik bir düşünce yatmaktadır: ilişkisel
yapıdan kaynaklanan aksaklıkları ortadan kaldırmak.

90 Paradox for Windows. User's guide... — s. 467-534

91 Anglo-American cataloguing rules / prep. under the dir.
of the Joint Steering Committee for Revision of AACR :
ed. by Michael Gorman and paul Winkler. — 2nd ed., 1988
rev. — Chicago : Ottawa : London : ALA : CLA : LA. c
1988. 1989

Bilindiği gibi kütüphane katalogları çok karmaşık veri tabanlarıdır. Bunun nedeni her alan için sayısı belli olan ancak kaçının ya da hangilerinin kullanılacağı hiçbir zaman için belli olmayan değişik alt alanların varlığıdır (kimi zaman da yokluğu). Bu alanları ilişkisel yapının tutanaklarında doğrusal olarak birbirleri ardına başlı başına alanlarmış gibi sıralamak olasıdır. Ancak bu sefer de kullanılamayan alanlar gereksiz bellek alanı kullanımı gibi sorunlar doğuracaklardır.

Buna bulunan çözüm basittir. Her alan için başlı başına bir veri tabanı dosyası açılmıştır. Bu dosyaya tüm alt alanlar başlı başına farklı tutanaklarda girilir. Tutanak yapısı içinde alt alan adı alanının yanı sıra bir anahtar alanı, bir de MARC alan kodlar ve altalan göstergelerini içeren yönlendirme alanı eklenmiştir: bunlara bazı yerlerde gerek duyulan ek alanlar da eklenmiştir, ki bununla da kimi yerde veri tabanının kendi yapısı içinde yetke dizinlerinin oluşması sağlanmıştır.

Anahtar alan bir yerde kütüphanelerde kullanılan demirbaş numarasıdır. Bu numaranın özelliği kataloglanan kitabı temsil etmesidir. Anahtar alanın değeri kataloglanan bir kitap için bütün veri tabanlarında aynı olduğundan bu numara üstünden bağlantı kurulacaktır. Yapı ilişkisel temele dayanmakla beraber birincil ya da ikincil anahtar alanlarının olmaması ilişkisel yapıyı ortadan kaldırmakta anahtar olarak

kullanılan öge doğrudan doğruya kataloglanan belgenin nesnesine yönelik olduğundan nesne yönelimlilik benzeri bir yapı ortaya çıkmaktadır. Bu yapı çerçevesinde anahtar alanın değeri dolaşım dosyasında da kullanıcı numarası ile ilişkilendirilerek hangi belgenin hangi okurda olduğu bu ilişki ile saptanabilecektir.

Yönlendirme alanında yer alacak değer doğrudan doğruya alan kodunun ya da alt alan göstergesinin sayısal karşılığı olacaktır.

Kütüphaneci bu arada bütün bu karmaşık yapıdan uzak kalarak kendi iş akışına göre belirlenmiş bir ekran "form"undan girişlerini yapacak. geliştirilecek uygulama yazılımı kataloglanan her kitaba bir anahtar numarası vererek ve bütün alt alanları gerekli veri tabanı dosyalarına bu anahtar numarası hangi alt alan olduğunu belirleyen yönlendirme kodu ile dağıtacaktır.

Sorgulmalarda Paradox for Windows'un güçlü sorgulama seçenekleri yardımıyla uygulama yazılımı, belli bir sorgulama "form"u ekranından girilen bir ya da bir den fazla sorgulama ögesini denk düştükleri veri tabanından tarayacak. yanıt olarak aldığı anahtar numaraları üstünden QBE yöntemiyle bulunduğu diğer alt alanları ve alanları anahtar numarası aynı olanları yönlendirme koduna göre ekranda geleneksel katalog kartı görüntüsü altında ya da tercih edilebilecek başka bir

biçimde bir araya getirecektir. Aynı yöntem raporlama çıktılarında da kullanılabilecektir.

Bu aşamada her veri tabanını dosyasını tek tek ele almak gereklidir:

B.K.V 12 ayrı veri tabanı dosyasından oluşur bunlar kişi adlarının bulunduğu KISI.DB, yayın adlarının bulunduğu YADI.DB, konu başlıklarının bulunduğu KONU.DB, basım kaydına ilişkin cerilerin bulunduğu BAS.DB, belgenin doğasına ilişkin verinin bulunduğu BELGE.DB, sınıflama kodlarının bulunduğu SINIF.DB, dizi kayıtlarının bulunduğu DIZI.DB, yayın ya da baskı yerlerinin bulunduğu YER.DB, yayıncı ya basımevinin bulunduğu YABA.DB, yayın ya da baskı tarihleini içeren TARİH.DB, notları içeren NOT.DB, ve ISBN ya da ISSN numaralarını içeren STAND.DB, veri tabanı dosyalarıdır.

KISI.DB

<u>Alan Adı</u>	<u>Türü</u>
ANAHTAR	N
YÖNLENDİRME	N
İŞLEV	MEMO
KİSİ ADI	MEMO
SAN	MEMO
DENETİM	N
TEKBİÇİM ADI	MEMO

Görüldüğü gibi, ANAHTAR adlı alan sayısal bir alandır. Bunun nedeni yazılımın kataloglardan kayıt eksiltilmesi durumunda bütün numaraları düzeltebilmesi ve yeni numaraları da basitçe sayarak kendi kendine verebilmesi içindir.

YÖNLENDİRME adlı sayısal alana önceden belirlenmiş sayısal kodları yazılımın kataloglama yapılırken alt alanlara göre kaydederek gerekli hallerde gene değerlendirebilecektir. Burada özellikle belirtmek gerekir ki kişi adları, sorumlu adı, emeğegeçen adı, basım sorumlusu adı v.b. alt alanlara ait olabilirler; hatta notlar alanından buraya gönderme bile yapılabilir. Buna göre yönlendirme kodu önceden Anglo-Amerikan kurallarının ilgili maddelerine göre belirlenir ve yazılımda saptanır.

İŞLEV alanı sorumluluk alanında yer alan işlev "yayına hazırlayan", "resimleyen" . "derleyen" v.b. türü bildirimlerin kısaltmalarını içerir. Uzunluğu belirsiz olduğundan "memo" türü bir alan olarak belirlenmiştir.

KİŞİ ADI alanı değişken uzunluklu bir alandır. Bu alan doğrudan doğruya kişi adının gövdede bulunan biçimini içerir.

SAN alanında ilgili kural gereği kişi adı İngiliz soyluluk sanı taşıyorsa bu içerilir.

DENETİM alanında kişi adının tüzel kişiye mi ait yoksa kişiye mi ait olduğu belirlenir.

TEKBIÇİM ADI alanı kişi adının özellikle temel giriş için alacağı biçimi içerir. Bu alan ile kişi adları için kendiliğinden bir yetke dizini de oluşur.

YADI.DB

<u>Alan Adı</u>	<u>Türü</u>
ANAHTAR	N
YADI	MEMO
YÖNLENDİRME	N
TEKBIÇİM ADI	MEMO

BASLIK değişken alanı ilgili kataloglama kuralı gereği başlık, bilgi başlığı, koşut başlık v.b. alt alanları içerir.

ANAHTAR sayısal alanı diğer veri tabanları ile bağlantıyı sağlar, YÖNLENDİRME alanı alt alanın türünü belirler.

KONU.DB

<u>Alan Adı</u>	<u>Türü</u>
ANAHTAR	N
KONU BAŞLIĞI	MEMO

KONU BAŞLIĞI değişken alanı konu başlığını içerir.

ANAHTAR sayısal alanı diğer veri tabanları ile bağlantıyı sağlar.

BELGE.DB

<u>Alan Adı</u>	<u>Türü</u>
ANAHTAR	N
GBB	MEMO
ÖZEL AYRINTI	MEMO
UZUNLUK	MEMO
FİZİKSEL ÖZELLİKLER	MEMO
BOYUTLAR	MEMO
BİRLİKTEKİ BELGELER	MEMO

ANAHTAR sayısal alanı diğer veri tabanları ile bağlantıyı sağlar.

Diğer değişken uzunluklu alanlar Kataloqlama Kuralları'nın ilgili maddeleri gereği belgenin doğasına iliski veri içerirler.

SINIF.DB

<u>Alan Adı</u>	<u>Türü</u>
ANAHTAR	N
YER #	MEMO
SINIFLAMA #	MEMO

ANAHTAR sayısal alanı diğer veri tabanları ile bağlantıyı sağlar.

YER # değişken uzunluklu alanı D.D.C. L.C.C. v.b. yerleştirme sistemi kodlarını içerir.

SINIFLAMA # değişken uzunluklu kodları U.D.C.. I.T.C. v.b. sınıflama sistemi kodlarını içerirler.

DIZI.DB

<u>Alan Adı</u>	<u>Türü</u>
ANAHTAR	N
DIZI ADI	MEMO
YÖNLENDİRME	MEMO
TEKBİÇİM DIZI	MEMO

ANAHTAR sayısal alanı diğer veri tabanları ile bağlantıyı sağlar. YÖNLENDİRME alanı alt alanın türünü belirler.

DIZI ADI değişken uzunluklu alanı dizi alanlarını içerir.

TEKBİÇİM DIZI değişken uzunluklu alanında zaman içinde değişiklik gösterebilecek dizi adları bir tekbiçimde toplanır. Böylelikle de veri tabanının yapısında kendiliğinden bir yetke dizini oluşur.

YER.DB

<u>Alan Adı</u>	<u>Türü</u>
ANAHTAR	N
YÖNLENDİRME	N
YER	MEMO
ÜLKE	MEMO

ANAHTAR sayısal alanı diğer veri tabanları ile bağlantıyı sağlar. YÖNLENDİRME alanı alt alanın türünü belirler.

YER değişken uzunluklu alanına belgenin yayın ya da basım yeri/yerleri alt alanları bulunur.

ÜLKE değişken uzunluklu alanına belgenin yayınlandığı ülke girilir. Bu alanın ilgili Kataloqlama Kuralında yeri vardır, ve ülke adına göre yapılacak taramalar için gerekli görülmüştür.

YABA.DB

<u>Alan Adı</u>	<u>Türü</u>
ANAHTAR	N
YÖNLENDİRME	N
İŞLEV	MEMO
YAYBAS	MEMO
TEKBİÇİM	MEMO

ANAHTAR sayısal alanı diğer veri tabanları ile bağlantıyı sağlar, YÖNLENDİRME alanı alt alanın türünü belirler.

İŞLEV sayısal alanı Kataloglama Kuralları gereği olup kuruluşun işlevini belirler.

YAYBAS değişken uzunluklu alanı yayıncının ya da basımevinin adını alt alanlarını içerir. Bu ad TEKBİÇİM adlı değişken uzunluklu alanı zaman içindeki değişiklikler göz önüne alarak bu adı bir tekbiçime oturtur.

TARİH.DB

<u>Alan Adı</u>	<u>Türü</u>
ANAHTAR	N
YÖNLENDİRME	N
BASLAMA	MEMO
SON	MEMO
CPRIGHT	N

ANAHTAR sayısal alanı diğer veri tabanları ile bağlantıyı sağlar, YÖNLENDİRME alanı alt alanın türünü belirler.

BASLAMA değişken uzunluklu alanı özellikle süreli yayınlar için yayına başlama tarihinin girileceği alt alana yöneliktir. SON değişken uzunluklu alanı gene süreli yayınlar için yayın bitiş tarihinin girileceği alt alana yöneliktir. Dizi yayın olmayalar BASLAMA alanına girilebilirler.

CPRIGHT alanında girilen tarihin copyright tarihi olup olmadığı belirlenir.

NOT.DB

<u>Alan Adı</u>	<u>Türü</u>
ANAHTAR	N
YÖNLENDİRME	N
NOT	MEMO

ANAHTAR sayısal alanı diğer veri tabanları ile bağlantıyı sağlar. YÖNLENDİRME alanı alt alanın türünü belirler.

NOT değişken uzunluklu alanına notlar alanının 22 değişik alt alanı girilir.

STAND.DB

<u>Alan Adı</u>	<u>Türü</u>
ANAHTAR	N
STANDART #	MEMO

ANAHTAR sayısal alanı diğer veri tabanları ile bağlantıyı sağlar.

STANDART # alanında ISBN numarası ve ISSN numarası belirtilir.

Günümüzde gerek kütüphanecilik gerekse bilgisayar konulu yayınlarda veri tabanı kavramı çoğunlukla kendisinden başka bir kavram olan veri tabanı yönetim sistemi kavramının yerine, sanki onun bir kısaltmasıymış gibi, yanlış olarak kullanılmaktadır. Oysa veri tabanı kavramı, aşağıda da ele alınacağı gibi, özellikle bilgisayar ortamındaki uygulamalarda, verilerin yüklü bulunduğu örgütlü teknik düzeneği (kütüphanecilik bakış açısıyla bir belgeyi) belirtirken, veri tabanı yönetim sistemi kavramı bu düzenek ile işlem yapmaya, onu söz konusu örgütlenme içinde yönetmeye yarayan bir tür uygulama yazılımını belirtmektedir.

Veri tabanı uygulama yazılımları veri tabanı yönetim sistemlerinde kimi özel uygulamalarda sıkça tekrarlanan yordamları oluşturan komutların kullanıcı tarafından belirlenen üst başlıklar altında bir araya getirilmesinden oluşan yazılımlardır.

Veri tabanı yaklaşımının getirdiği başlıca yenilikler özetle şöyle sıralanabilir:

1. Gereksiz tekrarlar önlenmiş. bellek tasarrufu sağlanmıştır.
2. Değişik veri tabanı dosyalarına veri tabanı yönetim sisteminden merkezi erişim sağlanmıştır.

3. Merkezi denetim sonucu veri güvenliği sağlanmıştır.
4. Veri tabanı yönetim sisteminden merkezi erişim sağlanarak basit işlemler için yazılım geliştirme gereği ortadan kalkmıştır.
5. Veri tabanı yapısı nedeniyle erişim hızı artmıştır.
6. Veri tek dosya yapısına yüklendiğinden verilerin arasındaki tutarsızlıklar tamamen ortadan kalkmıştır.
7. Veri dosyalarının yapısında standartlaşma sağlandığından özellikle kütüphaneler gibi kuruluşların arasında veri alışverişi kolaylaşmış, birbirlerinden bağımsız sistemler telefon hattıyla birbirlerine veri aktarmanın ötesinde birbirlerinin veri tabanlarına erişerek, hatta onlarla işlem yaparak, zaman ve veri taşıma masraflarından büyük kazançlar sağlanmıştır.

Veri tabanı geliştirmesi şu beş alt süreçten oluşan bir süreçtir:

1. Veri araştırması
2. Veri örnekleme
3. Veri tabanı tasarımı
4. Veri tabanı işletimi
5. Veri tabanı ayarlaması

Veri tabanları örgütlenme yapıları açısından zaman içinde farklılıklar göstermişler ve böylece de veri tabanlarının değişik türleri ortaya çıkmıştır. Ancak tarihsel açıdan konuya yaklaşıldığında, yaklaşım olarak veri tabanı düşüncesinin değişmediği, aksine değişimin yapısal nitelikli olduğu görülür. Bu yüzden ki yeni bir tür ortaya çıktığında eskisi etkisini yitirmektedir.

Veri tabanlarında yaşanan bu yapısal değişikliklerin başlıca nedeni veri tabanının yapısının giderek gerçekliğin bir örneğinin bilgisayar ortamındaki yansıması olmaktan, uygulama amacına yönelik olarak yeniden biçimlendirilmiş bir yapıya yönelmesidir. Örneğin ilk türlerden olan hiyerarsik veri tabanlarında veri tabanı yapısı uygulama (özellikle de sorgulama) kaygısı taşımaksızın, gerçekliğin felsefi anlamda tamamen koşut bir yansımasıdır. Ağ yapılı veri tabanlarında bu yapı biraz daha uygulamaya yönelik bir biçime sokulmaya çalışılmış, ancak elde edilen hiyerarsik veri tabanının yapısından biraz daha büyük bir karmaşa olmuştur.

Veri tabanı dünyasında ilk devrim niteliğini taşıyan gelişme yapı bakımından diğerlerinden oldukça yalın, özellikle sorgulamalarda büyük kolaylıklar sağlayan ilişkisel veri tabanlarının ortaya çıkmasıdır.

İlişkisel veri tabanlarındaki kimi aksaklıklar ve kısıtlamaların giderilmeye çalışılmasıyla da özellikle 90'lı

yıllara damgasını vuracak gibi görünen nesne yönelimli veri tabanları ortaya çıkmıştır.

Kütüphane alanında veri tabanlarının getirdiği yenilikler, bilgisayara dayalı uygulamaların tümleşik bir yapıya kavuşması, veri tabanlarının işlevleri nedeniyle MARC biçiminin fiziksel tutanak yapısına ihtiyaç duyulmaması, MARC biçiminin mantıksal tutanak yapısını kullanan, ve bu nedenle MARC-uyumlu olarak adlandırılan sistemlerin ortaya çıkması ve kütüphanecinin zamanla bir veri tabanı yöneticisi konumuna geleceği eğilimidir.

Paradox for Windows adlı veri tabanı yönetim sistemi özellikle kütüphanelerde gerçekleştirilebilecek uygulamalar için sağladığı geniş olanaklar, kolay kullanımı ve güçlü yapısı ile çözüm olanakları sağlayacak bir sistemdir.

Bölünmüş Katalog Veri Tabanı Yapısı adlı çözüm özellikle Paradox for Windows altındaki uygulamalar için geliştirilmiş, sistemin olanaklarını kullanıcının en üst düzeyde hizmetine sokan bir çözümdür.

KAYNAKLAR

American heritage dictionary. — 2nd college ed. — Boston :
Houghton Mifflin Company, c. 1982

Anglo-American cataloguing rules / prep. under the dir. of
the Joint Steering Committee for Revision of AACR ; ed. by
Michael Gorman and paul Winkler. — 2nd ed., 1988 rev. —
Chicago : Ottawa ; London : ALA ; CLA ; LA. c. 1988, 1989

Brackett, Michael H. Developing data structured databases /
Michael H. Brackett. — Englewood Cliffs, New Jersey, 1987

Codd, Edgar F. The relational model for databse management :
version 2 / E. F. Codd. — repr. with corr. — Reading,
Massachussets : Addison-Wesley, 1990

Crawford, Walt. MARC for library use : unserstanding the
USMARC formats / by Walt Crawford. — White Plains, NY :
London : Knowledge Industry Publications, 1984

Colak, Bahadır. Nedir bu Paradox Engine? / Bahadır Colak. —
s. 148-153. — Byte Türkiye. — c. 1. s. 4 (Mayıs 1994)

Date, C. J. Relational database writings, 1985-1989 / C J. Date; with a special cotribution by Andrew Warden. — repr with corr. — Reading, Massachussets : Addison-Wesley, 1990

Dbase 3 Plus. Inrtoduction to Dbase 3 Plus [bilgisayar dosyası]. — [y.y.] : Ashton-Tate, [1986?]

Hayes, Robert Mayo. Handbook of data processing for libraries / Robert M. Hayes. — 2nd ed. — Los Angeles : Melville Publishing Company, 1974

Heuer, Andreas. Objektorientierte datenbanken : Konzepte, Modelle, Systeme / Andreas Heuer. — Bonn... [ve öte.] : Addison-Wesley, 1992

Hopkinson, Alan. UNIMARC handbook / compiled and edited by Alan Hopkinson with the assistance of Sally H. McCallum and Stephen P. Davis. — London : IFLA International Office for UBC, 1983

Inside IBM's Distributed Data Management architecture / by R. A. Demers... [ve öte.]. — s. 459-487. — IBM Systems Journal. — vol 31. No. 3 (1992)

İlişkisel sistemler ve diğerleri. — s. 32-44 . — Bilgisayar . — (Temmuz 1993)

Keseoglu, Hasan S. Kataloglama Politikası / Hasan S. Keseoglu; Aynur Dursun. — s. [135]-174. — Kütüphanecilik dergisi : belge bilgi kütüphane araştırmaları. — s. 3 (1990)

Oxborrow, Elizabeth A. Databases and database control systems : concepts and issues / Elizabeth Oxborrow. — 2nd ed. — Bromley, Kent, England : Chartwell-Bratt, c. 1989

Paradox for Windows. Getting started. — Scotts Valley, California : Borland International, 1992

Paradox for Windows. Guide to ObjectPAL. — Scotts Valley, California : Borland International, 1992

Paradox for Windows. User's guide. — Scotts Valley, California : Borland International, 1992

Sauer, Hermann. Relationale Datenbanken : Theorie und Praxis, inklusive SQL-2 / Hermann Sauer. — 2. korrig. u. erw. Aufl. — Bonn... [ve öte.] : Addison-Wesley, 1992

Stein, Richard Marlon. Nesne veri tabanları / Richard Marlon Stein. — s. [80]-84. — Byte Türkiye. — c. 1. s. 3 (Nisan 1994)

Türkçe sözlük. — genişl. 7. bs. — [Ankara] : Türk Dil Kurumu, 1983

Udell, Jon. Componentware / Jon Udell. — s. 84-90. — Byte Türkiye. — c. 1. s. 4 (Mayıs 1994)

Wittgenstein, Ludwig. Tractatus logico-philosophicus = Logisch-philosophische Abhandlung / Ludwig Wittgenstein. — Frankfurt am Main : Suhrkamp Verlag, c. 1963, 1989

Yontar, Aysel. Bilgi bilime giriş [ders notları]. — [İstanbul] : [y.y.], [1992]. — Not : Metin teksirdir